

Optimization-Based Motion Planning for Autonomous Agricultural Vehicles Turning in Constrained Headlands

Chen Peng^{c,1}, Peng Wei^{a,1}, Zhenghao Fei^{c,*}, Yuankai Zhu^b, Stavros G. Vougioukas^a

^a*Department of Biological and Agricultural Engineering, University of California, Davis, CA, USA*

^b*Department of Mechanical and Aerospace Engineering, University of California, Davis, CA, USA*

^c*ZJU-Hangzhou Global Scientific and Technological Innovation Center, Hangzhou, Zhejiang, China*

一阶段通过经典的基于图案的转弯方法或方向图引导的混合A*算法
创建粗略路径，第二阶段通过数值优化器进一步优化路径，考虑车辆的运动学、控制和避碰限制，生成平滑轨迹

Abstract

Headland maneuvering is a crucial aspect of unmanned field operations for autonomous agricultural vehicles (AAVs). While motion planning for headland turning in open fields has been extensively studied and integrated into commercial auto-guidance systems, the existing methods primarily address scenarios with ample headland space and thus may not work in more constrained headland geometries. Commercial orchards often contain narrow and irregularly shaped headlands, which may include static obstacles, rendering the task of planning a smooth and collision-free turning trajectory difficult. To address this challenge, we propose an optimization-based motion planning algorithm for headland turning under geometrical constraints imposed by geometry and obstacles. Our method models the headland and the AAV using geometric primitives like convex polytopes, and calculates optimal and collision-free turning trajectories in two stages. In the first stage, we create a coarse path using either a classical pattern-based turning method or a directional graph-guided hybrid A* algorithm, depending on the complexity of the headland. The second stage refines this coarse path by feeding it into a numerical optimizer, which considers the vehicle's kinematic, control, and collision-avoidance constraints to produce a feasible and smooth trajectory. We demonstrate the effectiveness of our algorithm by comparing it to the classical pattern-based method in various types of headlands. The results show that our optimization-based planner outperforms the classical planner in generating collision-free turning trajectories within constrained headland spaces. Additionally, the generated trajectories from our planner are easier to track due to their smoothness, which respects the kinematic and control limits of the vehicle. In conclusion, our proposed approach successfully addresses the complex motion planning problems in constrained headlands, making it a valuable contribution to the autonomous operation of AAVs, particularly within orchards.

Keywords: Headland Turning, Trajectory Optimization, Autonomous Agriculture Vehicle, Constrained Headland

1. Introduction

Autonomous agricultural vehicles (AAVs), such as field robots and autonomous tractors, are gathering increased attention in modern farming operations. As research and investment in this technology continue to advance, the deployment of AAVs in the field is becoming more prevalent. A critical component of these autonomous operations is headland turning, which involves sharp turns performed by the AAVs at the end of the crop row. An optimized headland turning maneuver can enhance time and energy efficiency while ensuring operational safety. The motion planning for headland turning relies on a range of factors, including crop type, headland size and geometry, among others. The primary operating contexts for AAVs fall into two categories: open-field and orchard-based farming.

In open-field farming, crops are typically grown annually or, in some instances, multiple times per year. The cultivation layout can be easily modified within the field's boundaries at the beginning of each cultivation cycle to accommodate the

autonomous operation. Open fields usually provide sufficient space for AAVs to perform headland maneuvers, or alternatively, the cultivation layout can be adapted to facilitate these maneuvers. Therefore, research has mainly focused on optimal pre-planning of the cultivation layouts, coverage path planning (Jin and Tang, 2010; Spekken and de Bruin, 2013; Mier et al., 2023), and minimizing non-working distances for vehicles in headlands through global planning of optimal traversal sequences (Bochitis and Vougioukas, 2008). Due to the rela-

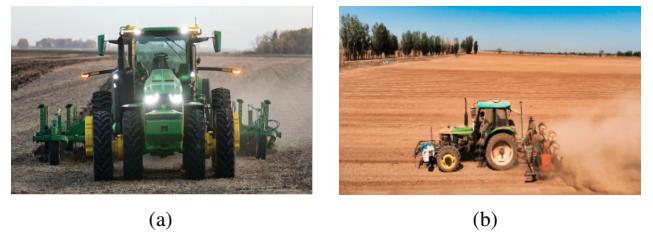


Figure 1: Left: John Deere's autonomous tractor operating in the field. Photo courtesy of Deere & Company. Right: CHCNAV's NX510 auto-steer retrofit kit that installed on a traditional tractor. Photo courtesy to CHC Navigation. .

*Corresponding author: Zhenghao Fei, zfei@zju.edu.cn

¹The authors contributed equally to this paper.

tively simple operational context and current technological advancements, Global Navigation Satellite System (GNSS)-based auto-steer systems for large vehicles, such as tractors, sprayers, and harvesters, are now commercially available and widely used in open-field farming. Additionally, headland turning maneuvers that do not take into account the obstacles and constrained spaces are also available for commercial auto-guidance products (Vougioukas, 2019). Figure 1(a) illustrates an example of these commercial auto-steer agricultural vehicles and Figure 1(b) shows a traditional tractor equipped with an auto-steer retrofit kit. Both of them can autonomously navigate pre-defined rows and execute a swath-to-swath turning in open headlands. To the best of our knowledge, these vehicles have nearly replaced human drivers’ driving work in open fields.

Unlike open fields, orchards may not always possess spacious, obstacle-free headlands, and expanding orchard headlands by replanting or removing plants can be costly, considering their high value and long lifespan. Most plants grown in orchards are perennial, and the lifespan of an orchard, such as an apple orchard, can range from 20 to 25 years (Lordan et al., 2018). Many existing orchards and crop fields have irregular and limited headland spaces, with some even containing obstacles within the headland. The reason is that these fields were not originally designed for autonomous operations, and orchard owners have sought to maximize land usage by providing only minimal headland space for human drivers. In some orchards, the headland space is so restricted that even human drivers can barely complete a turn, especially when the vehicle is equipped with some rigid-connected implements (e.g. Fig. 2(a)). Additionally, orchard boundaries may not always be quadrilateral. Instead, they may be common in irregular shapes, as shown in Fig. 2(b). The endpoints of tree rows may not be along a straight line and parallel to the boundary (see Fig. 2(c)). In some poorly designed orchards, obstacles such as electric poles, raised wells, and irrigation infrastructures may be present (see Fig. 2(d)), which pose further challenges for vehicle maneuvering in fixed patterns. These factors impede the deployment of autonomous operation systems within orchards, making it a complex task that demands careful and innovative solutions.

In this work, our primary focus is on tackling the challenging motion planning problem for AAV turning within constrained headlands, particularly in the real-world orchards described above. We summarized common challenges that AAVs may encounter in the constrained orchard headlands and presented them in Table 1. The first four challenges might not be problematic when there is sufficient headland space. However, when combined with the fifth challenge, they can render the motion planning problem very challenging. Moreover, even if an orchard has ideal headland conditions in most areas, a small percentage of less-than-ideal headland spaces can hinder the deployment of autonomous operations for AAVs. These challenges may also manifest in open-field farming where the cultivation layout is not optimized for autonomous navigation, especially when headland space is limited.

To the best of our knowledge, existing methodologies for AAV’s motion planning are insufficient for delivering effective turning maneuvers in constrained headlands. Thus, it is crucial

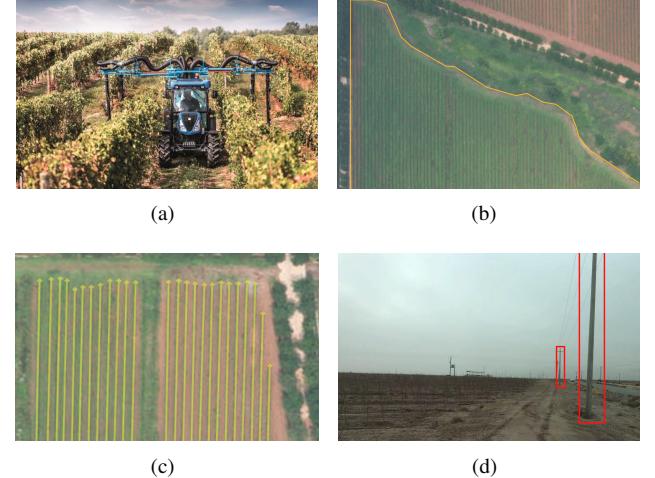


Figure 2: Example of (a) a tractor equipped with a mounted sprayer in a vineyard (photo courtesy to CNH Industrial Company); (b) irregular field boundary; (c) checkered crop row endpoints; (d) obstacles in headland space.

Table 1: Challenges that AAVs encounter in constrained orchard headlands.

Sources	Challenges
Tree row	1) The spacing between tree rows is relatively narrow compared to the vehicle’s width. 2) The endpoints of tree rows may not be collinear. 3) The vehicle must not run over existing crops in the field.
Field boundary	4) The field boundary can be irregular. 5) The headland space is confined relative to the vehicle’s size.
Environment	6) Static obstacles may be present within the headland area.

to develop advanced motion planning algorithms and control strategies that can address these challenges and enable AAVs to safely and efficiently navigate in a wide range of agricultural environments. By doing so, the adoption of AAVs can be accelerated, ultimately improving productivity and reducing labor costs in the farming industry. In response to these challenges, in this work, we propose a novel approach for planning turning motions that enable AAVs to navigate in these demanding environments. The main contributions of this work are as follows:

1. We present a generic representation of headland space, including irregular field boundaries, irregular crop row ends, and static obstacles. Similarly, we also introduce a generic way of modeling AAVs with or without rigidly connected implements.
2. We propose a novel framework for computing an optimal, feasible, and smooth trajectory for AAV turning in constrained headland space.
3. We evaluate the performance of the proposed algorithm in various headland environments and compare it with the

classic pattern-based method. We also demonstrate that our framework can be implemented on a real robot and solve complex headland-turning problems in real-world fields.

4. We open-source our headland-turning planning algorithm in a public repository. (https://github.com/AgRoboticsResearch/headland_trajectory_planning).

The rest of the paper is structured as follows: In Section 2, we review related work in the literature. Section 3 introduces our proposed optimization-based planning algorithm. Experiments and results are presented in section 4, followed by our discussions in section 5. Section 6 concludes the paper.

2. Literature Review

2.1. Dubins and Reeds-Shepp Curves

Dubins curves (Dubins, 1957) and Reeds-Shepp curves (Reeds and Shepp, 1990) are commonly used to generate turning paths, such as U -turns, Ω -turns, and switch-back turns (illustrated in Fig. 3), for solving problems related to coverage path planning and global traverse sequence planning problems (Bochtis and Vougioukas, 2008; Jin and Tang, 2010; Spekken and de Bruin, 2013; Mier et al., 2023). However, these paths suffer from two main drawbacks. First, they require the vehicle to turn at max steer, which puts high pressure on the steering system and may lead to off-track. Furthermore, they assume that the steering angle can be changed instantaneously, ignoring the steering system’s dynamic constraints, making it infeasible in reality. Recent work by He et al. (2023) proposed dynamic turning path planning for switch-back turns that alleviates the off-track problem. This approach, however, does not consider the steering dynamics constraints in the planning stage. Second, these turning path patterns are fixed by the turning type and do not take into account the field boundaries, crop rows, and other obstacles. In practice, the headland space may be restricted due to irregular field boundaries and checkered crop row endpoints even in the absence of obstacles, as described in Section 1. Applying these paths could easily interfere with the boundary, which is often designed to be tight for maximum land usage. When the field boundary and crop row ends are irregular, the modification of the cultivation layout may require a large unplanted area, leading to land wastage.

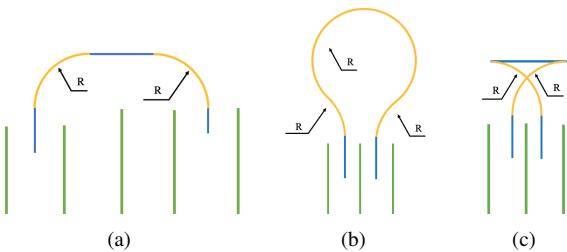


Figure 3: Illustrations of (a) U -turn, (b) Ω -turn, and (c) switch-back turn.

2.2. Continuous Curvature Path

Although Reeds-Shepp paths are widely used and easy to apply, they suffer from the issue of discontinuous curvature at the junctions of segments. This results in discontinuous steering angle commands, presenting challenges to the tracking control of non-holonomic vehicles. To accurately follow the trajectory, the vehicle may need to stop and steer. The problem becomes more serious when subsequent arcs have minimum radius arc for headland space-saving purposes. To solve this problem, Fraichard and Scheuer (2004) proposed a Continuous Curvature (CC) path that smooths the Reeds-Shepp path by ensuring continuous curvature. Cariou et al. (2010) presented a motion planner based on primitives connected with clothoid arcs to generate admissible trajectories. Sabelhaus et al. (2013) developed a method for generating continuous curvature headland trajectories and investigated the properties of different turning types with CC turns. Backman et al. (2015) introduced a method for generating a smooth turning path with continuous curvature and speed profiles, taking into account the vehicle’s maximum steering rate and linear acceleration. Plessen and Bemporad (2017) explored three methods for generating smooth reference trajectories and compared their tracking performance using linear time-varying model predictive control. Research on CC path planning has addressed the problem of generating smooth and trackable trajectories under steering constraints. However, these methods still plan their turning paths ignoring the constraints imposed by limited headland spaces.

2.3. Optimization-Based Planning Using a Simple Headland Space Model

Several research efforts have been made to model the headland spatial limits and incorporate them into turning path planning. For instance, Oksanen (2007) modeled the headland as parameterized lines, and calculated the optimal trajectory by solving an optimal control problem (OCP). Similarly, Tu and Tang (2019) employed a comparable headland space representation and explored a numerical optimization-based method to solve such an OCP. They also discovered that the headland boundary angle plays a critical role in the solution existence, particularly when the headland width is restricted. These studies highlight the growing interest among researchers in identifying feasible turning paths within constrained headland spaces. However, the headland space modelings in these works are overly simplified and fail to accommodate scenarios where the headland space exhibits irregularities.

2.4. Generic Motion Planning For Agricultural Vehicles

Researchers have also explored the generic motion planning problem in agricultural applications, which involves navigating vehicles from one pose to another while accounting for existing obstacles and environmental structures. Due to the non-convex and non-linear nature of this problem, various stochastic optimization techniques, such as genetic algorithm, simulated annealing, and neural networks have been employed to solve this problem (Noguchi and Terao, 1997; Makino et al., 1999; Ferentinos et al., 2002). However, these methods often suffer

from slow convergence and cannot guarantee optimal solutions due to the vast non-convex searching space. The research most closely related to our work is by [Vougioukas et al. \(2006\)](#), who proposed a two-stage optimal motion planner for the automatic planning of optimal tractor motions. In that work, a randomized planner is employed in the first stage to compute a feasible sub-optimal trajectory, which is then refined in the second stage under an optimal control framework based on the function-space gradient descent method. However, the computation time for both stages increases dramatically in the presence of obstacles due to the explicit calculation of the minimum distance function at every time step. Additionally, rather than employing constraints, collisions are incorporated as a penalty term into the cost function, which means a collision-free trajectory cannot be guaranteed.

3. Methodology

In this work, we present a hierarchical framework designed to calculate a safe and optimal trajectory for an AAV during headland turning. The framework operates in two stages: 1) In Stage I, we select a coarse path from a collection of classical headland turning path libraries. If none of the paths are feasible, we then apply a kino-dynamic path-searching algorithm, which operates based on the topological map of the field block, to find a feasible path. This coarse path meets the collision-free requirements and adheres to the kinematic constraints of the AAV, such as minimum curvature constraints. We prioritize the pattern-based turning over kino-dynamic searching due to its significant reduced computational time. Furthermore, the pattern-based turning path align closer to the optimized path when the headland space is ample and free of static obstacles. However, in instances of complex headland conditions, the searched path can circumvent obstacles, providing a more refined initial guess for the optimizer; 2) In Stage II, a numerical optimizer is applied to compute the optimal trajectory based on the coarse path identified in the first stage. This optimal trajectory minimizes a user-defined cost function that not only smooths the trajectory but also reduces the total path length and control efforts. The optimizer also ensures that the resulting trajectory satisfies collision-avoidance, vehicle kinematic, and control constraints. The flowchart that illustrates the planner's process is presented in Fig. 4.

3.1. System Dynamics

Given the state space \mathcal{X} and the control input space \mathcal{U} , the vehicle dynamics can be represented in a discrete form as follows:

$$\mathbf{x}_{i+1} = \mathcal{F}(\mathbf{x}_i, \mathbf{u}_i) \quad (1)$$

where $\mathbf{x}_i \in \mathcal{X}$ is the vehicle's state vector, $\mathbf{u}_i \in \mathcal{U}$ is the control input vector, $\mathcal{F} : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$ represents the vehicle's motion equation, and $i \in \{0, \dots, N\}$ denotes the time step index. The state and control input vectors are subjected to bounded box constraints of the form:

$$\mathbf{x}_{min} \leq \mathbf{x}_i \leq \mathbf{x}_{max}, \quad \mathbf{u}_{min} \leq \mathbf{u}_i \leq \mathbf{u}_{max} \quad (2)$$

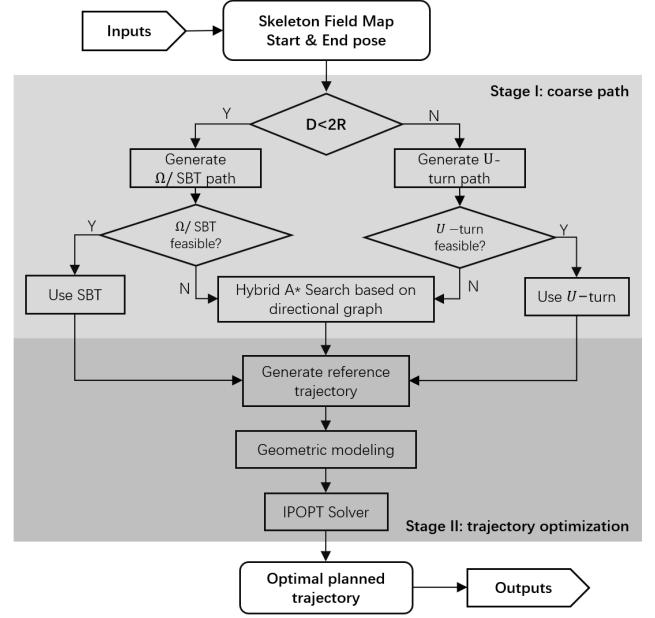


Figure 4: A flowchart of the proposed framework.

We use a simple kinematic bicycle model to capture the vehicle dynamics in this work. The discretized motion equations are provided in (3).

$$\begin{cases} x_{i+1} = x_i + v_i \cos \theta \Delta T \\ y_{i+1} = y_i + v_i \sin \theta \Delta T \\ \theta_{i+1} = \theta_i + v_i \tan \phi_i / L \Delta T \\ v_{i+1} = v_i + \dot{v}_i \Delta T \\ \phi_{i+1} = \phi_i + \dot{\phi}_i \Delta T \end{cases} \quad (3)$$

where $[x_i, y_i]^T$ represents the 2D position of the vehicle's real-wheel axle midpoint in the global frame. v_i denotes the vehicle's linear velocity along the direction of motion, while θ_i is the heading angle between v_i and the global x-axis. ϕ_i is the steering angle of the front wheel. L refers to the wheelbase length, and ΔT is the discretization time interval. Consequently, we have $\mathbf{x}_i = [x_i, y_i, \theta_i, v_i, \phi_i]^T$. The control input $\mathbf{u}_i = [\dot{v}_i, \dot{\phi}_i]^T$ consists of linear acceleration \dot{v}_i and steering rate $\dot{\phi}_i$.

3.2. Headland Environment Modeling

We model the headland environment by field boundaries, crop rows, and static obstacles, as illustrated in Fig. 5. The locations of them can be acquired using a handheld Real-Time Kinematic-Global Navigation Satellite System (RTK-GNSS) device. We will refer to this representation as a *map* throughout the paper. For the purposes of this study, we make the following assumptions: 1) we consider a 2D navigation problem, assuming negligible differences in elevation, 2) the map is pre-constructed and known to the vehicle, and 3) the vehicle's localization can be obtained from an RTK-GNSS device, with its error being negligible.

In this work, field boundaries, crop rows, and static obstacles within the area of interest, collectively referred to as *obstacles*,

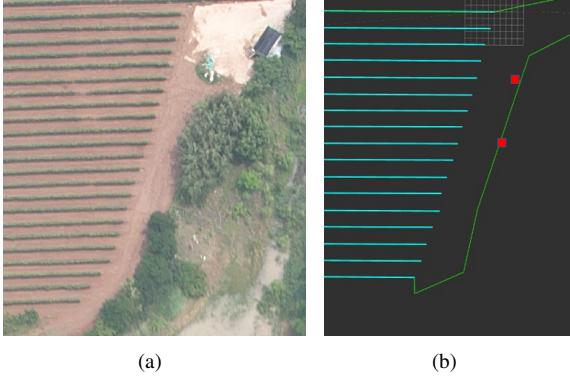


Figure 5: (a) The left figure provides a top-down view of the vineyard’s headland space. (b) The right figure is the representing map of the vineyard, where the polylines (green) represent field boundaries, the red squares represent the static obstacles, and the rectangles (cyan) represent crop rows.

are represented by *convex polygons*. The occupied spaces of these polygons are considered non-traversable for the vehicle and the shapes and locations of them remain constant over time. These convex polygons are denoted as $\mathbb{O}_1, \dots, \mathbb{O}_M \subset \mathbb{R}^2$, where M is the total number of obstacles. Each obstacle polygon can be represented as:

$$\mathbb{O}_m = \{z \in \mathbb{R}^2 : A_m z \leq \mathbf{b}_m\} \quad (4)$$

where $A_m \in \mathbb{R}^{l \times 2}$, $\mathbf{b}_m \in \mathbb{R}^l$, and $l \geq 3$ denotes the number of sides of polygon m , assuming a non-empty interior. We define a typical field and a non-typical field as follows:

1. A *typical field* is characterized by a headland space defined by two parallel lines. These lines represent the field boundary and the connection of crop row endpoints, respectively. We assume that the crop rows are straight, parallel, and equally spaced apart. Parameters of a typical field include the width of the headland (D), the width between two crop rows (w), the width of trees (w_{tree}), and the angle β between the headland line and the crop lines. This model is graphically represented in Fig. 6. A visual example of such a typical field can be seen in Fig. 7(a).
2. A *non-typical field*, which more accurately reflects real-world scenarios, is characterized by headland boundaries and row endpoints that cannot be simplified to straight lines. For instance, while the boundary in a typical field can be represented as a simple quadrilateral obstacle, the boundary in a non-typical field is represented using a polyline. Furthermore, the non-traversable areas can be denoted by multiple polygons, as shown in Fig. 7(b). Crop rows are represented as rectangles in this model.

3.3. Vehicle and Implements Modeling

An AAV equipped with implements may exhibit complex geometries. To represent the vehicle and its implements in a non-conservative manner in 2D space, multiple polygons are utilized to outline their occupied spaces. Figure 8 illustrates an

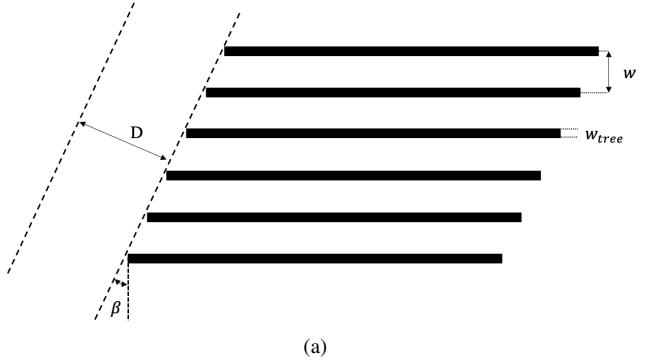


Figure 6: Diagram of a typical field parameterized by the width of the headland (D), the width between two crop rows (w), the width of trees (w_{tree}), and the angle (β) between the headland line and the crop lines.

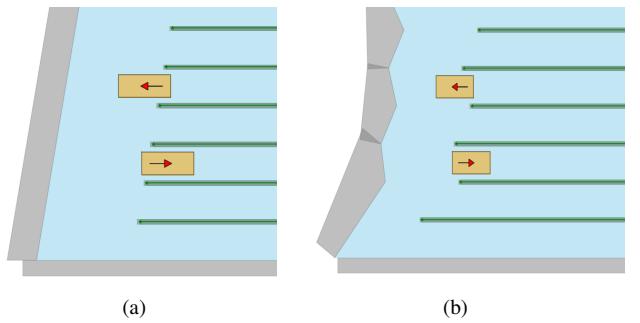


Figure 7: The field boundary (gray) is represented by (a) a quadrilateral in a typical field and (b) multiple polygons in a non-typical field. The crop rows are represented by rectangles (green).

example of an agricultural vehicle outfitted with a pruner and its corresponding polygon representation. In this study, only rigid-connected implements are considered for the AAV’s motion planning.

Given the current state \mathbf{x}_i , the vehicle and any attached implements are modeled as K *convex polygons*, denoted as $\mathbb{V}_1(\mathbf{x}_i), \dots, \mathbb{V}_K(\mathbf{x}_i) \subset \mathbb{R}^2$. The occupied space of $\mathbb{V}_k(\mathbf{x}_i)$ can be alternatively represented by a combination of rotation and translation applied to an initial convex set \mathbb{B}_k :

$$\mathbb{V}_k(\mathbf{x}_i) = \mathbf{R}(\mathbf{x}_i) \cdot \mathbb{B}_k + \mathbf{t}(\mathbf{x}_i), \quad \mathbb{B}_k = \{z \in \mathbb{R}^2 : \mathbf{G}_k z \leq \mathbf{g}_k\} \quad (5)$$

In this formula, $\mathbf{R}(\mathbf{x}_i) \subset \mathbb{R}^{2 \times 2}$ is a rotation matrix and $\mathbf{t}(\mathbf{x}_i) \subset \mathbb{R}^2$ is a translation vector. The initial convex set \mathbb{B}_k remains constant, where $\mathbf{G}_k \in \mathbb{R}^{j \times 2}$, $\mathbf{g}_k \in \mathbb{R}^j$, and $j \geq 3$ depends on the number of sides of either the vehicle or its implement, assuming a no-empty interior. From this point forward, we will refer to the vehicle and its implements collectively as the *vehicle*.

3.4. General Optimal Control Problem

The collision-avoidance condition can be formally expressed by (6), which must hold true at every time step. In other words, at each time step i , the intersection of the occupied spaces of the vehicle polygons and the obstacle polygons must be empty, meaning that there is no collision between the vehicle and any

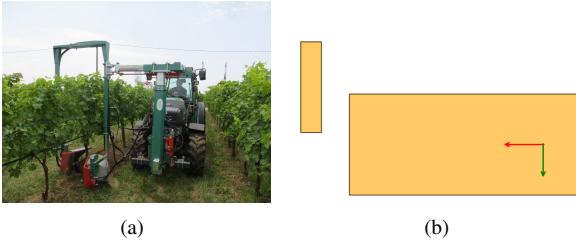


Figure 8: (a) An agricultural vehicle equipped with a summer pruner in the vineyard (Photo credit to CLEMENS Technologies) and (b) its polygonal modeling (top-down view) for motion planning.

of the obstacles.

$$\mathbb{O}_m \cap \mathbb{V}_k(\mathbf{x}_i) = \emptyset, \forall m \in \{1, \dots, M\}, \forall k \in \{1, \dots, K\} \quad (6)$$

We can formulate the headland motion planning as an optimal control problem. Given a feasible initial state \mathbf{x}_I and terminal state \mathbf{x}_T , a set of vehicle polygons $\{\mathbb{V}_1, \dots, \mathbb{V}_K\}$ and obstacle polygons $\{\mathbb{O}_1, \dots, \mathbb{O}_M\}$, and the total time steps N , we need to determine a state sequence $\mathbf{x} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N\}$ and a control input sequence $\mathbf{u} = \{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N-1}\}$ that minimizes a user-defined cost function $\mathcal{J}(\mathbf{x}, \mathbf{u})$. N is determined by the initial guess trajectory. This cost function may optimize factors such as total path length, control efforts, and trajectory smoothness, among others. The resulting state sequence must satisfy the collision-avoidance requirement at each time step. Additionally, the states and control inputs must also adhere to the vehicle’s equations of motion and their respective limitations. The representation of this generic OCP is provided as follows:

$$\begin{aligned} & \min_{\mathbf{x}, \mathbf{u}} && \mathcal{J}(\mathbf{x}, \mathbf{u}) \\ & \text{subject to} && \mathbf{x}_0 = \mathbf{x}_I, \mathbf{x}_N = \mathbf{x}_T \\ & && \mathbf{x}_{i+1} = \mathcal{F}(\mathbf{x}_i, \mathbf{u}_i) \\ & && \mathbf{x}_{\min} \leq \mathbf{x}_i \leq \mathbf{x}_{\max} \\ & && \mathbf{u}_{\min} \leq \mathbf{u}_i \leq \mathbf{u}_{\max} \\ & && \mathbb{O}_m \cap \mathbb{V}_k(\mathbf{x}_i) = \emptyset, \\ & && \forall i \in \{0, \dots, N-1\}, \forall m \in \{1, \dots, M\}, \\ & && \forall k \in \{1, \dots, K\} \end{aligned} \quad (7)$$

3.5. Collision-Avoidance Dual Problem

The collision-avoidance constraints in (7) are typically non-differentiable, making the optimization problem challenging to solve. We adopt the idea from the paper (Zhang et al., 2020) and transform the non-differentiable collision-avoidance constraints into smooth, differentiable constraints to facilitate computation. Given that $\mathbb{O}_1, \dots, \mathbb{O}_M$ and $\mathbb{V}_1(\mathbf{x}_i), \dots, \mathbb{V}_K(\mathbf{x}_i)$ are all convex sets and using the signed distance, $\text{dist}(\mathbb{V}_k(\mathbf{x}_i), \mathbb{O}_m)$, to write the collision-avoidance constraints (Schulman et al., 2014), we can take advantage of the strong duality of convex optimization (Boyd et al., 2004) and reformulates the non-differentiable collision-avoidance constraints into a set of smoothed convex constraints. The dual problem to $\text{dist}(\mathbb{V}_k(\mathbf{x}_i), \mathbb{O}_m) > d_{\min}$, com-

puted at time step i , is equivalent to:

$$\begin{aligned} \exists \lambda_i^{m,k} \geq 0 : & \begin{cases} -\mathbf{g}_k^T \boldsymbol{\mu}_i^{m,k} + (\mathbf{A}_m \mathbf{t}(\mathbf{x}_i) - \mathbf{b}_m)^T \lambda_i^{m,k} > d_{\min}^{m,k}, \\ \mathbf{G}_k^T \boldsymbol{\mu}_i^{m,k} + \mathbf{R}(\mathbf{x}_i)^T \mathbf{A}_m^T \lambda_i^{m,k} = 0, \\ \|\mathbf{A}_m^T \lambda_i^{m,k}\|_2 \leq 1 \end{cases} \end{aligned} \quad (8)$$

where $d_{\min}^{m,k}$ is the desired minimum distance between the two convex sets. $\lambda_i^{m,k}$ and $\boldsymbol{\mu}_i^{m,k}$ are the dual variables associated with the obstacle \mathbb{O}_m and the vehicle part $\mathbb{V}_k(\mathbf{x}_i)$, at time step i .

3.5.1. Optimization-Based Collision-Avoidance (OBCA)

By incorporating the duality in (8), the generic OCP outlined in (7) can be exactly reformulated with smooth and differentiable constraints. The effectiveness of this method has been validated in solving the autonomous valet parking problems (Zhang et al., 2020; Zhou et al., 2020). This reformulation, along with our chosen cost function, is presented below:

$$\min_{\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}} \sum_{i=0}^{N-1} \mathbf{u}_i^T \mathbf{Q} \mathbf{u}_i + \Delta \mathbf{u}_i^T \mathbf{P} \Delta \mathbf{u}_i + w(v_i \Delta T)^2 \quad (9a)$$

$$\text{subject to } \mathbf{x}_0 = \mathbf{x}_I, \mathbf{x}_N = \mathbf{x}_T \quad (9b)$$

$$\mathbf{x}_{i+1} = \mathcal{F}(\mathbf{x}_i, \mathbf{u}_i) \quad (9c)$$

$$\mathbf{x}_{\min} \leq \mathbf{x}_i \leq \mathbf{x}_{\max} \quad (9d)$$

$$\mathbf{u}_{\min} \leq \mathbf{u}_i \leq \mathbf{u}_{\max} \quad (9e)$$

$$-\mathbf{g}_k^T \boldsymbol{\mu}_i^{m,k} + (\mathbf{A}_m \mathbf{t}(\mathbf{x}_i) - \mathbf{b}_m)^T \lambda_i^{m,k} \geq d_{\min}^{m,k} \quad (9f)$$

$$\mathbf{G}_k^T \boldsymbol{\mu}_i^{m,k} + \mathbf{R}(\mathbf{x}_i)^T \mathbf{A}_m^T \lambda_i^{m,k} = 0 \quad (9g)$$

$$\|\mathbf{A}_m^T \lambda_i^{m,k}\|_2 \leq 1 \quad (9h)$$

$$\lambda_i^{m,k} \geq 0, \boldsymbol{\mu}_i^{m,k} \geq 0, \quad (9i)$$

$$\forall i \in \{0, \dots, N-1\}, \forall k \in \{1, \dots, K\},$$

$$\forall m \in \{1, \dots, M\}$$

The cost function (9a) is designed to minimize the control efforts, control input derivatives (i.e., speed and steering jerks), and overall path length. Equations (9b) establish the initial and terminal condition constraints, while (9c) represents the motion equation constraint for the vehicle. State and control input constraints are encoded in (9d) and (9e), respectively. Collision-avoidance is ensured by (9f), (9g), (9h), and all dual variables must remain positive, as mandated by (9i). In this work, we use the off-the-shelf solver, Interior Point Optimizer (IPOPT), to solve the optimization problem in (9).

3.6. Calculate Initial Guess

Given the strong non-linearity of the optimization problem described in (9), the feasibility and computation time of the solution are heavily dependent on the initial guess, also known as the warm start. Identifying an appropriate initial guess can pose a significant challenge for general optimal motion planning problems. However, in the specific context of row-to-row turning in agricultural fields, a variety of path patterns have been extensively studied and established. The selection among

these patterns can be strategically made considering the starting and ending poses, the turning radius, the physical dimension of the vehicle, and the headland space. In cases where a path generated through pattern-based strategies fails to be feasible, an algorithm based on a directed-graph is utilized to determine a viable path. Given the path and the desired speed, a reference trajectory is obtained. This reference trajectory serves as an initial guess for the optimization problem encapsulated in (9), which enhances computational efficiency and facilitates the discovery of the optimal solution more readily.

3.6.1. Approach 1: Use Pattern-Based Turning

The agriculture domain has seen the development of numerous pattern-based headland turning strategies, including U-turning, Ω -turning, and switch-back turning, among others (Jin and Tang, 2010). These classical turning patterns comprise arc and line segments. The arc radius should not be smaller than the controlled vehicle's minimum turning radius. In this work, the headland turning patterns are automatically selected based on the vehicle's minimum turning radius (R) and the skipped width (d) of start and end poses. If $d < 2R$, a switch-back or ω -turning path is applied. If $d > 2R$, a U-turn path is applied (Sabelhaus et al., 2015).

In the pattern-based turning approach, the vehicle is presumed to execute the turn within an unobstructed space. In orchard environments, this could necessitate adjusting the path outward to a certain distance to ensure avoidance of potential collisions with crop rows. In this work, we implement the classic pattern-based turning method, following the pseudocode provided in Algorithm 1. If the adjusted path also adheres to the boundary constraints, it is considered feasible and used to generate the initial guess trajectory for the path smoother of OBCA. Fig 9 offers an illustrative example of this path-generating method. The black dashed lines represent the paths under evaluation during the adjusting process, while the blue curve depicts the final collision-free path. Once a suitable turning pattern has been identified, an initial guess trajectory is obtained based on the generated path and a predetermined speed profile.

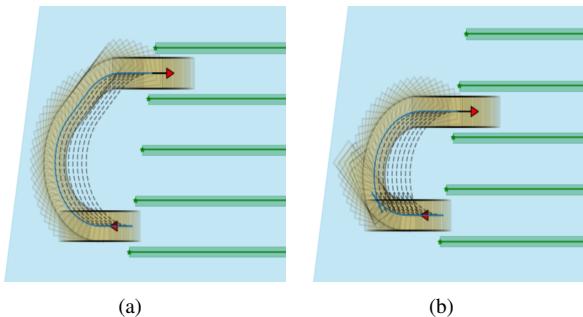


Figure 9: The process of generating the pattern-based method: a) U-turn path, when $d > 2R$, and b) Switch-back-turn path, when $d < 2R$.

Algorithm 1 Pattern-based planning algorithm

```

Input: start pose:  $(x_s, y_s, \theta_s)$ , end pose:  $(x_e, y_e, \theta_e)$ , turn radius:  $R$ , path resolution:  $ds$ , skeleton map:  $\mathbb{M}$ , shift step:  $\Delta d$ 
Output:  $\text{path} = [(x_0, y_0, \theta_0), \dots, (x_N, y_N, \theta_N)]$ 
 $d = abs(y_I - y_e)$ 
if  $d < 2R$  then
     $\text{path} = S\text{witchBackTurn}(x_s, y_s, \theta_s, x_e, y_e, \theta_e, R, ds)$ 
else
     $\text{path} = U - Turn(\text{start pose}, \text{end pose}, R, ds)$ 
end if
while  $\text{PathInterfereWithCropRows}(\mathbb{M}, \text{path})$  do
    for  $i = 1 : N$  do
         $x_i = \text{path}[i, 0], y_i = \text{path}[i, 1]$ 
         $x_i = x_i + \Delta d \cdot \cos(\theta_s)$ 
         $y_i = y_i + \Delta d \cdot \sin(\theta_s)$ 
    end for
end while

```

3.6.2. Approach 2: Use Hybrid A*

In situations where pattern-based turning paths (as described in Approach 1) fail to provide feasible solutions due to the constraints such as limited headland spaces and existing obstacles, a Hybrid A* planner is utilized to determine an initial guess. Hybrid A* is a search-based path planner that is capable of generating a smooth path in clustered environments for vehicles with non-holonomic constraints (Dolgov et al., 2010). The searching process includes two main steps. First, candidate nodes are created from the current ‘frontier’ nodes, which have the lowest cost. The cost is determined using a heuristic function defined by the user. Second, all feasible nodes are attempted to be connected to the goal node using an analytic expansion, such as the Reeds-Shepp or Dubins curves with a provided turning radius. The searching tree will be kept expanded unless a feasible path is found from the frontier node to the goal pose.

In scenarios such as parking lots, a directional graph can be constructed based on the map (Qin et al., 2020) and a heuristic cost function can be designed using the directional graph (Dolgov et al., 2010). The primary intuition behind this heuristic function is to assign high costs to path segments that deviate from the topological line. We adopt the idea of directional graph into agricultural fields which also exhibit topological structures like parking lots. This heuristic function has proven to be substantially more efficient than grid space-based heuristic functions (Qin et al., 2020), particularly when the vehicle needs to traverse multiple rows to reach a goal position during headland turning. For detailed proving and implementations, interested readers are referred to the paper (Dolgov et al., 2010; Qin et al., 2020). An example of directional searching in the agricultural field is presented in Fig. 10, where the path searching is prioritized in the area along the directional graph.

3.7. Trajectory Tracking

After obtaining a smoothed and optimal trajectory, we apply the nonlinear model predictive control (NMPC) to achieve an

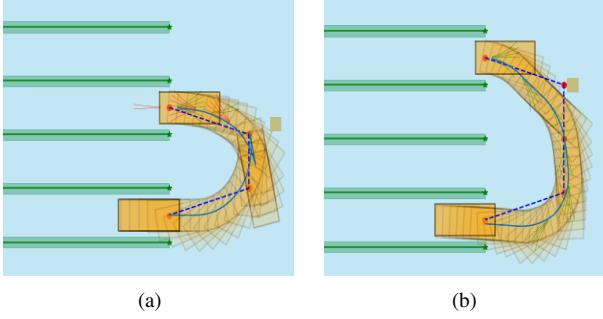


Figure 10: Hybrid A* path based on the directional graph in two different cases: a) Case I: $d < 2R$; b) Case II: $d > 2R$. The green curves represent the expanded nodes' paths in forward motions, while the red curves represent the paths in backward motions. The blue curves represent the searched path. The square represents a static obstacle in the headland.

accurate tracking. NMPC optimizes control inputs over a finite horizon for a nonlinear system, while satisfying user-defined constraints. It also exhibits robustness against disturbances. Constrained output path-following using MPC has been discussed in Faulwasser and Findeisen (2016). By discretizing the system, we can convert the OCP into a nonlinear programming problem, the objective of which is to determine a sequence of control inputs that minimize the deviations from a reference trajectory and reduce control efforts jointly, while adhering to constraints on state and control inputs. The formula of our NMPC is provided in (10).

$$\begin{aligned} \min_{\boldsymbol{u}} \quad & \boldsymbol{e}_{N_p}^T \boldsymbol{Q}_e \boldsymbol{e}_{N_p} + \sum_{i=0}^{N_p-1} \boldsymbol{e}_i^T \boldsymbol{Q} \boldsymbol{e}_i + \boldsymbol{u}_i^T \boldsymbol{R} \boldsymbol{u}_i + \Delta \boldsymbol{u}_i^T \boldsymbol{P} \Delta \boldsymbol{u}_i \\ \text{subject to} \quad & \boldsymbol{x}_0 = \boldsymbol{x}_{init} \\ & \boldsymbol{x}_{i+1} = \mathcal{F}(\boldsymbol{x}_i, \boldsymbol{x}_i) \\ & \boldsymbol{x}_{min} \leq \boldsymbol{x}_i \leq \boldsymbol{x}_{max} \\ & \boldsymbol{u}_{min} \leq \boldsymbol{u}_i \leq \boldsymbol{u}_{max} \\ & \Delta \boldsymbol{u}_{min} \leq \Delta \boldsymbol{u}_i \leq \Delta \boldsymbol{u}_{max} \end{aligned} \quad (10a-f)$$

In the cost function, $\boldsymbol{e}_i = \boldsymbol{x}_i - \boldsymbol{x}_i^{ref}$ refers to the difference between the current state \boldsymbol{x}_i and the reference trajectory state \boldsymbol{x}_i^{ref} . \boldsymbol{Q} and \boldsymbol{R} denote the weighting matrices. To improve the tracking performance, we take into account the speed jerk and steering jerk, denoted as $\Delta \boldsymbol{u}_i$. The corresponding weighting matrix is represented as \boldsymbol{P} in the cost function. N_p refers to the prediction steps. To make the problem easier to solve, we relax the terminal state constraint by incorporating it into the cost function and penalizing its difference $\boldsymbol{e}_N^T \boldsymbol{Q}_e \boldsymbol{e}_N$, where \boldsymbol{Q}_e denotes the penalty weight. In this work, a state-of-the-art MPC solver (Verschueren et al., 2022) is utilized to tackle the optimization problem in real time.

4. Experiments and Results

We conducted a series of experiments to evaluate the performance of our proposed methodology. The results from our optimization-based motion planner were compared against those obtained from classic pattern-based planners (Jin and Tang, 2010) across a range of headland configurations. We broadly categorized our experiments into four distinct scenarios:

- 1) *Typical field*: The first experiment benchmarked our planner against classic pattern-based planners in typical fields. The definition of a typical field is provided in Section 3.2.
- 2) *Non-typical field*: The second experiment compared the performance of our planner with that of classical pattern-based planners in two non-standard fields. The definition of a non-typical field is outlined in Section 3.2.
- 3) *Complex headland*: In the third experiment, we assessed our planner's performance in a complex headland with narrow spaces and static obstacles, a situation that poses a significant challenge to the classic pattern-based planner. This experiment demonstrated the capability of our planner to navigate more constrained headland spaces.
- 4) *Real-world field*: The final experiment evaluated our planner's performance in a real-world headland with artificial obstacles and boundaries, mimicking the complex headland tested in the previous experiment. This included assessing the performance of the entire autonomous navigation system, incorporating both turning trajectory planning and tracking, using a real robot in the field headland.

In scenarios 1 and 2, we considered the vehicle as a rectangle with dimensions of $3.8 \times 1.5 \text{ m}$. The wheelbase is assumed to be $L = 1.9 \text{ m}$. The steering angle is limited between $\delta \in [-0.6, 0.6] \text{ rad}$, the vehicle velocity is limited between $v \in [-1, 2] \text{ m/s}$, the linear acceleration is between $a \in [-0.6, 0.6] \text{ m/s}^2$ and the steering rate is constrained $\dot{\delta} \in [-0.7, 0.7] \text{ rad/s}$. The above parameters are reasonable and realistic for a medium-sized orchard tractor. The sampling time of the trajectory planner is selected as 0.2 s . We set the



Figure 11: A tricycle mobile robot used in the field experiments.

width of crop row $w = 2.5 \text{ m}$ and the treewidth is assumed to be $w_{tree} = 0.4 \text{ m}$, which is typical for vineyard fields. In scenarios 3 and 4, we implemented the developed planning and control algorithms on a tricycle mobile robot (shown in Fig. 11) within a challenging man-made environment. The robot has a wheelbase of 1.3 m and dimensions of $1.8 \times 1.2 \text{ m}$. The steering angle is constrained between $\delta \in [-0.6, 0.6] \text{ rad}$ and the velocity is constrained between $v \in [-1, 1] \text{ m/s}$. The linear acceleration is constrained to $a \in [-0.6, 0.6] \text{ m/s}^2$, and the steering rate is limited to $\dot{\delta} \in [-0.7, 0.7] \text{ rad/s}$.

4.1. Scenario I: Typical field

We first compared the performance of our planner against classic pattern-based planner in typical fields. Both planners were applied to plan a path that would enable the vehicle to transition from one row to another. The turning width, denoted as $W = n \times w$, is determined by the number of rows passed n and the crop width w . Two different headland turning cases were considered: (1) $W \geq 2R$, and (2) $W < 2R$. In case 1, the vehicle was directed to turn into the third row on the right, thereby allowing for a feasible forward-only path. Conversely, in case 2, the vehicle was guided to turn into the adjacent row on the right, necessitating a switch-back turning process. In both cases, we varied the headland width D from 4 m to 10 m with a resolution of 0.25 m and adjusted β from -30° to 30° with a resolution of 2.5° to explore the entire space of possible headlands. The evaluation metrics were chosen as the feasibility to provide a collision-free path. A generated path is considered feasible if it meets all collision-avoidance constraints. Additionally, our developed method must facilitate the optimizer to compute a feasible trajectory within a maximum time limit of 20 seconds. If the solver cannot return a feasible solution within this limit, it is deemed a failure.

The results of the first case can be viewed in Fig. 12(a), and those of the second case can be seen in Fig. 12(b). Each cell in those figures corresponds to a headland configuration. The color of each cell indicates the feasibility of the generated path. It should be noted that our planner's solution incorporates a comprehensive trajectory that takes into account the vehicle's operational constraints (e.g., maximum velocity and steering angle limits). In contrast, the classic pattern-based planner only provides a rudimentary path. It does not account for the vehicle's constraints except for minimal turning radius, and it might not be feasible for the vehicle to follow the path. Consequently, the trajectory produced by our planner is more realistic and easier to follow.

Figures 13 and 14 provide illustrative examples of the planned paths, which can help readers intuitively understand the difference between our planner's result and the classic planner's result. Figure 13 shows the planned path under case 1 using the forward-only setting for both planners, with an angle $\beta = 10^\circ$ and headland width $D = 6 \text{ m}$. Figure 14 presents the planned path under case 2, with both planners utilizing switch-back paths, featuring an angle $\beta = 10^\circ$ and headland width $D = 5.5 \text{ m}$. In both cases, our planner can determine a solution that successfully completes the turn, while the classic planner's solution is infeasible.

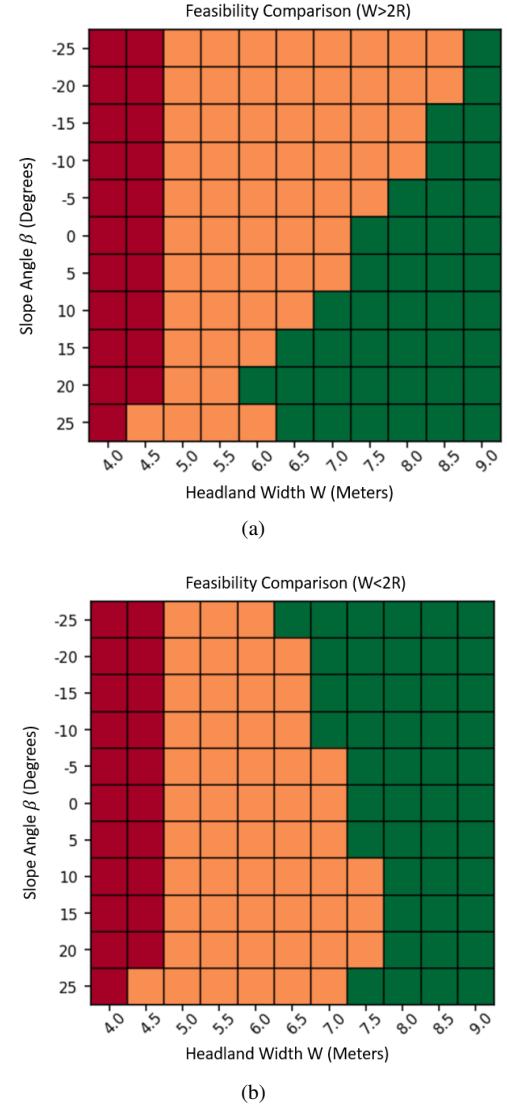
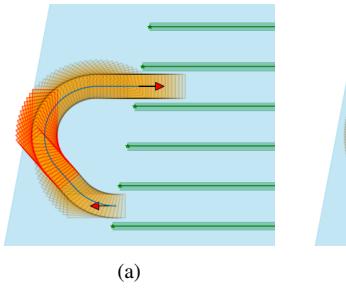


Figure 12: The feasibility results for different β and w values in two cases (a) $W > 2R$, and (b) $W < 2R$. The color of each cell represents the feasibility of the generated path. Red cells represent that neither the classic method nor our method can generate a feasible path. Yellow cells represent that the classic planner fails to generate a feasible path, while our method succeeds. Green cells represent that both the classic method and our method succeed in generating a feasible path.

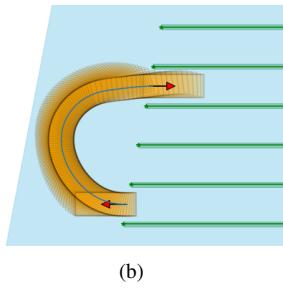
4.2. Scenario II: Non-typical field

The typical field model often understates the complexity of headland spaces. In real-world situations, headlands can be irregular, and the field boundaries and crop row ends do not necessarily align in straight lines. We designed a second experiment to assess our proposed planner's performance in two non-typical fields with irregular field boundaries and crop row ends. These non-typical field models, displayed in Fig. 15, are derived from actual field geometries using RTK-GNSS devices. Field 1 consists of 17 rows in total. Field 2 contains 46 rows in total and offers more challenge for the vehicle to make a turn in the headland compared to field 1.

We conducted exhaustive evaluations of headland turnings



(a)

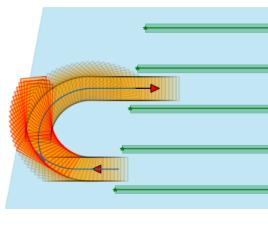
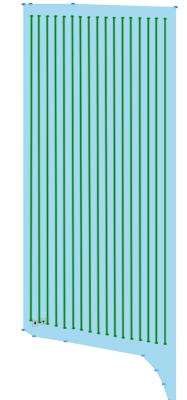


(b)

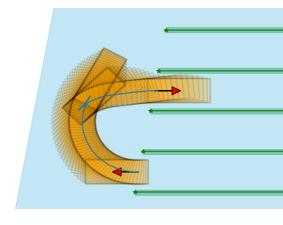
Figure 13: Comparison of path planning results for $\beta = 10^\circ$ and $D = 6.0 \text{ m}$. a) Infeasible path generated by the classic planner; b) Feasible path generated by our proposed planner. The blue curve is the planned path, and the vehicle's occupancy at each sampling step is drawn in the figure. Orange shapes denote safe poses, while red shading indicates collision instances with obstacles.



(a)

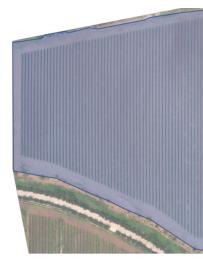


(a)

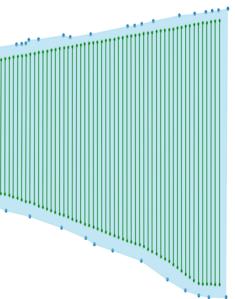


(b)

Figure 14: Comparison of path planning results for $\beta = 10^\circ$ and $D = 5.5 \text{ m}$. a) Infeasible path generated by the classic planner; b) Feasible path generated by our proposed planner.



(a)



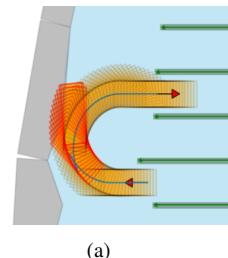
(b)

from all rows to their neighboring four rows, comparing the performance of our proposed planner against the classic pattern-based planner. In this study, we selected the success rate as our evaluation metric. The success rate results for both planners are presented in Table 2. These results indicate that our planner consistently outperforms the classic planner in terms of success rate across both fields. It achieved a 100% success rate in the field 1, where the classic planner only reached a 58% success rate. Even in a more constrained field, our planner maintained an 87% success rate, while the classic planner dropped to a rate of 9%. Figure 16 illustrates a non-typical field scenario where our planner successfully navigates the irregular headland space, while the classic method fails. However, in extremely challenging situation, our planner may struggle to provide a viable solution. Figure 17 presents two instances where our method fails due to excessively narrow headland spaces.

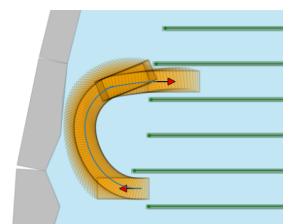
Table 2: The success rate results from the classic planner and our proposed planner in two non-typical field blocks.

Total Turnings	Our Planner		Classic Planner
	Success rate	Success rate	
Field 1	87	100%	58%
Field 2	292	87%	9%

Figure 15: Models of two non-typical fields from real world: (a) Field 1: a less challenging field with 17 rows, and (b) Field 2: a more challenging and constrained field with 46 rows.

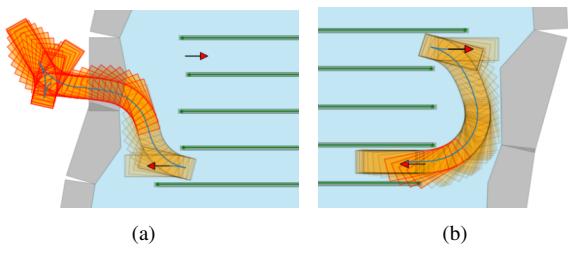


(a)



(b)

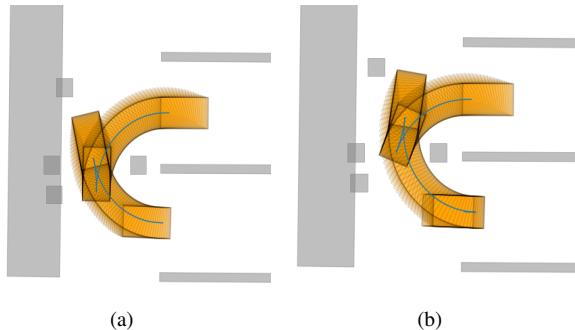
Figure 16: Our planner can utilize irregular space of the headland to maneuver while the classic planner failed to provide a safe path: (a) the classic planner; (b) our planner.



(a)

(b)

Figure 17: Two failure cases from our planner due to the excessively narrow headland spaces. The displayed path is the last solution from the solver before timeout.



(a) (b) (c)

Figure 18: Three man-made cases used to evaluate the planner’s performance in complex headland with narrow spaces and static obstacles.

4.3. Scenario III: Complex headland

Next, we assessed the performance of our planner in complex headlands with narrow spaces (at 4m width) and static obstacles. This evaluation aims to showcase our algorithm’s ability to handle more intricate headland conditions, particularly where the turning space is limited and additional collision-avoidance requirements should be enforced to avoid existing obstacle besides the field boundary and crop rows. It’s important to note that the classic pattern-based planner is unable to handle such complex scenarios.

We examined three different cases, as illustrated in Fig. 18. The planning results demonstrate that our planner is capable of generating a collision-free path within the space, ensuring safe navigation of the vehicle from the start to the end pose.

4.4. Scenario IV: Real-world field

In the final stage of our evaluation, we tested our pipeline, incorporating both planning and tracking modules, in a real-world field setting. The purpose of this experiment is to evaluate the performance of our autonomous navigation system in a realistic environment. The experiment was conducted in a vineyard located in Davis, CA, using a tricycle mobile robot, as shown in Fig. 11. The vineyard’s geometry was accurately mapped using an RTK-GNSS device and fed into our planner. To recreate the complexity of the headland conditions from the previous experiment (refer to Section 4.3), we artificially incorporated obstacles and boundaries in the headland space. All static obstacles’ locations were pre-known to the planner. Given the map of the field and the location of static obstacles, our planner could generate a feasible trajectory in the environment. The robot localization was achieved through an onboard RTK-GPS, fused with



(a)



(b)



(c)

Figure 19: Three testing cases used to evaluate the performance of our method on a mobile robot in a real-world field. The blue dashed line is the planned path and the red solid line is the actual path of the robot. The green rectangles visualize the obstacles’ locations in the headland. The field boundary and crop rows’ locations are projected to represent their relative positions graphically.

an IMU and wheel odometer, running an Extended Kalman Filter (EKF) in the background. Once our planner computed a trajectory, the tracking was executed by an NMPC, which was described in Section 3.7. All the computations were executed on an onboard computer equipped with an Intel i5-7300U CPU and 32 GB memory.

We present the planning and tracking results of three different cases in Figure 19. The blue dashed line represents the planned path, while the red solid lines shows the actual path. To help

visualize the obstacles' locations, we positioned paper boxes within the headland. Moreover, we projected the locations of the field boundaries and crop rows used in the planning onto the photograph to represent their relative positions graphically. The results clearly demonstrate our planner's ability to plan a feasible and smooth trajectory within a real-world setting that can be tracked by a real robot. The tracking error across all three cases remained below 0.1m. For supplementary video see: <https://www.youtube.com/watch?v=sf0uDFwpSfo>.

5. Discussion

We illustrated the efficacy of our planner in generating a smooth and optimal trajectory in Fig. 20. This figure presents the characteristics of a coarse path generated by the classic pattern-based planner and the optimized trajectory generated by our planner. These characteristics include curvature, heading, linear velocity, steering rate, and linear acceleration. The resultant trajectory from our planner is depicted in blue, while the orange represents the characteristics derived from the coarse path. The black dashed lines represent the bounds for each

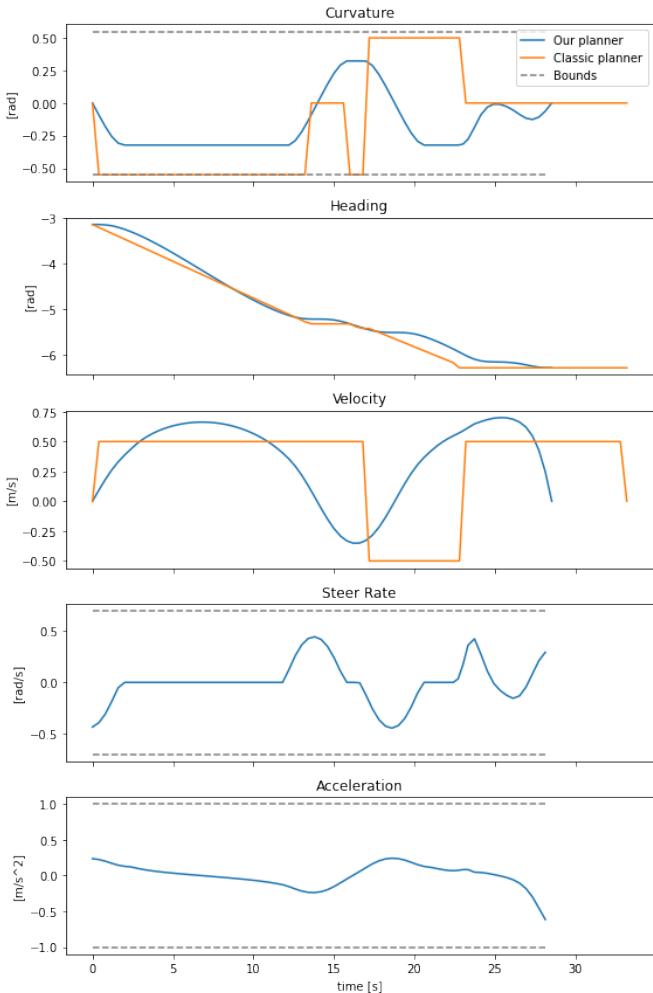


Figure 20: Comparison of the trajectory characteristics generated by a classic pattern-based planner and our proposed planner.

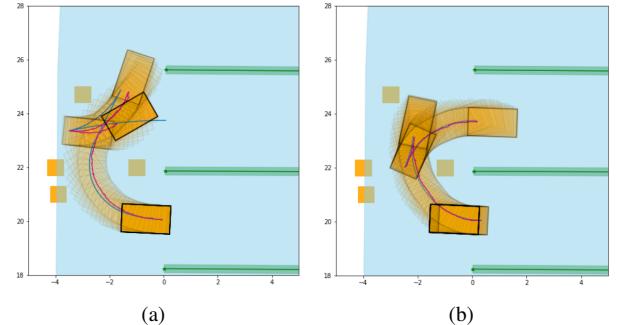


Figure 21: (a) The planned path from the Hybrid A* planner and its tracking results; (b) The planned path from our planner and its tracking results.

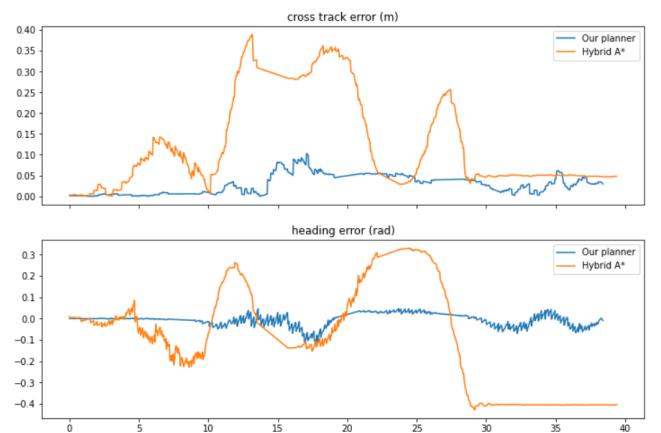


Figure 22: The cross-tracking errors and heading errors of the trajectory generated from the Hybrid A* planner (orange) and our planner (blue).

variable if exist. The results reveal that our method's output trajectory is significantly smoother than the coarse path and takes less time for the vehicle to complete. More importantly, the planned trajectory adheres to the vehicle's operational constraints without requiring instantaneous changes in values. This compliance with constraints makes the trajectory easier to track, demonstrating the advantage of our planning method over classic methods.

Moreover, we evaluated the NMPC's performance in tracking the trajectory generated by the Hybrid A* planner in the presence of static obstacles within the headland space, to confirm the improved trackability of the smoothed trajectory derived from our proposed planner. As depicted in Fig. 21, the mobile robot demonstrated superior efficiency in tracking the smoothed trajectory produced by our planner in contrast to the trajectory created by Hybrid A*. Additionally, according to Fig. 22, both the cross-tracking errors and heading errors from the Hybrid A* were noticeably larger than those encountered when tracking the smoothed trajectory generated by our planner.

Our method has effectively demonstrated its capabilities in both typical and non-typical field conditions, assuming sufficient space is available. However, under exceptionally challenging conditions, such as the extremely narrow headland spaces encountered in Section 4.2, our planner may struggle to

provide a solution within a given time window, or might not be able to provide a solution at all. A potential avenue for future research could be to enhance our method's adaptability and improve its success rate in generating feasible trajectories across a broader range of situations. This could potentially be achieved by supplying a more refined initial guess trajectory to the optimizer. It is also important to note that exceedingly challenging headland environments pose difficulties not only for our algorithm but also for human operators. While our algorithm might require more time to compute a solution, human drivers might need extra time for multiple maneuvers in these complex environments, extending the mission duration and total costs significantly.

Successful implementation of our algorithm is contingent upon enhanced computational capabilities and precise actuator control on the robot. Modern agricultural robots, such as autonomous electric tractors, present a promising platform for deploying our proposed pipeline given their ability to meet these technical and computational demands. Even though our algorithm was tested solely on a basic tricycle mobile robot during field experiments, we foresee that our pipeline can be applied to a broad spectrum of modern agricultural robots.

6. Conclusion

In conclusion, this paper presents a novel methodology for optimization-based motion planning in constrained headlands. Our approach consistently surpasses classic pattern-based method by generating feasible and optimal trajectories. By validating our method under various field configuration, we have highlighted its practical applicability and robustness. In typical field scenarios, our developed planner consistently outperforms the classic planner, achieving feasible and smooth trajectories in a larger number of situations. We also carried out a comparative analysis in two non-typical orchard field blocks, demonstrating the remarkable success rates of our planner that exceed those of the classic method by 78% and 42% in each respective field. Additionally, we have demonstrated its effectiveness in planning a safe and optimal trajectory in headlands with narrow spaces and static obstacles. Real-world experiments on an autonomous robot within an actual orchard further underscored the adaptability of our method to complex environments. The planned trajectory could be precisely tracked by the real robot, underscoring the robustness of our system. In essence, we offer an innovative solution to complex motion planning in constrained headlands, making it a valuable contribution to the autonomous operation of AAVs, particularly within orchards.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Backman, J., Piirainen, P., Oksanen, T., 2015. Smooth turning path generation for agricultural vehicles in headlands. *Biosystems Engineering* 139, 76–86. doi:[10.1016/j.biosystemseng.2015.08.005](https://doi.org/10.1016/j.biosystemseng.2015.08.005).
- Bochtis, D.D., Vougioukas, S.G., 2008. Minimising the non-working distance travelled by machines operating in a headland field pattern. *Biosystems Engineering* 101, 1–12. doi:[10.1016/j.biosystemseng.2008.06.008](https://doi.org/10.1016/j.biosystemseng.2008.06.008).
- Boyd, S., Boyd, S.P., Vandenberghe, L., 2004. Convex optimization. Cambridge university press.
- Cariou, C., Lenain, R., Thuiilot, B., Martinet, P., 2010. Autonomous maneuver of a farm vehicle with a trailede implement: Motion planner and lateral-longitudinal controllers. doi:[10.1109/ROBOT.2010.5509447](https://doi.org/10.1109/ROBOT.2010.5509447).
- Dolgov, D., Thrun, S., Montemerlo, M., Diebel, J., 2010. Path planning for autonomous vehicles in unknown semi-structured environments. *The international journal of robotics research* 29, 485–501.
- Dubins, L.E., 1957. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics* 79. doi:[10.2307/2372560](https://doi.org/10.2307/2372560).
- Faulwasser, T., Findeisen, R., 2016. Nonlinear model predictive control for constrained output path following. *IEEE Transactions on Automatic Control* 61. doi:[10.1109/TAC.2015.2466911](https://doi.org/10.1109/TAC.2015.2466911).
- Ferentinos, K.P., Arvanitis, K.G., Sigrimis, N., 2002. Heuristic optimization methods for motion planning of autonomous agricultural vehicles.
- Fraichard, T., Scheuer, A., 2004. From reeds and shepp's to continuous-curvature paths. *IEEE Transactions on Robotics* 20. doi:[10.1109/TRO.2004.833789](https://doi.org/10.1109/TRO.2004.833789).
- He, Z., Bao, Y., Yu, Q., Lu, P., He, Y., Liu, Y., 2023. Dynamic path planning method for headland turning of unmanned agricultural vehicles. *Computers and Electronics in Agriculture* 206. doi:[10.1016/j.compag.2023.107699](https://doi.org/10.1016/j.compag.2023.107699).
- Jin, J., Tang, L., 2010. Optimal coverage path planning for arable farming on 2d surfaces. *Transactions of the ASABE* 53, 283–295.
- Lordan, J., Francescato, P., Dominguez, L.I., Robinson, T.L., 2018. Long-term effects of tree density and tree shape on apple orchard performance, a 20 year study—part 1, agronomic analysis. *Scientia Horticulturae* 238, 303–317. doi:[10.1016/j.scienta.2018.04.033](https://doi.org/10.1016/j.scienta.2018.04.033).
- Makino, T., Yokoi, H., Kakazu, Y., 1999. Development of a motion planning system for an agricultural mobile robot, Society of Instrument and Control Engineers (SICE). pp. 959–962. doi:[10.1109/sice.1999.788679](https://doi.org/10.1109/sice.1999.788679).
- Mier, G., Valente, J., Bruin, S.D., 2023. Fields2cover: An open-source coverage path planning library for unmanned agricultural vehicles. *IEEE Robotics and Automation Letters* 8, 2166–2172. doi:[10.1109/LRA.2023.3248439](https://doi.org/10.1109/LRA.2023.3248439).
- Noguchi, T., Terao, H., 1997. Path planning of an agricultural mobile robot by neural network and genetic algorithm. *Computers and Electronics in Agriculture* 18. doi:[10.1016/s0168-1699\(97\)00029-x](https://doi.org/10.1016/s0168-1699(97)00029-x).
- Oksanen, T., 2007. Path planning algorithms for agricultural field machines. URL: <http://lib.tkk.fi/Diss/2007/isbn9789512290802/>.
- Plessen, M.M.G., Bemporad, A., 2017. Reference trajectory planning under constraints and path tracking using linear time-varying model predictive control for agricultural machines. *Biosystems Engineering* 153, 28–41. doi:[10.1016/j.biosystemseng.2016.10.019](https://doi.org/10.1016/j.biosystemseng.2016.10.019).
- Qin, Z., Chen, X., Hu, M., Chen, L., Fan, J., 2020. A novel path planning methodology for automated valet parking based on directional graph search and geometry curve. *Robotics and Autonomous Systems* 132, 103606.
- Reeds, J.A., Shepp, L.A., 1990. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics* 145. doi:[10.2140/pjm.1990.145.367](https://doi.org/10.2140/pjm.1990.145.367).
- Sabelhaus, D., Lammers, P.S., Helligen, L.P.M.Z., Röben, F., 2015. Path planning of headland turn manoeuvres. *Landtechnik* 70, 123–131. doi:[10.1515/lt.2015.2666](https://doi.org/10.1515/lt.2015.2666).
- Sabelhaus, D., Röben, F., zu Helligen, L.P.M., Lammers, P.S., 2013. Using continuous-curvature paths to generate feasible headland turn manoeuvres. *Biosystems Engineering* 116, 399–409. doi:[10.1016/j.biosystemseng.2013.08.012](https://doi.org/10.1016/j.biosystemseng.2013.08.012).
- Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., Pan, J., Patil, S., Goldberg, K., Abbeel, P., 2014. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research* 33, 1251–1270.
- Spekken, M., de Bruin, S., 2013. Optimized routing on agricultural fields by minimizing maneuvering and servicing time. *Precision Agriculture* 14, 224–244. doi:[10.1007/s11119-012-9290-5](https://doi.org/10.1007/s11119-012-9290-5).

- Tu, X., Tang, L., 2019. Headland turning optimisation for agricultural vehicles and those with towed implements. *Journal of Agriculture and Food Research* 1, 100009. doi:[10.1016/j.jafr.2019.100009](https://doi.org/10.1016/j.jafr.2019.100009).
- Verschueren, R., Frison, G., Kouzoupis, D., Frey, J., van Duijkeren, N., Zanelli, A., Novoselnik, B., Albin, T., Quirynen, R., Diehl, M., 2022. acados—a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation* 14. doi:[10.1007/s12532-021-00208-8](https://doi.org/10.1007/s12532-021-00208-8).
- Vougioukas, S., Blackmore, S., Nielsen, J., Fountas, S., 2006. A two-stage optimal motion planner for autonomous agricultural vehicles. *Precision Agriculture* 7, 361–377. doi:[10.1007/s11119-006-9022-9](https://doi.org/10.1007/s11119-006-9022-9).
- Vougioukas, S.G., 2019. Annual review of control, robotics, and autonomous systems agricultural robotics. *Annu. Rev. Control Robot. Auton. Syst.* 2019 2, 365–392. URL: <https://doi.org/10.1146/annurev-control-053018->, doi:[10.1146/annurev-control-053018-](https://doi.org/10.1146/annurev-control-053018-).
- Zhang, X., Liniger, A., Borrelli, F., 2020. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology* 29, 972–983.
- Zhou, J., He, R., Wang, Y., Jiang, S., Zhu, Z., Hu, J., Miao, J., Luo, Q., 2020. Autonomous driving trajectory optimization with dual-loop iterative anchoring path smoothing and piecewise-jerk speed optimization. *IEEE Robotics and Automation Letters* 6, 439–446.