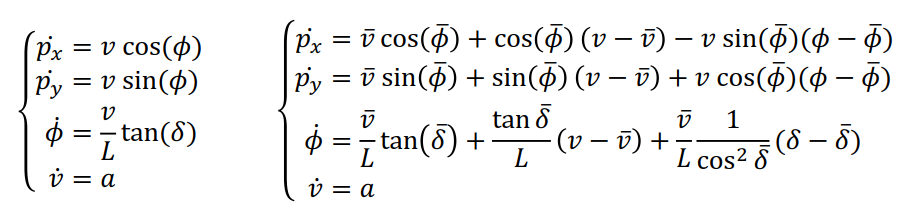
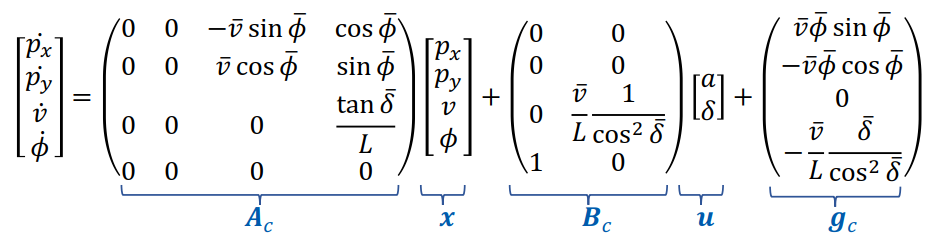
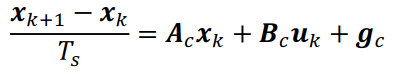
* **Get**

Convert linear model to discrete-time using forward Euler method

Linearization and discretization the model equation:









* **Set cost function**

The position error and polar angle error of the car are required to be as small as possible in the current position and trajectory:

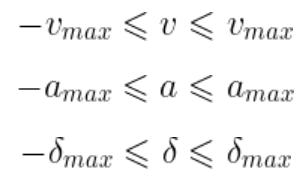
\large J=(x-x_{ref})^2+(y-y_{ref})^2+\rho (\phi-\phi_{ref})^2

State maxtrix ，and

So the cost funtion become:

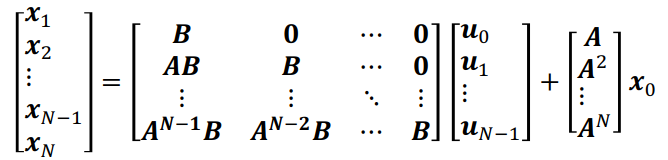
Note that when i=horizon:

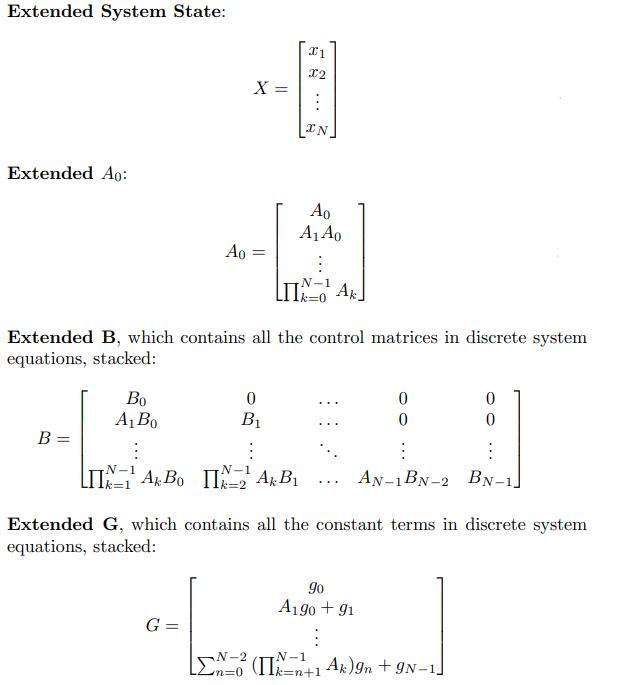
* **Set the constrain**



Note that the constrain of :

* **Solving quadratic programming problems**

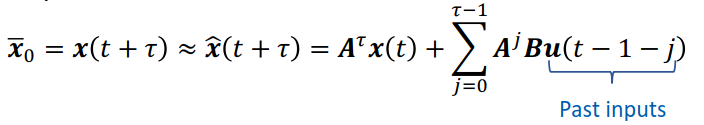




\large \textbf{q}_x = -\textbf{Q}^T\textbf{x}_{ref}

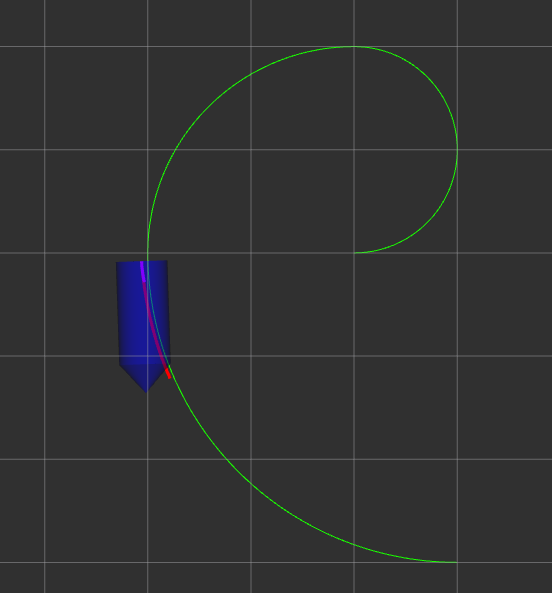
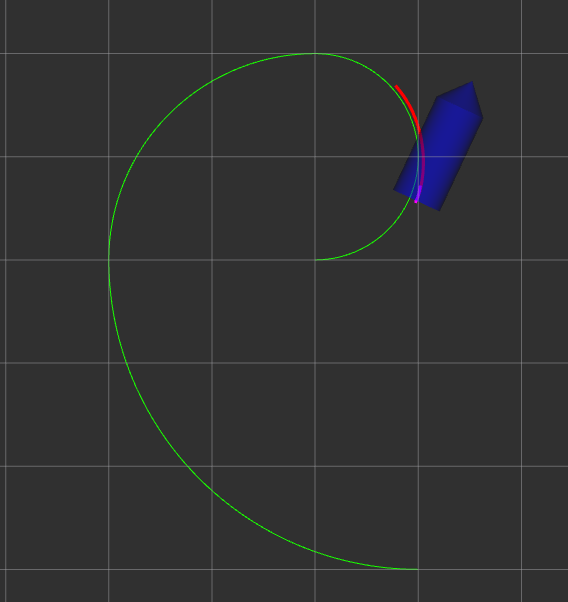
* **Get delay MPC**

The initial state of the system can be defined as the amount of state after the current moment

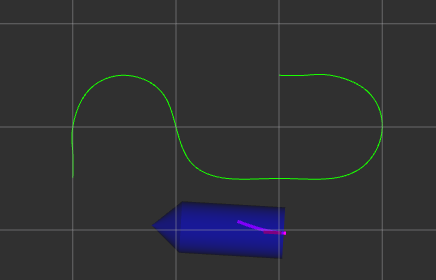


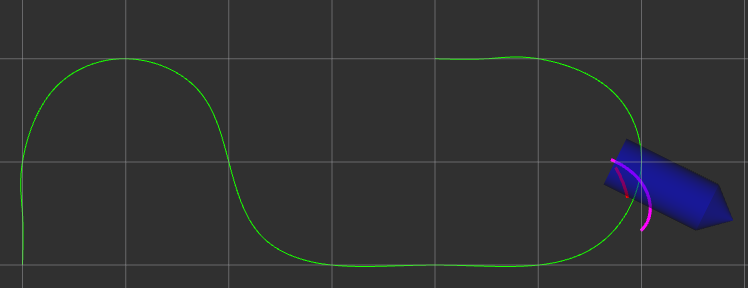
Let we get , and according , finally we can get from

* **Simulation result**

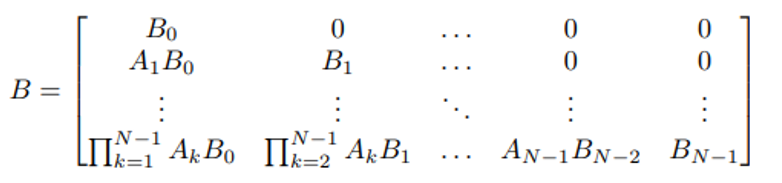


MPC performance is affected by the initial position and yaw angle and horizon, so the other two paths perform not very well





* **Bugs:**



Multiplying A sequentially from right to left causes numerical instability when N\_ is large.

Dynamic programming from top to bottom multiplies one A at a time will not cause numerical instability