Cyber Planner Code Report v0.1.0 (C++ Version)

Zirui Zhang, Yan Chen, Ruoqi Xu, and Fujun Ruan

***Abstract*—Traditional FRC teams commonly use AD* and forward-backward traversal methods for trajectory planning. However, these approaches show limitations when applied to the configuration space of cascading robotic arms, where the L2 norm is not meaningful. We present a new arm planning algorithm to overcome these limitations. The algorithm is based on a finite convex polygon set representation of objects, where path planning is built upon a Lipschitz continuous artificial potential field. In trajectory planning, constraints on current and voltage are incorporated, and time-optimal trajectories are derived using the Augmented Lagrangian method and the L-BFGS optimization algorithm.**

I. Introduction

PRACTICAL trajectory planning is crucial for maximizing a team's performance in FRC (FIRST Robotics Competition). Traditional AD* and forward-backward traversal have been widely adopted for path planning and trajectory optimization in drivetrain. However, these methods face significant challenges when applied to more complex robotic systems, such as multi-joint robotic arms, where the conventional L2 norm does not apply effectively in configuration space.

We handle these limitations in our new arm planning algorithm for the upcoming 2025 FRC season. In Section II, we leverage finite convex polytope sets to represent object, which enable us to esatblish a Lipschitz continuous artificial potential fields and optimize the path based on it. In Section III, we build our time optimal model based on current and voltage constraints, ensuring the system's efficiency and safety. Then Augmented Lagrangian method and the L-BFGS algorithm was introduced to solve the objective function.

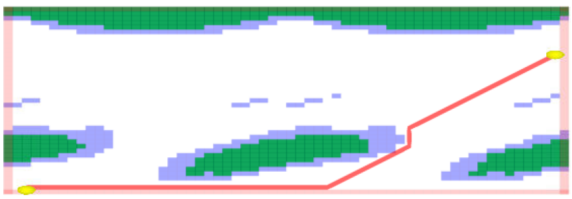
This method enhance the precision and performance of the robotic arm, overcoming the shortcomings of previous trajectory planning techniques, and offering significant advantages in dynamic and complex environments.

Z. Zhang, Y. Chen, R. Xu and F. Ruan are with Next Innovation STEM Center (see https://nifornextinnovation.com/).

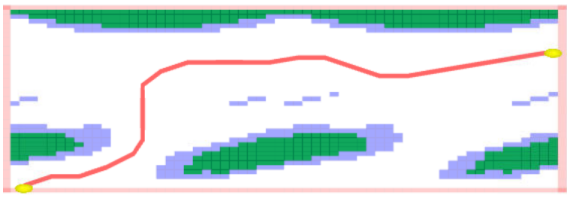
II. Path Planning

A. Improved A* Algorithm

As Image 1a shown, the original A* algorithm may always find a path that very close to the obstacles. This is often led by the heuristic function. However, if any errors occured in estimate or control, the robot may collide with the obstacles. To avoid this, we introduce artificial potential field to the cost function.



(a) Original A* Algorithm



(b) Improved A* Algorithm

Fig. 1: Difference between original and improved A* algorithm

Every grid in the map has a potential value which is calculated by the distance to the obstacles in a dynamic programming way. It can find us a heuristic path, but it is still discrete. To make the path continuous, we need to interpolate.

B. Polynomial Curve

The key idea of a polynomial curve is to make the discrete path continuous, not only in position but also in velocity and acceleration.

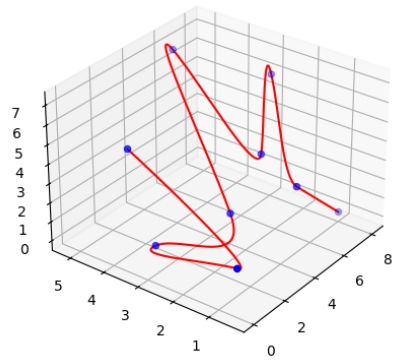


Fig. 2: 3D Cubic Spline

$$
\begin{aligned}
p_i &\in \mathcal{P} \\
p_i &= [x_i, y_i], \ \forall i \in [0, n] \\
q_{i,x}(s) &= a_{i,x}s^3 + b_{i,x}s^2 + c_{i,x}s^1 + d_{i,x} \\
\forall i &\in [0, n-1], s \in [0, 1)
\end{aligned} \tag{1}
$$

s is the normalized representation of the arc length.

The minimum order of polynomial to make the path continuous in position, velocity, and acceleration is 3. And to ensure the continuous, we need

$$
\begin{cases}
q_{i-1,x}(1) = q_{i,x}(0) = x_i \\
q'_{i-1,x}(1) = q'_{i,x}(0) \\
q''_{i-1,x}(1) = q''_{i,x}(0)
\end{cases}
, \forall i \in [1, n] \tag{2}
$$

If the object is stationary in the beginning and at the end, the velocity should be $q'_0(0) = q'_n(1) = 0$. By solving these equations we can uniquely determine a polynomial to fit those points.

$$
\begin{cases}
a_i = x_{i+2} - 2x_{i+1} + x_i \\
b_i = -x_{i+2} + 2x_{i+1,k} - x_i \\
c_i = x_{i+1} - x_i \\
d_i = x_i
\end{cases}
, \forall i \in [1, n-1] \tag{3}
$$

In natural form [12].

$$
\begin{bmatrix}
2 & 1 & & & \\
1 & 4 & 1 & & \\
 & \ddots & \ddots & \ddots & \\
 & & 1 & 4 & 1 \\
 & & & 1 & 2
\end{bmatrix}
\begin{bmatrix}
D_0 \\ D_1 \\ \vdots \\ D_{m-1} \\ D_m
\end{bmatrix}
=
\begin{bmatrix}
3(x_1 - x_0) \\ 3(x_2 - x_0) \\ \vdots \\ 3(x_m - x_{m-2}) \\ 3(x_m - x_{m-1})
\end{bmatrix} \tag{4}
$$

$$
\begin{cases}
a_i = 2(x_i - x_{i+1}) + D_i + D_{i+1} \\
b_i = 3(x_{i+1} - x_i) - 2D_i - D_{i+1} \\
c_i = D_i \\
d_i = x_i
\end{cases} \tag{5}
$$

C. Continuous Potential Field

In section II.A, we proposed a method to calculate the approximate and discrete potential field. However, the potential field often cause the path a severe shaking. Although you can set the potential field below a certain threshold to zero, this still does not make the system satisfactory.

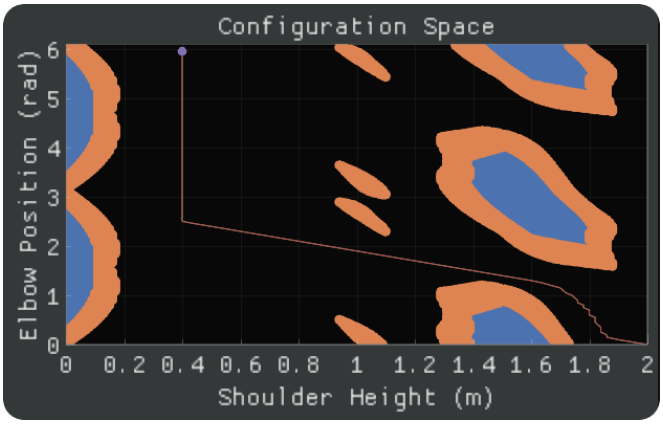


Fig. 3: The path is not smooth and has sharp jitters.

Any object can be represented by a set of convex polygons. The potential energy of a certain point is the minimum distance from the point to the edge of the polygon. In robot arm case, the certain point represent a kind of configuration of the robot arm. The potential energy is the minimum distance between the set of polygons that make up the current configuration of the robot and the set of polygons that make up the environmental obstacles.

If the point is inside the polygon, the distance is the minimum distance to the edge of the polygon. If the point is outside the polygon, the distance can be obtained by solving the Low Dimensional Quadratic Programming (LDQP) problem.

$$
\begin{aligned}
&\min_x ||x - p||^2 \\
&\text{s.t.} Ax \leq b
\end{aligned} \tag{6}
$$

p is the given point, and x is the point inside the polygon.

If the polygon is intersect or overlap with the other polygon, the distance is defined as the minimum distance from a vertex of a polygon contained by another polygon to an edge of that polygon. If the two polygons are not intersect, the distance can also be obtained by solving the LDQP problem.

$$
\begin{aligned}
&\min_{x,y} ||x - y||^2 \\
&\text{s.t.} Ax \leq b \\
&\quad Cy \leq d
\end{aligned} \tag{7}
$$

By doing so, we can establish a lipshitz continuous potential field and optimize the path by the gradient descent method. A probable objective function is

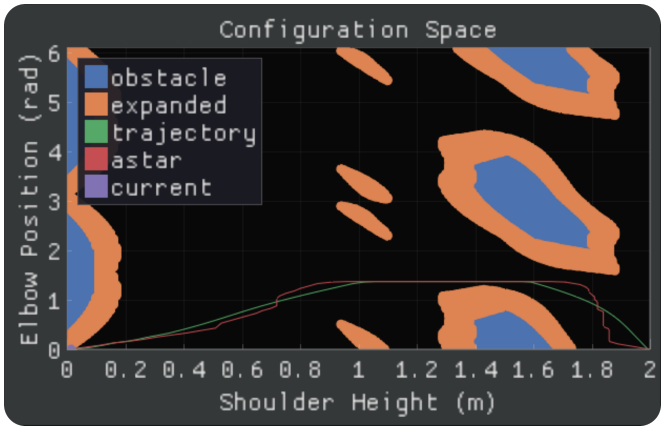


Fig. 4: The difference between the discrete and continuous potential field.

$$
\begin{aligned}
&\sum_{i=0}^{n} ||\exp(-d_{x_i})||^2 \\
&+\sum_{i=0}^{n} ||x_i - \bar{x}_i||_{\Sigma}^2 \\
&+\sum_{i=0}^{n-1} ||x_i - x_{i+1}||_{\Sigma}^2 \\
&+\sum_{i=1}^{n-1} ||x_{i-1} - 2x_i + x_{i+1}||_{\Sigma}^2
\end{aligned} \tag{8}
$$

$||x_i - \bar{x}_i||_1^2$ is the prior regularization term obtained by A* search. The Mahalanobis distance is used because the units of each dimension are different and the L2 norm is meaningless in configuration space. Squaring ensures differential flatness.

III. Trajectory Planning

Classical method to deal with the time parameterization problem in FRC is called forward and backward pass. Its essence is based on the trapezoidal curve or S-curve model, adjusting the time scale so that the motion planning meets the given constraints. However, there are difficulties in dealing multi-DOF trajectory planning, especially when it is not in Cartesian space [5], [7].

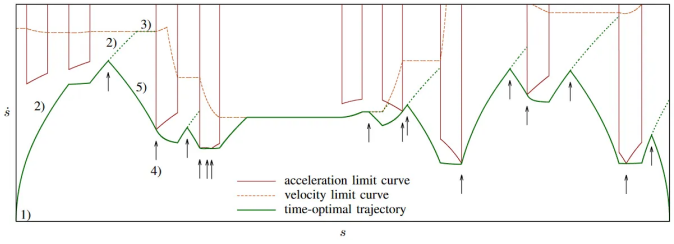


Fig. 5: Phase-plane trajectory [15]

Team 6328 proposed a Kairos solver in 2023. In their 23 season machine and code, they used an intake active avoidance strategy to avoid anti-collision. This allowed their solver to complete planning in a simpler way. So they use linear interpolation as the initial position of waypoints, and then adjusts the position of waypoints to make the speed, acceleration, and torque meet the constraints [10].

Our approach is very different from Team 6328. Instead of globally adjusting the time scale by assuming that the time intervals between points are consistent, we express the function of time with respect to arc length in discretized form.

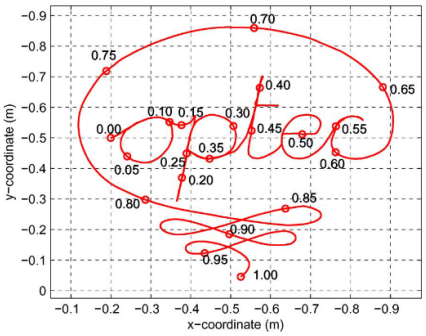


Fig. 6: The curve is uniquely determined by the arc length [2].

A. Time Optimal Path Parameterization

The total time of the trajectory is given by the integration of the whole path.

$$
T = \int_0^T 1 \mathrm{d}t = \int_0^L \frac{1}{\frac{\mathrm{d}s}{\mathrm{d}t}} \mathrm{d}s \tag{9}
$$

Velocity and acceleration constraints at elevator position x and arm radian θ are

$$
\begin{aligned}
-v_{x,\max} \le v_x \le v_{x,\max} \\
-a_{x,\max} \le a_x \le a_{x,\max} \\
-v_{\theta,\max} \le v_\theta \le v_{\theta,\max} \\
-a_{\theta,\max} \le a_\theta \le a_{\theta,\max}
\end{aligned} \tag{10}
$$

Given by chain rules, we have

$$
\begin{aligned}
\frac{\mathrm{d}x}{\mathrm{d}t} &= \frac{\mathrm{d}x}{\mathrm{d}s} \cdot \frac{\mathrm{d}s}{\mathrm{d}t} \\
\frac{\mathrm{d}^2x}{\mathrm{d}t^2} &= \frac{\mathrm{d}^2x}{\mathrm{d}s^2} \cdot \frac{\mathrm{d}s}{\mathrm{d}t} + \frac{\mathrm{d}x}{\mathrm{d}s} \cdot \frac{\mathrm{d}^2s}{\mathrm{d}t^2} \\
\frac{\mathrm{d}\theta}{\mathrm{d}t} &= \frac{\mathrm{d}\theta}{\mathrm{d}s} \cdot \frac{\mathrm{d}s}{\mathrm{d}t} \\
\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} &= \frac{\mathrm{d}^2\theta}{\mathrm{d}s^2} \cdot \frac{\mathrm{d}s}{\mathrm{d}t} + \frac{\mathrm{d}\theta}{\mathrm{d}s} \cdot \frac{\mathrm{d}^2s}{\mathrm{d}t^2}
\end{aligned} \tag{11}
$$

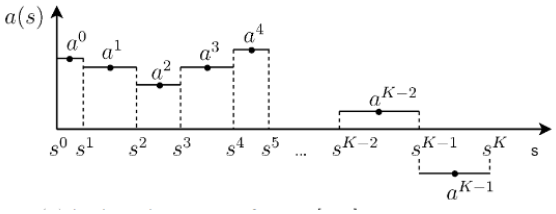
Denote

$$
a(s) = \frac{\mathrm{d}^2s}{\mathrm{d}t^2}, b(s) = \left(\frac{\mathrm{d}s}{\mathrm{d}t}\right)^2 \tag{12}
$$

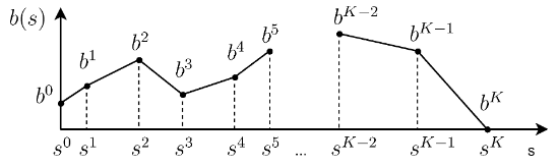
The problem is described by

$$
\begin{aligned}
\min_{a(s),b(s)} \quad & \int_0^L \frac{1}{\sqrt{b(s)}}\mathrm{d}s \\
& b'(s) = 2a(s) \\
& x'(0)\sqrt{b(0)} = 0 \\
& x'(L)\sqrt{b(L)} = 0 \\
& \theta'(0)\sqrt{b(0)} = 0 \\
& \theta'(L)\sqrt{b(L)} = 0 \\
\text{s.t.} \quad & -b(s) \le 0 \\
& [x'(s)]^2 b(s) \le v_{x,\max}^2 \\
& [\theta'(s)]^2 b(s) \le v_{\theta,\max}^2 \\
& x''(s)b(s) + x'(s)a(s) \le a_{x,\max} \\
& -x''(s)b(s) + x'(s)a(s) \le a_{x,\max} \\
& \theta''(s)b(s) + \theta'(s)a(s) \le a_{\theta,\max} \\
& -\theta''(s)b(s) + \theta'(s)a(s) \le a_{\theta,\max}
\end{aligned} \tag{13}
$$

1) Discrete Case: Considering the problem in discrete case, we have



(a) Discrete accelerations



(b) Discrete velocities

Fig. 7: Discretization of the derivative of time with respect to arc length [2].

We obtain the discretized object function

$$
\begin{aligned}
\min_{a,b} \quad & \sum_{k=1}^{K} \frac{2(s^{k+1}-s^k)}{\sqrt{b^{k+1}}+\sqrt{b^k}} \\
& \frac{b^{k+1}-b^k}{s^{k+1}-s^k} = 2a^k \qquad \forall k \in [0, K-1] \\
& x'(s^0)\sqrt{b^0} = 0 \\
& x'(s^K)\sqrt{b^K} = 0 \\
& \theta'(s^0)\sqrt{b^0} = 0 \\
& \theta'(s^K)\sqrt{b^L} = 0 \\
\text{s.t.} \quad & -b^k \le 0 \qquad \forall k \in [0, K] \\
& [x'(s^k)]^2 b^k \le v_{x,\max}^2 \qquad \forall k \in [0, K] \\
& [\theta'(s^k)]^2 b^k \le v_{\theta,\max}^2 \qquad \forall k \in [0, K] \\
& x''(s^k)b^k + x'(s^k)a^k \le a_{x,\max} \qquad \forall k \in [0, K-1] \\
& -x''(s^k)b^k + x'(s^k)a^k \le a_{x,\max} \qquad \forall k \in [0, K-1] \\
& \theta''(s^k)b^k + \theta'(s^k)a^k \le a_{\theta,\max} \qquad \forall k \in [0, K-1] \\
& -\theta''(s^k)b^k + \theta'(s^k)a^k \le a_{\theta,\max} \qquad \forall k \in [0, K-1]
\end{aligned} \tag{14}
$$

2) Second-Order Cone: To rewrite the problem to a second-order conic programming form, we first try to bound nonlinear term $\sqrt{b^k}$ with c^k, such that $\sqrt{b^k} \ge c^k, k \in [0, K]$. Going a step further, we introduce one more slack variable d^k, which satisfies $\frac{1}{c^{k+1}+c^k} \le d^k, k \in [0, K-1]$.

$$
\begin{aligned}
& \left\| \begin{matrix} b^k - 1 \\ 2c^k \end{matrix} \right\|_2 \le b^k + 1 \Leftrightarrow \begin{bmatrix} b^k + 1 \\ b^k - 1 \\ 2c^k \end{bmatrix} \in \mathcal{Q}^3 \\
& \left\| \begin{matrix} c^k + c^{k+1} - d^k \\ 2 \end{matrix} \right\|_2 \le c^k + c^{k+1} + d^k \\
& \qquad \Leftrightarrow \begin{bmatrix} c^k + c^{k+1} + d^k \\ c^k + c^{k+1} - d^k \\ 2 \end{bmatrix} \in \mathcal{Q}^3
\end{aligned} \tag{15}
$$

$$
\begin{aligned}
& \min_{a,b} \sum_{k=1}^{K} \frac{2(s^{k+1}-s^k)}{\sqrt{b^{k+1}}+\sqrt{b^k}} \Leftrightarrow \\
& \min_{a,b,c,d} \sum_{k=1}^{K} 2d^k(s^{k+1}-s^k) \\
& \text{s.t.} \quad \left\| \begin{matrix} c^k + c^{k+1} - d^k \\ 2 \end{matrix} \right\|_2 \le c^k + c^{k+1} + d^k \\
& \qquad \left\| \begin{matrix} b^k - 1 \\ 2c^k \end{matrix} \right\|_2 \le b^k + 1
\end{aligned} \tag{16}
$$

Rewriting the constraints of the optimization problem, especially the nonlinear terms $\sqrt{b_k}$, we get the final objective function [2].

$$
\min_{a,b,c,d} \sum_{k=1}^{K} 2(s^{k+1}-s^k)d^k
$$

Subject to

$$
\begin{aligned}
\begin{bmatrix} c^k + c^{k+1} + d^k \\ c^k + c^{k+1} - d^k \\ 2 \end{bmatrix} &\in \mathcal{Q}^3 && \forall k \in [0, K-1] \\
\begin{bmatrix} b^k + 1 \\ b^k - 1 \\ 2c^k \end{bmatrix} &\in \mathcal{Q}^3 && \forall k \in [0, K] \\
2(s^{k+1} - s^k)a^k + b^k - b^{k+1} &= 0 && \forall k \in [0, K-1] \\
[x'(s^0)]^2 b^0 &= 0 \\
[x'(s^K)]^2 b^K &= 0 \\
[\theta'(s^0)]^2 b^0 &= 0 \\
[\theta'(s^K)]^2 b^L &= 0 \\
-b^k &\leq 0, && \forall k \in [0, K] \\
[x'(s^k)]^2 b^k &\leq v_{x,\max}^2 && \forall k \in [0, K] \\
[\theta'(s^k)]^2 b^k &\leq v_{\theta,\max}^2 && \forall k \in [0, K] \\
x''(s^k)b^k + x'(s^k)a^k &\leq a_{x,\max} && \forall k \in [0, K-1] \\
-x''(s^k)b^k + x'(s^k)a^k &\leq a_{x,\max} && \forall k \in [0, K-1] \\
\theta''(s^k)b^k + \theta'(s^k)a^k &\leq a_{\theta,\max} && \forall k \in [0, K-1] \\
-\theta''(s^k)b^k + \theta'(s^k)a^k &\leq a_{\theta,\max} && \forall k \in [0, K-1]
\end{aligned} \tag{17}
$$

The constraints are divided into second-order cone constraints, equality constraints and inequality constraints [4].

$$
\begin{aligned}
\min_x \ & c^T x \\
\text{s.t. } & A_i x + b_i \in \mathcal{K}_i \\
& Gx = h \\
& Px \leq q
\end{aligned} \tag{18}
$$

3) Augmented Lagrangian Method: Therefore, we can relax the constraints with the Augmented Lagrangian method.

$$
\begin{aligned}
\mathcal{L}_\rho(x, \mu, \lambda, \eta) =& c^T x + \\
& \frac{\rho}{2} \sum_{i=1}^{m} ||P_{\mathcal{K}_i}(\frac{\mu_i}{\rho} - A_i x - b_i)||^2 + \\
& \frac{\rho}{2} ||Gx - h + \frac{\lambda}{\rho}||^2 + \\
& \frac{\rho}{2} ||\max[Px - q + \frac{\eta}{\rho}, 0]||^2
\end{aligned} \tag{19}
$$

$P_{\mathcal{K}}$ is the projection of a vector on a symmetric cone

$$
P_{\mathcal{K}}(v) = \arg\min_{x \in \mathcal{K}} ||v - x||^2 \tag{20}
$$

In second order cone case,

$$
P_{\mathcal{K}=\mathcal{Q}^n}(v) = \begin{cases} 0, & v_0 \leq -||v_1||_2 \\ \frac{v_0 + ||v_1||_2}{2||v_1||_2}(||v_1||_2, v_1)^T, & |v_0| < ||v_1||_2 \\ v, & v_0 \geq ||v_1||_2 \end{cases} \tag{21}
$$

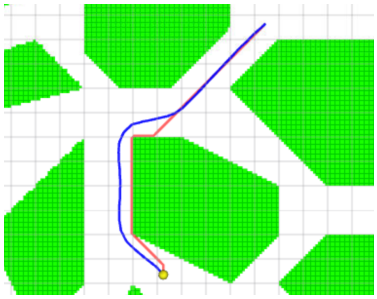
And the gradient of the augmented Lagrangian function is given by

$$
\begin{aligned}
\nabla \mathcal{L}_\rho(x, \mu, \lambda, \eta) =& c - \rho \sum_{i=1}^{m} A_i^T P_{\mathcal{K}_i}(\frac{\mu_i}{\rho} - A_i x - b_i) + \\
& \rho G^T (Gx - h + \frac{\lambda}{\rho}) + \\
& \rho P^T \{\max[Px - q + \frac{\eta}{\rho}, 0]\}
\end{aligned} \tag{22}
$$

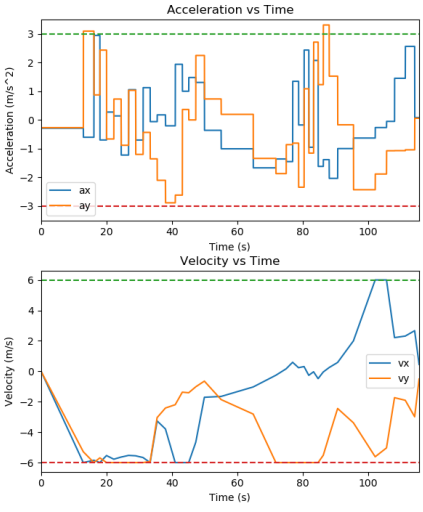
We can use a L-BFGS method to solve this convex and unconstrained function [1], [3], [16].

$$
\begin{aligned}
x &\leftarrow \arg\min_x \mathcal{L}_\rho(x, \mu, \lambda, \eta) \\
\mu_i &\leftarrow P_{\mathcal{K}_i}(\mu_i - \rho(A_i x + b_i)) \\
\lambda &\leftarrow \lambda + \rho(Gx - h) \\
\eta &\leftarrow \max[\eta + \rho(Px - q), 0] \\
\rho &\leftarrow \min[(1 + \gamma)\rho, \beta]
\end{aligned} \tag{23}
$$

γ is the growth rate of ρ and β is the upper bound of ρ, which is typically 10^3.



(a) The corresponding path



(b) Velocity and acceleration under constraints

Fig. 8: Our previous work showing constraints on velocity and acceleration [19].

B. Constraints on voltage and current

Although we can set constraints for speed and acceleration in segments instead of setting a global maximum

value, it is difficult to exhaust all possibilities and enumerate all the required constraints. A conservative constraint will waste control resources. Can we go one step further?

Under the guidance of the feedforward model, speed and acceleration together constitute a feasible control allocation. Hence, we redesign the speed and acceleration constraints into voltage and current constraints. Doing so enables the model to dynamically adjust the energy distribution on speed and acceleration, maximize the use of control resources on the field.

Elevator feedforward model, i_e means the ratio from motor radian to elevator position meter:

$$V_E = K_g + K_S \cdot \text{sign}(\dot{d}) + \frac{K_v}{i_e} \cdot \dot{d} + \frac{K_a}{i_e} \cdot \ddot{d} \tag{24}$$

Arm feedforward model, i_a means the ratio from motor radian to arm radian:

$$V_A = K_g \cdot \cos(\theta) + K_S \cdot \text{sign}(\dot{\theta}) + \frac{K_v}{i_a} \cdot \dot{\theta} + \frac{K_a}{i_a} \cdot \ddot{\theta} \tag{25}$$

The motor current at a given voltage and speed is:

$$I = -\frac{K_v}{R} \cdot \omega + \frac{1}{R} \cdot V \tag{26}$$

Notice that we used c^k to bound the lower side of $\sqrt{b^k}$. This is because

$$\frac{2(s^{k+1} - s^k)}{\sqrt{b^{k+1}} + \sqrt{b^k}} \leq \frac{2(s^{k+1} - s^k)}{c^{k+1} + c^k} \leq 2(s^{k+1} - s^k)d^k \tag{27}$$

In case, we introduce a second order cone to handle the chain constraints. The equality stands only when $\sqrt{b^k} = c^k$. How about directly let $\sqrt{b^k} = c^k$ instead of $\sqrt{b^k} \geq c^k$? So that,

$$\frac{2(s^{k+1} - s^k)}{\sqrt{b^{k+1}} + \sqrt{b^k}} = \frac{2(s^{k+1} - s^k)}{c^{k+1} + c^k} \leq 2(s^{k+1} - s^k)d^k \tag{28}$$

With quadratical constraints:

$$x^T J_j x - r_j^T x = 0, \ \forall j \in \mathcal{J} \tag{29}$$

By Augmented Lagrangian method, a dual variable ν is used to relax and bound the quadratic term. The problem is modeled by the following.

$$\begin{aligned}\mathcal{L}_\rho(x, \mu, \nu, \lambda, \eta) =& c^T x + \\ & \frac{\rho}{2} \sum_{i=1}^{m} ||P_{\mathcal{K}_i}(\frac{\mu_i}{\rho} - A_i x - b_i)||^2 + \\ & \frac{\rho}{2} \sum_{j=1}^{q} ||x^T J_j x - r_j^T x + \frac{\nu_j}{\rho}||^2 + \\ & \frac{\rho}{2} ||Gx - h + \frac{\lambda}{\rho}||^2 + \\ & \frac{\rho}{2} ||\max[Px - q + \frac{\eta}{\rho}, 0]||^2\end{aligned} \tag{30}$$

$$\begin{aligned}\nabla \mathcal{L}_\rho(x, \mu, \nu, \lambda, \eta) =& c - \rho \sum_{i=1}^{m} A_i^T P_{\mathcal{K}_i}(\frac{\mu_i}{\rho} - A_i x - b_i) + \\ & \rho \sum_{j=1}^{q} (x^T J_j - r_j^T + \frac{\nu_j}{\rho})(2J_j x - r_j) + \\ & \rho G^T (Gx - h + \frac{\lambda}{\rho}) + \\ & \rho P^T \{\max[Px - q + \frac{\eta}{\rho}, 0]\}\end{aligned} \tag{31}$$

$J^T + J = 2J$ in case J is symmetric.

$$\begin{aligned} x &\leftarrow \arg\min_x \mathcal{L}_\rho(x, \mu, \nu, \lambda, \eta) \\ \mu_i &\leftarrow P_{\mathcal{K}_i}(\mu_i - \rho(A_i x + b_i)) \\ \nu_j &\leftarrow \nu_j + \rho(x^T J_j x - r_j^T x) \\ \lambda &\leftarrow \lambda + \rho(Gx - h) \\ \eta &\leftarrow \max[\eta + \rho(Px - q), 0] \\ \rho &\leftarrow \min[(1 + \gamma)\rho, \beta] \end{aligned} \tag{32}$$

Till now, we obtain all the math tools to solve our problem.

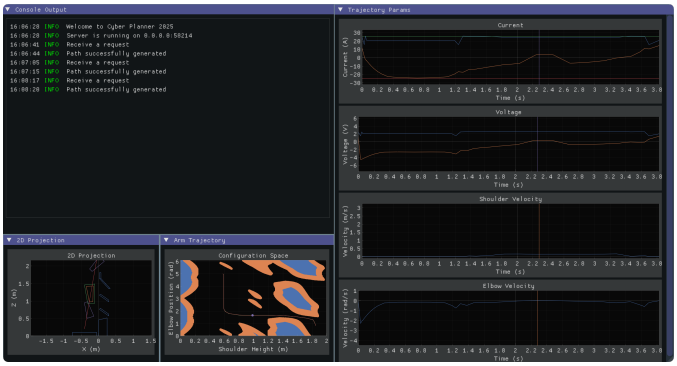


Fig. 9: Desktop Application Screenshot

References

[1] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," Mathematical Programming, vol. 45, no. 1–3, pp. 503–528, Aug. 1989, doi: 10.1007/BF01589116.

[2] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-Optimal Path Tracking for Robots: A Convex Optimization Approach," IEEE Trans. Automat. Contr., vol. 54, no. 10, pp. 2318–2327, Oct. 2009, doi: 10.1109/TAC.2009.2028959.

[3] D.-H. Li and M. Fukushima, "On the Global Convergence of the BFGS Method for Nonconvex Unconstrained Optimization Problems," SIAM J. Optim., vol. 11, no. 4, pp. 1054–1064, Jan. 2001, doi: 10.1137/S1052623499354242.

[4] F. Alizadeh and D. Goldfarb, "Second-order cone programming," Mathematical Programming, vol. 95, no. 1, pp. 3–51, Jan. 2003, doi: 10.1007/s10107-002-0339-5.

[5] F.-G. Declan, "A Dive into WPILib Trajectories," Dec. 2019. [Online]. Available: https://pietroglyph.github.io/trajectory-presentation

[6] G. Blekherman, P. A. Parrilo, and R. R. Thomas, Eds., Semidefinite Optimization and Convex Algebraic Geometry. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2012. doi: 10.1137/1.9781611972290.

[7] J. Russell, T. Bottiglieri, and A. Schuh, “Motion Planning and Control in FRC,” May 2015.
[8] J. Ji, X. Zhou, C. Xu, and F. Gao, “CMPCC: Corridor-Based Model Predictive Contouring Control for Aggressive Drone Flight,” in Experimental Robotics, B. Siciliano, C. Laschi, and O. Khatib, Eds., Cham: Springer International Publishing, 2021, pp. 37–46.
[9] L. Dong, Z. He, C. Song, and C. Sun, “A review of mobile robot motion planning methods: from classical motion planning workflows to reinforcement learning-based architectures,” Journal of Systems Engineering and Electronics, vol. 34, no. 2, pp. 439–459, 2023, doi: 10.23919/JSEE.2023.000051.
[10] Mechanical Advantage, Robot Code 2023. (2023). [Online]. Available: https://github.com/Mechanical-Advantage/RobotCode2023
[11] R. E. Banchs, “Natural Quartic Spline,” presented at the Research progress report #07 on Time Harmonic Field Electric Logging, The University of Texas at Austin, 1996.
[12] R. H. Bartels, J. C. Beatty, and B. A. Barsky, An introduction to splines for use in computer graphics and geometric modeling. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987.
[13] R. M. Freund, “Introduction to Semidefinite Programming (SDP)”.
[14] R. Madani, M. Ashraphijuo, M. Kheirandishfard, and A. Atamturk, “Parabolic Relaxation for Quadratically-constrained Quadratic Programming – Part I: Definitions & Basic Properties,” Aug. 07, 2022, arXiv: arXiv:2208.03622. doi: 10.48550/arXiv.2208.03622.
[15] T. Kunz and M. Stilman, “Time-Optimal Trajectory Generation for Path Following with Bounded Acceleration and Velocity”.
[16] W. Zhepei, LBFGS-Lite. (2021). [Online]. Available: https://github.com/ZJU-FAST-Lab/LBFGS-Lite
[17] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, “EGO-Planner: An ESDF-Free Gradient-Based Local Planner for Quadrotors,” IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 478–485, 2021, doi: 10.1109/LRA.2020.3047728.
[18] Y. Wu, Z. Ding, C. Xu, and F. Gao, “External Forces Resilient Safe Motion Planning for Quadrotor,” IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 8506–8513, 2021, doi: 10.1109/LRA.2021.3110316.
[19] Z. Zirui, CoTiMo. (2024). [Online]. Available: https://github.com/zhangzrjerry/cotimo
[20] Z. Zirui, FRC Auto Driver. (2022). [Online]. Available: https://github.com/ZhangzrJerry/FRC_AutoDriver