Otto-von-Guericke-University Magdeburg
Faculty of Electrical Engineering and Information Technology
Chair for Electromagnetic Compatibility

Non-Technical Project

Model Predictive Based Controller for Unmanned Ground Vehicle

Submitted on: January 16, 2022

Submitted by: Shailesh Kumar (229916)
born on May 05, 1996
in Raxaul-Bihar, India

Supervised by: M.Sc. Mohamed Soliman, Research Assistant Institute for Automation Engineering (IFAT)

Lehrstuhl für
Systemtheorie und Regelungstechnik
Prof. Dr.-Ing. Rolf Findeisen

Non-Technical Project

Matrikel-Nr: 229916

Shailesh Kumar

Model Predictive Based Controller for Unmanned Ground Vehicle

Herr Kumar hat dieses Forschungsprojekt mit großer Eigenständigkeit und sehr guten Ergebnissen bearbeitet.

Die Betreung erfolgte durch M.Sc. Mohamed Soliman

Die Leistung wird mit **1.3** bewertet.

Magdeburg, January 16, 2022

Prof. Dr.-Ing. Rolf Findeisen
Magdeburg, January 16, 2022

Abstract

The increasing day-to-day population leads to busy life schedules and availing more comfortable requirements, which forced to make advanced technical innovation in automation and control. Considering the present global scenario, the world is rapidly moving towards autonomous vehicles. Whether it is ground or air vehicle, the unmanned vehicle brings the revolution to peoples´ life. With the evolution of advanced technologies nowadays, more complicated tasks are demanded from these systems. The critical importance in designing such an autonomous system is to build a mathematical model that is promising enough during the controller design stage. To this end, this paper describes a mathematical model for an unmanned ground vehicle called a Hamster. To achieve the autonomy of the car, an overview of the Model Predictive Controller (MPC) is also presented. The efficacy of the MPC is illustrated via simulation results using a simple car model.

Objectives of the report:

The main objectives of this reports are:

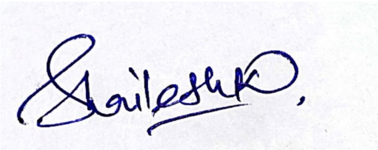
1. Learn how to read critically and learn how to tackle the problem through the literature review
2. Learn and enhance the scientific writing skills
3. Be familiar with a professional writing tool LaTeX
4. Learn how to formulate an optimal control problem using ACADO
5. Improve some programming skills using Matlab

Declaration by the candidate

The report entitled **Model Predictive Based Controller Unmanned Ground Vehicle** is conducted under the supervision of Mohamed Soliman, Research Assistant of Institute for Automation Engineering (IFAT) Laboratory for Systems Theory and Automatic Control Otto-von-Guericke University Magdeburg 39106 Magdeburg - Germany.

I hereby declare that this Non-Technical Project (NTP) thesis is my own work and effort and has not been submitted anywhere for any award. Where other sources of information have been used, they have been marked.
The work has not been presented in the same or a similar form to any other testing authority and has not been made public.

Magdeburg, January 16, 2022
SHAILESH KUMAR



Contents

Nomenclature

v	Velocity of the vehicle
a	Acceleration of the vehicle
u	Optimal control input
T_s	Sample time
J	Cost function
$\dot{x}$	State
y	Output
Q, R	Weighting matrices
R^n	Region with n dimension
D	Subset
x, y	Current position of the hamster
x_0, y_o	Position of the obstacle
R	Radius of the obstacle influence
K	Positive scaling factor
N_p	Prediction horizon
T_f	Simulation time
x_{1int}	Initial car position
x_{2int}	Initial car velocity
$X - Y - Z$	Coordinate system
x_1	Position constraints
x_{1ref}	Final car position
x_2	Velocity constraints

x_{2ref}	Final car velocity
δ_f	Steering angle
ψ	Heading angle
β	Slip angle
L	length of the vehicle
l_f	Distance between the front wheel to the center of vehicle
l_r	Distance between the rear wheel to the center of vehicle
w	Yaw angle

List of Acronyms

NTP Non-Technical Project

UGV Unmanned Ground Vehicle

UGVs Unmanned Ground Vehicles

MPC Model Predictive Control

PVC Poly vinyl chloride

ACADO Automatic Control and Dynamic Optimization

DARPA Defence Advanced Research Project Agency

RHPC Receding Horizon Predictive Control

EOD Explosive Ordnance Disposal

NMPC Nonlinear Model Predictive Control

GRRC Ground Robotics Reliability Center

PID Proportional Integral Derivative

AUGV Autonomous Unmanned Ground Vehicle

ROS Robot Operating System

CoG Center of Gravity

RCA Radio Corporation of America

MIMO Multi-input Multi-output

ESP Electronic Stability Program

SAE Society of Automotive Engineers

LIDAR Light Detection and Ranging

OCP Optimal Control Problem

SLAM Simultaneous Localization and Mapping

List of Figures

List of Tables

1 Introduction

Unmanned Ground Vehicle (UGV) has been in development for decades. The first UGV was developed in the 1930s decades for military purposes [1]. Over the period of time, many universities and research institutes found new UGV technologies for commercial use, like transportation services. The development of Unmanned Ground Vehicles (UGVs) has a promising application in industry and society, such as traffic congestion, pollution, accidents, and human casualties. Generally, the autonomous vehicle can be classified according to its structure, capability, environment, or operation, such as ground, water, and aerial vehicles. Autonomous vehicles have recently sparked a surge of interest in the academic, civilian, and research communities as essential solutions to various applications. Research on the autonomous vehicle has mainly focused on three main layers; perception, planning, and control [2]. Each of these layers presents very different specifications and is often addressed separately. This literature report aims to give an overview of the model-based predictive controller model for a UGV Model.
This reported work has also been motivated by the overwhelming applications of unmanned ground vehicle. **Chapter-1** covers the detailed literature survey of UGV with various applications. A short introduction of Model Predictive Control (MPC) has been given. **Chapter-2** covers the mathematical model of Unmanned Ground Vehicle (UGV) where the kinematics bicycle model has been chosen over the dynamic bicycle model and then a brief introduction of the Hamster has been presented with the mathematical equations. **Chapter-3** consists of a detailed study of MPC and their advantages/disadvantages. A Mathematical model of the system dynamic has been presented and the Automatic Control and Dynamic Optimization (ACADO) solver has been discussed as a proposed solver to our optimization problem. The Simulation result for the double integrator UGV model has been represented with the help of the MATLAB software package and the final outcome has been discussed in **Chapter-4**. AT the end, **Chapter-5** summarizes the final conclusion of the report. The future objective of the task and scope of the UGV has been briefly discussed with this the final goal of the project has been achieved.

1.1 General overview of Unmanned Ground Vehicle (UGV)

An unmanned ground vehicle (UGV) is any segment of mechanized equipment that travels across the ground surface and carries anything without the need for human intervention [3]. The vehicle will be equipped with sensors to monitor its surroundings. It will either make decisions independently or transfer the information to a human operator at a different location who will act on it by teleoperation (remote operation).
A vehicle powered by a human operator through an interface is known as a remote-operated UGV. The operator decides on all activities based on either direct visual observation or remote sensor use, such as digital video cameras. A remote-controlled toy car is a simple example of remote operation. The existence of UGV systems needs some organizing principles, and in fact, the taxonomy of UGV systems could be based on the following characteristics for each system [4].

- The aim of the development project (often, but not always, the performance of some application-specific mission).
- The reasons for selecting a UGV solution for this application are as follows: (e.g., hazardous environment, strength or endurance requirements, and size limitation).
- Various technical challenges; in terms of functionality, performance, or cost posed by the application.
- The planned operating area of the device (e.g., indoor environments, anywhere indoors, outdoors on roads, general cross-country terrain, etc.)
- The vehicle's locomotion mode (e.g., wheels, tracks, or legs)
- How the path of the vehicle is determined (i.e., control and navigation techniques employed).

1.1.1 Importance of an autonomous driving vehicle

Considering the present global scenario, the world is rapidly moving towards autonomous vehicles. Whether it is a ground or air vehicle, an unmanned vehicle brings a revolution in peoples' life. The increasing day-to-day population leads to a busy life schedule and availing more comfortable requirements forced to make advanced technical innovation in the field of autonomous driving. UGVs have many applications that play a vital role in today's life. Ranging from civilians to military missions such as Explosive Ordnance Disposal (EOD), reconnaissance, surveillance and combat; industrial and home usage, such as cleaning floors and harvesting crops; to special tasks such as rescue operations. UGVs, therefore, has drawn interest from many researchers and organizations, especially to the military, since the 1960s. The demand for them is ever-increasing [5].

Figure 1.1: The Multi-function Logistics and Equipment robot

There is a wide variety of UGVs are in use today. The versatile application and effective operation of UGV have drawn the world's eyes to it. The major talk about the UGV is that it follows the human command to achieve the target under various constraints. It reduces the undesired complexity which may cause by manned driving vehicles. Also, the need of the driver will be sorted out. We can see in Figure(1.1) [6]. It has been used for various purposes. There are numerous reasons for the successful future of the UGV.

- It avoids the need of a driver and also follows the given instruction very accurately, far better than human control.
- It can smoothly deal with the obstacle and follows the fed data perfectly.
- Repair itself without outside assistance.
- Work for an extended period of time without the need for human interference.
- It has self-optimization features that utilize and get the optimal solution as per the model.

1.2 A short history of Unmanned Ground Vehicle (UGV)

In October 1921, the first remote-controlled car was registered, which issued by Radio Corporation of America (RCA)'s Worldwide wireless magazine [7]. The car was driver-less and controlled wireless through radio, as shown in Figure(1.2) [7]. In the 1930s, the Soviet Union produced Teletanks, a machine gun-armed tank that could be operated remotely by radio from another tank. These were used in the II-world war(1939-1945). During the same era, the British developed a radio-controlled version of their Matilda II infantry tank, named "Black Prince," in 1941. The tank was developed to draw the fire of concealed anti-tank guns or demolition missions. Nevertheless, due to the costs of converting the tank's transmission system to Wilson-type gearboxes, it did not get success [4].

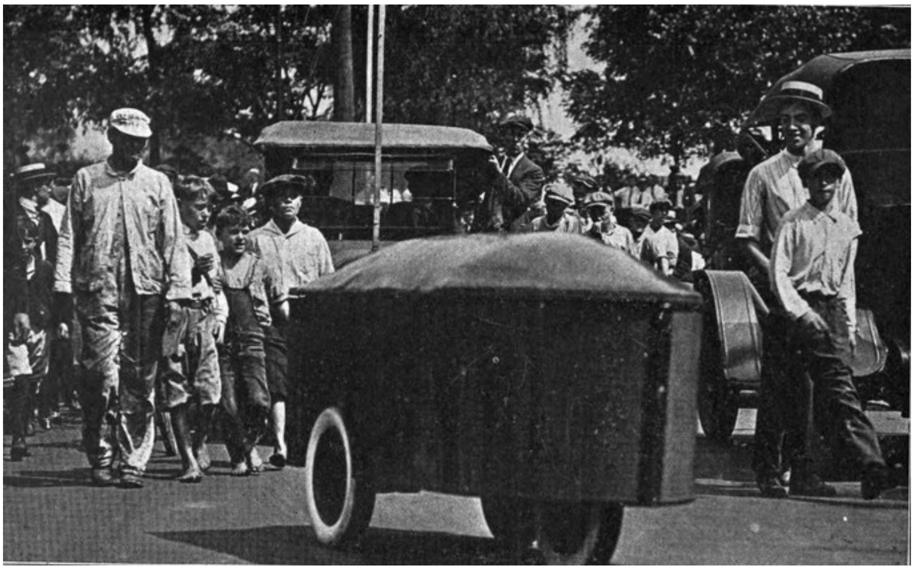


Figure 1.2: A radio controlled car, Ohio 1921

Shakey, the first significant mobile robot creation effort, was developed as a Defence Advanced Research Project Agency (DARPA) research study in the 1960s. It was a wheeled platform with a TV camera, sensors, and a computer to assist it in navigating its navigational tasks of picking up wooden blocks and putting them in specific areas based on commands. DARPA subsequently creates a number of semi-autonomous ground robots as well as the autonomous land vehicle. The first UGV was capable of fully autonomous navigation on and off roads at valuable speeds [7]. Due to the increasing importance of an autonomous vehicle and their applications, many national laws and regulations have been delivered to organized civil usage of UGV.

1.2.1 Applications of UGV

UGVs are being designed for civilian and military applications to perform unique, challenging, and dangerous tasks. They proved to be successful in various situations where using human labor would be too costly, dangerous, or impractical. Research and development of UGVs is often focused on specific applications and are therefore designed accordingly. For example, UGVs are broadly used in the military to transport weapon carrier, detection of warfare agents, and as a firefight vehicle, as shown in the Figure(1.3) [8].
The various field where UGV has been broadly used are Commercial and motion picture filming [9], Research and rescue [10], Forest fire detection/monitoring [11], Domestic policing and patrolling [12], Transport of goods [13]. Unmanned ground vehicles will

Figure 1.3: Unmanned ground vehicle

typically contain components such as a platform, sensors, control systems, communication links, and a guidance interface, depending on their application [14].
The UGV systems can be designed as per various applications. For example;

1. Space Applications
2. Civilian and commercial Applications
   - Agriculture
   - Manufacturing
   - Mining
   - Supply chain
3. Emergency response
4. Military Applications
5. Federal law enforcement

UGVs are still far from being effective, despite their numerous applications and successes. UGV reliability tests indicate that they need to be made safer and more reliable. It is essential to understand the root causes of failures to recognize areas for improvement in reliability. These four areas are critical for the safe and secure operation of UGVs. The Fishbone diagram Figure(1.4), shows the classification that we are using in the Ground Robotics Reliability Center (GRRC) to support reliable performance and operation of unmanned ground vehicles [5].

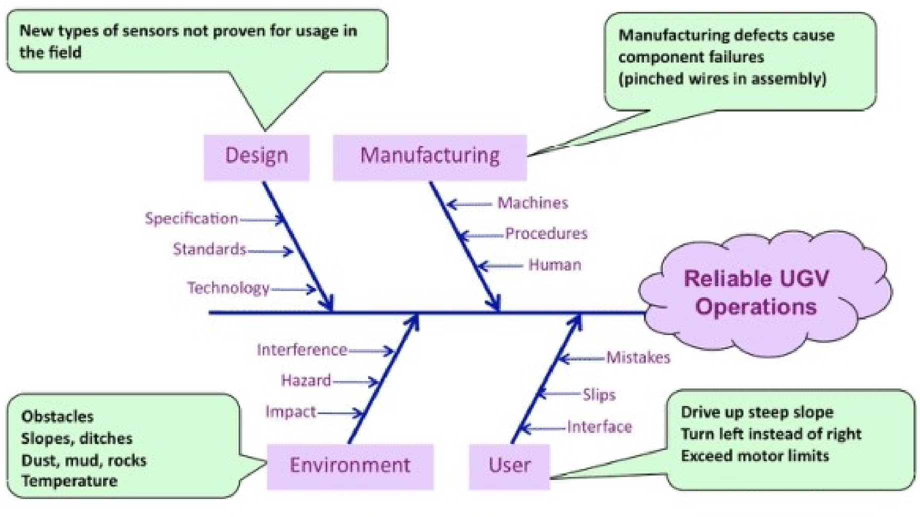


Figure 1.4: Four areas contribute to the reliable operations of UGVs

1.3 Model Predictive Control (MPC)

Model predictive control (MPC) is a sophisticated form of process control that involves meeting a collection of constraints to achieve the desired result. It is a feedback control algorithm that makes predictions about a process's outputs using a model [15]. It is appeared in the late 1970s and has steadily grown since then. The term model predictive denotes a wide variety of controls rather than a particular form of control technique. It is widely used at the current time in the process industry and applications to regulate a wide range of other procedures, from robotics to clinical anesthesia. Many other application areas such as; in the cement industry, drying towers, robot arms, chemical industry, Poly vinyl chloride (PVC) plants, steam generators, and many more [15]. MPC can control complex nonlinear systems that are constrained by nonlinear constraints. It operates by using a device model to optimize control signals over a given prediction horizon and then executing a portion of those optimized signals [16]. As a result, it appears to be an exciting and promising solution for overcoming the problems as mentioned earlier. There are various advantages of MPC which has been discussed in detailed in coming chapters.

1.4 Summary

In this chapter, brief introductory paragraphs have presented about the UGV and MPC. A brief history with various applications over the period of time has discussed. The reasons for the sudden hike in the interest of the UGV have been briefly summarized. Moreover, the reliability of autonomous vehicles has been discussed which would be a concern. The safety guaranty and efficient output lead the autonomous vehicle more reliable. An autonomous vehicle is essentially a self-driving robot that operates without human intervention and is based on artificial intelligence. The vehicle's sensor reads the surrounding, which is then used by a control algorithm to decide the next step in getting the optimum solution of the plant. Nowadays, UGV usage has been increasing gradually in intelligence, civilian, and study/research fields. Despite the rapidly increasing interest in Unmanned autonomous aircraft, watercraft, and ground vehicles, human operator interfaces remain burdensomely complex and costly to operate. In the next chapter, the vehicle's orientation, speed, and steering using the mathematical model of kinematic bicycle model have discussed.

2 Mathematical Model of Unmanned Ground Vehicle

2.1 Introduction

A unmanned ground vehicle (UGV) is a piece of mechanized equipment that travels around the ground without the intervention of a human aboard. It can be operated entirely autonomously or through remote control. Often it is also called driver-less vehicle or Autonomous vehicle [17]. It is designed and programmed in such a manner that it can fulfill all the requirements which are needed for a safe autonomous drive. It can be even more reliable than the manned vehicle because it has the capabilities to avoid obstacles, follow traffic rules, maintain a safe distance, and safe crossing at a turning point on the roads. Moreover, the reaction time of UGVs is faster than a manned vehicle and has self-optimizing capabilities according to the need. It is equipped with sensors to observe the surrounding environment and make self decisions for the next step or pass the information to a remote human operator to improve performance and reliability.

The autonomous vehicle development stages have been divided into five levels that leads to autonomy, as shown in Figure(2.1). Levels range from 0 (no assistance systems at all) to 5, which defines fully autonomous driving. However, fully autonomous driving is still far away from reality [18].

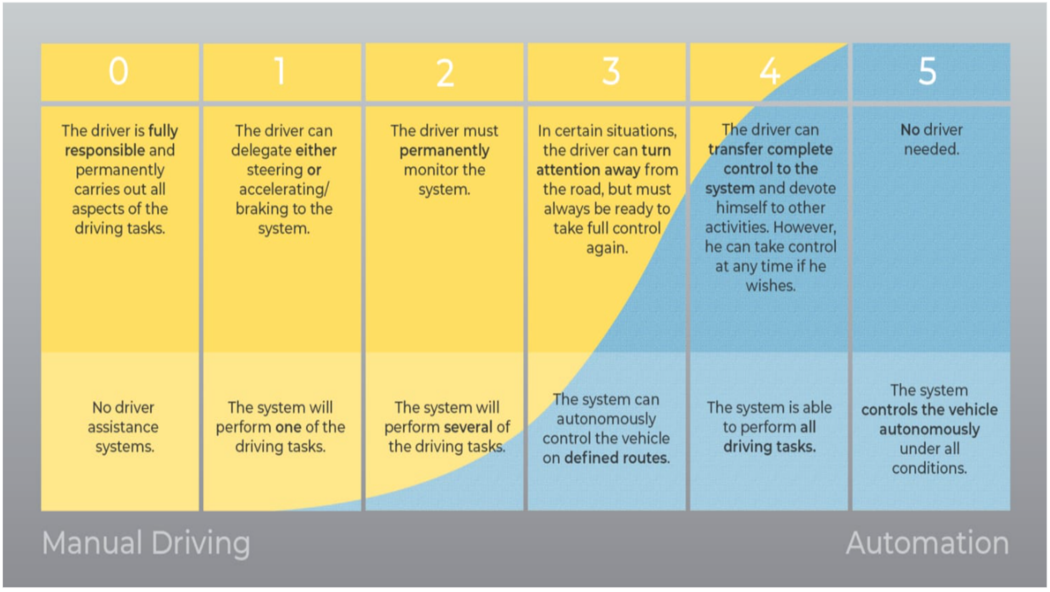


Figure 2.1: Gradual development of autonomous driving functions

Level of Autonomy: The Society of Automotive Engineers (SAE) divides the development stages up to autonomous vehicles into five levels, which define the extent to which vehicles can take over driver tasks.

Level 0 – Manual driving: In the 1990s, there was no unique assistance system like Electronic Stability Program (ESP), Parking assistance, or any other assistance system. Every function was operated entirely manually.

Level 1 – Driver assistance: In the current era, a vehicle without any assistance systems is hard to find. Today's vehicles are equipped with cruise control, preview capability, emergency brake assistance, and many more features. They support drivers to improve driving safety and comfort, but no way to fully substitute a driver has yet been discovered.

Level 2 – Partial automation: Various forms of assistance are combined so that the vehicle can execute driving maneuvers separately, such as parking or navigating stop-and-go traffic. The majority of current vehicle models are classified as level 2.

Level 3– Conditional automation: In this level, the car actually drives autonomously at certain level under certain conditions. This level of automation is only feasible in highway driving.

Level 4 – High automation: This level deals with the automation of all driving situations, where vehicle control can independently complete the journey on the highways and the city traffic.

Level 5 – Full automation: At this level, we come to actual autonomous driving, i.e., the driver becomes a complete passenger. The vehicle has no steering or pedals in this configuration; it is capable of driving entirely independently and is not designed for the passenger to interfere. Fully autonomous driving will only be permissible with secure environment exposure.

Level 3 and higher automated driving functions are only possible if vehicles can accurately sense their entire environment and extract action instructions accordingly. Sensors like ultrasonic sensors, radar, cameras, and the Light Detection and Ranging (LIDAR) sensor, which has high resolution and 3D measurement capabilities, will be used in such vehicles. UGV structure may vary from one to another, depend on the need and the applications, but generally, it consists of the following elements [14].

- **Sensors**: A robot needs to have a sensor(s) to observe its surroundings to optimize its movement. Sensors' accuracy is highly importance for UGVs.
- **Platform**: The platform gives locomotion, utility support, and power for UGVs.
- **Communication**: Communication is essential for different applications, where both accuracy and privacy of information exchange are important.
- **Control**: The level of autonomy and ability of the UGVs depends to a grand instant on its control system, which helps to control various obstacles using methods of the

control algorithm such as hierarchical learning, model predictive control,$\cdots$ etc.

- **System integration**: Within a UGV framework, the system-level architecture, configuration, sensors, and components play a significant role. A well-designed UGV would become self-sufficient, adaptable, and fault-tolerant, resulting in increased autonomy.

A Prototype of an unmanned ground vehicle is shown in Figure(2.2) [19], which is controlled autonomously or through a remote. A basic example of this principle would be a remote-controlled toy car.

Figure 2.2: A sample prototype of Unmanned ground vehicle

UGVs have many potential applications ranging from military to civilians, such as border surveillance, search and rescue missions, monitoring buildings and infrastructures, and even for home usages, such as harvesting crops and cleaning floors. Moreover, it has excellent potential for naval operation, playing a vital role in support of Marine Corps combat. Furthermore, an unmanned ground vehicle also used for industrial applications to automatically transport materials throughout a manufacturing warehouse, as shown in Figure(2.4) [20]. Since the last two decades, Autonomous Unmanned Ground Vehicle (AUGV) completely replaced human where it can be dangerous, inaccessible, or impossible for a human operator in both civilian and military applications such as critical rescue operation, fire fighting, mine detection, patrolling, as seen in Figure(2.3) [21] [22].
From the aforementioned examples and definition, we can say that the critical factor in AUGV is the onboard controller, which determines the behavior of the car based on the received sensor information. For safety reasons, it is recommended that any controller design should be tested in a simulation environment before being applied to the actual car. Then an excellent promising mathematical model that represents the car dynamics is a challenging point to build any simulation environment. So to say, the more accurate the model is, the more matched results between the simulated model and the actual model we

(a) Weaponized UGV Demonstrated During Live-Fire Exercise

(b) An UGV robot is patrolling the streets during the lockdown

Figure 2.3: UGV applications for Civilians and Military

Figure 2.4: Automated ground vehicle application in Industry/Warehouse

get. Often there is a trade-off between model complexity and computational power. In the upcoming section, we will present a mathematical model for a car, which is often used in literature.

2.1.1 Study of motion

In general, the study of the relationship between motion, forces, and the energy is defined as Mechanics [23]. Motion is the action of changing the location or position due to an acting force. The state of the body has further classified into three types.
Statics: It involves the study of bodies that are in equilibrium concerning a frame reference. The word statics may imply stationary, but it is exactly not correct. It has two aspects:

- The body can either be in rest (static equilibrium)
- Or the body can be moving with the constant velocity (Dynamic equilibrium)

Kinematics: It involves studying the motion parameters; velocity, acceleration, time, distance, displacement, etc. Without the reference to the forces that cause the motion.

Dynamics: It deals with the study of bodies that are in accelerated motion. It deals with the study of forces and the effect of force on the motion of the bodies. In addition to the study of motion, it also concerns the forces that cause the motion.

2.2 Kinematic bicycle model

Kinematics is about the geometric description of motions in space (e.g., based on different reference frames and coordinate systems). The kinematic bicycle model is the simplification of the kinematic four-wheels model, where the two rear and the two front wheels are lumped together into one rear and one front wheel, respectively, as shown below in the Figure(2.5) [24] [25].

Definition 2.1 [*Side slip angle* β] : *In-vehicle dynamics, slip angle or slide slip angle is the angle between the direction in which a wheel is pointing and the direction in which it is actually travelling (i.e., angle between the forward velocity vector and the line that connects the two-wheels).*

Definition 2.2 [*Steering angle* δ_f] : *Steering angle is the difference between the kinematic bicycle heading angle and heading angle of the front wheel.*

Definition 2.3 [*Heading angle* ψ] : *Heading angle is the orientation angle, which defines the entire orientation of the bicycle model in an X-Y plane.*

Definition 2.4 [*Yaw(rotation)* ω] : *It is defined as angular velocity, or the rate of change of the heading angle. which defines the entire orientation of the bicycle model in an X-Y plane.*

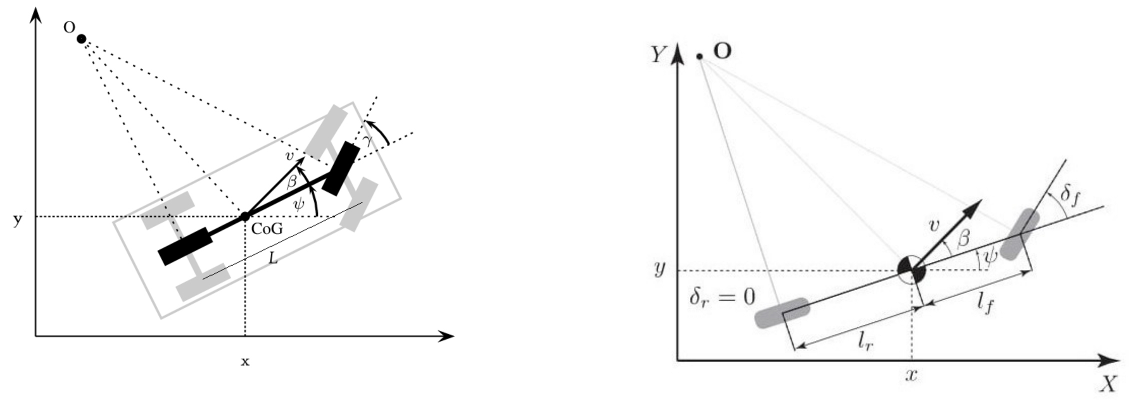
Definition 2.5 [L] : *It is the total length from rear wheel to the front wheel i.e: length of the kinematic bicycle model, i.e:* $l_f + l_r = L$.

To evaluate the motion's equation of kinematic bicycle model the following assumptions have been made [26]:

1. The slip angle of all wheels is Zero, and hence kinematic bicycle model behaves same as the four-wheel model.
2. It has the same instantaneous center of rotation O.
3. The vehicle reference point has been determined to be in the vehicle's centre, as indicated by Center of Gravity (CoG).
4. There is only one front wheel, hence only one steering angle required. Which reduces the complexity.

2.2.1 Equations of motion for Kinematic bicycle model

The following nonlinear continuous-time equations define a kinematic bicycle model: [24].



(a) Kinematic 4-wheels model lumped into 2-wheels bicycle

(b) Variables in co-ordinate system of 2- wheels lumped model model

Figure 2.5: Kinematic Bicycle Model of the vehicle

$$\dot{x} = v\cos(\psi + \beta) \tag{2.1a}$$

$$\dot{y} = v\sin(\psi + \beta) \tag{2.1b}$$

$$\dot{\psi} = \omega = \frac{v}{l_r}\sin(\beta) \tag{2.1c}$$

$$\dot{v} = a \tag{2.1d}$$

$$\beta = \tan^{-1}[\frac{l_r}{l_f + l_r}\tan(\delta_f)] \tag{2.1e}$$

Where equations (2.1a) and (2.1b) represent the vehicle linear translations in x-y plane with respect to the global frame of reference. Equations (2.1c) and (2.1e) represent the heading angle and side-slip angle, respectively. Where l_r and l_f is the distance from the center of the vehicle to the rear wheel and front wheel, respectively. The vehicle's velocity is v, and a is the acceleration of the center of mass, which applies in the same direction as the velocity. [24].

In-vehicle dynamic, the inertial frame is fixed to the earth, and we choose the coordinates X-Y-Z as the coordinates for the inertial frame. Compared to higher fidelity vehicle models or dynamic bicycle models, there are only two unknown parameters l_r and l_f to be determined. Hence, the system identification using a kinematic bicycle model is more superficial. The kinematic bicycle is a classic model that captures vehicle motion in normal driving conditions remarkably well. Due to its simplicity and compliance to the non-holonomic constraints of a car, the well-known kinematic bicycle model has long been used as a suitable control-oriented model for representing car [6]. Additionally, the proposed approach with the kinematic bicycle model is less computationally expensive than vehicle tire models. Moreover, at low vehicle speeds, where tire models become singular, it can be implemented. Furthermore, it improves the maneuverability and accuracy of path tracking [24] [26].

2.3 Hamster

Hamster is a smart Robot Operating System (ROS) based autonomous ground vehicle for Industry and Academic R&D. It is developed by Congniteam company. It is capable of running Simultaneous Localization and Mapping (SLAM), Path planning, Exploration, waypoint driving, and obstacle avoidance [27]. Hamsters are useful in civilian and military operations to perform various assignments such as; Fire fighting, Security and Surveillance, Landmine detection, Warehouse operation,$\cdots$, etc. It has also been broadly used for study and research purposes.
For a Hamster to perform fully autonomously, it must reliably detect its entire environment and derive control actions accordingly. In practical application, the most common emergency situation is obstacle avoidance. In this case, the Hamster can use a variety of sensors, including ultrasonic sensors, radar, and cameras, in addition to the LIDAR sensor. Figure(2.6) shows the main components of hamster while Table [2.1] gives the hardware specifications [27].
The Hamster model can be represented by the kinematic four-wheels model, where the two rear and the two front wheels are lumped together into one rear and one front wheel, respectively. It also follows the mathematical model of kinematic bicycle model represented by the set of equations (2.1)

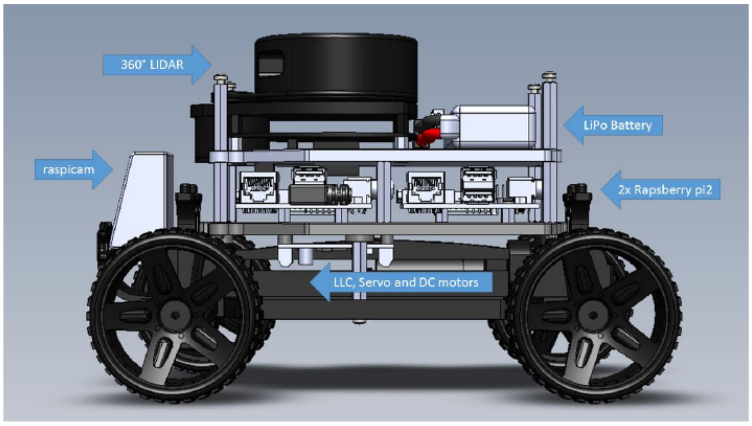


Figure 2.6: Hamster

2.3.1 Mathematical representation of Hamster

Since the Hamster mathematical model can be derived based on the kinematic bicycle model, the definitions, assumptions, and working principle are based on the discussion in section (2.2). So, the Hamster does satisfy the equation of the motion (2.1) of the kinematic bicycle model. It's one of the most basic models, and it's been used for motion planning for over a decade, with the hope of capturing enough of the nonholonomic constraints of real vehicle dynamics [28].
Our system model is given by the nonlinear continuous time equations from (2.1)

Table 2.1: Hamsters data

Hamster Parameters	
Parameters	Value
Maximum speed (v_{max})	1.2 m/s
Minimum speed (v_{min})	0.1 m/s
Steering angle(δ_{fmin})	-0.245rad (-14°)
Steering angle(δ_{fmax})	0.245rad (14°)
Center to rear wheel distance (l_r)	0.15m
Center to front wheel distance (l_f)	0.15m
Turning degree sensitivity	1°

2.4 Summary

This chapter discussed the brief Introduction of unmanned ground vehicle and their applications in various fields. The various autonomy level of UGV has been discussed. Also, the mathematical model of UGV has been presented. The Kinematic bicycle model is less computationally expensive than existing approaches that use vehicle tire models, so the kinematic bicycle model was chosen over the dynamic bicycle model. It also improves the maneuverability and accuracy of path tracking.

For describing slow-moving vehicles, a kinematic model is a suitable alternative. The kinematic model will lose fidelity as the speed increases, such as in a passenger car or a race car, due to the nonlinear characteristics of tire forces. Vehicle motion control and route planning applications often use kinematic bicycle models [29].

A brief introduction of the Hamster has been discussed, which works on the principle of the kinematic bicycle model. In the next chapter, we will be discussing the Model Predictive Control in detail.

3 Model Predictive Control

3.1 Introduction to Model Predictive Control

Model predictive control (MPC) is an advanced version of process control that involves meeting a set of constraints in order to achieve the desired result. It is optimization-based control, defined as a repetitive decision-making process as shown in Figure(3.2) [30].In other words, it is nothing but the repeated solution of an optimal control over finite horizon. The Prediction Horizon N_p (determines how far into the future the model behavior can be predicted.) is also known as Receding Horizon Predictive Control (RHPC) because of its forward-moving nature [31].
The MPC's main benefit is that it supports optimizing the current time slot while keeping upcoming time slots in consideration. It makes predictions about the plant's behavior using a model of the plant. It also employs an optimizer to ensure that the expected future plant performance is consistent with the desired reference. In the field of AUGV control, the MPC algorithm, which defines a multi-variable cost function under future reference state conditions and minimizes the function under multiple constraints, has sparked much interest. MPC models are intended to describe the behavior of complex dynamical systems in general. Extensive time delays and high-order dynamics, as these are difficult for Proportional Integral Derivative (PID) controllers to monitor, can be handled by MPC [31].
In the decade 1970, MPC gained much interest in the process industry with some contributions like; dynamic matrix control, model predictive based heuristic control, generalized predictive control [30]. Due to the simplicity of the algorithm and dynamic models, MPC strategies have been quickly adopted recently. In the last 2 decades it has diversified into wider range of applications e.g. Robotics,UGVs, Aerospace.
The layout of MPC is shown in the Figure(3.1) [15]. where the main components in MPC used:

1. A system model
2. An objective function
3. Optimizer
4. Algorithm for solving the problem of constrained optimization

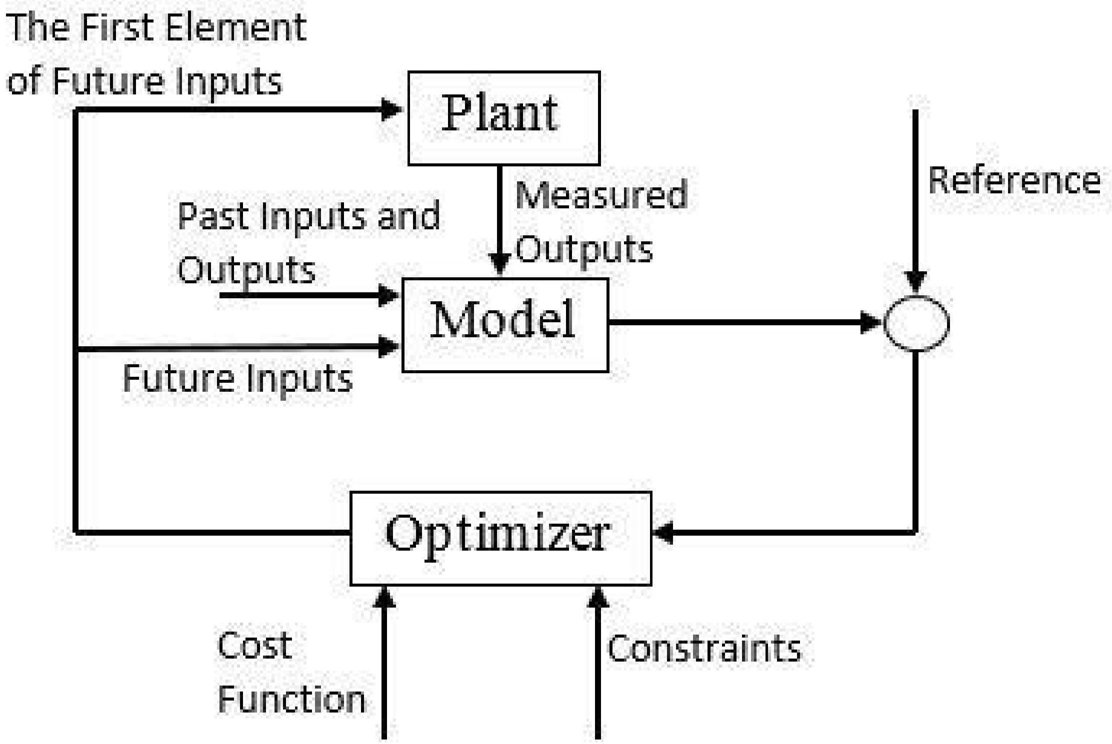


Figure 3.1: Block structure of MPC

3.1.1 Advantages of MPC over other methods

Among other optimal control techniques, MPC is one of the most popular controller, which computes the solution of an optimal control problem in a receding horizon fashion while taking multiple constraints in to account [15]. Due to its inability to deal with multiple obstacle control targets, the PID controller is not an ideal control approach for autonomous car route tracking. In the monitoring phase, the MPC ensures tracking accuracy and considers the vehicle's dynamic stability, i.e., the vehicle dynamics model is used as the controller model. Moreover, MPC resolves the issues of driving comfort induced by other controllers when the vehicle deviates from the target path. [32]
Since the 1980s, Model Predictive Controllers are being used in the process industry. [33]. With the increasing processing ability of microprocessors, their use has expanded to other fields as well. The research papers below give the detailed explanation of the applications of MPC in various field. Automotive [34] [35], Aerospace [30], Energy [36], Food Processing [15], Industrial Manufacturing [33] [15], Robotics [37] [15].
One of the most significant features is that it has the Preview capability, which helps to detect the disturbance before it affects the process output. It is similar to Feed-forward control(takes an appropriate control action in advance to eliminate disturbance effect on output). It has all these benefits, but at the same time, it necessitates the use of a powerful, fast processor with plenty of memory because it has to act in a less span of time

to protect the system efficiently. The table [3.1] below gives the Various advantages and disadvantages of MPC [31].

Table 3.1: Advantages and disadvantages of MPC

Advantages	**Disadvantages**
-Handling of Multi-input Multi-output (MIMO) system.	-Appropriate system model required.
-To compensate for the measurable disturbances, It introduces feed-forward control.	-It necessarily involves the use of a powerful, fast processor with huge memory storage capacity.
- Handling of Nonlinear systems.	-Real-time solution of the optimal control problem.
-Allows direct consideration of constraints and different type of cost functions .	-It is computationally expensive.
-Consideration of different types of cost functions.	
- Constraints satisfaction and relatively easy tuning.	

3.2 Principles of Model predictive control (MPC)

The Model predictive control scheme is an optimization-based control, which follows the repetitive decision-making process, as shown in Figure(3.2) [30]. It solves iteratively a finite horizon optimization problem by taking the system dynamics and various constraints into account. A common application of MPC is to minimize the distance of some states of the system with respect to a given reference.

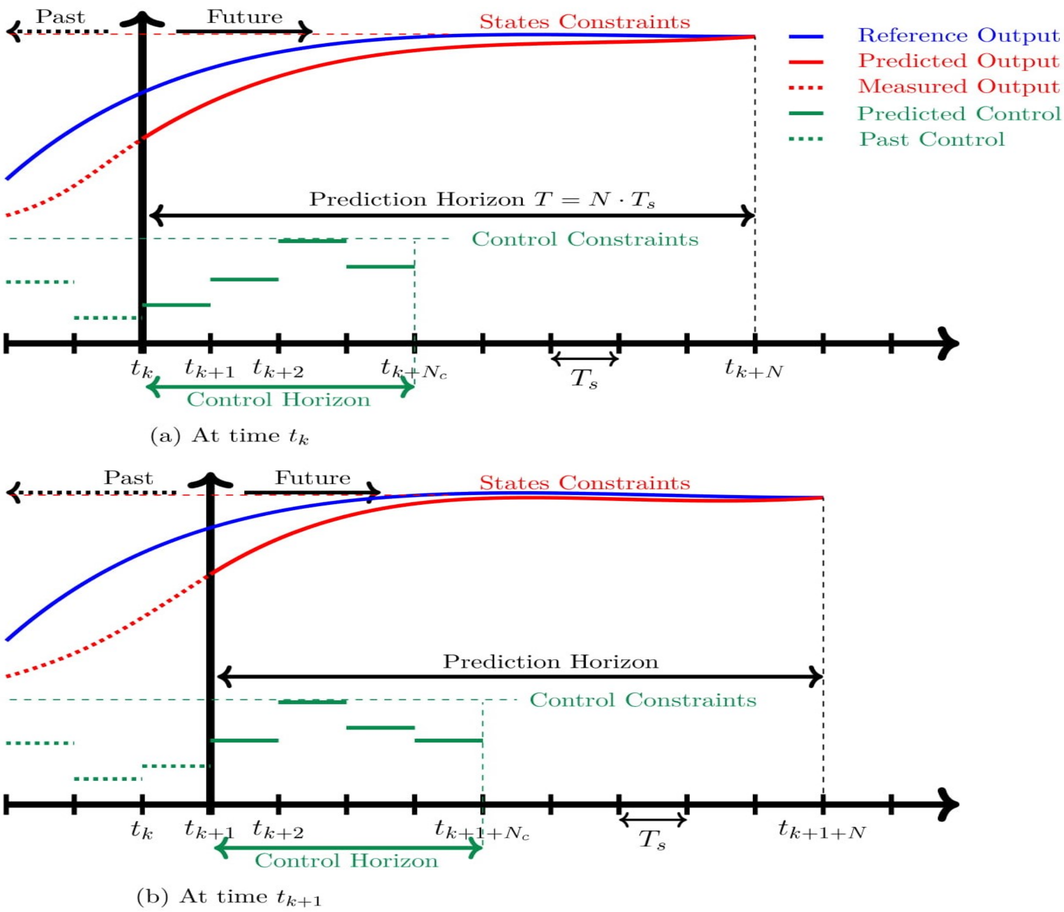


Figure 3.2: Principles of Model Predictive Control strategy

Principle of MPC strategy elaborates the reference and the predicted output based on the measured output, and the present and future control inputs. The future sequence of the system behavior is predicted over the horizon T = N · Ts, The optimal control sequence $\mathbf{u*} = [u(t_k), u(t_{k+1}), ..., u(t_{k+N})]$, for $t_k = 0, 1, 2, \ . \ . \ .$ is calculated to optimize a cost functional, which represents the performance index involving objectives on both states and inputs.

The first control signal candidate is plugged into the system, and the system states are measured/predicted in the next simulation step, this process is repeated until certain user defined criterion is achieved. Algorithm 1 explains how MPC works.

Table 3.2: MPC Algorithm

Algorithm 1: MPC Algorithm [30]

Result: Optimal Control Sequence $\mathbf{u^*}$ = [u (t_k), $u(t_{k+1})$, ..., $u(t_{k+N})$]
for t_k = 0, 1, 2, . . . do ;

1. Measure/estimate the system state $x(t_k) = x_0$ at the current time t_k;
2. Predict the future behavior using system model $f(x(\cdot); u(\cdot))$ based on the current state $x(t_k)$ and the calculated inputs;
3. Solve a finite horizon Optimal Control Problem (OCP) by minimize/maximize a cost function respect to the system constraints;
4. Calculate a sequence of optimal control inputs $\mathbf{u^*}$, then apply only $u(t_k)$;
5. Use the receding horizon strategy, whole procedure is repeated, go to step 1.

All the processes in the algorithm are repeated at each time step, which makes MPC an optimal solution for the many control problems for the UGV and aerospace applications. The principle behind the Success of Model Predictive Control is defined through the basic structure of MPC. The Block structure has been shown in Figure(3.1) [15], a broadly used model to predict future plant outputs based on past and current values and the proposed optimal future control actions. The optimizer calculates these actions by taking into account the various basic parameters; prediction horizon, control horizon, sample time, constraints, reference trajectory, and cost functions [15].
Definition 3.1 [*Prediction Horizon N_p*]: *Prediction horizon (N_p) dictates how far into the future the model will forecast.*
The user learns how to control the system more quickly and achieves improved performance when the prediction horizon is well suited to the lag between input and output. The value of the prediction horizon can be chosen manually depending upon the model. A more excellent value of the prediction horizon leads to more stability to the plant. At the same time, an increment in the value of the predictive horizon will also increase the process of calculation [31]. According to the principle, MPC system is based on discrete-time control methodology, and hence it only produces control action for discrete-time intervals. $[t_k, t_{k+1}, t_{k+2}..., t_{k+N}]$
Definition 3.2 [*Control Horizon N_c*]: *The control horizon(N_c) is the number of manipulated variables move to be optimized at the control interval k.*
Its value can be chosen between 1 and Prediction horizon N_p. However, practically it is chosen 10 to 20 percent of Prediction horizon N_p [31]. In most cases small (N_c) promotes an internally stable model.
Definition 3.3 [*Sample Time T_s*]: *Sample time is the set of time intervals that helps the*

prediction horizon to predict the future value.
The recommendation for choosing a prediction horizon is 20-30 samples. When a disruption occurs, if it is too big, the controller will not respond quickly enough. On the other hand, if the sample time is too short, the controller will respond to disruptions and setpoint changes much faster, but this places an unnecessary computation complexity on the controller. The time interval between the two-time sample is: $T_s = t_{k+1} - t_k$

Definition 3.4 [*Constraints*]: *A set of data which MPC should satisfy strictly depends on the requirement. It is the most important objective which MPC should perform successfully to get the optimal output.*
MPC can include constraints on the inputs, the rate of change of inputs, and the outputs. It is recommended that outputs be considered soft constraints and that hard constraints on inputs and the rate of change of inputs be avoided.

- Hard constraints: are the set of conditions for the variables that must be satisfied for the optimal solution. They cannot be violated.
- Soft constraints: are the set of conditions for the variables that can or cannot satisfy sometimes. It can be violated sometime to get better result.

Definition 3.6 [*Cost Functions J*]: *It is a mathematical equation that finds the error gap between the predicted outputs and the reference. It is a function to be minimized to get an efficient system output.*
The cost function consists of the multiplying of the difference between the output signal and reference signal. Smallest J will be the optimal solution.

3.2.1 System model and test setup

A hamster model has been taken as a sample model to present optimal result of an UGV using MPC control design technique. Various global test parameter has been considered. General nonlinear system dynamic representation in state space form:

$$\dot{x} = f(x(t), u(t)) \tag{3.1a}$$

$$y = h(x(t), u(t)) \tag{3.1b}$$

Where $\dot{x}$ is the state and y is the output equation. The system model is given by the nonlinear continuous time equation in the equation (2.1)
Cost function(J) which is consider as soft constraints is given by the equation (3.2),

$$J = \sum_{i=1}^{N_p} \|X_i - X_{ref}\|_Q^2 + \|U - U_{ref}\|_R^2 \tag{3.2}$$

Where Q & R are the defined as Weighting matrices, which penalize the cost function to get the optimal solution.

Such that,

$$\dot{x} = f(x(t), u(t)) \tag{3.3a}$$

$$v_{min} \leq v \leq v_{max} \tag{3.3b}$$

$$\delta_{fmin} \leq \delta_f \leq \delta_{fmax} \tag{3.3c}$$

$$x(t) \in X, u(t) \in U \tag{3.3d}$$

Where equation (3.3d) defines the range of the feasible state space set and control input set respectively. Equation (3.3b) and (3.3c) indicate the range of the velocity and the steering angle respectively.

3.3 MPC with ACADO toolkit

In recent years, the number of Linear and Nonlinear MPC applications for fast-dynamic systems has increased significantly as computing power has increased. There are many tools and software packages available to solve linear and Nonlinear Model Predictive Control (NMPC) like ACADO, CasADi [30]. ACADO is an open-source platform for dynamic optimization and automatic control. It provides a general framework for implementing a wide range of direct optimal control algorithms, such as model predictive control, state and parameter estimation, and robust optimization. To solve double integrator model of UGV, we proposed to use ACADO Toolkit, which provides a MATLAB interface to generate C++ code for NMPC [38]. It implements the real-time iteration scheme with a different algorithm to solve the OCP and incorporates it with the efficient numerical integrator in auto-generated C-code. The various process to interface the ACADO with MATLAB is given by ACADO Toolkit, GitHub [39].

3.4 Summary

This chapter discussed a brief introduction to MPC. A series of advantages of MPC have been discovered over other methods. Moreover, the algorithm of the MPC has been discussed and shown why it has been broadly used in various applications in the last few decades. MPC controller makes forecasts about future plant output. It also uses the optimizer, which finds an optimal sequence of control inputs that pushes the projected plant output as close to the setpoint as possible while minimizing the cost function. It also ensures that the steering wheel angle, velocity, and the car's position stay within the defined work space. These are assigned as constraints. Furthermore, Nonlinear systems that are not easy to be linearized could be an option for this controller [40]. A test setup for double integrator UGV model has been explained and executed in the next chapter.

4 Simulation Results

4.1 Introduction

The mathematical model of MPC and various physical constraints on Hamster/UGV have been taken to simulate our model. We approximate the Hamster model with a double integrator model subjected to the Hamster velocity constraints for simplicity. The aim of the Hamster is to reach the target position under the specified position and velocity constraints. The UGV model has been tested for two different scenarios to cross-verify the effectiveness. The system dynamics of the double integrator UGV model have been given below.

$$\dot{x} = f(x(t), u(t)) \tag{4.1a}$$

$$\dot{x_1} = x_2 \tag{4.1b}$$

$$\dot{x_2} = u \tag{4.1c}$$

Equation (4.1a) shows the state of linear dynamic system. The states of the UGV model is given by the equation (4.1b) and (4.1c), which is linearly dependent. Where x_1 and x_2 are the constraints on the position and velocity, respectively. u is the acceleration. The various other constraints on the UGV has been stated in the Table (4.1), where the initial and final car velocity is same for both the scenario, given by x_{2int} and x_{2ref}, respectively. Whereas, the initial and final car positions for the first scenario are stated by x_{1int}, x_{1ref}. Also, the initial and final car position for the second scenario is given by x_{1int}, x_{22ref}.

Table 4.1: Double integrator UGV model data

Simulation Parameters	
Parameters	value
Initial car position [x_{1int}]	0
Initial car velocity [m/s] [x_{2int}]	0
Final car position [x_{1ref}]	20
Final car position [x_{22ref}]	10
Final car velocity [m/s] [x_{2ref}]	0
Prediction horizon N_p	50
Sample time [sec](T_s)	0.4
Simulation time [sec](T_f)	12

4.2 Simulation of Hamster with Model Predictive Based Controller usign ACADO toolkit

The system dynamics under various constraints have been simulated using MATLAB -ACADO software packages. Also, the MATLAB code has been explained briefly.

MATLAB Coding

```

% This code is written as a part of NTP
%=========Year :April 2021===========%
clc;
clear all;
close all;

% Condition for compiling
EXPORT = 1;

DifferentialState x1;                    % Position
DifferentialState x2;                    % Speed  (dot(y)=dfx2))
Control u;                               % Control input

%% Diferential Equation

f = dot([x1; x2]) == [x2; u];

%define the step length
simTs = 0.4;                    % time interval
Num = 50;
numSteps = 2;
%define the objective function ( stage and end costs)
h = [diffStates; controls];
hN = [diffStates];
% Define number of sates and control signal
n_XD = length(diffStates);
n_U = length(controls);

%% SIMexport (Integrator Setting)
acadoSet('problemname', 'sim');

```

```
sim = acado.SIMexport( simTs );
sim.setModel(f);
sim.set( 'INTEGRATOR_TYPE',                'INT_IRK_RIIA5' );
sim.set( 'NUM_INTEGRATOR_STEPS',           numSteps        );

if EXPORT
    sim.exportCode( 'export_SIM' );

    cd export_SIM
    make_acado_integrator('./../integrate_DI')
    cd ..
end

%% MPCexport (optimal control setting)
acadoSet('problemname', 'NMPC');

ocp = acado.OCP( 0.0, simTs*Num, Num );
W_mat = eye(n_XD+n_U,n_XD+n_U);
WN_mat = eye(n_XD,n_XD);
W = acado.BMatrix(W_mat);
WN = acado.BMatrix(WN_mat);
ocp.minimizeLSQ( W, h);
ocp.minimizeLSQEndTerm( WN, hN );

ocp.subjectTo( 0 <= x1 <= 20 );     % Constraints on position
ocp.subjectTo( 0 <= x2   <= 1.2 ); % Constraints on speed

ocp.setModel(f);
mpc = acado.OCPexport( ocp );
mpc.set( 'HESSIAN_APPROXIMATION',       'GAUSS_NEWTON'       );
mpc.set( 'DISCRETIZATION_TYPE',         'MULTIPLE_SHOOTING'  );
mpc.set( 'PRINTLEVEL',                  'NONE'               );
mpc.set( 'INTEGRATOR_TYPE',             'INT_EX_EULER'       );
mpc.set( 'CG_USE_OPENMP ',              'YES');
mpc.set( 'NUM_INTEGRATOR_STEPS',         2*Num
    );
mpc.set( 'LEVENBERG_MARQUARDT',                   1e-4
                        );

```

```
if EXPORT
    mpc.exportCode( 'export_MPC' );
     copyfile('../../../../../../acado-master/external_packages/
        qpoases', 'export_MPC/qpoases', 'f')
%  copyfile('C:/Users/leeck/Desktop/ACADO/external_packages/
    qpoases','export_MPC/qpoases', 'f')
     cd export_MPC
     make_acado_solver('../acado_DI')
     cd ..
end
%==================================================%
% Simulation Parameters
N=50; % Tp = N*Ts (Ts = 0.4s)
dt =0.2;
input.N =50;            % predicition Horizon
X0 = [0  0 ]  ;
Xref = [10  0];
input.x = repmat(Xref,N+1,1);
Xref = repmat(Xref,N,1);

Uref = zeros(N,n_U);
input.u = Uref;

input.y = [Xref(1:N,:)  Uref];
input.yN = Xref(N,:);
% Weightening Matrices
input.W  = diag([0.02,    0.02,   0.02]);
input.WN  = diag([3,  3]);

 %============================================================%
disp('------------------------------HAMSTER ------------------------------')
disp('                 Simulation Loop'                 )

iter = 0; time = 0;
Tf = 12;
KKT_MPC = []; INFO_MPC = [];
controls_MPC = [];
state_sim = X0;
while time(end) < Tf
```

```
    input.x0 = state_sim(end,:);
    output = acado_DI(input);      % Solve our OCP

    input.x = output.x;
    input.u = output.u;
    input_sim.x = state_sim(end,:)';
    input_sim.u = output.u(1,:)';              % MPC control
    states = integrate_DI(input_sim);       % Simulate our system
    state_sim = [state_sim; states.value'];

    % Save the MPC step
    INFO_MPC = [INFO_MPC; output.info];
    KKT_MPC = [KKT_MPC; output.info.kktValue];
    controls_MPC = [controls_MPC; output.u(1,:)];
    controls = [controls; output.u(1,:)];

    iter = iter+1;
    nextTime = iter*dt;
    time = [time nextTime];
    result.CalTime = output.info.cpuTime;
end

 %==========Plot the results============%
 figure(1) % position graph
 plot(time',state_sim(:,1),'k','LineWidth',2)
 grid on
 xlabel('Time [sec]')
 ylabel('Position [m]')

 constV(1:1:length(time))=1.2;
 figure(2) %Velocity graph
 plot(time',state_sim(:,2),'k','LineWidth',2)
 hold on
 plot(time,constV,'r','LineWidth',1)
 xlabel('Time [sec]')
 ylabel('Velocity [m/sec]')
 legend('Velocity','Constraints')
 grid on
end
```

4.2.1 MATLAB code explanation

The above MATLAB code for the simulation of the double integrator model is executed, and the plots have been generated. The various syntaxes are explained briefly. In the code, line-17 shows the dynamic of the linear system. At the same time, Line-20,21 explains the sampling time and prediction horizon, respectively. The various constraints on the position and on the velocity have taken which has been given in line-57,58. It leads our system to strictly stay in the given velocity range and goes to zero when the target is reached. The weighting matrix for the cost function of the MPC has given in line-95,96. It is also known as tuning parameters that penalize the cost function for getting the optimal solution. Moreover, Line-84,85 explains the initial and final state and corresponding velocity of the UGV, as given in the table [4.1]. The code to plot the velocity and position graphs have stated in lines 129-145. Matlab.tikz package has been used to import the MATLAB graphs to LaTeX.

4.2.2 Simulation Results

The model predictive-based controller for the double integrator UGV model has been simulated using MATLAB-ACADO solver packages, and two different scenarios have been tested for the UGV. Figure(4.1) and (4.2) show the velocity and position of the UGV for the first scenario. Where x_{1ref} is the final state for the first scenario. The initial and final conditions and the different scenarios used to test our MPC controller are illustrate in table[4.1].

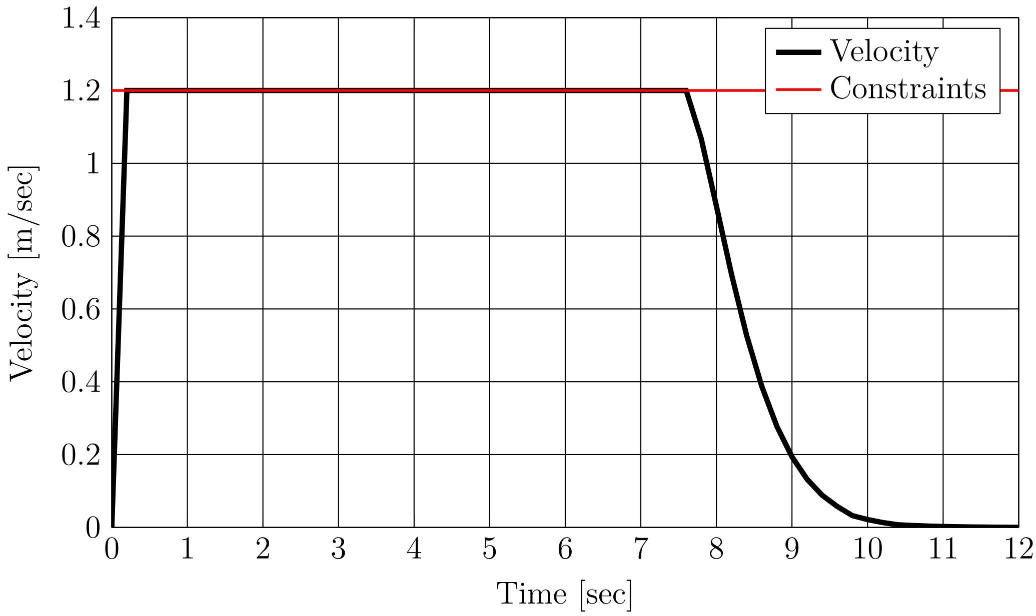


Figure 4.1: Velocity of the UGV

The above plots depict the evolution of the velocity[m/s] and position[m]. The velocity of the UGV is drastically changing as time passes. The MPC controller aware of the

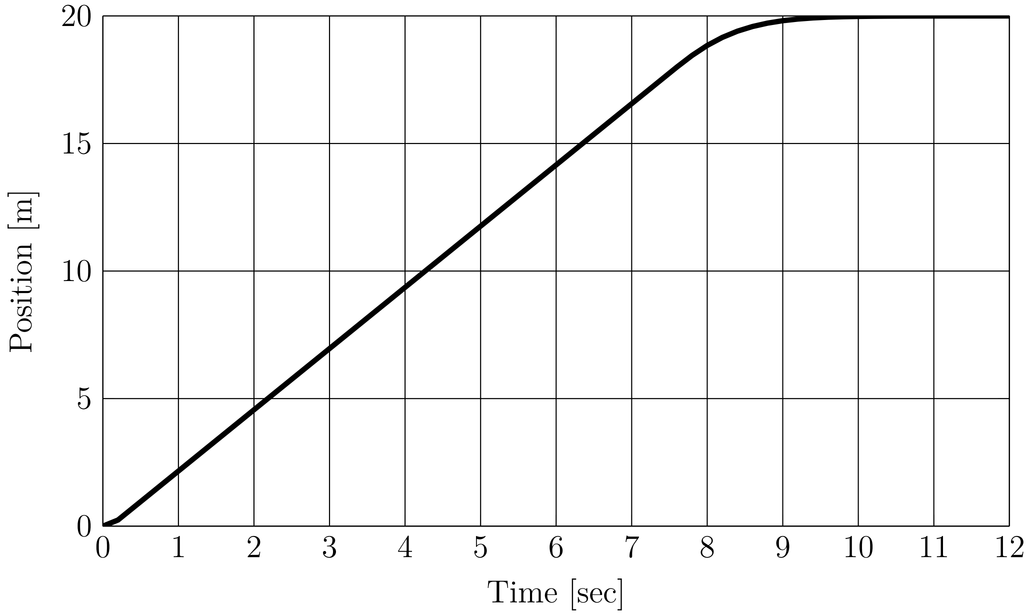


Figure 4.2: Position of the UGV

constraints, and it starts accelerating the UGV to the maximum allowable velocity of 1.2 [m/s], and once the car approaches the target, the velocity reduces and goes to zero when the car reached the target. This can be shown in the Figure (4.1) and (4.2). The target position is reached at 11sec, and hence the velocity of the UGV is strictly reduced to zero. For the second scenario as well, the model has been tested and implemented the code to check the effectiveness of the controller. Where x_{22ref} is the final state for the second scenario. The velocity and position of the UGV for the second scenario has been plotted in Figure(4.3) and (4.4).

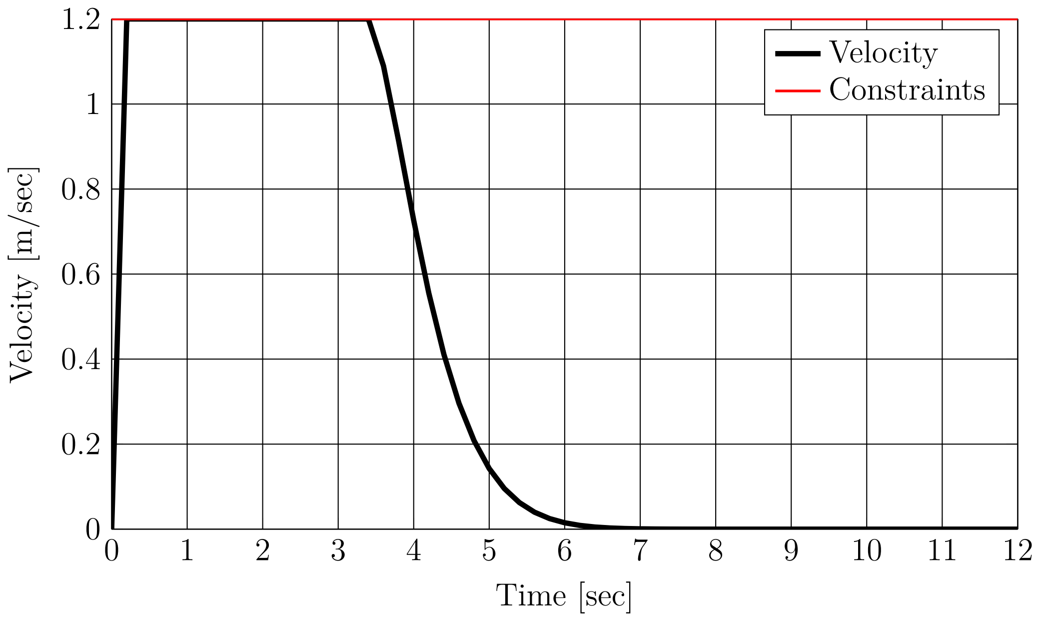


Figure 4.3: Velocity of the UGV

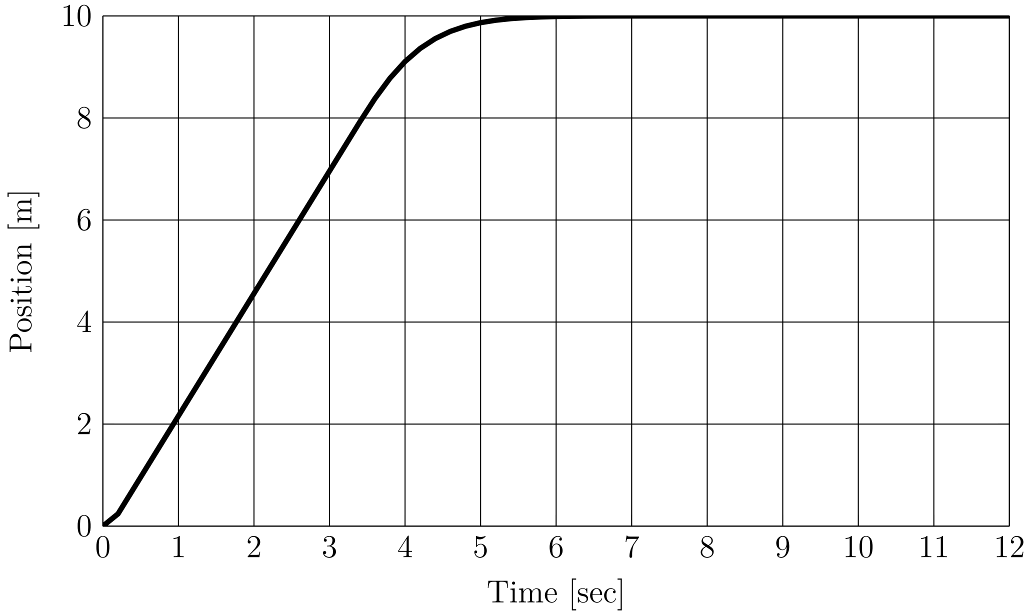


Figure 4.4: Position of the UGV

The plots above show how the velocity[m/s] and position[m] have changed over time. The MPC controller recognizes the constraints and accelerates the UGV to the maximum permissible velocity of 1.2 [m/s]. As the car approaches the target, the velocity decreases until it reaches zero at the target point. The target is completely reached at time 7sec, and thus the UGV's velocity is strictly zero.

4.3 Summary

The UGV's double integrator model was addressed in this chapter. Various constraints on the UGV have considered and implemented using MATLAB-ACADO software packages. In addition, the MATLAB code has been briefly explained. A Model Predictive based controller for the double integrator UGV model has been simulated, and the different outputs graphs have been plotted. Finally, the UGV reached the target point while strictly adhering to the velocity limit. It is re-verified by testing the second scenario by adjusting the state of the system. The conclusion and the future scope will be explored in the next chapter.

5 Conclusion and Future Outlook

This report studies both theoretical and practical aspects of UGV and MPC. In this work, MPC plays a significant role in solving different problems of UGV. Additionally, the MATLAB-ACADO software package helped to simulate the code and plot the output graphs.

First, we presented the general overview of Unmanned Ground Vehicle with a brief history. A short introduction has been given about MPC and the importance of the autonomous vehicle. The second part considered the detailed explanation of UGV with the mathematical model. The various autonomy level of UGV was discussed. It shows that how far away we are still from the actual autonomous vehicle. Moreover, the Model predictive controller has been discussed briefly with their algorithm. It is clear that MPC usage over other control methods is being increased over the period of time. The reasons for that have discussed in the chapter-3. The Hamster model with the cost function and various constraints has been explained in detail, which will continue in future project work. Furthermore, the double integrator model of UGV has been discussed and simulated under various constraints. The output graphs show that the UGV stays in the assigned position and velocity range. It goes to rest once the target is reached.

As the world moves ultimately towards autonomous vehicles, obstacle avoidance and safety challenges are the primary concern. So to say, our future work will give stress to using a more promising mathematical model for Hamster represented in chapter-2&3 for designing a nonlinear MPC controller. Moreover, the obstacle will be considered in future work, and the controller will ensure car safety by avoiding such obstacles. Mathematically, equation(5.1) will be added in the optimization problem.To this end, the problem becomes challenging as it becomes a non-convex optimization problem. Furthermore, the planned path is required to reduce the processing time, communication delay, and energy consumption as well [41].

$$g(x(t), u(t)) < 0 \tag{5.1}$$

Definition 5.1 [*Convex set*]: *A set $D \subseteq R^n$ is convex if, $\forall$ x,y $\in D$ there is a line segment connecting x and y and lies completely in D .*

If any subset of region R fails to make the line segment between two points then it is non-convex or concave set, as shown in Figure(5.1) [42]. In such case, due to the obstacle existence in the working region the problem becomes non convex. Generally, solving non-convex problem is challenging and will be the scope of our future work.

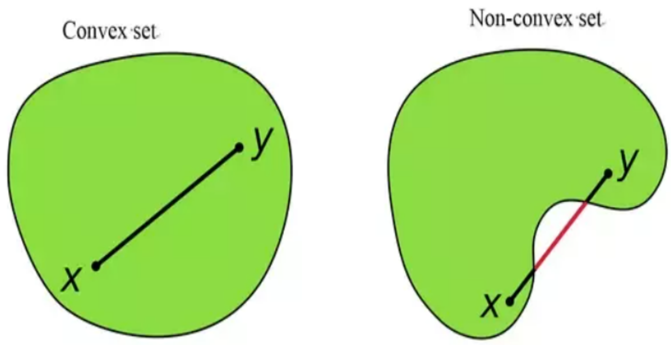


Figure 5.1: Convex and Non-convex set

In the end, looking at the versatile application and increasing demand, it is concluded that UGV is the future. Hence, making it more reliable would be an outstanding achievement, which gives a comfortable life to the upcoming generation.

Bibliography

[1] John Ebken, Mike Bruch, and Jason Lum. Applying unmanned ground vehicle technologies to unmanned surface vehicles. In *Unmanned Ground Vehicle Technology VII*, volume 5804, pages 585–596. International Society for Optics and Photonics, 2005.

[2] Philip Polack, Florent Altché, Brigitte Novel, and Arnaud de La Fortelle. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *kinematic bicycle model*, pages 812–818, 06 2017.

[3] Giuseppe Quaglia, Luca G Butera, Emanuele Chiapello, and Luca Bruzzone. Ugv epi. q-mod. In *Advances on Theory and Practice of Robots and Manipulators*, pages 331–339. Springer, 2014.

[4] Douglas W Gage. Ugv history 101: A brief history of unmanned ground vehicle (ugv) development efforts. Technical report, Naval command control and ocean surveillance center, 1995.

[5] D Tilbury and AG Ulsoy. Reliable operations of unmanned ground vehicles: Research at the ground robotics reliability center. In *Proceedings of IARP workshop on technical challenges for dependable robots in human environments*, pages 27–32, 2010.

[6] Hani Michael Sallum. *Development and implementation of a high-level command system and compact user interface for non-holonomic robots*. PhD thesis, Boston University, 2005.

[7] World wide wireless book. `https://archive.org/stream/WorldWideWirelessV2#page/n343/mode/2up`. Accessed: 2010-09-30.

[8] Rheinmetall AG. Rheinmetall Mission Master. `https://www.rheinmetall-defence.com/en/rheinmetall_defence/public_relations/themen_im_fokus/ugv/index.php`, 2016.

[9] Ioannis Mademlis, Vasileios Mygdalis, Nikos Nikolaidis, Maurizio Montagnuolo, Fulvio Negro, Alberto Messina, and Ioannis Pitas. High-level multiple-uav cinematography tools for covering outdoor events. *IEEE Transactions on Broadcasting*, 65(3):627–635, 2019.

[10] Teodor Tomic, Korbinian Schmid, Philipp Lutz, Andreas Domel, Michael Kassecker, Elmar Mair, Iris Lynne Grixa, Felix Ruess, Michael Suppa, and Darius Burschka. Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. *IEEE robotics & automation magazine*, 19(3):46–56, 2012.

[11] Chi Yuan, Youmin Zhang, and Zhixiang Liu. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Canadian journal of forest research*, 45(7):783–792, 2015.

[12] Mari Sakiyama, Terance D Miethe, Joel D Lieberman, Miliaikeala SJ Heen, and Olivia Tuttle. Big hover or big brother? public attitudes about drone usage in domestic policing activities. *Security Journal*, 30(4):1027–1044, 2017.

[13] Daeil Jo and Yongjin Kwon. Development of rescue material transport uav (unmanned aerial vehicle). *World Journal of Engineering and Technology*, 5(4):720–729, 2017.

[14] Phuoc-Nguyen Nguyen-Huu, J. Titus, Tilbury Dawn, Galip Prof, and Ulsoy. - 01 reliability and failure in unmanned ground vehicle (ugv). In *UGV*, 2009.

[15] E. F. Camacho and C. Bordons. *Model Predictive control*. Springer London, 2007.

[16] Huckleberry Febbo, Jiechao Liu, Paramsothy Jayakumar, Jeffrey L Stein, and Tulga Ersal. Moving obstacle avoidance for large, high-speed autonomous ground vehicles. In *2017 American Control Conference (ACC)*, pages 5568–5573. IEEE, 2017.

[17] H. Moon, J. Lee, J. Kim, and D. Lee. Development of unmanned ground vehicles available of urban drive. In *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 786–790, 2009.

[18] Autonomy level FLORIAN PETIT. https://www.blickfeld.com/blog/5-automation-levels/, five levels to autonomy. *blickfeld blog*, 2020.

[19] Cagatay olgar. grabcadcommunity library. `https://grabcad.com/library/robotic-tank-unmanned-ground-vehicle-ugv-1`, 2017.

[20] Themanufacturer Jonny Williamson. How is slam software powering the next-generation of autonomous industrial robots ?, 2013 https://www.themanufacturer.com/articles/slam-software-powering-next-generation-autonomous-industrial-robots/.

[21] Weaponized UGV Demonstrated During Live-Fire Exercise WEBSITE mike ball. `https://www.unmannedsystemstechnology.com/2019/06/weaponized-ugv-demonstrated-during-live-fire-exercise/`. Accessed: 2019-10-19.

[22] A UGV robot is patrolling the streets during the lockdown website. `https://www.inceptivemind.com/tunis-ugv-robot-p-guard-patrolling-streets-lockdown/12805/`. Accessed: 2020-04-20.

[23] The physics hypertextbook website. `https://physics.info/motion/`. Accessed: 2020-11-30.

[24] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099, 2015.

[25] Georg Schildbach. A new nonlinear model predictive control algorithm for vehicle path tracking. In *Advanced Vehicle Control: Proceedings of the 13th International Symposium on Advanced Vehicle Control (AVEC'16), September 13-16, 2016, Munich, Germany*, page 139. CRC Press, 2016.

[26] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[27] Cogniteam. V7 Hamster, a car-like-Robot, by cogniteam. `https://www.hamster-robot.com/`, 2019.

[28] Philip Polack, Florent Altché, Brigitte d'Andréa Novel, and Arnaud de La Fortelle. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *2017 IEEE intelligent vehicles symposium (IV)*, pages 812–818. IEEE, 2017.

[29] Hongbin Ren, Sizhong Chen, Lin Yang, and Yuzhuang Zhao. Optimal path planning and speed control integration strategy for ugvs in static and dynamic environments. *IEEE Transactions on Vehicular Technology*, 69(10):10619–10629, 2020.

[30] Phdthesis Mohamed Ibrahim. Design and real-time implementation of model predictive control for flight systems in presence of uncertainties. In *Model Predictive Control*, page 170, 2020.

[31] MathWorks Melda Ulusoy $https : //ch.mathworks.com/videos/understanding - model - predictive - control - part - 1 - why - use - mpc1526484715269.html?s_tid = srchtitle$. Understanding model predictive control, part 1: Why use mpc? In *Unmanned Ground Vehicle Technology VII*, 2016 ,.

[32] Hengyang Wang, Biao Liu, Xianyao Ping, and Quan An. Path tracking control for autonomous vehicles based on an improved mpc. *IEEE Access*, 7:161064–161073, 2019.

[33] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[34] Ryan Shaw. Obstacle avoidance of an unmanned ground vehicle using a combined approach of model predictive control and proportional navigation. *auburn university*, 2018.

[35] Mooryong Choi and Seibum B Choi. Mpc for vehicle lateral stability via differential braking and active front steering considering practical aspects. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 230(4):459–469, 2016.

[36] Susenjit Ghosh, Dhrupad Biswas, Desham Mitra, Somnath Sengupta, and Siddhartha Mukhopadhyay. Effect of soc uncertainty on mpc based energy management strategy for hevs. In *2020 IEEE 17th India Council International Conference (INDICON)*, pages 1–6. IEEE, 2020.

[37] F Künhe, J Gomes, and W Fetter. Mobile robot trajectory tracking using model predictive control. In *II IEEE latin-american robotics symposium*, volume 51. Citeseer, 2005.

[38] Boris Houska, Hans Joachim Ferreau, and Moritz Diehl. Acado toolkit—an open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[39] ACADO github, Acado toolkit. acado toolkit —, matlab-acado interface, 2014. [Online; accessed 29-September-2014].

[40] Royce Val C Malalis, Chyn Ira C Crisostomo, Romel S Saysay, Alexander C Abad, Lessandro Estelito O Garciano, and Renann G Baldovino. Design and simulation of a model predictive controller (mpc) for a seismic uniaxial shake table. *International Journal on Advanced Science, Engineering and Information Technology*, 10(3):1085, 2020.

[41] Harikumar Kandath, Titas Bera, Rajarshi Bardhan, and Suresh Sundaram. Autonomous navigation and sensorless obstacle avoidance for ugv with environment information from uav. In *2018 Second IEEE International Conference on Robotic Computing (IRC)*, pages 266–269. IEEE, 2018.

[42] Wikipedia contributors. Convex set — Wikipedia, the free encyclopedia 2021, 2004. [Online; accessed 22-July-2021].