Optimization based 2D platoon control with multiple behaviors implemented through a finite state machine

MSc Thesis Report

Shuhul Razdan

TU Delft

MSc Thesis Report

Optimization based 2D platoon control with multiple behaviors implemented through a finite state machine

by

Shuhul Razdan

to obtain the degree of Master of Science
in System and Control
at the Delft University of Technology,

Student number:	5049768	
Supervisors:	Ir. B. Kupers,	TNO-IVS
	Dr. ir. L. Ferranti,	TU Delft-CoR
	Dr. ir. S. Wahls,	TU Delft-DCSC

Delft University of Technology

TNO innovation for life

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abstract

With inland transportation increasing every passing day, vehicle platooning offers a good solution towards travelling more efficiently. Along with reducing traffic congestion on roads, platooning also leads to better fuel consumption among vehicles, fewer accidents, and most importantly, vehicle platoons can be made autonomous using optimization-based control techniques, such as Model Predictive Control (MPC). Much research has been done towards optimise fuel consumption through slip-streaming and by using topological data. A separate research field also looks into performing lanes changes and avoiding obstacles. However, both these research fields are disjointed and use a different model and MPC formulations to employ control. This project aims at developing an unified system that can implement longitudinal control (fuel optimisation), formation reconfiguration (lane changes) and collision avoidance using one model and MPC formulation. The platoon switches between these behaviours depending on the environment around the platoon. This report also highlights the limitations of the controller and gives concrete recommendations on how to deal with the shortcomings.

CHAPTER 1

INTRODUCTION

Inland freight transportation is becoming more important each passing day. Especially in Europe it is the primary way to carry large amount of goods across long distances. Inland freight transportation comprises of train, road and waterways transportation. Out of these three modes, road transportation is arguably the most extensively used and hence the most important. The importance is highlighted in Figure 1.1 [9].

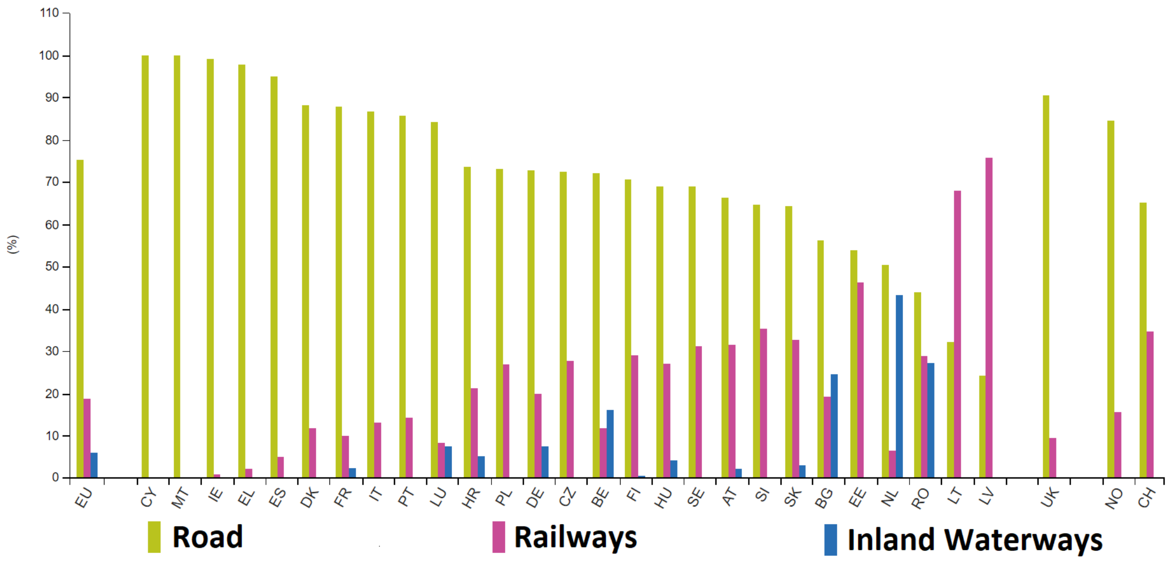

Figure 1.1: Modal Split of Inland Freight Transportation (EU), 2018 [9]

1.1 Why Platooning?

A lot of the road transportation can be done through platoons of trucks [30]. In transportation, platooning or flocking is a method for driving a group of vehicles together [50]. Platoons can be used mainly used to increase the capacity of the goods being transported at once. They have their advantages and disadvantages [20] as mentioned here:

Advantages of Vehicle Platooning

- Greater fuel economy due to reduced air resistance.
- Reduced congestion.
- Substantially shorter commutes during peak periods.
- Fewer traffic collisions.
- More comfort due to less braking and acceleration.
- More scope for automation resulting in less risk to human lives.

Disadvantages of Vehicle Platooning

- Some systems have failed in traffic, as they have been hacked by remote computers, creating a hazardous situation [42][14].
- Drivers would feel less in control of their driving, being at the hands of computer software or the lead driver.
- Drivers may be less attentive than usual, and they may not be able to react as quickly to adverse situations if the software or hardware were to fail.

As we see, the advantages are quite significant, especially when it comes to fuel-saving, comfort, and safety. Therefore, it is expected that platooning is a much more efficient way of freight transportation. Further, the disadvantages have also been made less significant with advances in fields such as cyber-security [37][10][7], control [45][30], and motion planning [11][12].

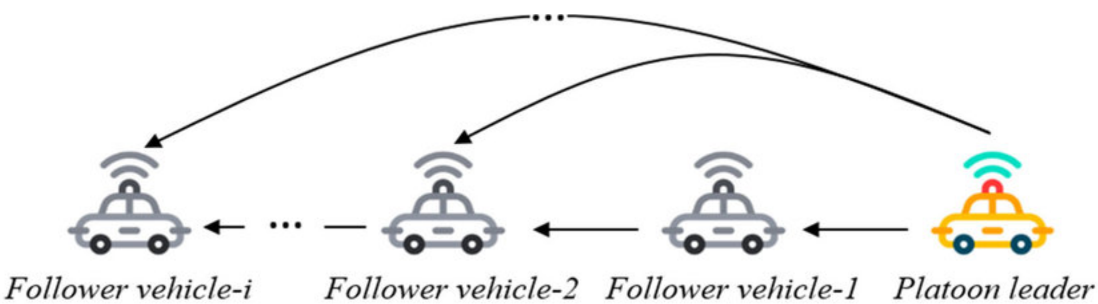

Figure 1.2: Representation of a platoon.

1.2 Autonomous Vehicles in Platooning

As mentioned above, one of the main advantages of platooning is that there is a great scope for the platoons to be partially or fully automated in the future [44]. There are several ways automating vehicle platoons and one of them is the optimization based approach which will be focused on in this report. Research has been carried on how platoons can navigate safely while optimising certain features [13][48] with the final goal being to make these platoon fully autonomous. These features include optimizing the amount of fuel (petrol vehicles) [28] or energy (EVs) that is required by the platoons to carry the goods from point A to point B. This can be achieved by allowing the vehicle behind the leading vehicle (see Fig 1.2) also known as follower vehicle (see Fig 1.2) to close up to a certain safe distance to the vehicle in front to reduce the drag force that they experience which leads to better fuel consumption. Optimisation algorithms also aim to reduce braking and consequent loss in fuel (energy in EVs) by using the data available to the algorithm regarding the future terrain that the platoon has to encounter. The vehicles make use of lifting and coasting [40][38] to optimize fuel consumption. This kind of optimisation is performed using *Longitudinal Control* of platoons. Extensive research has been carried out in this field, and it has been discussed in more detail in Section 3. The control can also be extended to other parameters such as safety [32] wherein we could optimise or constrain the distance between the vehicles, or we could also optimise comfort by putting constraints on the jerk and acceleration.

Longitudinal Control is an effective way of controlling a platoon or optimise a certain parameter (fuel, comfort, safety). However, this kind of control only deals with the platoon in the one dimension, i.e. in the longitudinal direction which is the direction in which the platoon is moving. As a result, it does not allow manoeuvres such as vehicle changing formations and order, changing lanes and avoiding obstacles. These manoeuvres might be less significant but are also very handy in scenarios wherein the platoon might need to navigate congestion. Therefore, we require the individual vehicles or the whole platoon to make lane changes that will need planning and control of lateral dynamics which will add another layer of complexity. The control in the second dimension is called *Lateral Control* and its combination with *Longitudinal Control* and *Collision Avoidance* is referred to as *2D Control* in this report.

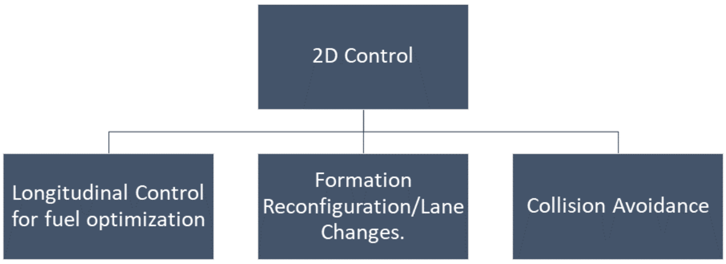

Figure 1.3: Breakdown of 2D Control

1.3 Related Work

This section will briefly discuss the literature on control of platoons to explain what the project is about. It will first highlight the work being done in Longitudinal Control and then in formation reconfiguration which encompasses the lateral manoeuvres such as lane changing. Then we shall conclude how they are similar or different, eventually leading to a thesis outline covered in this report.

1.3.1 Longitudinal Control

Longitudinal Control can be implemented for a platoon in several different ways. One way of employing control is using $\mathcal{L}_p$ String Stability [33]. Here, the platoon is treated as a string of vehicles and is modelled in one dimension. After modelling, the control algorithms aim to achieve specific tasks such as maintaining a target velocity for the vehicles using different gap policies [34] while minimising a norm function to maintain stability. The control architecture used is generally a variation of PID or, in some cases, LQR [1][18] which work along with a $\mathcal{L}_p$ constraint stabilisation to operate the platoon. Although computationally inexpensive, this method is also not the best as there can be no constraints on specific states. Also, it does not make use of the inter-vehicular gap to optimise for fuel consumption as it treats the platoon as a single block and therefore does not account for the distance between the vehicles. Further, this approach also does not do well when applied to non-linear systems and the results are not optimal. This is not desirable as the vehicle model used in this project will be non-linear.

To solve these problems associated with the String Stability approach, MPC can be used to control the platoon. The platoon is again modelled in 1D (no lateral dynamics), and this time inter-vehicular gap is also modelled and taken into account while optimising through a cost function to fulfil specific tasks, such as tracking references, maintaining safety, optimising fuel consumption, et cetera. There are several variations to how an MPC formulation can be designed to control platoons. The architecture can be centralised [22] wherein there is just one CPU in the platoon which has information of all the states of all trucks and then optimises over all parameters to generate optimal inputs. However, to implement a centralised architecture high grade communication modules are needed which can transmit the required information from the all trucks to the CPU and vice versa. Further, a powerful CPU is also needed to run the the extensive MPC problem. Having such a communication and computation in vehicles is not always practical and cost efficient. To fix this problem, the architecture could be changed to distributed [27][8] wherein each vehicle has its own CPU with only information about the states of the vehicles in its front and back. Here, the number of parameters over which the optimisation is done gets reduced, and the computation is essentially divided amongst several CPUs, leading to lesser computation time. The drawback sometimes is that now the CPU does not have information of other vehicles in the platoon (apart from the ones in its front and back) and it tries to estimate their states. This information might have some error which needs to be compensated for in some way.

1.3.2 Formation Reconfiguration and Collision Avoidance

Formation Reconfiguration and Lane Changes have been be implemented through model based approaches in literature by using a Kinematic Bicycle Model, which accounts for the lateral and longitudinal dynamics. One way of achieving reconfiguration is again to have a centralised [12] formulation of MPC and use the cost function to allow each of the vehicles in the platoon to have a reference lane or velocity. Again this formulation can be de-centralised [11] to reduce the computation associated. However, for a de-centralised architecture each vehicle does not have perfect knowledge of the states of other vehicles in the platoon and there needs to be a mechanism to account for the error in values of states.

Finally, the decision making can also be divided into a decision-maker, and a controller [46] wherein a separate layer determines the references for the controller. Then the MPC has only to follow the said references instead of generating them as well. In [24] one of the early approaches towards multi-robot control using MPC are discussed. Other de-centralised MPC approaches [5][39] employ alternating direction method of multipliers (ADMM), which leads to linearisation of collision avoidance constraints.

However, Kinematic Bicycle Model is not always enough as it assumes the slip angle of the vehicle to be zero which is not true in reality. Further, it cannot accommodate external forces acting on the vehicle into the the model. To deal with these concerns Dynamic Bicycle Model is used [35]. The work done in [23] shows how the the dynamic model can be used to perform lane changes and the model used in this project will be similar and work broadly on the same principles. The literature also goes into more detail while taking into account practical considerations such as lagged tire forces [6] and road topology [49]. Kinematic and Dynamic Bicycle model have many variations [25] based on different requirements and thus there are many options when it comes to modelling.

1.4 Conclusions from Related Work

After going through the literature, it became evident that research on Longitudinal Control and Reconfiguration & Avoidance is very disjointed. The reasons behind it are

1. Both approaches in the literature use different models to solve the problem wherein Longitudinal Control only encompasses the longitudinal dynamics. In contrast, the research in reconfiguration predominantly uses a Bicycle Model, which is different from a longitudinal model and focuses primaily on steering dynamics. The kinematic model does not incorporate external forces on the vehicle and thus goes farther away from the longitudinal model wherein aerodynamic, terrain and friction forces are accommodated. This problem can be dealt with by using a dynamic model but that also focuses more on lateral and tire dynamics in the literature.

2. Due to the differences in model formulation and vastly different requirements from both research fields, the formulation for the MPC is also vastly different. Longitudinal Control focuses more on conserving fuel, maintaining a gap, et cetera. Whereas during lanes changes and avoidance, these parameters are not essential, and instead, the MPC formulation focuses on safety and smooth tracking of lateral position.

1.5 Research Question

The main aim of the thesis is

- To build an end to end system which can implement several behaviours for a platoon, that are Longitudinal Control, Formation Reconfiguration and Collision Avoidance. The report will also investigate challenges one might face while building such a system and will also clearly highlight the limitations of the system. Alternatives and recommendations will also be proposed with regards to dealing with the limitations.

- Additionally, the platoon should be able to decide when to switch between these behaviours depending on the environment around the platoon.

- The tasks mentioned above should be achieved using a unified model and MPC formulation.

This is represented in Fig. 1.4.

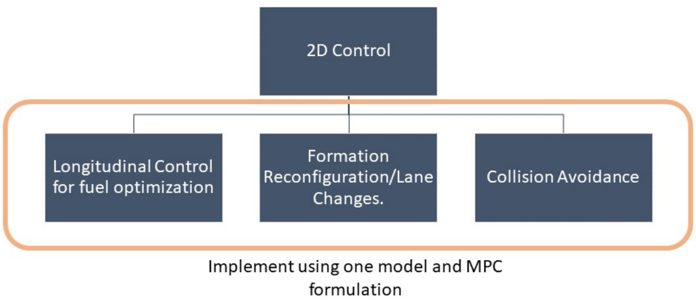

Figure 1.4: Research Question Representation

To generate such a system the following research questions will be answered.

1. Firstly, the question that will be dealt with will correspond to what model should be used for 2D Control? The model needs to incorporate all the dynamics related to 2D Control. To answer this question, first a system is designed achieve good results for Longitudinal Control in Chapter 3. Two approaches towards modelling will be discussed and it will be investigated why one is better than the other through simulation results. After developing the Longitudinal Control mechanism, the report will showcase the 2D model formulation in Chapter 4 which will combine Longitudinal forces (aerodynamic, terrain and friction) along with the lateral dynamics of a vehicle which are necessary for Reconfiguration and Avoidance behaviors. This will be done by appending the lateral dynamics in Dynamic Bicycle model with longitudinal states and forces. This way we will deal with the Point 1 from the conclusions above.

2. After answering the first research question, the next step will be to answer the question of how control will be applied to such a model using one MPC formulation? This will be achieved by having separate modes within the MPC running cost. Each mode will correspond to certain behaviors and the formulation will allow the platoon to switch between the modes depending on the environment around the platoon. The state machine to determine this switching will also be defined which will contain the conditions under which switching will take place. This will address Point 2 from the conclusions.

3. The thesis report will also answer the question regarding the effects of different tuning weights on the performance of platoon. Different modes of the platoon will be designed by having separate tuning weights for each behaviour. The reason for choosing these weights will be discussed and their effects will also be shown in simulations.

The performance of this new model and MPC formulation will be tested through several test cases and conclusions drawn from them will also be discussed in detail. The model formulation will be based on the literature on the dynamic bicycle model and longitudinal control and MPC formulation will be developed by borrowing ideas from the literature on all the fields mentioned above.

.

CHAPTER 2

PRELIMINARIES

2.1 Model Description

Before getting to how the control is employed on a vehicle it is essential first to choose what model to use to define the vehicle. It determines the state and dynamical equations and plays a vital role in defining constraints and cost functions. A model with fewer constraints is desirable as less computational power will be required to control it as the control algorithm will optimize over lesser variables.
However, sometimes a simpler model may not encompass all the required dynamics of a vehicle, resulting in sub-optimal performance. Models with higher complexity and more elaborate dynamics can easily solve this problem but, in turn, require more elaborate constraints to be defined during control. Thus, computation complexity is higher, and so is the time required for obtaining optimal control strategies. Therefore, it is best to choose a simple enough model in practice such that it encompasses not all but the significant dynamics of the system, thus resulting in a good balance.
In a platoon, the model for each vehicle needs to be defined which includes the states and inputs for each vehicle. These models could be identical (*homogeneous platoons*) or different (*heterogeneous platoons*). We assume that our platoon is *homogeneous* and we use the non-linear dynamic *Bicycle Model* to defined the model for each vehicle. To understand the Dynamical Bicycle Model better, the Kinematic Model has to be explained first.

2.1.1 Kinematic Bicycle Model

The model has been used used in several cases and has has given good results when it comes to encompassing the necessary dynamics of a four wheeled vehicle and it also gives good optimality results when it comes to control and motion planning [11]. The model essentially reduces a four wheeled vehicle or robot into a two wheeled bicycle like structure with longitudinal and lateral (steering) constraints. The model and its kinematic equations are provided below.

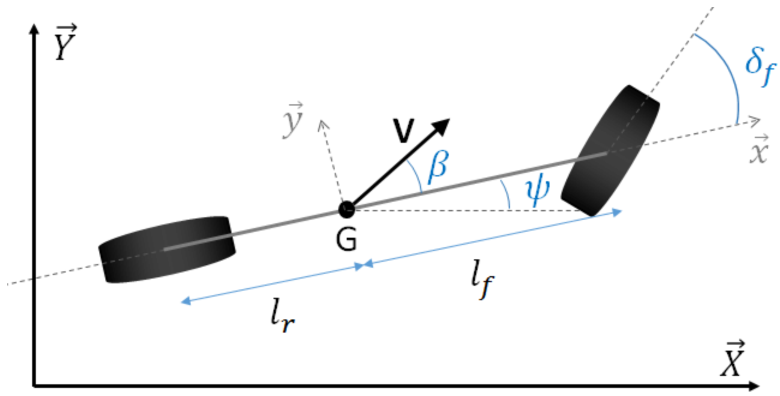

Figure 2.1: Non-linear Kinematic Bicycle Model

$$
\begin{aligned}
\dot{x} &= v\cos(\psi+\beta) \\
\dot{y} &= v\sin(\psi+\beta) \\
\dot{\psi} &= \frac{v\cos\beta}{l_f+l_r}(\tan\delta) \\
\dot{v} &= a
\end{aligned}
\tag{2.1}
$$

[11] where the state vector shall be $\mathbf{z} = [x, y, \psi, v]^T$ (x, y, ψ, and v are the longitudinal and the lateral position of the vehicle in global co-ordinates, the heading angle, and the velocity, respectively. The heading angle is calculated w.r.t the horizontal X axis. The control input vector is $u = [a, \delta]^T$ (a and δ are the acceleration and the steering angle, respectively), $\beta := \arctan\left(\tan\delta\left(\frac{l_r}{l_f+l_r}\right)\right)$ is the side slip angle, and l_f , l_r are the distance from the center of gravity to the front and rear axles, respectively. Using Euler discretization the model given above is discretized with a sample time Δt as

$$
\begin{aligned}
x(k+1) &= x(k) + \Delta t\, v(k)\cos(\psi(k)+\beta(k)) \\
y(k+1) &= y(k) + \Delta t\, v(k)\sin(\psi(k)+\beta(k)) \\
\psi(k+1) &= \psi(k) + \Delta t\, \frac{v(k)\cos\beta(k)}{l_f+l_r}(\tan\delta(k)) \\
v(k+1) &= v(k) + \Delta t\, a(k)
\end{aligned}
\tag{2.2}
$$

2.1.2 Dynamic Bicycle Model

At higher vehicle speeds, the assumption that the velocity at each wheel is in the direction of the wheel can no longer be made. This means that the slip angle (angle between velocity direction and wheel direction) is not zero. In this case, instead of a kinematic model, a dynamic model for lateral vehicle motion must be developed. The derivation for the state equations for the dynamic model is borrowed from [35].

The dynamics of the bicycle model with two degrees of freedom, namely lateral position y and the yaw angle Ψ is shown in Fig. 2.2. The lateral position is measured along the lateral vehicle axis to the COG of the vehicle, i.e. point **O**. The yaw angle ψ is measured w.r.t to the global **X** axis. The longitudinal velocity of COG is denoted by V_x.

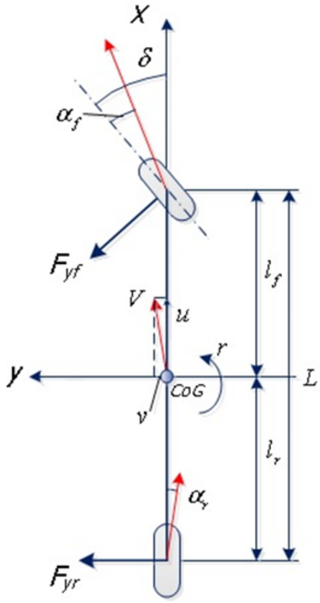

Figure 2.2: Dynamic Bicycle Model

The state space derived from the following dynamics is as follows [35]:

$$\frac{d}{dt}\begin{Bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & 0 & -V_x - \frac{2C_{\alpha f}\ell_f - 2C_{\alpha r}\ell_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2\ell_f C_{\alpha f} - 2\ell_r C_{\alpha r}}{I_z V_x} & 0 & -\frac{2{\ell_f}^2 C_{\alpha f} + 2{\ell_r}^2 C_{\alpha r}}{I_z V_x} \end{bmatrix} + \begin{Bmatrix} \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2\ell_f C_{\alpha f}}{I_z} \\ 0 \end{Bmatrix} \delta \tag{2.3}$$

The above state equations are enough to encompass the lateral dynamics required in this thesis project and will be used later. The model stated above is linear w.r.t the input δ which is the steering angle. However, the model used for the actual vehicles will have some addition to the states and will assume different inputs, e.g. V_x will no longer be constant and will be used as a state. The input will be the steering rate ϕ instead of the steering angle δ resulting in a very non-linear model for which a non-linear MPC solver will be used. All these changes will be discussed in Chapter 4.

2.2 Vehicle Control

Control can be carried out through several different control methods. The most basic control strategies are PID [51], [3]. If we go into more advanced controllers we see state-feedback [15] and Linear Quadratic Regulator (LQR) [30], [4]. LQR control method is very useful for continuous-time problems as it is optimal w.r.t an infinite horizon.

However, in reality no control problem is the infinite horizon as in reality control is applied for certain fixed amount of time. Thus. *Model Predictive Control* (MPC) is used to give better results for finite-horizon problems for non-linear systems [30], [21], [19]. MPC allows us to impose constraints on the optimization problem. MPC is also known as *Receding Horizon Control* (RHC) [47] as it involves repeatedly solving a constrained optimization problem, using predictions of future costs, disturbances, and constraints over a moving time horizon to choose the control action. RHC handles constraints, such as limits on control variables, directly and naturally, and generates sophisticated feed-forward actions. MPC also does better than other control methods when it comes to having non-linear models which will be used in this report.

2.2.1 Model Predictive Control

MPC optimizes a certain *cost function* $(J(x(k)))$ under some state or input constraints. The cost also comprises of running cost and terminal cost. Terminal cost is generally used to provide further stability whenever constraints are involved it is used to approximate the tail of the infinite horizon. A general MPC problem is formulated as [30]

$$\begin{aligned} J^*(x(t)) = \underset{\{u_k\}_{k=0}^{N-1}}{\text{minimize}} &\left\{ \sum_{k=0}^{N-1} l\left(x_k, u_k\right) + \overbrace{l_f\left(x_N\right)}^{\text{terminal cost}} \right\} \\ \text{subject to } & x_0 = x(t) \\ & \text{for } k = 0, \ldots, N-1 : \\ & x_{k+1} = f\left(x_k, u_k\right) \\ & x_k \in \mathcal{X} \\ & u_k \in \mathcal{U} \\ & x_N \in \mathcal{X}_f \text{ (terminal constraint)} \end{aligned} \tag{2.4}$$

where $l(x, u)$ is the running cost, x_0 is the state at time t. The next constraint is related to the dynamics of the system to ensure they are always followed. $\mathcal{X}$ and $\mathcal{U}$ are set of admissible states and inputs, respectively. The terminal constraint on x_N is introduced to ensure stability, and set $\mathcal{X}_f$ is generally chosen to be a subset of the level sets of l_f, which is in turn chosen as the solution to the infinite horizon Riccati Equation. $\mathcal{X}_f$ also needs to be chosen in such a way that the states and inputs do not exceed the set of admissible quantities. All the above statements are carefully derived using sophisticated mathematical analysis and are explained in [36].

2.2.2 Interior Point Method

The MPC formulation in this project will be non-linear due to the non-linear models and constraints. Therefore, the solver involved will also need a non-linear optimization algorithm. The algorithm used here is the Interior Point Method. The interior-point (IP) method for non-linear programming was pioneered by Anthony V. Fiacco and Garth P. McCormick in the early 1960s. The basis of IP method restricts the constraints into the objective function by creating a barrier function. This limits potential solutions to iterate in only the feasible region, resulting in an efficient algorithm regarding time complexity.

To ensure the program remains within the feasible region, a penalty parameter, μ, is added to "penalize" close approaches to the boundaries. This approach is analogous to the use of an invisible fence to keep dogs in an unfenced yard. As the dog moves closer to the boundaries, the more shock he will feel. In the case of the IP method, the amount of shock is determined by μ. A large value of μ gives the analytic centre of the feasible region. As μ decreases and approaches 0, the optimal value is calculated by tracing out a central path. A smooth curve is generated for the central path with small incremental decreases in μ during each iteration. This method is accurate but time-consuming and computationally intense. Instead, Newton's method is often used to approximate the central path for non-linear programming. Using one Newton step to estimate each decrease in μ for each iteration, a polynomial time complexity is achieved, resulting in a small zig-zag central path that convergences to the optimal solution.

The logarithmic barrier function is based on the logarithmic interior function:

$$B(x,\mu) = f(x) - \mu \log(x) = f(x) - \mu \sum_{i=1}^{m} \ln(x_i)$$

The IP method for NLP has been commonly used to solve Optimal Power Flow (OPF) problems. A set of non-linear equations is used to find the optimal solution of a power network in terms of speed and reliability. To solve these problems, the perturbation factor is used in addition to the typical Karush-Kuhn-Tucker (KKT) methods.

Starting with a general optimization problem:

$$\begin{aligned} &\min f(x) \\ \text{s.t. } &h(x) = 0 \\ &g(x) \leq 0 \end{aligned}$$

Modify the KKT conditions by adding convergence properties with slack variables and the perturbation factor:

$$\begin{aligned} &\nabla_x L(x, \lambda_h, \lambda_g) = 0h(x) = 0 \\ &g(s) + s = 0 \\ &[\lambda_g]\, s - \mu e = 0 \\ &(s, \lambda_g, \mu) \geq 0 \end{aligned}$$

Solve the nonlinear equations iteratively by Newton's methods. First determine Δx and $\Delta\lambda_h$ with reduced linear equations. Next, calculate slack variables and corresponding multipliers with: $\Delta s = -g(x) - s - \nabla g(x)\Delta x \ \Delta\lambda_g = -\lambda_g + \left[s^{-1}\right] * \mu e - [\lambda_g]\, \Delta s$

To calculate the perturbation factor, μ, use primal-dual distances:

$$\mu = \sigma * \text{pdad} = \sigma * \frac{\lambda_g^t s}{niq}$$

where σ defines the trajectory of the optimal solution, pdad is the primal-dual average distance, and niq accounts for the inequality constraints.

σ might assume the following values:

- $\sigma = 0$ corresponds to the affine-scaling direction where the optimal point is obtained through a non-perturbed solution of KKT
- $\sigma = 1$ corresponds to centralization direction where the non-optimal solution is found with a primal-dual distance equal to the initial value of μ

In a conventional primal-dual IP method, a constant value is assigned to σ (usually close to 0.1) for the iterations. This results in a search direction where 90% is defined towards the optimal point and 10% is allocated to trajectory of centralization.

.

CHAPTER 3

LONGITUDINAL CONTROL

Longitudinal Control as the name suggests deals with the control of vehicles or platoons of vehicles in a single dimension. It implies that planning and control are only performed longitudinally. Therefore, the platoon order that is the order in which the vehicles are moving in (relevant for heterogeneous platoons) is fixed and cannot be changed. As a result, the platoon can be controlled jointly, or each vehicle can be controlled separately. The joint approach [33] is less expensive computationally but is sub-optimal (as it does not make use of the inter-vehicular distance between the vehicles in the platoon), and vice versa is true for the case wherein each vehicle is controlled separately. The architecture of control can also be hierarchical [46] or just a single layer [22].

In this chapter, we discuss a centralised Eco-MPC approach that aims at maximising the fuel economy of a truck platoon by making use of lifting and coating techniques [40][38]. The platoon will already know the future terrain it will encounter, and thus the algorithm comes up with control strategies to employ less braking and acceleration to conserve fuel. The information regarding the future can be available through maps or sensors. The algorithm also tries to maintain a small inter-vehicular distance between the vehicles in the platoon to make use of the drag reduction caused by slip-streaming behind a vehicle in front [29][43]. Along with maintaining a desired gap between the vehicles, the vehicles aim to track a certain velocity provided by the user. In the centralised approach, the central computational unit is placed in one of the trucks in the platoon (generally the leading vehicle). This unit takes information from all the trucks regarding their current state, uses MPC to solve the problem and then sends back the desirable control inputs to achieve the goals assigned through the cost function.

In Section 3.1 we will discuss a centralised Eco-MPC approach wherein the control input is the acceleration to the vehicle. The section first discusses how the MPC problem was formulated and discusses the simulation results and effect of fuel conservation weights. Section 3.2 improves on the notion of Acceleration Control by now using jerk, i.e. the rate of change of acceleration as the input instead of the acceleration itself. Like Section 3.1 the problem formulation will also be presented here, and the simulation results will also be discussed. Finally, Section 3.3 will present the conclusions drawn from the research presented in this chapter and will also suggest some further improvements that could be made.

3.1 Acceleration Control

As mentioned above, in this section we will discuss the approach wherein we will use acceleration as an input to the platoon vehicles. The section is based on [22] compared to which changes have been made to dynamics of the system and also how the aerodynamic forces affect the vehicle.

3.1.1 Assumptions

The sensors on the vehicles are assumed to be fault free that is, the CPU receives perfect information about the states of the vehicles. Also, we assume that the communication systems between the CPU and vehicles are very fast and thus lead to no delay in communication which needs to be accounted for. Finally, we also assume there are no actuator faults and all the control actions are implemented perfectly on each vehicle. **These assumptions carry on to all sections and simulations in the report.**

3.1.2 Modelling of Platoon

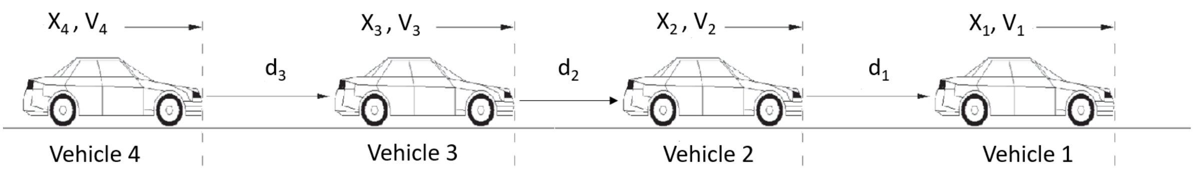

Figure 3.1: Platoon Model - Acceleration Control

The first in modelling the platoon and coming up with equations to model certain parameters is to decide what the states for our model will be. For our model the states and inputs are defined as

$$z_i = \begin{bmatrix} x_i & v_i \end{bmatrix}^T \tag{3.1}$$

$$u_i = a_i \tag{3.2}$$

where subscript i is an integer which indicates the i -th vehicle ($i = 1, 2, 3, 4, ...$) counted from the leading vehicle in the group. x_i refers to the position of the vehicle i (m), v_i is the velocity of the vehicle i ($\frac{m}{s}$) and similarly a_i signifies the acceleration of each vehicle ($\frac{m}{s^2}$). Further, the dynamics of the platoon can bee defined as

$$\begin{aligned} m_i\dot{v}_i(t) =& -f_{ai}(t) - f_{gi}(t) - f_{\mu i} + m_i u_i(t) \\ f_{ai}(t) =& \frac{1}{2}\rho A_i C_i v_i^2(t) \\ f_{gi}(t) =& m_i g \sin\theta\left(x_i(t)\right) \\ f_{\mu i}(t) =& \mu_i m_i g \\ d_i(t) =& x_i(t) - x_{i+1}(t) - l_i \end{aligned} \tag{3.3}$$

Parameters ρ, g, l_1, m_i, A_i and μ_i are the air density $(\frac{kg}{m^3})$, the gravity acceleration $(\frac{m}{s^2})$, the length of the preceding vehicle (m), the mass (kg), the projected frontal area (m^2) and the coefficient of all the other friction resistance of the i -th vehicle, respectively and variable d_i signifies the spacing between the vehicles (m). Functions f_{ai}, f_{gi} and $f_{\mu i}$ denote the aerodynamic drag, the grade resistance and all the other friction resistance of the i -th vehicle, respectively. $\theta\left(x_i\right)$ is the road gradient that depends on the location of the i -th vehicle and needs to be known beforehand. It will be discussed in this section and the next section. C_i refers to the drag coefficient of the i^{th} vehicle, and as seen in the equation, it is directly proportional to the force f_{ai} experienced by the vehicle. It is the highest for the vehicle in front as it is not following any vehicle. Subsequently, the coefficient for Vehicle 2 is lower, and it keeps decreasing as we go on due to the additive effect of drag reduction. Here, a straightforward model for drag is designed wherein each vehicle has a fixed coefficient based on its position in the platoon. These values are listed in the table below. In reality, more complex drag models are used wherein the coefficient depends on the spacing between the vehicles as well.

Co-Efficient	**Value**
$C1$	0.3
$C2$	0.275
$C3$	0.25
$C4$	0.2

Table 3.1: Drag Co-efficient Values

Similarly, we also design the road gradient $\theta\left(x_i\right)$ which will be depended on the position of the vehicle i. The equation to model it is defined as

$$\begin{aligned} \theta\left(x_i(t)\right) =& \hat{\theta}\left(\frac{1}{1+e^{s_1(x_a - x_i(t))}} - \frac{2}{1+e^{s_1(x_b - x_i(t))}}\right. \\ & + \frac{2}{1+e^{s_1(x_c - x_i(t))}} - \frac{2}{1+e^{s_1(x_d - x_i(t))}} \\ & \left. + \frac{1}{1+e^{s_1(x_e - x_i(t))}}\right), \end{aligned} \tag{3.4}$$

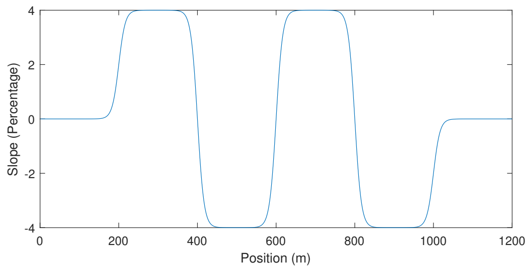

Figure 3.2: Slope Profile

and the values for the several variables is listed below and the profile w.r.t position is shown in 3.2

$\hat{\theta}$	0.04	x_c	600
s_1	0.12	x_d	800
x_a	200	x_e	1000
x_b	400	-	-

Table 3.2: Slope design variable

After defining all the above parameters the state space equations for 3 trucks can be described as follows

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ x(t) &= \begin{bmatrix} x_1 & v_1 & x_2 & v_2 & x_3 & v_3 \end{bmatrix}^T \\ u(t) &= \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T \end{aligned} \tag{3.5}$$

$$f(x(t), u(t)) = \begin{bmatrix} v_1 \\ -\frac{1}{m_1}(f_{a1} + f_{g1} + f_{\mu 1}) + u_1 \\ v_2 \\ -\frac{1}{m_2}(f_{a2} + f_{g2} + f_{\mu 2}) + u_2 \\ v_3 \\ -\frac{1}{m_3}(f_{a3} + f_{g3} + f_{\mu 3}) + u_3 \end{bmatrix} \tag{3.6}$$

3.1.3 MPC Formulation

For the platoon to work as desired, it needs to fulfil the following objectives:

1. The desired velocity is maintained;
2. The desired gap between the vehicle needs to be maintained;
3. Any collisions should be avoided;
4. The fuel consumption at the reference velocity and headway gap; needs to be minimised.
5. The acceleration should be moderated.

The objectives are taken into account through the cost function. The running cost consists of multiple elements associated to each of the objectives above. The running cost J^{MPC} is defined as

$$J^{\text{MPC}}(z_i, u_i, t) = \omega_v L_v(t) + \omega_d L_d(t) + \omega_u L_u(t) + \omega_a L_a(t) \tag{3.7}$$

where L_v, L_d, L_u and L_a are parts of the running cost function responsible for fulfilling separate tasks which were defined above. Further ω_v, ω_d, ω_u and ω_a are the weighing coefficients which are used put more or less emphasis on a particular task. Next, it is discussed how these parts help in fulfilling the objectives.

For velocity tracking, the cost function is

$$L_v(t) = \frac{1}{2}\left(v_1(t) - v_{ref}\right)^2 + \frac{1}{2}\left(v_2(t) - v_{ref}\right)^2 + \frac{1}{2}\left(v_3(t) - v_{ref}\right)^2$$

where v_{ref} is the desired velocity. For gap tracking, the cost is expressed as

$$L_d(t) = \frac{1}{2}\left(d_1(t) - d_{ref}\right)^2 + \frac{1}{2}\left(d_2(t) - d_{ref}\right)^2$$

where d_{ref} is the desired spacing. For conserving fuel the cost is

$$\begin{aligned} L_u(t) &= \tfrac{f_1(t)}{v_1(t)} + \tfrac{f_2(t)}{v_2(t)} + \tfrac{f_3(t)}{v_3(t)} \\ f_i(t) &= \begin{cases} \frac{P_i(t)}{\eta_i(P_i(t))Q} & \text{for } u_i \geq 0 \\ 0 & \text{for } u_i < 0 \end{cases} \end{aligned} \tag{3.8}$$

where f_i, P_i and η_i are the fuel consumption per unit time, the engine power output and the engine efficiency, respectively. Q = 34.5MJ/L is the calorific value of gasoline.

The engine power output P_i needed for driving a vehicle can be represented by

$$P_i(t) = \left(m_i \dot{v}_i(t) + f_{ai}(t) + f_{gi}(t) + f_{\mu i}\right) v_i(t) + P_c \tag{3.9}$$

where $P_c = 845.825$W is the constant power required when the vehicle is idling. Furthermore, based on the engine characteristics map [31] which shows the relationship between engine power output and fuel efficiency, the maximum efficiency η_i can be presented as a function of the engine power output P_i. η_i can be written as

$$\begin{aligned} \eta_i\left(P_i(t)\right) = & e_1 P_i^6(t) + e_2 P_i^5(t) + e_3 P_i^4(t) e_4 P_i^3(t) \\ & e_5 P_i^2(t) + e_6 P_i(t) + e_7 \end{aligned}$$

where $e_1 \ldots e_7$ are parameters shown in Table 3.3 .

e_1	-1.508×10^{-28}	e_5	-2.908×10^{-9}
e_2	3.448×10^{-23}	e_6	3.197×10^{-5}
e_3	-3.050×10^{-18}	e_7	0.127
e_4	1.313×10^{-13}	–	–

Table 3.3: Parameters that give the maximum efficiency in the equation

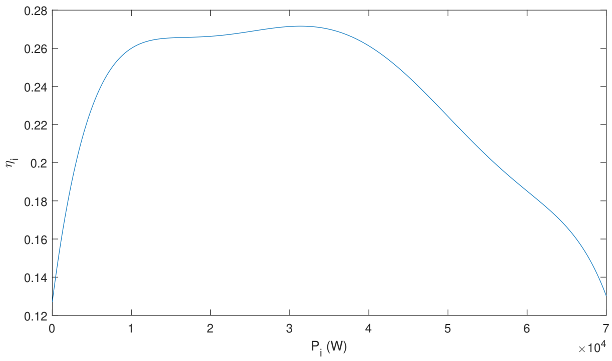

Figure 3.3: Efficiency w.r.t power output

Finally, for Task 5 we model the cost as

$$L_a(t) = \frac{1}{2}\left(u_1(t)^2 + u_2(t)^2 + u_3(t)^2\right) \tag{3.10}$$

This puts a penalty on large inputs and also ensures smooth inputs by reducing erratic behaviour.

Now, we define the input constraints to the problem, which are modelled as

$$|u_1(t)| \leq u_{1\,\mathrm{max}}, \quad |u_2(t)| \leq u_{2\,\mathrm{max}}, \quad |u_3(t)| \leq u_{3\,\mathrm{max}} \tag{3.11}$$

where the u_{max} denote the maximum input that we want to permit for the system.

After combining all the information gathered from above the final MPC formulation is as follows

$$\min_{u^i(t))} J^{\mathrm{MPC}}\left(\mathbf{z}^i, \mathbf{u}^i, \mathrm{t}\right)$$

subject to

$$\mathbf{z}^i(t+1) = f\left(\mathbf{z}^i(t), \mathbf{u}^i(t)\right), \tag{3.12a}$$

$$\mathbf{x}_{\mathrm{min}} \leq \mathbf{x}^i(t) \leq \mathbf{x}_{\mathrm{max}}, \tag{3.12b}$$

$$\mathbf{v}_{\mathrm{min}} \leq \mathbf{v}^i(t) \leq \mathbf{v}_{\mathrm{max}}, \tag{3.12c}$$

$$\mathbf{u}_{\mathrm{min}} \leq \mathbf{u}^i(t) \leq \mathbf{u}_{\mathrm{max}}, \tag{3.12d}$$

$$\mathbf{z}^i(0) = \mathbf{z}^i(t). \tag{3.12e}$$

3.1.4 Simulation Results

This section discuss the simulations carried out to determine the performance of the Acceleration Control approach. First, we will try simulations with three vehicles in the platoon. Table 3.4 defines the parameters related to the vehicles.

Parameter	Value
m_i	1480 kg
A_i	2.87 m^2
ρ	1.2 kg/m^3
μ	0.01
g	9.8 m/s^2
l	4.3 m
dt	0.04 s
N	20 steps

Table 3.4: Value of Parameters

In Table 3.4 dt is the time-step for the discretization of the continuous problem and N is the prediction horizon of the nonlinear MPC. Next, the constraints need to be applied to the states and inputs of the system. The constraints were borrowed from [22] and showed smooth and stable behavior for the platoon vehicles after conducting several trials. Table 3.5 summarises the constraints.

State and Inputs	Lower Limit	Upper Limit
Position (m)	0	∞
Velocity ($\frac{m}{s}$)	0	30
Acceleration ($\frac{m}{s^2}$)	-1.27	1.27

Table 3.5: State and Input Constraints

Since the platoon is homogeneous, i.e. all vehicles are the same, these constraints and parameters apply to all the vehicles. As mentioned in the MPC formulation, we want the vehicles in the platoon to track a certain velocity. A certain reference gap also needs to be maintained between the vehicles in the platoon. Following are the reference values

$$\mathrm{v_{ref}} = 27\, m/s \ (97.2\, km/h)$$

$$\mathrm{d_{ref}} = 4\, m$$

The initial conditions also need to be set before solving the problem below and they are listed in the table below.

Vehicle	Initial State
1	[24.6, 26]
2	[12.3, 26]
3	[0, 26]

Table 3.6: Initial State of each Vehicle

The above initial conditions leads to an initial gap between the vehicles $\mathrm{d_i}(0) = 8\, m$. Finally, the one of most important part of running an MPC problem is the tuning of weights in cost function (3.7). For this problem these weights refer to ω_v, ω_d, ω_u and ω_a. The optimal weights weights obtained after several trials and observations are listed in Table 3.7

Weight	Value
ω_v	600
ω_d	75
ω_u	100
ω_a	5

Table 3.7: Values of Weights for 3 vehicles

Results for 3 Vehicles

Using the above-mentioned parameters, weights, constraints and initial conditions, the behaviour showed in Fig. 3.4 is obtained. The first graph shows the velocity profile of the vehicles, the second one shows how the respective gaps between the trucks evolve, and the final graph shows the inputs sent into the system.

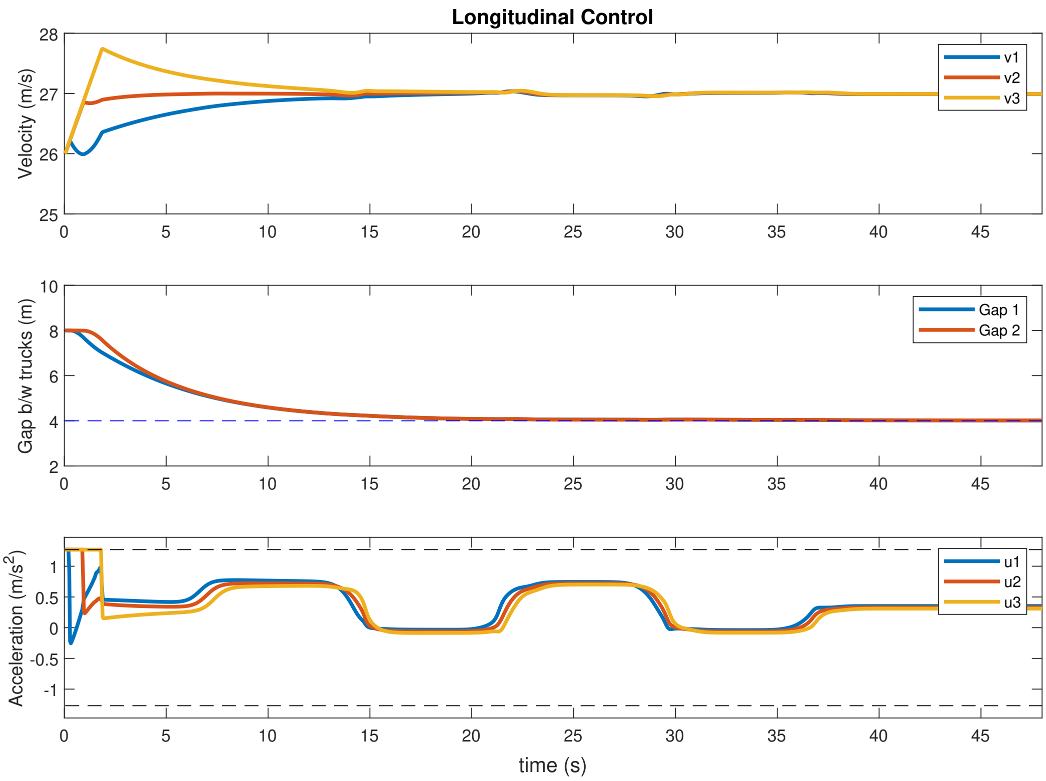

Figure 3.4: a) Velocity profile of all vehicles b) Gap b/w the vehicles c) Acceleration profile of the vehicles

The velocity of the vehicles increases as it has to go from 26 to 27 m/s. However, it can be noticed that the velocity of Vehicle 1 (leading vehicle) increases slower compared to the other velocities. This happens so that the vehicles behind can slowly close the gap to the vehicles in front with higher velocity. For vehicle 3, it even overshoots the reference velocity to gain on the vehicle in front. During this time, the gap comes down very quickly and starts converging to the reference value. As both gaps come closer to the reference value, the velocities of each vehicle also start converging towards their respective references. The effect of MPC can be seen here as it looks forward in the horizon and determines when the values for velocity and gaps may converge, leading to a very smooth convergence for both values. The acceleration also starts at the highest values so that all vehicles can reach their higher reference velocity but again a_1 drops to a lower value first to let the vehicles behind build a speed advantage to close the gap. Also, it can be seen that the acceleration also increases and decreases w.r.t to the slope the platoon encounters. The immediate increase in acceleration value from 0 to maximum at the start and then the sudden decrease is not desirable and not realistic too. This problem addressed in Section 3.2.

Results for 4 Vehicles

The problem can be easily extended to accommodate more vehicles in the platoon. Fig. 3.5 shows the behaviour for 4 trucks. The fourth added vehicle has all the same parameters and constraints as the other 3 to maintain the notion of a homogeneous platoon. However, the tuning needs to be done again, and the new weights are highlighted in Table 3.8.

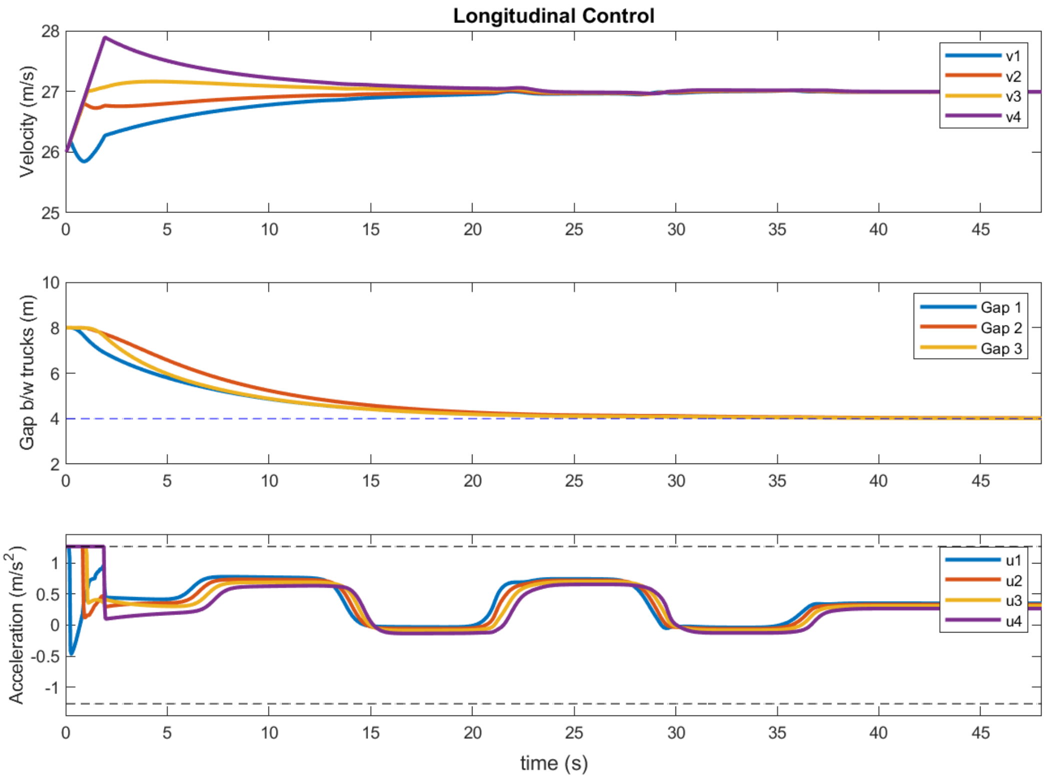

Figure 3.5: Simulation Results for 4 vehicles

All the principles and behaviours of states such as the vehicles building up a velocity advantage and smooth convergence of gap and velocity to its reference value, remain the same.

Weight	Value
ω_v	400
ω_d	50
ω_u	65
ω_a	3

Table 3.8: Values of Weights for 4 Vehicles

Fuel Saving

The above figures for acceleration control indicate how the acceleration control works well for tracking but does not showcase the fuel-saving part of the cost function. Table 3.9 shows weight ω_u effects the fuel consumption. All values were calculated on the same terrain and running the simulation for the same amount of time.

Value for ω_u	Fuel consumed by Vehicle 1 (mL)	Fuel consumed by Vehicle 2 (mL)	Fuel consumed by Vehicle 3 (mL)
0 (No fuel optimisation)	36.18	33.49	30.69
100	33.74 (-6.74%)	30.89 (-7.76%)	29.41 (-4.17%)
500	31.55 (-12.79%)	30.34 (-9.41%)	28.71 (-6.45%)

Table 3.9: Effect of Fuel Saving

Therefore, it can be seen that increasing the priority on fuel-saving does result in more fuel saving due the vehicles in the platoon now employing more lifting and coasting leading to less abrupt braking and accelerating. However, it should be stated that although $\omega_u = 500$ does lead to more fuel efficiency, it also results in the problem solving being slower (84.38s for ω_u = 100 and 198.64s for ω_u = 500). Also, more priority on fuel-saving leads to worse velocity and gap tracking for the vehicles. Also, higher values of ω_u result in worse exitflag behaviour in FORCESPRO due to scaling problems. These issues are also discussed in more detail in Appendix A. Therefore, the value $\omega_u = 100$ was realised as the optimal value after multiple simulations and observations.

3.2 Jerk Control

As shown in the above section, acceleration control works well to fulfil all the tasks that it was assigned. It can do the tracking well on both fronts and also ensures that the constraints are obeyed. With correct tuning of weights, smooth behaviour of inputs and smooth convergence of velocity and gap to their reference is also ensured. However, it comes up short in one aspect, i.e. the sudden increase and decrease of input at the start of the simulation. As can be seen in the bottom right graph of Fig. 3.4 the acceleration value jumps from 0 at the start to the maximum allowable almost instantly, which in reality will not be possible. Also, after around 2 seconds, all inputs start to come down from the maximum possible to establish a status quo, and this decrease is also very sudden. Studying these values suggested that this decrease resulted in a jerk (rate of change of acceleration) values of up to 13.75 m/s^3, which is very high. For reference ideal jerk values for driving on the highway are generally between -2 m/s^3 and +2 m/s^3 [16]. Values higher than 6 m/s^3 can also cause severe whiplash and values greater than 10 m/s^3 can also break the driver's neck. Apart from the safety concerns, this kind of jerk is also impossible to implement in real-life vehicles.

Thus, there needs to be a limit to what the jerk can be to ensure that the manoeuvres are safe and feasible in real life. A handy way to achieve this is to employ Jerk Control. As the name suggests, now the control input into the system is jerk instead of acceleration. This way, the amount of jerk into the system can be regulated on top of having control over the permissible acceleration values.

3.2.1 Modelling of Platooning

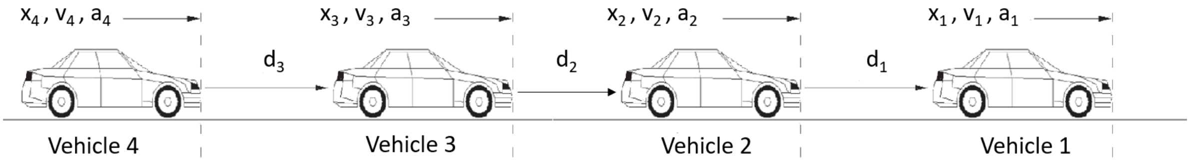

Figure 3.6: Platoon Model - Jerk Control

The only fundamental change to the model now is to assign acceleration to be a state of the system, and jerk becomes the input. This results in each truck being described by a state space of 3 states instead of 2 states as it was in (3.1). It results in the following state-space model for 3 trucks

$$
\begin{aligned}
m_i\dot{v}_i(t) &= -f_{ai}(t) - f_{gi}(t) - f_{\mu i} + m_i a_i(t) && (3.13)\\
\dot{a}_i &= u_i && (3.14)\\
f_{ai}(t) &= \frac{1}{2}\rho A_i C_i(d(t)) v_i^2(t) && (3.15)\\
f_{gi}(t) &= m_i g \sin\theta\left(x_i(t)\right) && (3.16)\\
f_{\mu i} &= \mu_i m_i g && (3.17)\\
d_i(t) &= x_i(t) - x_{i+1}(t) - l_i && (3.18)
\end{aligned}
$$

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), u(t))\\
x(t) &= \begin{bmatrix} x_1 & v_1 & a_1 & x_2 & v_2 & a_2 & x_3 & v_3 & a_3 \end{bmatrix}^T\\
u(t) &= \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T,
\end{aligned}
\tag{3.19}
$$

$$f(x(t), u(t)) = \begin{bmatrix} v_1 \\ -\frac{1}{m_1}(f_{a1} + f_{g1} + f_{\mu 1}) + a_1 \\ u_1 \\ v_2 \\ -\frac{1}{m_2}(f_{a2} + f_{g2} + f_{\mu 2}) + a_2 \\ u_2 \\ v_3 \\ -\frac{1}{m_3}(f_{a3} + f_{g3} + f_{\mu 3}) + a_3 \\ u_3 \end{bmatrix} \tag{3.20}$$

The main difference that can be noticed here lies with (3.14) which has been added such that now acceleration is a state and its derivative is the input to the system, i.e. the jerk.

3.2.2 MPC Formulation

This section highlights the differences in the formulation when compared to the formulation in Section 3.1.3. Compared to the formulation acceleration control, the running cost function only changes w.r.t to 2 tasks, i.e. Task 4 and 5.

For Task 4 i.e. fuel optimization the formulation of L_u changes to

$$\begin{aligned} L_u(t) &= \tfrac{f_1(t)}{v_1(t)} + \tfrac{f_2(t)}{v_2(t)} + \tfrac{f_3(t)}{v_3(t)} \\ f_i(t) &= \begin{cases} \frac{P_i(t)}{\eta_i(P_i(t))Q} & \text{for} \quad a_i \geq 0 \\ 0 & \text{for} \quad a_i < 0 \end{cases} \end{aligned} \tag{3.21}$$

The only difference here is that the value of f_i now depends on a_i instead of u_i as now the input is jerk and acceleration is a state.

For Task 5, due to the same reason, the formulation of L_a is

$$L_a(t) = \frac{1}{2}\left(a_1(t)^2 + a_2(t)^2 + a_3(t)^2\right) \tag{3.22}$$

Additionally, the state constraints will now increase in dimensionality due to an additional state, i.e. acceleration. Finally, after making the above changes, the final MPC formulation looks like this

$$\min_{u^i(t))} J^{\text{MPC}}\left(\mathbf{z}^i, \mathbf{u}^i, \text{t}\right)$$

subject to

$$\mathbf{z}^i(t+1) = f\left(\mathbf{z}^i(t), \mathbf{u}^i(t)\right), \tag{3.23a}$$

$$\mathbf{x}_{\min} \leq \mathbf{x}^i(t) \leq \mathbf{x}_{\max}, \tag{3.23b}$$

$$\mathbf{v}_{\min} \leq \mathbf{v}^i(t) \leq \mathbf{v}_{\max}, \tag{3.23c}$$

$$\mathbf{a}_{\min} \leq \mathbf{a}^i(t) \leq \mathbf{a}_{\max}, \tag{3.23d}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}^i(t) \leq \mathbf{u}_{\max}, \tag{3.23e}$$

$$\mathbf{z}^i(0) = \mathbf{z}^i(t) \tag{3.23f}$$

3.2.3 Simulation Results

This section shows the simulations done for Jerk Control in order to determine if the additions made to obtain the new formulation work as desired. The parameters used to obtain the dynamic equations remain the same and are used from Table 3.4. Regarding the constraints, Table 3.5 needs to be appended to Table 3.10.

State and Inputs	Lower Limit	Upper Limit
Position (m)	0	∞
Velocity ($\frac{m}{s}$)	0	30
Acceleration ($\frac{m}{s^2}$)	-1.27	1.27
Jerk ($\frac{m}{s^3}$)	-2	2

Table 3.10: State and Input Constraints

The reference values also remain the same and initial conditions also need to be appended to accommodate the extra state as shown in Table 3.11

Vehicle	Initial State
1	[24.6, 26, 0]
2	[12.3, 26, 0]
3	[0, 26, 0]

Table 3.11: Initial State of each Vehicle

Finally, since the MPC formulation has already been adjusted towards now recognising acceleration as a state and not an input, there is no need to alter the tuning weights of the running cost and they can be directly used from Table 3.7.

Results for 3 Vehicles

Implementing the changes stated above, we obtain Fig. 3.7 wherein the same information is displayed as Fig. 3.4.

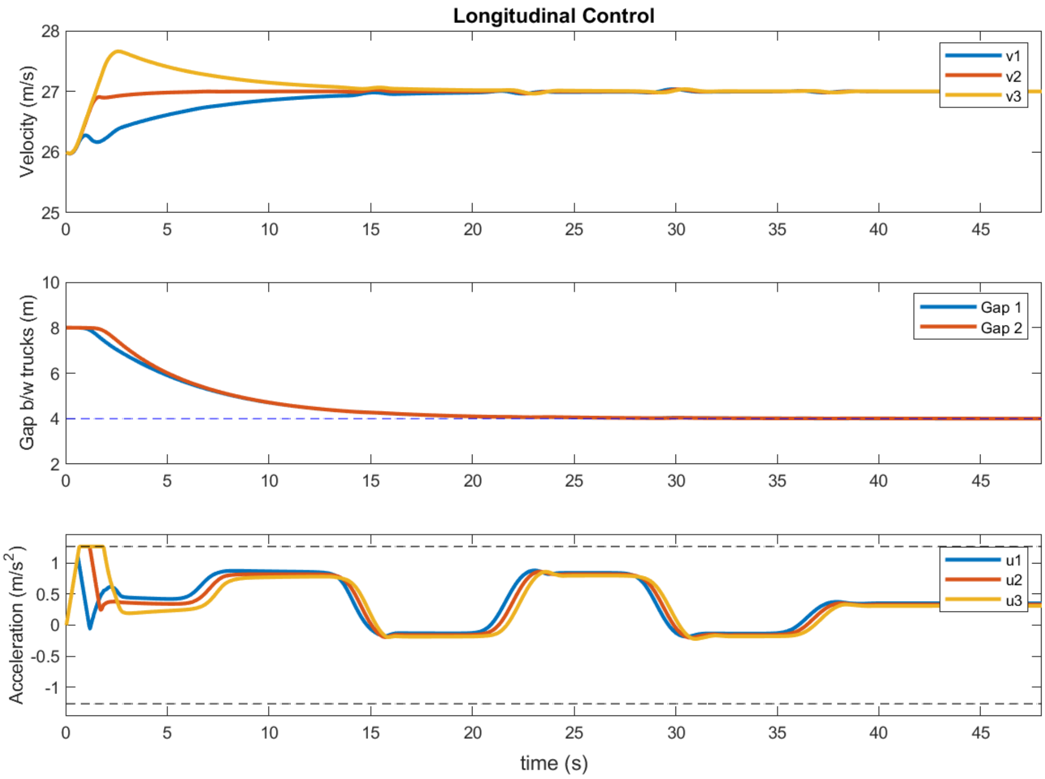

Figure 3.7: Simulation Results for 3 Trucks

The behaviour of the vehicles in the platoon is largely similar to Acceleration Control in

- The vehicles have broadly the same velocity profile as in Fig. 3.4 i.e. the follower vehicles still try and build up a speed advantage to close in the gap to the reference value.
- The gap still goes down smoothly due to the speed advantage and also converges smoothly to the reference value like the velocity.
- Acceleration accommodates the terrain changes to maintain smooth convergence along with velocity and gap tracking.

Crucially, the difference lies in the behaviour of the inputs. It can be seen that as opposed to Fig. 3.4 the acceleration does not jump from 0 to maximum value instantaneously and rather goes towards it slowly for all vehicles. Further, when the trucks are close to their reference gap and velocity, the acceleration also does not drop suddenly. Again it decreases at a much slower rate due to the limits on negative jerk. Therefore, it can be concluded that jerk control works as desired for this system. This problem can also be scaled up to 4 vehicles by making the changes that were made before, and the results are shown in Fig. 3.8.

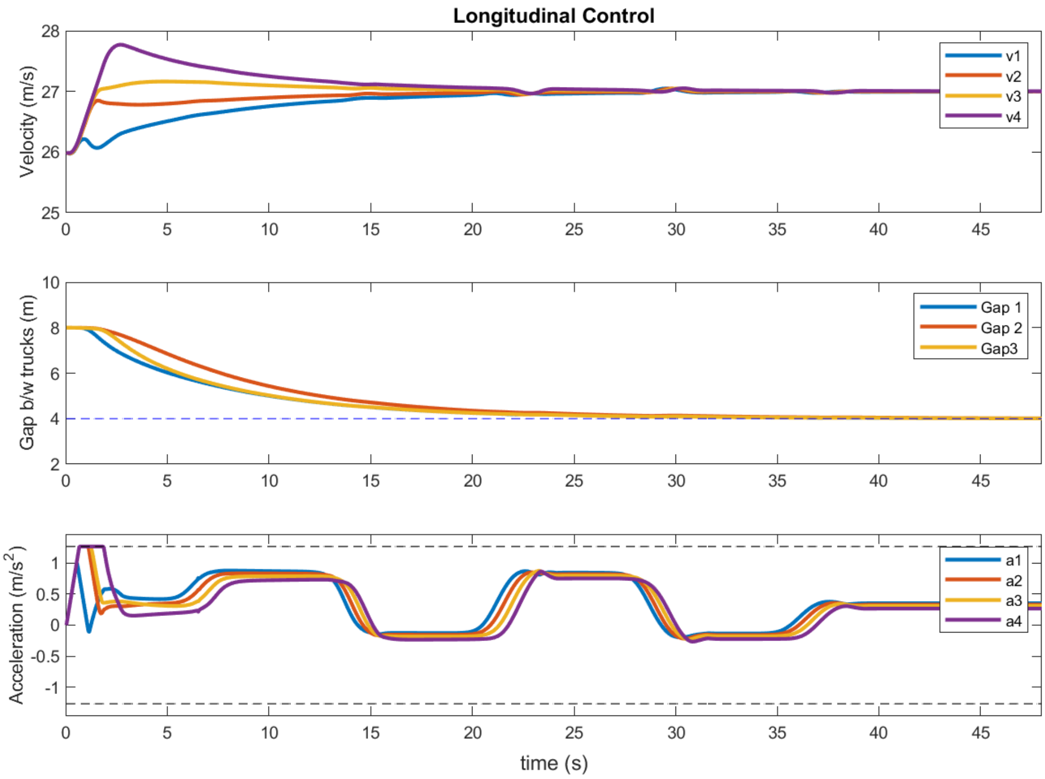

Figure 3.8: Simulation Results for 4 Trucks

Jerk Control without Tracking

Tracking is an important part of Longitudinal Control. The algorithms are built around maintaining a target/reference velocity and gap between vehicles, as this is sometimes the main aim of the platoon. However, a lot of the literature also focuses on saving as much fuel as possible. One way to do this is by removing the tracking requirement altogether and allow the vehicles to operate in a safe region. In our case, this would mean that there would be no velocity and gap tracking. As a result, the cost function will focus much more on fuel saving and also, the platoon will not need to constantly brake and accelerate (especially during climbing and coming down slopes) to maintain the reference velocity or gap. Instead, it can stay within a safety range defined for both these parameters, saving significant amounts of fuel. The drawback of this approach is that now the system needs to have an emergency braking mode for the case wherein the platoon might need to brake for safety reasons and go out of the bounds defined for velocity.

For the jerk control system defined in this chapter, we just need to alter constraints for velocity and add constraints on the gap between the vehicles. These constraints are as follows

$$\begin{aligned} 25\,m/s \le v_i \le 29\,m/s \\ 3\,m \le d_i \le 10\,m \end{aligned} \tag{3.24}$$

The weights for tuning were $\omega_u = 1250$ and $\omega_a = 5$ and the results using these weights are shown in Fig. 3.9.

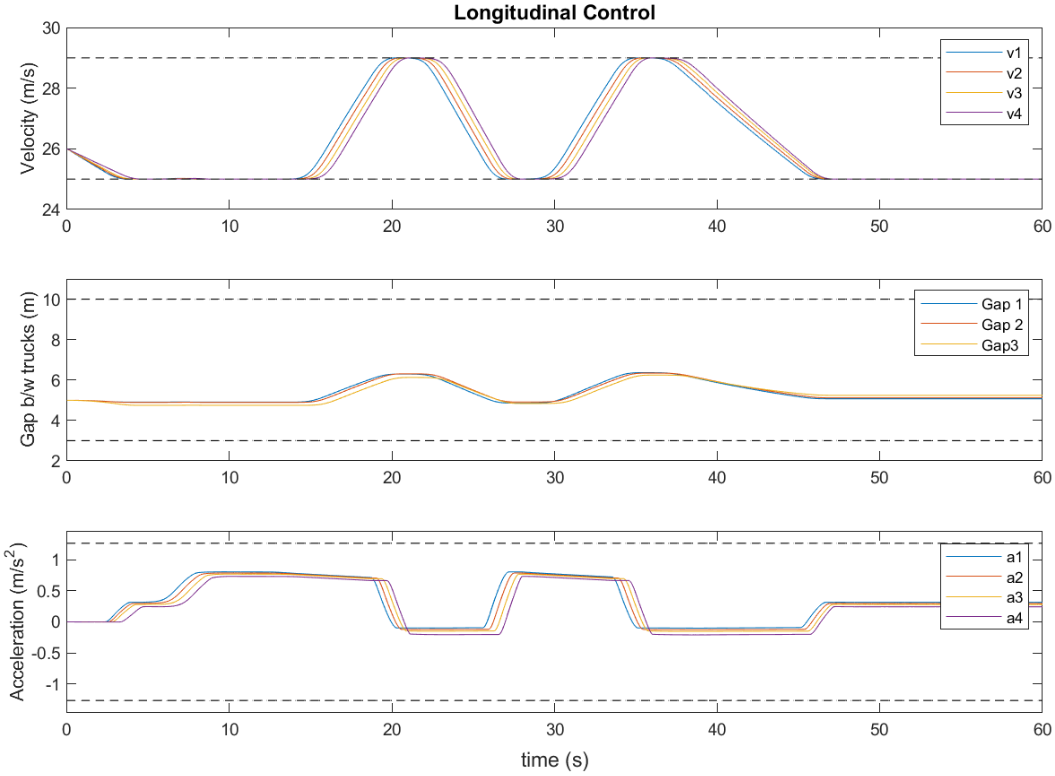

Figure 3.9: Simulation Results for 4 Trucks without tracking

As expected the system can now work in range of velocity and gap values instead of just trying to achieve a reference. This leads to much less braking and acceleration which results in more lifting and coasting as well. The effect on fuel saving is summarised in Table 3.12.

Case with Jerk Control	**Fuel consumed by Vehicle 1 (mL)**	**Fuel consumed by Vehicle 2 (mL)**	**Fuel consumed by Vehicle 3 (mL)**	**Fuel consumed by Vehicle 4 (mL)**
With velocity and gap tracking	106.01	102.88	99.42	92.55
Without velocity and gap tracking	91.15 (-16.30%)	88.04 (-14.42%)	85.13 (-14.37%)	79.17 (-14.49%)
Without tracking lax weights	82.15 (-22.51%)	79.33 (-22.89%)	76.63 (-22.92%)	71.98 (-22.22%)

Table 3.12: Fuel Saving with no Tracking

Therefore, it can be concluded that taking away tracking can increases fuel efficiency. The final row in the table showed data when the constraints for velocity were altered to be

$$22 \leq v_i \leq 30 \tag{3.25}$$

resulting in a broader range of velocity in which the vehicle can work, leading to even more fuel efficiency.

3.3 Conclusion

This chapter focused on coming up with an MPC formulation that is capable of employing Longitudinal Control on a platoon of vehicles. The challenges accomplished in this chapter were

1. The formulation can fulfil all tasks.
2. It can also bee seen that it easy to add more number of trucks into the platoon by appending the state space and the running cost of the MPC. Therefore, the system designed is scalable. However, adding extra trucks does make the problem solving slower. The total solve time for the jerk control problem of three trucks was 93.81 s and for four trucks it took 226.10 s.
3. The formulation for fuel-saving and making use of drag resistance works well.
4. The model is well-tuned after several tests and works well for all scenarios.

Therefore, it can be concluded that the proposed centralised Eco-MPC works well and thus can be used to further develop the end to end 2D control, which will be discussed in the next chapter.

.

CHAPTER 4

TWO DIMENSIONAL CONTROL

Two Dimensional Control comprises of control in 2 dimensions (in terms of position) as the name suggests. One dimension comprises of *Longitudinal Control* wherein the control is only applied assuming that the platoon or vehicle does not need to change lanes at any point. The second dimension, however, considers that there are multiple lanes on the road and allows the platoon to switch lanes, break formation, and come back into formation. This kind of lane changing requires *Lateral Control* of the platoon. This additional dimension of control gives the platoon more flexibility. However, implementing *Lateral Control* requires knowledge of motion planning and especially collision avoidance which was not needed in Longitudinal Control. Lane changing in itself is also not easy to implement. Highways, where vehicles arrive at high speeds on the adjacent lane, are very susceptible to accidents, and many of them are due to lane changes [41]. Therefore, lane changing may provide versatility, but it is also more convoluted and not as straightforward as *Longitudinal Control*.

There are several ways to implement lane changes and collision avoidance in a platoon. The decision making and control architecture can be *centralised* [12], i.e. there is just one CPU that takes in information about the states of all vehicles, determines the optimal input to fulfil given tasks by solving a nonlinear MPC problem and communicates the optimal inputs to the respective vehicles in the platoon. The architecture can also be *distributed* [11] wherein each vehicle has its own CPU, and this unit only takes relevant information regarding the vehicles around the vehicle in the platoon and only decides upon the optimal input for that particular vehicle by solving a less complicated nonlinear MPC problem. This problem is less convoluted than the centralised one as there the CPU has information of the states for all vehicles and needs to determine the input for all vehicles and thus needs to solve a much bigger MPC problem. In the distributed case, the computation time goes down significantly [11][46][17]. However, each vehicle CPU does not have perfect information about the vehicles in the platoon, and there is some error that needs to be accounted for somehow.

In all of the cited literature, the model being used is a Kinematic Bicycle model. However, in the following sections, a Dynamic Bicycle model is used to model the vehicle's dynamics. The reason for this is also stated later. This decision making architecture is centralised wherein the CPU will be placed in the leading vehicle of the platoon.

4.1 Combining Behaviours

As mentioned before, the novel aspect of this thesis project is to implement Longitudinal Control along with Formation Reconfiguration and Collision Avoidance using a single model and MPC formulation. Therefore, we want an end-to-end control architecture that can employ fuel-saving when in formation and reconfiguration along with avoidance based on the environment around the platoon. Since the requirements such as model and MPC formulation for both the behaviours are very different, the research is very disjointed between these two topics. There is no clear link in the literature. The thesis project aims to provide that link using a Dynamic Bicycle model (DBM) to model each vehicle in the platoon and using this model to formulate the MPC.

Section 4.2 will discuss how DBM is used to model the platoon. Section 4.3 discusses how the non-linear MPC is formulated for the novel model and finally simulations will be discussed in detail in Section 4.4.

4.2 Modelling of Platoon

As mentioned in Section 3.1.2 the first step in deciding the model is to determine which parameters should be chosen as the states for the model. Starting with what has already done, the model for Longitudinal Control has already been developed and it also worked as expected. Therefore, the basic formulation of that model is still needed. The equations are again listed here

$$
\begin{aligned}
m_i \dot{v}_i(t) &= -f_{ai}(t) - f_{gi}(t) - f_{\mu i} + m_i a_i(t) && (4.1)\\
\dot{a}_i &= u_i && (4.2)\\
f_{ai}(t) &= \frac{1}{2}\rho A_i C_i(d(t)) v_i^2(t) && (4.3)\\
f_{gi}(t) &= m_i g \sin\theta\,(x_i(t)) && (4.4)\\
f_{\mu i} &= \mu_i m_i g && (4.5)\\
d_i(t) &= x_i(t) - x_{i+1}(t) - l_i && (4.6)
\end{aligned}
$$

It can be noticed that the equations have been borrowed from Jerk Control as one of the control parameters will be Longitudinal Jerk due to the drawbacks of choosing Longitudinal Acceleration as an input, as was discussed above. As it now becomes relevant, x_i represents the longitudinal position of the vehicle in world coordinates. Therefore, the equation will be altered from the dynamics defined for the Longitudinal Control section (shown later).

We augment the longitudinal model with some states that describe the lateral dynamics of the vehicle. For this purpose, the Dynamic Bicycle Model (DBM) is used. The Dynamics Bicycle model is chosen firstly because, at higher speeds, the assumption that the velocity at each wheel is in the direction of the wheel can no longer be made. Therefore, the Kinematic model does not hold true. We aim to incorporate external forces such as drag resistance, slope forces friction into our model, which is also not possible if a Kinematic model is used.

The DBM has been described in Section 2.1.2. Taking the equations from the state space, we get the following

$$\dot{y_i} = v_{x,i}\sin(\psi_i) + v_{y,i}\cos(\psi_i) \tag{4.7}$$

$$\dot{v_{y,i}} = -\left(\frac{C_{\alpha f} + C_{\alpha r}}{m v_{x,i}}\right) v_{y,i} - \left(v_{x,i} + \frac{C_{\alpha f}\ell_f - C_{\alpha r}\ell_r}{m v_{x,i}}\right) r + \frac{C_{\alpha f}}{m}\delta_i \tag{4.8}$$

$$\dot{\psi_i} = r_i \tag{4.9}$$

$$\dot{r_i} = -\left(\frac{\ell_f C_{\alpha f} - \ell_r C_{\alpha r}}{I_z v_{x,i}}\right) v_{y,i} - \left(\frac{{\ell_f}^2 C_{\alpha f} + {\ell_r}^2 C_{\alpha r}}{I_z v_{x,i}}\right) r + \frac{\ell_f C_{\alpha f}}{I_z}\delta_i \tag{4.10}$$

where y_i is the lateral position of the vehicle i in the global co-ordinates, $v_{y,i}$ is the lateral velocity of the vehicle, ψ_i represents yaw angle of the vehicle w.r.t the X axis and therefore the r_i represents the yaw rate of the vehicle. This model sufficiently encompasses all the lateral dynamics required for the project and therefore the equations are used as it is. In the model above the control variable is the steering angle of the vehicle δ_i. Therefore, using input constraints we can limit the values of the steering angle itself. However, it is also essential we limit the steering rate of the system as the there are mechanical and safety limits to how fast a steering can or should be rotated. Therefore, in the model the steering rate is taken as input and the steering angle now becomes a state of the system defined by

$$\dot{\delta_i} = \phi_i$$

where ϕ_i is the steering rate for vehicle i. Furthermore, (4.1) and (4.2) are now written as

$$\dot{v}_{x,i}(t) = \frac{1}{m_i}\Big(- f_{ai}(t) - f_{gi}(t) - f_{\mu i}\Big) + a_{x,i}(t) \tag{4.11}$$

$$\dot{a_{x,i}} = j_i \tag{4.12}$$

$$\tag{4.13}$$

as now these equations on define the longitudinal velocity and longitudinal acceleration respectively. The input vector for each vehicle now is of length 2 and has both jerk and steering rate

$$u_i = \begin{bmatrix} u_{1,i} & u_{2,i} \end{bmatrix}^T$$
$$u_{1,i} = j_i$$
$$u_{2,i} = \phi_i$$

Therefore, the final model $\mathbf{z}^i(t+1) = f\left(\mathbf{z}^i(t), \mathbf{u}^i(t)\right)$ is obtained by discretizing the continuous time model. The continuous time model for each vehicle reads out to be

$$\dot{x_i} = v_{x,i}\cos(\psi_i) - v_{y,i}\sin(\psi_i) \tag{4.14}$$

$$\dot{y_i} = v_{x,i}\sin(\psi_i) + v_{y,i}\cos(\psi_i) \tag{4.15}$$

$$\dot{v}_{x,i} = -\frac{1}{m}(f_{ai}(t) - f_{gi}(t) - f_{\mu i}) + a_{x,i}(t) \tag{4.16}$$

$$\dot{v}_{y,i} = -\left(\frac{C_{\alpha f} + C_{\alpha r}}{m v_{x,i}}\right) v_{y,i} - \left(v_{x,i} + \frac{C_{\alpha f}\ell_f - C_{\alpha r}\ell_r}{m v_{x,i}}\right) r + \frac{C_{\alpha f}}{m}\delta_i \tag{4.17}$$

$$\dot{a}_{x,i} = u_{1,i} \tag{4.18}$$

$$\dot{r_i} = -\left(\frac{\ell_f C_{\alpha f} - \ell_r C_{\alpha r}}{I_z v_{x,i}}\right) v_{y,i} - \left(\frac{{\ell_f}^2 C_{\alpha f} + {\ell_r}^2 C_{\alpha r}}{I_z v_{x,i}}\right) r + \frac{\ell_f C_{\alpha f}}{I_z}\delta_i \tag{4.19}$$

$$\dot{\delta_i} = u_{2,i} \tag{4.20}$$

$$\dot{\psi_i} = r_i \tag{4.21}$$

The novel aspect to this model formulation is the accommodation of drag and slope forces along with the lateral dynamics if the DBM.

States and Inputs	**Description**
x_i	x co-ordinate of the COM of vehicle
y_i	y co-ordinate of the COM of vehicle
$v_{x,i}$	Longitudinal velocity of the COM of vehicle
$v_{y,i}$	Lateral velocity of the COM of vehicle
$a_{x,i}$	Longitudinal acceleration of COM of vehicle
r_i	Yaw rate of the vehicle
δ_i	Steering angle of the vehicle
ψ_i	Yaw angle of the vehicle
j_i	Longitudinal Jerk of the COM of vehicle
ϕ_i	Steering rate of the vehicle

Table 4.1: Description of states and inputs

Finally, we append the drag model to account for the lateral position of the vehicle in the platoon. If the difference between the lateral position of two vehicles running in sequence is below a certain value then the vehicle behind will feel the effect of slip-streaming i.e. a reduced drag co-efficient and otherwise it will be of the nominal value.

Condition	$\mathbf{c_1}$	$\mathbf{c_2}$	$\mathbf{c_3}$
$\|y_1 - y_2\| \leq 0.375$ m and $\|y_2 - y_3\| \leq 0.375$ m	0.3	0.275	0.25
$\|y_1 - y_2\| \leq 0.375$ m and $\|y_2 - y_3\| \geq 0.375$ m	0.3	0.275	0.3
$\|y_1 - y_2\| \geq 0.375$ m and $\|y_2 - y_3\| \leq 0.375$ m	0.3	0.3	0.25
$\|y_1 - y_2\| \geq 0.375$ m and $\|y_2 - y_3\| \geq 0.375$ m	0.3	0.3	0.3

Table 4.2: Conditions for drag co-efficient values

4.3 MPC Formulation

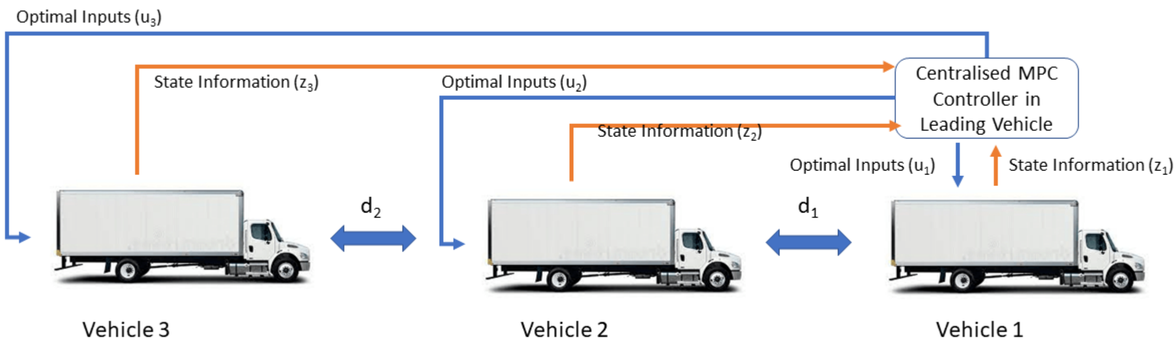

Figure 4.1: Flow of Information in a Centralised Control Architecture

As we saw in the MPC formulation of Longitudinal Control, there were a certain number of tasks that needed to be achieved through different parts of the running cost function J^{MPC}. In 2D control, the principle remains the same, i.e. we will still achieve the tasks at hand through different parts of the running cost function, but it will have additional tasks compared to before, and the priority of each task will vary depending on the environment around the platoon.

In addition to tasks that were defined in Longitudinal Control, there are more tasks that the platoon needs to fulfil w.r.t reconfiguration and collision avoidance. These tasks are

1. All vehicles in the platoon need to avoid collisions with any obstacles on the road, which could also be moving traffic.
2. All vehicles in the platoon also need to maintain a safe distance amongst themselves while performing any manoeuvres.
3. Each truck has to stick to a reference lane as well as it can.

4.3.1 Collision Avoidance

For this project, collision avoidance is implemented using Repulsive Potential Fields [26]. Potential Fields is a prevalent way of maintaining a safe distance from an obstacle in the field of robotics. In principle, as the robot/vehicle gets closer to the obstacle, the repulsive potential function U^{rep} function starts to increase in value significantly. This potential function primarily depends on the distance between the robot/vehicle, and this distance can be calculated in a number of ways. For this project, the distance is the Euclidean distance between the vehicle in the platoon and the obstacle i.e.

$$\text{D}(\mathbf{z^i}(\text{t})) = \sqrt{(\mathbf{x^i}(\mathbf{t}) - \mathbf{x^{obs}}(\mathbf{t}))^2 + (\mathbf{y^i}(\mathbf{t}) - \mathbf{y^{obs}}(\mathbf{t}))^2}$$

where $(\mathbf{x^i}, \mathbf{y^i})$ are the co-ordinates of vehicle i and $(\mathbf{x^{obs}}, \mathbf{y^{obs}})$ are co-ordinates of the obstacle. The formulation can be extended to multiple obstacles easily. **It is assumed that the**

platoon has information about the co-ordinates of the obstacles or it has an advanced radar system which can measure the distance to the obstacle accurately. Further, the potential function $\mathrm{U^{rep}}$ is defined as

$$\mathrm{U^{rep}}(\mathbf{z^i}(t)) = \begin{cases} \frac{1}{2}\left(\frac{1}{\mathrm{D}(\mathbf{z^i}(\mathrm{t}))} - \frac{1}{Q^*}\right), \mathrm{D}(\mathbf{z^i}(\mathrm{t})) \leq Q^* \\ 0\,, \mathrm{D}(\mathbf{z^i}(\mathrm{t})) > Q^* \end{cases} \tag{4.22}$$

where Q^* is the distance of influence corresponding to the potential field. Q^* decides when the potential field will be activated. It can be seen that when the Euclidean distance is more than Q^* the potential is not active. This is a general practice while implementing a repulsive potential field and saves on unnecessary computation when the obstacle is too far away from the vehicle. Q^* is design parameter and is chosen by the programmer after some testing. If it chosen to be too small then the vehicle will not have enough time to avoid the obstacle and if it is too big then there will unnecessary computation on the CPU which will make the problem solving slower.

Repulsive Potential Fields are used in this project as they are easy to implement but do require good tuning. Further, as the potential function $\mathrm{U^{rep}}$ itself acts as a cost function which needs to be minimised to ensure avoidance, it can be easily incorporated as part of the running cost which will be minimised when needed. Alternatively, hard constraints could have also been implemented for avoidance by using non-linear constraints. This method required more time to be implemented and hence was not used in this report. The benefit with hard constraints is that guarantee collision avoidance and in case its not possible, render the problem infeasible. For potential fields, good amount of tuning can ensure avoidance for many number of test cases but there is never a 100% guarantee unlike hard constraints.

4.3.2 Lane Tracking

2D control adds another dimension in which the trucks are controlled, namely the lateral dimension. It means that the platoon needs to stick to a single lane as much as possible, i.e. maintaining a stable lateral position in addition to maintaining proximity w.r.t to longitudinal distance. For this project, lane tracking is implemented through the running cost by using an error term.

$$\mathrm{L_y} = |\mathbf{y}_i - \mathrm{y}_i^{\mathrm{Ref}}|$$

where $\mathrm{y}_i^{\mathrm{Ref}}$ corresponds to the center of the reference lane in the simulation environment. All the assumptions mentioned Section 3.1 also hold true here. In addition, the the lane information is also assumed to available through a camera system or a map and as mentioned before there is assumed to be no error in the measurements. There are no hard constraints on the lane the vehicle should be in, as we want the platoon to be able to change lanes in case there are obstacles in front. Hence, it is incorporated into the running cost so that with good tuning the solver can make correct decisions regarding when to stick to the reference lanes and when to change lanes to avoid collisions.

4.3.3 Final Formulation

As mentioned above the tasks are still achieved through a running cost function $\mathrm{J}^{\mathrm{MPC}}$ wherein different parts of it are responsible for respective tasks. The function is defined as

$$\mathrm{J}^{\mathrm{MPC}}\left(\mathbf{z}^i, \mathbf{u}^i, \mathrm{t}\right) = \omega_y \mathrm{L_y} + \omega_v \mathrm{L_v} + \omega_d \mathrm{L_d} + \omega_u \mathrm{L_u} + \omega_a \mathrm{L_a} + \omega_c \mathrm{L_c} + \omega_{cd} \mathrm{L_{cd}} \tag{4.23}$$

Task	**Running Cost Term**	**Weighting Term**	**Cost Formulation**
Lane Tracking	$\mathrm{L_y}$	ω_y	$\sum_{i=1}^{m} \lvert \mathbf{y}_i - \mathrm{y}_i^{\mathrm{Ref}} \rvert$
Velocity Tracking	$\mathrm{L_v}$	ω_v	$\sum_{i=1}^{m} \lvert \mathbf{v}_i - \mathrm{v}_i^{\mathrm{Ref}} \rvert$
Gap Tracking	$\mathrm{L_d}$	ω_d	$\sum_{i=1}^{m} \lvert \mathbf{d}_i - \mathrm{d}_i^{\mathrm{Ref}} \rvert$
Fuel Saving	$\mathrm{L_u}$	ω_u	$\sum_{i=1}^{m} \frac{f_i}{v_i}$
Acceleration Moderation	$\mathrm{L_a}$	ω_a	$\sum_{i=1}^{m} \mathbf{a_i}^2$
Collision Avoidance with Obstacles	$\mathrm{L_c}$	ω_c	$\sum_{i=1}^{m} \sum_{j=1}^{n} \mathrm{U}_{\mathrm{ij}}^{\mathrm{rep}}$
Collision Avoidance within the Platoon	$\mathrm{L_{cd}}$	ω_{cd}	$\sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} \mathrm{U}_{\mathrm{ij}}^{\mathrm{rep}}$

Table 4.3: Running Cost Description

The breakdown of the above cost function is given in table 4.3. The number of vehicles in the platoon is defined by m and number of obstacles is n. Therefore, as can be seen in the formulation for collision avoidance with obstacles, each vehicle in the platoon has a repulsive potential field corresponding to each obstacle. Similarly, for the formulation of collision avoidance within the platoon, each vehicle has a repulsive potential field corresponding to every other vehicle.

The most important and fascinating aspect of the formulation is switching between Longitudinal Control and Collision Avoidance behaviours. The way this formulation achieves that is by changing the weighting terms when needed. Therefore, there are different modes, each of which corresponds to the platoon trying to achieve a different set of tasks.

The table below first shows which Mode corresponds to what behaviour and highlights what tasks are essential for a particular behaviour and which tasks are unimportant. The mode switching is performed by a simple Finite State Machine (discussed below) and it primarily depends on the distance between the platoon and the obstacles.

Mode	Behaviours	Task Priority						
		Lane Tracking	Velocity Tracking	Gap Tracking	Fuel Saving	Acc. Moderation	Avoidance with Obstacle	Avoidance Within
0	Leader Change	MED	MED	MED	MED	MED	LOW	**HIGH**
1	Longitudinal Control	LOW	**HIGH**	**HIGH**	**HIGH**	MED	LOW	LOW
2	Reconfiguration and Avoidance	**HIGH**	MED	LOW	MED	MED	**HIGH**	**HIGH**

Table 4.4: Task importance w.r.t Behaviour.

The corresponding weights assigned to each tasks for each behaviour are mentioned in the next section with the simulation results. Now that all parts of the formulation have been defined and described the final formulation can be written as

$$\min_{u^i(t))} \mathrm{J}^{\mathrm{MPC}}\left(\mathbf{z}^i, \mathbf{u}^i, \mathrm{t}\right)$$

subject to

$$\mathbf{z}^i(t+1) = f\left(\mathbf{z}^i(t), \mathbf{u}^i(t)\right), \tag{4.24a}$$

$$\mathbf{x}_{\min} \leq \mathbf{x}^i(t) \leq \mathbf{x}_{\max}, \tag{4.24b}$$

$$\mathbf{y}_{\min} \leq \mathbf{y}^i(t) \leq \mathbf{y}_{\max}, \tag{4.24c}$$

$$\mathbf{v_x}_{\min} \leq \mathbf{v_x}^i(t) \leq \mathbf{v_x}_{\max}, \tag{4.24d}$$

$$\mathbf{a_x}_{\min} \leq \mathbf{a_x}^i(t) \leq \mathbf{a_x}_{\max}, \tag{4.24e}$$

$$\delta_{\min} \leq \delta^{\,i}(t) \leq \delta_{\max}, \tag{4.24f}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}^i(t) \leq \mathbf{u}_{\max}, \tag{4.24g}$$

$$\mathbf{a_y}_{\min} \leq \mathbf{a_y}^i(t) \leq \mathbf{a_y}_{\max}, \tag{4.24h}$$

$$\mathbf{z}^i(0) = \mathbf{z}^i(t), \tag{4.24i}$$

The constraints on $\mathbf{a_y}^i(t)$ is introduced as a non-linear constraint in the solver and $\mathbf{a_y}$ is defined as

$$\mathbf{a_y} = -\frac{C_{\alpha f} + C_{\alpha r}}{m v_x} v_y + \frac{l_r C_{\alpha r} - l_f C_{\alpha f}}{m v_x} r + \frac{C_{\alpha f}}{m} \delta$$

4.4 Simulation Results

This section will discuss several uses cases to test our 2D control algorithm just like Section 3.1.4 and 3.2.3. All use cases here will be with a platoon of 3 vehicles. The parameters used here are different from before as now more information was needed for the vehicle's dynamics. So a model was needed wherein all information was present for lateral dynamics as well. The data was borrowed from TNO Car Labs. This was the best detailed set of parameters available for a vehicle and hence they were used in this project.

Parameter	Value
m_i	1845 kg
A_i	2.87 m^2
ρ	1.2 kg/m^3
μ	0.01
g	9.8 m/s^2
$C_{\alpha f}$	1.2×10^5 N/rad
$C_{\alpha r}$	2.2×10^5 N/rad
I_z	3.58×10^3 $kg\,m^2$
l_f	1.33 m
l_r	1.47 m
dt	0.04 s
N	50

Table 4.5: Value of Parameters

where $C_{\alpha f}$ and $C_{\alpha r}$ are the front and rear cornering stiffness of the vehicle respectively. I_z is the moment of inertia and l_f and l_r are the length of the front and rear axles of the vehicle. Notice that the prediction horizon has also increased for MPC in this problem as the one used before was not long enough such that the platoon could look enough in the future to make the correct decision on avoidance and reconfiguration. The values for parameters in Table 4.5 here correspond to a car model but it can be adapted such that it now represents a truck instead. Theoretically, if the parameters $m_i, A_i, C_{\alpha f}, C_{\alpha r}, I_z, l_f$ and l_r are increased to the right value they will represent a truck.

The constraints of the system were also largely decided using the data from Renault Car Labs and they were altered a bit so as to limit the overshoot when trying to achieve a required lateral position. The following table summarises the constraints.

State and Inputs	Variable	Lower Limit	Upper Limit
Longitudinal Position (m)	x_i	0	∞
Lateral Position (m)	y_i	1	9.5
Longitudinal Velocity ($\frac{m}{s}$)	$v_{x,i}$	0	30
Longitudinal Acceleration ($\frac{m}{s^2}$)	$a_{x,i}$	-1.27	1.27
Steering Angle (rad)	δ_i	-0.15	0.15
Yaw Angle (rad)	ψ_i	-2π	2π
Lateral Acceleration ($\frac{m}{s^2}$)	$a_{y,i}$	-2	2

Table 4.6: State and Input Constraints

The lane width for each lane in the Netherlands is 3.5 m, and the use cases have 3 lanes, and therefore the width of the whole road is 10.5 m. The limits on lateral position are decided while taking the vehicle's width in mind so that there is no sideways collision between the vehicle and barriers.

Regarding the reference values, in addition to the reference for velocity and gap as was in Longitudinal Control, their also a reference now for the Lateral Position. The references are as follows

$$\mathrm{v_{ref}} = 26\,m/s$$

$$\mathrm{d_{ref}} = 4\,m$$

Although the reference is dependent on the user, i.e. in which lane he/she would want the platoon to be in. For the uses cases below, the reference lane is always the middle lane. Thus,

$$\mathrm{y_{ref}} = 5.25\,m$$

which is the center of the middle lane. Further, the initial conditions for the vehicles are

Vehicle	Initial State
1	[13.6, 1.75, 25, 0, 0, 0, 0, 0]
2	[6.8, 1.75, 25, 0, 0, 0, 0, 0]
3	[0, 1.75, 25, 0, 0, 0, 0, 0]

Table 4.7: Initial State of each Vehicle

As the length of each vehicle $l = l_f + l_r$ which is 2.8 m, the gap initially between the vehicle is already 4 m. After tuning, the optimal parameters for the MPC formulation were also obtained. However, here as was mentioned above, each Mode has a different set parameters. the parameters match the principles from Table 4.4. The tuned weights are given in Table 4.8

Mode	$\omega_{\mathbf{y}}$	$\omega_{\mathbf{v}}$	$\omega_{\mathbf{d}}$	$\omega_{\mathbf{u}}$	$\omega_{\mathbf{a}}$	$\omega_{\mathbf{c}}$	$\omega_{\mathbf{cd}}$
Leader Change (0)	0.9	2	0.9	5	0.35	0	8
Longitudinal Control (1)	0.8	1.5	1.2	5	0.35	0	0
Reconfiguration and Avoidance (2)	0.075	0.7	0	2	0.5	30	2

Table 4.8: Tuning Weights

It is also essential to define the conditions in which a certain mode switch is required. If any of the vehicles in the platoon is within 70 m of the obstacle then the mode switches from 1 to 2 and when each truck is sufficiently far from all obstacles, in our case 100 m then the mode switches from 2 to 1. This is shown in Fig. 4.2.

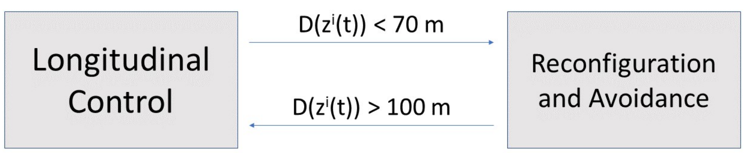

Figure 4.2: Decision Making Description

The use cases will show how the platoon goes around single and multiple moving obstacles on the road. It will also show how the platoon deals with two obstacles in opposite directions of the platoon and finally the it will also be showcased how the platoon changes the leader effectively.

4.4.1 Going around 1 Obstacle

The first use cases to be discussed will have only one larger moving obstacle in the middle lane. As mentioned above, vehicles in the platoon have to achieve a constant velocity of 26 m/s as one of the tasks. Therefore, the obstacle has a lower velocity compared to the platoon. This velocity was of the obstacle is set to be constant at 16 m/s. Therefore the platoon approaches the obstacle at a relative velocity of roughly 10 m/s. The starting position of the obstacle is set to be [100,5.25].

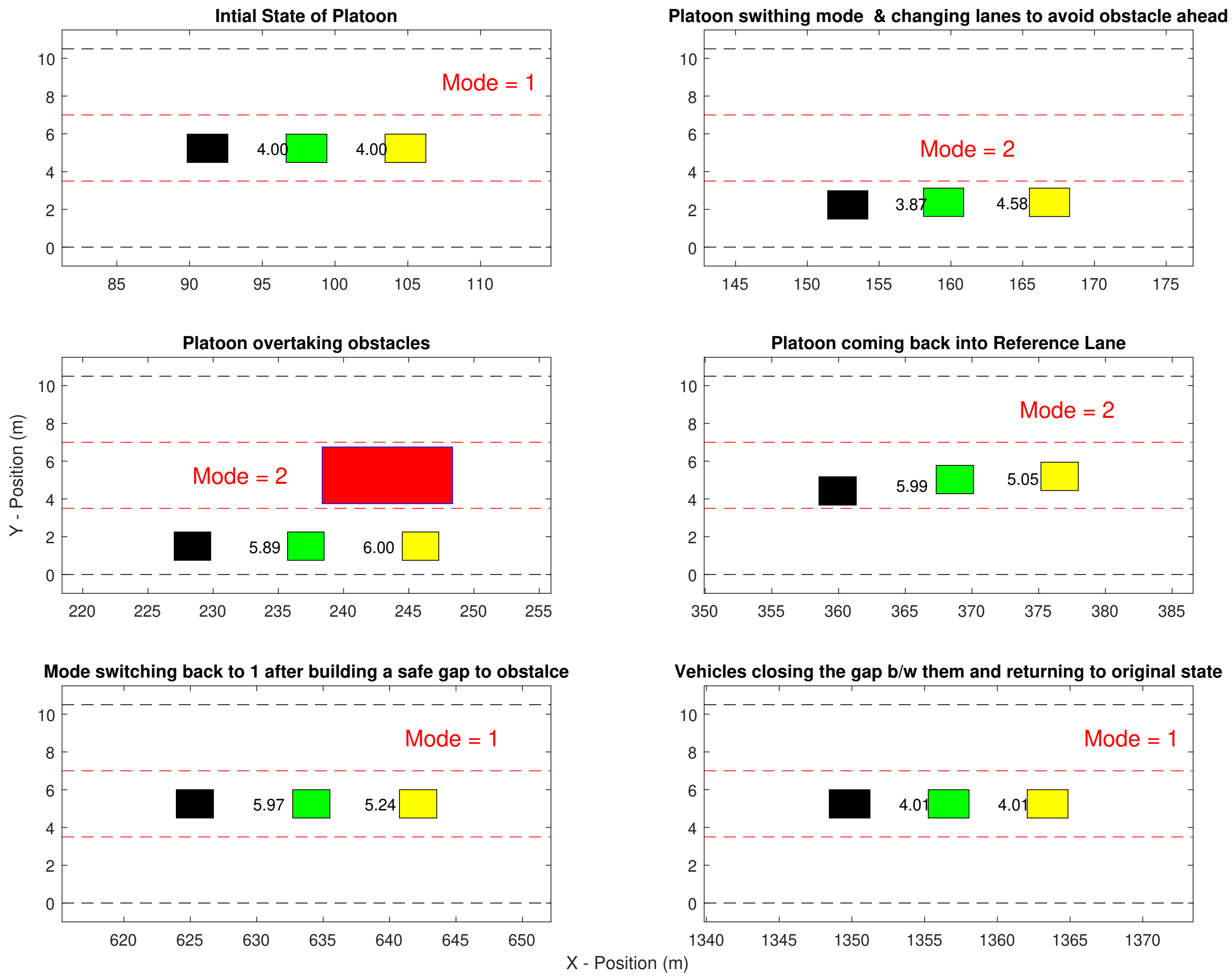

Figure 4.3: Simulation Results for 3 Trucks to go around 1 Obstacle

The platoon will initially start in the middle lane, and the lateral position reference for the platoon during the simulation is also set to be in the middle lane. Therefore, the platoon should try and be in the middle lane as much as possible unless it has to avoid any obstacles and ensure safety. Fig. 4.3 shows the behaviour of the platoon and gives a birds-eye view of the simulation. The simulation is shown chronologically by row i.e. the top left window shows the first instance, and the bottom right shows the last.

As can be seen, the platoon is initially has a gap of 4 m, and the platoon is initially in Mode 1. This means that it is just trying to primarily save fuel and maintain a reference gap, as seen in the first window. At t = 3.88 s the leading vehicle comes within 70 m of the obstacle, and thus the platoon now switches to Mode 2. As soon as that happens, the platoon, realising that there is an obstacle in front in the same lane, starts to switch lanes to occupy a free lane. to make the lane change while being close to the reference velocity, the vehicles have to accelerate. Due to this, a larger gap opens up between the vehicles due to the vehicle accelerating first, followed by the follower vehicles one by one. The most critical and challenging part for the use case is showcased in the third window, wherein the platoon overtakes the obstacle. Due to the repulsive potential field platoon maintains a safe distance to the obstacle while also staying within the confines of the road boundaries and trying to stay as close as possible to the reference gap and velocity.

After each vehicle builds a safe gap to the obstacle behind it starts to switch back to the reference lane in the middle, as seen in the fourth window. At t = 24.16 s the platoon builds ups a big enough gap (100 m) in front of the obstacle resulting in the Mode switching back to 1 (5^{th} window). Eventually, the platoon achieves its reference gap to come back to the state it was initially (6^{th} window).

Fig. 4.4 shows the velocity profile of the vehicles where it can be seen that the platoon tries to achieve the reference gaps at the end by keeping the velocity if the green vehicle at 26 m/s. Also, other vehicles have a velocity larger (black) and smaller (yellow) than the green vehicle till the gap is achieved, which is very desirable for fuel saving. The acceleration also does not have sudden changes due to restrictions on the jerk, and the effect of slopes can also be noticed in that graph.

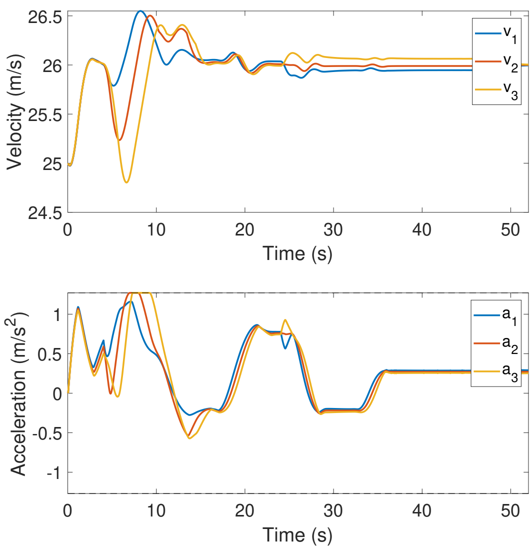

Figure 4.4: Velocity and Acceleration profile. Reconfiguration and Avoidance behavior b/w 4s and 24s

4.4.2 Going around 2 Obstacles

This use case will further build upon the previous use case of going around 1 obstacle. In the last use case simulation, we saw that the MPC formulation did well to go around one obstacle. However, the interesting part was the decision making of the MPC. The platoon has the option of going around the obstacle from the top lane or the bottom lane. However, due to minor errors in tracking, the MPC always decided to steer the platoon towards the bottom lane.

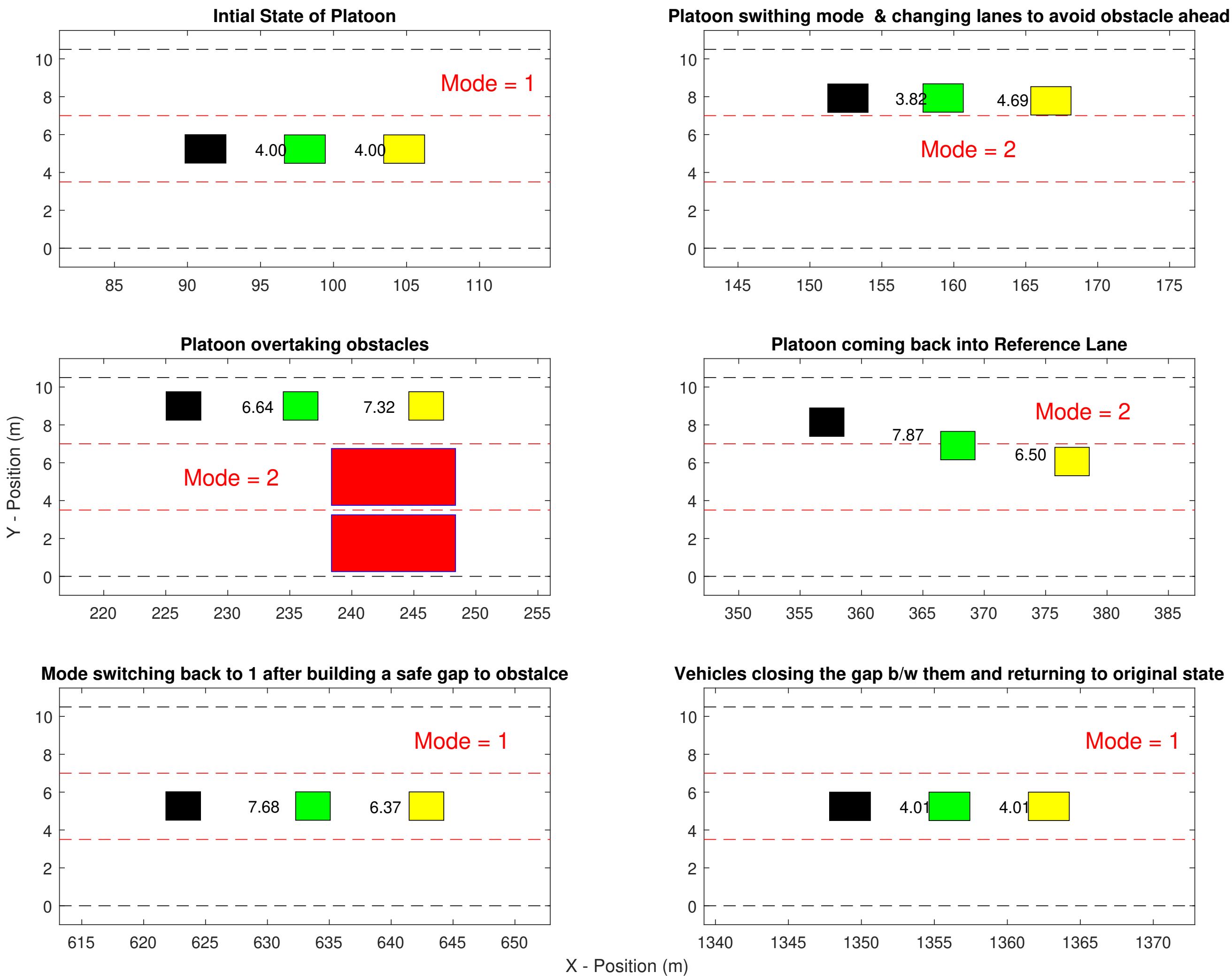

Figure 4.5: Simulation Results for 3 Trucks to go around 2 Obstacles

In this use case, we will limit the platoon's ability to go into the bottom lane and make it go into the top lane. It will be done by now having two moving obstacles run side by side while occupying the centre and the bottom lanes. Therefore, now the platoon cannot go into the bottom lane to avoid collisions and will need to go through the top lane. This way, it can be verified that lateral dynamics work well on both sides and see how well the decision making is for the algorithm.

The initial gap between vehicles remains the same, and so does the initial condition for each vehicle in the platoon. Reference lanes also remain the lanes in the middle. The obstacles also have the same initial condition, i.e. [100,5.25], and they also have the same velocity as before. Fig. 4.5 Shows the simulation results for the use case in the same order as Fig. 4.3. The window again shows the platoon in Mode 1 implementing Longitudinal Control on the platoon. Again at $t = 3.88\ s$, the platoon comes within 70 m of the obstacles and switches to Mode 2. The difference now clearly arises in window 2, wherein the platoon now steers towards the top lane instead of the bottom lane. It is what was expected of the platoon. Therefore the algorithm does well in decision making. The platoon again seamlessly overtakes both the obstacles and start steering back into the reference lane once they are far enough away in front of the obstacles (window 4). Finally, Mode switches back to 1 when the platoon is 100 m away from the obstacles. The gaps are also reduced to the reference values gradually to obtain the initial state of the platoon again.

Fig. 4.6 again shows the velocity profile and the acceleration of the vehicles. Here it can also be seen that the platoon gradually tries to attain the reference velocity while also attaining the reference gap, and the acceleration again is very smooth and again shows the effect of slopes.

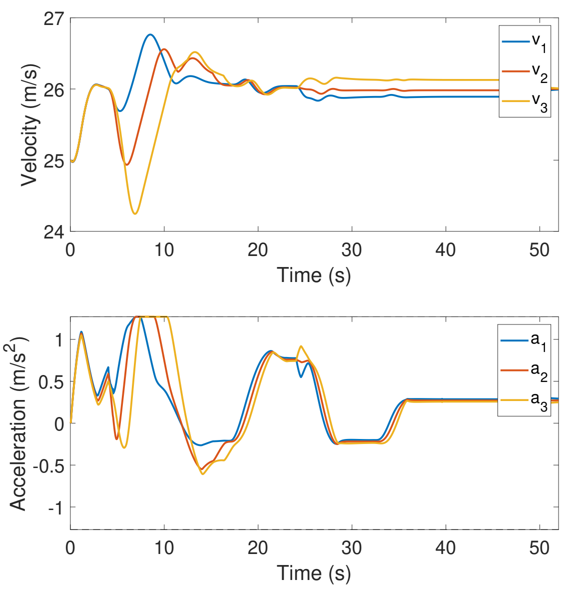

Figure 4.6: Velocity and Acceleration profile. Reconfiguration and Avoidance behavior b/w $4s$ and $24s$

4.4.3 Faster Obstacle from Behind and Slower one in front

The last use case showed platoon behaviour with 2 obstacles running side by side. In this section, the use case will show a variation of having 2 moving obstacles. Instead of having two slower obstacles in front of the platoon, one obstacle will be faster than the vehicles in the platoon and will approach the platoon from behind. The obstacle in the middle will be as it is, but the one in the bottom lane will be the faster one approaching from behind.

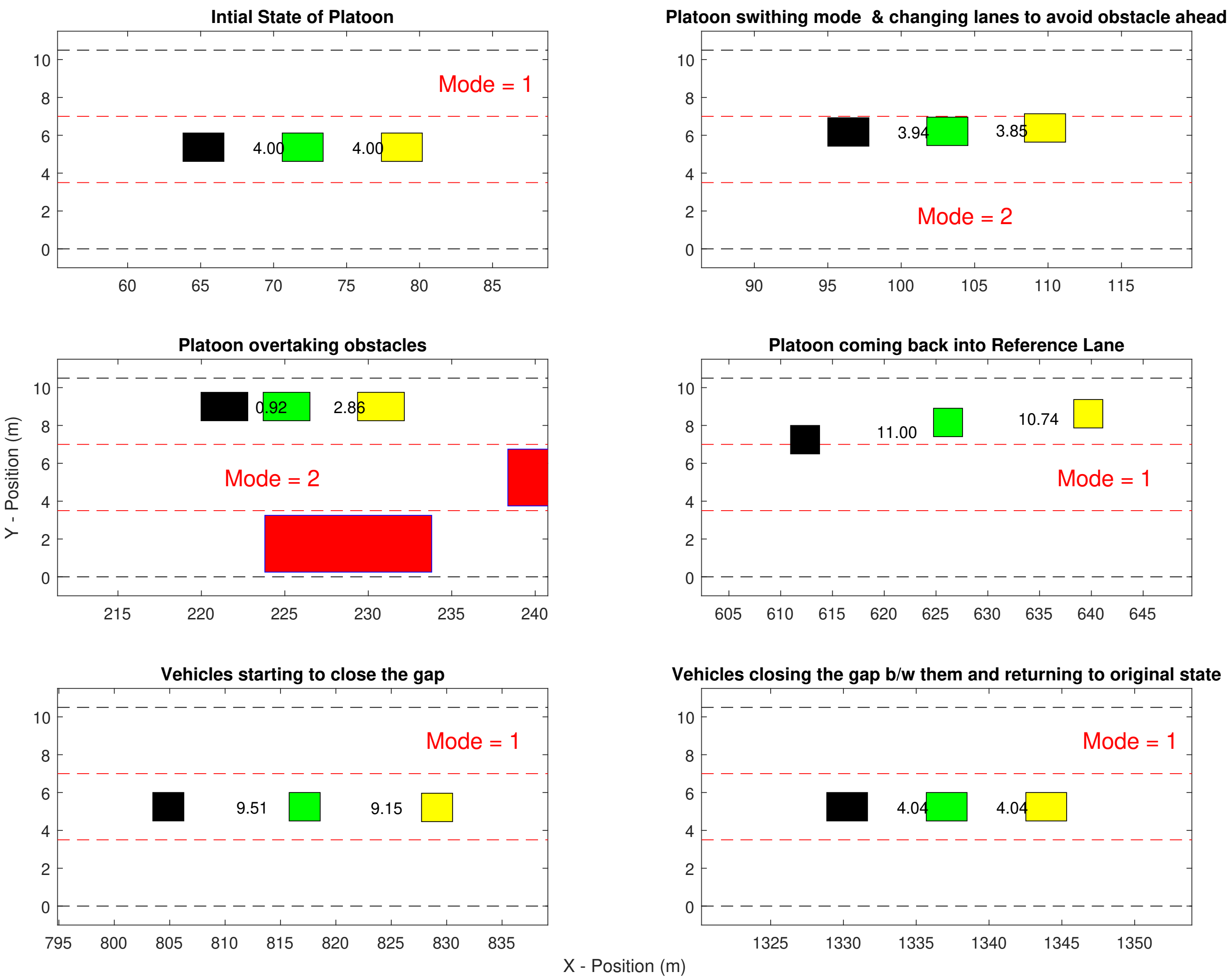

Figure 4.7: Simulation Results for 3 Trucks with Faster Obstacle from behind and Slower Obstacle in Front

For this test case, the velocities of the platoon and obstacles needed to be altered to keep the use case more realistic. The faster-moving obstacle has the initial state of [-60, 1.75] and has a high velocity if 30 m/s. The platoon now has reduced initial velocity if 20 m/s and v_{ref} also now changes to 21 m/s. Further the velocity of the vehicle in front is also decreased from 16 m/s to 10 m/s.

In this test again, we aim to test the decision making ability of the MPC solver. As the faster obstacle approaches from behind, the platoon should now again choose to take the top lane as the platoon will not be able to execute the overtake from the bottom lane and avoid the incoming obstacle. Fig. 4.7 shows the simulation results of the test case. The first window remains the same as before. Crucially, as seen in the second window the platoon again decides to steer towards the top lane as was required of it. Here, the MPC has to work a bit harder as before, both vehicles had the same behaviour (running side by side with same velocity). Now their behaviours were very different and the MPC had to deal with two different "types" of problems. The gap between the trucks (in window 2) is much smaller than the two cases before as the platoon essentially encounters two repulsive potential fields from opposite directions (front and back) until the faster obstacle overtakes the platoon. Therefore, the platoon compresses till that moment reducing the gap between the trucks.

The third window proves that the platoon cannot overtake from the bottom lane without collision and shows that the platoon can overtake from the right lane. In this case, the vehicles also have to wait longer to move back into their reference due to the potential fields from the faster vehicle in front having its effect. Therefore, the last vehicle steers first as opposed to the leading vehicle in the other two cases. The last 2 windows show the same behaviour as before.

Fig. 4.8 shows the velocity profile and acceleration of the platoon vehicle and the results are again as expected and desirable.

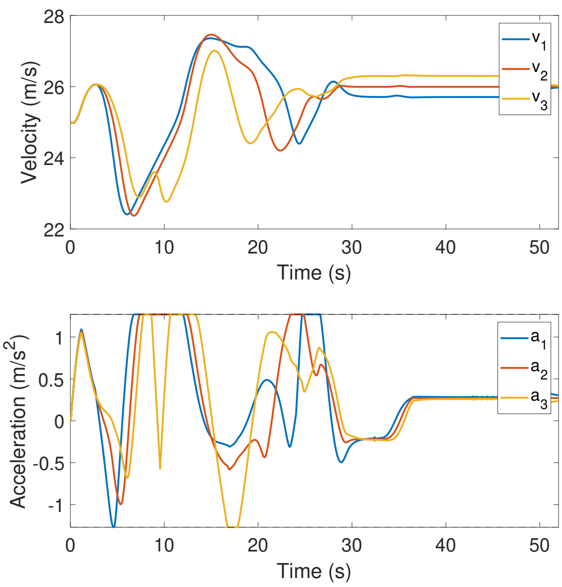

Figure 4.8: Velocity and Acceleration profile. Reconfiguration and Avoidance behavior b/w 2.6s and 24s

4.4.4 Leader Change

Finally, the last requirement from the platoon in this project was the ability to change the leader of the platoon. This behaviour is helpful in case there is a failure in the CPU of the leader vehicle, and one of the vehicles behind has a spare CPU equipped to lead the platoon autonomously. This behaviour has been designed separately from Longitudinal Control and Reconfiguration Avoidance and has been coined as Mode 0.

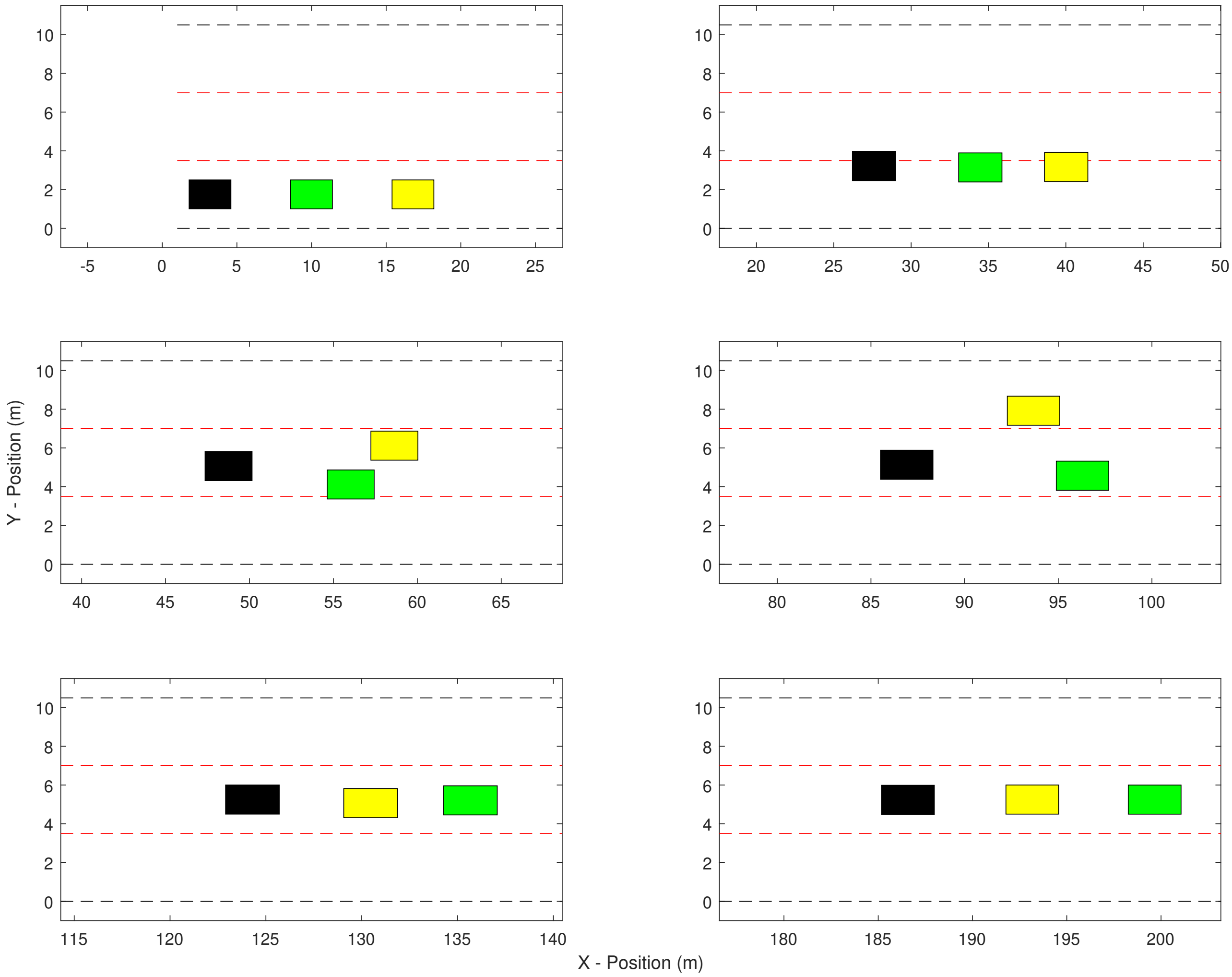

Figure 4.9: Simulation Results for 3 Trucks while changing leader

In this case, the trucks will start from the bottom lane and will change leaders while switching lanes. It is imperative to notice that this manoeuvre can only be implemented when there is no obstacle around the road is clear ($\omega_c \downarrow$). The leader change behaviour is primarily implemented by changing the formulation of the gaps between the trucks as follows

$$d_1(t) = x_1(t) - x_2(t) - l \quad \longrightarrow d_1(t) = x_2(t) - x_1(t) - l$$
$$d_2(t) = x_2(t) - x_3(t) - l \quad \longrightarrow d_1(t) = x_1(t) - x_3(t) - l$$

This way platoon now aims to switch the first and second vehicles in the platoon as now the formulation assumes that Vehicle 2 is now the leader. Tuning for this behaviour was based on giving more importance to the potential fields amongst the vehicles in the platoon ($\omega_{cd} \uparrow$). Fig. 4.9 shows the simulation for this behaviour. While changing lanes, the leading vehicle uses extra steering to go to the top lane so that the new leader can overtake it and without collisions. Vehicle 1 decelerates so that Vehicle 2 can overtake (Fig. 4.10), and Vehicle 2 accelerates longer than it can build a gap between it and Vehicle 3 in which Vehicle 1 can slot in. Vehicle 3 also decelerates at around $t = 4\ s$ to increase the gap to Vehicle 2. At this point, Vehicle 1 accelerates again to slot in smoothly.

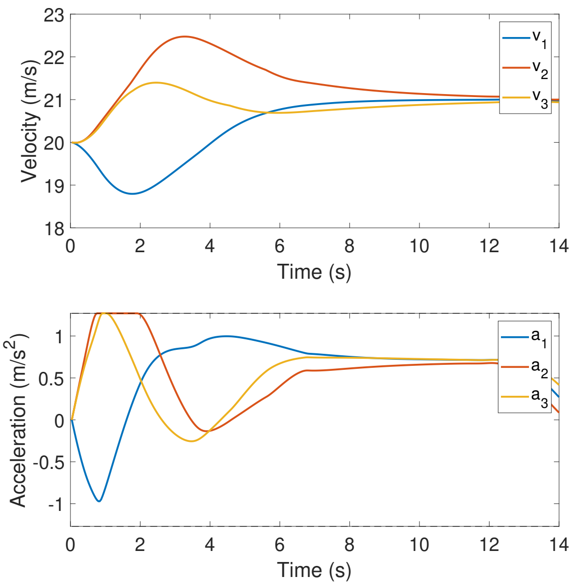

Figure 4.10: Acceleration and Velocity profile

Therefore, the MPC formulation can also implement leader changes by changing formulation for the gap between the vehicle and is a great tool to have in emergencies.

4.5 Difficult Use Case

This section discusses the limitations associated with the controller that has been designed for 2D Control of platoons. The limitations are showcased by providing reasoning regarding why the particular scenario or behavior is a limitation and then discussing what has been and what more can be done to deal with the limitation. To support what has already been done to move in the right direction w.r.t dealing with limitation, simulation results and future recommendations will also be given.

In the above test cases we witnessed that all the scenarios can be dealt with using a single set of weights that were defined in Table 4.8. There was no change in the weights during the simulation thus leading to the conclusion that the tuning weights were versatile enough to deal with the scenarios mentioned above. However, there were some other challenging test cases for which the tuning weights are not good enough and problem associated with the scenario remained unsresolved. As a result, it was required that the controller had to be tuned specifically for the test case and so it was inferred that the controller was not versatile enough to deal with these limiting test cases. Automatic weight switching outside of two modes (Mode 1 and Mode 2) is not possible for the finite state machine defined in Figure 4.2 and also it is not practical to have separate weights for particular test scenarios. For this precise reason these test cases were said to be limiting and remain unresolved by the automated controller. One such test case is discussed below.

4.5.1 Laterally Opposing Potential Fields

All the above use cases were fulfilled a single set of tuning parameters, and thus this set was deemed to be working nicely. However, the test case presented in this section acts as the limiting case towards the versatility of the tuned weights of the running cost. In this case, the platoon will attempt to travel between two slower obstacles present in the top and the bottom lane. This use case is especially challenging as the platoon will experience opposing potential fields from different directions due the obstacles being on either side of the platoon. However, unlike Section 4.4.3 where the fields were opposing in the longitudinal direction (from the front and back), this time the fields will be opposing in the lateral direction. It was expected that the intensity of the potential fields from each obstacle to the platoon should have be almost equal and opposite, resulting in the MPC solver deciding to balance the cost w.r.t both fields and moving straight. However, it decided to minimise the cost associated with one of the obstacles resulting in the platoon making the wrong decision initially to move to the top or the bottom lanes which are occupied by the obstacles which eventually led to infeasibility. Therefore, the tuning for this case had to be done separately. The weights for lane tracking were increased so that the platoon stays in lane ($\omega_v \uparrow$), and also the effect of potential fields was reduced ($\omega_c \downarrow$) so that it does not overpower everything. Further, velocity and gap tracking were also given more preference (ω_v, $\omega_d \uparrow$) so that the platoon does not deviate too much from its initial position as it is not needed. The final weights which resulted in the behavior in Fig. 4.11 are shown in Table 4.9.

Weight	**Value**
ω_y	0.9
ω_v	2
ω_d	0.9
ω_u	5
ω_a	0.35
ω_c	0
ω_{cd}	8

Table 4.9: Separate Weights for this Use Case

As seen in Fig. 4.11 the platoon makes the decision of not changing lanes just goes through the middle lane safely while avoiding collisions. Fig. 4.12 also shows velocity and acceleration which behaves as expected.

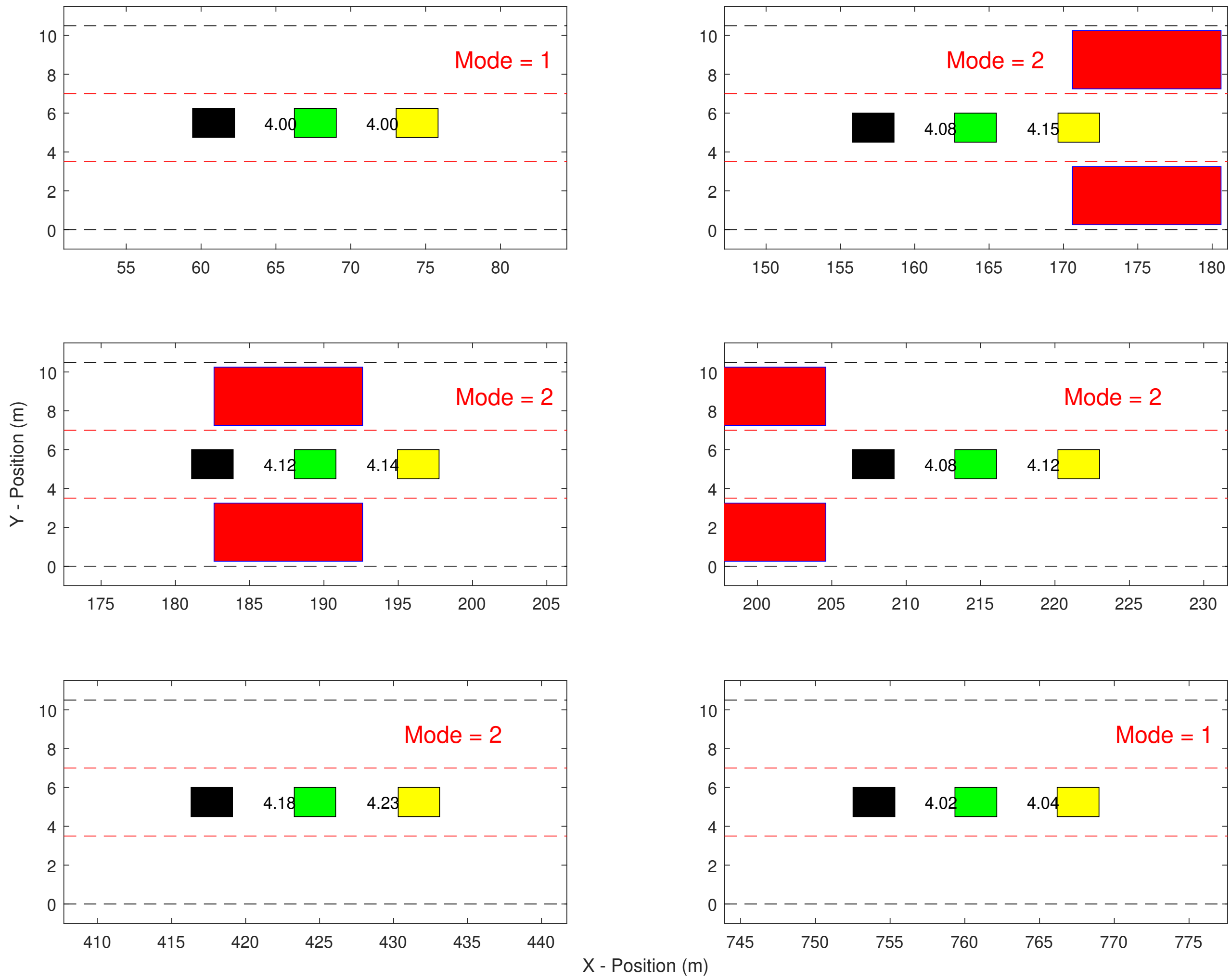

Figure 4.11: Simulation Results for 3 Trucks passing between 2 Obstacles

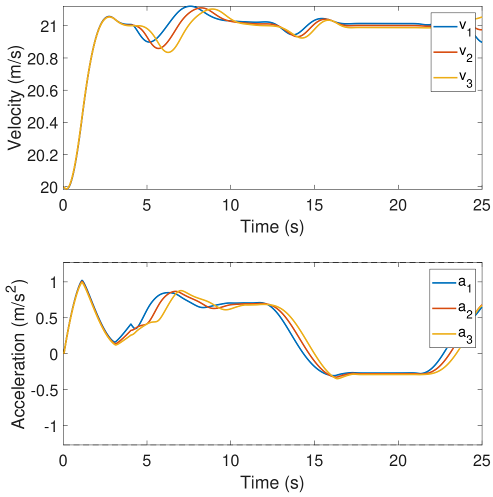

Figure 4.12: Velocity and Acceleration profile. Reconfiguration and Avoidance behavior b/w $3.2s$ and $24s$

The problem encountered in this test case can be generalised to any scenarios wherein the platoon experiences laterally opposing potential fields and therefore the set of weights defined in Table 4.9 have to be used in that. However, as mentioned above changing weights online for a particular scenario is not ideal. It also may not be possible to do so as the platoon would somehow need to identify and look out for such a scenario (wherein it has laterally opposing potential fields) separately for which complex sensor fusion systems and architectures would be required. Ideally, we would like to alter our already existing controller in some way such that it can handle this particular scenario as well. Following are two ways it can be done.

1. Instead of using Repulsive Potential Fields for collision avoidance the controller can be altered to implement avoidance using hard constraints. For implementing this, the avoidance part in the cost function associated with potential fields will need to be removed and instead non-linear constraints would be introduced into the problem such that their would be no collisions. Therefore, each vehicle in the platoon will have a non-linear constraint associated with each obstacle in the environment. A computationally efficient way to implement hard constraints is indicated in [2]. Using hard constraints would also guarantee more safety when it comes to other test cases and therefore would overall be more versatile.

2. Automated truck platoons designed by TNO for testing have a human operator in the leading vehicle of the platoon. This is done to provide additional safety to the platoon through human oversight. If this holds true, then the scenario specific mode can be designed to be triggered by the human present in the leading vehicle. When he/she encounters a scenario in front of them wherein there are two obstacles on either side of the platoon, he/she can trigger these weights by the push of a button and platoon can navigate safely between obstacles. This solution does deal with a separate kind of automation wherein the assumption regarding human oversight does hold true. This is different compared to what has been implemented in this project.

4.6 Extension: Following the Obstacle

Section 4.4.1 - 4.4.3 discussed scenarios wherein the platoon aimed to overtake the slow moving obstacles in front of the platoon. It was and effective way of travelling as the velocity of the obstacles was very slow and overtaking was the preferred decision. However, there might be cases wherein the velocity of the slower obstacle in front might be fast enough that it would be better for the platoon to follow the obstacle instead of making the "effort" of overtaking it. In addition, the platoon would also want the freedom to switch to overtaking mode (Mode 2) if the velocity of the obstacle drops below a certain threshold. This behavior is not possible with the current formulation and again new tuning weights would be needed to implement such a behaviour.

To solve this problem and to incorporate the Follower behaviour the finite state machine in Fig. 4.2 was appended instead of having a separate set of weights. The new FSM is shown in Fig. 4.13.

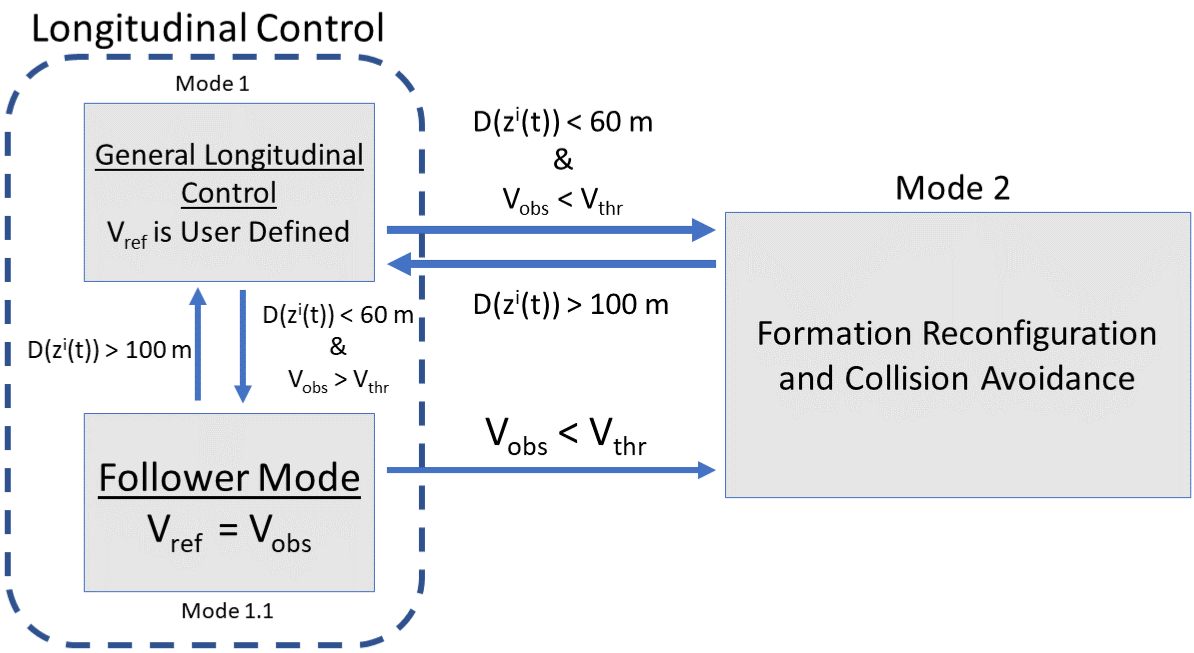

Figure 4.13: Updated Finite State Machine

As can be seen a new secondary Follower Mode is introduced in the FSM. This mode is activated when the platoon is in the vicinity of the obstacle (within 60 m) and the velocity of the obstacle (v_{obs}) is above a certain threshold (v_{thr}) which can be defined by the user. If the velocity is lower than the threshold then the platoon aims at overtaking the obstacle through Mode 2 as before. Follower Mode has the same tuning weights as the general longitudinal control mode (Mode 1) as defined in Table 4.8. The only difference is the velocity reference. For the follower mode (Mode 1.1), it is change from the user defined reference to the velocity of the obstacles. As a result, the platoon tries to match the velocity of the obstacle and thus maintains a constant a distance to the obstacle. The conditions for switching to Mode 1 remain the same. The FSM also allows for transition from Mode 1.1 to Mode 2 if the velocity of the obstacle drops below the threshold. Essentially, Mode 1 and 2 remain the same.

For simulating the use case for this formulation

$$\begin{aligned} \mathrm{v_{thr}} &= 16\,\mathrm{m/s} \\ \mathrm{v_{ref}} &= 21\,\mathrm{m/s} \\ \mathrm{v_{initial}} &= 21\,\mathrm{m/s} \\ \mathrm{v_{obs}} &= \begin{cases} 19 & 0\,\mathrm{s} \le t \le 12\,\mathrm{s} \\ 10 & 12\,\mathrm{s} \le t \le \infty \end{cases} \end{aligned}$$

Therefore, we expect the platoon to switch to mode 1.1 when it comes within 60 m of the obstacle and as soon as the velocity of the obstacle drops to 10 m/s at t = 12 s, the mode switches to 2 and the platoon looks to overtake the obstacle.

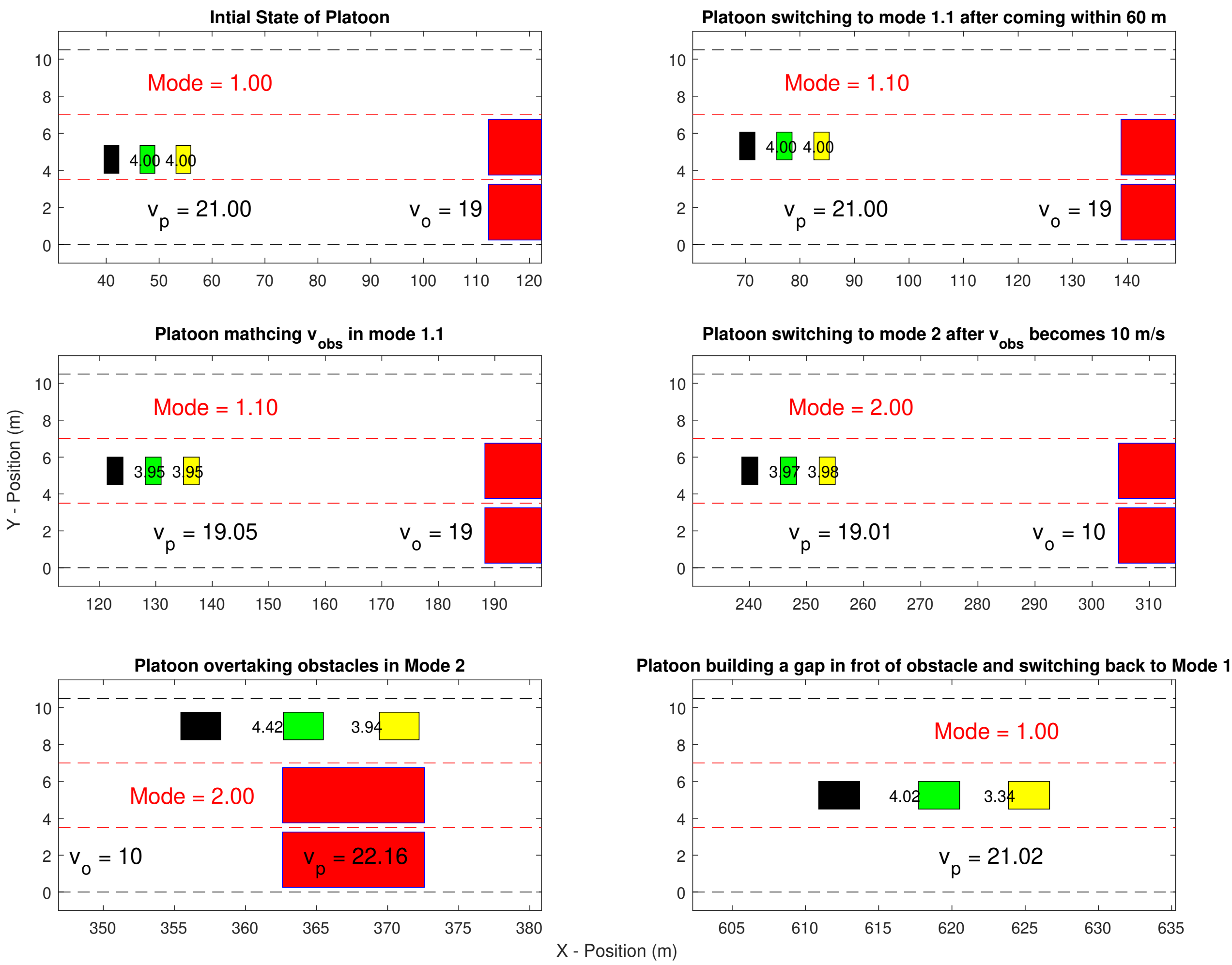

Figure 4.14: Simulation Results for Follower Mode

This is exactly what is observed in Fig. 4.14. The mode switches to 1.1 in window 2 after the platoon comes within 60 m of the obstacles. It maintains a constant distance to the obstacle as seen in window 3 and the platoon velocity ($\mathrm{v_p}$) is equal to the obstacle velocity ($\mathrm{v_o}$). The mode switches to 2 at t = 12 s as soon as the velocity of the obstacles drops below the threshold value. After this the behavior is same as section 4.4.1 and 4.4.2. The shape of the vehicles vary due to the limits on the subfigures.

After performing the simulation and looking at the results it can be concluded that the obstacle mode follower has been implemented successfully.

4.7 Conclusion

The chapter was focused on building an end to end system which can implement several behaviours depending on the environment around the platoon. Following were the conclusions

1. A different modelling approach was proposed for four-wheeled vehicles.
2. The model can incorporate external forces such as drag resistance, terrain forces and friction on the vehicle/platoon while also having lateral dynamics.
3. A novel MPC formulation was also proposed to fulfil tasks corresponding to all behaviours.
4. The formulation could incorporate the model from the previous chapter seamlessly with minimal changes.
5. The formulation for Reconfiguration and Avoidance was versatile, i.e. a single set of parameters was good enough for multiple use cases, meaning that the weights do not need to be changed for every scenario around the platoon.
6. Limitations w.r.t to the versatility of weights was also investigated in detail and after drawing certain conclusions from the limiting test cases, different solutions were suggested to deal with the limitation with minimal changes to controller that has already been developed.
7. Both behaviours were combined seamlessly, and switching between them was also smooth by varying the tuning weights during the simulation and designing a basic FSM based on certain conditions as shown in Fig. 4.2.
8. The collision avoidance also worked well when tuned adequately such that it kept a safe distance while also not violating the state and input constraints. Limits regarding avoidance was also discussed in section 4.5.1 and recommendations on how to deal with these limitations were also provided.

Therefore, it can be concluded that the model modelling technique and MPC formulation work well.

.

CHAPTER 5

CONCLUSION

This thesis project aimed at primarily bridging the gap between two disjointed research fields for vehicle platoons. After going through the literature, it was evident that there was no single approach that dealt with all the different behaviours required to implement an end to end system for platoon control. The literature either focused on Longitudinal Control and optimizing fuel or focused on reconfiguration, lanes changes and avoidance and optimizing that.

Initially, it was thought that both behaviours could be combined by having two separate models and two different MPC formulations and switching between them when needed. However, this would be very cumbersome as the algorithm would have to change everything about the problem while switching, which would be computationally expensive and might also have implications on the system's stability as switching MPCs is not always safe. After diving into the literature of Dynamic Bicycle Model, it also became clear that it would be possible to do everything needed using one model and one MPC formulation.

First, the system for Longitudinal Control was developed and tested based on different parameters, and it was concluded that the system and algorithm worked well. Then this system was integrated with the lateral dynamics of the Bicycle Model to obtain an end to end model and consequently a novel MPC formulation which also worked well. The system performed well for several different use cases and its abilities were tested thoroughly, and tuning weights were also studied.

5.1 Limitations and Recommendations for Future Work

1. The fuel model is used in this project is not as detailed as it can be and relies on regression. More elaborate or physics based fuel models can be used, which also consider gear ratios the vehicle is using. This will give more accurate readings of how much fuel is being used and as a result more fuel can also be saved using this information.

2. As mentioned in Chapter 3, the aerodynamic drag modelled here does not depend on the distance between the vehicles whereas in reality that is the case and therefore more involved drag models should be used. Using them would also test the ability of the MPC formulation in the platoon to close the gap based on reduction in drag resistance and will again be closer to reality.

3. The model does not accommodate tyre forces. After accommodating these forces in the model, tests can be carried on how the vehicle acts on different weather conditions such as rain, wherein the friction force will come down, and tyre forces will be different.

4. The communication between the trucks in the platoon has been assumed to be instantaneous or in other words it is assumed that the process of the state information of each truck being sent to the controller and the the controller sending back the optimal outputs, takes less than 0.04 s, i.e. the discretization step size. In reality the communication and the computation of the inputs cannot be this fast. Therefore, the system cannot deal with problems wherein the communication and computation process takes longer than the discretization step of the system. The recommendation here would be to model a delay in communication and then accommodating the the worst case delay into the dynamics so that the system becomes more robust to the delays in real systems.

5. Further, the simulation can also be performed using high fidelity simulation software such as Sim Mechanics. It would also lead to testing for a model mismatch between the model used in the algorithm and the high fidelity model, e.g. Sim Mechanics. For this project the simulation and computation is done fully in MATLAB and therefore there is no knowledge how the algorithm will act when applied to a real and more convoluted system.

6. Collision Avoidance can also be implemented using hard constraints instead of potential fields that would provide more assurance when it comes to avoidance. Repulsive fields can be tuned well to avoid collisions but they can never guarantee avoidance like hard constraints. As mentioned in section 4.5.1, having hard constraints will also give the platoon more versatility w.r.t dealing with a wider range of use cases.

.

Appendices

APPENDIX A

SOLVER

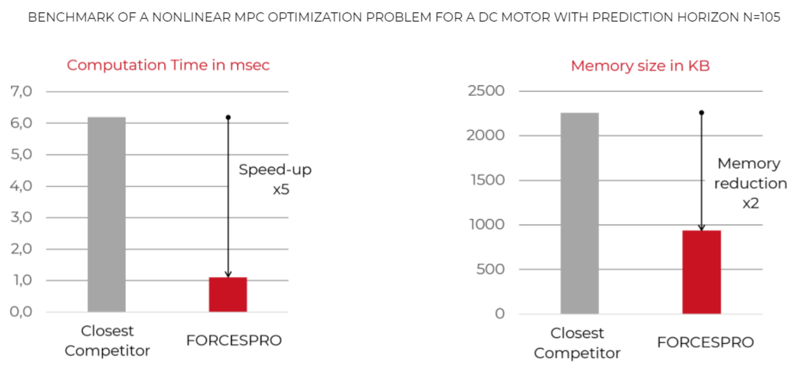

Figure A.1: Example of the performance advantage of FORCESPRO

The primary requirement for the solver being used for this project was for it to be relatively fast and in solving in non-linear MPC optimization problems and that it can be used seamlessly in the future for real world testing. Keeping this in mind FORCESPRO has been used for solving MPC problems in this thesis. FORCESPRO enables users to generate tailor-made solvers from a high-level mathematical description of an optimization problem. The numerical software is designed specifically for the purpose of fast, embedded optimization. The algorithms are developed specifically for the mathematical structure of optimal control problems, which makes it the fast solver. The solvers generated have a very small code size that can be embedded on any hardware platform. Most importantly FORCESPRO can be used with MATLAB thus there being no need to learn a new language altogether. In addition to generating the solver in MATLAB, FORCESPRO also generates a SIMULINK block for the same which can be used instantly.

However, the platform also has its disadvantages. Although FORCESPRO can be used with MATLAB the user needs to download the whole client from the server and put effort into integrating this client/solver with MATLAB and the user needs to learn several commands. There are also several different parameters in which effect the MPC formulation so the user also needs to get a good grasp of what each of these parameters/functions do. FORCESPRO also requires the problem to be scaled properly i.e. if the states and inputs having varying magnitudes then solver would like them to be scaled well before use otherwise it may run into computation problem. Finally, the debugging process can also be a bit cumbersome as the user does not know how the solver is being generated and he/she is also unaware of the of the details of how the solver works inside.

BIBLIOGRAPHY

[1] String-stable cacc design and experimental validation, a frequency-domain approach. *IEEE Transactions on Vehicular Technology*, 59(9):4268–4279, 2010. ISSN 0018-9545. doi: 10.1109/TVT.2010.2076320. 4

[2] Bruno Brito, Boaz Floor, Laura Ferranti, and Javier Alonso-Mora. Model predictive contouring control for collision avoidance in unstructured dynamic environments. *IEEE Robotics and Automation Letters*, 4(4):4459–4466, 2019. doi: 10.1109/LRA.2019.2929976. 54

[3] C. Chang and Z. Yuan. Combined longitudinal and lateral control of vehicle platoons. In *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, pages 848–852, 2017. doi: 10.1109/ICCSEC.2017.8446824. 11

[4] J. Chen, W. Zhan, and M. Tomizuka. Autonomous driving motion planning with constrained iterative lqr. *IEEE Transactions on Intelligent Vehicles*, 4(2):244–254, 2019. doi: 10.1109/TIV.2019.2904385. 11

[5] Linying Chen, Hans Hopman, and Rudy R. Negenborn. Distributed model predictive control for vessel train formations of cooperative multi-vessel systems. *Transportation Research Part C: Emerging Technologies*, 92:101 – 118, 2018. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2018.04.013. URL http://www.sciencedirect.com/science/article/pii/S0968090X18305011. 5

[6] Mooryong Choi and Seibum B. Choi. Model predictive control for vehicle yaw stability with practical concerns. *IEEE Transactions on Vehicular Technology*, 63(8):3539–3548, 2014. doi: 10.1109/TVT.2014.2306733. 5

[7] P. Darby and R. Gottumukkala. Decentralized computing techniques in support of cyber-physical security for electric and autonomous vehicles. In *2019 IEEE Green Technologies Conference(GreenTech)*, pages 1–5, 2019. doi: 10.1109/GreenTech.2019.8767163. 2

[8] Christian Earnhardt, Ben Groelke, John Borek, and Chris Vermillion. Hierarchical model predictive control approaches for strategic platoon engagement of heavy-duty trucks.

IEEE Transactions on Intelligent Transportation Systems, pages 1–13, 2021. doi: 10.1109/TITS.2021.3076963. 5

[9] Eurostat. Freight transport statistics. URL `https://ec.europa.eu/eurostat/statistics-explained/index.php/Freight_transport_statistics`. iv, 1

[10] A. Ferdowsi, S. Ali, W. Saad, and N. B. Mandayam. Cyber-physical security and safety of autonomous connected vehicles: Optimal control meets multi-armed bandit learning. *IEEE Transactions on Communications*, 67(10):7228–7244, 2019. doi: 10.1109/TCOMM.2019.2927570. 2

[11] Roya Firoozi, Laura Ferranti, Xiaojing Zhang, Sebastian Nejadnik, and Francesco Borrelli. A distributed multi-robot coordination algorithm for navigation in tight environments, 2020. 2, 5, 9, 10, 34

[12] Roya Firoozi, Xiaojing Zhang, and Francesco Borrelli. Formation and reconfiguration of tight multi-lane platoons. *Control Engineering Practice*, 108:104714, 2021. ISSN 0967-0661. doi: https://doi.org/10.1016/j.conengprac.2020.104714. URL `https://www.sciencedirect.com/science/article/pii/S0967066120302847`. 2, 5, 34

[13] G. Guo and Q. Wang. Fuel-efficient en route speed planning and tracking control of truck platoons. *IEEE Transactions on Intelligent Transportation Systems*, 20(8):3091–3103, 2019. doi: 10.1109/TITS.2018.2872607. 3

[14] Roland E. Haas and Dietmar P. F. Möller. Automotive connectivity, cyber attack scenarios and automotive cyber security. In *2017 IEEE International Conference on Electro Information Technology (EIT)*, pages 635–639, 2017. doi: 10.1109/EIT.2017.8053441. 2

[15] P. Hamelin, P. Bigras, J. Beaudry, P. Richard, and M. Blain. Discrete-time state feedback with velocity estimation using a dual observer: Application to an underwater direct-drive grinding robot. *IEEE/ASME Transactions on Mechatronics*, 17(1):187–191, 2012. doi: 10.1109/TMECH.2011.2154338. 11

[16] L. L. Hoberock. A Survey of Longitudinal Acceleration Comfort Studies in Ground Transportation Vehicles. *Journal of Dynamic Systems, Measurement, and Control*, 99 (2):76–84, 06 1977. ISSN 0022-0434. doi: 10.1115/1.3427093. URL `https://doi.org/10.1115/1.3427093`. 25

[17] W.-H Hucho and G Sovran. Aerodynamics of road vehicles. *Annual Review of Fluid Mechanics*, 25:485–537, 11 2003. doi: 10.1146/annurev.fl.25.010193.002413. 34

[18] P. A. Ioannou and C. C. Chien. Autonomous intelligent cruise control. *IEEE Transactions on Vehicular Technology*, 42(4):657–672, 1993. doi: 10.1109/25.260745. 4

[19] M. Islam and M. Okasha. A comparative study of pd, lqr and mpc on quadrotor using quaternion approach. In *2019 7th International Conference on Mechatronics Engineering (ICOM)*, pages 1–6, 2019. doi: 10.1109/ICOM47790.2019.8952046. 12

[20] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen. A survey on platoon-based vehicular cyber-physical systems. *IEEE Communications Surveys Tutorials*, 18(1):263–284, 2016. doi: 10.1109/COMST.2015.2410831. 2

[21] V. K, M. Ambalal Sheta, and V. Gumtapure. A comparative study of stanley, lqr and mpc controllers for path tracking application (adas/ad). In *2019 IEEE International Conference on Intelligent Systems and Green Technology (ICISGT)*, pages 67–674, 2019. doi: 10.1109/ICISGT44072.2019.00030. 12

[22] Anan Kaku, Md. Abdus Samad Kamal, Masakazu Mukai, and Taketoshi Kawabe. Model predictive control for ecological vehicle synchronized driving considering varying aerodynamic drag and road shape information. *SICE Journal of Control, Measurement, and System Integration*, 6(5):299–308, 2013. doi: 10.9746/jcmsi.6.299. URL https://doi.org/10.9746/jcmsi.6.299. 5, 16, 17, 22

[23] Shivaram Kamat. Lane keeping of vehicle using model predictive control. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pages 1–6, 2019. doi: 10.1109/I2CT45611.2019.9033958. 5

[24] T. Keviczky, K. Fregene, F. Borrelli, G. J. Balas, and D. Godbole. Coordinated autonomous vehicle formations: decentralization, control synthesis and optimization. In *2006 American Control Conference*, pages 6 pp.–, 2006. doi: 10.1109/ACC.2006.1656517. 5

[25] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099, 2015. doi: 10.1109/IVS.2015.7225830. 5

[26] Jean-Claude Latombe. *Potential Field Methods*, pages 295–355. Springer US, Boston, MA, 1991. ISBN 978-1-4615-4022-9. doi: 10.1007/978-1-4615-4022-9_7. URL https://doi.org/10.1007/978-1-4615-4022-9_7. 38

[27] Aaron Lelouvier, Jacopo Guanetti, and F. Borrelli. Eco-platooning of autonomous electrical vehicles using distributed model predictive control. 2017. 5

[28] J. Liu, B. Pattel, A. S. Desai, E. Hodzen, and H. Borhan. Fuel efficient control algorithms for connected and automated line-haul trucks. In *2019 IEEE Conference on Control Technology and Applications (CCTA)*, pages 730–737, 2019. doi: 10.1109/CCTA.2019.8920650. 3

[29] Bogdan Marcu and Fred Browand. Aerodynamic forces experienced by a 3-vehicle platoon in a crosswind. In *SAE Technical Paper*. SAE International, 03 1999. doi: 10.4271/1999-01-1324. URL https://doi.org/10.4271/1999-01-1324. 16

[30] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, 2016. doi: 10.1109/TIV.2016.2578706. 2, 11, 12

[31] R. Pfiffner, Lino Guzzella, and Christopher Onder. Fuel-optimal control of cvt powertrains. *IFAC Proceedings Volumes*, 34:17–22, 03 2001. doi: 10.1016/S1474-6670(17)34371-9. 20

[32] J. Ploeg. Cooperative vehicle automation: Safety aspects and control software architecture. In *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*, pages 6–6, 2017. doi: 10.1109/ICSAW.2017.69. 3

[33] J. Ploeg, N. van de Wouw, and H. Nijmeijer. Lp string stability of cascaded systems: Application to vehicle platooning. *IEEE Transactions on Control Systems Technology*, 22(2):786–793, 2014. doi: 10.1109/TCST.2013.2258346. 4, 16

[34] R. Rajamani and C. Zhu. Semi-autonomous adaptive cruise control systems. *IEEE Transactions on Vehicular Technology*, 51(5):1186–1192, 2002. doi: 10.1109/TVT.2002.800617. 4

[35] Rajesh Rajamani. *Lateral Vehicle Dynamics*, pages 15–46. Springer US, Boston, MA, 2012. ISBN 978-1-4614-1433-9. doi: 10.1007/978-1-4614-1433-9_2. URL https://doi.org/10.1007/978-1-4614-1433-9_2. 5, 10, 11

[36] J.B. Rawlings, D.Q. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017. ISBN 9780975937730. URL https://books.google.nl/books?id=MrJctAEACAAJ. 12

[37] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin. The security of autonomous driving: Threats, defenses, and future directions. *Proceedings of the IEEE*, 108(2):357–372, 2020. doi: 10.1109/JPROC.2019.2948775. 2

[38] Green Car Report. Saving gas by lifting the pedal:engine braking vs coasting. URL https://www.greencarreports.com/news/1113597_saving-gas-by-lifting-the-pedal-engine-braking-vs-coasting-video. 3, 16

[39] F. Rey, Z. Pan, A. Hauswirth, and J. Lygeros. Fully decentralized admm for coordination and collision avoidance. In *2018 European Control Conference (ECC)*, pages 825–830, 2018. doi: 10.23919/ECC.2018.8550245. 5

[40] Payman Shakouri, A. Ordys, Paul Darnell, and Peter Kavanagh. Fuel efficiency by coasting in the vehicle. *International Journal of Vehicular Technology*, 2013, 08 2013. doi: 10.1155/2013/391650. 3, 16

[41] Mohamed Shawky. Factors affecting lane change crashes. *IATSS Research*, 44(2):155 – 161, 2020. ISSN 0386-1112. doi: https://doi.org/10.1016/j.iatssr.2019.12.002. URL http://www.sciencedirect.com/science/article/pii/S0386111219300020. 34

[42] Aman Singh and Madhusudan Singh. An empirical study on automotive cyber attacks. In *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pages 47–50, 2018. doi: 10.1109/WF-IoT.2018.8355124. 2

[43] Lun Tsuei and Ömer Savaş. Transient aerodynamics of vehicle platoons during in-line oscillations. *Journal of Wind Engineering and Industrial Aerodynamics - J WIND ENG IND AERODYN*, 89:1085–1111, 10 2001. doi: 10.1016/S0167-6105(01)00073-3. 16

[44] S. Tsugawa, S. Jeschke, and S. E. Shladover. A review of truck platooning projects for energy savings. *IEEE Transactions on Intelligent Vehicles*, 1(1):68–77, 2016. doi: 10.1109/TIV.2016.2577499. 3

[45] V. Turri, B. Besselink, and K. H. Johansson. Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning. *IEEE Transactions on Control Systems Technology*, 25(1):12–28, 2017. doi: 10.1109/TCST.2016.2542044. 2

[46] V. Turri, B. Besselink, and K. H. Johansson. Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning. *IEEE Transactions on Control Systems Technology*, 25(1):12–28, 2017. doi: 10.1109/TCST.2016.2542044. 5, 16, 34

[47] Stanford University. Receding horizon control: Automatic generation of high-speed solvers. URL `https://web.stanford.edu/~boyd/papers/code_gen_rhc.html`. 12

[48] V. Vohra, M. Wahba, G. Akarslan, R. Ni, and S. Brennan. An examination of vehicle spacing to reduce aerodynamic drag in truck platoons. In *2018 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1–6, 2018. doi: 10.1109/VPPC.2018.8604977. 3

[49] Junjie Wang, Hongsai Zhu, Jingyu Xu, and Yan Su. Linear time-varying mpc-based steering controller for vehicle trajectory tracking considering the effect of road topography. In *2020 5th International Conference on Control, Robotics and Cybernetics (CRC)*, pages 256–260, 2020. doi: 10.1109/CRC51253.2020.9253449. 5

[50] Wikipedia. Platoons (automobile). URL `https://en.wikipedia.org/wiki/Platoon_(automobile)`. 2

[51] P. Xavier and Y. Pan. A practical pid-based scheme for the collaborative driving of automated vehicles. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 966–971, 2009. doi: 10.1109/CDC.2009.5400734. 11