Path Planning for Autonomous Vehicles based on Nonlinear MPC with using a Kinematic Bicycle Model

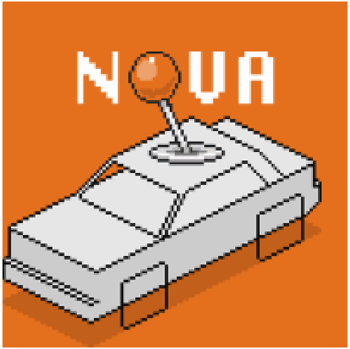
Jonas Wagner

MECH 6V29 - MPC
Fall 2023

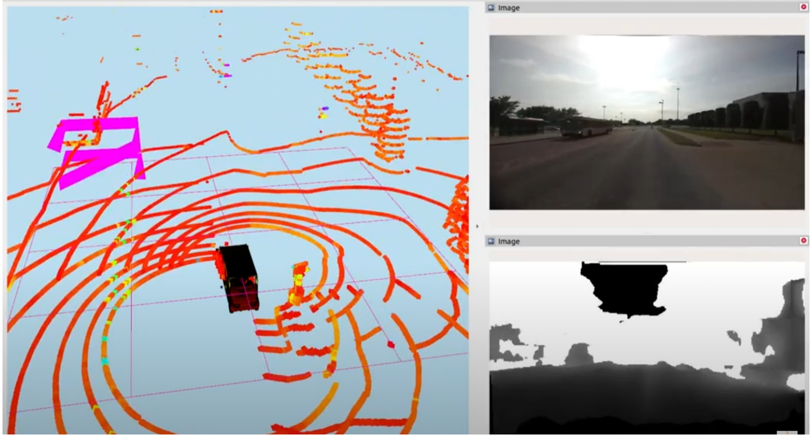
Outline

Outline

NOVA: Hail Bopp [1]

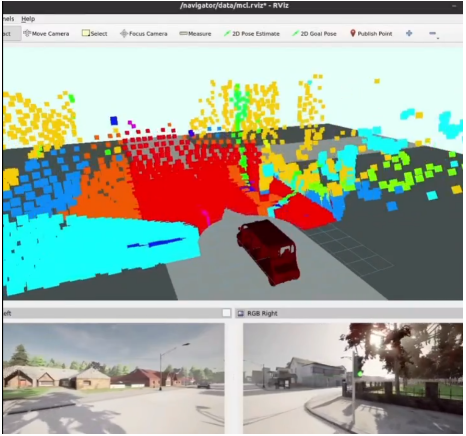


NOVA: Perception [1]



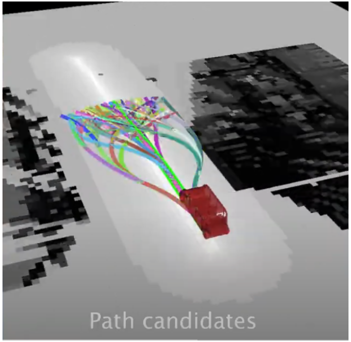
NOVA: Navigator [1]

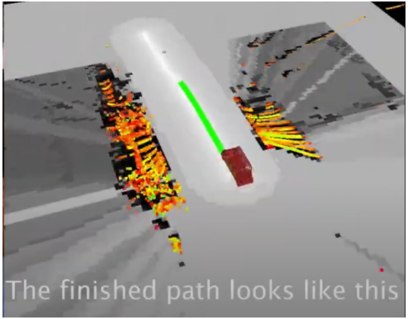
- Navigator is the self-driving software stack being developed by NOVA.
- Simulations done in then deployed to Hail Bopp.



Path Planning Objective [1]

Current Approach (random path generation and ranking)



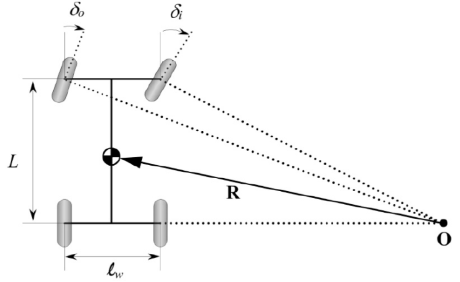


Objective: Develop NL-MPC based path planning that is better than the current aproach. (Spoiler: NL-MPC is too complicated for this portion of Navigator's stack at this stage and many more efficient techniques exist. MPC will become more benificial when robustness/security gaurentees are required/desired).

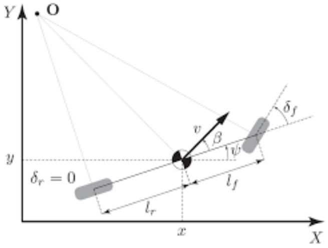
Outline

Vehicle Kinematic Models [2, 3]

Akerman Steering Model



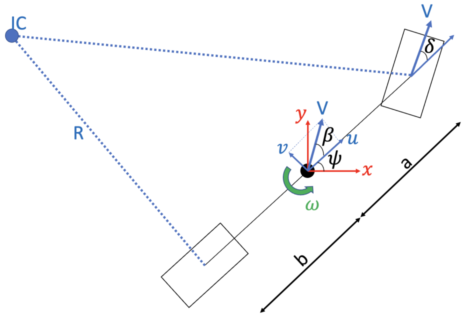
Bicycle Approximation



Akerman steering can be approximated as a bicycle model in most cases.

$$\delta \approx \frac{\delta_o + \delta_i}{2}$$

Kinematic Bicycle Model [4]



Simple nonlinear kinematics model equations:

$$\begin{cases} \dot{x} = V\cos(\psi + \beta) \\ \dot{y} = V\sin(\psi + \beta) \\ \dot{\psi} = \dfrac{V\cos(\beta)}{l_f + l_r}(\tan(\delta_f) - \tan(\delta_r)) \\ \dot{\theta} = \psi \end{cases} \tag{1}$$

where

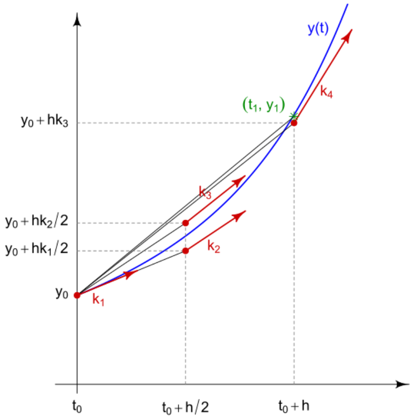
$$\beta = \tan^{-1}\left(\frac{l_f\tan(\delta_r) + l_r\tan(\delta_f)}{l_f + l_r}\right) \tag{2}$$

$\delta_r = 0$, $l_f = a = 0.7[m]$, and $l_r = b = 0.7[m]$.

Outline

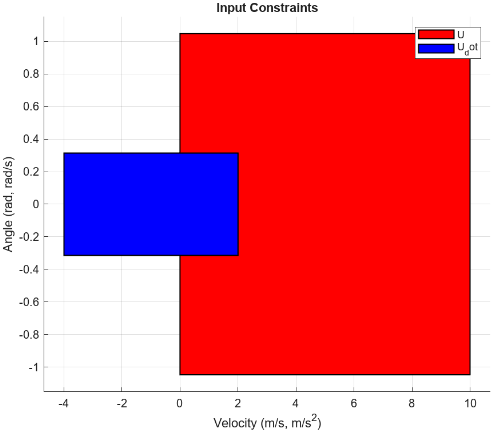
Model Discretization[5]

Discretized with $\Delta t = 1$ [s] using **RK4 method** (euler method doesn't work without very small Δt)



Input Constraints

$V \in [0, 10]$ [m/s]
$\theta \in \pm\pi/3$ [rad]
$\dot{V} \in [-4, 2]$ [m/s$\hat{2}$]
$\dot{\theta} \in \pm\pi/10$ [rad/s]



MPC Optimization Problem

Parameters:

- $N = 15$
- $\Delta t = 1$ [s]
- Multiple $q(x_k, u_k)$ and $p(x_n)$ tested
- $h(x_k, u_k) \mid u_k \in \mathbf{U} \wedge \dot{u}_k \in \dot{\mathbf{U}}$
- (for closed-loop sim: $T = 10$ [s])

$$J_0^*(x_0) = \min_{U_0} \sum_{k=0}^{N-1} q(x_k, u_k) + p(x_N)$$

$$\text{s.t.}$$

$$x_{k+1} = f(x_k, u_k),\ k \in \{0, 1, \ldots, N-1\}$$

$$h(x_k, u_k) \leq 0, \quad k \in \{0, 1, \ldots, N-1\}$$

Outline

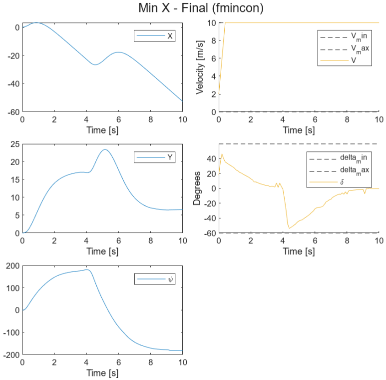
fmincon vs ipopt (Computation Time)

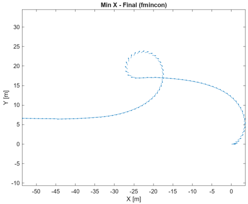
The results and computation time are GREATLY affected by the selection of the nonlinear optimization solver used.
The following is a simple comparison of computation time (just for demonstration... not a controlled case study)

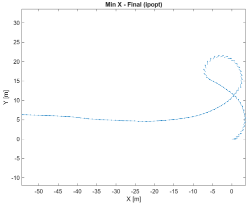
	FMINCON	**IPOPT**
Min Y - Unstructured	243.79	164.75
Min Y - Final	235.91	155.6
Min Y - Ref X and theta	438.67	71.79
Min X - Final	**1077.6**	**164.58**
Min X with +180	395.71	191.69
Min X with special ref	**344.52**	**165.16**
Max Y - Final	**331.71**	**140.49**
Max Y - Ref X theta	**665.82**	**85.492**

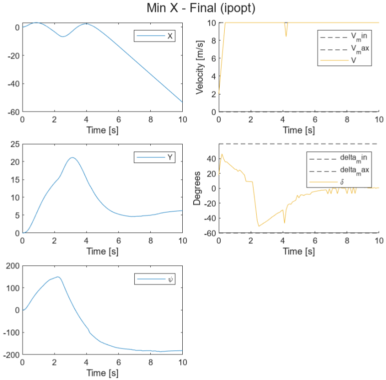
fmincon vs ipopt (Min X w/ Final - Both Fail)

$$J = x_{N+1}$$



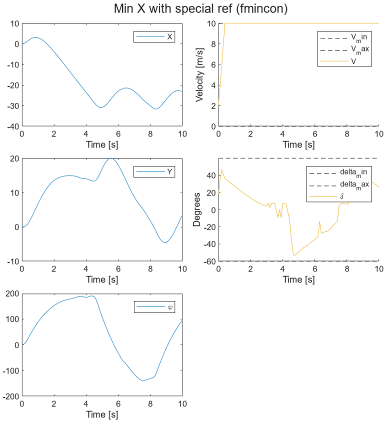


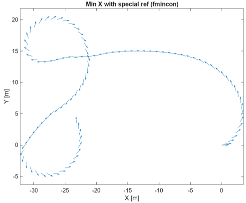


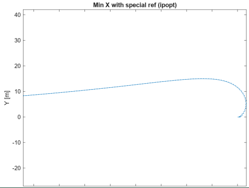


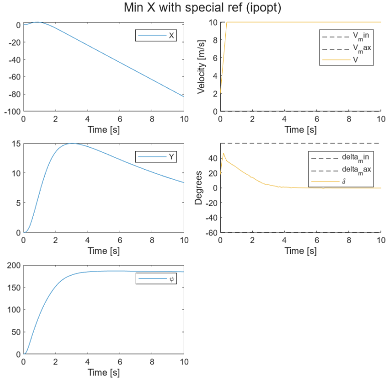
fmincon vs ipopt (Min X w/ special ref - fmincon fails)

$$J_k = x_k + (y_k - 0)^2/100 + (\theta_k - \pi)^2$$



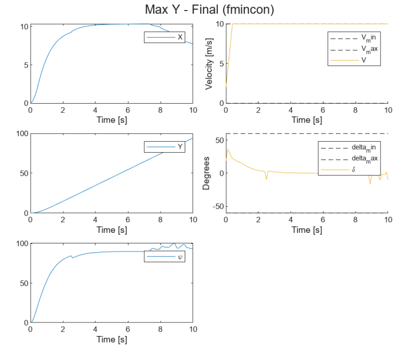


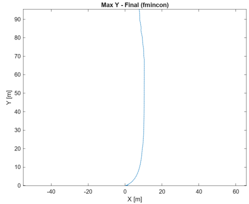


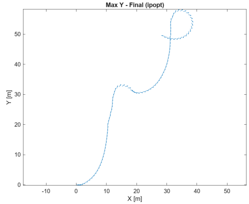


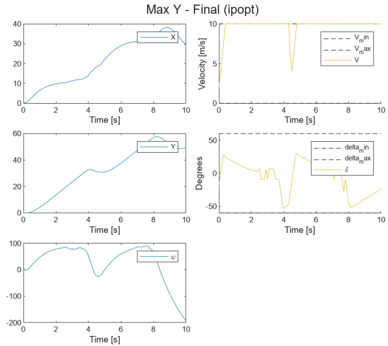
fmincon vs ipopt (Max Y w/ Final - ipopt fails)

$$J = -y_{N+1}$$



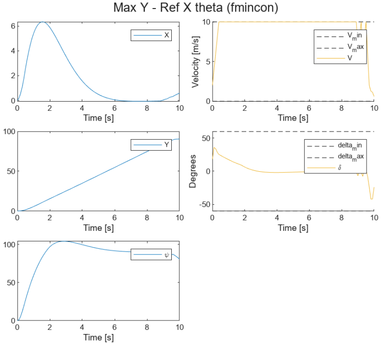


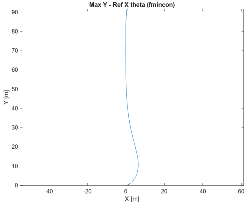


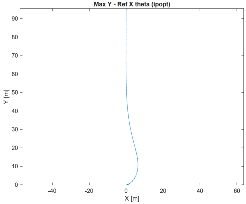


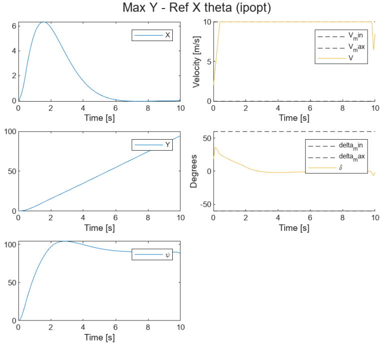
fmincon vs ipopt (Max Y with refs)

$$J_k = -y_k + (x_k - 0)^2 + (\theta_k - \pi/2)^2$$









Bibliography I

NOVA.
nova-utd.github.io, 2023.

Daniele Casanova.
On minimum time vehicle manoeuvring: The theoretical optimal lap.
2000.

Peeroon Pete Ramanata.
Optimal vehicle path generator using optimization methods.
PhD thesis, Virginia Tech, 1998.

Rajesh Rajamani.
Lateral Vehicle Dynamics, pages 15–46.
Springer US, Boston, MA, 2012.

Bibliography II

wikipedia.
Runge–kutta methods.