Part 1: Control System Design

a

I will implement the algorithm introduced in

Turri, V., Carvalho, A., Tseng, H.E., Johansson, K.H. and Borrelli, F., 2013, October. Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. In 16th international IEEE conference on intelligent transportation systems (ITSC 2013) (pp. 378-383). IEEE.

I provide a table which illustrates all the parameter values I have used during the implementation. CoG stands for Center of Gravity.

The Parameter	Value
Car Mass	2050 kg
Car Inertia Around CoG	3344 $kg\,m^2$
Friction Coefficient between the Ground and Car	0.3
Air Density	1.225 $kg\,m^3$
Aerodynamic Drag Coefficient of the Car	0.31
Car Frontal Cross-Section Area	4 m^2
Distance of CoG to Front	1.1 m
Distance of CoG to Back	0.9 m
C_{fL} C_{fU} C_{rL} C_{rU}	3000/4 4000/4 3000/4 4000/4

b

The given state-space system is not Time-Invariant. Hence, a simple LQR would fail. But MPC method enables me to control the given **Time-Variant** system. The procedure is as follows:

1. Assume that the system is **Time-Invariant**. Compute the corresponding input with a proper horizon length.
2. Apply the iteration for one step to compute the next state values.
3. Update the system dynamics, i.e., recompute the matrices A and B.
4. Go to first step.

c - Mathematical Analysis of the System Dynamics

The equations governing the system are given as

$$m\ddot{y} = -m\dot{x}\dot{\psi} + C_{fL}\alpha_f + C_{rL}\alpha_r \tag{1}$$
$$I\ddot{\psi} = aC_{fL}\alpha_f - bC_{rU}\alpha_r \tag{2}$$
$$\dot{e}_\psi = \dot{\psi} - \dot{x}\psi_r \tag{3}$$
$$\dot{e}_y = \dot{y} + \dot{x}e_\psi \tag{4}$$
$$\dot{\delta} = u \tag{5}$$

where α_f and α_r are expressed as

$$\alpha_f = \frac{\dot{y} + a\dot{\psi}}{\dot{x}} - \delta \qquad\qquad \alpha_r = \frac{\dot{y} - b\dot{\psi}}{\dot{x}} \tag{6}$$

The state vector 's' can be expressed as follows

$$s = \begin{bmatrix} \dot{y} \\ \dot{\psi} \\ e_\psi \\ e_y \\ \delta \end{bmatrix} \tag{7}$$

Now, put all the equations in state-space form

$$\dot{s} = \begin{bmatrix} \frac{C_{fL}+C_{rL}}{m\dot{x}} & \frac{a\,C_{fL}-b\,C_{rL}}{m\dot{x}} - \dot{x} & 0 & 0 & -\frac{C_{fL}}{m} \\ \frac{a\,C_{fL}-b\,C_{rU}}{I\dot{x}} & \frac{a^2\,C_{fL}+b^2\,C_{rU}}{I\dot{x}} & 0 & 0 & -\frac{a\,C_{fL}}{I} \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & \dot{x} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} s + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ -\dot{x}\psi_r \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

Notice that the state-space in equation 8 has an offset term. Hence, it is **not** linear. But, with algebraic manipulation, I can get cancel out the offset term.

$$\dot{s} = As + Bu + C_{offset} \tag{9}$$

Define a new variable $\bar{s}$ such that $s = \bar{s} + C_s$. Then

$$\dot{\bar{s}} = \dot{s} = As + Bu + C_{offset} = A(\bar{s} + C_s) + Bu + C_{offset} \tag{10}$$
$$= A\bar{s} + Bu + A\,C_s + C_{offset} \tag{11}$$

If there exist a C_s such that $A\,C_s + C_{offset} = 0$, the equation 11 turns out to be **linear**. Note that A is not invertible. Hence, I do not guarantee that there exists such a C_s matrix. But it is possible to develop an algorithm which can compute one if such a C_s exists.

Finally I have a Linear Time Variant system

$$\dot{\bar{s}} = \bar{s}A(t) + Bu \tag{12}$$

The system is **time variant** because the parameters of matrix A depends on the velocity of the car ($\dot{x}$) and the radius of the road ($r = 1/\psi_r$). Note that the road curvature ψ_r only affects the state offset C_s.

A Few Comments on Offset

The original state-space model given in equation 8 has an offset term C_{offset}. Hence, the system is not linear. But the mathematical tools such as LQR design methods, MPC design methods are all defined for linear systems. Hence, we need to deal with the offset.

When I define $\bar{s}$ and compute the corresponding C_s, I declare that the original state vector s has to have an offset. The algorithm will try to make the state vector $\bar{s}$ go to zero. Hence, in steady-state, the original state vector s will be nonzero.

Consider that a vehicle is moving in a circular road. In such a scenario, one can imagine that the angle of the wheel cannot be zero. The wheel angle has to be nonzero so that the vehicle can follow the lane. Notice that the wheel angle is just one of the terms in the original state vector s. Such an intuition can be carried out for other terms of the original state vector. Hence, the computed offset term C_s is well expected and required for lane keeping.

Discretization

The MPC design method is valid only for discrete systems. But, the state-space system given in equation 12 is continuous. Hence, I need to discretize the system. Note the following notation:

The discretized state vector $\bar{s}$ is represented by d_i where $i \epsilon \{0, 1, 2....\}$

The discretization period is given by Δt.

We have the equality

$$d_k = \bar{s}|_{t=k\times\Delta t} \tag{13}$$

For discretization, I propose two methods:

Method 1: Forward Euler Method

I use the approximation $d_{k+1} - d_k = \Delta t \times \dot{\bar{s}}|_{t=k\times\Delta t}$ where the term $\dot{\bar{s}}|_{t=k\times\Delta t}$ stands for the derivative of the state vector $\bar{s}$ at time instant $t = k \times \Delta t$.

$$\dot{d}_{k+1} = d_k + \Delta t \, \dot{\bar{s}}|_{t=k\times\Delta t} \tag{14}$$

$$= d_k + \Delta t \left(A_k \, d_k + Bu \right) = \left(I + \Delta t \, A_k \right) x_k + \Delta t \, B \;\; u \tag{15}$$

$$\rightarrow \qquad A_d = I + \Delta t \, A \qquad\qquad B_d = \Delta t \, B \tag{16}$$

Method 2: Formal Discretization Formula

Use the well known formulas

$$A_d = e^{A\times\Delta t} \qquad\qquad B_d = \int_0^{\Delta t} e^{A\times\tau} B \, d\tau \tag{17}$$

Both method result in approximate performances with the fact that the second method turns out to be slightly more stable. Note that the computation of eq 17 is much more complex than eq 16. Since MATLAB has appropriate tools to compute 17, I will be using formal discretization formula during the simulations.

System Constraints

One fundamental assumption during the derivation of LTV model of the system was that the variable δ is small. We need to satisfy

$$\cos(\delta) \approx 1 \qquad\qquad \sin(\delta) \approx \delta \tag{18}$$

Hence, I define the limits of δ as in equation 19

$$-\frac{\pi}{10} \;<\; \delta \;<\; \frac{\pi}{10} \tag{19}$$

Let's check the assumption on the limits of δ.

$$\cos\left(\frac{\pi}{10}\right) = 0.9511 \qquad\qquad \sin\left(\frac{\pi}{10}\right) = 0.3090 \tag{20}$$

Input of the system is equal to derivative of the wheel angle δ. As I explain, the variable δ is restricted to a very narrow region. Hence, the input cannot have large values. Let's compute the maximum value of the input.

$$u_{max} = \frac{\delta_{max} - \delta_{min}}{Step\ Size} = \frac{\pi/10 - (-\pi/10)}{0.1} = 2\pi \;\; \frac{rad}{sec} \tag{21}$$

Hence, I do not need to implement a new restriction for input.

Cost Matrices

Refer to equation 21. The input of the system is restricted to a narrow region. Hence, the cost of the input is already low. I will not be focused on the cost matrix R. Rather, I will be focused on matrix Q.

The constraints related to states are already implemented. There remains the problem that the vehicle needs to be on the center of the road, i.e., $e_y = 0$. Hence, I define the output of the system as

$$y = e_y = C\bar{s} \tag{22}$$

where $C = [\,0\,0\,0\,1\,0\,]$. Then

$$Q = C^T C \tag{23}$$

Lets see the performance of the cost matrices for dual mode MPC design. The state vector will be nonzero initially. We will observe that the initial condition will be rejected. Examine the Figure 1.

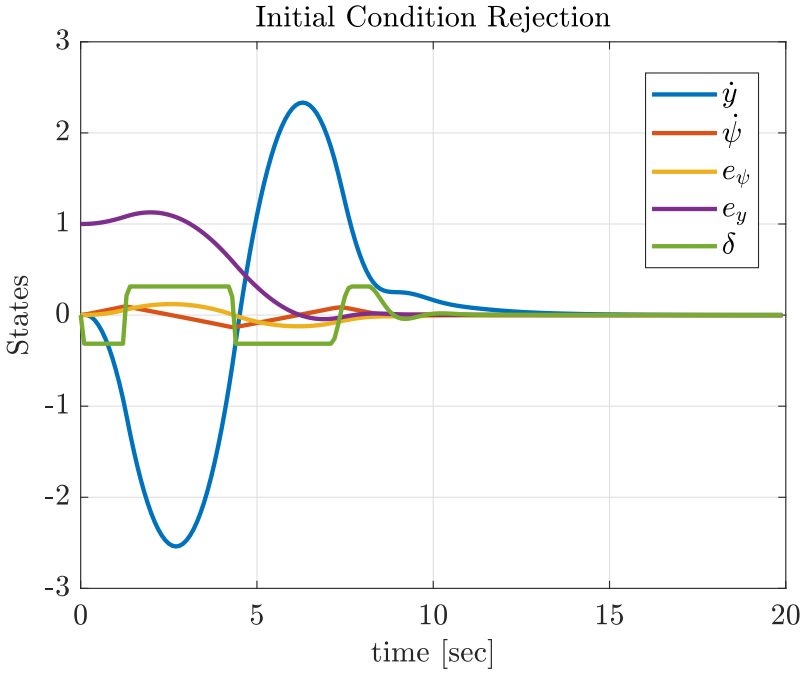

Figure 1: Initial Condition $= \bar{s}_0 = [0\,0\,0\,1\,0]^T$. Longitudinal Velocity $\dot{x} = 20\,m/sec$ and road radius is $500\,m$. Dual Mode MPC Method is executed. Offset term C_s is not added to the states.

Possible Disturbances

Until that point, I have assumed that the longitudinal velocity $\dot{x}$ and the road radius r is constant. But, during the operation, they can exhibit deviations.

The disturbances are modeled in MATLAB as follows

$$\dot{x}_{k+1} = \dot{x}_k + \dot{x}_{desired} \times 0.02 \times (0.5 - rand) \tag{24}$$

$$r_{k+1} = r_k + r_{estimated} \times 0.02 \times (0.5 - rand) \tag{25}$$

$\dot{x}_{desired}$ stands for the desired longitudinal velocity. $r_{estimated}$ is the estimated radius of the road. 'rand' is a built-in MATLAB function that generates a random variable between 0 and 1.

With the expressions 24 and 25, the velocity of the vehicle and the road radius change by time. Now, we have the time variance issue. Examine Figure 2. I focus on the state e_y. The oscillations are slightly increased. But the controller performance is acceptable.

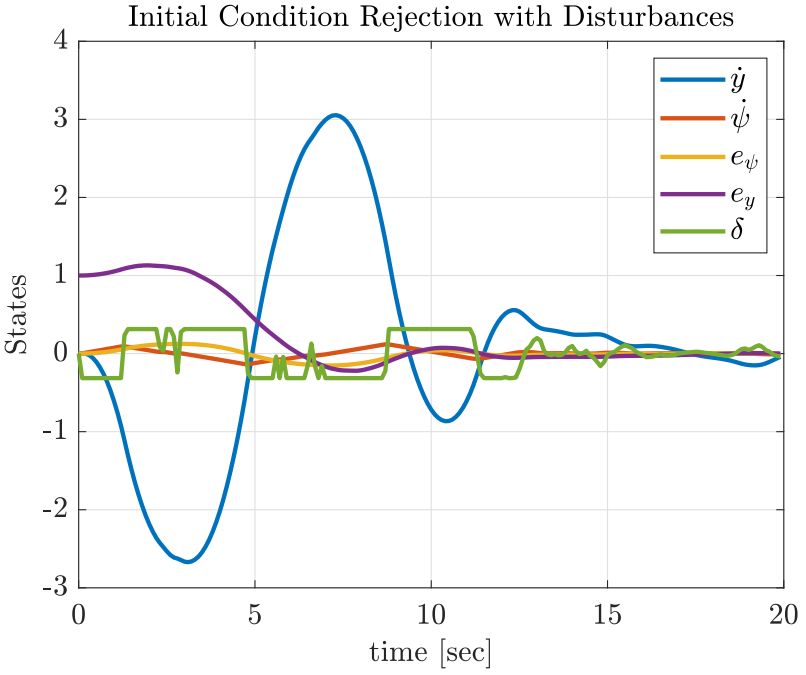

Figure 2: Initial Condition $= \bar{s}_0 = [00010]^T$. Desired longitudinal velocity $\dot{x}_{desired} = 20\ m/sec$ and the estimated road radius is $500\ m$. Dual Mode MPC Method is executed. Offset term C_s is not added to the states.

d - Obstacle Avoidance without Disturbances

Up to this point, I have shown that the controller is capable of keeping the vehicle on the lane, i.e., $\bar{s} = 0$ at steady-state. The controller performance is also observed with various disturbances. Now, I perform obstacle avoidance.

Assumption: The car needs to move 1 meters in lateral axis to avoid the obstacle.

Notice that: the car needs approximately 10 seconds to obtain $e_y = 0$ in Figure 2. That 10 seconds correspond to a movement of 1 meter in lateral axis.

Conclusion: The car needs notice the obstacle at least 10 seconds before the collision.

Remark: Sampling period is 0.1 seconds. 10 seconds with the sampling period 0.1 second correspond to 100 steps.

Conclusion: The horizon length of the MPC must be greater than 100.

In case we detect an obstacle, we need a restriction on the state e_y. Hence, depending on the position of the vehicle and the position of the obstacle, we need to update the restriction of the state e_y for the whole horizon. The results are provided in 3 and 4, for longitudinal velocities 20 meters per second and 10 meters per second respectively.

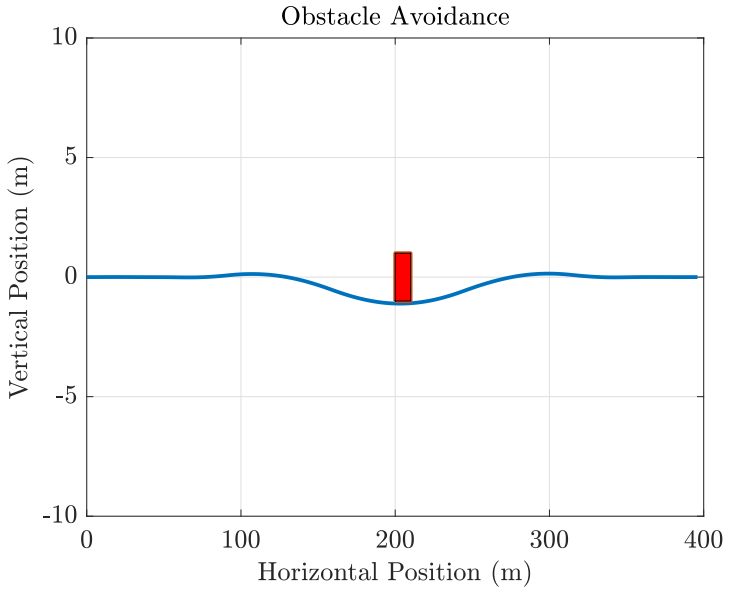

Figure 3: The red box represents the obstacle. Blue line is the trajectory of e_y. Longitudinal Velocity $\dot{x} = 20\, m/sec$ and road radius is $500\, m$. Dual Mode MPC Method is executed.

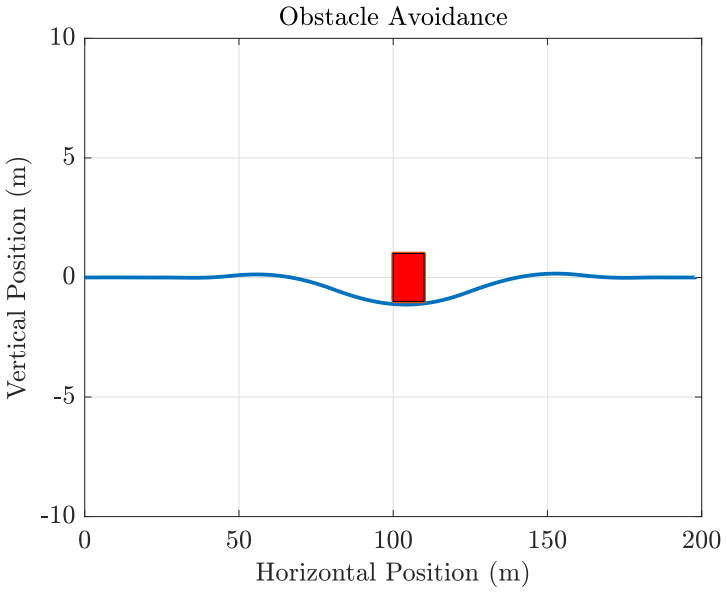

Figure 4: The red box represents the obstacle. Blue line is the trajectory of e_y. Longitudinal Velocity $\dot{x} = 10\, m/sec$ and road radius is $500\, m$. Dual Mode MPC Method is executed.

d - Obstacle Avoidance - Disturbances are Included

When I include the disturbances, I observe high oscillations. Hence, some modifications are executed.

A New Cost Matrix

It turns out that the cost matrix Q fails when disturbances are introduced in obstacle avoidance problem. By experimental observations, I update my Q matrix as follows:

$$Q = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{26}$$

Notice that only the term 0.5 is new.

Having this new modification, obstacle avoidance problem is repeated. Examine 5.

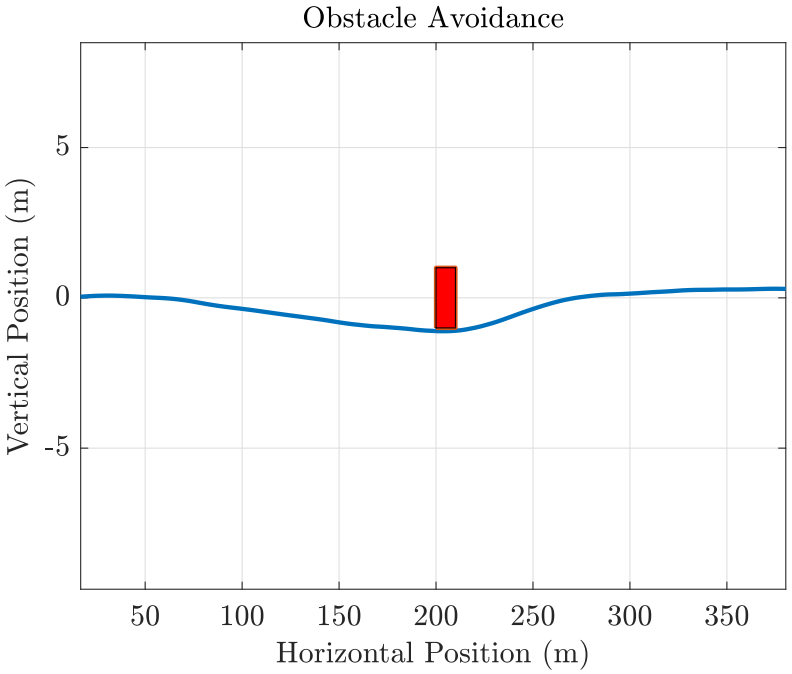

Figure 5: The red box represents the obstacle. Blue line is the trajectory of e_y. Longitudinal Velocity $\dot{x} = 20\, m/sec$ and road radius is $500\, m$. Dual Mode MPC Method is executed. Disturbances are included during the simulation.