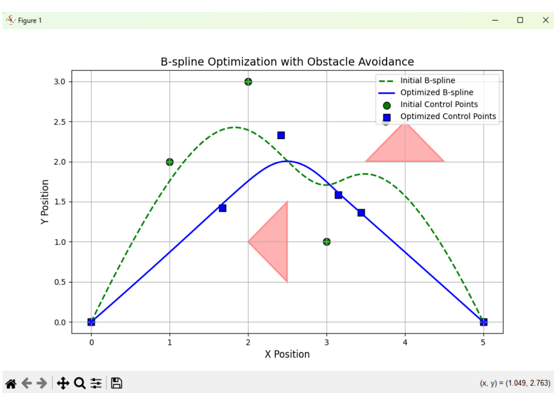
Obstacle Avoidance using B-spline Functions

Introduction

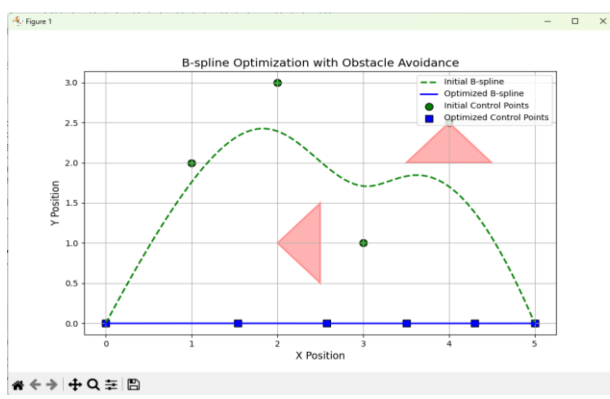
- Navigation in complex environments is a fundamental challenge for robots and automated systems.
- Obstacle avoidance is a critical component of successful navigation.
- B-spline curves offer a flexible and smooth way to represent paths.
- This presentation details a manual adjustment approach to B-spline-based obstacle avoidance.

B-spline Basics

- B-splines are piecewise polynomial curves defined by a set of control points.
- Moving a control point only affects a local region of the curve, providing intuitive control.
- The curve lies within the convex hull of its control points.



Control Point no. 5 is [3.75,2.5]



Control Point no. 5 is [4,2.5]

Problem Formulation - The Obstacle Avoidance Problem

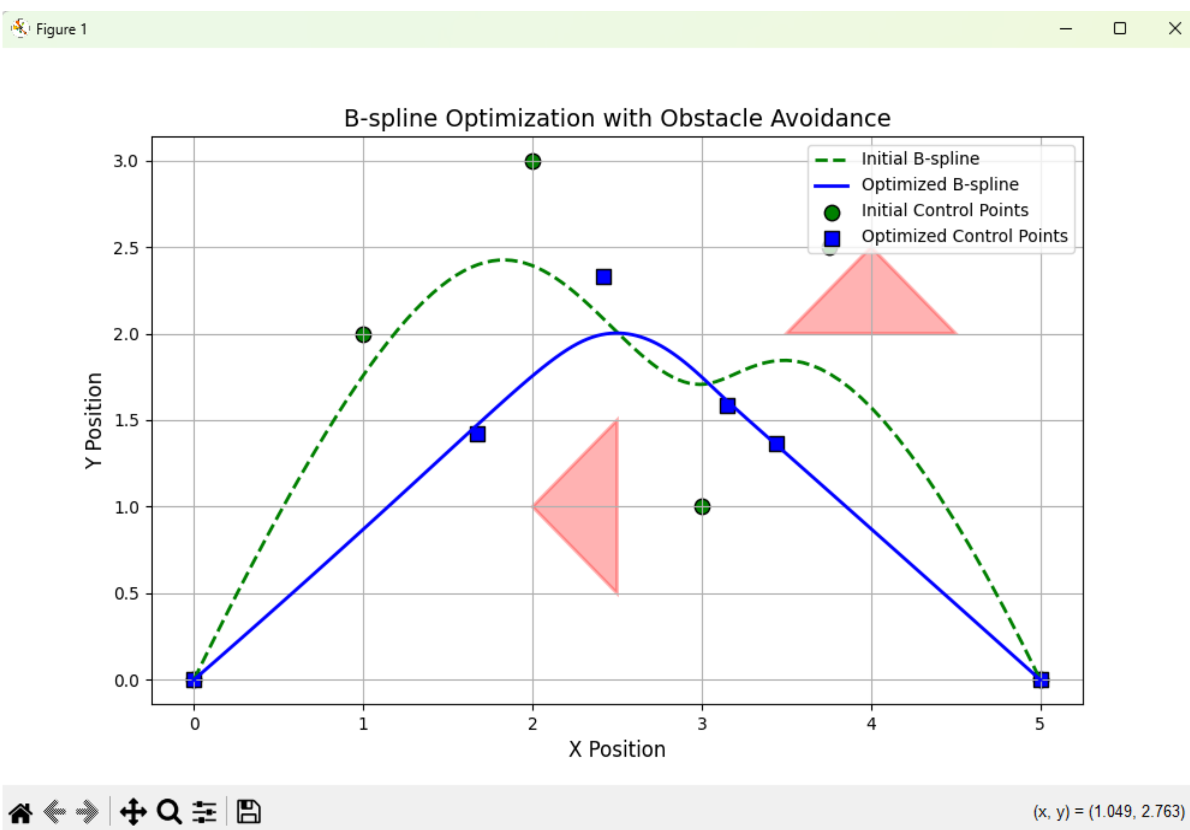
- Given a set of initial control points defining a B-spline curve and a set of obstacles.

- The goal is to adjust the control points such that the B-spline curve avoids all obstacles.
- We aim to achieve this while maintaining the smoothness and general shape of the path.

Manual Control Point Adjustment

- Control points were iteratively adjusted using a Python script and visual feedback.
- The process involved:
  - Visualizing the B-spline curve and obstacles.
  - Manually modifying the coordinates of control points in the code.
  - Re-plotting the curve to assess the impact of adjustments.
- This process was repeated until a satisfactory path, clear of all obstacles, was achieved.

Results – Successful Path adjustment



- The manually adjusted curve successfully avoids all obstacles.
- The path maintains a smooth shape, suitable for navigation.
- The green dashed line shows the initial B-spline, and the blue line shows the manually adjusted path. The red polygons represent the obstacles. The circles and squares represent the initial and adjusted control points, respectively.

Real-world applications – applications and scalability

- Robotics: Path planning for robots in warehouses, manufacturing, or exploration.
- Autonomous Vehicles: Lane changing and obstacle avoidance for self-driving cars.
- Computer Graphics: Creating smooth curves and animations.
- Scalability: While manual adjustment is suitable for simple scenarios, automated optimization techniques are necessary for complex environments with many obstacles or dynamic environments.

Conclusion and future-works

- Manual adjustment of B-spline control points offers an intuitive way to achieve obstacle avoidance.
- The process highlights the importance of understanding B-spline properties and their relationship to control points.
- Future work will focus on implementing automated optimization algorithms for greater efficiency and scalability.