Non-overshooting Model Predictive Control (MPC) Design for Vehicle Lateral Stability

by

Monish Dev Sudhakhar

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved January 2023 by the
Graduate Supervisory Committee:

Yan Chen, Chair
Yi Ren
Zhe Xu

ARIZONA STATE UNIVERSITY

May 2023

ABSTRACT

Advanced driving assistance systems (ADAS) are one of the latest automotive technologies for improving vehicle safety. An efficient method to ensure vehicle safety is to limit vehicle states always within a predefined stability region. Hence, this thesis aims at designing a model predictive control (MPC) with non-overshooting constraints that always confine vehicle states in a predefined lateral stability region. To consider the feasibility and stability of MPC, terminal cost and constraints are investigated to guarantee the stability and recursive feasibility of the proposed non-overshooting MPC. The proposed non-overshooting MPC is first verified by using numerical examples of linear and nonlinear systems. Finally, the non-overshooting MPC is applied to guarantee vehicle lateral stability based on a nonlinear vehicle model for a cornering maneuver. The simulation results are presented and discussed through co-simulation of CarSim® and MATLAB/Simulink.

ACKNOWLEDGEMENTS

I am very thankful to my advisor, Dr. Yan Chen, whose guidance was constructive in planning my career path while completing my thesis. I had the opportunity to hone my control theory knowledge and get practical exposure by being able to work on multiple projects within Dynamic Systems & Control Laboratory.

I want to thank the committee members, Dr. Yi Ren & Dr. Zhe Xu, for their time and effort in reviewing my thesis. I would like to thank my parents and my brother for their support all these years. I will carry this positive energy all through my life.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1 INTRODUCTION

1.1 Background and Motivation

The growth of autonomous guided vehicles in driving scenarios is gaining much importance and has obtained rapid growth with the help of artificial intelligence and enhanced sensor technologies. Complete autonomous driving in natural urban settings has remained an important but elusive goal [2]. Beyond that appropriate for a research lab, significant engineering effort must go into a system to ensure maximal reliability and safety in all conditions [2]. Vehicle safety has changed drastically over the years; today, newer cars are safer than ever [1]. As many ADAS functions focus on longitudinal motion, minimal research is done considering the lateral motion as in Figure 1

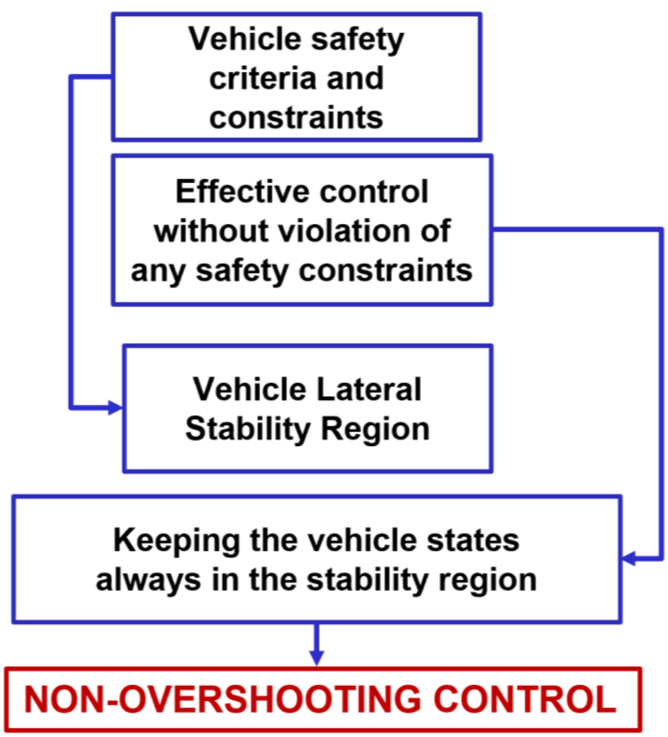

Figure 1 Non-overshooting MPC With Applications to Vehicle Stability Control. [19]

The practical need for non-overshooting phenomenon is needed when the system must follow a safety constraint in some emergencies like the docking of a ship/boat where the constraints play a significant role, and there is a need to make sure that the system satisfies it at all instant (feasible). The vehicle's lateral dynamics will dominate and may cause safety issues in a typical driving scenario [3]. Critical maneuvers like a lane change, double lane change, cornering, and obstacle avoidance include a lateral motion. A significant number of safety-related accidents happened in the last few years, primarily due to lateral vehicle movement. Therefore, vehicle lateral stability control should also be thoroughly considered in developing AGVs [3]. With that in mind, the primary method to control vehicle stability is based on a predefined or estimated stability region, as in Figure 1, typically described by lateral vehicle states (such as yaw rate and lateral speed) on a phase plane [3]. The effect of model predictive control (MPC) on this issue is of great importance due to the controller's ability to control the vehicle states with constraints. The presence of the constraint makes the vehicle states be in the region and not overshoot a particular value. In model predictive control (MPC), inequality constraints for non-overshooting design could be applied at each sampling time during the entire prediction horizon to ensure that the control inputs will not make the system output overshoot. If the boundaries are considered references, a non-overshooting control design concerning the references can satisfy the vehicle stability requirements [2].

MPC's underlying issues regarding stability and feasibility should be addressed. This thesis addresses those problems by adding specific parameters in the last step of the horizon. The stability and feasibility of the model predictive control (MPC) are ensured ideally with the help of terminal cost and terminal constraint [5]. The long-term goal of this project is to propose a detailed study on a control algorithm using the non-overshooting constraint and terminal cost/constraints such that we could use it in an autonomous vehicle to improve driving safety. The control purpose is to keep vehicle states operating within the stability region. To achieve this purpose, whenever the vehicle states approach or pass through the boundaries of the stability region, the controller will maintain or drag vehicle states back into the region [2][3]. Hence a suitable terminal cost under a specific terminal region is developed and represented as an approximation to the infinite horizon problem.

1.2 Literature Review

Theoretically, the non-overshooting MPC has been implemented in two categories generally, linear systems & nonlinear systems. Although the non-overshooting control design for the specific systems has been discussed, the general dynamic system has yet to be addressed by many. The main idea was derived from the works of Yiwen Huang and Yan Chen [3]. In this project, through the advantages of handling constraints, MPC is utilized as an appropriate approach to develop a uniform non-overshooting control design for general dynamic systems. The flow of the study concerning the numerical simulations with the linear/nonlinear systems & stability region approach for a non-overshooting model predictive control (MPC) in this research is based on the work from Dr. Yan Chen [3]. The estimation of this region is based on his other paper [4], where he discusses vehicle lateral stability regions by a local linearization method, which guarantees both vehicle's local stability and handling stability. As per the literature, the terminal cost function brings stability to the systems, and the terminal equality constraint brings feasibility to the system. The definition of terminal constraint was taken from lecture notes on MPC by UC-Berkeley, Stanford, and Ohio State Universities [13] [14] [15]. D.Q Mayne and D. Limon predominantly did the prior work on this. Hence, their paper [12] on the subjective study of model predictive control is beneficial in understanding the flow. The theoretical understanding and the baseline for the terminal cost and terminal constraint functions is based on F. Borrelli's [6] work in his book. The stability proof and the definition of invariant set theory were very relevant to this thesis. A model predictive control variant employs a terminal cost $F(x)$ and a terminal constraint in the optimal control problem PN(x). It is the version attracting the most attention in the current research literature. It has superior performance compared to zero states and terminal constraint set MPC and can handle a much more comprehensive range of problems than terminal cost MPC (D.Q.Mayne) [13]. Finally, the choice of terminal cost from the nonlinear systems is still fuzzy, and there is ongoing research for the best cost function. Authors like David Scramuzza from ETH – Zurich addressed this in his paper by considering an essential assumption regarding the terminal penalty matrix addressed in this thesis. The terminologies have been used as per the SAE definitions [18]

1.3 The Problem Statement

As discussed, non-overshooting model predictive control is critical in emergencies [2], where the system must obey a specific constraint indefinitely. This research accelerates the development and popularization of the stability region based MPC. However, as mentioned above, the MPC has stability and feasibility issues [12]. The system seems stable for specific initial points and general conditions but only for some scenarios. It is one of the most significant disadvantages of MPC. We need to ensure that stability and feasibility are guaranteed so that we can have a robust MPC that is suitable for practical applications in an efficient way. Before implementing the model with the vehicle controller, we need to understand the system's response in general. The problem formulation is dealt with and carried out in such a way that it addresses the following questions,

1. How to analyze the systems using non-overshooting constraints and terminal cost/constraint utilizing the approach of MPC algorithm for both linear & nonlinear systems?
2. How can it be extended to vehicle lateral stability control, guaranteeing that vehicle states are constantly within the stability region?

1.4 Outline

The remainder of the paper is as follows. Chapter 2 discusses the introduction to model predictive control (MPC) and its theory. The influence of terminal cost and terminal constraint on the MPC is discussed elaborately with the support of elaborate proofs accordingly in chapter 3. Chapter 4 presents the non-overshooting model predictive control (MPC) idea with the simulation results. Chapter 5 describes the theory behind the non-linear vehicle model and the CarSim® environment. Chapter 6 concludes the thesis presented and discusses the future work that could take this research forward.

CHAPTER 2: MODEL PREDICTIVE CONTROL

2.1 Background of Model Predictive Control (MPC)

Model Predictive Control was initially developed for chemical applications to control the transients of dynamic systems with hundreds of inputs and outputs, subject to constraints. It actively encompasses optimal control strategy in a receding horizon approach. We know that optimization is the backbone of the optimal control process because of the need to find the minimum/maximum of the objective function.

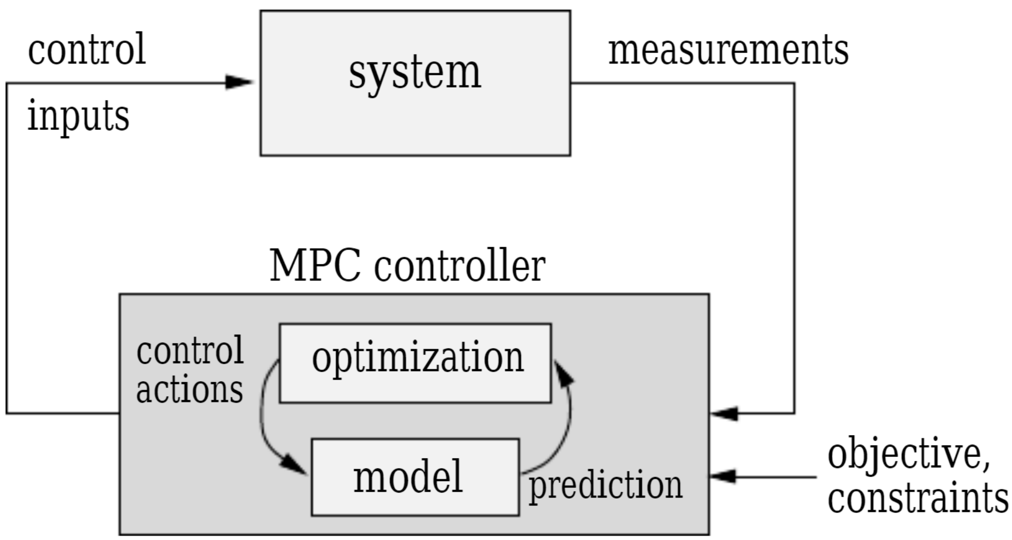

Figure 2 Layout of Model Predictive Control [24]

The further understanding for the need of optimization in optimal control can be understood by looking at the following problem formulations. [19]

The optimization problem for minimizing the function f(x) is,

$$\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) \tag{1}$$

subject to $\mathrm{c}(x) \leq 0, \mathrm{ceq}(x) = 0, \mathrm{A} * x \leq \mathrm{b}, Aeq * x = beq, \mathrm{lb} \leq x \ \leq \mathrm{ub}$

Similarly, the optimal control problem is formulated as ,

$$\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) \tag{2}$$

subject to $\mathrm{c}(x) \leq 0, \mathrm{ceq}(x) = 0, \mathrm{A} * x \leq \mathrm{b}, \mathrm{Aeq} * x \leq \mathrm{b}, \mathrm{lb} \leq x \leq \mathrm{ub}$

$$\dot{x} = \mathrm{h}(x)$$

where c(x), ceq(x), A, b, Aeq, Beq are the equality and inequality coefficients.

From (1) & (2), we come to know that the optimal control finds the future values in addition to solving an minimization problem. This is utilized in the model predictive control actively.

There are many methods for solving an optimization and optimal control problem. Optimization pre-dominantly uses:

1. Analytical methods (E.g.: KKT conditions)
2. Iterative methods(E.g.: SQP)

Where the fmincon is the non-linear programming solver [20] as the addition of non-linear constraints makes the problem non-linear and complex. It is very useful in simulation with tools like MATLAB and this thesis has the active usage of fmincon. For the optimization problem in (1), the syntax of the fmincon problem would be, [20]

$$x = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, nonlcon)$$

For the optimal control problem, most used methods are. [19]

1. Dynamic programming
2. Receding horizon Control (MPC)

We can understand the relation of optimal control with the model predictive control from this approach. Model predictive control is one of the solutions to optimal control problems, and it can be done in stages explained in the next section. Hence the applications of model predictive control (MPC) have increased by multiple folds due to the ability of it to satisfy the constraints and the explicit usage of model and tuning parameters. An optimal control strategy is an open loop control, and it cannot account for the system-model mismatch between the predicted and actual behavior.

2.1 What is model predictive control?

MPC is a form of control in which the control action is obtained by solving online an infinite horizon optimal control problem at each sampling instant in which the initial state is the current state of the plant. Optimization yields an infinite control sequence, and the first control action in this sequence is applied to the plant, as in Figure 3. MPC differs, therefore, from conventional control in which the control law is precomputed offline. However, this is not an essential difference; MPC implicitly implements a control law that can, in principle, be computed offline.

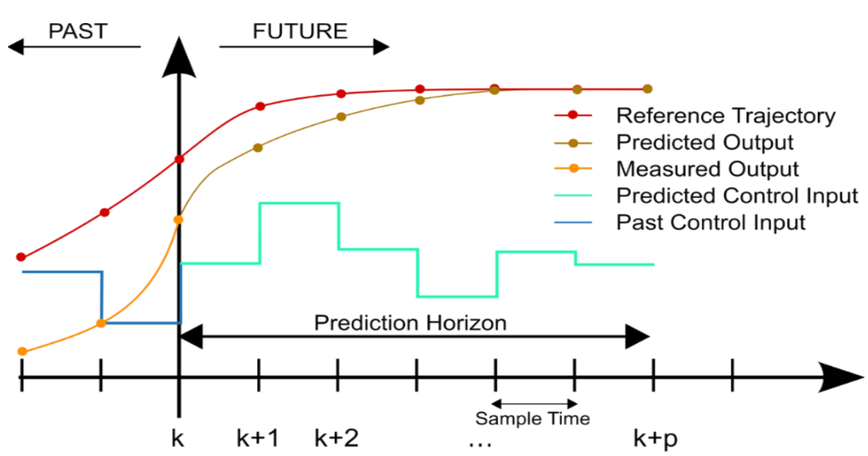

Figure 3 Working Principle of RHC[20]

The design of model predictive control needs the understanding of the following components:

1. Plant model
2. Cost Function
3. Prediction Horizon
4. Control Horizon

The plant model represents the system's dynamics, which can be linear or nonlinear. It varies according to the application. The cost function is the function of state and control variables that can be linear or quadratic. Generally, it is considered quadratic to prevent the error from going to negative values while solving the problem. The prediction horizon, generally denoted by N, defines the time window in which the cost function is minimized, and the states are predicted according to this value. The control horizon denoted by P tells us the number of times the control variable (Moving Variable) is solved for a given problem.

It can be from one to the value of N but can be, at most, the prediction horizon as the system cannot afford to have more control effort.

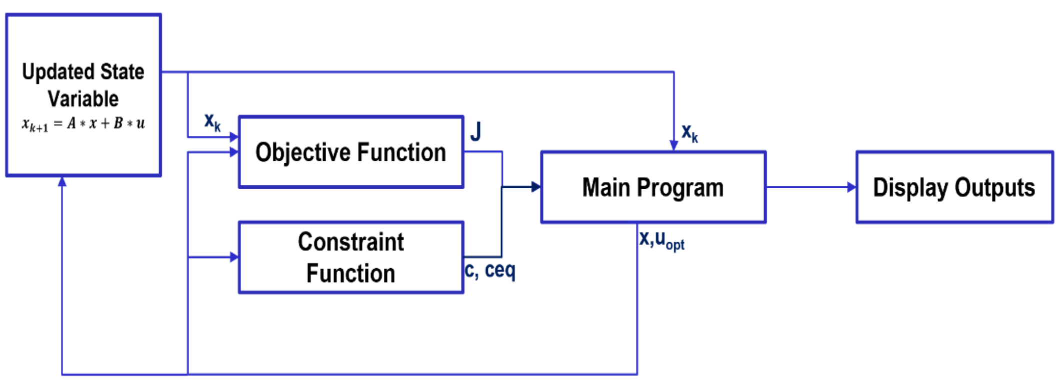

Figure 4 Flowchart of the MPC Programming

Primarily, suppose the current state of the system is controlled. In that case, model predictive control (MPC) obtains an optimal control input by solving an open-loop problem for this initial state, and then a specific control action is applied to the plant, as in Figure 4. Then the state variable is updated and sent to the objective function again, where an optimization problem is solved. The computed optimal manipulated input signal is applied to the process only during the following sampling interval. At the next step, a new optimal control problem based on new state measurements is solved over a shifted horizon. An infinite horizon suboptimal controller can be designed by repeatedly solving finite time optimal control problems in a receding horizon way. At each sampling time, starting at the current state, an open-loop optimal control problem is solved over a finite horizon in Figure 3. The resulting controller is called a Receding Horizon Controller (RHC). A receding horizon controller where the finite time optimal control law is computed by solving an optimization problem online is usually referred to as Model Predictive Control (MPC). [6].

2.2 Limitations of Model Predictive Control

In MPC, when we solve the optimization problem over a finite horizon repeatedly at each time step, we hope that the controller resulting from this "shortsighted" strategy will lead the example in the forthcoming sections indicates that at least two problems may occur.

First, the controller may lead us into a situation where after a few steps, the finite horizon optimal control problem that we need to solve at each time step is infeasible, i.e., there does not exist a sequence of control inputs for which the constraints are obeyed. Second, even if the feasibility problem does not occur, the generated control inputs may not lead to trajectories that converge to the origin, i.e., that the closed-loop system is asymptotically stable. (F. Borrelli) [6].

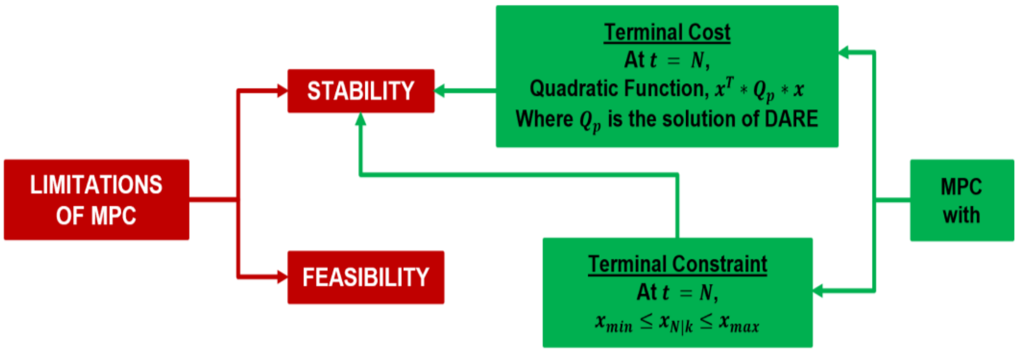

Figure 5 Common Issues in MPC

It is a well-known fact that there are differences in the closed-loop trajectories between the actual and predicted system response in an MPC. The optimal control problem's design is being solved instantly, and we must ensure stability and feasibility are guaranteed at every step. The inclusion of terminal cost and constraints plays a significant role in this, as in Figure 5. Those effects are studied with the help of the conclusion drawn from the work of notable authors. Therefore, conditions are derived on how the terminal cost with the penalty matrix and the terminal constraint set should be chosen to ensure closed-loop stability and feasibility.

CHAPTER 3 : THEORETICAL STUDY

The following table gives an overview of the theorems and lemma considered for the numerical simulations in the next section.

Theorem 1	Consider a linear discrete system, $$x_{k+1} = A * x_k + B * u_k$$ If we were to solve for the infinite horizon problem as below, i.e., $k \to \infty$ $$J_\infty^*(x(k)) = \min_{u_0,u_1,\dots} \sum_{k=N}^{\infty} x_k' Q x_k + u_k' R u_k$$ The choice of terminal penalty matrix, P in the terminal cost and it requires no control effort. The above problem approximates as $x_N^T P x_N$, where P should be a solution of ARE. It can be understood in two parts , one being from 0 to N and the other from N to ∞. $$J_k(x_k) = \min_{u_0,\dots u_{n-1}} \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + x_N^T P x_N$$	Finite horizon problem to be equivalent to an infinite horizon problem.
Theorem 2	Necessary and sufficient conditions for the predictions generated by the tail u_{k+1} to be	Recursive Feasibility

	feasible at time $k+1$ whenever the MPC optimization at time k is feasible are, 1. The constraints are instantaneously satisfied at all points in X_f. 2. X_f is invariant in mode 2 , which is equivalent to requiring that $(A+BK)x_{N\|k} \in X_f$ for all $x_{N\|k} \in X_f$	
Theorem 3	If X_f is control invariant, then the individual states, $X_{N-1}, X_{N-2}, \ldots, X_1$ € X_f are control invariant too.	Subset of an invariant set is also invariant of the control input.
Lemma 1	Effect of Positive invariant set on persistent feasibility	Positive invariant definition
Lemma 2	Effect of control invariant set on persistent feasibility	Control invariant definition
Theorem 4	The simplest approach is to design a terminal region such that , $x_n \in X_f = 0$ so that we can have a feasible control input at all instances.	Feasibility proof for the system with terminal equality constraints
Theorem 5	If a function is of the form,	

	$V(0) = 0\ and\ V(x) > 0, \forall x \in \Omega \setminus \{0\}\ V(x_{k+1}) - V(x_k) \leq -\alpha(x_k) \forall x_k \in \Omega \setminus \{0\}$ where $\alpha: R^n \to R$ is a continuous positive definite function. Then $x = 0$ is asymptotically stable in Ω	Lyapunov Function definition
Theorem 6	If a system is assumed to satisfy all the lemma and theorem discussed above. With the assumption(A3) as below, $min_{v \in U,}^{Ax+Bv \in X_f} (-p(x) + q(x, v) + p(Ax + Bv)) \leq 0, \forall x \in X_f.$ Then, the origin of the system is asymptotically stable with domain of attraction X_0	Shows the stability in the form of convergence criteria which ensures stability of the system.

Table 1 Overview of Theorems & Lemmas [6]

3.1 Infinite Horizon Problem

The shortsighted or the finite horizon brings certain disadvantages to the model predictive control as there is always some model mismatch. Hence, transforming a finite horizon problem into an infinite one is critical. If we solve the RHC problem for N = 1 (as done for LQR), then the open-loop trajectories are the same as the closed-loop trajectories [6] [15].

As solving the problem over an infinite horizon does not seem practical, an equivalent cost function mimics an infinite horizon is formulated.

Theorem 1

For a system, $x_{k+1} = Ax_k + Bu_k$, the infinite horizon cost

$$J_{\infty}^{*}(x(0)) = \min_{u_0,u_1,\dots} \sum_{k=0}^{\infty} x_k' Q x_k + u_k' R u_k$$

can be substituted as a terminal cost function $x_N^T P x_N$, where x_N is the terminal state and P is the solution of ARE which approximates the problem. [6]

Let us consider a system defined as,

$$x_{\mathrm{k+1}} = \mathrm{A}x_{\mathrm{k}} + \mathrm{B}u_{\mathrm{k}} \tag{3}$$

Then the finite time optimal cost function to be solved is,

$$\mathrm{J_k}(x_{\mathrm{k}}) = \min_{\mathrm{u_0,\dots u_{n-1}}} \sum_{\mathrm{k=0}}^{\mathrm{N-1}} \left(x_{\mathrm{k}}^{\mathrm{T}} \mathrm{Q} x_{\mathrm{k}} + u_{\mathrm{k}}^{\mathrm{T}} \mathrm{R} u_{\mathrm{k}}\right) + x_{\mathrm{N}}^{\mathrm{T}} \mathrm{P} x_{\mathrm{N}} \tag{4}$$

subject to $x_N = Ax_{N-1} + Bu_{N-1}$

$\mathrm{J_k}(x_N) = x_N^T Q_P x_N + \sum_{k=0}^{N-1}(x_k^T Q x_k + u_k^T R u_k)$. For the finite problem as in,

$$J_N(x_N) = \min_{u_0,\dots u_{n-1}} \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + x_N^T Q_P x_N$$

According to the principle of optimality the optimal one step cost-to-go can be obtained,

$$\mathrm{J}_{\mathrm{N-1}}^{*}(x_{\mathrm{N-1}}) = \min_{u_{\mathrm{N-1}}} x_N^T Q_P x_N + x_{N-1}^T \mathrm{Q} x_{N-1} + u_{N-1}^T R u_{N-1} \tag{5}$$

$$x_N = Ax_{N-1} + Bu_{N-1},\ P = P_N \tag{6}$$

Substituting (6) into the objective function (5),

$$\mathrm{J}^*_{\mathrm{N-1}}(x_{\mathrm{N-1}}) = \min_{\mathrm{u_{N-1}}}\{x^T_{N-1}(\mathrm{A}^T\mathrm{P_N}\mathrm{A} + \mathrm{Q})x_{N-1} + 2x^T_{N-1}\mathrm{A}'\mathrm{P_N}\mathrm{B}u_{N-1} \tag{7}$$

$$+ u^T_{N-1}(\mathrm{B}^T\mathrm{P_N}\mathrm{B} + \mathrm{R})u_{N-1}\}$$

We note that the cost-to-go $\mathrm{J}^*_{\mathrm{N-1}}(x_{\mathrm{N-1}})$ is a positive definite quadratic function of the decision variable u_{N-1}. We find the optimum by setting the gradient to zero and obtain the optimal input

$$u^*_{\mathrm{N-1}} = -(\mathrm{B}^T\mathrm{P_N}\mathrm{B} + \mathrm{R})^{-1}\mathrm{B}^T\mathrm{P_N}\mathrm{A}x_{N-1} \tag{8}$$

and the one-step optimal cost-to-go,

$$\mathrm{J}^*_{\mathrm{N-1}}(x_{N-1}) = x^T_{N-1}\mathrm{P_{N-1}}x_{N-1} \tag{9}$$

where we have defined,

$$\mathrm{P_{N-1}} = \mathrm{A}^T\mathrm{P_N}\mathrm{A} + \mathrm{Q} - \mathrm{A}^T\mathrm{P_N}\mathrm{B}(\mathrm{B}^T\mathrm{P_N}\mathrm{B} + \mathrm{R})^{-1}\mathrm{B}^T\mathrm{P_N}\mathrm{A}. \tag{10}$$

At the next stage, consider the two-step problem from time $N-2$ forward, we can state the optimal solution as per the solution at previous step u^*_{N-1},

$$u^*_{N-2} = -(B^TP_{N-1}B + R)^{-1}B^TP_{N-1}Ax_{N-2}.$$

The optimal two-step cost-to-go is

$$J^*_{N-2}(x_{N-2}) = x^T_{N-2}P_{N-2}x_{N-2},$$

where we defined,

$$P_{N-2} = A^T P_{N-1} A + Q - A^T P_{N-1} B (B^T P_{N-1} B + R)^{-1} B^T P_{N-1} A$$

Continuing in this manner, at some arbitrary time k the optimal control action is,

$$u^*(\mathrm{k}) \;= -(\mathrm{B}^T \mathrm{P}_{\mathrm{k}+1} \mathrm{B} + \mathrm{R})^{-1} \mathrm{B}^T \mathrm{P}_{\mathrm{k}+1} \mathrm{A} x_k \;= F(\mathrm{k}), \text{ for } \mathrm{k} = 0, \dots, \mathrm{N} - 1 \tag{11}$$

where,

$$\mathrm{P}_{\mathrm{k}} = \mathrm{A}^T \mathrm{P}_{\mathrm{k}+1} \mathrm{A} + \mathrm{Q} - A^T P_{k+1} B (B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A \tag{12}$$

and the optimal cost-to-go starting from the measured state $x(k)$ is,

$$\mathrm{J}_{\mathrm{k}}^*\big(x(\mathrm{k})\big) = x_k^T \mathrm{P}_{\mathrm{k}} x_k \tag{13}$$

Equation (12) is called Discrete Time Riccati Equation or Riccati Difference Equation (RDE) and is initialized with $P_N = P$ and is solved backwards, i.e., starting with P_N and solving for P_{N-1}, etc. Note from that the optimal control action $u^*(k)$ is obtained in the form of a feedback law as a linear function of the measured function.

Infinite Horizon Problem

This controller corresponds to an infinite horizon control law. Notice that it is stabilizing and has a reasonable stability margin. Nominal stability is a guaranteed property of infinite horizon controllers as we prove in the next section. For continuous processes operating over a long time, it would be interesting to solve an infinite horizon problem. The problem becomes difficult if we were to solve for the infinite horizon problem as below, i.e., $N \rightarrow \infty$ [6]. The reason for that is the increase in the prediction horizon value which is not a good practice to follow because there are chances where the system may not follow the dynamics of the system in the presence of uncertainties.

$$J_{\infty}^{*}(x(0)) = \min_{u_0,u_1,\dots} \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k \tag{14}$$

To prove the cost from $N \to \infty$, is $x_N^T P x_N$, We need to do prove by using the dynamic programming approach by solving it recursively backwards.

Since the prediction must be carried out to infinity, application of the batch method becomes impossible. On the other hand, derivation of the optimal feedback law via dynamic programming remains viable. We can initialize,

$$P_k = A^T P_{k+1} A + Q - A^T P_{k+1} B (B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A \tag{15}$$

with the terminal cost matrix $P_0 = Q$ and solve it backwards for $k \to -\infty$. Let us assume for the moment that the iterations converge to a solution P_{∞}. Such P_{∞} would then satisfy the Algebraic Riccati Equation (ARE)

$$P_{\infty} = A^T P_{\infty} A + Q - A^T P_{\infty} B (B^T P_{\infty} B + R)^{-1} B^T P_{\infty} A \tag{16}$$

Then the optimal feedback control law is,

$$u^{*}(k) = -(B^T P_{\infty} B + R)^{-1} B^T P_{\infty} A \, x_k \, , k = 0, \cdots, \infty \tag{17}$$

and the optimal infinite horizon cost is

$$J_{\infty}^{*}\big(x(0)\big) = x_0^T \, P_{\infty} \, x_0 \tag{18}$$

Controller is referred to as the asymptotic form of the Linear Quadratic Regulator (LQR) or the ∞-horizon LQR.

3.2 Terminal constraint

Based on the discussion by Kouvaritakis & Cannon (Chapter 2.4 & 2.5) in their book [6]

about the invariant sets, the guarantees of closed-loop stability and convergence derived in the previous section rely on the assumption that the predictions generated by the tail, u_k satisfy constraints at each time k. However due to short sighted strategy, the finite horizon optimal control may lead to a infeasible solution where the constraint satisfaction cannot be ensured at all times [15].

It is clear that u_k only satisfies constraints over the first $N-1$ sampling intervals of the prediction horizon, Since the optimal predictions at time k, namely $\{u_{1|k}, \dots, u_{N-1|k}\}$, necessarily satisfy constraints. However, for u_{k+1} to be feasible at $k+1$, there is also a need that the N^{th} element of to satisfy constraints i.e., at u_{k+1}, $(u_N = Kx_N)$, and this requires extra constraints to be introduced in the MPC optimization at $k = N$.

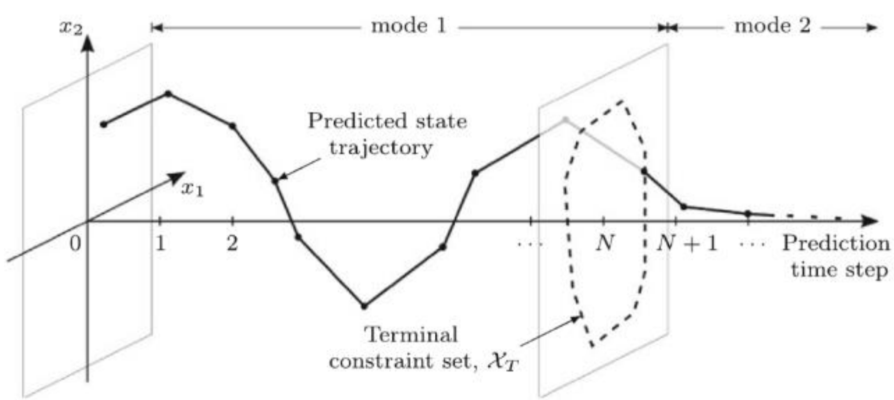

Figure 6 Terminal Constraint [7]

Extra constraints that are introduced to ensure feasibility of the tail are known as terminal constraints since they apply to mode 2 predictions (which are governed by time-invariant feedback) in Figure 6, and are therefore equivalent to constraints on the terminal state prediction $x_{N|k}$. For convenience we denote the region in which $x_{N|k}$ must lie to satisfy given terminal constraints as X_f [6] , where X_f is called terminal region.

Note that for stable systems without state constraints $K_\infty(X_f) = \mathbb{R}^n$ always, i.e., the choice of X_f is less critical. For unstable systems K∞(X_f) is the region over which the system can operate stably in the presence of input constraints and is of eminent practical importance [6].

Constraint means that it can be defined in either one of the two terminal points. It can be zero state constraints or the final state constraints. With just zero state constraints in the initial point, we still are not sure whether the solution is feasible. The terminal constraints must be constructed to ensure feasibility of the MPC optimization recursively [6], so that the tail predictions necessarily satisfy constraints, including the terminal constraints themselves. i.e., The constraint satisfaction at the last step of the horizon must happen successively. When we include constraints at both the terminal points , which is approximated as terminal cost (that represents an infinite horizon problem), then it represents a terminal constraint. Johan [11] addresses about approximating the terminal set and arriving at a terminal constraint through complex hull calculation and LMI methods and he concludes that those methods does not explicitly categorize the feasible set for all systems. Hence the easy way to determine the terminal set to understand the system's response for the input and define soft/hard constraints in such a way that it satisfies it from N to ∞.

Invariant Set

An MPC is called recursively feasible if it always keeps the states in a region from where the online optimization problem has a feasible solution. One way to achieve this is to restrict the states within a pre-computed robust controlled invariant set. These robust controlled invariant sets, however, can have a complicated geometry. Therefore, restricting the states in these sets may introduce too many state constraints and render the online optimization problem computationally expensive to solve. [12]

Consider the MPC problem formulation as below,

$$J_k(x_k) = \min_{u_0,\dots u_{n-1}} \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + x_N^T Q_P x_N \tag{19}$$

subject to,

$$x = A\, x_{k+1} + B x_k,$$

$$x_k \in X,$$

$$u_k \in U, where\ k = 0, \dots, N-1$$

$$x_N \in X_f$$

$$x_0 = x(t)$$

The following sets which are very important to prove feasibility and stability. C_∞: The maximal control invariant set C_∞ is only affected by the sets X and U, the constraints on states and inputs. It is the largest set over which we can expect any controller.
X_0: A control input U_0 can only be found, i.e., the control problem is feasible, if $x(0) \in X_0$. The set X_0 depends on X and U, on the controller horizon N and on the controller terminal set X_f. It does not depend on the objective function and it has generally no relation withC_∞.
O_∞ : The maximal positive invariant set for the closed-loop system depends on the controller and as such on all parameters affecting the controller, i.e., X, U, N, X_f and the objective function with its parameters P, Q and R. Clearly $O_\infty \subseteq X_0$ because if it were not there would be points in O_∞ for which the control problem is not feasible. Because of invariance, the closed loop is persistently feasible for all states $x(0) \in O_\infty$. Clearly, $O_\infty \subseteq C_\infty$. [6]

We can now state necessary and sufficient conditions guaranteeing persistent feasibility by means of invariant set theory.

Theorem 2

Necessary and sufficient conditions for the predictions generated by the tail u_{k+1} to be feasible at time $k+1$ whenever the MPC optimization at time k is feasible are, [10] [6]

1. The constraints are instantaneously satisfied at all points in X_f

2. X_f is invariant in mode 2 , which is equivalent to requiring that $(A + BK)x_{N|k} \in X_f$ for all $x_{N|k} \in X_f$

If X_f satisfies (i) and (ii), then the MPC optimization is,

$$J_k(x_k, u_k) = \min_{u_0,\dots u_{n-1}} \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + x_{k|N}^T Q_P x_{k|N}$$

subject to,

$$u_{min} \leq u \leq u_{max} , x_{min} \leq x \leq x_{max}$$

$$x_k \in X,\ u_k \in U ,\ x_n \in X_f ,\ 0 \leq k \leq N - 1$$

where the terminal set is given by the output $\boldsymbol{x}$, $x_i - x_{ref} \leq 0$ $x_N \in X_f$ is guaranteed to be feasible at all times $k > 0$ provided it is feasible at $k = 0$.

Theorem 3

Consider the RHC law (13) as in with N ≥ 1. If X_f is a control invariant set for a system, then the RHC is persistently feasible.

Proof: If X_f is control invariant, then $X_{N-1}, X_{N-2}, \ldots, X_1$ are control invariant and Lemma 2 establishes persistent feasibility for all feasible u.

Lemma 1

Let O_∞ be the maximal positive invariant set for the closed-loop system $x(k + 1) = f_{cl}(x(k))$ with constraints $x(t) \in X, u(t) \in U, \forall t \geq 0$. The RHC problem is persistently feasible if and only if $X_0 = O_\infty$. [6]

Proof:

For the RHC problem to be persistently feasible X_0 must be positive invariant for the closed-loop system. We argued above that $O_\infty \subseteq X_0$. As the positive invariant set X_0 cannot be larger than the maximal positive invariant set O_∞, it follows that $X_0 = O_\infty$.

As X_0 does not depend on the controller parameters P, Q and R but O_∞ does, the requirement $X_0 = O_\infty$ for persistent feasibility shows that, in general, only some P, Q and R are allowed. The parameters P, Q and R affect the performance. The complex effect they have on persistent feasibility makes their choice extremely difficult for the design engineer. In the following we will remedy this undesirable situation. We will make use of the following important sufficient conditions for persistent feasibility.

Lemma 2

Consider the RHC law from the definition above with $N \geq 1$. If X_1 is a control invariant set for a system, then the RHC is persistently feasible. Also, $O_\infty = X_0$ is independent of P, Q and R. [6]

Proof: If X_1 is control invariant then, by definition, $X_1 \subseteq Pre\ (X_1)$. Also recall that $Pre\ (X_1) = X_0$ from the properties of the feasible sets. (Note that $Pre\ (X_1) \cap X = Pre\ (X_1)$ from control invariance). Pick some $x \in X_0$ and some feasible control u for that x and define $x^+ = Ax + Bu \in X_1$. Then $x^+ \in X_1 \subseteq Pre\ (X_1) = X_0$. As u was arbitrary (if it is feasible) $x^+ \in X_0$ for all feasible u. As X_0 is positive invariant, $X_0 = O_\infty$. As X_0 is positive invariant for all feasible u, O_∞ does not depend on P, Q and R.

A terminal constraint is $x(k+N) \in X_f$, where X_f is the terminal set [13]. To ensure that the tail satisfies constraints over the first N steps of the prediction horizon at time $k+1$, we need to include a terminal constraint, [7]

$$x(k+N) \in X_f \begin{cases} u_{min} \leq K * x_{N|k} \leq u_{max} \\ x_{min} \leq x_{N|k} \leq x_{max} \end{cases} \tag{20}$$

However, an optimization problem with terminal constraints will be computationally complex. From the optimization point of view, as being feasible means to satisfy certain constraint conditions at every time step in a recursive manner and optimization takes more time to find the minimum with the effects of constraints in place whereas an infeasible problem gives up the search after some time by returning a value which may or may not be a feasible solution. Hence the control input must be finite all the time indefinitely.

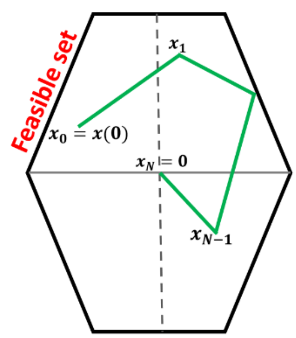

Figure 7 Feasible Set

Let us assume that we measure a state $x(t)$ at some time instance t. We then obtain $U_t^*(x(t))$ by solving an optimization problem. If the optimization renders an empty set $U_t^*(x(t)) = \Phi$, then it means that the problem is infeasible. Then the optimization stops at that point and the first control action u_t^* from U_t^* is applied and then the problem moves to next time step $t+1$.This process is continued, and this is called online receding horizon control. [6]

What if we can find feasible u for that all X at that point. That's when, we can guarantee recursive feasibility. That's the reason we need a constrained optimization solver at the time t. We need to define a set of X for which u is always feasible and could return a solution. That set is called the terminal set/region and it denoted by X_f. Authors are still researching about the best method to arrive a suitable terminal region for a system and the common method to do that is to establish a convex set. But the simplest approach is to design a terminal region such that , $x_n \in X_f = 0$. [15]

Theorem 4

The simplest approach is to design a terminal region such that , $x_n \in X_f = 0$ so that we can have a feasible control input at all instances. [F.Borrelli, M. Morari and C. Jones] [15]

1. Assume feasibility of x_0 & let $\{u_0^*, u_1^*, \dots, u_{N-1}^*\}$ is the optimal control sequence computed at x_o to give $\{x_o, x_2, \dots . x_n\}$ as the corresponding state trajectory.
2. Apply u_0^* , then $x(1) = A * x_0 + B * u_0^*$.
3. At $x(1)$, the control sequence is $\{u_1^*, \dots, u_{N-1}^*, 0\}$ is feasible. i.e., $x_{n+1} = 0$.

For some $x_N \in X_f$, At $x(1)$, the control sequence is $\{u_1^*, \dots, u_{N-1}^*, v(x_N)\}$ is feasible.

i.e., $x_{N+1} = A * x_N + B * v(x_N) \in X_f$

3.3 Terminal Cost

Having ensured the feasibility, we need to focus on stability of an MPC. Persistent feasibility does not guarantee that the closed-loop trajectories converge towards the desired

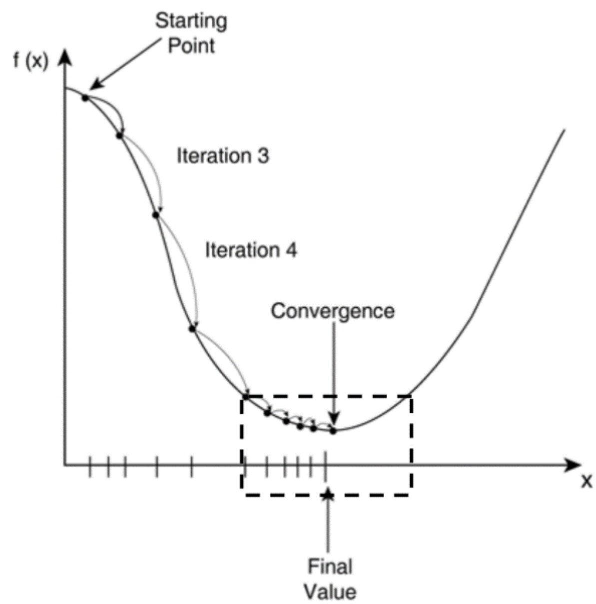

Figure 8 Property of Terminal Cost [7]

equilibrium point [6]. Hence, we need to establish a additional term called terminal cost. One of the earliest proposals for modifying cost function to ensure closed-loop stability was the addition of a terminal cost. The proposal was made in the context of predictive control of unconstrained linear system for which the choice $x_N^T P x_N$ is appropriate [13]. The matrix P, the terminal value of the Riccati difference equation, is chosen so that the sequence obtained by solving the Riccati difference equation in reverse time with terminal condition is monotonically non-increasing and it represents the cost as the infinite horizon. A monotonically non increasing function tells us the convergence is ensured consistently and consistent convergence is said to be stable most of the times. One other point to note is that errors at any stage of the computation are not amplified but are attenuated as the computation progresses.

The main reason for considering the addition of terminal cost in MPC is to increase the stability. The stability of the problem can be ensured by finding the Lyapunov function for the closed loop system. We will show next that if the terminal cost and constraint are appropriately chosen, then the value function $J_0^*(x(.)$ is a Lyapunov function. [6]

Terminal cost is a quadratic function which is calculated at the last step of the prediction horizon. It approximates the infinite horizon to a finite horizon problem. Terminal constraint is a set that is defined in the form of constraints on the state variable such that it remains in the feasible set for all time, t. The terminal set is defined is based on the system responses.

We try to prove the result by simulating the spring mass system below as in and we find that the convergence characteristics increases as we increase the prediction horizon for the same control horizon. As per the definition of stability. Hence when N reaches infinity , the convergence characteristics is improved. and closed loop stability can be guaranteed by using the terminal cost and suitable terminal regions [13]. The response of the system converges to the steady state closed loop eigen values [7]. This tells that the nominal stability and a better stability margin can be guaranteed in an infinite horizon controller. Hence, the terminal cost is chosen to be the value function of infinite horizon unconstrained optimal control problem, there exists a set of initial states for which MPC is actually optimal for the infinite horizon constrained optimal control problem and therefore inherits its associated advantages [7] . Therefore, conditions will be derived on how the terminal weight P chosen according to theorem 1 can be used in a quadratic function which is proved to be a Control Lyapunov Function (CLF) such that closed-loop stability is ensured.

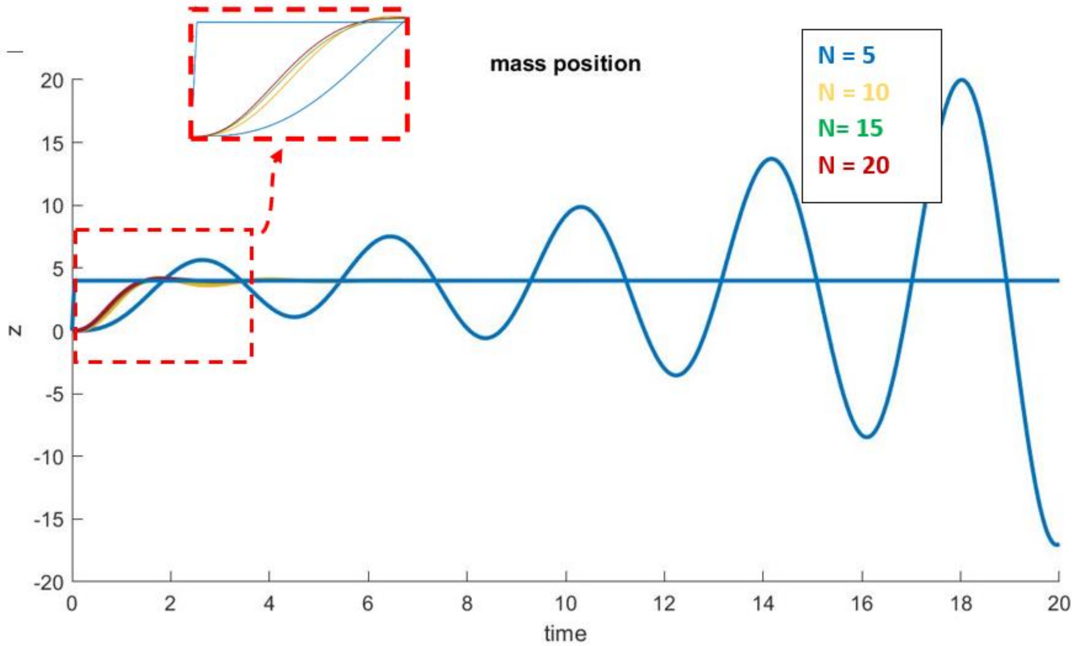

Figure 9 System Oscillation When N Increases

Theorem 5

Consider the equilibrium point $x = 0$ of a discrete system. Let $\Omega \subset R^n$ be a closed and bounded set containing the origin. Assume there exists a function $V: R^n \to R$ continuous at the origin, finite for every $x \in \Omega$, and such that,

$$\mathrm{V}(0) = 0 \text{ and } \mathrm{V}(x) > 0, \forall x \in \Omega \setminus \{0\}\ \mathrm{V}(x_{\mathrm{k+1}}) - \mathrm{V}(x_k) \leq -\alpha(x_k) \forall x_k \in \Omega \setminus \{0\} \tag{21}$$

where $\alpha: R^n \to R$ is a continuous positive definite function. Then $x = 0$ is asymptotically stable in Ω. [13]

Theorem 6

The Control Lyapunov Function should follow these conditions as given below, [13] [6] [15]

Consider system (3), the RHC law (19) and the closed-loop system.

(A0) The stage cost $q(x, u)$ and terminal cost $p(x)$ are continuous and positive definite functions.

(A1) The sets X, X_f and U contain the origin in their interior and are closed.

(A2) X_f is control invariant, $X_f \subseteq X$

(A3) $min_{v \in U,}^{Ax+Bv \in X_f} (-p(x) + q(x, v) + p(Ax + Bv)) \leq 0, \forall x \in X_f$.

Then, the origin of the closed-loop system (3) is asymptotically stable with domain of attraction X_0.

Proof:

From hypothesis (A2), Theorem 3 and Lemma 1, we conclude that $X_0 = O_\infty$ is a positive invariant set for the closed-loop system (3) for any choice of the cost function. Thus, persistent feasibility for any feasible input is guaranteed in X_0.

Next, we prove convergence and stability. We establish that the function $J_0^*(\cdot)$ in (18) is a Lyapunov function for the closed-loop system. Because the cost J_0, the system and the constraints are time-invariant we can study the properties of J_0^* between step $k = 0$ and step $k + 1 = 1$.

Consider problem (3) at time $t = 0$. Let $x(0) \in X_0$ and let $U_0^* = \{u_0^*, \dots, u_{N-1}^*\}$ be the optimizer of problem and $x_0 = \{x(0), x_1, \dots, x_N\}$ be the corresponding optimal state trajectory. After the implementation of u_0^* we obtain $x(1) = x_1 = Ax(0) + Bu_0^*$.

For $t = 1$. We will construct an upper bound on $J_0^*(x(1))$. Consider the sequence $U_1 = \{u_1^*, \dots, u_{N-1}^*, v\}$ and the corresponding state trajectory resulting from the initial state $x(1), X_1 = \{x_1, \dots, x_N, Ax_N + Bv\}$.

Because $x_N \in X_f$ and (A2) there exists a feasible v such that $x_{N+1} = Ax_N + Bv \in X_f$ and with this v the sequence $U_1 = \{u_1^*, \dots, u_{N-1}^*, v\}$ is feasible. Because U_1 is not optimal $J_0(x(1), U_1)$ is an upper bound on $J_0^*(x(1))$. Since the trajectories generated by U_0^* and U_1 overlap, except for the first and last sampling intervals, it is immediate to show that,

$$\mathrm{J}_0^*\big(x(1)\big) \le \mathrm{J}_0(x(1), \mathrm{U}_1) = \min_{\mathrm{u}_0, \dots \mathrm{u}_{n-1}} \sum_{k=0}^{N-1} \mathrm{q}(x_k, u_k^*) + \mathrm{q}(x_\mathrm{n}, \mathrm{v}) + \mathrm{p}(\mathrm{A}x_N + \mathrm{Bv}) \qquad (22)$$

Adding $J_0^*(x(0)$ and subtracting from $q(x_0, u_0^*)$ will give us the cost from 1 to $N + 1$,

$$\begin{aligned} \mathrm{J}_0^*(x(1)) &\le \mathrm{J}_0(x(1), \mathrm{U}_1) \\ &= J_0^*(x(0)) - \mathrm{q}(x_0, u_0^*) - \mathrm{p}(x_N) + \big(\mathrm{q}(x_N, \mathrm{v}) + \mathrm{p}(\mathrm{A}x_N + \mathrm{Bv})\big) \end{aligned} \qquad (23)$$

Let $x = x_0 = x(0)$ and $u = u_0^*$. Under assumption (A3), (11) becomes,

$$\mathrm{J}_0^*(\mathrm{A}x + \mathrm{B}u) - \mathrm{J}_0^*(x) \le -\mathrm{q}(x, u), \forall x \in \mathrm{X}_0 \qquad (24)$$

Equation (24) and the hypothesis (A0) on the stage cost $q(\cdot)$ ensure that $J_0^*(x)$ strictly decreases along the state trajectories of the closed-loop system for any $x \in X_0, x \neq 0$. In addition to the fact that $J_0^*(x)$ decreases, $J_0^*(x)$ is lower-bounded by zero and since the state trajectories generated by the closed-loop system starting from any $x(0) \in X_0$ lie in X_0 for all $k \geq 0$, equation is sufficient to ensure that the state of the closed-loop system converges to zero as $k \to 0$ if the initial state lies in X_0. We have proven that in order to prove stability via we must establish that $J_0^*(x)$ is a Lyapunov function. For continuity at the origin, we will show that $J_0^*(x) \leq p(x), \forall x \in X_f$ and as $p(x)$ is continuous at the origin (by hypothesis (A0)) $J_0^*(x)$ must be continuous as well. From assumption $(A2), X_f$ is control invariant and thus for any $x \in X_f$ there exists a feasible input sequence $\{u_0, \dots, u_{N-1}\}$ for problem (12.6) starting from the initial state $x_0 = x$ whose corresponding state trajectory is $\{x_0, x_1, \dots, x_N\}$ stays in X_f, i.e., $x_i \in X_f \forall i = 0, \dots, N$. Among all the input sequences $\{u_0, \dots, u_{N-1}\}$ we focus on the one where u_i satisfies assumption (A3) for all $i = 0, \dots, N-1$. Such a sequence provides an upper bound on the function J_0^*.

$$J_0^*(x_0) \leq \left(\sum_{i=0}^{N-1} q(x_i, u_i)\right) + p(x_N), x_i \in X_f, i = 0, \dots, N$$

which can be rewritten as

$$J_0^*(x_0) \leq \left(\sum_{i=0}^{N-1} q(x_i, u_i)\right) + p(x_N) \quad = p(x_0) + \left(\sum_{i=0}^{N-1} q(x_i, u_i) + p(x_{i+1}) - p(x_i)\right) x_i \in X_f, i = 0, \dots, N$$

which from assumption (A3) yields,

$$\mathrm{J}_0^*(x) \leq \mathrm{p}(x), \forall x \in \mathrm{X_f} \tag{25}$$

In conclusion, there exist a finite time in which any $x \in X_0$ is steered to a level set of $J_0^*(x)$ contained in X_f after which convergence to and stability of the origin follows.

Remark

- The assumption on the positive definiteness of the stage cost $q(\cdot)$ in can be relaxed as in theorem 6 as in standard optimal control.
- The procedure outlined in theorem 6 is, in general, conservative because it requires the introduction of an artificial terminal set X_f to guarantee persistent feasibility and a terminal cost to guarantee stability.
- A function $p(x)$ satisfying assumption (A3) of theorem 6 is often called the control Lyapunov function.

Model predictive control employs both a terminal cost F(x) and a terminal constraint $x(k+N) \in X_f$ in the optimal control problem and is the version attracting most attention in current research literature [68]. It has improved performance when compared with zero state and terminal constraint set MPC and can handle a much wider range of problems than just terminal cost MPC. This is one of the basic strategies to ensure stability and feasibility together. The action of convergence will make the system to reach the optimal point while satisfying the set of constraints which ensures feasibility at every time step. If the terminal constraint is removed from the optimization problem, then the optimal cost may not be a Lyapunov function for the system and, moreover, the feasibility may be lost.

However, there are some predictive controllers with guaranteed stability which do not consider an explicit terminal constraint. It is important to note that for certain cases without terminal constraints, terminal cost may ensure feasibility if it attains the optimal point at every iteration that belongs to the feasible set as in Figure 6. However, it is not guaranteed always. With, there are significant research done in removing terminal constraints because of the complexity in computation depending on the system chosen. It is evident that the constraint arises because one often has a local, rather than a global, control Lyapunov function (CLF) for the system being controlled. In a few situations, a global CLF is available, in which case a terminal constraint is not necessary [13]. When you remove terminal constraints, we are indirectly reducing the domain of attraction which means the ability of the controller to reach the optimal point is reduced. Hence , the removal for terminal constraints has significant effects on the system,

1. Domain of attraction is decreased.
    - Let us assume that A is an invariant set,
    - There exists a neighborhood of A, called the domain of attraction for A which is denoted as B(A). Domain of attraction is the region of the phase space, over which iterations are defined, such that any point (any initial condition) in that region will be asymptotically iterated.
    - It consists of all points b that "enter A in the limit t → ∞". More formally, B(A) is the set of all points b in the phase space with the following property.
    - Similarly, we have terminal constraint, x_N that help us to enter X_f when t → ∞. When we remove terminal constraint , we reduce the ability of the controller to reach the terminal region.

If additional weights to terminal cost assuming the set is control invariant, the domain of attraction can be increased, and we don’t require additional constraints [21]. In the sense of Lyapunov, the convergence characteristics is increased as we make the terminal cost to attain the further minimum value (negative value) by weighting it by a constant. Sometimes terminal constraints can enhance stability characteristics of a system. Stability is a direct result of incorporating the stability constraint x(T) = 0 in the optimal control problem [13]. De Nicolao, Magni and Scattolini (1996a) and Magni and Sepulchre (1997) employed the terminal constraint $x(k + N) = 0$ to establish closed-loop stability when the system is nonlinear and unconstrained. This ideal in cases when $N \rightarrow \infty$, then $x \rightarrow 0$. However , it is not true for all systems. While there is a lot of research on how to find terminal constraint set, there is a significant study to eliminate the use of terminal constraint due to complexity issues which is addressed in the later section of the paper.

The terminal region is regarded as a good estimation of the stability region [7]. In practice, sometimes choosing a good terminal cost is enough (i.e., don't need to enforce a terminal control invariant condition), though you may be sacrificing guarantees [16]. While there are other modern methods to arrive at a suitable terminal set in the form of polytope, terminal sets in general are not well understood by practitioners & requires advanced tools to compute polyhedral or LMI [6]. $x_n \in X_f = 0$ is the simplest choice but it has small region of attraction when the N is small.

CHAPTER 4: NON-OVERSHOOTING MODEL PREDICTIVE CONTROL

4.1 Problem Statement

Consider a general non-linear system which is the form, [3]

$$x(k+1) = f\big(x(k), u(k)\big)$$
$$y(k) = hx(k)$$

The objective of the MPC design is to find a control input that minimizes the following cost function,

$$J(k) = \sum_{j=1}^{N-1} \left[x\big(k+j \ |k\big) \ - x_{ref})^T] \,)\right] * Q * \left[x(k+j \mid k\,) - x_{ref}\right] + \sum_{j=1}^{P-1} \Delta u(k+j \mid k)^T * R * \Delta u(k+j \mid k) \tag{26}$$

Subject to, $x(k+j|k) \in X$, $u(k+j|k) \in U$, $0 \leq \text{j} \leq \text{P}$

Symbol	Parameter
Q & R	Weighting matrices.
x_{ref}	Reference State
N	Prediction horizon
P	Control Horizon
X & U	Subset of $\mathbb{R}^n$ & $\mathbb{R}^m$
$\Delta u(k+j \mid k)$	$u(k+j \mid k) - u(k+j-1 \mid k)$

Table 2 Parameter Information

To achieve the non-overshooting MPC design of system outputs with respect to the references, four different inequality constraints (C1-C4) are proposed as follows, [3]

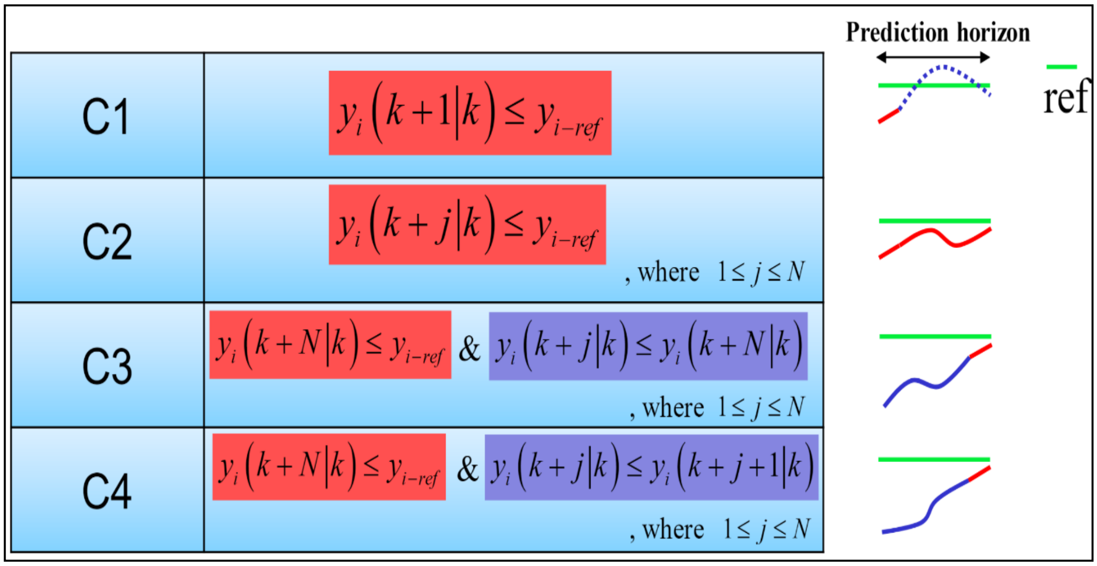

Figure 10 Non-overshooting Constraints [19]

where $y_i(k+N|k)$is the predicted value of the i^{th} system output at the j^{th} prediction horizon step and the y_{i-ref} is the i^{th} reference variable.

The action of different constraint has different effect on the optimal control sequence and there by generating appropriate system response.

C1: $\boldsymbol{y_i(k+1|k) \le y_{i-ref}}$

This makes sure that the first prediction value is less than the reference value of the system.

C2: $\boldsymbol{y_i(k+j|k) \le y_{i-ref}}$, where $\boldsymbol{1 \le j \le N}$

This makes sure that the prediction values from 1 to N to be less than the reference value of the system.

C3: $\boldsymbol{y_i(k+N|k) \leq y_{i-ref}}$ & $\boldsymbol{y_i(k+j|k) \leq y_i(k+N|k)}$, where $\boldsymbol{1 \leq j \leq N-1}$

This has 2 constraints . One is the terminal constraint which makes sure that the predicted value at the last step of the horizon is less than or equal to the desired value. While this is being ensured, it tells us that the system does not overshoot at the last step and hence the next constraint is constructed in such a way that the values predicted from 1 to N-1 is less than the terminal value.

C4 : $\boldsymbol{y_i(k+N|k) \leq y_{i-ref}}$ & $\boldsymbol{y_i(k+j|k) \leq y_i(k+j+1|k)}$, where $\boldsymbol{1 \leq j \leq N-1}$

The first constraint is same as the terminal constraint tin the C3 case. The second one is constructed in such a way that is ensures monotonic characteristic due to the constraint chain between each time step.

4.2 Linear System – Numerical Simulations

Consider a general spring mass damper system which is the form ,

$$\begin{aligned} \dot{x} &= \boldsymbol{A}x + \boldsymbol{B}u \\ y &= \boldsymbol{C}x + \boldsymbol{D}u \end{aligned} \tag{27}$$

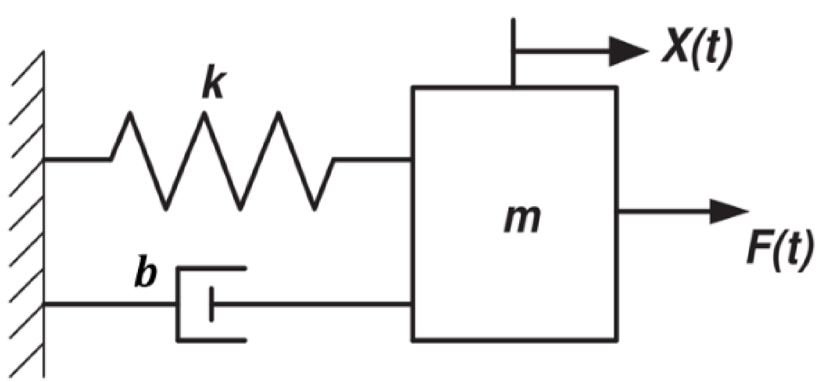

Figure 11 Mass Spring Damper System

Symbol	Parameter	Value
m	mass	5 kg
k	Spring Constant	0.1 N/m
b	Damping Coefficient	0.1 N.s/m
Ts	Sample Time	0.1 s
$\boldsymbol{x(0)}$ **&** $\boldsymbol{x_{ref}}$	Initial state & reference State	[0,0] & [4,0]

Table 3 System Properties & Values

Here the optimization problem is,

$$J(k) = \sum_{j=1}^{N-1} \left[x\left(k+j \mid k\right) - x_{ref})^T]\,)\right] * Q * \left[x(k+j \mid k) - x_{ref}\right] + \tag{28}$$

$$\sum_{j=1}^{P-1} \Delta u(k+j \mid k)^T * R * \Delta u(k+j \mid k)$$

Where $Q = diag[10 \quad 1]$, $R = 1$, N= P = 10.

As per the simulation responses below, the spring mass damper system is simulated as per the constraints defined above in Figure 10. For four non-overshooting constraints, C1 gives the shortest settling time. C2 and C3 cause similar but longer settling time compared with C1. C4 generates the longest settling time among the four constraints. From C1 to C4, since the constraints become stricter, the optimized control input sequence becomes more conservative, which may cause a longer settling time.

The presence of overshoot in C4 as in Figure 13 is due to the conservative action of the controller. The value of control input (force) must decrease eventually once the steady state is reached. But in this case, a distortion in control is observed even after the mass reaches the desired position. So when the output y fall below $y_{\mathrm{i-ref}}$, the condition $y_i(k+N|k) \leq y_{i-ref}$ is satisfied and the other predicted states are less than the terminal state at N as per $y_i(k+j|k) \leq y_i(k+j+1|k)$, where $1 \leq j \leq N-1$. Hence the error between the closed loop trajectory increases and this causes the change in the control action constantly and this constant fluctuation develops for a while resulting in overshooting to ensure that the mass in the desired position.

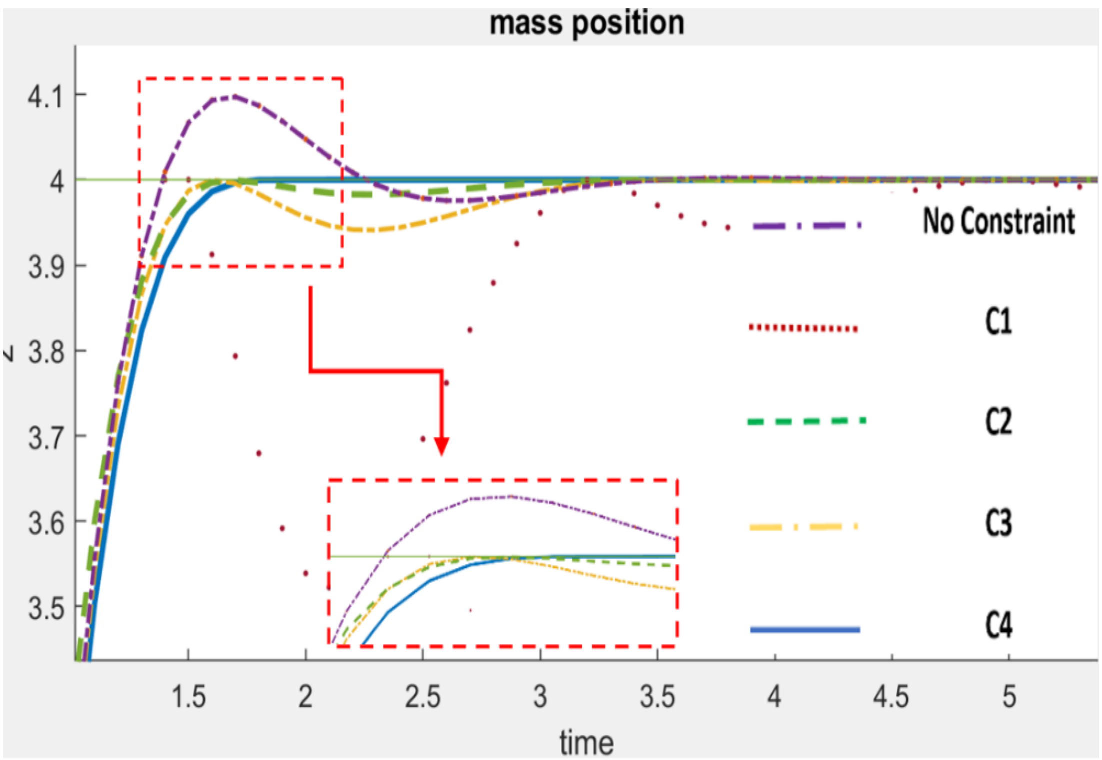

Figure 12 System Response for the Constraints C1- C4 [3]

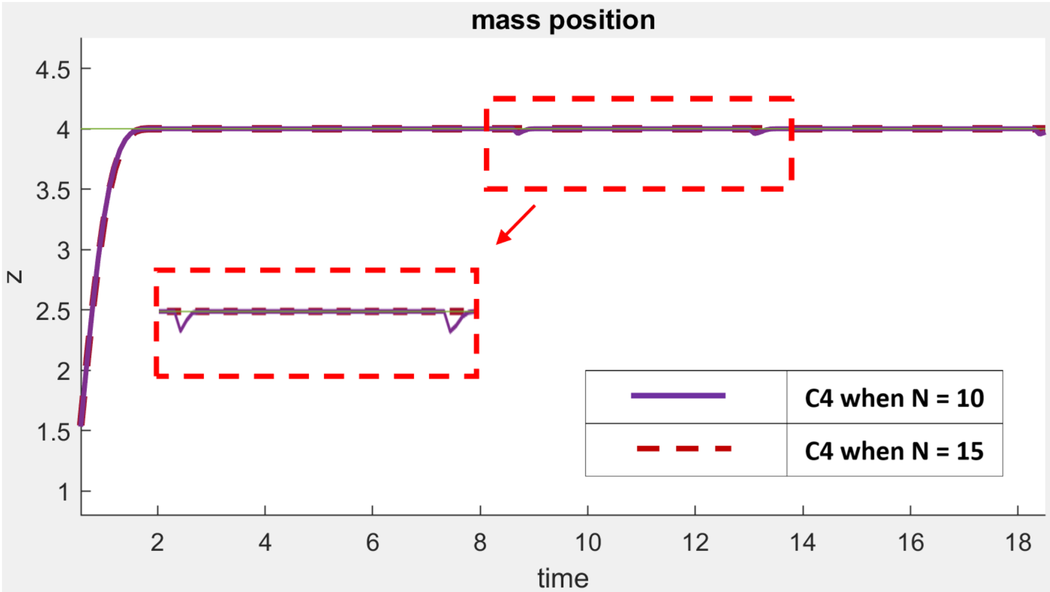

Figure 13 System Response for C4 When N Increases

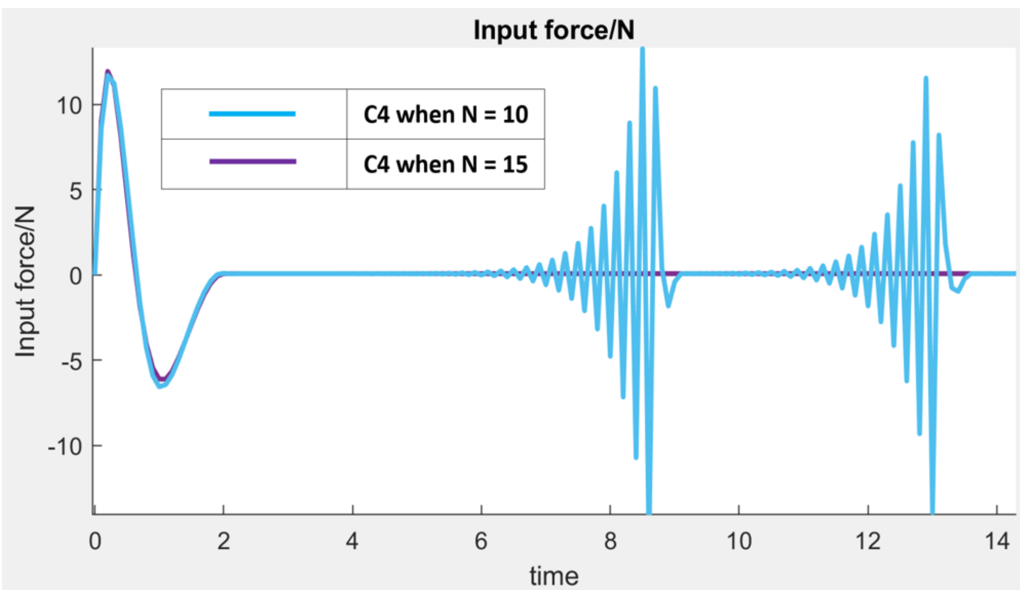

Figure 14 Input response of the System Response for C4

While we can make the system to not overshoot by modifying the horizon as in Figure 13, but the non-overshooting can happen anytime because of the system dynamics. Fox example, the system is overshooting at N = 10 and non-overshooting at N = 15. To deal with this kind of undesirable behavior due to design parameters , we need to make sure that the constraint terminal state must be equal to the desired state so that error is zero for tracking scenario. As the strictest constraint is C4 , we would consider the same by replacing the $y_i(k+N|k) \leq y_{i-ref}$ with $y_i(k+N|k) - y_{i-ref} = 0$ i.e., terminal equality constraint. This leads to a new design C5.

For the optimization problem as below,

$$J(k) = \sum_{j=1}^{N-1} \left[x\left(k+j \mid k\right) - x_{ref})^T] \,)\right] * Q * \left[x(k+j \mid k) - x_{ref}\right] + \sum_{j=1}^{P-1} \Delta u(k+j \mid k)^T * R * \Delta u(k+j \mid k) + \sum_{j}^{N} \left[x\left(k+N \mid k\right) - x_{ref})^T\right] * Q_p * \left[\, x\left(k+N \mid k\right) - x_{ref}\right]$$

where, $x(k+N\,|k) - x_{ref} \in X_f = 0$,

When $x_n(k+N\,|k) - x_{ref} = 0$, then $x_n(k+N+1\,|k) = 0 \;\forall\, k$ This method is one of the approaches, however it is very trivial. The same thing can be extended to general terminal region X_f,

At $x(1)$, $\{u_1^*, \ldots, u_{N-1}^*, v(x_n)\}$ is feasible: x_n is in $X_f \rightarrow v(x_n)$ is feasible and x_{n+1} = A*x_n + B*$v(x_n)$ in X_f, where v is the control input at N. The same result ensures persistent feasibility if the set defined is control variant (i.e. $v(x_n) \,\forall\, x_n \in X_f$), and positive invariant (i.e. $X_f \subseteq X$).

The common way to adopt a terminal region is by constructing an ellipsoidal set. D. Limon address the general terminal region to be $X_f = P(x_n) \leq \alpha$, where $\alpha > 0$.

i.e. $X_f = \{ x \in \mathbb{R}^n \mid P(x_n) \leq \alpha \}$, where $\alpha > 0$ is chosen such that $x \in X$, $K.x \in U$

$\Rightarrow X_f = \{ x \in \mathbb{R}^n \mid [x(k{+}N \mid k) - x_{ref})^T] * Q_p * [x(k + N \mid k) - x_{ref}] \} \leq \alpha$, For general cases, $\alpha = 1$.

C5: $\boldsymbol{y_i(k + N|k) - y_{i-ref} = 0}$ **&** $\boldsymbol{y_i(k + j|k) \leq y_i(k + j + 1|k)}$**, where** $\boldsymbol{1 \leq j \leq N - 1}$

The system is simulated in Figure 9 with the proposed constraints and the system response is better as the non-overshooting can be achieved. Apart from this, recursive feasibility can be guaranteed as we use a terminal equality constraint. In this case, the fluctuations are reduced because of the terminal equality constraint term in C5. It makes sure that there is a feasible control input to drive the state to zero at the end of the prediction horizon.

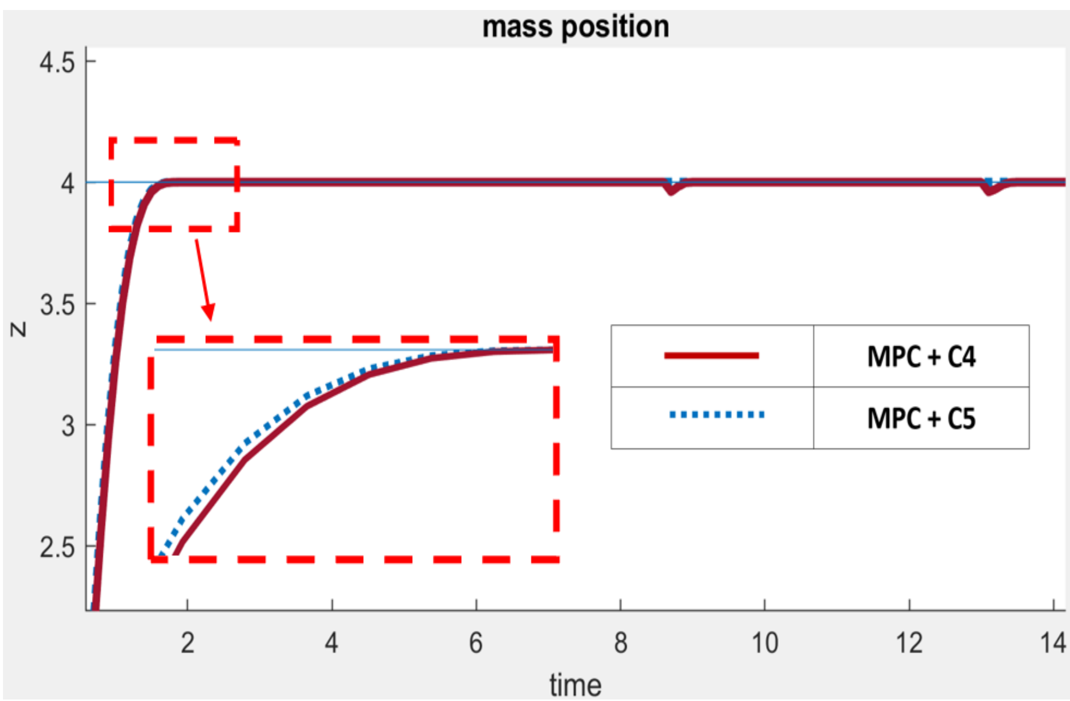

Figure 15 System Response for C4 & C5

The same system is simulated with a sampling time of 0.05s. When the sampling time gets reduced , the controller tends to have a better system response as seen in Figure 10.

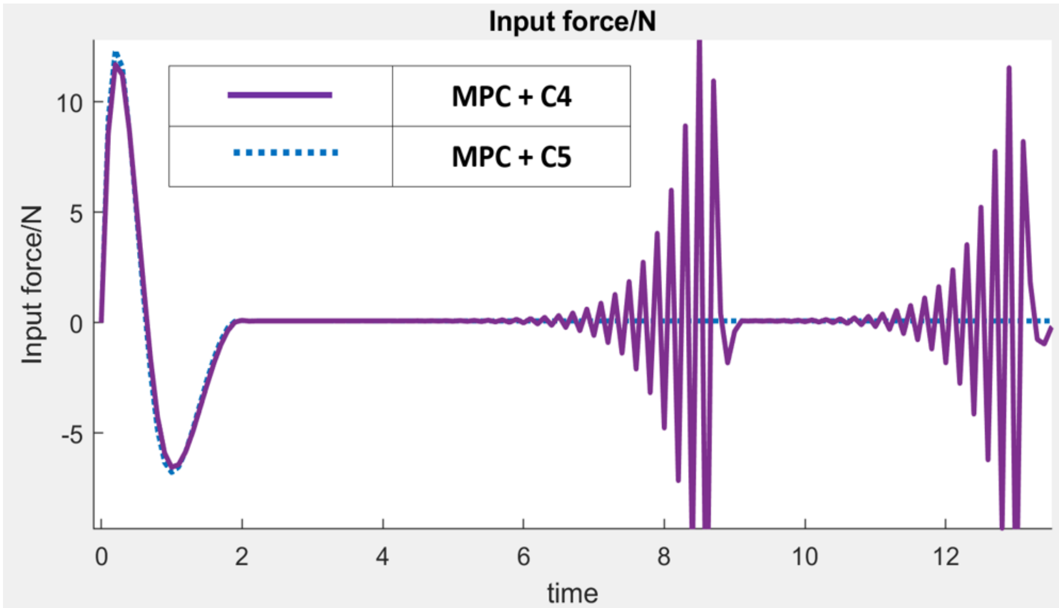

Figure 16 Input Responses of the System for C4 & C5

However, as the sampling time becomes small, the computational effort increases dramatically as C5. Moreover, as R decreases, the control input becomes more aggressive, which usually causes impractical control efforts and may violate the system capability. Therefore, to reduce the overshoot percentage and oscillation, small sampling time periods and R values are probably not practical and effective.

Effect of terminal cost.

With the support of literature, we know that terminal equality constraints ensure recursive feasibility in which there is a feasible control input for all time. But for the cases without the terminal equality constraint, terminal cost ensures that stability indefinitely upon the introduction of terminal cost which is a Lyapunov function.

The matrix Q_p, the terminal value of the Riccati difference equation, is chosen so that the sequence obtained by solving the Riccati difference equation in reverse time with terminal

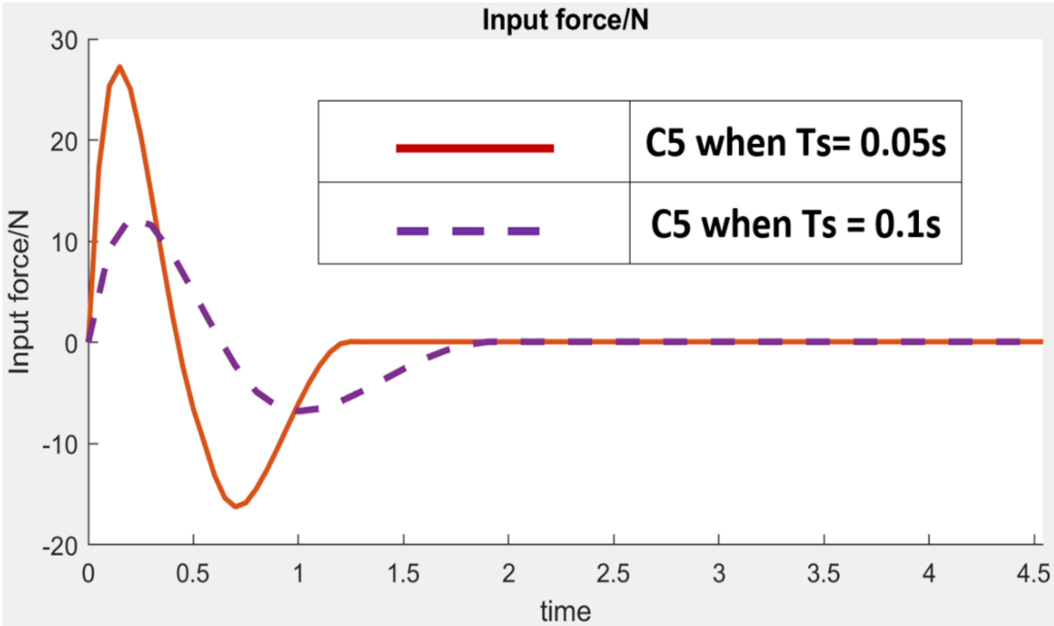

Figure 17 Input Responses of the System for Ts = 0.05s & Ts = 0.1s

sequence obtained by solving the Riccati difference equation in reverse time with terminal condition is monotonically non-increasing.

A monotonically non decreasing function tells us the convergence is ensured consistently and consistent convergence is said to be stable most of the times. One other point to note is that errors at any stage of the computation are not amplified but are attenuated as the computation progresses.

Terminal cost,

$$P(x(k+N|k)) = [x(k+N \mid k) - x_{ref})^T] * Q_p * [x(k+N \mid k) - x_{ref}],$$

$Q_p = dare(A, B, Q, R)$, Terminal constraint, $x(k+N|k) - x_{ref} = 0$.

Then the optimization problem become as follows with the additional term being added to the cost function as per the previous definition. This also increases the computation time as per the simulation process as MATLAB tend to take more time with this term.

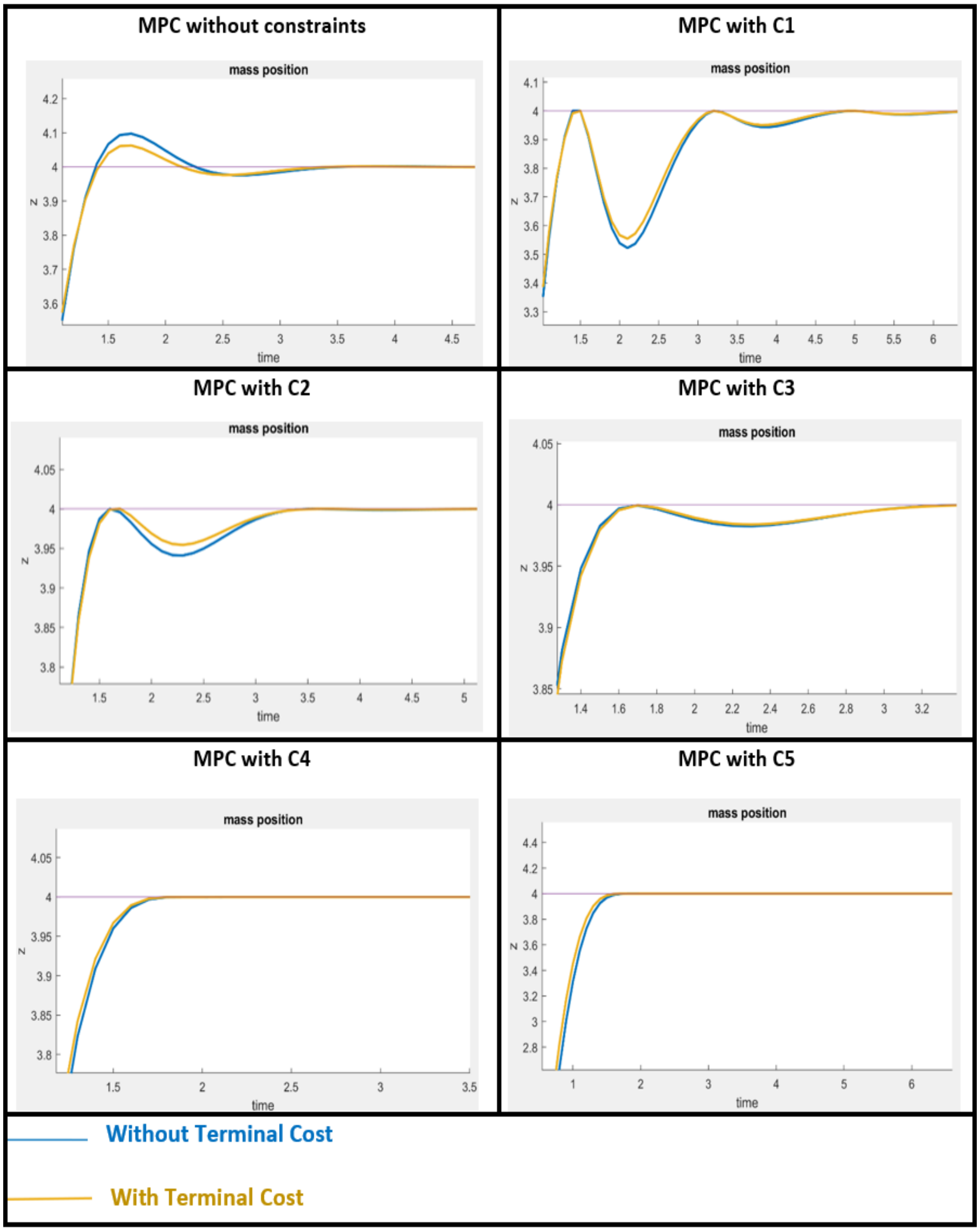

Figure 18 Effect of Terminal Cost on System Response

$$J(k) = \sum_{j=1}^{N-1} \left[x\left(k+j \mid k\right) - x_{ref})^T]\,)\right] * Q * \left[x(k+j \mid k) - x_{ref}\right] +$$

$$\sum_{j=1}^{P-1} \Delta u(k+j \mid k)^T * R * \Delta u(k+j \mid k) \tag{29}$$

$$+\sum_{j}^{N} \left[x\left(k+N \mid k\right) - x_{ref})^T\right] * Q_p * [\,x\left(k+N \mid k\right) - x_{ref}\right]$$

From the simulation results in Figure 19 , we could see that the convergence is improved upon the addition of the terminal cost. The settling time is also improved upon the addition of terminal cost. Hence this could be one of the effective approaches to design a MPC controller with guaranteed stability and feasibility with terminal cost and terminal equality constraints. This idea is presented through the simulation in figure 13 with the proposed C4 & C5 design. It is evident is that system has a quick convergence and less settling time while ensuring non-overshooting characteristic.

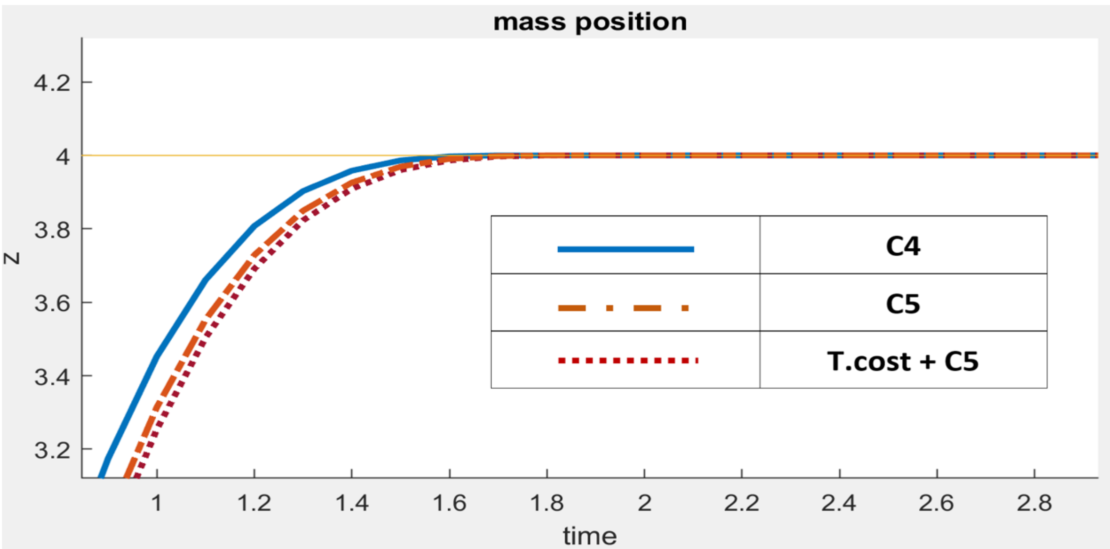

Figure 19 System Output for the MPC with C4, C5 & Terminal Cost + C5

4.3 Nonlinear system – Numerical Simulations

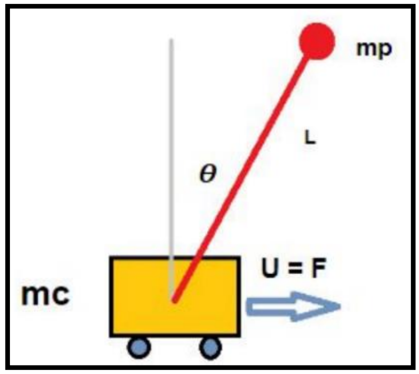

Figure 20 Inverted Cart Pendulum.

Consider a cart-pendulum system with the following dynamics,

$$x = [z \quad \dot{z} \quad \theta \quad \dot{\theta}]^T$$

$$\dot{x} = \begin{bmatrix} \dot{z} \\ \dfrac{F - K_d\dot{z} - m_pL\dot{\theta}^2\sin\,\theta + m_pg\sin\,\theta\cos\,\theta}{m_c + m_p\sin^2\,\theta} \\ \dot{\theta} \\ \dfrac{(F - K_d\dot{z} - m_pL\dot{\theta}^2\sin\,\theta)\cos\,\theta + (m_c + m_p)g\sin\,\theta}{L(m_c + m_p) - m_pL\cos^2\,\theta} \end{bmatrix} \quad (30)$$

where z, θ, and u are the cart position, pendulum angle, and input force applied on the cart. The parameter values are $m_p = m_c = 1$ kg, $L = 0.5$ m, and $K_d = 10\ \mathrm{N \cdot s/m}$. The initial condition is $x(0) = [0 \quad 0 \quad -\pi \quad 0]^T$ and reference is $x_{\mathrm{ref}} = [4 \quad 0 \quad 0 \quad 0]^T$. Weighting matrices are $Q = \begin{bmatrix} 10 & 0 \\ 0 & I_3 \end{bmatrix}$ and $R = [0.1]$. The prediction and control horizons are 10-time steps with the sampling step at $T_s = 0.1s$.

As observed in Figure 21, the overshoots of the nonlinear system responses are like those of the linear system example for the MPC design without and with the non-overshooting constraints C1-C4. For other characteristics of system responses, the oscillations are still observed for C1-C3, but the amplitudes become smaller as the constraints become stricter. For C4, since the system response is required. Unlike linear system ,we don't observe any overshoot or fluctuation in the results for different horizon. This is because we use aggressive control (R =0.1) for our system.

The definition of C1 to C4 is extended to the proposed nonlinear system and the system responses are recorded as well. However, as the sampling time becomes small, the computational effort increases dramatically. Moreover, as R decreases, the control input becomes more aggressive, which usually causes impractical control efforts and may violate the system capability. Therefore, to reduce the overshoot percentage and oscillation, small sampling time periods and R

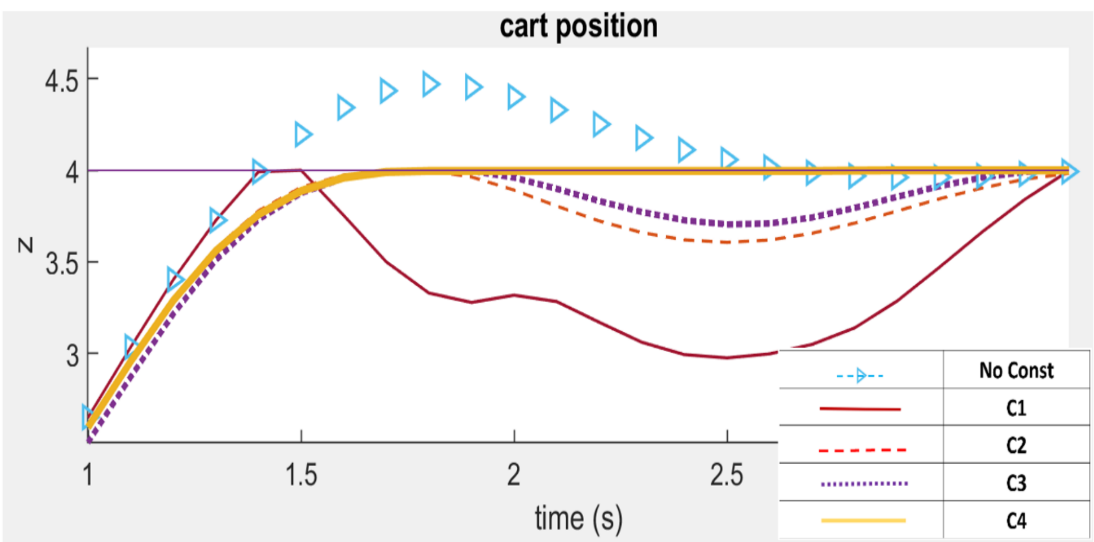

Figure 21 System Responses for C1 – C4 [3]

values are probably not practical and effective.

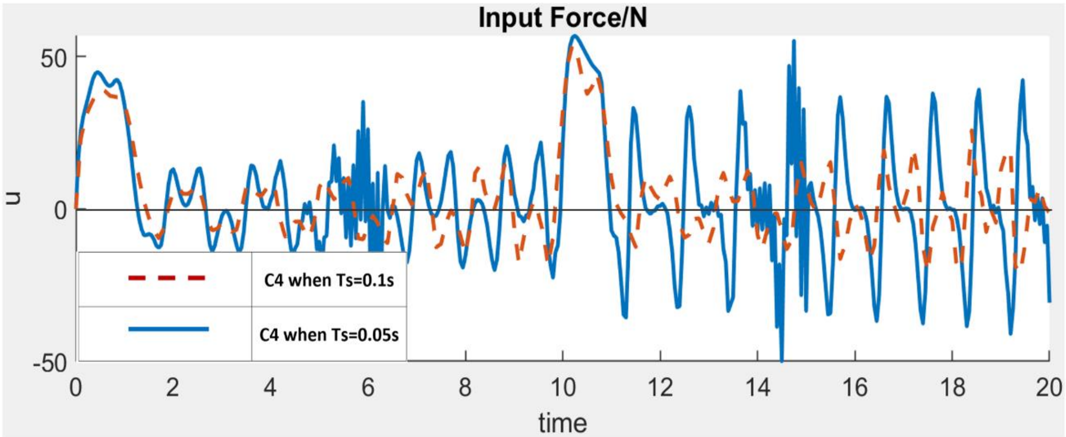

Figure 22 Input Force Profile When Ts = 0.1s & Ts = 0.05s

As explained, we don't observe any kinds of fluctuation as in linear system and we could conclude that the C4 is the strictest constraint with less settling time without many oscillations. As the strictest constraint is C4 , we would consider the same by replacing the $y_i(k+N|k) \leq y_{i-ref}$ with $y_i(k+N|k) - y_{i-ref} = 0$ i.e., terminal equality constraint. This leads to a new design,C5.

C5:$\boldsymbol{y_i(k+N|k) - y_{i-ref} = 0}$ **&** $\boldsymbol{y_i(k+j|k) \leq y_i(k+j+1|k)}$**, where1** $\boldsymbol{\leq j \leq N-1}$

When we try to extend the proposed non overshooting design here, we see that the system response better with quick convergence and less settling time when compared to other non-overshooting constraint C4 as in Figure 23. Moreover, we could ensure feasibility of the nonlinear MPC upon the addition of terminal equality constraint. [13] The convergence cannot be guaranteed always as it depends on the value of R when the sampling time changes.

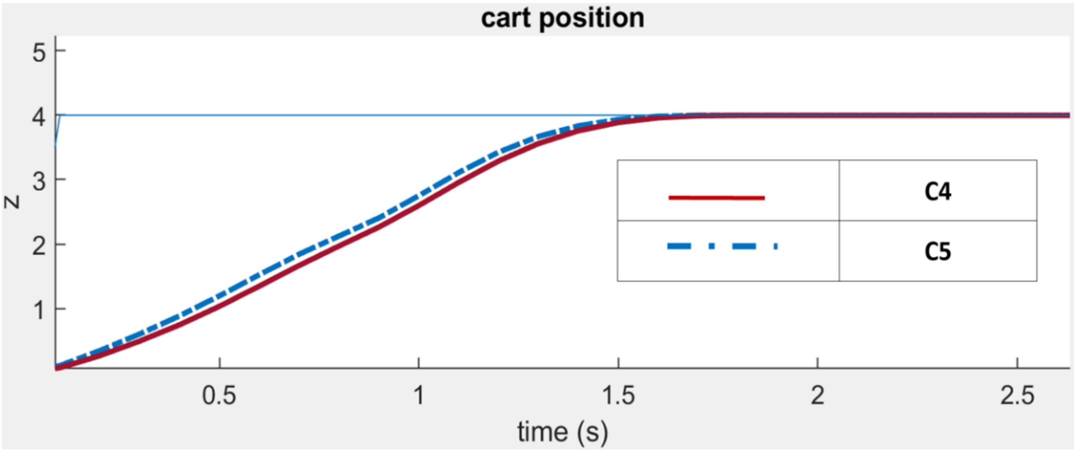

Figure 23 System Response for System with C4 & C5

The effect of predication horizon plays a major role here as the system tend to have a varied response as in Figure 24 and specific purpose for longer and shorter values of N. We notice that the prediction horizon with less N has better convergence, however it might have some adverse effect on the system due to the short-sighted behavior. If we choose a prediction horizon long enough is that the system will be more prone to disturbance and uncertainties and hence these methods are not ideal choice. We need to select the horizon in such a way that it satisfies all the dynamics of the system [20]. Hence N = 10 is chosen as one of the ideal values in carrying out the simulation.

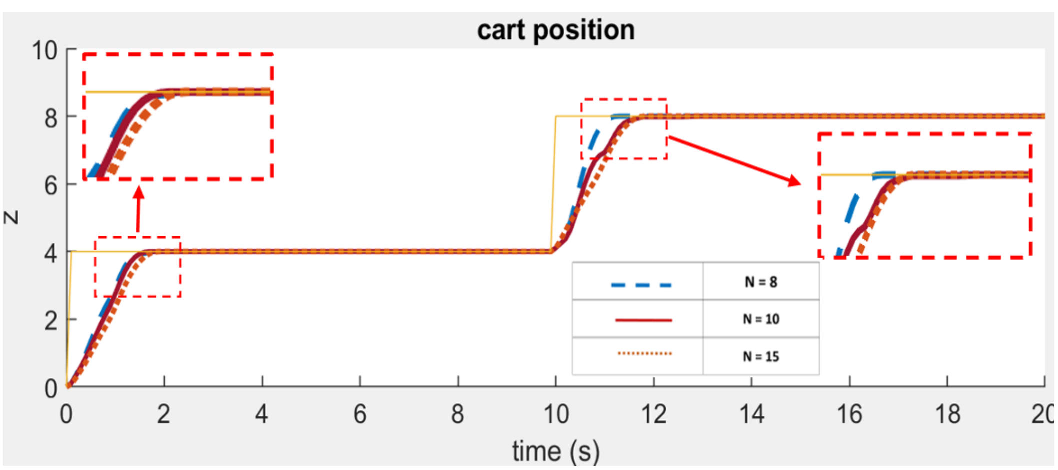

Figure 24 System Reponse for the Different Values of N

Effect of terminal cost.

As per the theory presented in this thesis, we know that the terminal cost is considered as a control Lyapunov function which is a monotonically increasing function where the convergence and stability are guaranteed.

Terminal cost, $P(x(k+N|k)) = [x(k+N \mid k) - x_{ref})^T] * Q_p * [\, x(k+N \mid k) - x_{ref}]$, The selection of Q_p plays a major role here and it cannot be obtained by solving the Riccati solution as this is not a linear system. As per the literature, the choice of Q_p is positive definite matrix. This idea is derived by the study of literature by multiple authors in their research on non-linear MPC tracking. We tend to present the same idea in this paper by considering the custom penalty matrix for the sake of trajectory convergence.

$$Q_p = diag(50,5,10,1)$$

Terminal constraint, $x(k+N\,|k) - x_{ref} = 0.$

Then the optimization problems become,

$$J(k) = \sum_{j=1}^{N-1} [x(k+j \mid k) - x_{ref})^T])] * Q * [x(k+j \mid k) - x_{ref}] + \sum_{j=1}^{P-1} \Delta u(k+j \mid k)^T * R * \Delta u(k+j \mid k) + \sum_{j}^{N} [x(k+N \mid k) - x_{ref})^T] * Q_p * [\, x(k+N \mid k) - x_{ref}] \tag{31}$$

The effect of terminal cost on the system response with every non overshooting constraint is presented individually in Figure 25. The common characteristic is that it reduces the oscillation for the response as in C1 – C4, and also it helps in better & smooth convergence. There is a significant decrease in the settling time of the system.

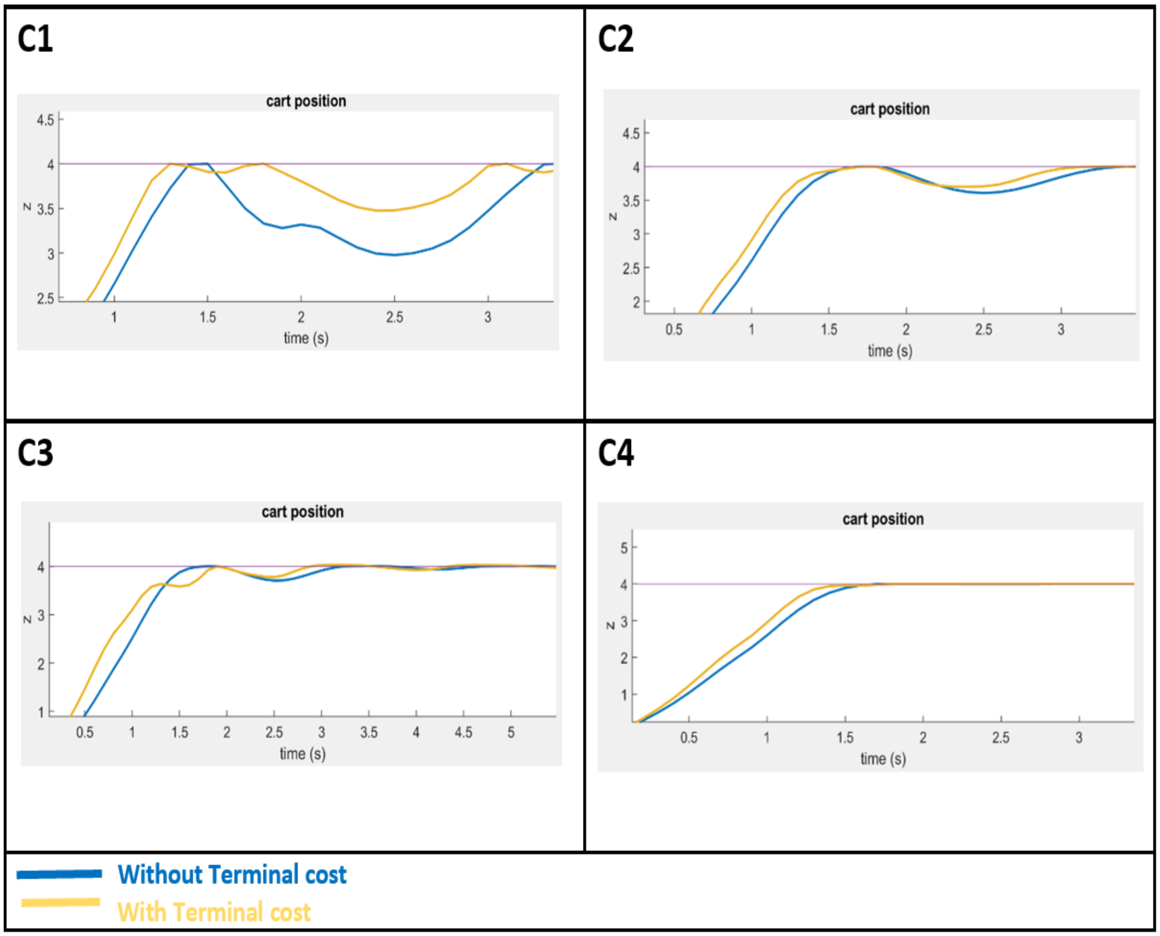

Figure 25 Effect of Terminal Cost on MPC with C1 to C4

The same idea ids extended to the design C5 below as in Figure 26 where the convergence is getting improved on the addition of terminal cost. The results are presented in a cumulative way as shown in Figure 27 for the better understanding. Hence by the theoretical study and the application of the practical scenarios, it is confirmed that the characteristics are similar for both linear and nonlinear systems.

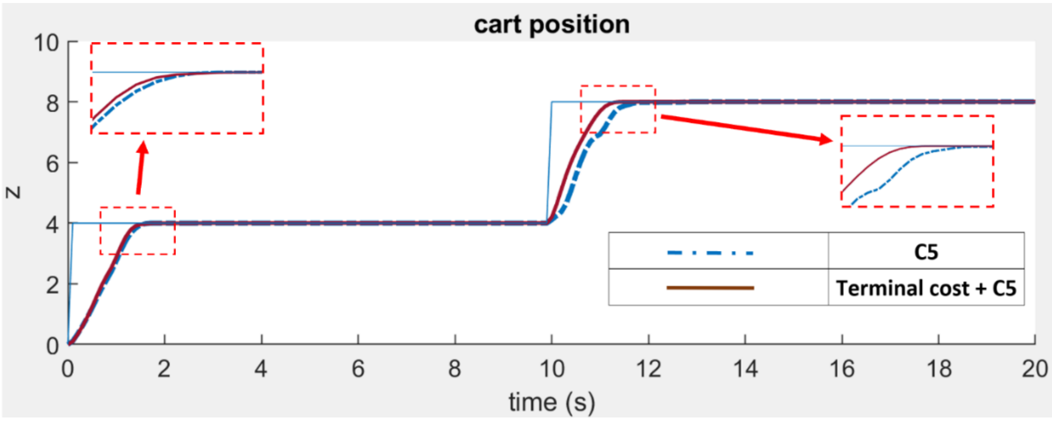

Figure 26 Effect of Terminal Cost on MPC with C5

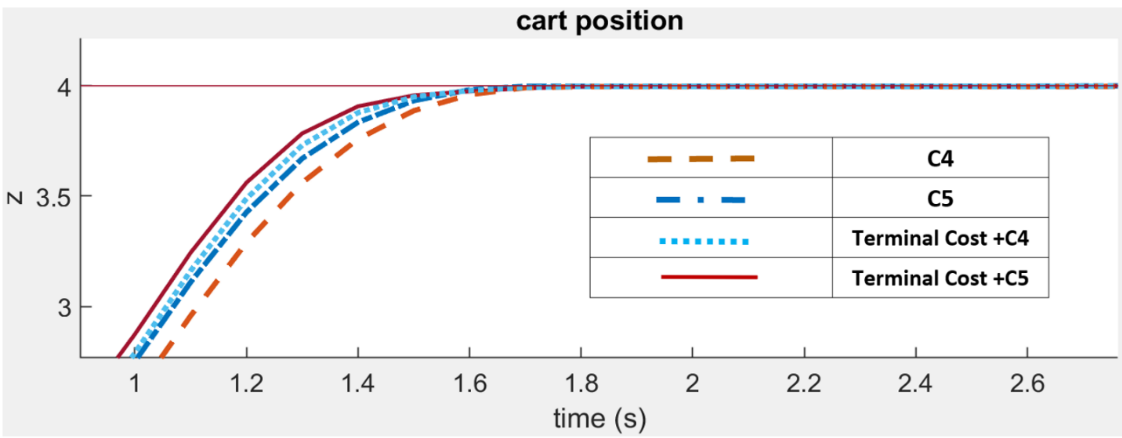

Figure 27 Effect of Terminal Cost on C4 & C5

$Convergence: Terminal\ cost + C5 > Terminal\ cost + C4 > C5 > C4$

$Settling\ Time:\ \ C4 > C5 > Terminal\ cost + C4 > Terminal\ cost + C5$

CHAPTER 5: VEHICLE LATERAL STABILITY CONTROL

5.1 Vehicle Model

Having seen the simulation results from the linear and non-linear system, we extend the same to the vehicle model. Here is the non-vehicle model considered as below,

$$m_v\left(\dot{V}_y + V_x r\right) = \left(F_{yfl} + F_{yffr}\right)\cos\delta_f + F_{yrl} + F_{yrr} + F_{yAFS}, \qquad (32)$$

$$I_z\dot{r} = l_f\left[\left(F_{yfl} + F_{yfr}\right)\cos\ \delta_f + F_{yAFS}\right] - l_r\left(F_{yyl} + F_{yrr}\right) \qquad (33)$$

$$+ l_s\left(F_{yfl} - F_{yfr}\right)\sin\delta_f,$$

where m_v, I_z, δ_f, V_x, V_y, and r are the vehicle mass, yaw moment of inertia, front steering angle, vehicle longitudinal velocity, lateral velocity, and yaw rate. l_s, l_f, and l_r are the wheel track, front wheelbase, and rear wheelbase, respectively. F_{yi} ($i = fl, fr, rl, rr$) are the lateral forces, which are calculated by 2D LuGre tire model, on four wheels, respectively. F_{yAFS} is the additional tire lateral force generated by the AFS control. [1]

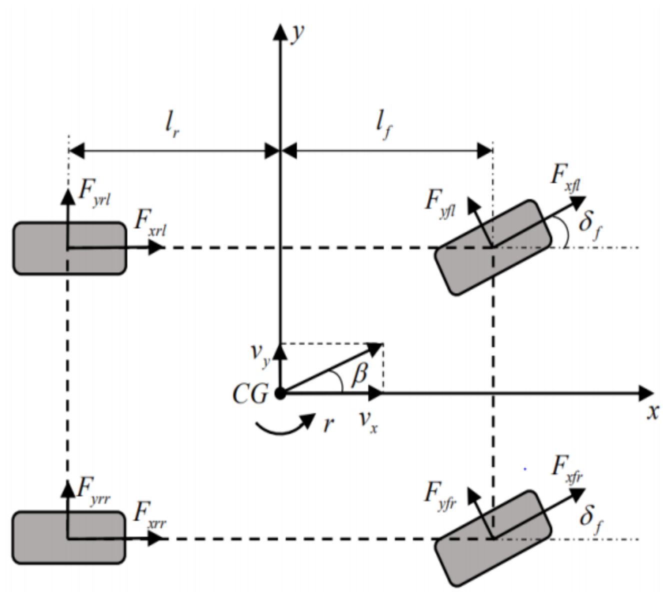

Figure 28 Lateral Vehicle Model

Tire models play a major role in the stability analysis. There are many models which has been discussed before and they follow different criteria according to the practicality of the applications. For example, magic formula tire model is a nonlinear model for the static tire with steady state data [22]. The 2D LuGre model is more suitable to real world applications & it is nonlinear and dynamic in nature which has better tire force characteristics [23]. Hence a 2D LuGre model has been considered. Stability region was generated by considering the tire and vehicle instabilities.

The stability region is presented in figure above. The boundaries are constructed by using the polytope function from the MATLAB as the data points are already defined. The yaw rate is the defined as a function of lateral velocity. [4]

$$\text{i.e. } r = f(V_y) \text{, where } V_{y-min} \leq V_y \leq V_{y-mid}$$

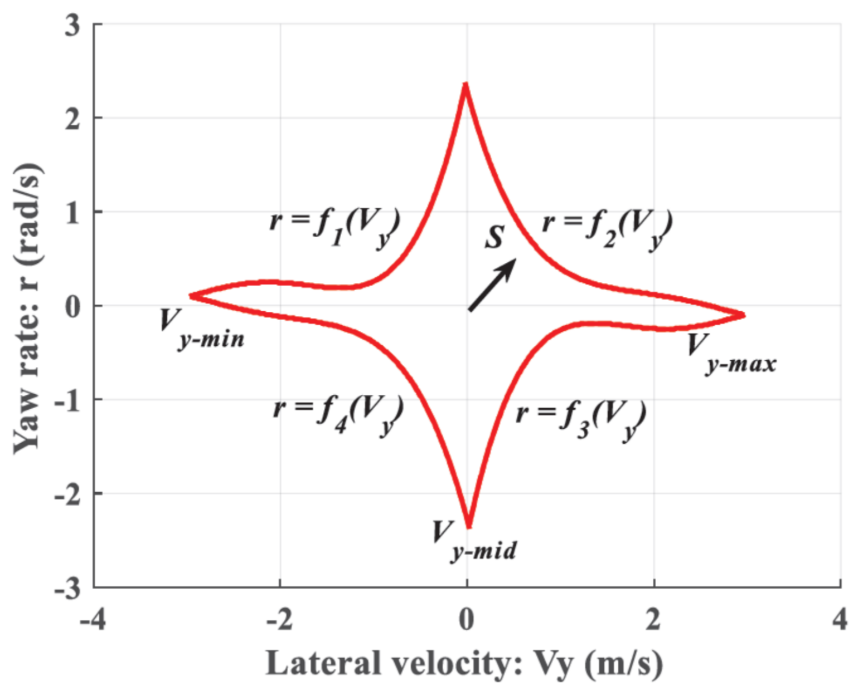

Figure 29 Stability Region [3]

The behavior of the stability region tends to enlarge with the varying longitudinal velocity V_x. But as the V_x remains constraint, the region gets shifted by vector $\vec{S}$ by the change in the steering angle δ_f. [3]The shifting vector, $\vec{S} = (V_{y-shift}, r_{shift})$, where the $V_{y-shift} = \frac{V_x l_r \delta_f}{l_f + l_r}$, $r_{shift} = \frac{V_x \delta_f}{l_f + l_r}$ makes the vehicle stability region real time implementable.

5.2 Test Environment

The constraint formulated as derived based on the previous study done on linear and non-linear systems. The 2012 Hatchback C class vehicle from the CarSim® database is simulated for the above the problem formulation with the following parameters as shown the table.

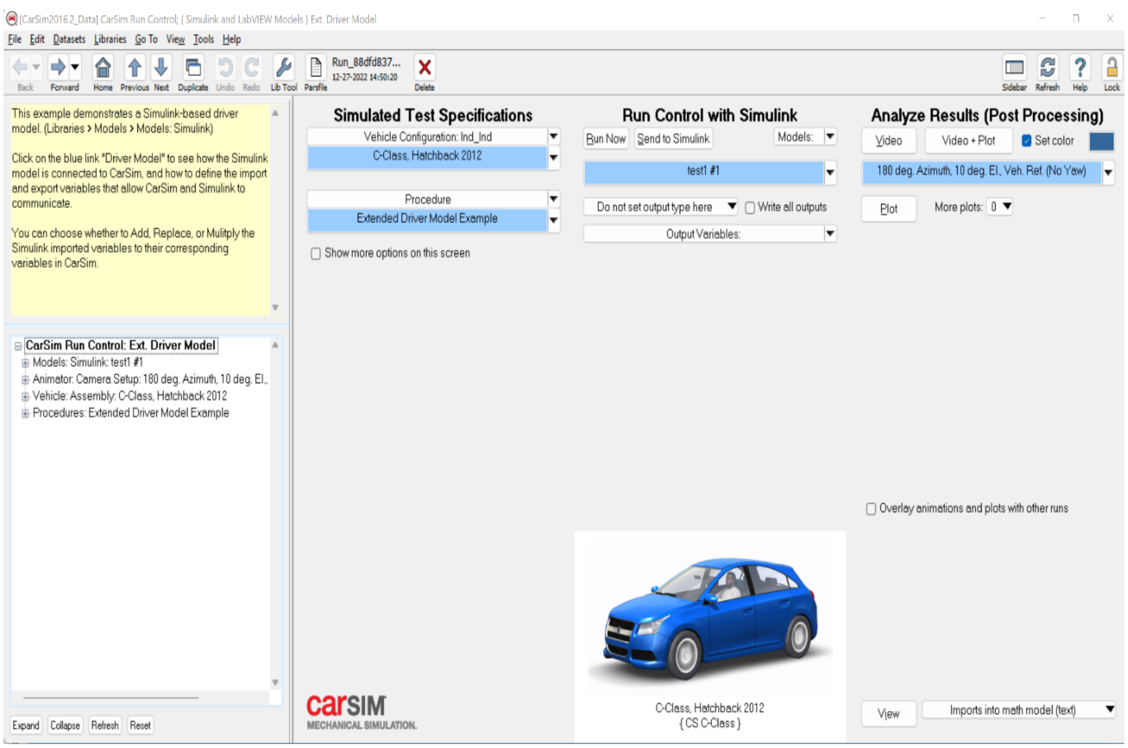

Figure 30 2012 Hatchback model setup [25]

Symbol	Parameter	Value
m_v	Vehicle Mass	1270 kg
g	Gravity Constant	9.8 m/s^2
I_z	Yaw Inertia	1536.7 $kg.m^2$
L	Wheelbase	2.91 m
l_f	Front Wheelbase	1.11m
l_f	Rear Wheelbase	1.8 m
l_s	Half of the vehicle track	0.835 m
T_s	Sampling Time	0.1 s
N & P	Prediction & Control Horizon	10

Table 4 Vehicle Parameters

The vehicle speed is fixed at 25 m/s. The steering angle is shown in Figure 4, in which the front wheel steering angle starts at zero and ramps up to 0.18 rad in 1.5 seconds and then keeps constant. MPC design and it is sent to the CarSim® vehicle as a input along with the steering profile externally. The experiment is carried out via the co-simulation using the MATLAB/Simulink & CarSim®. The inputs are generated by the external controller using the non-overshooting the vehicle states are obtained as the outputs from the model and fed back into the system for further computation.

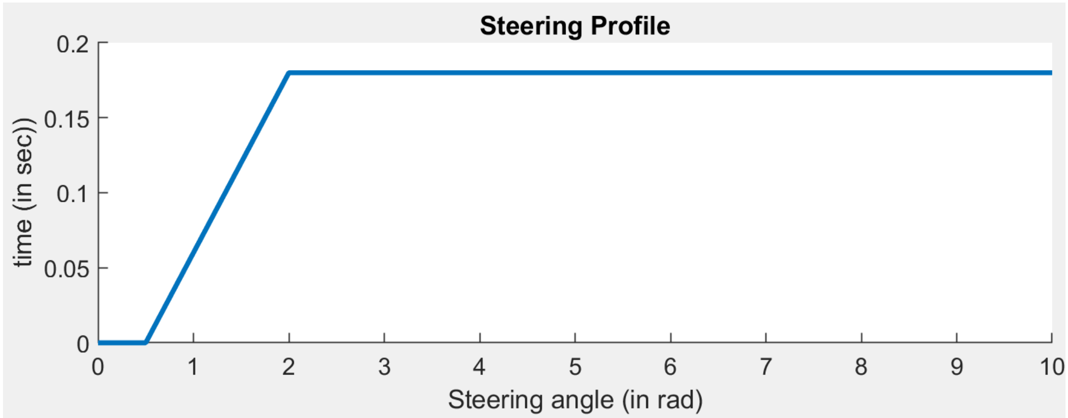

Figure 32 Steering Input in radians.

,

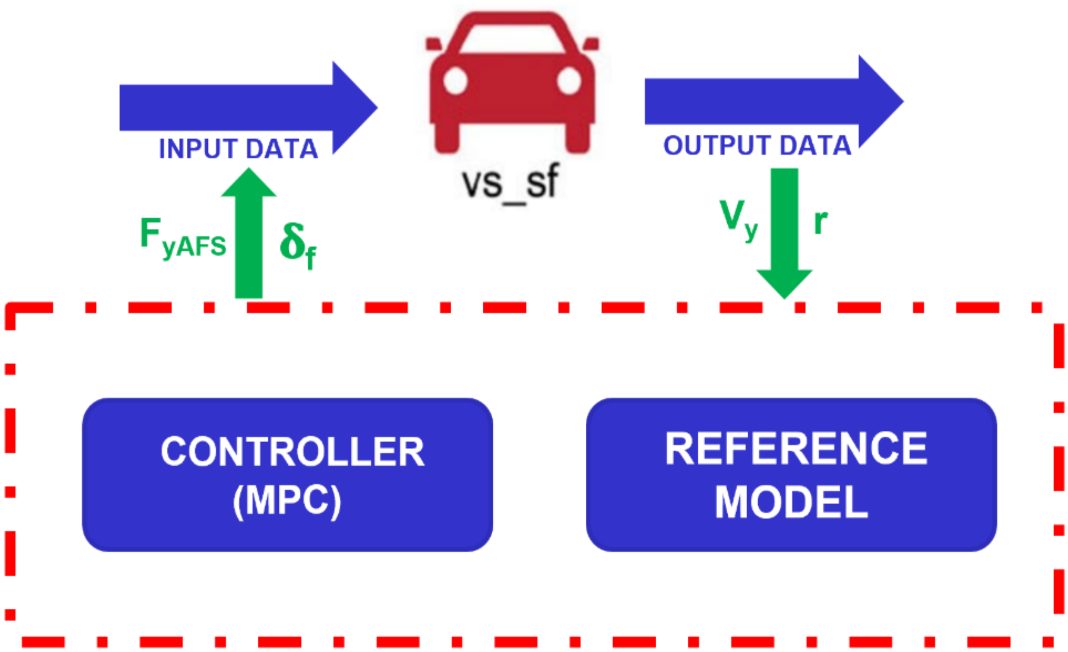

Figure 31 System Configuration with CarSim® at DSCL, ASU Poly

The same setup is done via Simulink as per Figure 33 Simulink Model As shown in the figure, the open loop function generates the steering angle. The lateral tire force due to the active front wheel steering angle is generated by the model predictive control algorithm using the non-overshooting design.

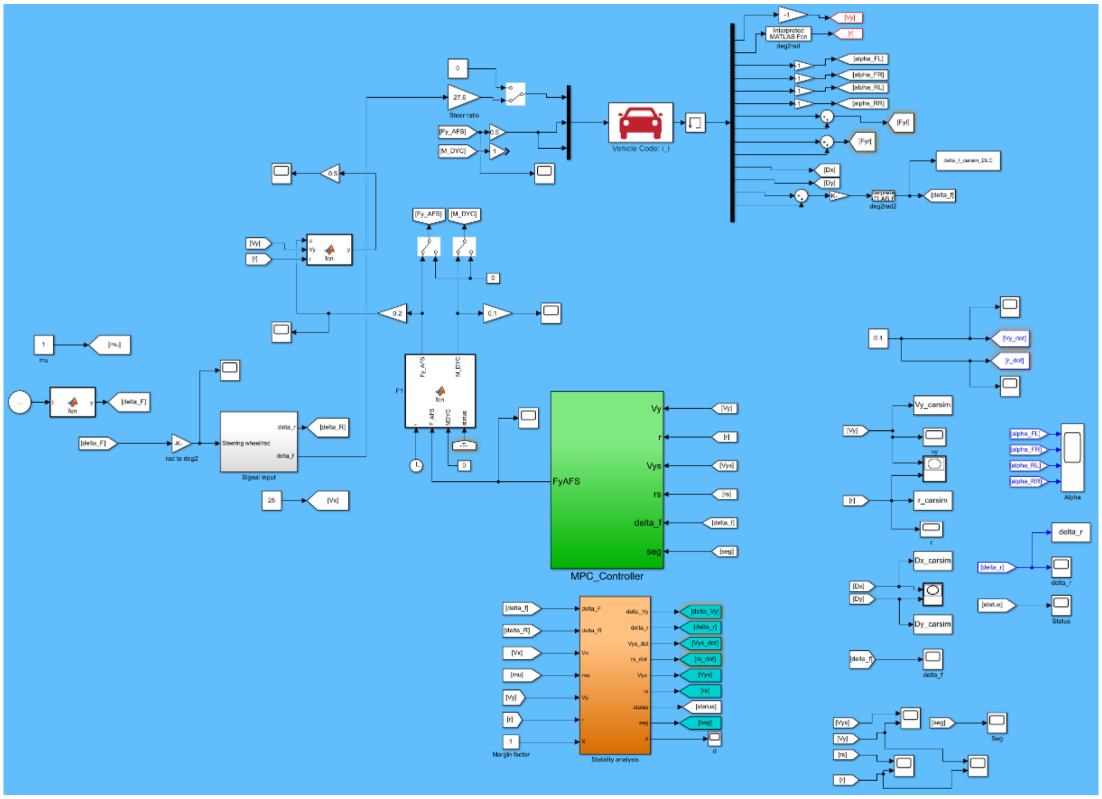

Figure 33 Simulink Model

We can see the shift in the vehicle trajectory from the CarSim result. The step steer and the lateral force cause the vehicle to move in the left direction, having a steering angle of 0.18 rad/s. This idea is implemented in different stages in the later sections of the thesis.

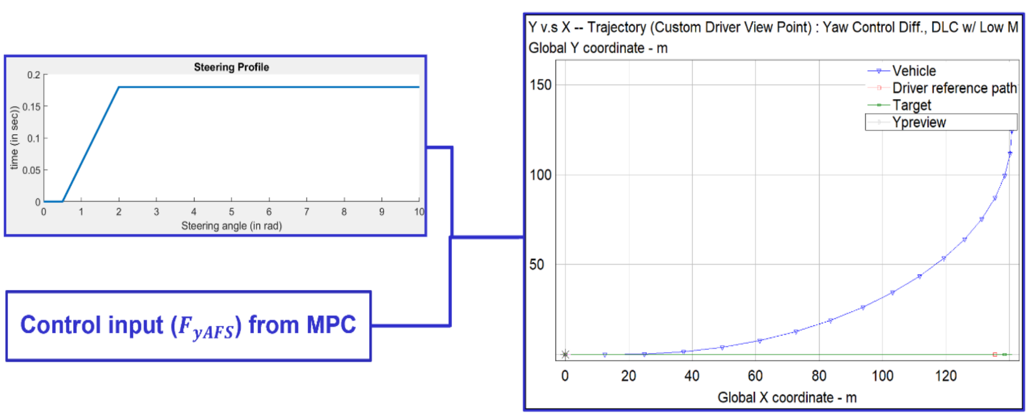

Figure 34 Vehicle Path With/Without Control

5.3 Non-overshooting MPC for Vehicle Stability Control

The control objective is to minimize the following cost function when the vehicle states are going to pass the stability boundary $x_s \rightarrow (V_{yS}, r_s)$,

$$J(k) = \sum_{j=1}^{N-1} [x(k+j \mid k) - x_s)^T])] * Q * [x(k+j \mid k) - x_s] + \sum_{j=1}^{P-1} \Delta u(k+j \mid k)^T * R * \Delta u(k+j \mid k) \quad (34)$$

Where $Q = [10 \quad 1; 1 \;\; 10]$, $R = 0.001$, N= P = 10 & $\Delta u(k+j \mid k) = F_{y-AFS}(k+j \mid k) - F_{y-AFS}(k+j-1 \mid k)$.

The implemented non-overshooting design is done in four stages by the mathematical formulation of stability region boundary. [3]

1. $r \leq f_1(V_y)$ & $V_y \in [V_{y-min}, V_{y-mid}]$
2. $r \leq f_2(V_y)$ & $V_y \in [V_{y-mid}, V_{y-max}]$
3. $r \leq f_3(V_y)$ & $V_y \in [V_{y-mid}, V_{y-max}]$
4. $r \leq f_4(V_y)$ & $V_y \in [V_{y-min}, V_{y-mid}]$

Among the ones defined , the fourth one is adopted to achieve the best control objective based on our previous discussions. We need to understand that the control is applied as per the change in segment numbers, and we try to present only the effects of specific cases in our following discussions

5.4 Simulation Results and Discussions

The aim of the analysis is to keep the vehicle states close the stability boundary as much as possible through the non-overshooting design MPC [3]. The continuous variations of the vehicle states and regions were monitored and verified in the simulation. The stability point in the boundary are found by the projection method discussed by Y.Chen, and Y. Huang [4][3]. This observation can be further verified by comparing the actual vehicle state with the closest point on the boundary for case 4 as below. The reason for simulating the design with C4 design is to see the response of the best non-overshooting design and how to take this forward.

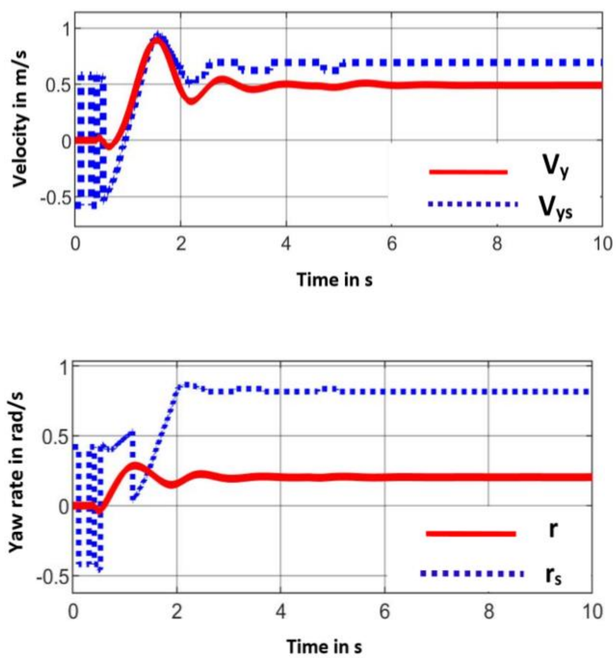

Figure 35 Vehicle States & Its Closest Boundary in C4 Design

As mentioned before , the aim of the simulation is to have the vehicle states closer to the stability boundary as much as possible. The vehicles states movement and its status are recorded below. This tells us the vehicle states has tried to overshoot the region and it was steered back due to the control formulation. However, this observation does not mean the vehicle states are out of the stability region.

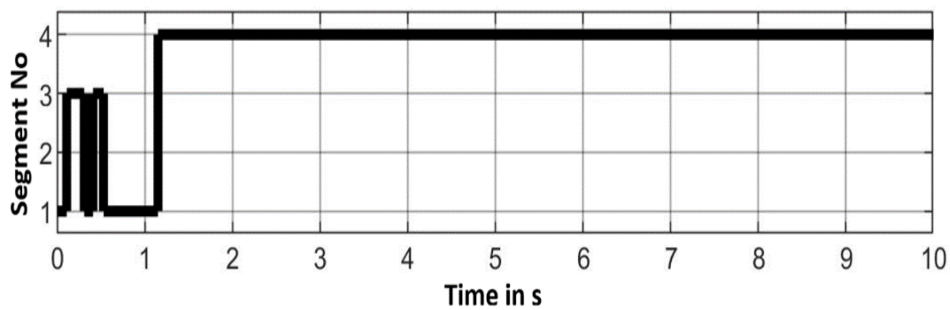

Figure 36 Segment Profile for C4

However, we tend to extend the proposed design in terms of C5 as we tend to get to have the feasible response with better convergence as in linear and the non- linear systems. From the idea derived before, we propose equality constraints at the terminal step of the horizon. The case 5 is proposed as : $r \leq f_4(V_y)$ & $V_y \in [V_{y-min}, V_{y-mid}]$ & with the $y_i(k+N|k) - y_{i-ref} = 0$ & $y_i(k+j|k) \leq y_i(k+j+1|k)$, where$1 \leq j \leq N-1$. The system is simulated as in figure 37 , and we observe the vehicle states to be closer to the boundary as the result of the terminal equality constraint. However, this approach is not trivial [6] [12]. We observe the vehicle states are much closer to the boundaries as per the conclusion in [3][4]. Also, the convergence and the oscillations are better than the in C4 design. This is due to the action of the terminal equality constraint (Otherwise called as stability constraint) that helps with convergence. The movement of vehicle states are presented in the figure below where the fluctuation in the stability boundary values is caused due to the movement of vehicle states.

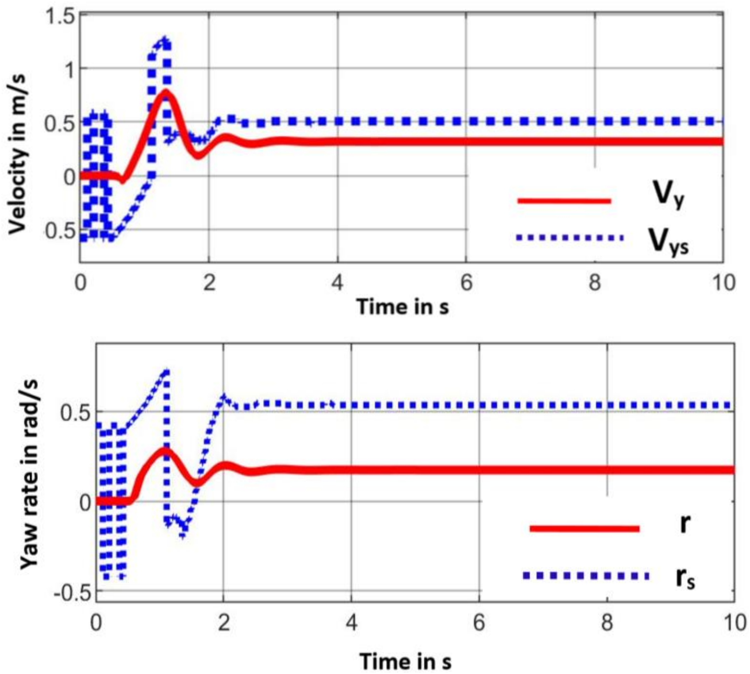

Figure 37 Vehicle States & Its Closest Boundary in C5 Design

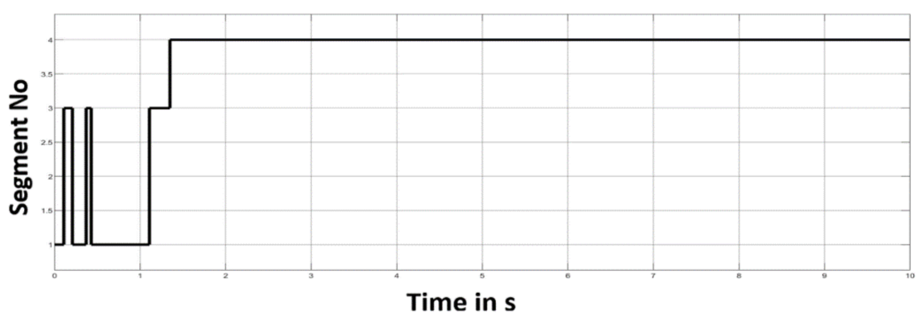

Figure 38 Segment Profile in C5 Design

The stability boundary when compared to the previous cases and thus we validate that the C5 has the best characteristic among all the non-overshooting design.

$$\boldsymbol{Convergence: C5 > C4 \left(wrt\ to\ stability\ point,\ x_s \rightarrow (V_{ys}, r_s)\right)}$$

$$\boldsymbol{Oscillation: C5 < C4 \left(wrt\ to\ state\ point,\ x \rightarrow (V_y, r\ )\right)}$$

Effect of terminal cost,

As discussed earlier ,the purpose of the terminal cost function is to bring stability to the system. While the stability proof can be evaluated theoretically for a linear system. The nonlinear systems need certain assumptions for the terminal penalty matrix used in the cost function. There is still on going to find a best way to arrive at a penalty matrix and As per authors like David Scramuzza , the terminal penalty matrix $\boldsymbol{Q_p}$ shall be equal to the $\boldsymbol{Q}$ which is used from $\mathbf{1}$ to $\boldsymbol{N-1}$ steps. [17] (2022)

The problem formulation using the terminal cost is as follows,

$$J(k) = \sum_{j=1}^{N-1}\{ [x(k+j \mid k) - x_s]^T * Q * [x(k+j \mid k) - x_s]\} + \Delta u(k+j \mid k)^T * R * \Delta u(k+j \mid k) + [x_n(k+j \mid k))^T])] * Q_p * [x_n(k+j \mid k) - x_s] \quad (35)$$

where $Q = [10 \quad 1; 1 \ \ 10]$, $R = 0.001$, N = P = 10 , $\Delta u(k+j \mid k) = F_{y-AFS}(k+j \mid k) - F_{y-AFS}(k+j-1 \mid k)$, $Q_p = Q$ is the terminal penalty matrix. The system is simulated with the presence of terminal cost on C5 as we concluded that the stability and feasibility ca be guaranteed by the addition of the terminal cost and terminal constraint. The response shows the states and its respective stability boundaries. From the segment plot down below, we can infer the movement of vehicle states within the segments along the boundaries. The associated advantages of terminal cost, which is Lyapunov function and terminal constraint brings the states closer to the boundary than the C5 design which is ideal necessity according to the application and research methodology presented.

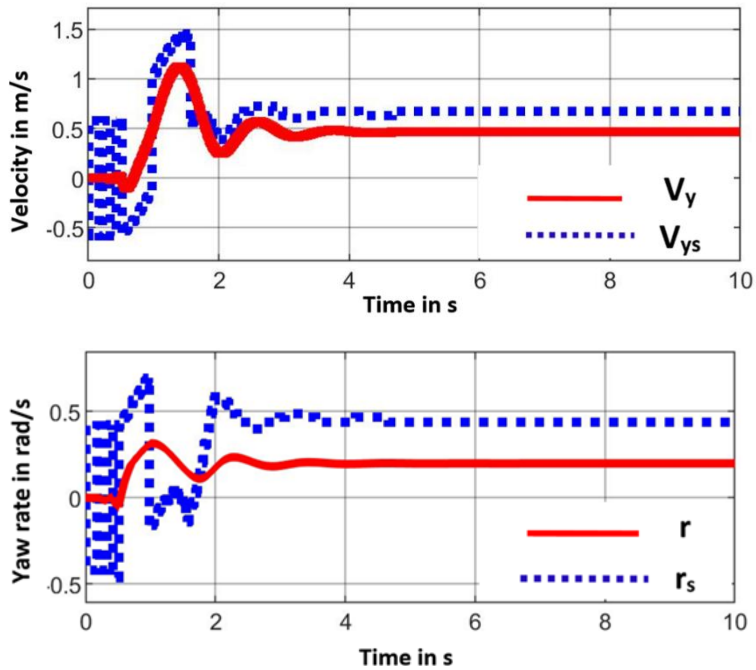

Figure 39 Vehicle States & Its Closest Boundary for Terminal Cost & C5

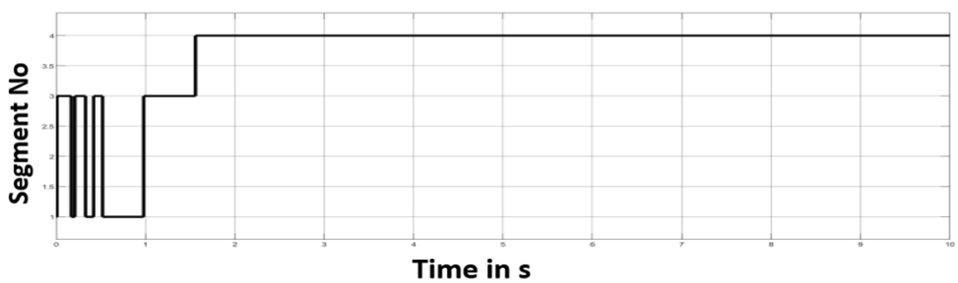

Figure 40 Segment Profile for the Design with Terminal Cost & C5

$$Convergence : Ter.Cost + C5 > C5 > C4$$

$$(wrt\ to\ stability\ point,\ x_s \rightarrow (V_{ys}, r_s))$$

$$Oscillation:\ C5 < Ter.Cost + C5 < C4$$

$$\left(wrt\ to\ state\ point,\ x \rightarrow (V_y, r\ )\right)$$

CHAPTER 6: CONCLUSION

The system's safety is essential and is the primary motivation behind this thesis [1]. In a nutshell, restricting the states within a pre-defined region ensures that the system/actuator does not reach its saturation value, which is very important for the system. This is a critical thing to ensure in emergency scenarios like docking of a boat, lane change, cornering in a tight turn, etc., and this thesis proposes a novel method for doing that. Non-overshooting model predictive control (MPC) ensures that the safety and theory of terminal cost and terminal constraints address its limitations. The existing non-overshooting control is fused with this theory, and a new design is proposed. Thus, benefiting from the proposed algorithm's non-overshooting characteristic, the vehicle's maneuverability is significantly improved in front wheel steering (FWS). The effectiveness of this algorithm is therefore checked in terms of its convergence through the simulation results from MATLAB/Simulink and CarSim® environment.

REFERENCES

1. NHTSA report on vehicle safety (https://www.nhtsa.gov/newer-cars-are-safer-cars).

2. Automated Vehicles for Safety, US Department of transportation. https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety

3. Yiwen Huang and Yan Chen, Stability Region based Vehicle Lateral Control Using Non-Overshooting MPC, 2019 American Control Conference (ACC) Philadelphia, PA, USA, July 10-12, 2019.

4. Y. Huang and Y. Chen, "Vehicle lateral motion control based on estimated stability regions," in Proceedings of the ASME Dynamic Systems and Control Conference (Invited Paper), 2017.

5. S. Inagaki, I. Kushiro, M. Yamamoto, "Analysis on vehicle stability in critical cornering using phase-plane method," JSAE Review, Vol. 16, No. 2, pp. 287-292, 1995.

6. F. Borrelli, A. Bemporad, M. Morari, Predictive Control for linear and hybrid systems, Date Oct 26, 2016.

7. James B. Rawlings , David Q. Mayne, Moritz M. Diehl, Model Predictive Control: Theory, Computation, and Design, 2nd Edition,

8. Basil Kouvaritakis, Mark Cannon, Model Predictive Control: Classical, Robust and Stochastic.

9. Keerthi, S. S., & Gilbert, E. G. (1988). Optimal, infinite horizon feedback laws for a general class of constrained discrete time systems: Stability and moving-horizon approximations. Journal of Optimization Theory and Application, 57, 265-293

10. Chen, C. C., & Shaw, L. (1982). On receding horizon feedback control. Automatica, 18, 349-352

11. Johan Löfberg, Feasibility analysis of MPC with ellipsoidal terminal state constraints, DOI June 14, 2000, Report no LiTH-ISY-R-2216.

12. Basil Kouvaritakis, Mark Cannon, Model Predictive Control: Classical, Robust and Stochastic.(Mark Cannon, MPC Lecture Notes, Oxford university.)

13. D.Q Mayne, J.B Rawlings, C.V Rao & P.O.M Scokert, Constrained model predictive control: Stability and optimality, Elsevier, Automatica 36 (2000) 789}814
14. Wei Zhang, Model Predictive Control: Theoretical Aspects Lecture Notes, Ohio State University

15. F. Borrelli, C. Jones, M. Morari, Model Predictive Control Algorithm, Feasibility and Stability, University of California Berkeley, ETH Zurich.

16. Dimitry Gorinevsky, Model Predictive Control, Stanford University

17. Sihao Sun, Angel Romero, Philipp Foehn, Elia Kaufmann & Davide Scaramuzza. "A Comparative Study of Nonlinear MPC and Differential-Flatness-Based Control for Quadrotor Agile Flight", IEEE TRANSACTIONS ON ROBOTICS, 2022.

18. SAE new standard: J3016, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems", SAE International, 2016

19. Yan Chen, EGR-560 Vehicle Dynamics and Control, 2022

20. MATLAB documentation (https://www.mathworks.com/help/matlab/)

21. D. Limon, T. Alamo, F. Salas, and E. F. Camacho, On the stability of constrained MPC without terminal constraint, IEEE transactions on automatic control, vol. 51, no. 5, May 2006.

22. H. B. Pacejka, and E. Bakker, "Magic formula tyre model," Vehicle System Dynamics, Vol. 21, pp. 1-18, 1993.

23. W. Liang, J. Medanic, and R. Ruhl, "Analytical dynamic tire model," Vehicle System Dynamics, Vol. 46, No. 3, pp. 197-227, 2008.

24. Michele Arnold, Rudy R. Negenborn, Goran Andersson, Bart De Schutter, Multi-Area Predictive Control for Combined Electricity and Natural Gas Systems, ETH Zurich

25. CarSim® Database and Simulation Environment, DSCL, ASU Poly Campus