ADVANCED QUADRATIC OPTIMIZATION BASED SMOOTHING FRAMEWORK FOR CURVED PATH PLANNING IN SUSTAINABLE AGRICULTURAL OPERATIONS

WEERAHANNEDIGE DANANJALA POKURUMALIE FERNANDO
Bachelor of Science, University of Sri Jayewardenepura, Sri Lanka, 2022

A thesis submitted
in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

Department of Mathematics and Computer Science
University of Lethbridge
LETHBRIDGE, ALBERTA, CANADA

ADVANCED QUADRATIC OPTIMIZATION BASED SMOOTHING FRAMEWORK FOR CURVED PATH PLANNING IN SUSTAINABLE AGRICULTURAL OPERATIONS

WEERAHANNEDIGE DANANJALA POKURUMALIE FERNANDO

Date of Defence: December 16, 2025

Dr. S. Hossain Co-Supervisor	Professor	Dr. Scient.
Dr. S. Das Co-Supervisor	Professor	PhD.
Dr. R. Benkoczi Thesis Examination Committee Member	Professor	PhD.
Dr. P. Ghazalian Thesis Examination Committee Member	Professor	PhD.
Dr. W. Osborn Chair of the Thesis Examination	Associate Professor	PhD.

Abstract

This research introduces an advanced quadratic optimization framework for the creation of smooth, curvature-continuous, and steerable waylines in precision agriculture. The framework combines curvature control, coverage uniformity, and steering feasibility into a single mathematical model to reduce skips and overlaps during field operations. Three smoothing techniques; B-spline, Bézier, and Non-Uniform Rational B-Spline (NURBS) are applied and compared to determine their impact on curvature continuity and field coverage efficiency. An extensive sensitivity analysis is performed to understand the effect of swath width tolerance, angular deviation, objective function cost weights, number of characteristic points of the spline curve, and solver configurations using IPOPT and Gurobi. Additionally, a versatile reference-line propagation method enables waylines to be created from either the field boundary or an interior line, thus, increasing flexibility in irregular geometries. The findings reveal that the NURBS-based method offers the greatest geometric flexibility and the most effective field utilization when compared to the other tested methods. The framework is shown to be performant and stable across irregular real-world fields, consequently, it is a scalable and practical solution for optimization-driven path planning in agriculture aiming at sustainability.

Acknowledgments

I would like to extend my deepest gratitude to my thesis supervisor, Dr. Shahadat Hossain, and my co-supervisor, Dr. Saurya Das, for their continuous guidance, encouragement, and invaluable support throughout this research. I also extend my sincere gratitude to Dr. Robert Benkoczi and Dr. Pasqual Ghazzelion for serving on my examination committee and for their thoughtful feedback and encouragement. The collaboration with Verge Ag made this research possible. The support of the Mitacs Accelerate Program and, along with that, the Verge Ag expertise and industry insights, greatly facilitated the development and the validation of the practical aspects of this study. I am grateful to my parents, my two sisters, and all my family members for their endless love, sacrifices, and belief in me. Their support has been the foundation upon which this achievement stands. My heartfelt thanks also go to my friends and relatives for their constant encouragement and kindness throughout this journey. Last but not least, I dedicate this work to my beloved husband, whose patience, understanding, and unwavering support have been my greatest strength. This journey would not have been possible without the guidance and support of all those mentioned above. I am truly grateful.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Precision Agriculture and Field Operations

Agriculture remains one of the most challenging industries. One of the main reasons for this is the ever-increasing global population that requires more and more food production but at the same time, the natural resources are limited. Meeting such challenges without fail necessitates innovative solutions that help to increase the output and at the same time, cut back on the wastes and the negative impact on the environment. Accordingly, precision agriculture has become one of the main pillars of such a strategy as it offers technology-enabled and data-driven solutions that can enhance the efficiency of agricultural operations. Today, many agricultural practices depend on the use of Global Navigation Satellite System (GNSS)-based guidance systems, automation, and machine control technologies that are geared towards optimizing field operations, reducing input usages, ensuring that the most effective utilization of the available resources is carried out. In reality, agriculture production is based on doing a series of field operations which include plowing, seeding, fertilizing, spraying, irrigating, and harvesting. Every operation needs thorough coverage of the field to guarantee a uniform application of the inputs and to increase the output efficiency and consistency. The power of these operations is determined not only by the machinery performance but also by how the operations are systematically organized within the field. If these operations are inefficiently planned, it may lead to the wastage of resources, the increase of fuel consumption, needless compaction of the soil caused by the doubling of routes, and finally, lose in yields due to the fact that some parts may be under-

treated or over-treated. Therefore, the organizing of field operations has a major impact on making sure that they will be carried out in a way, which is not only effective but also considerate of the environment. The concept of systematic field coverage is not complicated to understand, however it is a lot more complicated when the actual implementation is done under real-world conditions. Usually, agricultural fields are not in the shape of a rectangle or a square and are not uniform either; in fact, they most often have irregular edges shaped by topography, land ownership patterns, and natural features. Apart from this, fields may also contain some internal barriers such as bodies of water, tree lines, drainage channels, or built structures that may hinder the continuity of operations. Such irregularities turn out to be significant challenges in the planning of efficient paths as the traditional methods such as drawing simple contour lines or A-B lines can not always ensure full coverage without leaving some areas unworked, overlapping, or inefficient maneuvers. Addressing these issues requires more advanced planning methods that are capable of adapting to the geometry of the field while still ensuring smooth, continuous, and resource-efficient operations. The critical point of planning field operations is to generate routes for agricultural machines that can commute to execute different operations. Such routes can be split into two groups: (i) operational tracks, i.e., the paths along which activities can be directly carried out for input application or harvesting, and (ii) non-operational maneuvers, for example, turning between successive tracks or traveling to stations where refilling and unloading are done. The present research focuses primarily on the operational track generation process, as this component is fundamental to ensuring efficient coverage of the field. The idea of designing the operational tracks is far from being straightforward: in a number of instances, they cannot be represented as simple straight lines because of the irregularity of the fields in the real world. Instead, the task is to create curved, continuous tracks that machines can follow without any difficulty and optimized to reduce skips, overlaps, and the inefficient traversing of areas.

1.2 Problem Statement

In modern farming, how the machines travel through the field is directly linked to the efficiency, operating costs, and the crop yield. To a great extent, the effectiveness of such a movement depends upon the design of the tracks, as they are the ones that define the area of the field that can be covered, the count of the turning operations, and the capability of the devices to follow the planned routes.

The company collaborating in this research provides GNSS-based guidance systems for agricultural machinery to perform operations such as seeding, spraying, and harvesting. Currently,it utilizes a geometric procedure to generate tracks. A smooth reference line is first defined, usually based on the field boundary or specified by the user. From this reference, parallel tracks are created through geometric offsetting (for example, using GEOS line offsetting). The GEOS (Geometry Engine – Open Source) line offsetting operation generates a new line at a specified distance from an existing one, which is achieved by shifting the original line to the left (for positive distances) or to the right (for negative distances). The resulting offset line corresponds to the boundary of the buffer created around the input line. After the new line is generated, sharp angles beyond a specified threshold are smoothed using Bézier curves.

The biggest limitation in the offsetting algorithm used by the company is its strict assumption that tracks should never overlap. Although this avoids double coverage, it often leaves unworked strips between tracks, particularly in irregular fields. Moreover, the subsequent smoothing of sharp angles after offsetting can exacerbate this problem by creating even larger skipped areas between adjacent tracks and producing paths whose steerability is uncertain under real operating conditions. Importantly, farmers have indicated that in a some cases, a small amount of overlap may be allowed and even advantageous. As an illustration, responses from growers indicate that permitting overlaps up to a slightly maximum perpendicular distance can aid in the suppression of weed growth in untreated strips, thereby reducing the competition for nutrients. Apart from these problems, the paths cre-

ated do not consider the constraints of machinery steering, thus they usually require turning radii that are not feasible from an operational point of view. Lastly, the smoothing operation only changes the way the paths appearance from the visual point of view but it does not solve the steering feasibility, width-width consistency, or the handling of overlaps and skips problems which are deeper issues.

B-spline based quadratic optimization framework for wayline generation has been introduced in the literature. Although this technique lays a strong foundation, it is still quite constrained in terms of its extension to some real-world fields. Essentially, the method in its initial version is limited to the generation of waylines that only depart from a field edge, which may not always be possible in real-world situations and, even when it is, may not be lead to the most optimal results, especially in the case of irregularly shaped fields. Furthermore, the formulation lacks a systematic analysis of parameter sensitivity or weighting effects. When the method is being adjusted to real-world conditions, the tuning of such parameters is very important because the relative impact of path length, smoothness, and coverage on operational efficiency and solution quality.

In order to adapt the original method for the practical implementation of the proposed research, the research first applies the framework to real-world field data obtained from the industry partner and then extends it by introducing new smoothing techniques. These smoothing formulas are being compared systematically, as the degree of smoothness has a direct impact on the main performance metrics that are of interest; namely, skips, overlaps, and curvature continuity. Moreover, a detailed sensitivity analysis is performed to determine how changes in key parameters, such as offset distance, working width, and objective function weights, influence the final coverage quality and optimization results. Besides that, the method is being generalized to allow the generation of waylines not only from the edges of the fields but also from the reference lines that are situated in the interior of the field. The importance of this flexibility lies in the fact that the efficiency of coverage as well as the amount of skips and overlaps depend heavily on the selection of the reference line.

These extensions, in combination, help to close the gap between the theoretical formulation and its practical application in the real world, thus making it possible to have a more robust and adaptable framework for the performance of field operations that are not only reliable but also flexible.

1.3 Research Objectives

The primary objective of this study is to create smooth and steerable waylines for the company's agricultural fields by applying a new optimization-based method that surpasses their current methods. It is a frequent circumstance that the company's current track generating and smoothing operations lead to large areas of skips which the farmers find undesirable, and at the same time, if smoothing is neglected, there will be sharp turns that are hard to steer by heavy agricultural machinery. To fix this trade-off between coverage efficiency and path smoothness, the present work uses a quadratic optimization framework which is capable of generating continuous and easily steerable paths with the least possible skips and overlaps. Moreover, the study presents and compares the effectiveness of the different smoothing methods, i.e., Bézier, B-spline, and Non-Uniform Rational B-spline (NURBS) formulations, each providing different characteristics of the curvature control and flexibility. These methods are compared in a systematic way for their effectiveness in balancing the consistency with the reference line geometry as well as the operational smoothness and coverage efficiency. The execution is carried out using real field data provided by the industry partner so that the results are practically relevant. Furthermore, to conduct a comprehensive sensitivity analysis to determine how changes in the key parameters and solver configurations affect the optimization performance and the overall framework stability.

1.4 Research Questions

- How can efficient, feasible tracks be generated for irregular fields while minimizing the skips and overlaps?

- How do different smoothing techniques (Bézier, B-spline, and NURBS) influence the smoothness, curvature continuity, and coverage efficiency of the generated paths?
- How does reference line choice affect coverage, skips, and overlaps?
- Which solver and parameter choices provide efficient results?

1.5 Contributions

This study makes several significant contributions toward increasing the efficiency of agricultural wayline generation through optimization-based path planning. The main contributions can be summarized in the following points:

- Application of the quadratic optimization framework in real-world fields: The current optimization method was expanded and tailored to incorporate irregular boundaries and operational requirements of actual fields provided by the industrial partner, thus making the results usable in a practical setting and not only in theoretical ones.
- Integration of multiple curve-smoothing techniques: Besides the B-spline smoothing, the Bézier and NURBS curve formulations were utilized and tested for achieving higher geometric continuity and steering feasibility in different field geometries.
- Comprehensive parameter sensitivity and solver analysis: A detailed, systematic study was conducted to understand how key model parameters, e.g., such as angular offset tolerance, swath width deviation, number of spline points, and objective weight distribution alongside a comparative evaluation of different solvers (IPOPT and Gurobi) on skip, overlap, and computational performance.
- Flexible reference line initialization: The system was changed to allow the creation of the reference line from the edge of the field or the center point, providing more flexibility in terms of different working strategies and the field layout.

1.6 Thesis Organization

This thesis is arranged in five chapters, each focusing on a particular aspect of the study and, when combined together, leading to the design of an optimization framework for the generation of agricultural waylines that are smooth, steerable, and aware of the terrain. Chapter 1 introduces the background and motivation of the research, emphasizing the role of precision agriculture and the necessity for efficient and sustainable field coverage. Moreover, it points out the limitations of the existing geometric and optimization-based path generation methods, explains the research problem, and defines the objectives and research questions. The chapter wraps up with a brief of the main contributions and a synopsis of the thesis structure. Chapter 2 presents a comprehensive review of agricultural coverage path planning-related works. The chapter looks into geometric decomposition, curvature-constrained optimization, and spline-based trajectory generation, besides slope awareness and soil-erosion mitigation integration studies. The review of the literature sets the theoretical base and pinpoints the research gaps the current thesis has resolved. Chapter 3 details the designed optimization framework for generating smooth, steerable, and terrain-aware waylines. After introducing the agricultural terminology and describing the operation context, the chapter proceeds with the creation of the baseline two-dimensional quadratic optimization model for continuous-curvature wayline generation. Implementation of B-spline, Bézier, and NURBS smoothing techniques is also elaborated in the chapter. Chapter 4 describes the experiments carried out and the results obtained. The input data and solver configurations used for the evaluation are also discussed in the chapter. The chapter features extensive sensitivity and solver analyses that investigate how changes in parameters like swath width tolerance, angular deviation, number of spline points and weight distribution influence coverage performance as well as the effect of different solver. The chapter concludes with the comparative results of various smoothing techniques (B-spline, Bézier, and NURBS), and the flexible reference-line initialization experiments. Chapter 5 wraps up the thesis by reflecting on the main findings and contributions of the work and outlining

potential future directions.

Chapter 2

Related Work

Coverage Path Planning (CPP) is a crucial task in precision agriculture, the main goal is to keep the field covered entirely and efficiently, at the same time, to reduce the operational time, energy consumption, and soil degradation. Previous studies mostly relied on geometric and heuristic algorithms inspired by manual AB-line driving, a method in which the working machines follow straight, parallel tracks to cover an area systematically. Hameed *et al.* [2, 3] and Bochtis and Sørensen [4] formulated the problem of field coverage as a sequence optimization challenge and in doing so, they were the first to introduce the concepts of field blocks, headlands, and B-patterns to reduce non-operational travel and turning maneuvers. The pattern-based strategies, however, only worked well for convex and regular-shaped fields as they were limited to two-dimensional representation and did not take into account the vehicle kinematics or field topography, thus, leading to the area that was covered twice or even more and the inconveniences caused by the irregularities of the terrains or slopes. Subsequent geometric decomposition and heuristic algorithms, including boustrophedon and trapezoidal coverage, further advanced CPP by enabling full-area coverage under obstacle constraints [5, 6]. Oksanen and Visala [5] developed a trapezoidal decomposition framework that selected driving direction and subfield partitioning based on coverage efficiency, while Jin and Tang [7] introduced headland-first planning to minimize maneuvering losses and reduce overlapping tracks. These methods were well suited for structured environments but lacked adaptability to uneven terrain and vehicle motion limits. The practical operation of agricultural robots, however, requires

curvature-bounded and dynamically feasible paths, motivating research into nonholonomic path models and continuous-curvature formulations. Path feasibility and smooth maneuvering have been major considerations in modern CPP frameworks. Because agricultural machines are constrained by a minimum turning radius, Dubins and Reeds–Shepp curves have been widely used to connect field tracks with feasible transitions [8, 9, 10, 11]. Yet, such primitives introduce curvature discontinuities that cause abrupt steering variations. Backman and Oksanen [11] introduced a continuous-curvature turning model for agricultural vehicles that keeps the steering rate and acceleration changes within limits, thus producing smoother headland transitions. In the same manner, Sabelhaus *et al.* [12] created continuous-curvature turn methods to make the path more feasible and to reduce time losses at field boundaries. These developments have been the basis for curvature-constrained optimization that is used to generate smooth, physically feasible trajectories for autonomous agricultural operations. Spline-based parameterizations are now the standard choice to create curvature-continuous and differentiable paths. Elbanhawi *et al.* [13] found that cubic B-splines are capable of retaining continuity and enable curvature control while at the same time providing local control over the path segments. Besides, B-spline parameterizations permit the direct analytical calculation of curvature and slope that beat simple Bézier concatenations in smoothness and control accuracy [14, 15]. These features have been used in the navigation of autonomous ground and aerial vehicles, and in agricultural path smoothing for vehicle guidance and steering stability. This research is extending these benefits by bringing spline-based smoothness and curvature regularization right into a quadratic optimization framework. Non-uniform rational B-splines (NURBS) have been recognized as a higher-order generalization of B-splines, which can provide rational weighting and local geometric control that make it possible to generate both curvature-bounded and constraint-satisfying motions. *Jalel et al.* [16] proposed a path-planning framework where NURBS curves were not only employed for smoothing but were also fused into the optimization process itself. *Xie and Qin* [17] proposed an optimization approach that automatically de-

termines that NURBS knot vector by minimizing a smoothness (energy) functional, rather than choosing knots manually. *Höffmann et al.* [18] employed the principle to expand the use of NURBS in agricultural robotics by creating a nonlinear optimization problem for generating a headland path. In summary, research works have demonstrated that the rational weighting feature of NURBS can be used for curvature regulation and the generation of smooth path. The current study, propagates the idea to a convex quadratic optimization framework that models the three objectives of curvature smoothness, overlap and skip minimization. Optimization-based approaches to CPP perform path generation as a constrained optimization problem balancing smoothness, path length, and swath spacing objectives. Dong *et al.* [1] presented a quadratic programming model for continuous curvature wayline generation. Wan *et al.* [19] and Chien *et al.* [20] introduced cubic spline interpolation methods for the guidance of agricultural robots which helped in maintaining stable steering while improving path consistency across different terrains. These optimization frameworks marked a change from heuristic or geometric planning to continuous, curvature-bounded, and mathematically optimal trajectories. Nevertheless, the majority of these works have been confined to flat fields with no consideration of slopes or environmental sustainability. In fields with highly uneven surface, variations in swath spacing due to surface slope will result in non-uniform coverage, if two-dimensional coverage path planning (CPP) is used. Hameed [21] took the concept of coverage planning to three dimensions and introduced a side-to-side 3D method where terrain slope is taken into account for the generation of waylines in order to minimize skip and overlap areas. By using digital elevation models (DEMs) for the real surface projection, it is possible to have slope-consistent spacing and thus obtain better accuracy in field operations. Groen and Nørremark [22] also showed that using surface data for route optimization can have a great impact on the efficiency of the harvesting process. Besides coverage efficiency, environmental sustainability is becoming a major concern that is being addressed by precision agriculture. A number of recent publications pinpoint that the way the vehicle changes its track concerning the slope is a

major factor that determines soil erosion and runoff. Zhao et al. [23] and Spekken et al. [24] demonstrated that contour-aligned patterns have the potential to reduce the risk of erosion significantly as compared to the patterns that go straight downhill. Nevertheless, few optimization-based CPP models have been designed to consider erosion mitigation explicitly in their objectives. Optimization modeling environments like Pyomo [25] have made it possible to integrate geometric, kinematic, and environmental objectives into a single mathematical framework. Nonlinear solvers such as IPOPT and Gurobi [26] are turning fast solutions to large-scale, nonconvex agricultural optimization problems. Open-source CPP frameworks like Fields2Cover [27] have also been important in the reproducibility of agricultural robotics by the modularization of headland and swath planning. Nevertheless, these implementations are still primarily two-dimensional and have not considered the integration of slope- and erosion-aware objectives. In this way, the current study formulates a single Pyomo-based quadratic optimization model that regulates curvature-feasible smoothness. In summary, coverage path planning research has evolved from geometric decomposition to curvature-constrained and optimization-driven frameworks. However, there are still some gaps in adaptability to different terrains and eco-friendliness of the environment. The framework presented in this thesis overcomes these drawbacks by integrating smooth quadratic optimization, thereby advancing the state of of coverage path planning for autonomous agricultural systems.

Traditional coverage path planning methods usually depend on heuristic lane generation or post-processing smoothing processes. However, this thesis presents agricultural track generation as a unified optimization framework. The method is therefore directly minimizing operational-quality losses, e.g., skips/overlaps, while also abiding by machine-feasible curvature constraints, hence, it is guaranteed to be implementable. One of the main contributions is the thorough experimental comparison of B-splines, Bézier, and NURBS representations, alongside the novel curvature-aware NURBS weighting strategy which is capable of significantly improving local conformance in tight turns. The framework has

been verified on irregular-shaped geometries of fields and solvers typical to such problems, IPOPT and Gurobi, have been used for the solutions. The results demonstrate that the proposed framework achieves superior boundary adherence and coverage efficiency compared to purely geometric methods.

Chapter 3

Optimization Framework for Smooth and Steerable Wayline Generation

Efficient field coverage in advanced precision agriculture is largely dependent on the creation of waylines that are smooth, continuous paths which indicate the route of operation for agricultural machinery like seeders, sprayers, and harvesters. In general, these machines are of considerable size, have a wide working width and a large turning radius, thus they are very sensitive to curvature of the path and the distance between the paths. In such cases where the curvature of the waylines is high or the waylines are poorly spaced can lead to unsteerable turns, excessive overlap, or skipped regions, which in turn directly influence the efficiency of the operation and the consumption of fuel. The industrial collaborator involved in this research has a wayline generation method based on geometry with less complexity. The first step in the operation is the use of a reference line which is either taken from the field boundary or is typed in by the user. The next waylines are made by the GEOS (Geometry Engine Open Source) offsetting operation, which generates new lines at a certain lateral distance by moving the original line to the left or right, according to the direction of the offset. Once the offsetting has been done, the Bézier smoothing is used to sharp corners and to make the curves smooth which is also the requirement for the automatic steering systems. The method presents has a number of limitations. The reason is that the offsetting operation used to produce the tracks does not take into account curvature or length constraints, so the tracks are usually not controlled for curvature, thus the turns can be beyond the feasible steering limits of the machine. In addition, just straightforward

offsetting cannot change the distances between the neighboring passes especially on the curved parts so that there can be some regions which are skipped or overlapped and thus coverage accuracy is compromised. In those cases where the offset lines are subsequently smoothed, the problem can become even more severe because the curvature changes usually increase the spacing errors between the neighboring passes. In order to overcome such obstacles, this study upgrades the current optimization-based framework for the generation of waylines with the aim of introducing innovative smoothing techniques that help to improve curvature control, continuity, and general path geometry. As a result, the final method establishes a single framework for the generation of smooth, steerable, and operationally feasible waylines which can conform to the actual field topography with the least possible coverage losses. The next section provides the basic terms and parameters that are used in this research.

3.1 Agricultural Terminology and Operational Context

Various local terms are used in precision farming to explain the geometry as well as the functioning of the machinery paths. Understanding these geometric features is a prerequisite for the formulation of the path optimization problem. Wayline is a track that has been planned for the vehicle to follow, that is used to carry out a field operation such as seeding, spraying, or harvesting. The working width or the swath width (implement width) is the width of the agricultural machine that can be used effectively for one pass. The turning radius shows the least curve that the vehicle can make during turns. A field is a definition of an area where agricultural operations are carried out. The outer edge of this area called the field boundary, specifies the spatial extent of the movement of the machinery and is a polygon in the computational model. The waylines together make a route or path plan. Any route therefore has a start point and an end point that are defined, and can also consist of working time as well as non-working time (the time during which the machine is maneuvering or transiting between tracks and headlands). Headlands are essentially the outer

courses around the field boundary that offer room for turning equipment. A headland is usually only driven at the start or the finish of the operation and is vital for the easy change of the interior tracks. The term headland travel or headland moving is used in relation to journeys along these headlands which are made up of non-operational maneuvers. A waypoint can be understood as a geographic coordinate that determines a target location along a route. Sequential waypoints thus represent a seamless navigation path, which is how the vehicle is made to follow the right track. In the optimization model, these are basically the instances of the continuous reference line or the smoothed trajectory. The very first waypoint is the trace of the route start, i.e., the place where the machine is going to be loaded with the task, and the final waypoint is the establishment of the route end. Field elevation defines the height of the ground surface in relation to a reference level, while slope refers to the steepness or incline of the land, which can be calculated from changes in elevation. Increasing slope indicates that the ground is going up or down more steeply, thus affecting the stability of the machinery and the impact on the soil. Heavy slope in the direction of the travel can also cause soil erosion, i.e. the loss or movement of the top layer of soil due to water runoff or wind, which is generally intensified on steep or bare slopes. The term precision agriculture is a technology-based management strategy that uses detailed, site-specific information derived from geospatial data, sensor systems, and automation technologies to achieve greater field efficiency and sustainability. The optimization framework presented in this thesis is an example of such a system, which creates locally optimal, data-driven guidance paths (often referred to as guidance lines or AB lines) for GPS-based vehicle control systems. After defining the important geometric terms, it is necessary to explain the mathematical construction of smooth and continuous waylines. As the continuity and curvature of these trajectories have a significant impact on the steering capability of agricultural machinery, it is very important to use the correct curve representations. Hence, this chapter investigates the main methods of smoothing employed in this study starting with clamped B-splines and Bézier representations, which are the basis for the later NURBS-based ex-

tension.

3.2 Smoothing Techniques for Wayline Generation

One of the essential features of vehicle motion that is also necessary for accurate field coverage is the generation of smooth and continuous waylines. The resulting trajectories' curvature and continuity change significantly depending on the mathematical formulation used for smoothing. Hence, it is crucial to find appropriate curve representations that can be combined with the optimization framework. Consequently, this chapter deals with the possible smoothing methods starting with clamped B-splines and then moving to Bézier that offer the mathematical basis for creating the steering of the vehicle and geometrically consistent waylines.

3.2.1 B-spline Smoothing

The difference in the geometry of the path resulting from the B-spline framework can be very large just by the different interpretation and selection of spline parameters such as knot vectors, polynomial degree, control points, and basis functions. The features of the obtained route in terms of smoothness, continuity, and flexibility are the result of these components taken together. Thus understanding the idea and producing the mathematical B-splines is the main point in order to have the possibility of changing and adjusting the manner to the most effective one.

Definition of B-splines

A B-spline of degree k is one that is determined step-by-step by the Cox–de Boor formula.

For $k = 0$ (piecewise constants), the basis functions are

$$N_{i,0}(u) = \begin{cases} 1, & \text{if } u_i \leq u < u_{i+1}, \\ 0, & \text{otherwise.} \end{cases}$$

For $k \geq 1$, the basis functions are defined as

$$N_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u),$$

where $\{u_i\}$ is the non-decreasing knot sequence.

B-splines are k degree piecewise polynomials that have up to $k-1$ degree continuity at the knots, thus they represent smooth and numerically stable curves.

B-spline Fundamentals

A B-spline curve is characterized by four components: the knot vector, the basis functions, the spline's degree, and the control points.

- **Knot Vector:**

  The *knot vector* is a non-decreasing sequence of real numbers that divides the B-spline's parametric domain into intervals, determining where and how the piecewise polynomial segments connect.

  $$U = \{u_0, u_1, \ldots, u_m\}, \quad u_i \leq u_{i+1},$$

  Its structure strongly influences the smoothness, endpoint behavior, and overall shape of the resulting curve. Depending on the arrangement and repetition of the knots, several categories can be defined. [28]

  - **Uniform Knot Vector:** In a uniform knot vector, all knots are evenly spaced:

    $$U = \{0, 1, 2, \ldots, m\}.$$

The variable m is the total number of knots in this sequence. The index m is not arbitrary; it is strictly defined by the number of control points $(n+1)$ and the degree (k) of the curve: $m = n+k+1$. This produces a consistent parameterization across the curve and equal influence of all control points. However, uniform spacing does not guarantee that the curve passes through the first and last control points. Uniform B-splines are suitable for periodic or smoothly repeating paths where strict endpoint interpolation is not required.

– **Open Uniform Knot Vector:** A special form of the uniform knot vector where the first and last knots are repeated $(k+1)$ times, with k being the degree of the spline:

$$U = \{\underbrace{0,\ldots,0}_{k+1}, 1, 2, \ldots, n-k, \underbrace{n-k+1,\ldots,n-k+1}_{k+1}\}.$$

The repeated endpoints *clamp* the curve to the first and last control points, ensuring interpolation at the boundaries, while the interior knots remain uniformly spaced. This configuration combines uniform parameter spacing with predictable endpoint behavior and is the most common form used in engineering and computer-aided geometric design.

– **Non-Uniform Knot Vector:** In a non-uniform knot vector, the spacing between knots is irregular:

$$U = \{u_0, u_1, \ldots, u_m\}, \quad \text{where } u_{i+1} - u_i \text{ may vary.}$$

Non-uniform distributions allow finer local control over curve shape regions with closely spaced knots produce tighter curvature and reduced continuity, while widely spaced knots yield smoother transitions. This flexibility is useful when local geometric features require increased precision or curvature adaptation.

Clamped vs. Unclamped B-splines. In addition to knot spacing, the repetition of endpoint knots determines whether a B-spline is *clamped* or *unclamped*.

- **Clamped B-splines:** The first and last knots are repeated $(k+1)$ times, enforcing

$$\mathbf{C}(u_0) = \mathbf{P}_0, \quad \mathbf{C}(u_m) = \mathbf{P}_n.$$

  Thus, the curve goes through its first and last control points directly, which is required when waylines must start and end precisely on field boundaries or user-defined references.

- **Unclamped B-splines:** Endpoint knots are not repeated, thus the curve smoothly approach, but does not have to go through, its end control points. Such a setting ensures greater smoothness and can be used for periodic or continuous trajectories.

The present study uses an open uniform, clamped B-spline formulation. This method is preferred because it guarantees that the waylines created extend to the edges of the fields only and that the spacing remains uniform and the curvature is smooth throughout the field.

- **Degree:** The degree k is the polynomial order of the basis functions. Higher degree means more smoothness of the curves. Usually, $k = 2$ (quadratic) or $k = 3$ (cubic) are taken.

- **Control points:** A set of points $\mathbf{P}_i \in \mathbb{R}^d$ that outline the shape of the curve. Each point influences the curve locally.

- **B-spline curve equation:** The degree k with $n+1$ control points is written as,

$$\mathbf{C}(u) = \sum_{i=0}^{n} N_{i,k}(u)\,\mathbf{P}_i, \quad u \in [u_k, u_{m-k}].$$

This weighted sum ensures local control, smoothness, and numerical stability.

e.g. Clamped B-Splines

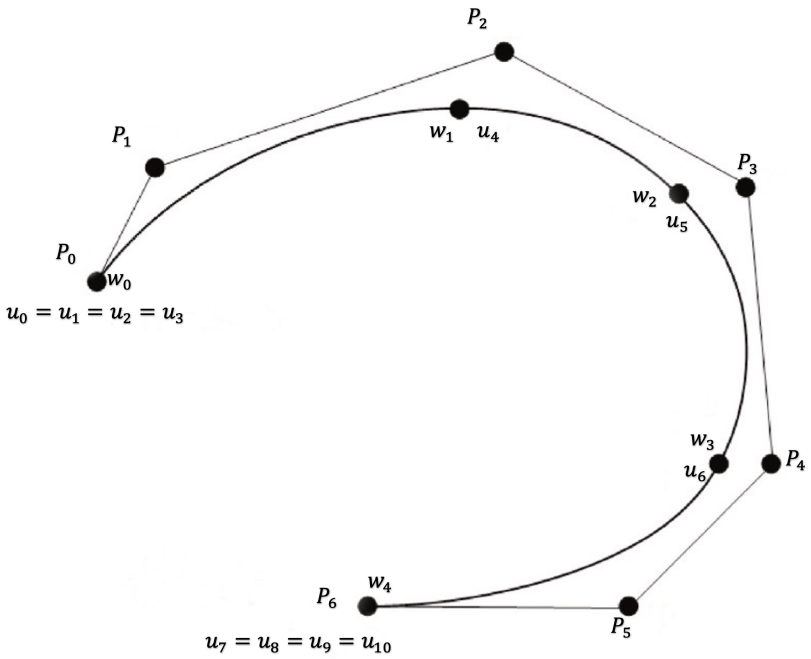


Figure 3.1: Clamped B-spline example. u_0 to u_{10} are knots, P_0 to P_6 are control points, w_0 to w_4 are way points.

Figure 3.1 shows an example of a clamped b-spline of degree(k) 3. The parametric clamped b-spline curve $C^i(u)$ is,

$$\mathbf{C}^i(u) = N_0(u)\,P_{i-1} + N_1(u)\,P_i + N_2(u)\,P_{i+1} + N_3(u)\,P_{i+2}, \qquad 0 \le t \le 1, \tag{3.1}$$

where $N_i(u)$ the basis functions,

$$N_0(u) = \frac{(1-u)^3}{6}, \qquad N_1(u) = \frac{4-6u^2+3u^3}{6}, \qquad N_2(u) = \frac{1+3u+3u^2-3u^3}{6}, \qquad N_3(u) = \frac{u^3}{6}. \tag{3.2}$$

3.2.2 Bézier Smoothing

The company collaborating in this research develops GNSS-based guidance systems for agricultural machinery that perform field operations such as seeding, spraying, and harvesting. In its internal software, Bézier-based smoothing is used to create smooth and continuous guidance paths for vehicle control. The optimization framework adopted in this study, initially employs B-spline smoothing to generate continuous waylines from discrete waypoints obtained after optimization. In order to enable a comparative evaluation between the two curve representations under the same optimization setting, this work implements the same optimization framework but replaces the B-spline smoothing stage with a Bézier-based formulation. This adaptation allows for a direct, quantitative, and qualitative comparison between Bézier and B-spline smoothing approaches in terms of path smoothness, curvature behavior, and suitability for agricultural field operations.

Bézier Curve Formulation

A Bézier curve defines a smooth path between a set of control points $\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n$ using Bernstein basis polynomials. The parameter $t \in [0,1]$ controls interpolation along the curve. The following formulations summarize the commonly used degrees [29].

Linear Bézier Curve (Degree 1). Given two control points $\mathbf{p}_0$ and $\mathbf{p}_1$, the linear Bézier curve is:

$$\mathbf{p}(t) = (1-t)\mathbf{p}_0 + t\mathbf{p}_1, \quad t \in [0,1]. \tag{3.3}$$

It represents a straight line between $\mathbf{p}_0$ and $\mathbf{p}_1$.

Quadratic Bézier Curve (Degree 2). For three control points $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$:

$$\mathbf{p}(t) = (1-t)^2\mathbf{p}_0 + 2t(1-t)\mathbf{p}_1 + t^2\mathbf{p}_2, \quad t \in [0,1]. \tag{3.4}$$

The curve passes through the endpoints $\mathbf{p}_0$ and $\mathbf{p}_2$.

Cubic Bézier Curve (Degree 3). Given four control points $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$:

$$\mathbf{p}(t) = (1-t)^3\mathbf{p}_0 + 3t(1-t)^2\mathbf{p}_1 + 3t^2(1-t)\mathbf{p}_2 + t^3\mathbf{p}_3, \quad t \in [0,1]. \tag{3.5}$$

The first and last control points define the endpoints, while $\mathbf{p}_1$ and $\mathbf{p}_2$ control the exit and entry tangents.

General Bézier Curve

The general Bézier curve of degree n is defined as:

$$\mathbf{p}(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \mathbf{p}_i, \quad t \in [0,1], \tag{3.6}$$

where $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ are the binomial coefficients.

Properties of Bézier Curves

1. **Endpoint Interpolation:** $\mathbf{p}(0) = \mathbf{p}_0$ and $\mathbf{p}(1) = \mathbf{p}_n$.

2. **Tangent Direction:** The tangents at the endpoints are given by

$$\mathbf{p}'(0) = n(\mathbf{p}_1 - \mathbf{p}_0), \quad \mathbf{p}'(1) = n(\mathbf{p}_n - \mathbf{p}_{n-1}). \tag{3.7}$$

3. **Convex Hull Property:** The entire curve lies within the convex hull of its control points, i.e., $\mathbf{p}(t) \in \mathrm{CH}(\mathbf{p}_0, \ldots, \mathbf{p}_n)$.

4. **Affine Invariance:** Applying any linear transformation T to the control points transforms the curve equivalently:

$$T[\mathbf{p}(t)] = \text{Bézier curve of } \{T(\mathbf{p}_i)\}.$$

5. **Smoothness and Differentiability:** A Bézier curve is smooth and differentiable for

all $t \in (0,1)$. When joining multiple Bézier segments, C^1 or C^2 continuity can be enforced by aligning tangents at the junctions.

Bézier-based Smoothing Implementation

Curve representation. Given control points $\{P_i\}$, the Bézier curve of degree n is

$$C(t) = \sum_{i=0}^{n} B_i^n(t)\, P_i, \qquad t \in [0,1], \quad B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}.$$

We evaluate $C(t)$ on a uniform grid $t_k = k/(N-1)$, $k = 0, \ldots, N-1$.

Derivatives and curvature. First and second derivatives are computed analytically from the control polygon:

$$C'(t) = n \sum_{i=0}^{n-1} B_i^{n-1}(t)\,(P_{i+1} - P_i), \qquad C''(t) = n(n-1) \sum_{i=0}^{n-2} B_i^{n-2}(t)\,(P_{i+2} - 2P_{i+1} + P_i).$$

Curvature is reported as

$$\kappa(t) = \frac{|x''(t)y'(t) - x'(t)y''(t)|}{\left(x'(t)^2 + y'(t)^2\right)^{3/2}}$$

The smoothing formulations described above establish the mathematical foundation for representing continuous and curvature controlled waylines. Curvature directly affects the vehicle's steerability. However, smoothness alone does not guarantee complete field coverage or operational feasibility. To systematically control curvature, spacing, and alignment between adjacent paths, these curve representations must be integrated within an optimization framework. The following section therefore presents the baseline 2D quadratic optimization formulation, which combines geometric constraints and coverage objectives to generate steerable and spatially consistent waylines across the entire field.

3.3 Baseline 2D Quadratic Optimization Formulation

The starting point of this research is a quadratic optimization framework proposed in the literature for generating waylines in two-dimensional agricultural fields [1]. This formulation treats wayline generation as an optimization problem, where a smooth reference line is propagated across the field while minimizing deviations and curvature subject to coverage constraints. The method represents each wayline as a B-spline curve, and the optimization objective balances coverage efficiency with smoothness to ensure that generated paths are feasible for machinery to follow.

The 2D wayline generation using quadratic optimization can be broken down into two main stages:

Stage 1: Reference Curve Generation

- One curved side of the field boundary is selected as the starting reference.
- This reference line $\mathbf{P}_{\text{ref}}$, will guide the creation of the first curved wayline.

Stage 2: Optimisation-Based Wayline Generation

Each new wayline is generated by solving a quadratic programming problem based on the previous one. The decision variables are the 2D coordinates of waypoints (x_i, y_i) along the curve.

Objective Function

The total cost to minimise is a weighted sum of three components:

$$\min w_{Straightness} \cdot \text{cost}_1 + w_{Length} \cdot \text{cost}_2 + w_{Skip/overlapWidth} \cdot \text{cost}_3 \tag{3.8}$$

Where:

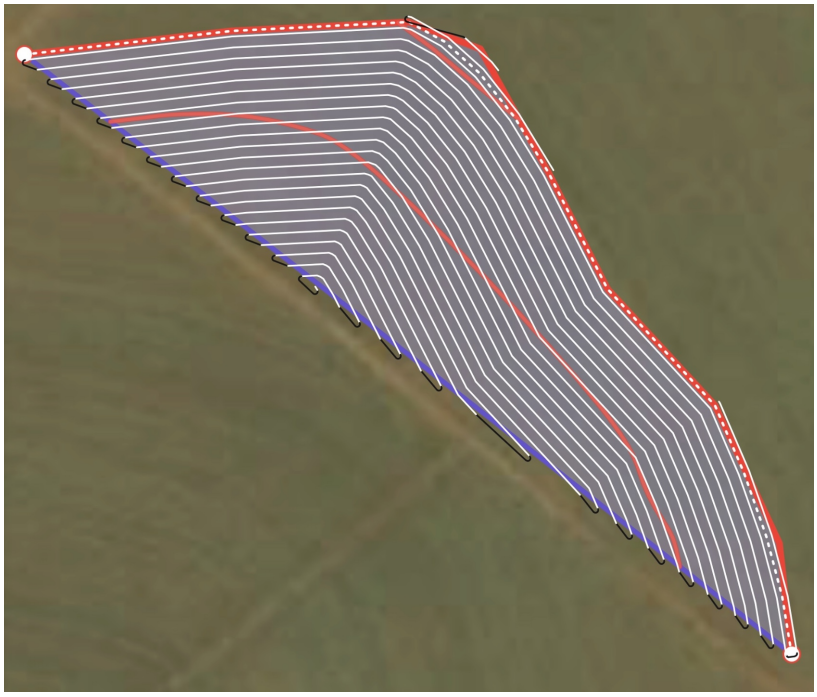


Figure 3.2: Field example showing the reference line (red), generated waylines (white), and field area (purple), extracted from the field management system of the partnering company. The waylines shown here were generated using company's existing workflow.

- **Straightness cost:** encourages smoother paths by penalizing sharp angles between consecutive points:

$$\text{cost}_1 = \sum_{i=1}^{n-2} \left[(x_{i-1} + x_{i+1} - 2x_i)^2 + (y_{i-1} + y_{i+1} - 2y_i)^2 \right] \tag{3.9}$$

- **Length cost:** promotes shorter, more direct paths overall paths:

$$\text{cost}_2 = \sum_{i=0}^{n-2} \left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 \right] \tag{3.10}$$

- **Skip/overlap cost:** minimizes spacing errors between adjacent passes by aligning new waylines closely with a set offset distance (usually equal to the implement; effective working width of the agricultural tool):

$$\text{cost}_3 = \sum_{i=0}^{n-1} \left[(x_i - (x_i^{\text{ref}} + \hat{e}_x^i \cdot w))^2 + (y_i - (y_i^{\text{ref}} + \hat{e}_y^i \cdot w))^2 \right] \tag{3.11}$$

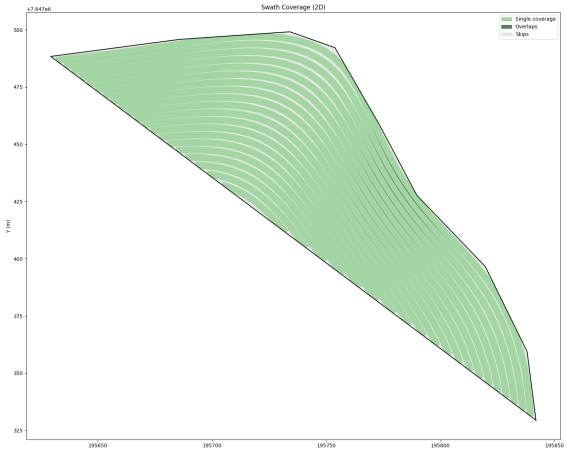


Figure 3.3: Illustration of skip and overlap between adjacent passes.

Here, $\hat{e}_x^i$, $\hat{e}_y^i$ are components of the unit normal vector at point $\mathbf{P}_i^{\text{ref}}(x_i^{\text{ref}}, y_i^{\text{ref}})$, and w is the implement width (working width).

Constraints

- **Swath width constraint:** controls how far each point on the new wayline can deviate from the ideal offset distance.

$$(1-\gamma)w \leq \frac{x_i - x_i^{\text{ref}}}{\hat{e}_x^i} \leq (1+\gamma)w \tag{3.12}$$

$$(1-\gamma)w \leq \frac{y_i - y_i^{\text{ref}}}{\hat{e}_y^i} \leq (1+\gamma)w \tag{3.13}$$

  Here, (x_i, y_i) are the optimized waypoints, $(x_i^{\text{ref}}, y_i^{\text{ref}})$ are points on the reference line, and $\hat{e}_x^i, \hat{e}_y^i$ are the components of the unit normal vector at point i, and γ controls the allowed deviation from the ideal implement width. It defines the tolerance range for how far the new wayline can be offset from the reference path. Typical values are between 0.00 and 1.00.

- **Offset direction constraint:** ensures the direction of the offset stays within a limited

angle from the normal of the previous line.

$$\frac{y_i - y_i^{\text{ref}}}{x_i - x_i^{\text{ref}}} \in \left[\frac{\hat{e}_y^i}{\hat{e}_x^i} - \mu, \frac{\hat{e}_y^i}{\hat{e}_x^i} + \mu \right] \tag{3.14}$$

Here, the parameter μ controls the extent to which the direction of the offset vector can differ from the normal direction of the reference line. In this way, a smooth and consistent spacing pattern is preserved while there is still some freedom in the case of sharp or curved regions.

- **Curvature constraint:** enforces a maximum turning curvature based on the vehicle's physical limits (e.g. minimum turning radius),Curvature at a point is approximated as:

$$k_i = \frac{(x_{i-1} + x_{i+1} - 2x_i)^2 + (y_{i-1} + y_{i+1} - 2y_i)^2}{d_s^2} \tag{3.15}$$

$$(x_{i-1} + x_{i+1} - 2x_i)^2 + (y_{i-1} + y_{i+1} - 2y_i)^2 \leq (d_s^2 \cdot k_{\max})^2 \tag{3.16}$$

Where:

- d_s = gives the average spacing between waypoints
- $k_{\max} = 1/R_{\min}$ = is the maximum allowable curvature, where $R_{\min}$ is machine's minimum turning radius

First, the optimisation problem is solved to get the waypoints. These waypoints are then interpolated by using clamped cubic B-splines to produce a continuous, smooth curve. The curve is within the kinematic constraints of the machine, and it is capable of accurate and efficient coverage of the field. The quadratic wayline generation method for the baseline in this research was chosen because of its proven effectiveness and reliability, as has been demonstrated in prior work. As per the initial research, the optimisation-based formulation effectively balances smoothness, skip, and overlap by integrating curvature and coverage objectives into a single quadratic programming framework. The approach goes on to sim-

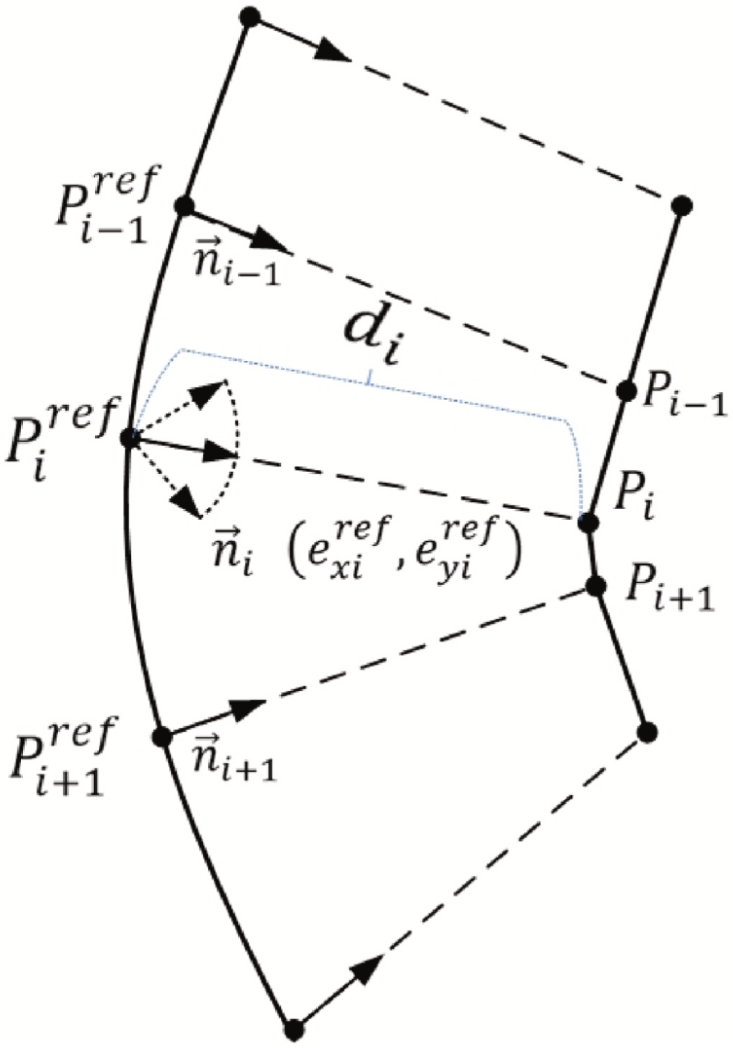


Figure 3.4: Illustration of two consecutive tracks [1].

plify the curvature constraint to a linear one, thus allowing the computation to be carried out while the path is still continuous and the vehicle is still steerable. Results of experiments in the referred paper achieved coverage of more than 97%, overlap less than 0.8%, and skip less than 2.3%, thus greatly exceeding the performance of a commercial navigation system (AllyNav) while at the same time producing smoother trajectories with the curvature being reduced up to 77%. Nevertheless, these results have only been validated on a small number of test fields, mostly of simple geometry, and consequently, they should be considered as initial confirmation rather than a thorough evaluation. This algorithm, therefore, constitutes a suitable and tested baseline for our work, given the evidence obtained. We have taken this

baseline further in the next section by considering more complex and irregular field geometries to investigate the effect of boundary shape and layout on coverage efficiency and path feasibility. This step is driven by practical requirements from farmers, who are most concerned with minimizing skips and overlaps while ensuring maximum field coverage with smoothly steerable waylines.

3.4 Extension of Clamped B-Splines to NURBS for Enhanced Curvature Control and

Non-Uniform Rational B-Splines (NURBS) smoothing is an advanced and adaptable mathematical technique for creating smooth and continuous curves or surfaces from a set of discrete data points or control vertices. It basically takes the standard B-spline system one step further by adding non-uniform parameterization and rational weighting, thus allowing not only the accurate representation of intricate freeform geometries but also the mathematically generated shapes like circles and ellipses. With NURBS smoothing, the curve is represented using a mix of control points, weights, basis functions, and a knot vector that specifies the parameter space. Such a framework grants local control and adjustment of the curvature to even a small portion of the curve without the need to change the whole curve, thus NURBS are perfect to be used for path generation, surface modeling, and boundary reconstruction in engineering and geometric optimization problems.

A NURBS curve $C(u)$ of degree p is mathematically expressed as:

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\, w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u)\, w_i}, \quad u \in [u_0, u_m] \tag{3.17}$$

where:

- P_i: control points defining the general shape of the curve.
- w_i: weights associated with each control point, influencing the curve's pull towards that point.

- $N_{i,p}(u)$: B-spline basis functions of degree p, constructed recursively using the knot vector.
- $\{u_0, \ldots, u_m\}$: knot vector that divides the parameter domain and determines the influence range of each control point.

The principal parts of NURBS smoothing are described below:

Control Points

The control points (or control vertices) are the points that visually represent the foundation of the curve. Essentially the curve is "pulled" towards every individual control point thereby shaping the whole design. The count as well as the location of the points decide whether the curve is more like a flexible or a rigid one. More control points mean more precise local control although too many points could make the computations more complex.

Knots and the Knot Vector

A knot is a scalar value that splits the parametric domain into intervals. The array of knots, called a knot vector, specifies how the curve blends between control points. The distribution and multiplicity of knots influence the smoothness of the curve. A non-uniform knot vector can have uneven distances between knots and can therefore have characteristics like sharp corners or local refinements.

Basis Functions

B-spline basis functions are the functions which decide the blending mathematically between control points. They are piecewise polynomial functions that guarantee smooth transitions between spans. The degree of the basis function (p) decides the smoothness of the curve: linear ($p = 1$) for straight segments, quadratic ($p = 2$) for gentle bends, and cubic ($p = 3$) for extremely smooth curves, which are mostly used in path optimization.

Weights

Each control point has an associated weight that adjusts its influence on the curve. By a larger weight the curve is attracted more to the control point, while if the weight is smaller then the effect is less. Through the adjustment of these weights NURBS are able to be the exact models of any conic like circle and ellipse

Degree and Order

The degree of a NURBS curve is the highest power of the polynomial in its basis functions that is directly affect the curves smoothness and curvature. The order of the curve is degree plus one (Order $=$ Degree $+ 1$). Higher degree curves provide smoother and more flexible shapes, however, they also increase the computational cost. In fact, cubic ($p = 3$) NURBS curves are most commonly used because they are flexible and efficient at the same time.

Spans

A span is the part of a NURBS curve that lies between two consecutive knot values. A span is uniquely determined by a set of control points and basis functions, thus it represents a small portion of the entire curve. The NURBS curve can be considered as the result of multiple spans that are smoothly connected.

Advantages of NURBS Smoothing:

- Provides both local and global shape control.
- Can represent even the most complicated geometrical shapes.
- Uses non-uniform knots to guarantee smoothness of the curve and to allow increasing or decreasing of the parameter.
- Enables efficient and smooth curve generation suitable for optimization and path planning tasks.

In this research, NURBS smoothing is employed to refine reference lines and operational boundaries, producing curvature-continuous and geometrically accurate paths for coverage optimization. The resulting smoothed trajectories minimize discontinuities and enhance the feasibility and quality of autonomous field operations. [18, 16]

3.5 NURBS Representation and Curvature Evaluation

In this work, each reference or fitted path is represented as a Non-Uniform Rational B-Spline (NURBS) curve of degree p. A NURBS curve $C : [u_p, u_{m-p}] \to \mathbb{R}^2$ is defined by control points $\{P_i\}$, weights $\{w_i\}$, and a knot vector $\mathbf{U} = \{u_0, \ldots, u_m\}$:

$$C(u) \;=\; \frac{\sum_{i=0}^{n} N_{i,p}(u)\, w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u)\, w_i}, \qquad u \in [u_p,\, u_{m-p}], \tag{3.18}$$

where $N_{i,p}(u)$ are the B-spline basis functions of degree p defined on $\mathbf{U}$ (Cox–de Boor recursion).

Knot Vector (Clamped, Non-Uniform)

We use a *clamped* knot vector, i.e., the first and last knots have multiplicity $p+1$, which guarantees interpolation at the endpoints:

$$\mathbf{U} \;=\; \big[\underbrace{u_0, \ldots, u_0}_{p+1},\; u_{p+1}, \ldots, u_{m-p-1}, \underbrace{u_m, \ldots, u_m}_{p+1}\big].$$

Control Points

The control points $\{P_i\}$ are obtained by constructing a cubic (p=3) interpolatory B-spline through the input samples.

Rational Weights

To improve local accuracy in areas where the control polygon exhibits sharp bends, we apply curvature-aware weights:

$$w_i \;=\; 1 \;+\; \alpha\,\frac{\angle(P_{i-1},P_i,P_{i+1})}{\pi}, \quad i=1,\dots,n-1, \qquad w_0 \leftarrow w_0(1+\beta),\;\; w_n \leftarrow w_n(1+\beta), \tag{3.19}$$

where $\angle(P_{i-1},P_i,P_{i+1}) \in [0,\pi]$ is the turning angle of the control polygon at P_i, $\alpha > 0$ controls curvature emphasis, and $\beta > 0$ adds mild endpoint anchoring. This keeps the interpolation intact but increase the local pull in high-turn regions.

Basis Functions (Cox–de Boor)

The B-spline basis $N_{i,p}(u)$ on $\mathbf{U}$ is defined recursively by

$$N_{i,0}(u) = \begin{cases} 1, & u_i \le u < u_{i+1}, \\ 0, & \text{otherwise}, \end{cases} \qquad N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u)$$

Curvature

Curvature of a curve is a measure of the sharpness with which it bends at a point.

It is defined as the rate of change of the tangent direction. For a parametric curve $C(u) = (x(u), y(u))$, the curvature is expressed as:

$$\kappa(u) = \frac{\left|\, x'(u)\,y''(u) - y'(u)\,x''(u)\,\right|}{\left(x'(u)^2 + y'(u)^2\right)^{3/2}}. \tag{3.20}$$

A larger curvature value indicates a sharper turn, while a smaller value corresponds to a smoother or straighter portion of the curve.

Chapter 4

Results and Discussion

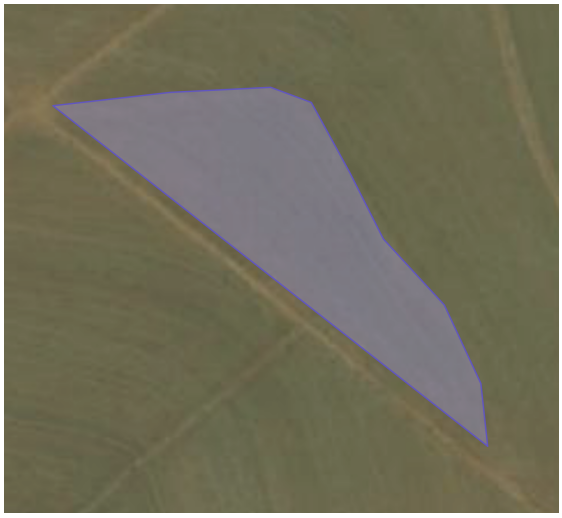
4.1 Input Data and Preprosessing

The raw input data are provided as JSON files exported from the agricultural field management system of the industrial collaborator. Each JSON file contains the geographic and operational information required for path planning and field analysis.

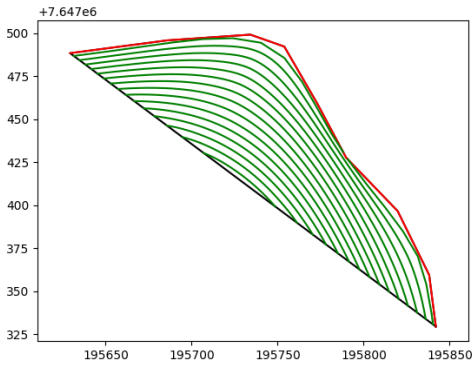
For the purpose of this research, the relevant attributes are extracted from these JSON files and converted into a structured CSV format, which serves as the standardized input to the proposed workflow. Because agricultural machinery and optimization algorithms require inputs in metric units, the geographic coordinates originally provided by the company in the WGS84 geographic coordinate system (degrees) must be transformed into a projected coordinate system (meters).

For this research, the relevant attributes are extracted and converted into structured CSV files, which serve as the standardized inputs to the proposed optimization workflow. The two primary inputs are described as follows:

- **Field boundary:** Defines the outer perimeter of the agricultural land. Boundaries may consist of a single polygon (outer ring) or multiple rings representing holes or disjoint field sections.
- **Reference line:** A geometric line that is predefined and used as the initial shape for the iterative generation of waylines.



(a) Field boundary



(b) Generated waylines based on the reference line (shown in red).

Figure 4.1: Visual representation of the input and the resultant output. (a) The left panel displays the field boundary. (b) The right panel indicates the reference line (in red) along with the created waylines that are calculated with respect to the reference line.

4.2 Experimental Setup

Computing Environment. All the experiment were carried out in a Python 3.9 virtual environment running on a macOS system with Apple Silicon architecture. The `Pyomo` modeling framework was utilized to create the optimization models.

Software Framework. The computational workflow utilized a blend of open-source and commercial scientific computing libraries. The essential numerical and array operations were handled by `NumPy` and `SciPy`, whereas geometric processing and spatial operations were carried out with the help of `Shapely`. The manipulation of tabular data and the preprocessing were done with the help of `Pandas`, and the entire figures and plots were created using `Matplotlib`.

Optimization Solvers. Two solvers were employed in this research: In this research, we used `IPOPT 3.14.19` solver and the `Gurobi-12.0.3` solver for quadratic programming

formulations. All solver runs were coordinated via the `Pyomo` interface.

4.3 Sensitivity Analysis and Solver Experiments

A systematic sensitivity Analysis was carried out to grasp the impact of key parameters on coverage and path feasibility. These experiments were designed in such a way that only one parameter was varied at a time while the rest were kept constant. The parameters were: (i) the angular offset tolerance μ (limits the deviation of the generated pass lines from the perpendicular direction), (ii) the swath width tolerance γ (indicates the permissible deviation from the swath width), (iii) the objective function weight distribution $w_{Curvature} : w_{Length} : w_{SwathWidth}$ referring to straightness, path length, and skip/overlap minimization costs, respectively and (iv) the number of characteristic points defining the reference spline curve, which influences wayline smoothness, coverage accuracy, and scalability across different field sizes. Besides these fundamental sensitivity parameters, the efficiency of two nonlinear solvers *Ipopt* and *Gurobi* in terms of solution quality and computation time under the same problem settings was also compared. While doing so, the resulting *skip area* (uncovered gaps), *overlap area* (double-covered regions), and *summation of the skips and overlap areas* for each setting were recorded, Throughout all the optimization runs, the curvature constraint (maximum steering curvature $k_{\max} = 1/R_{\min}$; where $R_{\min}$ is the minimum turn radius) was observed, which means that turns with small radii were not allowed even if that position led to better coverage. The goal function was to find ranges of parameters that achieve the two targets of minimizing skips and overlaps and at the same time having the curvatures that are smooth and drivable.

4.3.1 Effect of Skips and Overlaps Cost Weight

A sensitivity analysis was performed on the three cost coefficients to see how the weighting of the objective function affects the quality and the number of the waylines generated. $w_{Straightness}$ (smoothness weight), w_{Length} (path length weight), and $w_{Skips/Overlaps}$

(skips and overlaps weight) represented the three cost coefficients. The coefficients were normalized in such a way that $w_{Straightness} + w_{Legth} + w_{Skips/Overlaps} = 1$. In every trial, $w_{SwathWidth}$ was changed gradually within the range $[0.0, 1.0]$ and $w_{Straightness}$ and w_{Legth} were recalculated proportionally to keep the normalization constant. The optimization problem was solved for each set of weights with the same field boundary, reference line (Field157 - Figure 4.1), and solver settings. The resulting waylines were then analyzed to determine the areas of the skip and overlap from which the corresponding skip and overlap areas were calculated. Geometrical and operational parameters that describe the experiments were constant across the experiments were; working width 7 m, minimum turning radius 3 m, steering curvature 0.12 m^{-1}, reference-line extension 1000 m, allowable deviation from the working width 0.8 m, allowable deviation from the offsetting direction 0.01 rad, number of characteristic points for the generated cubic B-spline curve (70), and number of points used for displaying the curve (1000).

Highlighting the coverage-related weight $w_{Skips/Overlaps}$ is especially significant, as it is the factor that very closely controls the trade-off between path efficiency and coverage completeness. In real-world agricultural practices, complete and even field coverage is a very important goal, as too many skips or overlaps lead to the waste of inputs, the decrease of yield stability, and the increase of time and fuel consumption. When the optimizer progressively raises the importance of $w_{Skips/Overlaps}$, it is first of all a question of reducing the coverage errors, at the potential expense of longer or less smooth paths. Lower $w_{Skips/Overlaps}$ values, on the other hand, concentrate the attention on the reduction of path length and the maintenance of curvature smoothness, thus allowing small unprocessed or overlapped regions.

This analysis quantifies that changes in the value of $w_{Skips/Overlaps}$ affect the balance between the three competing objectives coverage accuracy, smoothness, and path efficiency and identify appropriate weight combinations that produce trajectories that are operationally feasible and efficient in the field.

Table 4.1: Sensitivity of total cost vs. Skips/Overlaps cost weight

Length Cost W	Straightness Cost W	Skips/Overlaps Cost W	Skips	Overlaps	Total
0.500	0.500	0.00	4705.75	147.71	4853.46
0.450	0.450	0.10	2892.58	45.24	2937.82
0.400	0.400	0.20	1712.43	65.75	1778.18
0.350	0.350	0.30	1186.34	93.35	1279.69
0.300	0.300	0.40	894.09	85.95	980.04
0.250	0.250	0.50	706.76	80.66	787.42
0.200	0.200	0.60	561.83	67.97	629.80
0.150	0.150	0.70	459.30	52.84	512.14
0.100	0.100	0.80	360.92	36.64	397.56
0.050	0.050	0.90	295.62	42.57	338.19
0.040	0.040	0.93	274.90	117.63	392.53
0.025	0.025	0.95	271.48	127.41	398.89
0.020	0.020	0.96	271.28	121.88	393.16
0.010	0.010	0.98	271.78	112.27	384.05
0.000	0.000	1.00	277.98	116.79	394.77

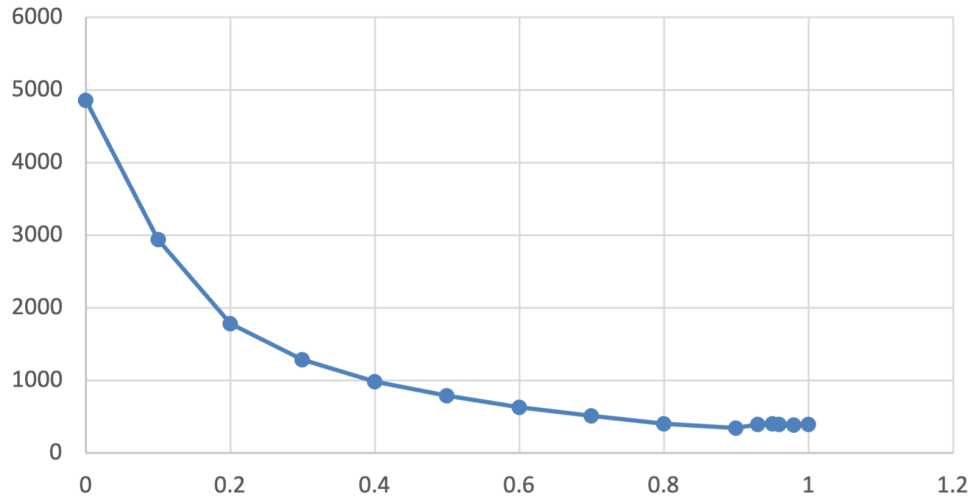


Figure 4.2: Sensitivity analysis results for parameter Skips/Overlaps Cost Weights

As observed in Table 4.1, increasing the weight assigned to the Skips/Overlaps term leads to a substantial reduction in the total cost. When the Skips/Overlaps weight increases from 0.00 to 0.9, the total cost drops from 4853.46 to 338.19, indicating that emphasizing swath width uniformity in the objective function has a pronounced positive effect on cover-

age efficiency. Changing the weight that is assigned to the Skips/Overlaps term to a higher value leads to a substantial reduction in the total cost, as it can be seen from the data in Table 4.1. The reduction in skip areas is the main reason for most of this improvement, and skip areas decrease very quickly as the solver gives higher priority to uniform spacing between adjacent swaths. Still, the total cost goes on stabilizing and fluctuating slightly beyond a Skips/Overlaps weight of about 0.9. Which basically means that Skips/Overlaps weighting is the main factor that plays a significant role in coverage efficiency. The best trade-off between the minimization of skips and the maintenance of path smoothness is at around a Skips/Overlaps cost weight 0.9.

4.3.2 Effect of Swath Width Tolerance

Sensitivity analysis has been performed to see the effect of the swath width tolerance parameter *gamma* on the overall coverage performance. The parameter γ defines the allowable deviation from the nominal swath width during optimization, thereby controlling how strictly adjacent passes are required to maintain uniform spacing. To isolate the effect of this parameter, all other parameters and solver settings were kept constant while γ was varied incrementally within the range $[0.0, 1.0]$. For each value of γ, the optimization problem was solved using the same field boundary and reference line (Field157 - Figure 4.1), and computed the skip and overlap areas. The fixed geometric and operational parameters used in all experiments were as follows: a working width of 7 m, minimum turning radius of 3 m, maximum steering curvature of 0.12 m^{-1}, reference-line extension of 1000 m, allowable deviation from the offsetting direction of 0.01 rad, number of characteristic points for the generated cubic B-spline curve (70), objective function cost weights (swath width, length, and curvature) each set to 1, and number of points used for displaying the curve (1000). The recorded results enabled evaluation of how increasing the allowable swath width tolerance affects the trade-off between redundant coverage and unprocessed regions, providing insight into the balance between geometric adaptability and coverage efficiency.

Table 4.2: Sensitivity analysis results for parameter gamma

γ	Skips	Overlaps	Total Cost
0.00	267.64	416.63	684.27
0.10	277.35	479.43	756.78
0.20	313.15	806.23	1119.38
0.30	303.75	847.76	1151.51
0.40	297.44	850.20	1147.64
0.50	368.16	868.85	1237.01
0.60	299.50	475.85	775.35
0.70	296.34	38.13	334.47
0.80	295.55	48.00	343.55
0.90	296.83	45.50	342.33
1.00	296.60	40.83	337.43

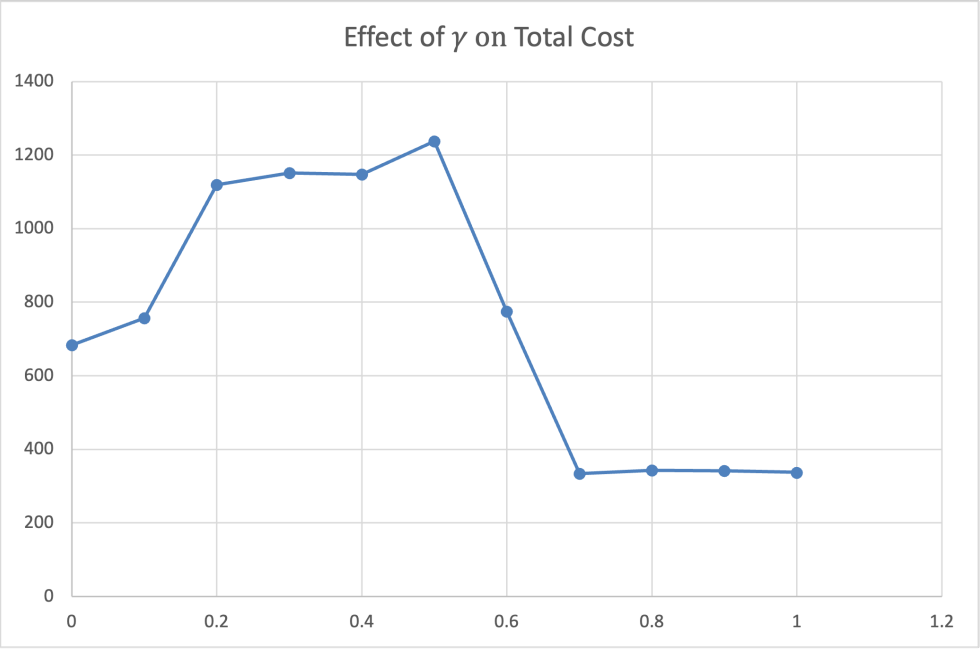


Figure 4.3: Sensitivity analysis results for parameter gamma

Table 4.2 presents the outcomes of the sensitivity analysis on parameter γ. The findings indicate that there is a clear non-linear relationship between γ and the total costs, thus explaining how modifications of this weighting parameter influences the trade-off between skips and overlaps. For instance, at smaller γ values, i.e., $\gamma \leq 0.4$, the total cost is kept quite high, it is more than 1100 units for the majority of configurations. As γ increases beyond 0.5, total performance improvement is very sharp: the total cost goes down significantly to about 350–340 at γ=0.7–1.0. The observed "cost jump" between γ=0.20 and 0.50 is a manifestation of mathematical tension between the hard curvature constraints ($R_{\min}$) and the penalty for swath width deviation. In this intermediate range, the optimizer lacks sufficient "geometric breathing room(tight constraint)" to align the tracks efficiently with the irregular field boundaries, forcing the solver to "cluster" waylines in curved sections to maintain steerability. This results in a massive spike in overlap costs, which account for approximately 70% of the total cost during the peak. At $\gamma \geq 0.7$, the framework undergoes a phase shift where the increased tolerance allows the solver to transition from a restricted local compromise to a globally optimal configuration, successfully harmonizing smoothness with coverage precision and drastically reducing total operational waste. The sensitivity analysis is consistent with the idea that values of γ around 0.70 to 1.00 are the best as far as the trade-off between skips and overlaps is concerned, which leads to a more efficient and robust coverage path plan.

4.3.3 Effect of Angular Tolerance

The angular tolerance parameter μ was part of the sensitivity analysis to figure out its impact on curvature adaptability and coverage performance. The parameter μ is a variable that determines the extent to which the offset direction can be different from the local normal direction of the reference path. In other word, this parameter restricts the extent to which each generated wayline can be different from the ideal perpendicular offset, thus limiting the directional flexibility of the neighboring passes. To assess its effect, all other

optimization parameters were held constant while μ was varied within the range $[0.01, 1.00]$ radians. For each tested value, the optimization problem was solved using the same field boundary and reference lineFigure 4.1), and the resulting waylines were post-processed to measure skip and overlap areas. All experiments were conducted using the following fixed geometric and operational parameters: working width 7 m, minimum turning radius 3 m, maximum steering curvature 0.12 m^{-1}, reference-line extension 1000 m, allowable deviation from the nominal working width 0.8 m, number of characteristic points for the generated cubic B-spline curve 70, objective function cost weights for swath width, length, and curvature each set to 1, and number of points used for visualizing the curve 1000. This analysis made it possible to measure the extent to which angular flexibility affects the balance between geometric conformity, curvature smoothness, and overall field coverage efficiency.

Table 4.3: Sensitivity analysis results for parameter μ

μ	Skips	Overlaps	Total Cost
0.01	1155.83	82.56	1238.39
0.11	1144.05	82.62	1226.67
0.07	1148.08	82.52	1230.60
0.21	1048.70	83.43	1132.13
0.31	1138.73	82.57	1221.30
0.41	1137.97	82.58	1220.55
0.51	1137.22	82.56	1219.78
0.61	1136.36	82.56	1218.92
0.71	1135.45	82.56	1218.01
0.91	1134.32	82.48	1216.80
1.00	1130.81	82.45	1213.26

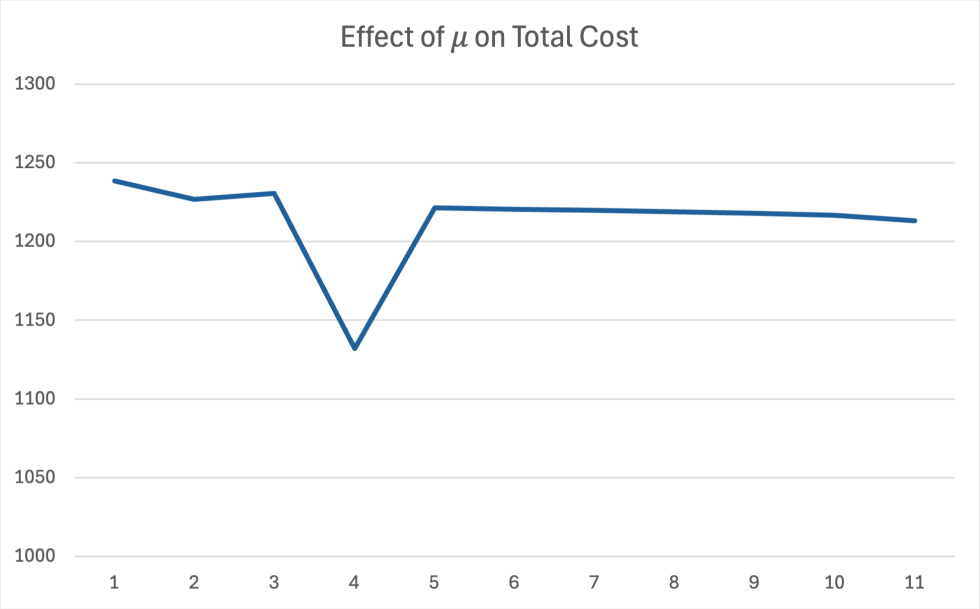


Figure 4.4: Sensitivity analysis results for parameter mu

Table 4.3 presents the findings from the sensitivity analysis for the parameter μ that controls the allowable deviation of the offset direction from the local normal direction of the reference path. The findings show that changes in μ influence the total cost mostly by the effect of the skip term, while the overlap values change very little from test case to test case. Hence, the findings convey that μ still influence path curvature regularization, but its effect on total coverage performance is quite negligible under the given test scenarios.

4.3.4 Solver Experiments

This section is about the comparison of the performances of the two nonlinear solvers, *IPOPT* and *Gurobi*, by means of computation times over multiple field configurations. For each field the model parameters, weights, and optimization settings were kept the same so that the differences in performance could only be attributed to the solvers. Both solvers were called via the *Pyomo* interface that ensured a uniform environment for modeling and execution. The next subsection gives a detailed description of the Pyomo framework and its integration with these solvers.

Pyomo Framework

Pyomo (Python Optimization Modeling Objects) is an open-source algebraic modeling language (AML) that is integrated into the Python programming environment. It delivers a high-level, object-oriented structure for defining, examining, and solving mathematical optimization problems. Pyomo enhances the capabilities of Python by adding symbolic objects for optimization modeling, i.e., decision variables, objective functions, and constraints, thus giving users the ability to represent complex mathematical models in an easily understandable and highly adaptable syntax. Being a Python-based AML, Pyomo takes the best features from established modeling languages such as AMPL and GAMS and combines them with the strength of a general-purpose programming language. The integration allows for advanced modeling workflows, data-driven optimization, and hassle-free communication with other scientific computing libraries like NumPy, SciPy, and Matplotlib. Pyomo accommodates a wide range of optimization paradigms such as linear, nonlinear, mixed-integer, quadratic, stochastic, and dynamic models constrained by differential or algebraic equations [25]. One of the distinguishing characteristics of Pyomo is that it can handle both *concrete* and *abstract* model formulations. Concrete models have all the data defined right in the model file, which is suitable for quick prototyping and testing. Abstract models, on the other hand, separate the model structure from the data, thus enhancing modularity and reusability of the different instances of the problem. Besides that, Pyomo's hierarchical and object-oriented architecture also allows the use of *blocks* and nested components for the structuring and extending of models. Pyomo interfaces with a wide range of commercial and open-source solvers, including CPLEX, Gurobi, IPOPT, and GLPK, through tightly and loosely coupled interfaces. This flexibility allows efficient solution of problems ranging from small-scale academic examples to large-scale industrial systems. The framework also provides built-in tools for model transformation, parallel computation, and integration with cloud-based solver services such as NEOS.

In this research, Pyomo was used as the primary modeling framework to implement and

solve the formulated optimization problems. The decision variables, objective function, and constraints were programmatically defined within Pyomo. The model was solved using two solvers, IPOPT and Gurobi, to compare solver performance.

4.3.5 Ipopt vs. Gurobi

To evaluate solver efficiency and numerical behavior, both *IPOPT* and *Gurobi* were tested across five representative field configurations. For each field, the optimization problem was solved twice while maintaining identical model formulations and parameter settings; once using *IPOPT* and once using *Gurobi* while keeping all geometric, operational, and model parameters fixed. The experiments were done in such a way that the solver was the only variable. Hence, the observed differences can only be attributed to the different solver algorithms. Next we show the comparative results and analysis of the solver's performance. The findings were assessed in terms of skips, overlaps, total cost (sum of skips and overlaps), and solver time. The experimental design thus ensured that any performance differences were due to the inherent features of the solvers and not to different problem configurations.

Table 4.4: Comparison of IPOPT and Gurobi Solver for different fields

Field	IPOPT				Gurobi			
	Skips	Overlaps	Total Cost	Solver Time (s)	Skips	Overlaps	Total Cost	Solver Time (s)
Field 110	9538.27	7006.64	16544.91	37.64	9874.49	6856.94	16731.43	6.02
Field 157	1855.79	72.05	1927.84	4.96	1855.24	71.93	1927.17	1.10
Field 279	73838.14	135193.47	209031.14	38.70	74797.91	119999.29	194797.2	12.91
Field 311	607.54	0.00	607.54	7.98	603.25	0.00	603.25	1.19
Field 442	131972.64	15873.98	147846.62	38.32	136177.70	15724.34	151902.04	11.31

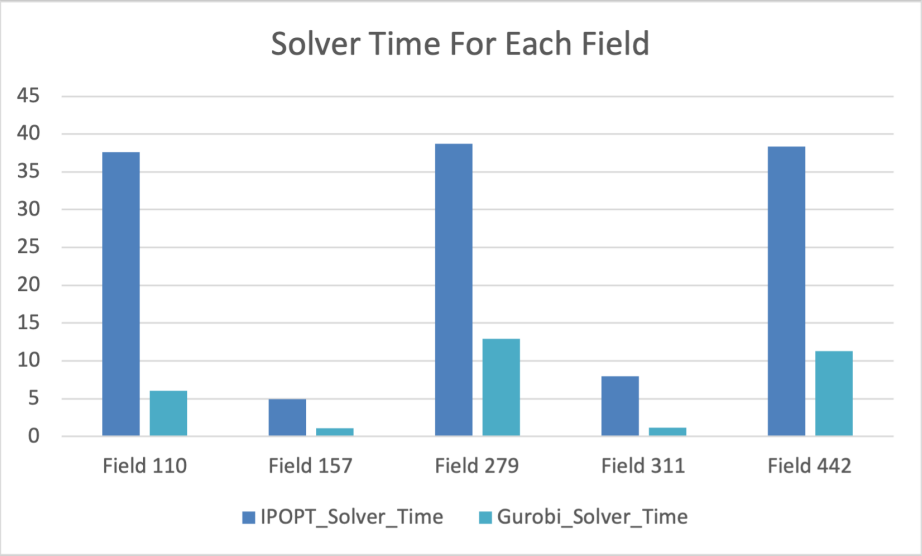


Figure 4.5: Solver time by field (IPOPT vs. Gurobi). Each bar represents the computation time required for convergence under identical optimization settings.

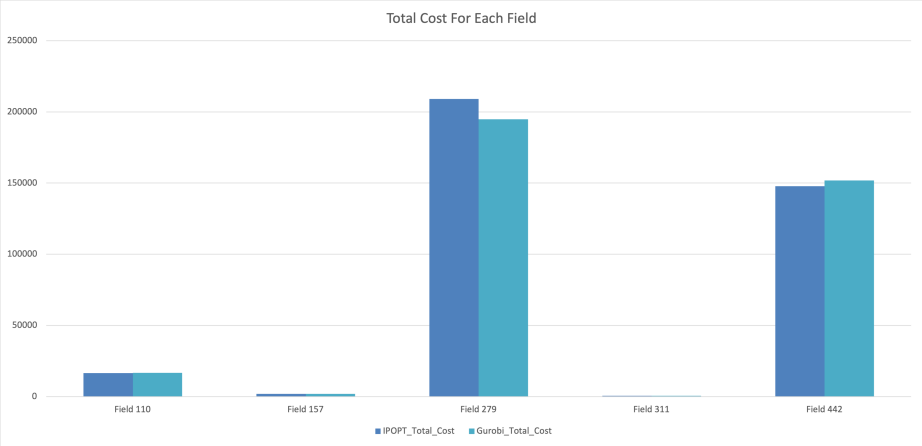


Figure 4.6: Total cost (skips + overlaps) by field (IPOPT vs. Gurobi). Results show solver performance consistency across all test fields.

The results shown in Table 4.4 illustrate how IPOPT and Gurobi solvers performed under the same optimization conditions. In other words, all parameters and weights referring to the swath width, curvature, straightness, and path length were the same for both cases. Therefore, the variations that have appeared can only be attributed to the differences in the solver algorithms and numerical strategies. In all the test fields, both solvers managed to converge to solutions that are feasible and which satisfy the same optimization constraints.

Yet, the two performance profiles differ substantially with respect to the aspects of computational efficiency and numerical robustness. The Gurobi solver was constantly able to converge faster with the time of the solver typically being one order of magnitude less than that of IPOPT. For example, the time taken by Gurobi and IPOPT for completion of the optimization on larger or more irregular fields is about 10–15 and 30–40 seconds respectively. In fact, this difference highlights Gurobi's power as a well-tuned commercial mixed-integer and quadratic programming solver. Regarding the quality of the solution, both solvers produced almost similar skip and overlap values in all the test fields proving that Gurobi's speed is not reason enough for accuracy or coverage performance to be compromised. However, the total skip and overlap costs are relatively high because there was no field-specific parameter tuning done. The main point of this experiment was to ensure a fair comparison between two solvers under identical model configurations rather than to reach optimal coverage efficiency for each field.

In summary, the comparison of the solvers, IPOPT and Gurobi solvers can produce accurate and consistent results when the same optimization settings are used, but their computational characteristics are quite different. On the one hand, IPOPT is a nonlinear solver that is capable of robust convergence and hence, is still good for research-oriented analyses or model extensions that involve complex constraints. On the other hand, Gurobi is much faster and can handle a large number of variables which is why it is a better choice for real-time field applications or those of high scale. The results of the study reveal that the optimization framework can be implement using either solver effectively and the deciding factor would be mainly the operational context and the performance requirements.

4.3.6 Number of Characteristic Points of the Spline Curve

An additional sensitivity analysis of the number of characteristic points (Characteristic Points define the resolution of the spline) used to define the spline curve was performed to measure the effect of reference path resolution on the optimization result. This analysis

was performed by two different experiments; the first experiment the same field boundary and optimization settings were kept while the number of spline control points to represent the reference line at different levels of discretizations was changed. Thus, it was possible to evaluate the dependence of path smoothness and coverage accuracy on the changes in input resolution. In the other experiment, the same number of characteristic points was used for geometrically similar fields of different scales to study the scalability and constancy of the formulation. For every configuration, the optimization was carried out with the same solver settings, and the skip and overlap metrics were captured to measure the impact of spline point density on the quality of the solution. The tables below present the summary of the results of both scenarios in terms of skips, overlaps, and total cost. To examine the optimization performance sensitivity to the spline discretization, the number of control points defining the reference spline was systematically varied. This experiment addresses the question of how the total cost, calculated as the sum of skips and overlaps, changes with the increasing density of spline control points when all other optimization parameters are kept constant. The goal is to understand the influence of control points on coverage.

Table 4.5: Scenario 1: Effect of spline control point count on skips, overlaps, and total cost

Number of Control Points	Skips	Overlaps	Total Cost
10	3347.96	82.35	3430.31
20	1224.28	129.00	1353.28
30	687.75	111.17	798.92
40	501.74	72.78	574.52
50	411.80	49.80	461.6
60	352.94	37.55	390.49
80	296.60	40.83	337.43
90	284.59	55.09	339.68
100	271.71	45.80	317.51

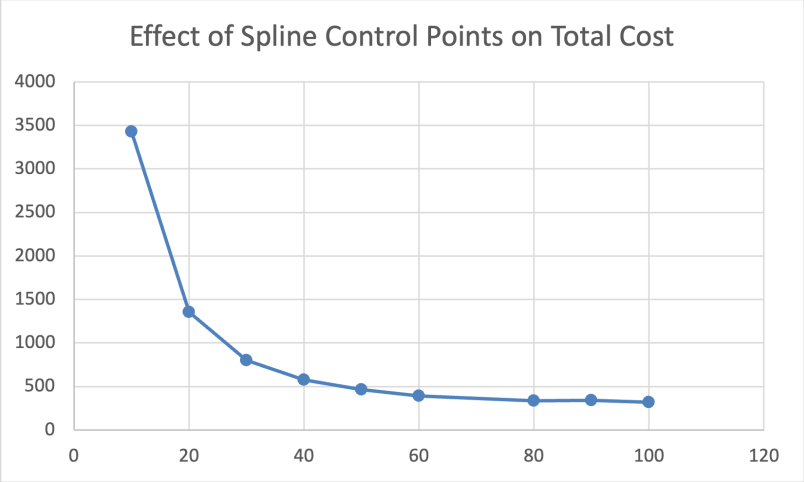


Figure 4.7: Effect of spline control point count on total coverage cost.

The quality of the coverage paths that have been generated is directly influenced by the number of control points that are used to define the spline reference line. As illustrated in Table 4.7, a curve that is represented by only a few control points thus lacks the flexibility to wrap tightly around the field geometry, which in turn leads to large deviations from the

adjacent paths. These deviations correspond to an increase in the number of skips and overlaps, which in turn results in a higher total cost. When the number of control points is increased, the spline has more strength, which means that the optimization can better adjust the curvature of the path to fit the local field variations. Therefore, both the skips and the overlaps are reduced considerably, thus indicating that the coverage is more efficient and the transitions between the tracks are smoother. Nonetheless, for the reference Field 157 (Figure 4.1), it is an indication that the improvement in the total cost becomes very small beyond approximately 60-80 control points, which means that the spline representation is already dense enough to capture the main features of the field. Essentially, raising the number of control points enables the spline to better represent the field geometry, thereby ensuring smoother and more accurate path derivation. Nevertheless, this enhancement is only up to a certain level of detail, after which the spline is sufficiently detailed to represent the basic shape of the field.

In the next experiment, to measure the scalability and consistency of the designed optimization framework, a simple Field 157 (Figure 4.1) with a curved boundary was geometrically transformed into four different size variations, both bigger and smaller, by uniformly scaling. Each scaled field retained the same reference line and optimization parameters to ensure a fair comparison. The performance metrics; skips, overlaps, and total cost, were measured for each scale. Additionally, the *Total Cost (%)* was calculated as the ratio of total cost to field area, expressed as a percentage using the relation:

$$\text{Total Cost } (\%) = \frac{\text{Total Cost} \times 100}{\text{Field Area}}.$$

Table 4.6: Scenario 2: Performance consistency across similar shaped fields in different scales using the same number of control points

Field Scale	Field Area	Skips	Overlaps	Total Cost	Total Cost (%)
0.1x	118.30	27.27	0.29	27.56	23.30
0.5x	2957.52	320.91	14.82	335.73	11.35
1.0x	11830.08	1156.99	80.02	1237.01	10.46
1.5x	26617.68	2835.33	195.95	3031.28	11.39
2.0x	47320.32	5426.29	363.13	5789.42	12.23
2.5x	73938.00	9119.16	582.23	9701.39	13.12

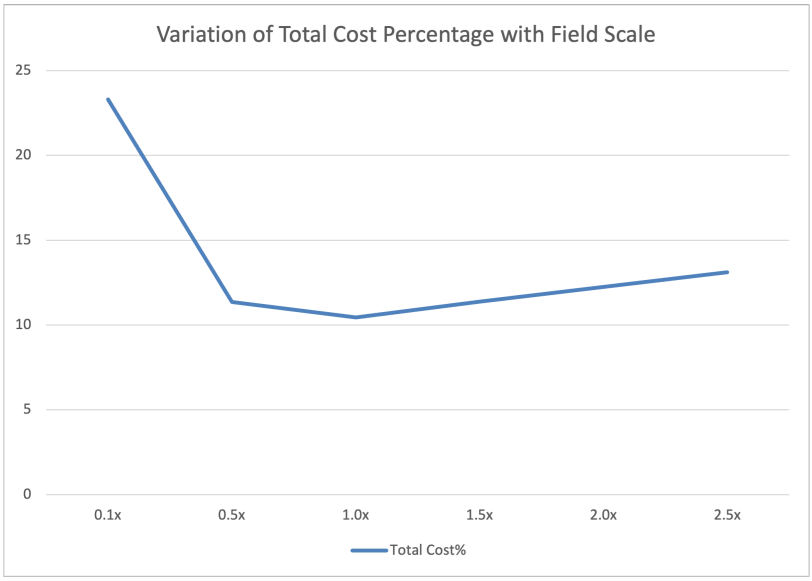


Figure 4.8: Effect of field scale on coverage efficiency, showing how the total cost percentage varies across geometrically scaled versions of the same field.

As illustrated in Figure 4.8 and summarized in Table 4.6, the total cost percentage decreases sharply when the field scale increases from 0.1× to 0.5× and stabilizes around 10–13% for larger scales. This indicates that the optimization framework maintains a consistent coverage efficiency once the field reaches a practical operational size. The high cost percentage at the smallest scale (0.1×) can be attributed to the limited spatial resolution. At

such points the optimization behavior is mainly influenced by the curvature of the reference line and path spacing, thus resulting in proportionally larger skips and overlaps. With the increase in the field area, these edge and curvature effects fade away, and the framework attains smoother and more balanced coverage patterns. After the 1.0× scale, the total cost percentage shows a slight upward trend, which is indicative of minor efficiency losses that are associated with scaling larger fields without re-tuning the optimization parameters. This behavior is an indication of the reliability of the quadratic optimization-based track generation method whereby coverage quality is almost constant within a wide range of field sizes and at the same time it is also suggesting that adaptive parameter scaling can be far more effective in very large or complex fields. Overall, the findings show that keeping the same number of control points across similar shaped fields of different scales maintains stable performance once the field reaches a practical operational scale.

By performing the sensitivity analysis and looking at the impact of objective weights, geometric tolerances, solver configurations, and spline resolution on the optimization result, the physical behavior of the formulation was essentially understood. These experiments revealed considerable information on how parameter tuning affects coverage performance, curvature continuity, and computational efficiency. With this knowledge, the subsequent phase of this work is devoted to investigating and comparing various smoothing methods. The goal of this examination is to refine the baseline optimization method by continuity of the trajectory and geometric adaptability while keeping the quality of the field coverage quality. Although these parameters were examined separately to understand their isolated effect on coverage performance and curvature smoothness, they are interdependent and strongly influenced by the geometry and irregularity of the field in reality. Hence, parameter tuning should not be considered as a fixed setting but rather as an adaptive process that adjusts to the characteristics of a specific field. To support this, future development of the framework could improved end-user interface that allows operators to interactively adjust the key parameters, thus enabling the generation of more realistic and field-adaptive way-

lines. For consistency, the sensitivity analysis was performed using Field 157 only which is a representative case from the real world where growers have reported very high skip and overlap rates. The next section gives an overview and analysis of the smoothing methods that have been taken into consideration in this research.

4.3.7 B-Spline Smoothing

The B-spline smoothing method was used as the baseline approach to generate continuous and feasible tracks across all test fields. The outcomes below demonstrate the produced tracks, the change of curvature along each route, as well as the skip and overlap metrics.

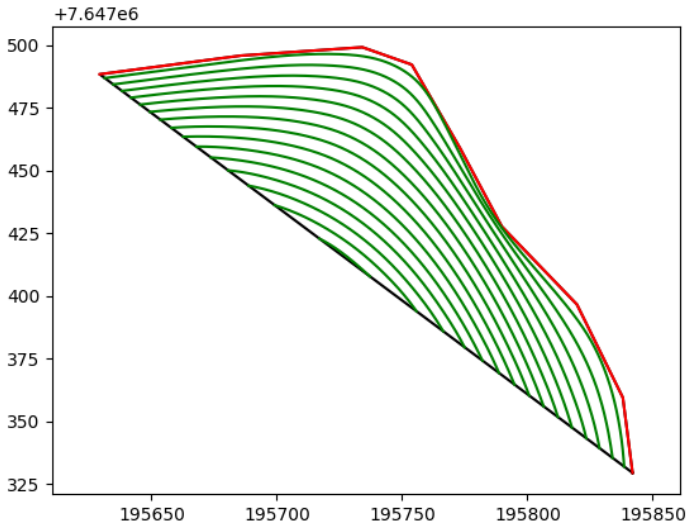


Figure 4.9: Generated tracks for Field 157 using cubic B-spline smoothing.

Figure 4.9 illustrates how the B-spline model produces a smooth and continuous series of parallel tracks that fit the agricultural field boundary well. Overall alignment remains consistent with the optimized reference path.

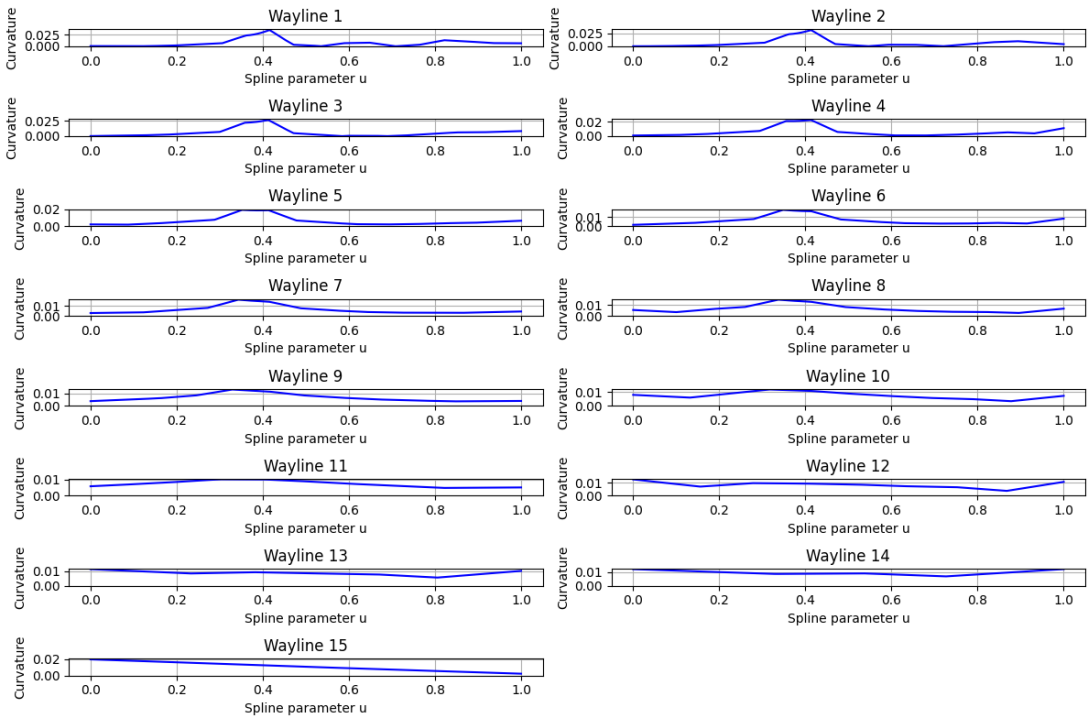


Figure 4.10: B -spline curvature κ versus normalized spline parameter $u \in [0,1]$ for each track. The horizontal axis is spline parameter, while the vertical axis is curvature (1/m) computed from first and second derivatives of the fitted B -spline; higher κ indicates tighter turning demand.

Figure 4.9 demonstrates that the B-spline model provides a smooth and continuous set of parallel tracks that fit very well to the field boundary. Overall alignment remains consistent with the optimized reference path. The curvature profiles in Fig. 4.10 show that all the generated waylines are continuous and of low curvature and there are no sudden oscillations or discontinuities. Small curvature peaks can be found at similar positions of the spline parameter for all tracks and these points correspond to a gentle geometric bending of the respective field region. Thus, it is confirmed that the optimization produces globally smooth trajectories, local curvature changes being gradual and occurring in the same spatial area.

The total duration of this experiment was 2.86 seconds. The average computation time for one wayline was 0.143 seconds, which is a clear indication of the great performance of the quadratic programming formulation and the spline-based implementation. These are

the proof of the B-spline formulation being a reliable tool for the creation of smooth and continuous coverage paths. However, its limited local control and sensitivity to boundary curvature motivated the exploration of alternative smoothing methods.

4.3.8 Bézier Smoothing

After the cubic B-spline method had been assessed, Bézier smoothing was put in place as a different method for producing continuous-curvature waylines. The overall framework, including the structure of the optimization model and the wayline generation process, was kept identical to that used in the B-spline experiments. The smoothing formulation was changed only to use Bézier curve representation to provide a fair basis for the performance comparison of the two methods. The experimental settings, such as the boundary geometry, reference line, work width, and solver configurations, were kept the same so that the impact of the smoothing method could be directly evaluated.

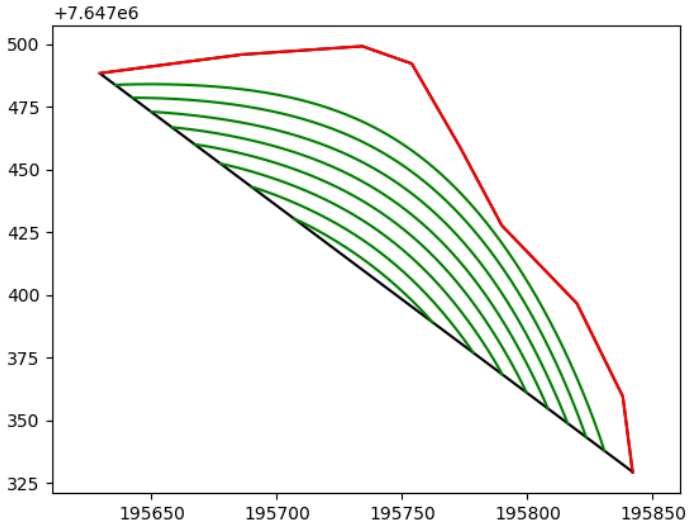


Figure 4.11: Generated tracks for Field 157 using cubic Bezier smoothing.

Figure 4.11 illustrate the waylines generated using Bézier smoothing technique. While B-splines offer local control of the curve shape due to their segmented formulation, Bézier curves have global control, i.e., any change in a control point affects the entire curve. There-

fore, the Bézier based waylines from the same reference line under identical experimental conditions show a less precise interpretation of that reference compared to the B-spline approach. Because of this global sensitivity, coverage accuracy is often lower, which is the reason for the slightly higher skip and overlap values that are present, although the resulting trajectories are still smooth and continuous.

The total runtime for the Bézier-based experiment was 2.71 s, with an average computation time of 0.136 s per wayline, demonstrates the efficiency of the quadratic programming formulation and the curve-based smoothing implementation.

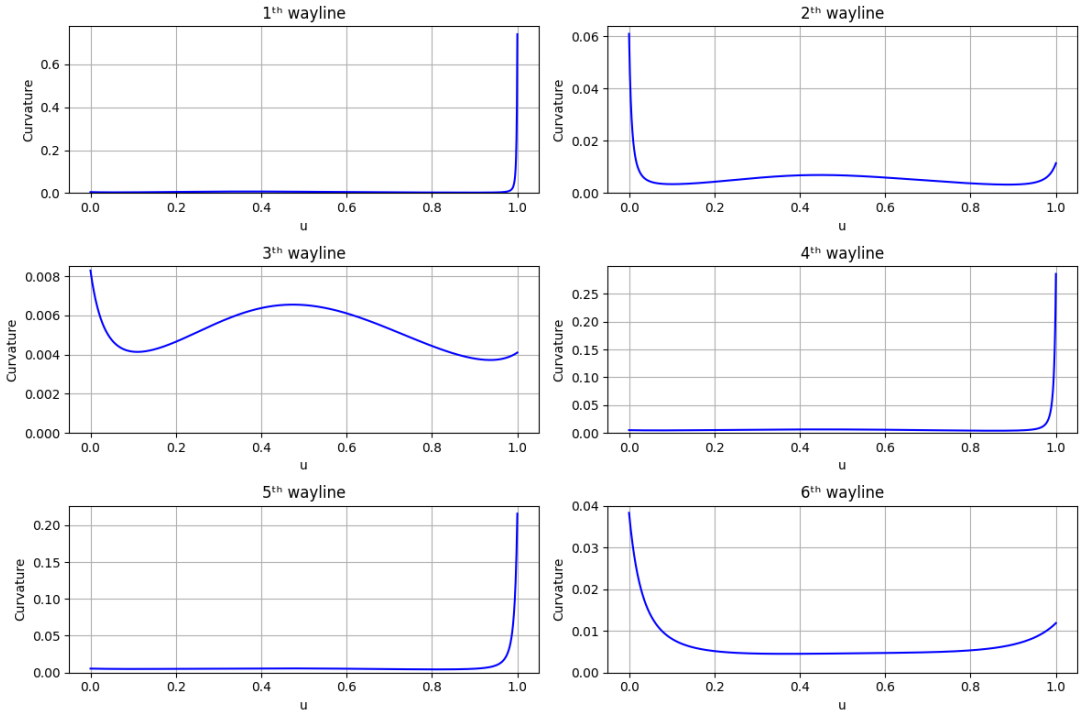


Figure 4.12: Curvature variation κ versus spline parameter across generated Bézier tracks. Higher curvature values correspond to tighter turns.

Figure 4.12 demonstrates how curvature changes along the normalized spline parameter u for the waylines generated utilizing Bézier smoothing method. The figures show that the Bézier formulation is able to keep the curvature continuous and smooth for all the waylines, satisfying the requirement of the curvature continuity of the optimization model. However, the magnitudes of the curvature exhibit more significant spikes around the end

points ($u = 0$ and $u = 1$). The reason for this is, near the endpoints, the shape of the curve is mainly determined by the first and last control points as well as their corresponding tangent directions. Therefore, if there is any slight change in these points, it will result in an increase in the curvature at the boundaries. Although these fluctuations are still within the permissible operational limits, they account for the relatively higher skip and overlap values, which is an indication that there has been a slight decrease in coverage precision along with the maintenance of overall trajectory smoothness.

4.3.9 NURBS Smoothing

To further improve local adaptability and geometric flexibility, the Non-Uniform Rational B-spline (NURBS) smoothing method was implemented as an extension of the baseline B-spline formulation. Unlike conventional B-splines, NURBS introduce a weighting parameter for each control point, allowing localized adjustment of curve influence and enabling the exact representation of conic sections. This additional degree of control enhances the model's ability to conform to irregular or highly curved field boundaries while maintaining curvature continuity. The subsequent analysis presents the runtime characteristics, curvature variation, and coverage performance obtained using the NURBS-based wayline generation framework.

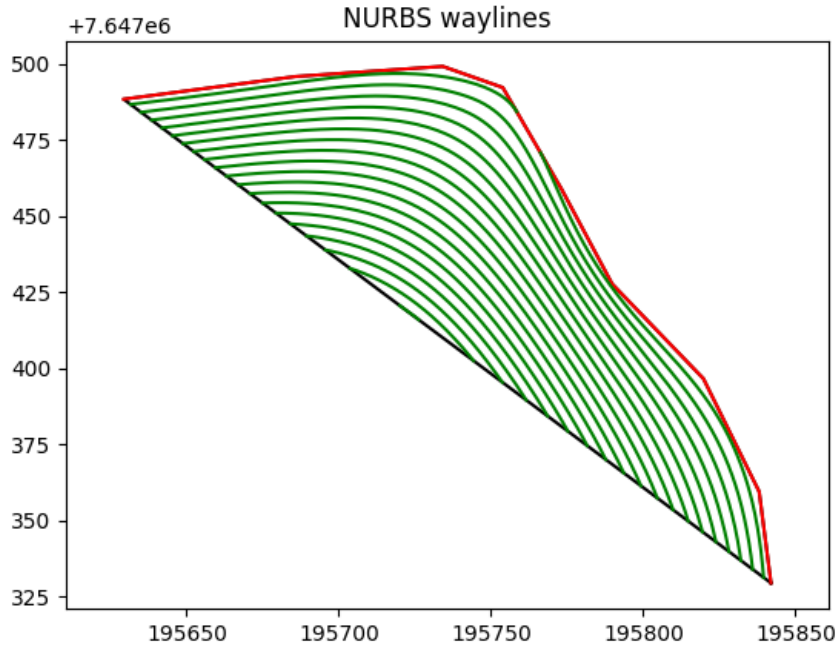
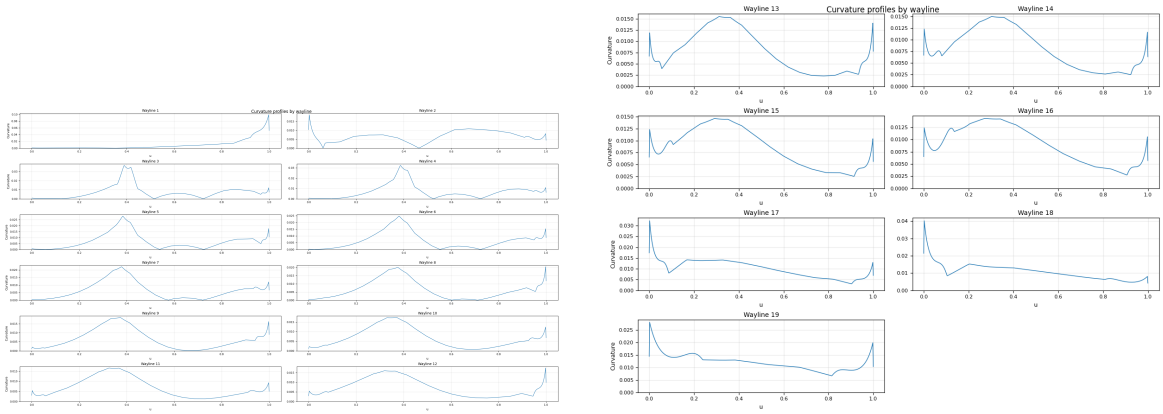


Figure 4.13: Generated tracks for Field 157 using NURBS smoothing.

Figure 4.13 shows the waylines generated using the NURBS smoothing approach. NURBS-based paths exhibit a more balanced curvature distribution and well fitted to the field boundary compared to the Bézier and B-spline results. The weighting feature in the NURBS expression gives more precise control in the locality, thus the curve can smoothly follow the boundary irregularities without causing an area of high curvature or distortion. In fact, the resulting waylines not only keep the spatial pattern of the field consistent but also have a higher geometric adaptability in the corners where both Bézier and B-spline curves are likely to be far from the boundary. Hence, the NURBS smoothing technique essentially performs like a hybrid of local control properties of B-splines and global flexibility of Bézier features, thereby yielding a set of trajectories that are smooth, well-aligned and boundary-aware resulting in better field coverage quality.



(a) Curvature profiles for waylines 1–12 (b) Curvature profiles for waylines 13–19

Figure 4.14: Curvature variation κ across generated NURBS waylines, plotted with respect to the spline parameter $u \in [0, 1]$.

Figure 4.14 plots the curvature profiles of the waylines generated using the NURBS smoothing method. Overall, the curvature is smooth and continuous over the entire parameter range, which is an evidence that the NURBS formulation is an effective way of preserving geometric continuity while providing higher local adaptability. Compared to the Bézier results, the curvature peaks are less pronounced and occur more gradually, which is an indication that there is better control of local curvature changes. The weighting mechanism that is inherent to NURBS enables a certain control point to have more or less influence which helps to regulate the curvature variations without compromising the path smoothness. It is true that slight increases of the curvature can still be detected close to the endpoints (u = 0 and u = 1), but their magnitudes are much smaller than those in the Bézier case, thus it can be concluded that the introduction of rational weights successfully successfully mitigates endpoint sensitivity. The middle parameter parts ($0.3 \leq u \leq 0.7$) show almost constant curvature values, suggesting consistent turning smoothness throughout the field. Collectively, these findings provide evidence that NURBS smoothing achieves a favorable balance between global smoothness and local geometric control, resulting continuous-curvature trajectories that are computationally stable and feasible. The total runtime of the optimization process was 4.26 seconds, with the average solver time per wayline being 0.213 seconds.

Because of the local weighting property of the NURBS formulation, it is anticipated that the skip and overlap magnitudes will be lower for NURBS as compared to the other smoothing methods. Having presented the individual results for the B-spline, Bézier, and NURBS smoothing methods and compares their performance based on curvature characteristics, and the accuracy of the reference line representation by each method, the main goal of this comparison is to see how the mathematical model of each curve type, control structure, and continuity properties, influence both the quality of coverage and geometric smoothness. The comparison of these features makes it possible to identify the relative advantages and limitations of each smoothing approach, offering insights into their suitability for field coverage optimization.

Table 4.7: Skip/Overlap comparison across smoothing methods (areas in m^2).

Fields	**B-spline**		**Bézier**		**NURBS**	
	Skips	**Overlaps**	**Skips**	**Overlaps**	**Skips**	**Overlaps**
157	3616.74	92.34	7530.65	0.00	1848.96	82.28

Units: m^2

For Field 157, the total field area was measured as 11,830.08 m^2. Based on the same boundary, the coverage results were notably different for the three smoothing methods. The B-spline representation covered 8213.34 m^2 of the field, which is equivalent to a coverage ratio of 69.43%. The Bézier method, on the other hand, had a lower covered area of only 4299.43 m^2, and thus the coverage ratio amounted to 36.34%. In contrast, the NURBS method was able to cover the largest area, with 9981.12 m^2 covered and the respective ratio of 84.37%.

These results clearly proves, adding rational weights in the NURBS formulation significantly boosts its capability to follow the field boundary while also keeping the curvature smooth and continuous. The local weighting feature of NURBS makes the method more flexible in areas with strong boundary curvature or irregular edges thus it is less of

a skipped and overlapped areas. For instance, the B-spline keeps the equilibrium between coverage and stability, whereas the Bézier method which is limited by its global control shows a weaker conformity to the reference line and consequently the lowest coverage ratio. Overall, the NURBS smoothing method has the greatest geometric adaptability and the most effective way of the available field area, although its average computational time was slightly higher compared to the other two smoothing methods. Beyond the inherent characteristics of each smoothing composition, the quality of the reference line has been a very important factor in the determination of the overall field coverage. Since each next wayline is created by moving the offsets from the reference line, any deviation or misinterpretation in the first wayline will be very directly affecting the alignment of the entire set of tracks. A well-interpreted reference line makes the first wayline not only to be very close to the field orientation but also to be quite consistent in the offsets and to have very few gaps or overlaps near the boundaries.

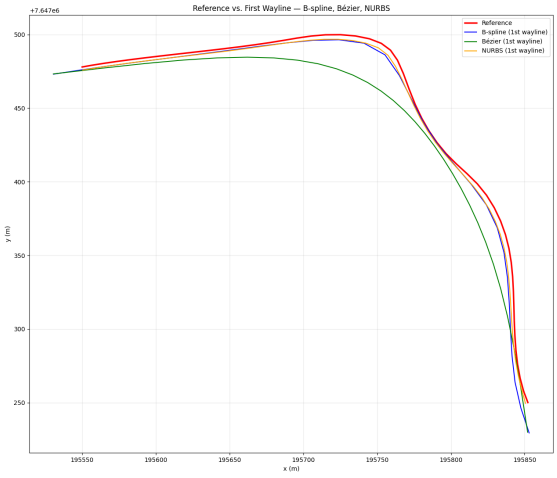


Figure 4.15: Comparison between the reference line and the first wayline generated by each smoothing method (B-spline, Bézier, and NURBS). The alignment quality of the first wayline reflects each method's ability to interpret and preserve the reference geometry.

Different smoothing methods resulted in visibly different first wayline shapes relative to the reference line as is evident from the first waylines comparison in Figure 4.15. The NURBS-based method is the one that most closely preserves the reference-line geometry,

as it stays aligned almost everywhere along the path. In contrast, the Bézier formula deviates from the reference to a greater extent, especially near the regions of high curvature, which is a direct result of global control property of the curve. Such deviation of the first track is carried over to the subsequent offsets, causing positional inconsistencies and larger unserviced regions. Moreover, in the case that the shape of the initial wayline does not properly represent the reference line, the effective coverage width of subsequent tracks will be reduced even though the working width stays the same. As a result, fewer valid waylines are generated within the same field boundary, further decreasing total coverage efficiency. The B-spline approach is the one that falls between these two, providing both features, conformity and smoothness, in a balanced way. Therefore, the closeness of the first wayline to the reference line can be considered as a main factor that indirectly measures the level of the coverage quality and the efficiency of the path planning. Considering these points, the role of the reference line in fact determining the quality and adaptability of the produced coverage paths becomes very clear. The geometric accuracy of the first wayline with respect to the reference line is what mainly determines the initial conformity; however, the reference line's shape and placement are the ones that ultimately decide how the subsequent tracks will be laid out in the field. Different reference line configurations whether being the result of the field boundary, centerline, or optimized alignment can produce very different curvature distributions, turning behaviors, and coverage efficiency. Therefore, in the following section, we analyze the reference line flexibility results, highlighting how variations in its geometry.

4.4 Reference Line Flexibility: Bidirectional Propagation from a Mid-field Seed

Building on the preceding discussion, the accuracy of the first wayline with respect to the reference line was one of the key factors that affected the overall coverage performance. The next stage of the research is focused on improving the adaptability and the

user-friendliness of the process of generating the reference line. Although the B-spline, Bézier, and NURBS methods were able to produce smooth and efficient paths, their performance still depended on the placement of the initial reference line. In real field operations, it might be more convenient to start coverage from a central or operator-defined location rather than going strictly along a boundary edge for some instances. To allow such flexibility, the system was upgraded to support reference lines that are started from the middle of the field, thus providing a more versatile and realistic evaluation of wayline generation and coverage effectiveness. Typically, boundary geometries of real agricultural fields are often irregular due to natural terrain constraints, drainage channels, or land ownership divisions. Thus, it might not always be possible or efficient to generate waylines solely from the edge of a field to obtain a good coverage pattern. In case the first reference line is limited to a very curved boundary, the subsequent propagation might not be able to reduce the area of skips and overlaps effectively, especially where the field orientation changes significantly. Allowing users to choose a different starting line (such as one close to the center of the field) makes better alignment between adjacent passes, reduces accumulated geometric error, and improves overall coverage efficiency.

To overcome this limitation, the wayline creation method was restructured to allow for variable starting point along any reference line within the area of the field instead of it being limited to just an edge of the field. In this implementation, a central reference line is selected, and waylines are propagated outward in both directions. This flexibility is advantageous for complex and irregular fields, because the process of subdividing the field into smaller and locally consistent subregions can improve both computational stability and operational efficiency. We do the same optimization pipeline twice, under the same conditions with the only difference being that the propagation side parameter is changed. We keep all solver configurations, objective weights, spacing targets, and curvature bounds the same. The left- and right-propagated families thus formed are patched to the boundary and displayed both separately and together.

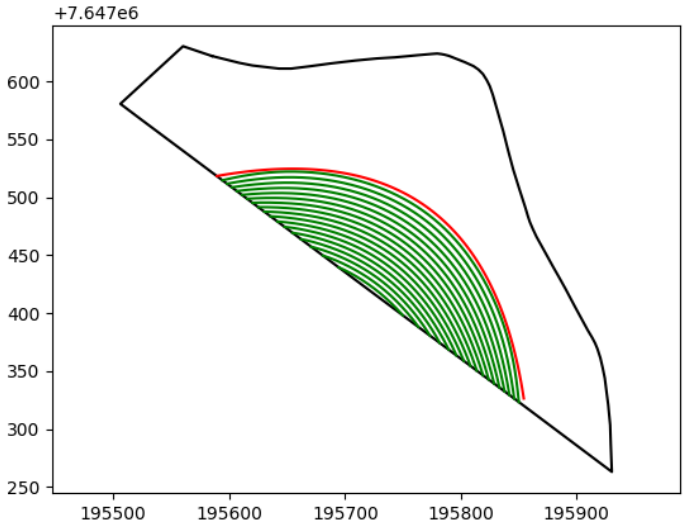


Figure 4.16: Tracks generated from the mid-field reference in the left direction.

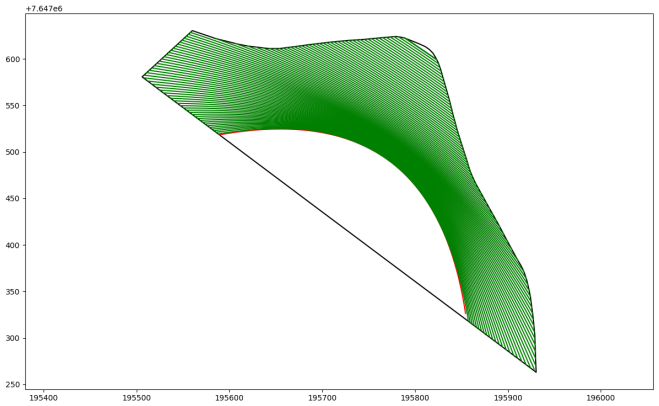


Figure 4.17: Tracks generated from the mid-field reference in the right direction.

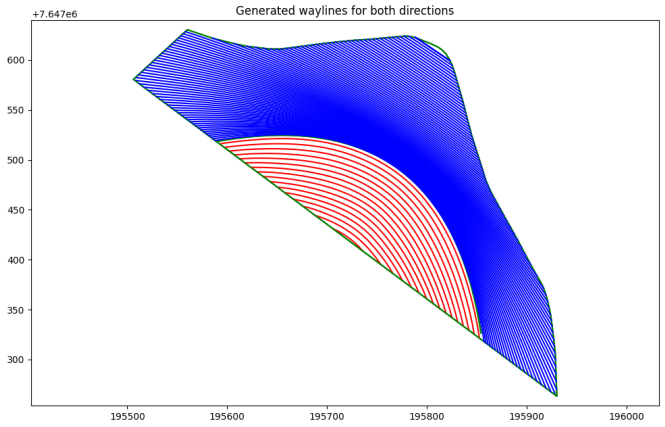


Figure 4.18: Combined overlay of bidirectional propagation results showing complete field coverage.

By using this extension, the proposed system becomes more efficient and flexible. With the option to choose a reference line from the field inside, users are able to break down complicated shapes into smaller parts, thus reducing curvature distortion and increasing the total coverage efficiency of the optimization framework. The subsequent improvement of the reference-line generation system, was motivated by the necessity to obtain a more accurate and complete representation of the field.

Chapter 5

Conclusion and Future Work

5.1 Summary and Conclusion

This research has successfully developed quadratic optimization based wayline generation method to the industrial partner as a realistic solution based on for their main operational problems. Their problems included minimizing skips and overlaps while making sure that the generated tracks remain smooth and can be easily guided. The introduced approach integrates curvature control, coverage uniformity, and spatial alignment into a single optimization framework, thus presenting a theoretically sound and practically reliable base for agricultural path planning. This research goes beyond the company's current geometric method to also introduce new smoothing expressions for this quadratic optimization framework, e.g., Bézier, B-spline, and Non-Uniform Rational B-spline (NURBS). A comprehensive comparative assessment has indicated that the use of rational weighting in the NURBS equation enhances the capacity of the path to follow the reference line while also making sure that the path is continuous and that the curvature is allowable. The NURBS-based method is able to achieve the minimum areas of skips and overlaps as well as the most accurate representation of the reference line among the tested methods, hence it is the most effective smoothing technique for complex real-world applications. Moreover, the detailed sensitivity analysis in this work has served to demonstrate the framework's stability and versatility. By altering parameters such as swath width tolerance, angular deviation, number of points defining the splines, and objective-function weights one at a time, the analysis identified the parameter settings that deliver excellent coverage performance. For instance,

giving a greater weight to the swath width term instead of employing equal weighting for curvature, length, and swath width cost weights led to curves that were not only smooth and easily steerable but also had the skip and overlap regions significantly reduced. These findings offer practical guidance for parameter tuning in operational settings and demonstrate the flexibility of the model under different field conditions. Furthermore, the method was tested using two solvers; IPOPT and Gurobi to compare the solution quality and the Computational time and identified among these two solvers the solution quality remains almost same for both solvers while the time taken is significantly less in Gurobi solver. For clarity and consistency, Field 157 (Figure 4.1)was chosen as the representative example for this entire work, since it was one of the fields where the growers had reported higher levels of skips and overlaps. By means of the suggested quadratic optimization framework, we managed to generate smooth and curvature-continuous waylines for diverse irregular real-world agricultural fields that were supplied by the industry partner. The illustrations of several complicated field layouts are given in Appendix A. The method demonstrated capability in dealing with diverse geometries while also ensuring high coverage efficiency and steering feasibility. Essentially, the framework is a great solution that is scalable, computationally efficient, and is able to meet the requirements of modern precision agriculture, thus it can reliably generate paths that minimize the operational inefficiencies of a wide range of field environments.

5.2 Future Research Directions

While the present framework effectively addresses two dimensional optimization and is successful in producing smooth and steerable waylines for agricultural fields, it leaves several aspects unexplored. Basically, extending the quadratic optimization framework to three-dimensional, terrain-aware wayline generation is a significant ongoing work direction. The purpose of this extension is to embed the elevation data deep into the optimization process so that the resulting tracks not only look but are the actual surface geometries of the

field. Correspondingly, the incorporation of soil erosion modeling is at the same stage of progress, which will motivate the framework to become environmentally friendly by emitting and reducing the erosion risks associated with different path orientations. Collectively, the progress of these features will let the system be able to yield the most realistic, terrain-sensitive, and environmentally friendly coverage plans. Moreover, beyond the extensions, obstacle handling can be integrated to facilitate the automatic recognition and evasion of those in the field, like trees, ponds or poles, thus the framework's resistance and practicality in complicated field environments will be enhanced. Also, the reference line generation can be improved by segmentation and adaptive propagation methods enabling the division of large or irregular fields into smaller subregions which can be independently optimized. This would make possible multi-directional and context-aware wayline generation along with saving the total efficiency of coverage. Additionally, the next steps should focus on enhancing computation performance through parallelization of the optimization and smoothing stages. The framework will be a good fit for real-time or large-scale applications if multi-core or GPU-based computation is utilized. Last but not least, the incorporation of adaptive parameter tuning and automated solver selection can help the system become more user-friendly by allowing it to change dynamically with different field geometries and operational constraints without the need for human intervention. In summary, ultimate goal of these future directions is to transform the proposed framework into a real-time, terrain-aware, adaptive, and sustainable optimization platform for efficient agricultural field coverage.

Bibliography

[1] L. Dong, W. Yang, L. Wen *et al.*, "Continuous curvature wayline generation using quadratic optimisation for agricultural machine guidance," *Biosystems Engineering*, vol. 238, pp. 22–37, 2024.

[2] I. A. Hameed, D. D. Bochtis, C. G. Sørensen, and M. Nørremark, "Automated generation of guidance lines for operational field planning," *Biosystems Engineering*, vol. 107, no. 4, pp. 294–306, 2010.

[3] I. A. Hameed, D. D. Bochtis, and C. G. Sørensen, "An optimized field coverage planning approach for navigation of agricultural robots in fields involving obstacle areas," *International Journal of Advanced Robotic Systems*, vol. 10, p. 231, 2013.

[4] D. D. Bochtis and C. G. Sørensen, "The vehicle routing problem in field logistics: Part i," *Biosystems Engineering*, vol. 104, no. 4, pp. 447–457, 2009.

[5] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *Precision Agriculture*, vol. 8, no. 1–2, pp. 39–55, 2007.

[6] E. Garcia and C. Gonzalez de Santos, "Complete coverage path planning for agricultural field machines," *Robotics and Autonomous Systems*, vol. 19, no. 1, pp. 75–83, 2000.

[7] J. Jin and L. Tang, "Optimal coverage path planning for arable farming on complex field topographies," *Transactions of the ASABE*, vol. 53, no. 6, pp. 1699–1707, 2010.

[8] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[9] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[10] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *Journal of Field Robotics*, vol. 26, no. 8, pp. 651–668, 2009.

[11] J. Backman, P. Piirainen, and T. Oksanen, "Smooth turning path generation for agricultural vehicles in headlands," *Biosystems Engineering*, vol. 139, pp. 76–86, 2015.

[12] D. Sabelhaus, L. P. Meyer zu Helligen, and P. Schulze-Lammers, "Using continuous-curvature paths to generate feasible headland turn manoeuvres," *Biosystems Engineering*, vol. 116, no. 4, pp. 399–409, 2013.

[13] M. Elbanhawi, M. Simic, and R. N. Jazar, "Solutions for path planning using spline parameterization," in *Nonlinear Approaches in Engineering Applications*. Springer, 2018, pp. 277–306.

[14] M. Elbanhawi and M. Simic, "Continuous path smoothing for car-like robots using b-spline curves," *Journal of Intelligent and Robotic Systems*, vol. 77, pp. 1–15, 2014.

[15] T. Maekawa, T. Noda, S. Tamura, T. Ozaki, and K. Machida, "Curvature continuous path generation for autonomous vehicle using b-spline curves," *Computer-Aided Design*, vol. 42, no. 4, pp. 350–359, 2010.

[16] S. Jalel, P. Marthon, and A. Hamouda, "A new path generation algorithm based on accurate nurbs curves," *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, p. 75, 2016. [Online]. Available: https://doi.org/10.5772/63072

[17] H. Xie and H. Qin, "Automatic knot determination of nurbs for interactive geometric design," *Proceedings of the International Conference on Shape Modeling and Applications (SMI)*, pp. 267–275, 2001. [Online]. Available: https://doi.org/10.1109/SMA.2001.923389

[18] M. Höffmann, S. Patel, and C. Büskens, "Weight-optimized nurbs curves: Headland paths for nonholonomic field robots," in *2022 8th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE, 2022, pp. 81–85.

[19] K. Wan, D. Zhao, and Y. Li, "Fast path planning method for agricultural robot automatic guidance based on cubic spline interpolation," *Computers and Electronics in Agriculture*, vol. 210, p. 107993, 2023.

[20] J.-C. Chien, C.-L. Chang, and C.-C. Yu, "Fast path planning method for agricultural robot automatic guidance based on cubic spline interpolation in strip farming," in *2022 International Conference on System Science and Engineering (ICSSE)*. IEEE, 2022, pp. 63–67.

[21] I. A. Hameed, "Side-to-side 3d coverage path planning approach for agricultural robots to minimize skip or overlap areas between swaths," *Computers and Electronics in Agriculture*, vol. 128, pp. 36–45, 2016.

[22] T. Groen and M. Nørremark, "Increasing the efficiency of harvesting machinery through improved route planning," *Agricultural Engineering International: CIGR Journal*, vol. 23, no. 1, pp. 45–55, 2021.

[23] W. Zhao, Z. Li, and Y. Chen, "Optimization of agricultural field operations considering soil erosion and terrain slope," *Agronomy*, vol. 13, no. 5, p. 1151, 2023.

[24] M. Spekken, J. E. Molin, and S. de Bruin, "Planning machine paths and row crop patterns on steep surfaces to minimize soil erosion," *Computers and Electronics in Agriculture*, vol. 127, pp. 688–700, 2016.

[25] W. E. Hart, M. L. Bynum, G. A. Hackebeil, C. D. Laird, B. L. Nicholson, J. D. Siirola, J.-P. Watson, and D. L. Woodruff, *Pyomo: Optimization Modeling in Python*, 3rd ed. Cham: Springer, 2020.

[26] L. T. Biegler, *Optimization Modeling with Python*. SIAM, 2020.

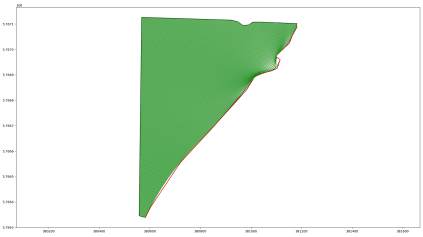
[27] G. Mier, J. Valente, and S. de Bruin, "Fields2cover: An open-source coverage path planning library for unmanned agricultural vehicles," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2166–2172, 2023. [Online]. Available: https://github.com/Fields2Cover/Fields2Cover

[28] Y. Liu, "Effect of knot vectors on b-spline curves and surfaces," MEC 572 Term Paper, State University of New York at Stony Brook, 2021.

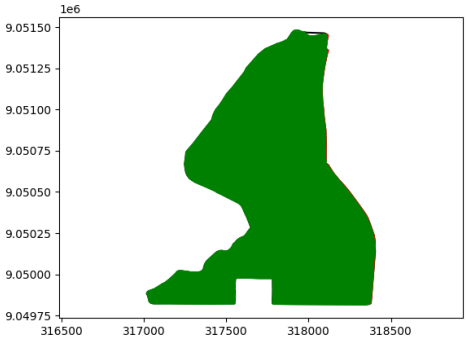
[29] J. Wyss-Gallifent, "Math431: Bézier curves," https://www2.math.umd.edu/~jwr/MATH431/, 2021, department of Mathematics, University of Maryland.

Appendix A

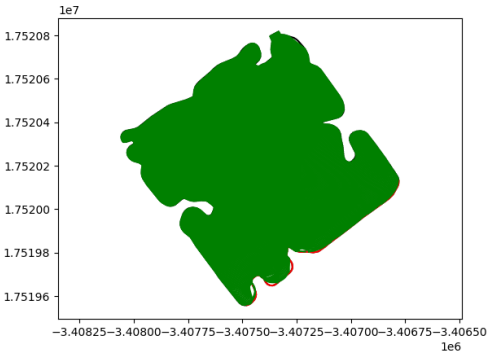
Complex Field Senarios



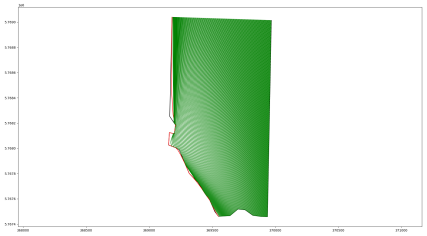
(a) Field 1



(b) Field 2



(c) Field 3



(d) Field 4

Figure A.1: Waylines generation results for Complex real world fields