Bezier Curve Optimization

1. Change to basis of the trajectory from monomial polynomial to Bernstein polynomial
2. Make use of properties of Bezier Curve to generate equations and inequations

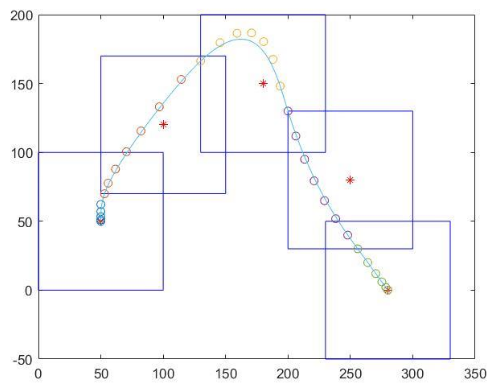
    a. Endpoint interpolation. The Bezier curve always starts at the first control point, ends at the last control point, and never pass any other control points.

    b. Convex hull. The Bezier curve $B(t)$consists of a set of control points ci are entirely confined
    within the convex hull defined by all these control points.

    c. Hodograph. The derivative curve $B'(t)$of a Bezier curve $B(t)$ is called as hodograph, and it is also a Bezier curve with control points defined by $n\cdot(c_{i+1}-c_i)$, where n is the degree.

    d. Fixed time interval. A Bezier curve is always defined on

3. Carefully write the code in MATLAB with respect to different equations and inequations.
4. Solve it by QP and display the optimized curve.



Simulation Resutls

Discussion

1. The picture I got is different from other student's result. I have checked many times and don't know why. I hope that the teaching assistant may give me some suggestions
2. The following formula (8) in [1] is suspicious, as it is against (12)

$$a_{\mu j}^{0,i} = c_{\mu j}^{i}, a_{\mu j}^{l,i} = \frac{n!}{(n-l)!} \cdot (a_{\mu j}^{l-1,i+1} - a_{\mu j}^{l-1,i}), \ \ l \geq 1, \ \ (8)$$

$$a_m^- \leq n \cdot (n-1) \cdot (c_{\mu j}^{i} - 2c_{\mu j}^{i-1} + c_{\mu j}^{i-2})/s_j \leq a_m^+. \quad (12)$$

3. How to scale the time is still not clear for me, which needs further efforts.

Reference

[1] Gao, F., Wu, W., Lin, Y., & Shen, S. (2018, May). Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 344-351). IEEE.