Model predictive control of Cartpole model

Hyun In Lee *

Abstract

This report deals with the applying Model predictive control method to CartPole model.Model predictive control is a powerful technique for optimizing the performance of constrained systems. The performance of MPC was done with respect to (1)different initial state (2)different horizon length (3) implicit MPC or explicit MPC. The performance was quantified by feasibility of state,input / stability / and cost function.

1 Introduction

MPC is powerful tool to solve the control optimization problem. In this report, MPC was implemented to control the CartPole model. First ,finding the maximal control invariant set , maximal positive invariant set, and feasible state set was done to find feasible initial state. Then , the implementation procedure was divided into two method depending on which method will I use to solve this MPC optimization problem : implicit MPC method and explicit MPC method. Shortly , implicit MPC method is the online method to compute controller and states and explicit MPC is the MPQP method that compute the controller offline and use it to get the states. After I choose which method to use to solve the problem, 3 analysis was done : (1) divide the feasible X set into 225 points and set them into initial points and see the behaviour of its state trajectory and (2) vary the horizon length N with fixed initial states and see the feasibility and stability of the state trajectory and (3) calculate the computation time for (1) and (2) then compare the computation time with explicit and implicit case.

2 Cart pole problem

2.1 Implement MPC to cart pole model

To accommodate the understanding of MPC , I have solved the cart-pole problem via MPC. The cart pole is widely used example to test the performance of the various control method such as PID, LQR , MPC and even reinforcement learning.

*Department of Mechanical Engineering, University of Seoul National university. Email: gksdid5@snu.ac.kr

The reason that I have selected cart- pole as example of MPC is that first , various controller performance of cart pole example exists which provides good base or base criteria to compare the performance of the new controller(MPC) and second , still the core mechanism of the cart-pole system which is the under-actuated system that balance the object to unstable equilibrium point is widely used in the field.

2.2 System dynamic and constraints

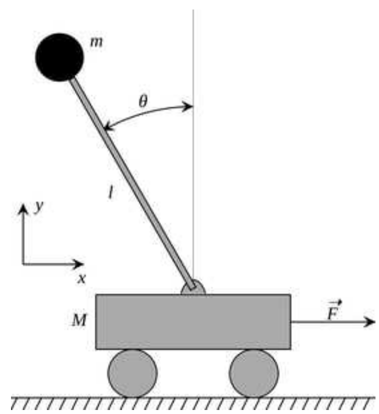


FIGURE 1: CartPole

The equation of motion is derived by Lagrangian method. Let the mass of cart, mass of pole, rod length as M, m, l . Then L = T - V could be written as

$$L = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{\theta}^2 l^2 + 2\cos\theta\dot{\theta}\dot{x}l + \dot{x}^2) - mgl\cos\theta \tag{1}$$

The Lagrangian dynamic could be solved by below method.The equation of motion is obtained with respect to θ and

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = u \qquad \Rightarrow \qquad ml^2\ddot{\theta} + ml\cos\theta\ddot{x} - mgl\sin\theta = 0 \tag{2}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0 \qquad \Rightarrow \qquad ml\cos\theta\ddot{\theta} + (M+m)\ddot{x} - ml\ddot{\theta}^2\sin\theta = u \tag{3}$$

Then use the state-space representation with state-space variables $x_1 = \theta, x_2 = x, x_3 = \dot{\theta}, x_4 = \dot{x}$ and the equation could be linearized around $\theta = 0$. Then the final representation of the linearized dynamic equation is given as

$$\begin{bmatrix} \dot{\theta} \\ \dot{x} \\ \ddot{\theta} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{mg+Mg}{Ml} & 0 & 0 & 0 \\ -\frac{mg}{M} & 0 & & 0 \end{bmatrix} \begin{bmatrix} \theta \\ x \\ \dot{\theta} \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{Ml} \\ \frac{1}{M} \end{bmatrix} u \tag{4}$$

Above equation could be formulated as

$$\dot{z} = Az + Bu$$

If we update the equation every dt time step, then discrete time system dynamic could be expressed as

$$z(k+1) - z(k) = dt(Az(k) + Bu(k))$$

$$z(k+1) = (Adt + I_4)z(k) + (Bdt)u(k))$$

In this problem , I have set the variables as $M = 1, m = 1, l = 1, g = 9.8$ and $dt = 0.01$, then finally the system dynamic could be formulated as

$$z(k+1) = \begin{bmatrix} 1 & 0 & 0.01 & 0 \\ 0 & 1 & 0 & 0.01 \\ 0.196 & 0 & 1 & 0 \\ -0.098 & 0 & 0 & 0 \end{bmatrix} z(k) + \begin{bmatrix} 0 \\ 0 \\ -0.01 \\ 0.01 \end{bmatrix} u \tag{5}$$

where eigenvalues of matrix A are 1.0,1.0,1.0443,0.9557 which denotes this system is unstable due to existence of eigenvalues that exceed 1.0

The input and state constraint set X, U is

$$\begin{bmatrix} -\pi/2 \\ -5 \\ -10 \\ -10 \end{bmatrix} <= \begin{bmatrix} \theta \\ x \\ \dot{\theta} \\ \dot{x} \end{bmatrix} <= \begin{bmatrix} \pi/2 \\ 5 \\ 10 \\ 10 \end{bmatrix} \tag{6}$$

and

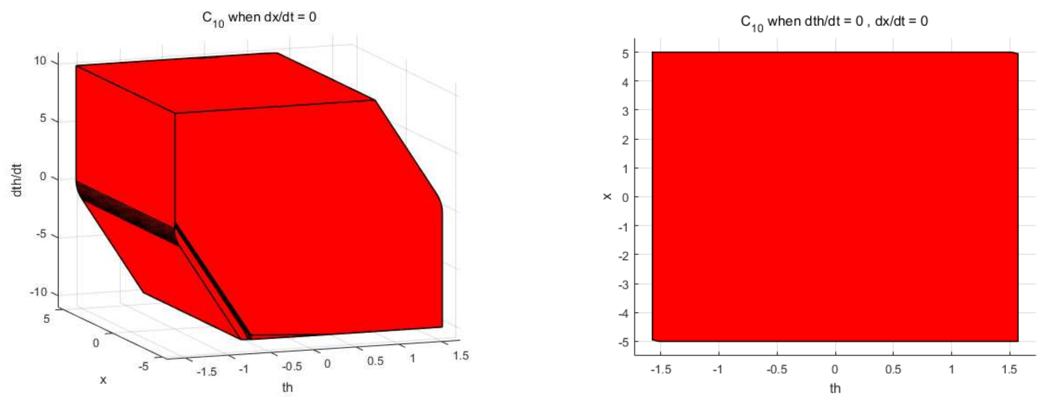
$$-50 <= u <= 50 \tag{7}$$

2.3 Terminal set and Terminal constraints

The terminal set should be selected carefully because it determines the feasibility of solution. 2 possible set is possible for terminal set, the maximal control invariant set C_∞ and maximal closed loop invariant set O_∞ The algorithm to get C_∞ and O_∞ is recursive method starting from X. Since the set is defined in R^4, the set is shown as figure2 and figure3 in lower dimension when $\dot{x} = 0$ and when $\dot{x} = 0, \dot{\theta} = 0$. The result is plotted with $(x, \theta, \dot{x}, \dot{\theta} = 0)$ and $(x, \theta, \dot{x} = 0, \dot{\theta} = 0)$.

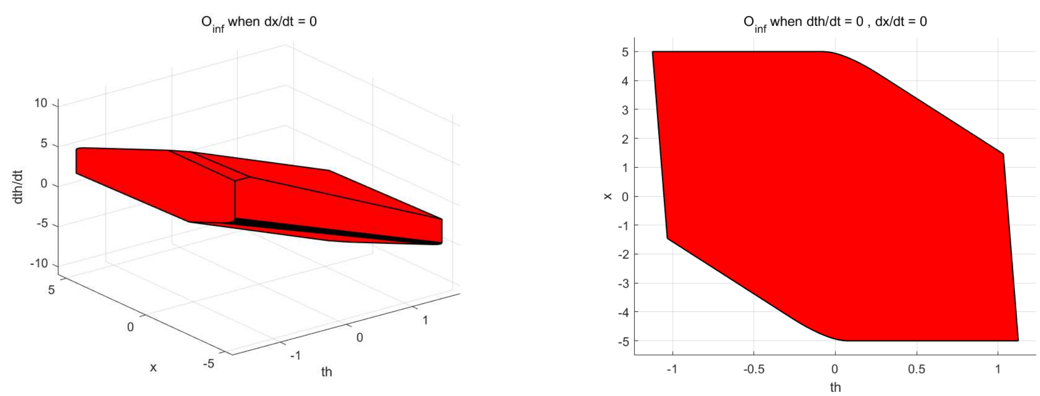
Unfortunately, the computation time to get maximal control invariant set is too long,so insteadC_{10} which means 10 recursive step control invariant set was estimated.Also, the weight matrix is chosen as the solution of Algebraic Riccati Equation. As a result , in figure2, C_{10} set does not show critical area reduction with compare with default feasible x set X . Only the left bottom or right upper vertex area get reduction. However,in figure3, the O_∞ set shows significant reduction of area compared to X

Therefore , setting the terminal set as maximal positive invariant set O_∞ and terminal cost as P_∞ guarantees the feasibility and stability, respectively.



(A) $C_{10}, (x, \theta, \dot{x}, \dot{\theta} = 0)$ (B) $C_{10}, (x, \theta, \dot{x} = 0, \dot{\theta} = 0)$

FIGURE 2: C_{10} set



(A) $O_\infty, (x, \theta, \dot{x}, \dot{\theta} = 0)$ (B) $O_\infty, (x, \theta, \dot{x} = 0, \dot{\theta} = 0)$

FIGURE 3: O_∞ set

2.4 Implicit MPC solution

Implicit MPC is optimal solution obtained by iterative numerical procedure. two observations will be done on this section . First , implicit MPC solution with respect to different initial state with same MPC horizon length and Second, implicit MPC solution with different MPC horizon length with same initial state.

For both case , the cost function is designed as

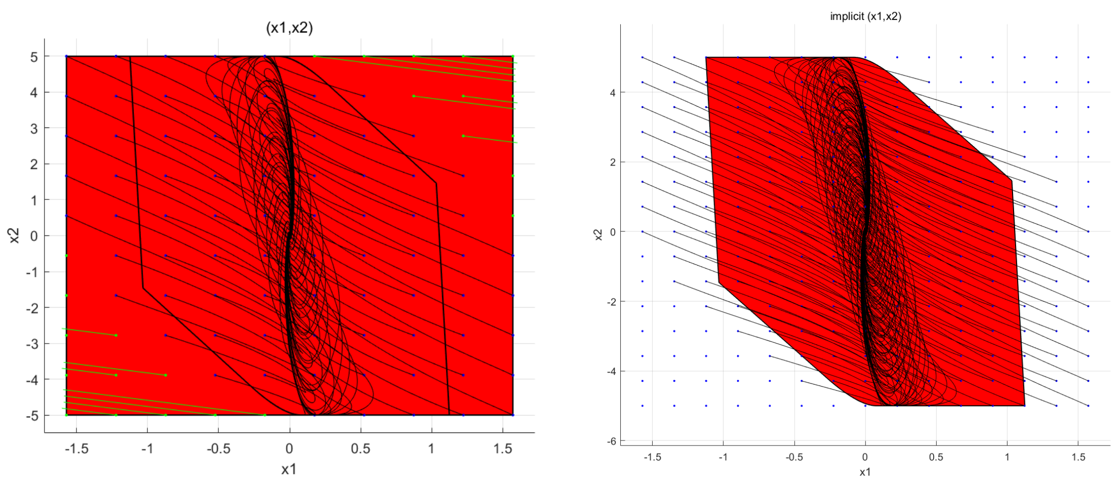
$$J(X) = x_N^T P x_N + \sum_{t=0}^{N-1} x_k^T Q x_k + u_k^T R u_k$$

where N and coefficient matrix are setted as

$$N = 50, Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 2, P = P_\infty$$

2.4.1 Solution with different initial states

The MPC problem with different initial state was done with 100 initial states(or 225 initial states) when $\dot{x} = 0, \dot{\theta} = 0$ and N=50 . Those initial states were equally distributed in 100 points(or 225 points) within feasible x set$X$(the outer contour of figure4). In figure4(a) , the outer rectangular contour describes the feasible x set X and inner hexagon contour is the maximal positive invariant set O_∞. The green colored line is infeasible initial point and black colored line is the feasible initial points.The figure4(b) is just same analysis what I have done on figure4(a) but divide the feasible x set X into 225 points and have done same experiment.



(A) feasibility of 100 points in X

(B) feasibility of 225 points in X

FIGURE 4: feasibility of initial points in X

From above results , we could observe all initial state points inside the O_∞ goes to origin as time goes by and points inside the C_∞ also converges to the origin. But the initial points lies between between feasible x set X and C_∞ diverges. The green points denotes the points that diverge and blue points denotes convergence points.

2.4.2 Solution with different horizon length

Now , fix the initial point as [-1.5 , 2, 0 , 0] and do experiment when N = 3, 5, 7, 10, 20, 30, 50, 70, 100, 200].

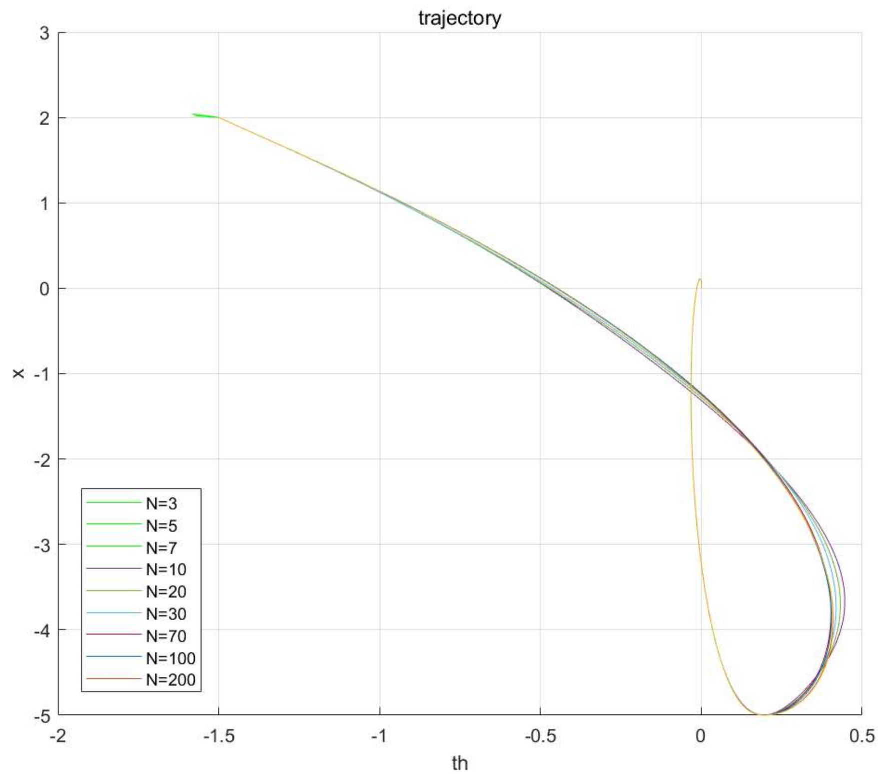


FIGURE 5: closed loop trajectory of different horizen length

From above result when N=3,5,7 the trajectory diverges but N = 10,20,30,70,100,200 the trajectory converges to the origin. Also there are some trajectory difference among N=10,20,30,70,100,200. The computation time for each horizon length with initial point[-1.5 , 2, 0 , 0] was done. The result is shown in table1.

TABLE 1: implicit solution computation time

Horizen length N	**3**	**5**	**7**	**10**	**20**	**30**	**50**	**70**	**100**	**200**
Computation time[sec]	2.09	0.60	0.43	24.14	33.37	32.19	41.65	50.74	65.30	124.3740

2.4.3 Computation time of 2.4.1 and 2.4.2

The computation time of implicit solution could be divided into two types of computation time : (1) time for solving optimizer function which computes the MPC controller and (2) time for getting all states x and u until x, u converges to the origin which get $x_0, u_0, x_1, u_1, ..., x_{N-1}, u_{N-1}, x_N$. The MatLab function : clock() was used to compute the computation time. The figure7 shows the computation time to compute controller and get all states and inputs that have done for 100 different initial points.(The closed trajectory loop is shown in figure4(a)).

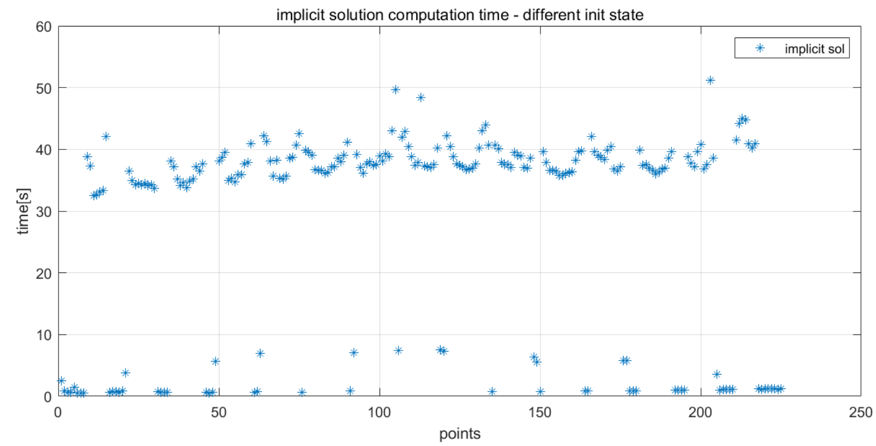


FIGURE 6: computation time when different initial state(N=50)

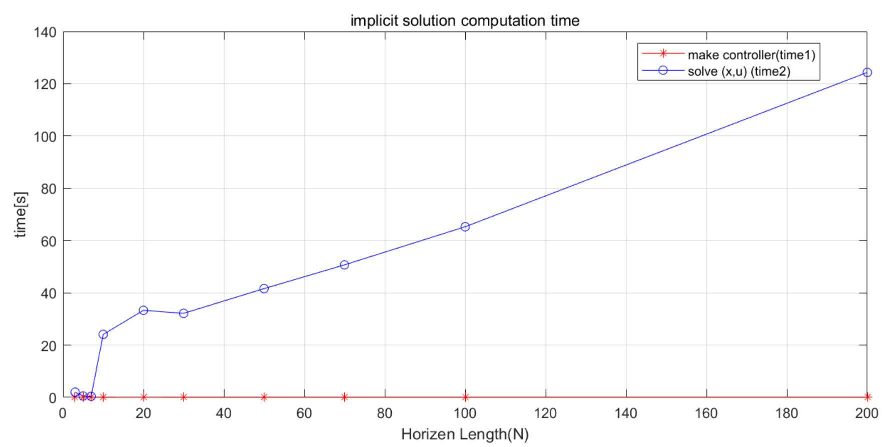


FIGURE 7: computation time when different horizon length($x0 = [-1.5, 2, 0, 0]$)

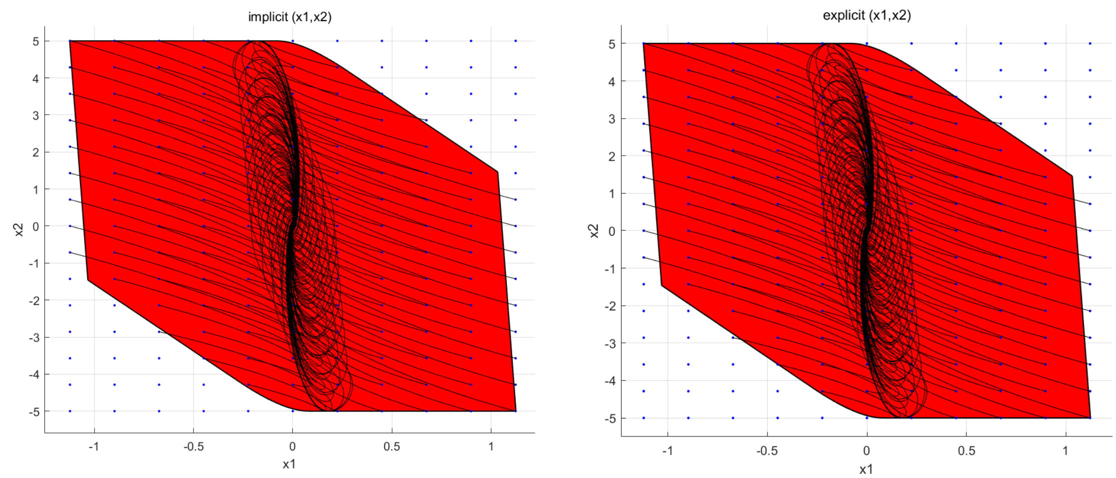
From figure6, if points are feasible solution, then the computation time is about 30 40 seconds. That time is almost identical for all points. Also figure7 shows if N increases, then time to get states and inputs $(x_0, u_0, x_1, u_1, ..., x_{N-1}, u_{N-1}, x_N)$ is increasing linearly.

2.5 Explicit MPC solution

Explicit MPC solution is method that evaluating the explicit representation of the MPC feedback law, which is obtained off-line using parametric programming (referred to as explicit MPC) The same procedure was done what I have done on section 2.4.1 and 2.4.2 . After that, the computation time was calculated with same initial point [-1.5 , 2, 0 , 0] . The computation time is sum ot two type of time that (1) solve mpc problem to explicit form and (2) calculate all states x and u for simulation time .

2.5.1 solution with different initial states

Unfortunately, due to the high computation time of explicit MPC , the explicit MPC solution for N=3 is only computed by personal notebook. when N=3, the implicit solution of 225 different initial points and explicit solution of 225 different points was compared. The implicit,explicit MPC result which is the close loop trajectory when N=3 is shown in figure8.



(A) implicit MPC when N=3

(B) explicit MPC when N=3

FIGURE 8: feasibility of initial points in X

From figure8 , I could check that implicit solution and explicit solution have same feasible points.So I could conclude that implicit or explicit MPC solution does not influence the feasibility of the initial points. Also the computation time for implicit solution and explicit solution when N=3 is given as figure9,10,11

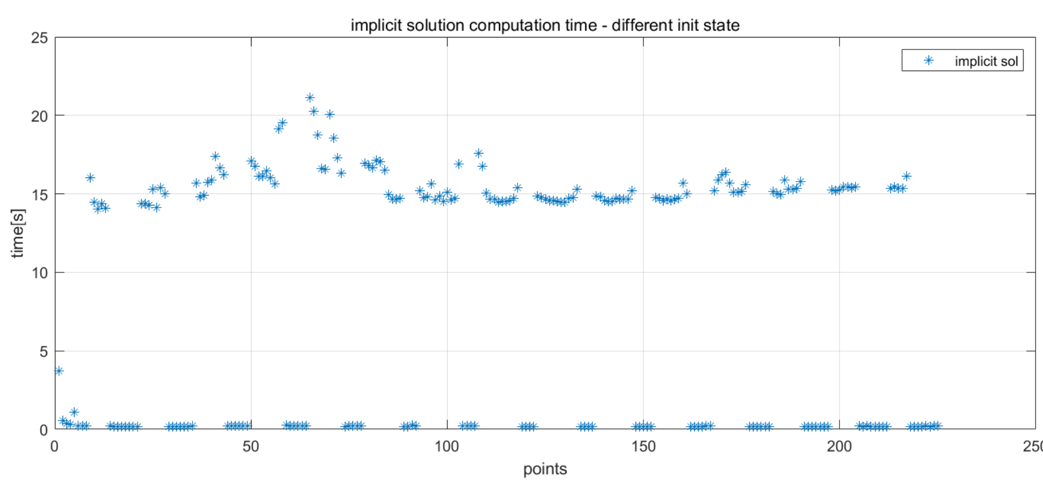


FIGURE 9: implicit solution computation time when N=3

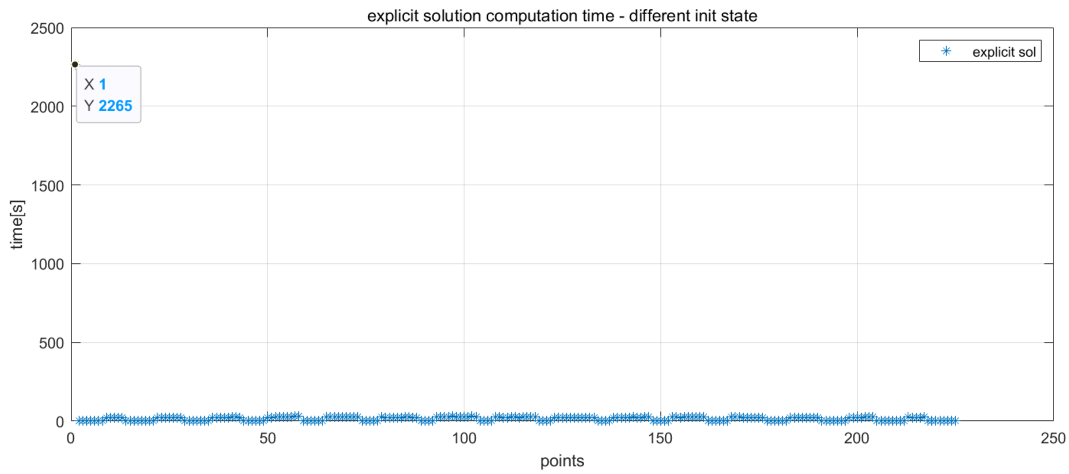


FIGURE 10: explicit solution computation time when N=3

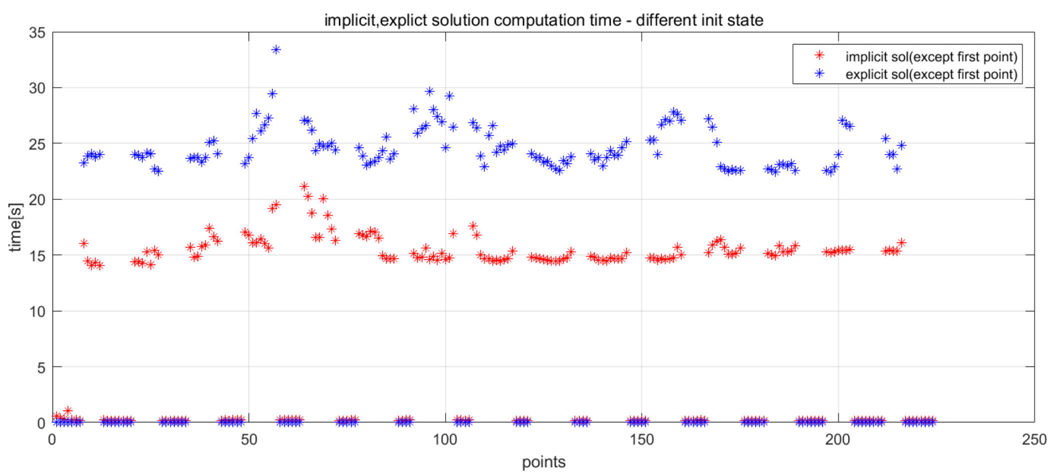


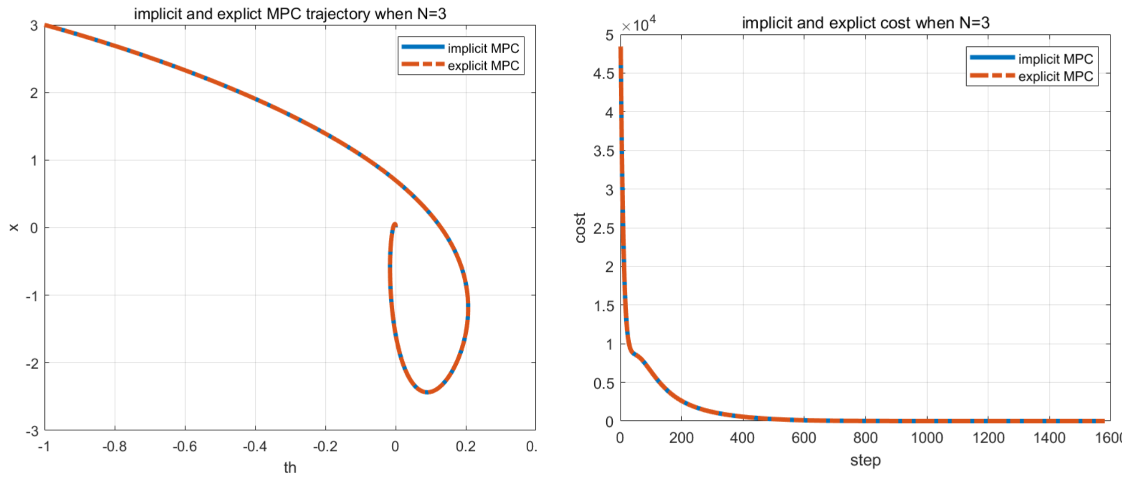
FIGURE 11: implicit explicit solution(except first point) computation time when N=3

Figure9, figure10 respectively shows the computation time of implicit,explicit solution.Figure11 shows the comparison result of figure9 and figure 10.From figure9, we could check that average time for implicit MPC took about 15 20 seconds. From figure10, The first point includes time for computing controller and it took about 2250 seconds, and after computing the explicit MPC controller , average time for getting (x,u) took about 20 30 seconds.

For small N , I have expect that explicit solution is much faster than the implicit solution but it's not observed when N=3. I think this is because , explicit MPC has about 4682 critical regions so it took additional time for searching the critical region that the initial point x_0 is inside and get the corresponding function output.

2.5.2 Solution with different horizon length

Also only when horizon length N=3 is compared in the case of implicit solution and explicit solution. Figure12 shows the trajectory or implicit MPC, explicit MPC and cost value of implicit MPC, explicit MPC.



(A) trajectory of implicit/explicit MPC

(B) explicit MPC when N=3

FIGURE 12: cost of implicit/explicit MPC

From figure12 , I could check that explicit MPC and implicit MPC shows same trajectory and same cost value per stage. So I could also conclude that implicit or explicit MPC solution does not influence the stability of the solution.

3 Conclusion

By solving above examples, there are some conclusions that I have make. . .

1. Set terminal set as maximal positive invariant set (O_∞) could guarantees the recursive feasibility.(Also, maximal control invariant set C_∞ is available and guarantees the recursive feasibility)
2. weight matrix as P_∞ which is the solution of algebraic Ricatti equation guarantees the stability of the solution.
3. solve identical MPC with different horizon length yields slightly different trajectory of the solution and this behavior could be observed easily when set the initial state in the maximal control invariant set but the point is lies on very close to the border between infeasible X set and the maximal control set
4. The implicit and explicit yields same answer which means both method yields same stability , recursive feasibility and performance.

5. The explicit form is useful when sample time is short or online computing power is low but implicit form is useful when sample time is long