POLITECNICO DI TORINO

Master Degree in Mechatronic Engineering

Master Degree Thesis

Optimal Trajectory Generation via Robust Model Predictive Control for a Four-Wheel Steering Electric Unmanned Ground Vehicle

Supervisors
prof. Fabrizio Dabbene
Martina Mammarella

Candidate
Ruotolo Vincenzo Pio
matricola: 257668

Academic Year 2020-2021

Abstract

The purpose of the present thesis is focused on the design of a Model Predictive Control approach, able to generate a feasible trajectory for a 4 steering wheel electric Unmanned Ground Vehicle in the Agriculture 4.0 scenario. A priori knowledge of the environment and static obstacles therein (i.e. vine rows) is given by a processed point-cloud map of a vineyard, provided by UAV recognitions. Based on this information, the problem of generating a feasible trajectory (guidance) and tracking it (control) are addressed by a suitable GNC system.

In literature it is possible to find a high amount of guidance and control strategies. About control strategies, it is possible to find approaches that depends on the application. Commonly control strategies are PID, LQR and H-inf usually for linear systems, Feedback linearization and Sliding mode for nonlinear systems. Recently have been developed new advanced control strategies derived by the contribution of fields outside control, as Model Predictive Control, Fuzzy logic, Neural Networks and Data-Driven.
Instead, about guidance strategies, a survey [5] comes to our help, listing lots of different methods. One of the most famous is the Rapidly-Expanding Random Tree (RRT), applied for the first time by LaValle in [1] and in [2], which adopts a stochastic search. Another method mentioned is Model Predictive Control, already listed in the control strategies. The MPC is a well known approach in controls, but in guidance, its standard formulation it is not a global optimal method, but it finds an optimal solution over a finite time horizon, and does not cover the entire path to generate. In order to cover the remaining path it is used to add a suitable cost-to-go in the problem, also called terminal cost, alongside a terminal constraint.
A complete guidance and navigation system using MPC, can be found in the proposed papers [9] and [10], which is used for lance change in automated highway driving context. Another approach based on Mixed-Integer Linear Programming for path planning is shown in [11].
It has been choosen the Model Predictive Control strategy to develop our GNC system, trying to pass over the problem of finding a global optimal solution, and providing a feasible trajectory.

This thesis is organized as follow: in part I are introduced the context of the thesis, an overview of possible system models for modelling the 4-steering wheel UGV, the optimization theory and a MPC overview; in part II it is are described the two design approaches; in part III are described the implementation of each approach and their results, drawing conclusions.

Contents

List of Tables

List of Figures

Part I

Introduction

Chapter 1

Thesis Framework

1.1 AgriTech: Robotics in Agriculture

Agricultural technology or in short Agritech, stands for the use of new technologies in agriculture fields. As mentioned on the report launched during the World Government Summit, called ”Agriculture 4.0 – The Future Of Farming Technology” [3], the new challenges to face are Demographics, Scarcity of natural resources, Climate change, and Food waste. As a consequence, future agriculture will have to make wide use of cutting-edge technologies, based on automated robotic systems, big data analysis and IoT (Internet of Things). This new wave of techonological improvement has been called Agritech, since it is a dramatic change in the way the agriculture has been conceived until now. Using these techonlogies, reduce the required time and cost in production, increasing the quality and efficiency, avoiding the waste of resources. Several applications have been already implemented in agricolture such as seeding, grove supervision, chemical applications, weed control, harvesting, end so on. Even if the agricultural environment is not structured as well as the other economic sectors, the flexibility of robots allows them to be exploited, taking into account the complexity of the tasks and the environments. This makes it so that this is a field in constant development and day after day new researches add new opportunities of growth.
Here we focus on automated robotic systems, in particular on automation of motion, which implies the use of unmanned aerial and ground vehicles. So as to take part in this initiative, the Dipartimento di Scienze Agrarie, Forestali, Alimentari (DISAFA) of University of Turin, in partnership with Politecnico di Torino and the Institute of Electronics, Computer and Telecommunication Engineering of National Research Council of Italy (CNR-IEIIT), has developed a four wheel steering unmanned ground vehicle.

1.2 Case study: Four-Wheel Steering Electric Unmanned Ground Vehicle

The unmanned ground vehicle under study is shown in in fig. 1.1, with parameters in tab. 1.1. It has 4 steering wheels with two Ackermann steering mechanisms (ASM), respectively for the front and rear axes, allowing the UGV to steer in small spaces. Furthermore, the UGV can move with a crab-wise motion, letting the UGV moving in a straight line, on a different direction other than its longitudinal axis.
The greatest advantage of a four wheel steering vehicle consists in reaching higher steering angles with respect to the common two wheel steering one, letting it to reduce the turning radius. This is a very common condition in Agritech, where ground vehicles are expected to moves in narrow spaces, which may be ***vineyard***, characterized by linear and long paths, and ***greenhouse***, which path is most of time irregular and narrow. Both cases are shown in fig. 1.2, where it can be noticed that our UGV fits well the first case, while for the second it may be more suitable using UGV of smaller dimensions.

Figure 1.1: UGV

Quantity	Value	Description
m	$800kg$	mass
l_f	$0\ 75m$	front axis length
l_r	$0\ 75m$	rear axis length
L	$1\ 5m$	UGV length
W	$1m$	UGV width
I_z	$112kg/m^3$	inertia
v_{max}	$2\ 13m/s$	maximum velocity
δ_{max}	$46°$	maximum steering angle
δ_{max}	$107°/s$	maximum steering rateo
R_w	$0\ 254m$	wheel radius
W_w	$0\ 203m$	wheel width
C_f, C_r	$10000N/rad$	cornering stiffness

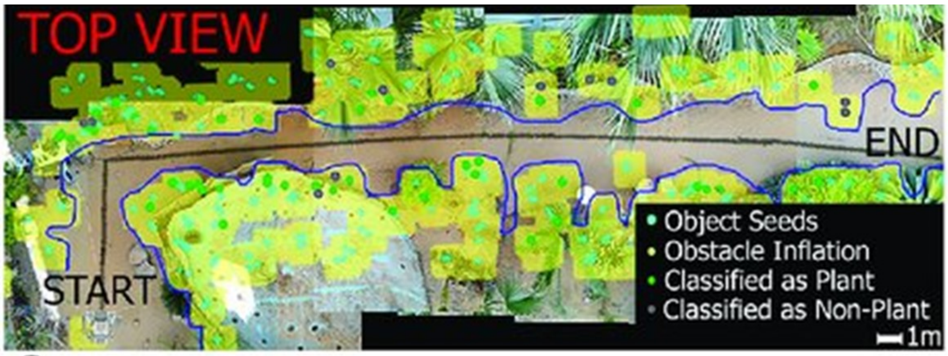
Table 1.1: UGV parameters

In aerospace, the process of design a system able to control the motion of a vehicle is called GNC (Guidance, Navigation and Control) [4], where:

- ***Guidance*** is the process of determination of a feasible trajectory
- ***Navigation*** is the process of determination of the vehicle state at a certain instant
- ***Control*** is the process of application of forces to the vehicle needed to perform the desired trajectory

This thesis deals only with Guidance and Control parts, then in the determination of a feasible trajectory from the inital position to a final goal, and the application of the suitable forces to perform that trajectory, meanwhile Navigation is not addressed directly, but we expected that UGV mounts a set of sensors that allows us to obtain information about its state at each instant of time.
For both Guidance and Control it has been chosen to use the Model Predictive Control approach in this thesis, which, even if it is well-known in control, in guidance it needs some improvement in order to overcome problems related infeasibility over the complete track. The strength of this approach consists in solving an optimization problem taking into account a prediction of a complex MIMO (Multiple Input Multiple Output) system, physical and design constraints, and the possibility to trade-off between user specified optimization parameters.

(a)



(b)

Figure 1.2: Agritech environments: a) Vineyard; b) Greenhouse;

Chapter 2

Reference frames and Coordinate systems

In this chapter it is given a description of the reference frames and the coordinate systems used in this thesis, since in the design, models can be expresseed in different coordinate systems.

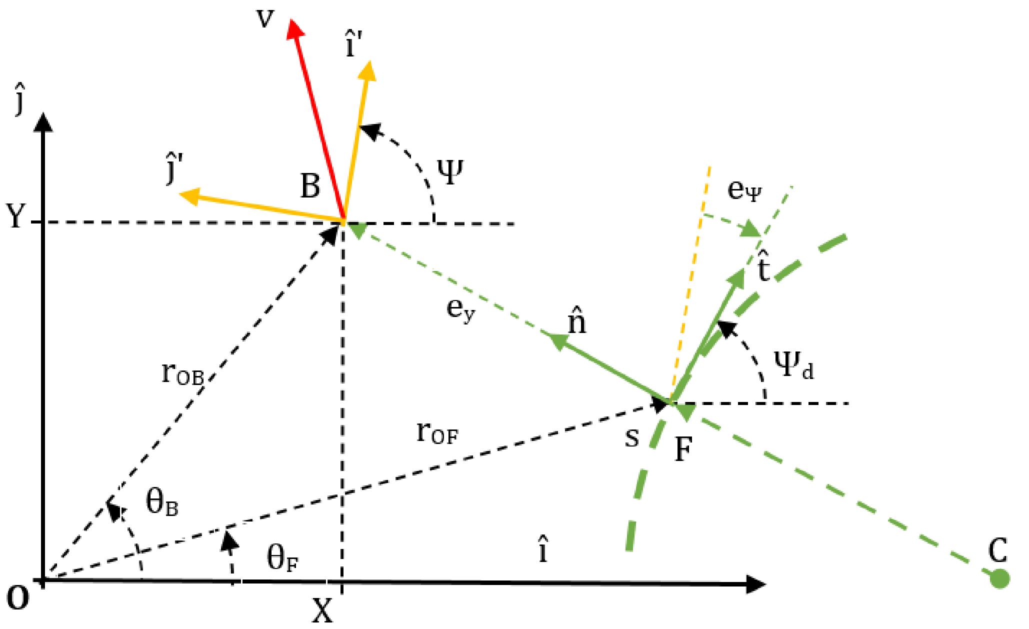


Figure 2.1: Global, body and track frames

It is considered a moving body frame B in a plane with respect to an inertial global frame O, and a Frenet frame F.

- Global frame $O(\hat{i}, \hat{j})$ defined as an inertial frame.
- Body frame $B(\hat{i}', \hat{j}')$ defined with the exes fixed to the body.
- Frenet frame $F(\hat{t}, \hat{n})$ defined by projecting the body to the track, with the axes tangent and normal to the curve.

The position of the body can be defined in cartesian coordinates (x, y) or curvilinear coordiantes (s, e_y).

2.1 Inertial-Frenet-Body transfrom

2.1.1 Rotation matrix

Rotation matrix between the inertial frame O, body frame B and Frenet frame F:

$$\begin{aligned} R_B^O(\psi) &= \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \\ R_F^O(\psi_d) &= \begin{bmatrix} \cos\psi_d & -\sin\psi_d \\ \sin\psi_d & \cos\psi_d \end{bmatrix} \\ R_B^F(e_\psi) &= \begin{bmatrix} \cos e_\psi & -\sin e_\psi \\ \sin e_\psi & \cos e_\psi \end{bmatrix} \end{aligned} \tag{2.1}$$

2.1.2 Curvilinear coordinates

The body can be expressed in curvilinear coordinates (s, e_y), which can be defined with respect to the Frenet frame F:

$$\begin{aligned} \vec{s} &= \vec{r}_{OF}^{F} \\ \vec{e}_y &= \vec{r}_{FB}^{F} \end{aligned} \tag{2.2}$$

According to the figure 2.2, the curvilinear abscissa s can be expressed in polar coordinates

$$s = \rho\theta \tag{2.3}$$

where: ρ : curve radius
$c_c = 1/\rho$: curvature
θ : angle travelled

The angle travelled θ, taking into account the rotation direction, can be expressed as:

$$\theta = \theta_0 + \theta' = \theta_0 - (90° - \psi_d) \tag{2.4}$$

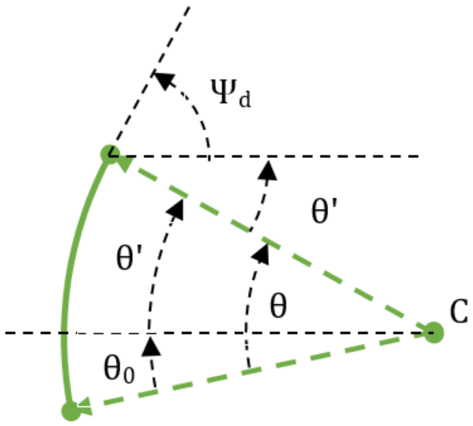


Figure 2.2: Angles relation

where θ_0 is an offset of the starting point to a specific position on the horizonal inertial axis.

Deriving the equation above:

$$\frac{d}{dt}(\theta) = \frac{d}{dt}(\theta_0 - (90° - \psi_d)) = \frac{d}{dt}(\psi_d) \tag{2.5}$$

which proves the relation of the angle travelled along the curve θ and the angle with respect to the tangent to the curve itself.

Deriving the curvilinear abscissa in polare coordinates in 2.3 with respect to the time, considering a constant curvature:

$$\vec{\dot{s}} = \frac{d}{dt}(\rho\theta)\hat{t} = \rho\frac{d}{dt}(\theta)\hat{t} = \rho\omega_d\hat{t} \tag{2.6}$$

then the angular velocity:

$$\omega_d = \frac{\dot{s}}{\rho} = \dot{s}c_c \tag{2.7}$$

It is defined the deviation e_ψ from the body heading to the curve (desired) heading, is:

$$e_\psi = \psi - \psi_d \tag{2.8}$$

2.1.3 Body with respect to the global frame

The position of the body frame B with respect the global frame O is:

$$\vec{r}^{\,O}_{OB} = \begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} r_{OB}\cos\theta_B \\ r_{OB}\sin\theta_B \end{bmatrix} \tag{2.9}$$

The velocity of the body in its reference frame $\vec{v}_B^B$, can be expressed in the global frame O as:

$$\vec{v}_B^O = R_B^O(\psi)\vec{v}_B^B \Rightarrow \vec{v}_B^B = R_O^B(\psi)\vec{v}_B^O \tag{2.10}$$

The acceleration of the body in its reference frame $\vec{a}_B^B$, can be expressed in the global frame O, deriving the equation above:

$$\begin{aligned}
\vec{a}_B^O &= \frac{d}{dt}(\vec{v}_B^O) = \frac{d}{dt}(R_B^O(\psi)\vec{v}_B^B) = \dot{R}_B^O(\psi)\vec{v}_B^B + R_B^O(\psi)\dot{\vec{v}}_B^B \\
&= \vec{\omega}_B^O \times R_B^O(\psi)\vec{v}_B^B + R_B^O(\psi)\dot{\vec{v}}_B^B = R_B^O(\psi)(\vec{\omega}_B^O \times \vec{v}_B^B + \dot{\vec{v}}_B^B) \\
\Rightarrow & R_O^B\vec{a}_B^O = \vec{\omega}_B^O \times \vec{v}_B^B + \dot{\vec{v}}_B^B
\end{aligned} \tag{2.11}$$

then:

$$\vec{a}_B^B = \vec{\omega}_B^O \times \vec{v}_B^B + \dot{\vec{v}}_B^B \tag{2.12}$$

where: $\vec{a}_B^B = R_O^B\vec{a}_B^O$

2.1.4 Frenet frame

The position of the Frenet frame F with respect the global frame O is:

$$\vec{r}_{OF}^O = \begin{bmatrix} x_F \\ y_F \end{bmatrix} = \begin{bmatrix} r_{OF}\cos\theta_F \\ r_{OF}\sin\theta_F \end{bmatrix} \tag{2.13}$$

The velocity of the body in its reference frame $\vec{v}_B^B$, can be expressed in the Frenet frame F as:

$$\vec{v}_B^F = R_B^F(e_\psi)\vec{v}_B^B \Rightarrow \vec{v}_B^B = R_F^B(e_\psi)\vec{v}_B^F \tag{2.14}$$

2.1.5 Body with respect to the Frenet frame

The position of the body can be expressed as:

$$\vec{r}_{OB}^O = \vec{r}_{OF}^O + \vec{r}_{FB}^O = \vec{r}_{OF}^O + R_F^O(\psi_d)\vec{r}_{FB}^F \tag{2.15}$$

According to [13], deriving the equation above with respect the time:

$$\begin{aligned}
&\frac{d}{dt}(\vec{r}_{OB}^O) = \frac{d}{dt}(\vec{r}_{OF}^O) + \frac{d}{dt}(R_F^O(\psi_d)\vec{r}_{FB}^F) \\
\Rightarrow & \vec{v}_B^O = \frac{d}{dt}(\vec{r}_{OF}^O) + \dot{R}_F^O(\psi_d)\vec{r}_{FB}^F + R_F^O(\psi_d)\dot{\vec{r}}_{FB}^F \\
\Rightarrow & \vec{v}_B^O = \dot{\vec{r}}_{OF}^O + \vec{\omega}_F^O \times R_F^O(\psi_d)\vec{r}_{FB}^F + R_F^O(\psi_d)\dot{\vec{r}}_{FB}^F \\
\Rightarrow & R_O^F(\psi_d)\vec{v}_B^O = R_O^F(\psi_d)\dot{\vec{r}}_{OF}^O + \vec{\omega}_F^O \times \vec{r}_{FB}^F + \dot{\vec{r}}_{FB}^F \\
\Rightarrow & R_O^F(\psi_d)R_B^O(\psi)\vec{v}_B^B = \dot{\vec{r}}_{OF}^F + \vec{\omega}_F^O \times \vec{r}_{FB}^F + \dot{\vec{r}}_{FB}^F \\
\Rightarrow & R_B^F(e_\psi)\vec{v}_B^B = \dot{\vec{r}}_{OF}^F + \vec{\omega}_F^O \times \vec{r}_{FB}^F + \dot{\vec{r}}_{FB}^F
\end{aligned} \tag{2.16}$$

where: $R_B^F(e_\psi) = R_O^F(\psi_d)R_B^O(\psi)$
$\vec{\dot{r}}_{OF}^F = \dot{s}\hat{t}$
$\vec{\dot{r}}_{FB}^F = \dot{e}_y\hat{n}$

Passing from the cartesian coordinates to the curvilinear coordinates, using the transformation in 2.2:

$$\begin{aligned}
R_B^F(e_\psi)\vec{v}_B^B &= \vec{\dot{s}} + \vec{\omega}_d \times \vec{e}_y + \vec{\dot{e}}_y \\
\Rightarrow R_B^F(e_\psi)\vec{v}_B^B &= \dot{s}\hat{t} + \dot{s}c_c\hat{k} \times e_y\hat{n} + \dot{e}_y\hat{n} \\
\Rightarrow R_B^F(e_\psi)\vec{v}_B^B &= \dot{s}\hat{t} + \dot{s}e_y c_c(-\hat{t}) + \dot{e}_y\hat{n} \\
\Rightarrow R_B^F(e_\psi)\vec{v}_B^B &= \dot{s}(1 - e_y c_c)\hat{t} + \dot{e}_y\hat{n}
\end{aligned} \tag{2.17}$$

which can be decompoed in the Frenet frame axes:

$$\begin{bmatrix} \dot{s} \\ \dot{e}_y \end{bmatrix} = R_B^F(e_\psi) \begin{bmatrix} v_x \\ v_y \end{bmatrix} \begin{bmatrix} \frac{1}{1-c_c e_y} \\ 0 \end{bmatrix} \tag{2.18}$$

while the deviation angle velocity:

$$\vec{\dot{e}}_\psi = \frac{d}{dt}(\psi - \psi_d)\hat{k} = \vec{\omega} - \vec{\omega}_d = \vec{\omega} - \dot{s}c_c\hat{k} \tag{2.19}$$

Chapter 3

System Modeling

A system model is an important part of the MPC for prediction, which consists in a set of kinematics/dynamics equations that governs the system. In general it is spent a quite large amount of time building a suitable system model, but we can say that holds the following rule: the simplest model that gives accurate enough prediction.

It is possible to distinguish two types of models: kinematic models and dynamic models. The first are mainly used in low-speed conditions, where forces and inertia does not affect too much the vehicle; while the second keep into account forces and their effects.
In this chapter is given a description of the most common models used to represent a physics of a car. These models are commonly expressed in the so-called State-Space representation [6], which is a set of differential equations:

$$\begin{cases} \dot{x}(t) = \frac{d}{dt}x(t) = f(x(t), u(t)) \quad \text{state equation} \\ y(t) = h(x(t)) \quad \text{output equation} \end{cases} \tag{3.1}$$

where: $x(t)$: state
$u(t)$: input
$y(t)$: output

In other words, a system behaviour can be described by a function of its states and inputs.

3.1 Point mass kinematic

The point-mass linear kinematic [9] is the easiest way to represent a moving object in the space, which is expressed by this set of equations:

$$\begin{cases} \dot{X} = V_x \\ \dot{Y} = V_y \\ \dot{V}_x = u_1 \\ \dot{V}_y = u_2 \end{cases} \tag{3.2}$$

where: X, Y : coordinates with respect to the inertial frame
V_x, V_y : longitudinal and lateral velocity in inertial frame

3.2 Bicycle kinematics

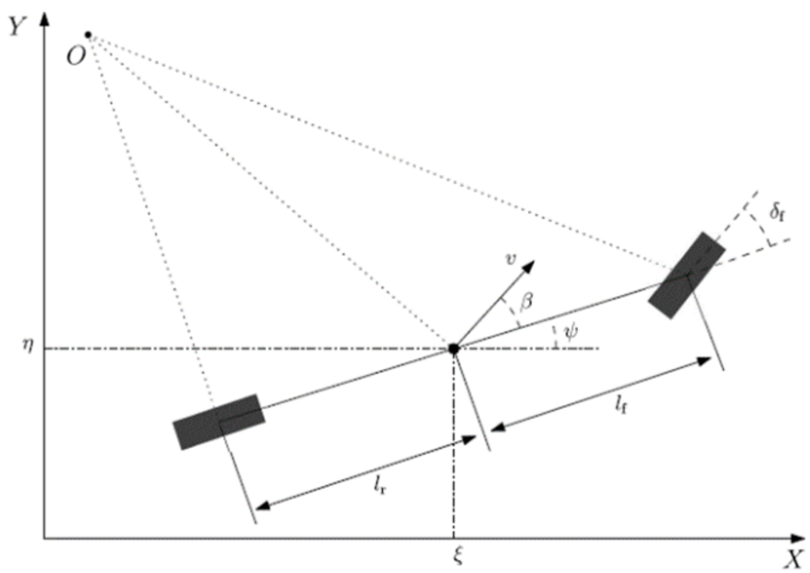


Figure 3.1: Bicycle kinematic

The kinematic model of a bicycle [12] is the simplest bycicle model, expressed by the following set of equations:

$$\begin{aligned}
\dot{X}(t) &= v\cos(\psi+\beta) = V_x\cos(\psi) - V_y\sin(\psi) \\
\dot{Y}(t) &= v\sin(\psi+\beta) = V_x\sin(\psi) + V_y\cos(\psi) \\
\dot{\psi}(t) &= \frac{v\cos(\beta)}{l_f+l_r}(\tan(\delta_f)+\tan(\delta_r))
\end{aligned} \tag{3.3}$$

where: $X, Y:$ coordinates with respect to the inertial frame
$V_x, V_y:$ longitudinal and lateral velocity with respect to the inertial frame
$v:$ velocity in body frame
$\delta_f, \delta_r:$ front and rear steering angles
$\psi:$ heading with respect to the inertial frame
$\beta = \arctan\left(\frac{l_f\tan(\delta_r)+l_r\tan(\delta_f)}{l_f+l_r}\right):$ slip angle

which is a nonlinear system of the form:

$$\dot{\xi}(t) = f(\xi(t), u(t)) \tag{3.4}$$

where: $\xi(t) = \begin{bmatrix} X & Y & \psi \end{bmatrix}^T$ state
$u(t) = \begin{bmatrix} \delta_f & \delta_r & v \end{bmatrix}^T$ input

3.3 Bicycle dynamics

The bycicle dynamics described in [12] and [15], is one of the most used models, since it is rich enough to integrate the relevant physical aspects of a car, but still not to complex as a full car model.

3.3.1 Equation of motion

From equations 2.12, the acceleration of the body in its own frame B, is:

$$\begin{aligned}
\vec{a}_B &= \vec{\omega}_z \times \vec{v}_B + \dot{\vec{v}}_B = \omega_z\hat{k} \times (v_x\hat{i} + v_y\hat{j}) + (\dot{v}_x\hat{i} + \dot{v}_y\hat{j}) \\
&= \omega_z v_x\hat{j} + \omega_z v_y(-\hat{i}) + (\dot{v}_x\hat{i} + \dot{v}_y\hat{j}) \\
&= (\dot{v}_x - \omega_z v_y)\hat{i} + (\dot{v}_y + \omega_z v_x)\hat{j}
\end{aligned} \tag{3.5}$$

then:

$$\vec{a}_B = \begin{bmatrix} a_x \\ a_y \end{bmatrix} = \begin{bmatrix} \dot{v}_x - \omega_z v_y \\ \dot{v}_y + \omega_z v_x \end{bmatrix} \tag{3.6}$$

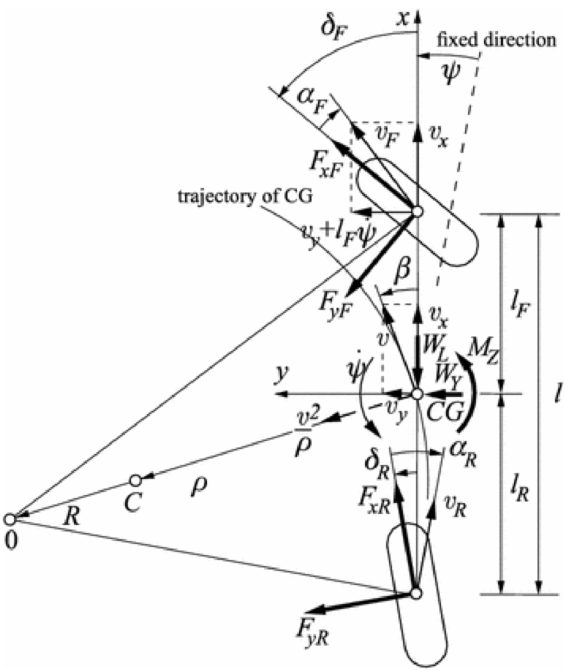


Figure 3.2: Bicycle dynamics

According to the equilibrium principle, expressed in dynamics, it is obtained the set of equation:

$$\begin{aligned} m(\dot{v}_x - \omega_z v_y) &= F_{xf} + F_{xr} \\ m(\dot{v}_y + \omega_z v_x) &= F_{yf} + F_{yr} \\ I_z \dot{\omega}_z &= l_f F_{yf} - l_r F_{yr} \end{aligned} \tag{3.7}$$

then:

$$\begin{aligned} \dot{v}_x &= \frac{1}{m}(F_{xf} + F_{xr}) + \omega_z v_y \\ \dot{v}_y &= \frac{1}{m}(F_{yf} + F_{yr}) - \omega_z v_x \\ \dot{\omega}_z &= \frac{1}{I_z}(l_f F_{yf} - l_r F_{yr}) \end{aligned} \tag{3.8}$$

where: $m:$ mass

$l_f, l_r:$ front and rear wheel distance from the center of mass

$F_{xf}, F_{xr}:$ forces on x body axis of front (f) and rear (r) wheel

$F_{yf}, F_{yr}:$ forces on y body axis of front (f) and rear (r) wheel

in which the forces can be decomposed in the tire-fixed frame:

$$\begin{cases} F_{x,i} = F_{l,i} cos(\delta_i) - F_{s,i} sin(\delta_i) \\ F_{y,i} = F_{l,i} sin(\delta_i) + F_{s,i} cos(\delta_i) \end{cases} \tag{3.9}$$

where: $F_{l,i} = \{F_{lf}, F_{lr}\}$: force on longitudinal axis of front (f) and rear (r) wheel
$F_{s,i} = \{F_{sf}, F_{sr}\}$: force on perpendicular axis of front (f) and rear (r) wheel
$\delta_i = \{\delta_f, \delta_r\}$: front (f) and rear (r) wheel steering angles

3.3.2 State Space equation

Replacing the equations 3.9 into 3.8, it is obtained the complete formulation:

$$\begin{aligned}
\dot{v}_x &= \frac{1}{m}(F_{lf}\cos(\delta_f) - F_{sf}\sin(\delta_f) + F_{lr}\cos(\delta_r) - F_{sr}\sin(\delta_r)) + \omega_z v_y \\
\dot{v}_y &= \frac{1}{m}(F_{lf}\sin(\delta_f) + F_{sf}\cos(\delta_f) + F_{lr}\sin(\delta_r) + F_{sr}\cos(\delta_r)) - \omega_z v_x \\
\dot{\omega}_z &= \frac{1}{I_z}(l_f(F_{lf}\sin(\delta_f) + F_{sf}\cos(\delta_f)) - l_r(F_{lr}\sin(\delta_r) + F_{sr}\cos(\delta_r)))
\end{aligned} \tag{3.10}$$

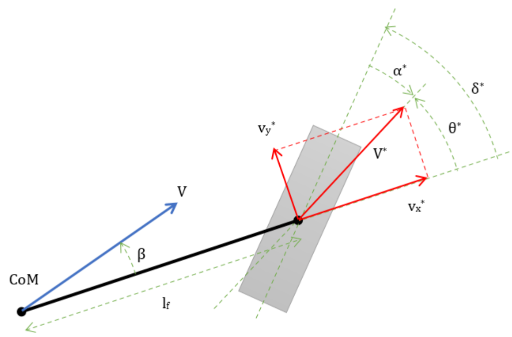


Figure 3.3: Sideslip angle

Sideslip angle The sideslip angle is defined in [12] as:

$$\alpha_* = \delta_* - \theta_* \tag{3.11}$$

where: $\theta_* = \arctan\left(\frac{v_{x*}}{v_{y*}}\right)$
v_{x*}, v_{y*} : longitudinal and lateral wheel velocities in fixed body-frame

While v_{x*} coincides with the CoM longitudinal velocity, the v_{y*} is the sum of the CoM lateral velocity and the tangential velocity of the wheel respect to the CoM due to the angular velocity, then:

$$\begin{aligned}
v_{xf} &= v_{xr} = v_x \\
v_{yf} &= v_y + l_f\omega_z \\
v_{yr} &= v_y - l_r\omega_z
\end{aligned} \tag{3.12}$$

where: v_{xf}, v_{yf} : front wheel velocities in fixed body-frame
v_{xr}, v_{yr} : rear wheel velocities in fixed body-frame

Replacing the equation above in the first one, it is obtained the sideslip angle of the front and the rear wheel:

$$\begin{aligned} \alpha_f &= \delta_f - \arctan\left(\frac{v_y + l_f\omega_z}{v_x}\right) \\ \alpha_r &= \delta_r - \arctan\left(\frac{v_y - l_r\omega_z}{v_x}\right) \end{aligned} \tag{3.13}$$

3.4 Tire's linear dynamic

For small sideslip angles, the wheel lateral force can be well approximated proportional to the sideslip angle itself [15]:

$$F_{si} = C_i(F_z, \mu)\alpha_i \tag{3.14}$$

C_i is the wheel "Cornering stiffness", depending on the normal force acting on the wheel and the friction coefficient with the surface.
Here, an estimate of this coefficient has been done evaluating the dynamic evolution of the dynamic system model, with different values of the cornering stiffness, until a realistic behaviour has been achieved.
Since it directly affects the wheel lateral forces, it is worthy to notice that small values of it, make difficult to steer, while too high values generate excessively high lateral forces, which are not realistic.

Chapter 4

Optimization

4.1 Introduction

Optimization is the other foundamental part of the MPC, for obtaining the optimal input to apply to the system.
Optimization, as exposed in [7], consists in minimizing an objective function $f(x)$, through a decision variable x, subjected to constraints $g(x), h(x)$, which can be expressed In the standard form:

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & g(x) \leqslant 0 \\ & h(x) = 0 \end{aligned} \tag{4.1}$$

Objective function The objective function is called also cost/loss function since it implies a penalization of the terms involved, which represent some performance indices. The penalty is often expressed as the $l_2 norm$ which is always convex and can be interpreted in penalizing the energy dissipation.
The performance indices depend on what we want to control. Most of the time, these indices are in opposition from each other, then it is performed a trade-off between tracking a reference and input effort, then between ***stability*** and ***performance***.

Decision variable The decision variable or minimizer is the solution of the optimization problem. If a solution exists, the problem is ***feasible***, and the values that minimizes the objective function is the solution, which results in an ***optimal value***. Feasibility and optimality are the main issues of an optimization problems.

Constraints The constraints can be related to the system model or to some design parameters. Usually, physics constraints are hard ***constraints***, since they cannot be violated; while design constraints are often ***soft constraints*** where their violation is allowed within some defined tolerances.

4.2 Optimization Programs

The first distinction in optimization problems is between ***Convex*** and ***Non-Convex*** problems. Convex problems are special case of the Non-Convex ones, where the local optimum is unique and corresponde to the global optimum. Convex optimization problems can be often reformulated according a family of optimization programs the most common of which are:

- Linear Programs (LP)

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & A_{eq}x = b_{eq} \end{aligned} \tag{4.2}$$

- Quadratic Programs (QP)

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}x^T H x + c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & A_{eq}x = b_{eq} \end{aligned} \tag{4.3}$$

- Second-Order Cone Program (SOCP)

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & F_0 + \sum_{i=1}^{m} x_i F_i \succeq 0 \end{aligned} \tag{4.4}$$

Chapter 5

Model Predictive Control

5.1 MPC Introduction

Model Predictive Control is a control approach, containing different implementations, based on specific key principles [8]:

- ***Prediction***: predict the system future behavior.
- ***Optimization***: solve an optimization problem.
- ***Receding horizon***: apply only the first solution of the optimization problem and shift forward the horizon.

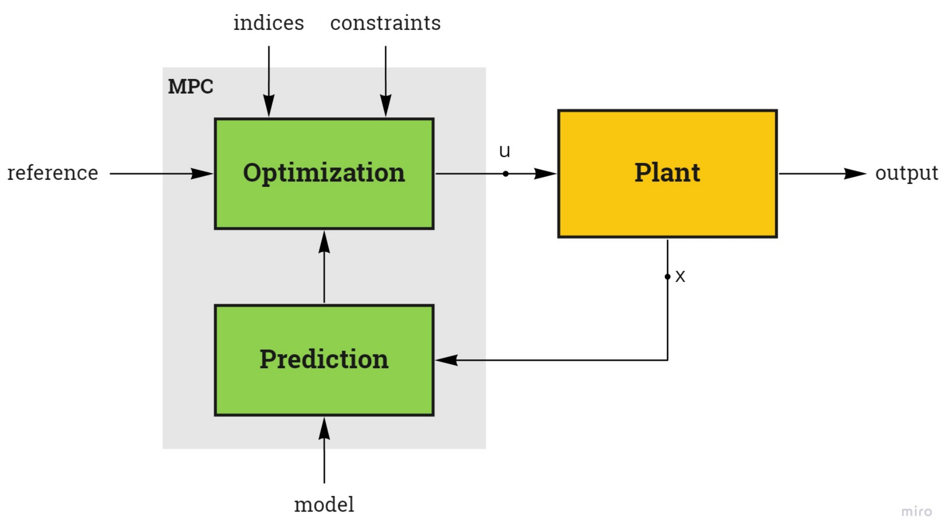


Figure 5.1: MPC block scheme

Taking as reference the fig. 5.1, it is possible to outline its algorithm, in fact at each sampling time k, the MPC:

1. performs a ***prediction*** of the future behavior of the system over a time horizon N_p, through a model of the system and its past states and inputs:

$$x(i+1|k) = f(x(i|k), u(i|k)) \tag{5.1}$$

2. obtains an optimal control sequence by solving an ***optimization problem*** over a finite prediction horizon N_p:

$$\begin{aligned} \min_{u_k} \quad & \sum_{i=0}^{N_p-1} l(x(t), u(t)) \\ \text{s.t.} \quad & \text{system model} \\ & \text{design constraints} \end{aligned} \tag{5.2}$$

3. applies only the first term of the input sequence to the system and shifting forward the time horizon, according to ***receding horizon*** principle, and obtain the resulting state which is used as initial state for the new prediction:

$$u_k = u^*(0|k) \in U_k^* \tag{5.3}$$

5.2 MPC Formulations

5.2.1 Standard formulation

The classic MPC is formulated according the following assumptions:

- linear system model (LTI)
- no uncertainties affects the system model
- no disturbances affects the real system

The prediction model is a LTI system usually in state-space representation in discrete time domain:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \rightarrow \begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{cases} \tag{5.4}$$

The optimization problem consist in minimizing a quadratic cost (square loss) subject to linear constraints

$$
\begin{aligned}
\min_{U_k} \quad & \sum_{i=0}^{N_p-1} \left(\left\| y_{i|k} - y_{r_{i|k}} \right\|_Q^2 + \left\| u_{i|k} \right\|_R^2 \right) \\
\text{s.t.} \quad & x_{i+1|k} = Ax_{i|k} + Bu_{i|k} \\
& x_k = x_{0|k} \\
& x_{min} \leqslant x_{i|k} \leqslant x_{max} \\
& u_{min} \leqslant u_{i|k} \leqslant u_{max} \\
& \Delta u_{min} \leqslant \Delta u_{i|k} \leqslant \Delta u_{max}
\end{aligned}
\tag{5.5}
$$

which can be always reformulated as QP:

$$
\begin{aligned}
\min_{U_k} \quad & \sum_{i=0}^{N_p-1} U_k^T H U_k + 2f^T U_k + g \\
\text{s.t.} \quad & A_{ineq} U_k = b_{ineq}
\end{aligned}
\tag{5.6}
$$

The solution is the optimal control input sequence:

$$
U_k^* = \begin{bmatrix} u_{0|k}^* & u_{1|k}^* & \cdots & u_{N_p-1|k}^* \end{bmatrix}^T \tag{5.7}
$$

from which is applied only the first input to the system:

$$
u_k^* = u_{0|k}^* \tag{5.8}
$$

Observation 1 MPC performance degrade easily as the nonlinearities and disturbances affect the system.

5.2.2 Nonlinearities formulation

Most of systems (if not all) are nonlinear, then many strategies are been developed to deal with nonlinearities:

- **Adaptive MPC**: the system model is linear, but the system matrices are parameter dependant, constant in prediction.
  - ***LPV MPC*** (Linear Parameter Varying MPC): It is on considering a set of finite operative points, around which linearizing the system.
  - ***SL MPC*** (Successive Linearization MPC): It is based on linearizing online the system at each iteration according to a varying operative point (current/equilibrium state).
- **LTV MPC** (Linear Time Varying MPC): the system model is linear, but the system matrices are time dependant.

- **NMPC** (Nonlinear MPC): the system model is nonlinear, implying the loss of convexity.
- ***Hybrid MPC***: the plant is coupled with a controller requiring logic actions, which involves in integer constraints.

5.2.3 Uncertainties formulation

In presence of uncertainties, instead, according to the properties of these disturbances, you should follow one of the two approaches:

- **RMPC** (Robust MPC): If the disturbance is bounded it is possible to design a robust MPC that ensure that the constraints are satisfied for all the possible disturbances sequence.
- **SMPC** (Stochastic MPC): If the disturbance is unbounded it is possible to design a stochastic MPC to ensure that the constraints are satisfied on a specific probability.

Some approaches to RMPC and SMPC are:

- ***Min-max MPC***: The optimization problem is performed with respect to all possible trajectory evolutions.
- ***Constraint Tightening MPC***: The optimization is subject to enlarged constraints by a given margin, so that the trajectory feasibility is met.
- ***Tube-based MPC***: The optimization is done over a nominal dynamic model, while a feedback controller ensures the convergence of the actual state to the nominal one.

5.2.4 Data-Driven formulation

The need not to rely completely on a priori defined model of the system, pushed to develop the **Data-Driven MPC** approach:

- ***LMPC*** (Learning MPC) [16]: The MPC solve the optimization problem over a number of iterations of the same task, in which the terminal cost and the terminal constraints ensure increasing of performance at each iteration.

5.3 MPC Improvement

5.3.1 Prediction and Control Horizon

In order to reduce the overall time for solving the optimization by the solver, it is possible to reduce the sequence of inputs to take into accounts. This is translated into optimizing over a control horizon which is smaller than the prediction horizon, usually between 1-5.

$$\begin{aligned}
\min_{U_k} \quad & \sum_{i=0}^{N_p-1} \left\| y_{i|k} - y_{r_{i|k}} \right\|_Q^2 + \sum_{i=0}^{N_c} \left\| u_{i|k} \right\|_R^2 \\
\text{s.t.} \quad & x_{i+1|k} = Ax_{i|k} + Bu_{i|k}, \quad i \in \left[1, N_p\right] \\
& x_k = x_{0|k} \\
& x_{min} \leqslant x_{i|k} \leqslant x_{max}, \quad i \in \left[1, N_p\right] \\
& u_{min} \leqslant u_{i|k} \leqslant u_{max}, \quad i \in [0, N_c - 1] \\
& \Delta u_{min} \leqslant \Delta u_{i|k} \leqslant \Delta u_{max}, \quad i \in [0, N_c - 1]
\end{aligned} \tag{5.9}$$

where: $U_k = \begin{bmatrix} u_{0|k} & u_{1|k} & \cdots & u_{N_c-1|k} \end{bmatrix}^T \in \mathbb{R}^{n_u N_c}$

5.3.2 Optimization and Prediction sampling time

In general, the requirements for the optimization (computation cost) do not match the requirements for system evolution (system dynamics). Due to that the sampling time for prediction can be different from the sampling time for the system:

$$T_{sys} = \frac{T_{mpc}}{m}$$

where: m is an integer

5.3.3 Soft constraints handling

In some conditions, in case of infeasibility, it is possible to violate some of these constraints, adding a slack variable to the problem.

$$\begin{aligned}
\min_{U_k} \quad & \sum_{i=0}^{N_p-1} \left\| x_{i|k} - r_{i|k} \right\|_Q^2 + \sum_{i=0}^{N_c} \left\| u_{i|k} \right\|_R^2 + \rho\epsilon^2 \\
\text{s.t.} \quad & x_{i+1|k} = Ax_{i|k} + Bu_{i|k}, \quad i \in \left[1, N_p\right] \\
& x_k = x_{0|k} \\
& x_{min} \leqslant x_{i|k} \leqslant x_{max}, \quad i \in \left[1, N_p\right] \\
& u_{min} \leqslant u_{i|k} \leqslant u_{max}, \quad i \in [0, N_c - 1] \\
& \Delta u_{min} \leqslant \Delta u_{i|k} \leqslant \Delta u_{max}, \quad i \in [0, N_c - 1] \\
& y_{min} - \epsilon V_{min} \leq y_{i|k} \leq y_{max} + \epsilon V_{max}
\end{aligned} \tag{5.10}$$

where: $\rho \gg Q$

5.4 Feasibility and Stability issues

Considering the standard MPC formulation in close-loop, recursive feasibility and closed-loop stability are not guaranteed, since the difference between predicted response and closed-loop response.
The MPC is ***recursive feasible***, if the system is initialized from a feasible initial state x_0, then there exists a feasible solution to the same problem for all the future samplings.
The MPC is ***asymptotically stable***, if the system converges asymptotically to steady-state at infinite time.

Standard MPC solves locally (over a finite prediction horizon) the infinite horizon optimization problem for which stability is guaranteed if the optimization is feasible:

$$\begin{aligned} J^*_\infty(x_0) \min_{U_k} \quad & \sum_{k=0}^{\infty} h(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k) \\ & x_k \in X_c \\ & u_k \in U_c \end{aligned} \tag{5.11}$$

The infinite horizon optimization problem can be reformulated as MPC problem through the ***dual mode approach***:

- Mode I: optimal input of the MPC over a finite horizon optimization
- Mode II: optimal input of the stabilizing feedback law over an infinite horizon optimization

Replacing the tail (infinite horizon) with the cost-to-go at horizon edge, it is obtained the compelte formulation:

$$\begin{aligned} \min_{U_k} \quad & \sum_{i=0}^{N_p-1} h(x_k, u_k) + V_f(x_{N_p}) \\ \text{s.t.} \quad & x_{i+1|k} = f(x_{i|k}, u_{i|k}) \\ & x_{i|k} \in X_c \\ & u_{i|k} \in U_c \\ & x_{N_p|k} \in X_f \end{aligned} \tag{5.12}$$

where: $V_f(x_{N_p})$ terminal cost (cost-to-go)
X_f terminal constraint set (terminal region)

Definition The closed loop stability and the recursive feasibility of the system are guaranteed, if for any $x_0 \in \Gamma_{N_p}$, the following conditions are satisfied:

1. V is a Lyapunov function:

$$V(x) \geq min_{u \in U_c} h(x,u) + V(f(x,u))$$

2. X_f is an invariant set

$$f(x,u) \in X_f$$

where: $x_0 \in \Gamma_{N_p}$ domain of attraction

The objective is to obtain the largest domain of attraction.

5.5 Discussion

The MPC approach has become popular for many advantages with respect to common control techniques since is an extremely flexible control design, able to cope to many different challenges. At the same time, it is a performance-oriented method easy to tune, allows to handle constraints, and can be easily used for MIMO systems. One of the biggest drawbacks of the MPC is its computational cost in term of resources and time, beside of the fact that it is a suboptimal optimization approach.

Part II

MPC Design

Chapter 6

2-layer MPC

The first design is a 2-layer MPC, which is a common approach used in [9] and [10] for autonomous drive lane change, where it is applied for automated highway driving, where a high-level MPC is used for guidance, while a low-level MPC for control. The difference in this case, consists in planning not just a manoeuvre, but instead a full trajectory.
The high-level planner utilizes a point-mass kineamtic model and linear collision avoidance constraints to computes a manoeuvre. Has been provided a discrete reference points between the intial position and the goal, in practice, ordered series of waypoints has been provided in order to divide the full path into smaller pieces. At this level is addressed the obstacle avoidance into MPC contraints, in a way that is ensured convexity since the MPC is convex. Since the UGV has to move in a path laterally delimited, it is possible to consider the space of free movement, as a subset of the entire available space. Provided a full description of the map in terms of cartesian coordinates, only the convex subset of the map in which the UGV is located, is active into the constraints.
While the low-level control system uses a non-linear bycicle kinematic model, in order to compute the control inputs, having as reference the trajectory obtained above. Due to the nonlinear nature of the bicycle models, a so-called Successive Linearization Adaptive MPC or SL MPC is exploited, as proposed in [14]. This MPC formulation consists in using a LPV (Linear Parameter Varying) system for prediction, obtained by linearizing the nonlinear system around a varying operative point, which can be in general an equilibrium point or the actual state of the system. In this way, it is possible to guarantee the full range of operative conditions, continuing to preserve the simplicity of a linear MPC.

6.1 MPC Formulation

6.1.1 Prediction

At $k-th$ sampled, the predicted state at instant i, is:

$$\xi_{i+1|k} = A_d \xi_{i|k} + B_d u_{i|k} + K_d, \quad \xi_{0|k} = \xi_k \tag{6.1}$$

The state prediction over the horizon can be stack as:

$$\bar{\xi}_k = \bar{A}_d \xi_k + \bar{B}_d U_k + \bar{K}_d \tag{6.2}$$

where: $\bar{\xi}_k = \begin{bmatrix} \xi_{1|k} \\ \vdots \\ \xi_{N_p|k} \end{bmatrix} \in \mathbb{R}^{n_x N_p}; \quad U_k = \begin{bmatrix} u_{0|k} \\ \vdots \\ u_{N_c-1|k} \end{bmatrix} \in \mathbb{R}^{n_u N_c};$

$$\bar{A}_d = \begin{bmatrix} A_d \\ A_d^2 \\ \vdots \\ A_d^{N_p} \end{bmatrix} \in \mathbb{R}^{n_x N_p, n_x};$$

$$\bar{B}_d = \begin{bmatrix} B_d & 0^{n_x,n_u} & \cdots & 0^{n_x,n_u} \\ A_d B_d & B_d & \cdots & 0^{n_x,n_u} \\ \vdots & \vdots & \ddots & \vdots \\ A_d^{N_c} B_d & A_d^{N_c-1} B_d & \cdots & B_d \\ \vdots & \vdots & \ddots & \vdots \\ A_d^{N_p-1} B_d & A_d^{N_p-2} B_d & \cdots & A_d^{N_p-N_c} B_d \end{bmatrix} \in \mathbb{R}^{n_x N_p, n_u N_c}$$

$$\bar{K}_d = \begin{bmatrix} K_d \\ A_d K_d + K_d \\ \vdots \\ \sum_{i=0}^{N_p-1} A_d^p K_d \end{bmatrix} \in \mathbb{R}^{n_x N_p};$$

6.1.2 Optimization Problem

The optimization problem consists in minimizing over the error between the state and the reference over a prediction horizon, and at the same time the command input, over a short control horizon, both of them as square loss. The constraints are the common system model constraint and the physical constraints on state, inpput and input rate.

$$
\begin{aligned}
\min_{U_k} \quad & \sum_{i=1}^{N_p} \left\|\xi_{i|k} - ref_{i|k}\right\|_Q^2 + \sum_{i=0}^{N_c-1} \left\|u_{i|k}\right\|_R^2 \\
\text{s.t.} \quad & \xi_{i+1|k} = A_d(p)\xi_{i|k} + B_d(p)u_{i|k} + K_d(p), \quad i \in [1,\ ...,\ N_p] \\
& \xi_k = \xi_{0|k} \\
& \xi_{min} \leqslant \xi_{i|k} \leqslant \xi_{max}, \quad i \in [1,\ ...,\ N_p] \\
& u_{min} \leqslant u_{i|k} \leqslant u_{max}, \quad i \in [0,\ ...,\ N_c - 1] \\
& \Delta u_{min} \leqslant \Delta u_{i|k} \leqslant \Delta u_{max}, \quad i \in [0,\ ...,\ N_c - 1]
\end{aligned}
\tag{6.3}
$$

Objective function

$$
\begin{aligned}
J(\xi_k) &= \sum_{i=1}^{N_p} \left\|\xi_{i|k} - ref_{i|k}\right\|_Q^2 + \sum_{i=0}^{N_c-1} \left\|u_{i|k}\right\|_R^2 \\
&= \left\|\bar{\xi}_k - ref_k\right\|_{\bar{Q}}^2 + \left\|U_k\right\|_{\bar{R}}^2 \\
&= U_k^T \left(\bar{R} + \bar{B}_d^T \bar{Q} \bar{B}_d\right) U_k + 2\left(\bar{A}_d \xi_k + \bar{K}_d - ref_k\right)^T \bar{Q}\bar{B}_d U_k \\
&+ \left(\bar{A}_d \xi_k + \bar{K}_d - ref_k\right)^T \bar{Q} \left(\bar{A}_d \xi_k + \bar{K}_d - ref_k\right)
\end{aligned}
\tag{6.4}
$$

where: $ref_k = \begin{bmatrix} ref_{1|k} & \cdots & ref_{N_p|k} \end{bmatrix}^T \in \mathbb{R}^{n_x N_p}$

$$
\bar{Q} = \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q \end{bmatrix} \in \mathbb{R}^{n_x N_p, n_x N_p}
$$

$$
\bar{R} = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R \end{bmatrix} \in \mathbb{R}^{n_u N_c, n_u N_c}
$$

Constraints

State constraints

$$
\xi_{min} \leq \xi_{i|k} \leq \xi_{max} \Rightarrow \begin{cases} -\bar{B}_d U_k \leq -\bar{\xi}_{min} + \bar{A}_d \xi_k + \bar{K}_d \\ \bar{B}_d U_k \leq \bar{\xi}_{max} - \bar{A}_d \xi_k - \bar{K}_d \end{cases}
\tag{6.5}
$$

where: $\bar{\xi}_{min} = \begin{bmatrix} \xi_{min} & \cdots & \xi_{min} \end{bmatrix}^T \in \mathbb{R}^{n_x N_p}$
$\bar{\xi}_{max} = \begin{bmatrix} \xi_{max} & \cdots & \xi_{max} \end{bmatrix}^T \in \mathbb{R}^{n_x N_p}$

Input constraint

$$
u_{min} \leq u_{i|k} \leq u_{max} \Rightarrow \begin{cases} -U_k \leq U_{min} \\ U_k \leq U_{max} \end{cases}
\tag{6.6}
$$

where: $U_{min} = \begin{bmatrix} u_{min} & \cdots & u_{min} \end{bmatrix}^T \in \mathbb{R}^{n_u N_p}$
$U_{min} = \begin{bmatrix} u_{max} & \cdots & u_{max} \end{bmatrix}^T \in \mathbb{R}^{n_u N_p}$

Input rate constraint

$$\Delta u_{min} \leq \Delta u_{i|k} \leq \Delta u_{max} \Rightarrow \begin{cases} -U_k \leq -\Delta U_{min} - U_{k-1} \\ U_k \leq \Delta U_{max} + U_{k-1} \end{cases} \tag{6.7}$$

where: $\Delta U_{min} = \begin{bmatrix} \Delta u_{min} & \cdots & \Delta u_{min} \end{bmatrix}^T \in \mathbb{R}^{n_u N_p}$
$\Delta U_{min} = \begin{bmatrix} \Delta u_{max} & \cdots & \Delta u_{max} \end{bmatrix}^T \in \mathbb{R}^{n_u N_p}$
$\Delta u_{i|k} = u_{i|k} - u_{i|k-1} \rightarrow \Delta U_k = U_k - U_{k-1}$
$\Delta U_k = \begin{bmatrix} \Delta u_{0|k} & \cdots & \Delta u_{N_c-1|k} \end{bmatrix}^T$
$U_{k-1} = \begin{bmatrix} \Delta u_{0|k-1} & \cdots & \Delta u_{N_c-1|k-1} \end{bmatrix}^T$

6.1.3 QP formulation

$$\begin{aligned} \min_{U_k} \quad & U_k^T H U_k + 2 f^T U_k + g \\ \text{s.t.} \quad & A_{ineq} U_k \leq b_{ineq} \end{aligned} \tag{6.8}$$

where: $H = \left(\bar{R} + \bar{B}_d^T \bar{Q} \bar{B}_d \right)$
$f = \left(\bar{A}_d \xi_k + \bar{K}_d - ref_k \right)^T \bar{Q} \bar{B}_d$
$g = \left(\bar{A}_d \xi_k + \bar{K}_d - ref_k \right)^T \bar{Q} \left(\bar{A}_d \xi_k + \bar{K}_d - ref_k \right)$

$$A_{ineq} = \begin{bmatrix} -\bar{B}_d \\ \bar{B}_d \\ -I^{n_u N_p} \\ I^{n_u N_p} \\ -I^{n_u N_p} \\ I^{n_u N_p} \end{bmatrix} ; \quad b_{ineq} = \begin{bmatrix} -\bar{\xi}_{min} + \bar{A}_d \xi_k + \bar{K}_d \\ \bar{\xi}_{max} - \bar{A}_d \xi_k - \bar{K}_d \\ -U_{min} \\ U_{max} \\ -\Delta U_{min} - U_{k-1} \\ \Delta U_{max} + U_{k-1} \end{bmatrix}$$

6.2 High-level MPC

The high-level MPC is a classical linear MPC, using a point mass kinematic model for prediction. The optimization has a quadratic cost function which minimizes the difference of the state/output from a reference and the input, while the constraints include obstacle avoidance conditions beside the physical and design contraints.
The reference, here, is a sequence of given waypoints along the path, from the initial position to the final goal, and updated according the UGV position. In other words, once the UGV has reaced the waypoint set, the reference is updated with the next waypoint.

6.2.1 Point mass kinemtic model

The point mass kinematic model 3.2 is a LTI system that can be expressed in the state space form:

$$\dot{\xi}(t) = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} X \\ Y \\ V_x \\ V_y \end{bmatrix}}_{\xi(t)} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} \dot{V}_x \\ \dot{V}_y \end{bmatrix}}_{u(t)} = A\xi(t) + Bu(t) \tag{6.9}$$

where: $\xi(t) = \begin{bmatrix} X & Y & V_x & V_y \end{bmatrix}^T$ state
$u(t) = \begin{bmatrix} \dot{V}_x & \dot{V}_y \end{bmatrix}^T$ input

The continuos system has to be discretized according to the Euler approximation 10.7, as follow:

$$\xi_{k+1} = A_d\xi_k + B_d u_k \tag{6.10}$$

6.2.2 Obstacle Avoidance

The obstacle avoidance can be part of the optimization problem constraints, taking into account that the constraints has to be convex in order to keep the optimization convex. Exploiting the nature of the environment, the space can be though as a series of connected rectangular corridors, each of these is convex by definition.
Thus obstacle avoidance can be set as constraints over the position (state) of the UGV, where the minimum and maximum values are the border coordinates of each corridor. In other words, the UGV's state position is constrained to move within the borders of a corridor in which it is located. This means that it must be active only one corridor at time, which corrispond to a single set of obstacle avoidance contraints over the UGV's position state, to keep the proble convex.

Considering a generic corridor characterized by the following borders expressed through cartesian coordinates:

$$\begin{aligned} x_{s_j} &= [x_{s_{j,min}}, x_{s_{j,max}}] \\ y_{s_j} &= [y_{s_{j,min}}, y_{s_{j,max}}] \end{aligned} \tag{6.11}$$

it is active, if and only if it satisfies the following conditions:

- **Condition I**: the UGV position lies inside a range for which that constraints exists (obstacle's borders):

$$S_I = \begin{cases} x_{I_j} = x_{s_j} \Leftrightarrow y_{s_{j,min}} \leq y_k \leq y_{s_{j,max}} \\ y_{I_j} = y_{s_j} \Leftrightarrow x_{s_{j,min}} \leq x_k \leq x_{s_{j,max}} \end{cases} \tag{6.12}$$

- **Condition II**: the borders are within an activation range r_{act} with respect to the UGV position:

$$S_{II} = \begin{cases} x_{II_j} = x_{s_j} \Leftrightarrow \left|x_k - x_{s_j}\right| \leq r_{act} \\ y_{II_j} = y_{s_j} \Leftrightarrow \left|y_k - y_{s_j}\right| \leq r_{act} \end{cases} \tag{6.13}$$

The union of the two conditions above, returns the unique active corridor:

$$\begin{aligned} x_{act} = x_{s_j} &\Leftrightarrow \begin{cases} y_{s_{j,min}} \leq y_k \leq y_{s_{j,max}} \\ \left|x_k - x_{s_j}\right| \leq r_{act} \end{cases} \\ y_{act} = y_{s_j} &\Leftrightarrow \begin{cases} x_{s_{j,min}} \leq x_k \leq x_{s_{j,max}} \\ \left|y_k - y_{s_j}\right| \leq r_{act} \end{cases} \end{aligned} \tag{6.14}$$

Furthermore, it is possible to take into account a safety distance r_{safe} from the obstacle border:

$$\begin{aligned} X_c = [X_{min}, \quad X_{max}] = [x_{act_{min}} + r_{safe}, \quad x_{act_{max}} - r_{safe}] \\ Y_c = [Y_{min}, \quad Y_{max}] = [y_{act_{min}} + r_{safe}, \quad y_{act_{max}} - r_{safe}] \end{aligned} \tag{6.15}$$

Then it is possibile to set the constraints:

$$\begin{aligned} \underbrace{x_{act_{min}} + r_{safe}}_{X_{min}} \leq x_k \leq \underbrace{x_{act_{max}} - r_{safe}}_{X_{max}} \\ \underbrace{y_{act_{min}} + r_{safe}}_{Y_{min}} \leq y_k \leq \underbrace{y_{act_{max}} - r_{safe}}_{Y_{max}} \end{aligned} \tag{6.16}$$

6.2.3 Constraints

State constratins are related to the position and velocity expressed with respect to the ineratial frame O. About the position it ha been set the map boundaries, but these constrained are overwritten at each sample time by the constrains obstained in by the obstacle avoidance algorithm. Meanwhile for velocity it has been set the absolute value of $3m/s$ as a boundaries box.

$$\xi_c = [\xi_{min} \quad \xi_{max}] = \begin{bmatrix} X_{min} & X_{max} \\ Y_{min} & Y_{max} \\ V_{x,min} & V_{x,max} \\ V_{y,min} & V_{y,max} \end{bmatrix} = \begin{bmatrix} 0m & +20m \\ 0m & +10m \\ -3m/s & +3m/s \\ -3m/s & +3m/s \end{bmatrix} \tag{6.17}$$

Input constraints are related to the accellerations expressed with respect to the ineratial frame O, and it has been choosen the absolute value of $0.1m/s^2$ as a boundaries box.

$$u_c = [u_{min} \quad u_{max}] = \begin{bmatrix} \dot{V}_{x,min} & \dot{V}_{x,max} \\ \dot{V}_{y,min} & \dot{V}_{y,max} \end{bmatrix} = \begin{bmatrix} -0.1m/s & +0.1m/s \\ -0.1m/s & +0.1m/s \end{bmatrix} \tag{6.18}$$

6.2.4 Trajectory generation

The trajectory is generated appling the optimal solution $u*$, sample by sample, to the same system model used by the MPC, but with a smaller sample time:

$$x_{k+1} = A_d x_k + B_d u_k^* \tag{6.19}$$

where: $T_{sys} = \frac{T_s}{\alpha}$

6.3 Low-level Adaptive SL MPC

The low-level MPC used for control, requires at least an adaptive formulation, since the bycicle kinemtics model is nonlinear. For this reason has been chosen the Successive Linearization MPC, using as operative point the actual state and input of the system. This choice came from evaluating [14], where good performance are achieved with this kind of approach.
The optimization problem has a quadratic cost function, minimizing the error between the state of the system and the reference obtained from the high-level MPC.

6.3.1 Bicycle kinemtic model

The model used in prediction is the bicycle model 3.3, assuming only the frontal steering angle:

$$\begin{cases} \dot{X}(t) = v\cos(\psi + \beta) \\ \dot{Y}(t) = v\sin(\psi + \beta) \\ \dot{\psi}(t) = \frac{v\cos(\beta)}{l_f + l_r}\tan(\delta_f) = \frac{V}{l_r}\sin(\beta) \end{cases} \tag{6.20}$$

where: X, Y : coordinates with respect to the inertial frame
V_x, V_y : longitudinal and lateral velocity with respect to the inertial frame
v : velocity in body frame
θ_f, θ_r : front and rear steering angles
ψ : heading with respect to the inertial frame
$\beta = \arctan\left(\frac{l_f\tan(\delta_r) + l_r\tan(\delta_f)}{l_f + l_r}\right)$: slip angle

Linearization

The nonlinear system has to be linearized around the operative point $p = (\xi_0, u_0)$. According to 10.2, the system can be approximated through the Taylor expansion:

$$\dot{\xi}(t) = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A(p)} \xi(t) + \underbrace{\begin{bmatrix} -v\sin(\psi_0+\beta_0) \\ v\cos(\psi_0+\beta_0) \\ v\cos(\beta_0) \end{bmatrix}}_{B(p)} u(t) + \underbrace{\begin{bmatrix} v\cos(\psi_0+\beta_0) \\ v\sin(\psi_0+\beta_0) \\ v\cos(\beta_0) \end{bmatrix}}_{K(p)} \tag{6.21}$$

where: $\xi(t) = \begin{bmatrix} X & Y & \psi \end{bmatrix}^T \in \mathbb{R}^{n_x}$ state
$u(t) = \Delta\beta \in \mathbb{R}^{n_u}$ input

obtaining the following LPV system:

$$\dot{\xi}(t) = A(p)\xi(t) + B(p)u(t) + K(p) \tag{6.22}$$

Discretization

For prediction, the system is discretized according to the Euler approximation 10.7, as follow:

$$\xi_{k+1} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{A_k(p)} \xi_k + \underbrace{\begin{bmatrix} -V\sin(\psi_0+\beta_0)T_s \\ V\cos(\psi_0+\beta_0)T_s \\ V\cos(\beta_0)T_s \end{bmatrix}}_{B_k(p)} u_k + \underbrace{\begin{bmatrix} -V\sin(\psi_0+\beta_0)T_s \\ V\cos(\psi_0+\beta_0)T_s \\ V\cos(\beta_0)T_s \end{bmatrix}}_{K_d(p)} \tag{6.23}$$

obtaining:

$$\xi_{k+1} = A_d(p)\xi_k + B_d(p)u_k + K_d(p) \tag{6.24}$$

6.3.2 Constraints

The state constraints is related to the pose of the UGV, and for the position, are set the map boundaries, as for the High-Level MPC, but in this case these ones are not updated.

$$\xi_c = \begin{bmatrix} \xi_{min} & \xi_{max} \end{bmatrix} = \begin{bmatrix} X_{min} & X_{max} \\ Y_{min} & Y_{max} \\ \psi_{min} & \psi_{max} \end{bmatrix} = \begin{bmatrix} 0m & +20m \\ 0m & +10m \\ -\pi^\circ & +\pi^\circ \end{bmatrix} \tag{6.25}$$

Input constraint and input rate constraint are respectively the slip angle limits and slip angle rate limits.

$$u_c = \begin{bmatrix} u_{min} & u_{max} \end{bmatrix} = \begin{bmatrix} \beta_{min} & \beta_{max} \end{bmatrix} = \begin{bmatrix} -0.5rad/s & +0.5rad/s \end{bmatrix} \tag{6.26}$$

$$\Delta u_c = \begin{bmatrix} \Delta u_{min} & \Delta u_{max} \end{bmatrix} = \begin{bmatrix} \Delta\beta_{min} & \Delta\beta_{max} \end{bmatrix} = \begin{bmatrix} -0.1^\circ/s & +0.1^\circ/s \end{bmatrix} \tag{6.27}$$

Chapter 7

Adaptive MPC in curvilinear frame

The second design developed was inspired by [17], in which a MPC formulation is used with a model with respect to a curvilinaer reference frame, called Frenet frame, expressed in curvilinaer coordinates. In this way, due to the nature of the environment, the obstacles can be direcly are addressed directly into contraints, in particular, as the maximum distance with respect a defined path centreline. For this reason the reference is porvided as a complete description of a geometric path. In our case we decided to use again a Successive Linearization formulation of the MPC, but with a bycicle dynamics model. Furthermore, it has been taken into account in the cost function, the minimization of the input rate, in order to have an impact on the steering rate, which is a performance parameter.

7.1 Bicycle dynamics in a curvilinear reference

From the previous sections, a dynamic model of the bicycle in a curvilinear reference can be implemented under the following assumptions:

- Small sideslip angles ($\alpha \leq \pm 5°$)
- Front steering wheel command ($\delta_r = 0$)
- Rear traction ($F_{lr} = T, F_{lf} = 0$)
- The forces are multiplied by a factor of 2 in order to take into account the physics of 4 wheels

the dynamics in 3.10 can be semplified as:

$$\begin{aligned} \dot{v}_x &= \frac{1}{m}(-F_{sf}\sin(\delta_f) + T) + \omega_z v_y \\ \dot{v}_y &= \frac{1}{m}(F_{sf}\cos(\delta_f) + F_{sr}) - \omega_z v_x \\ \dot{\omega}_z &= \frac{1}{I_z}(l_f F_{sf}\cos(\delta_f) - l_r F_{sr}) \end{aligned} \tag{7.1}$$

while the sideslip angle, from 3.13, under the assumption of small angles, is:

$$\begin{aligned} \alpha_f &= \delta_f - \arctan\left(\frac{v_y + l_f\omega_z}{v_x}\right) \simeq \delta_f - \frac{v_y + l_f\omega_z}{v_x} \\ \alpha_r &= -\arctan\left(\frac{v_y - l_r\omega_z}{v_x}\right) \simeq -\frac{v_y - l_r\omega_z}{v_x} \end{aligned} \tag{7.2}$$

The tire's forces are then:

$$\begin{aligned} F_{sf} &= 2C_f\alpha_f \simeq 2C_f\left(\delta_f - \frac{v_y + l_f\omega_z}{v_x}\right) \\ F_{sr} &= 2C_r\alpha_r \simeq 2C_r\left(-\frac{v_y - l_r\omega_z}{v_x}\right) \end{aligned} \tag{7.3}$$

While the link between the dynamics from the cartesian coordinate into the curvilinear cooridnate from 2.18 is:

$$\begin{aligned} \dot{s}(t) &= \frac{v_x\cos e_\psi - v_y\sin e_\psi}{1 - c_c(s)e_y} \\ \dot{e}_y(t) &= v_x\sin e_\psi + v_y\cos e_\psi \\ \dot{e}_\psi(t) &= \omega_z - \frac{v_x\cos e_\psi - v_y\sin e_\psi}{1 - c_c(s)e_y}c_c(s) \end{aligned} \tag{7.4}$$

Putting toghether 7.1 7.3 7.4 it is obtained the complete dynamics:

$$\begin{aligned} \dot{v}_x(t) &= a + \frac{2}{m}\left(-C_f\left(\delta_f - \frac{v_y + l_f\omega_z}{v_x}\right)\sin(\delta_f)\right) + v_y\omega_z \\ \dot{v}_y(t) &= \frac{2}{m}\left(C_f\left(\delta_f - \frac{v_y + l_f\omega_z}{v_x}\right)\cos(\delta_f) + C_r\left(-\frac{v_y - l_r\omega_z}{v_x}\right)\right) - v_x\omega_z \\ \dot{\omega}_z(t) &= \frac{2}{I_z}\left(l_f C_f\left(\delta_f - \frac{v_y + l_f\omega_z}{v_x}\right)\cos(\delta_f) - l_r C_r\left(-\frac{v_y - l_r\omega_z}{v_x}\right)\right) \\ \dot{s}(t) &= \frac{v_x\cos e_\psi - v_y\sin e_\psi}{1 - c_c(s)e_y} \\ \dot{e}_y(t) &= v_x\sin e_\psi + v_y\cos e_\psi \\ \dot{e}_\psi(t) &= \omega_z - \frac{v_x\cos e_\psi - v_y\sin e_\psi}{1 - c_c(s)e_y}c_c(s) \end{aligned} \tag{7.5}$$

where: $\xi = \begin{bmatrix} v_x & v_y & \omega_z & s & e_y & e_\psi \end{bmatrix}^T$
$u = \begin{bmatrix} \delta_f & a \end{bmatrix}^T$

7.1.1 Linearization

The nonlinear system has to be linearized around the operative point $p = (\xi_0, u_0)$. According to 10.2, the system can be approximated through the Taylor expansion:

$$\frac{d}{dt}(\delta\xi(t)) \approx \frac{\partial f}{\partial \xi}\delta\xi(t) + \frac{\partial f}{\partial u}\delta u(t) \Rightarrow \delta\dot{\xi}(t) \approx A(p)\delta\xi(t) + B(p)\delta u(t) \tag{7.6}$$

where: $\delta\xi(t) = \xi(t) - \xi_0 = \begin{bmatrix} \delta v_x & \delta v_y & \delta\omega_z & \delta s & \delta e_y & \delta e_\psi \end{bmatrix}^T$
$\delta u(t) = u(t) - u_0 = u(t) = \begin{bmatrix} \delta_f & a \end{bmatrix}^T$

then:

$$\delta\dot{\xi}(t) = A(p)\delta\xi(t) + B(p)u(t) \tag{7.7}$$

7.1.2 Discretization

For prediction, the system is discretized according to the Euler approximation 10.7, as follow:

$$\delta\xi_{k+1} = A_d\delta\xi_k + B_d u_k \tag{7.8}$$

7.2 Prediction

At k-th sampled, the predicted state is:

$$\delta\xi_{i+1|k} = A_d(p)\delta\xi_{i|k} + B_d(p)u_{i|k}, \quad \delta\xi_{0|k} = \delta\xi_k \tag{7.9}$$

State prediction:

$$\delta\bar{\xi}_k = \bar{A}_d(p)\delta\xi_k + \bar{B}_d(p)U_k \tag{7.10}$$

where: $\bar{\delta\xi}_k = \begin{bmatrix} \delta\xi_{1|k} \\ \vdots \\ \delta\xi_{N_p|k} \end{bmatrix} \in \mathbb{R}^{n_x N_p}; \quad U_k = \begin{bmatrix} u_{0|k} \\ \vdots \\ u_{N_c-1|k} \end{bmatrix} \in \mathbb{R}^{n_u N_c};$

$$\bar{A}_d(p) = \begin{bmatrix} A_d(p) \\ A_d^2(p) \\ \vdots \\ A_d^{N_p}(p) \end{bmatrix} \in \mathbb{R}^{n_x N_p, n_x};$$

$$\bar{B}_d(p) = \begin{bmatrix} B_d(p) & 0^{n_x,n_u} & \cdots & 0^{n_x,n_u} \\ A_d(p)B_d(p) & B_d(p) & \cdots & 0^{n_x,n_u} \\ \vdots & \vdots & \ddots & \vdots \\ A_d^{N_c}(p)B_d(p) & A_d^{N_c-1}(p)B_d(p) & \cdots & B_d(p) \\ \vdots & \vdots & \ddots & \vdots \\ A_d^{N_p-1}(p)B_d(p) & A_d^{N_p-2}(p)B_d(p) & \cdots & A_d^{N_p-N_c}(p)B_d(p) \end{bmatrix} \in \mathbb{R}^{n_x N_p, n_u N_c}$$

7.3 Optimization Problem

In this case, our objective function consists in minimizing the state variation $\delta\xi_{i|k}$, the input $u_{i|k}$ and also the input rate $\Delta u_{i|k}$, each one as quadratic loss. This choise came from the need to penalize big variation in the steering angle command, in order to avoid possible discontinuities. As in the first approach, the constraints are the prediction model constraint, the state, input and input rate.

$$\begin{aligned} \min_{U_k} \quad & \sum_{i=1}^{N_p} \left\|\delta\xi_{i|k}\right\|_Q^2 + \sum_{i=0}^{N_c-1} \left(\left\|u_{i|k}\right\|_R^2 + \left\|\Delta u_{i|k}\right\|_S^2\right) \\ \text{s.t.} \quad & \delta\xi_{i+1|k} = A_d(p)\delta\xi_{i|k} + B_d(p)u_{i|k}, \quad i \in [1,\ \dots,\ N_p] \\ & \delta\xi_{0|k} = \delta\xi_k \\ & \delta\xi_{min} \leq \delta\xi_{i|k} \leq \delta\xi_{max}, \quad i \in [1,\ \dots,\ N_p] \\ & u_{min} \leqslant u_{i|k} \leqslant u_{max}, \quad i \in [0,\ \dots,\ N_c - 1] \\ & \Delta u_{min} \leqslant \Delta u_{i|k} \leqslant \Delta u_{max}, \quad i \in [0,\ \dots,\ N_c - 1] \end{aligned} \tag{7.11}$$

7.3.1 Objective function

The objective functione consists in minimizing the square root of the state variation, the input and the input rate.

$$\begin{aligned} J(\xi_k) &= \sum_{i=1}^{N_p} \left\|\delta\xi_{i|k}\right\|_Q^2 + \sum_{i=0}^{N_c-1} \left(\left\|u_{i|k}\right\|_R^2 + \left\|\Delta u_{i|k}\right\|_S^2\right) \\ &= \left\|\bar{\delta\xi}_k\right\|_{\bar{Q}}^2 + \|U_k\|_{\bar{R}}^2 + \|\Delta U_k\|_{\bar{S}}^2 \\ &= U_k^T\left(\bar{B}_d^T\bar{Q}\bar{B}_d + \bar{R} + \bar{S}\right)U_k + 2\left(\left(\bar{A}_d\delta\xi_k\right)^T\bar{Q}\bar{B}_d - U_{k-1}\bar{S}\right)U_k \\ &+ \left(\bar{A}_d\delta\xi_k\right)^T\bar{Q}\left(\bar{A}_d\delta\xi_k\right) + (U_{k-1})^T\bar{S}U_{k-1} \end{aligned} \tag{7.12}$$

where: $\bar{Q} = \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q \end{bmatrix} \in \mathbb{R}^{n_xN_p,n_xN_p}$

$$\bar{R} = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R \end{bmatrix} \in \mathbb{R}^{n_uN_c,n_uN_c}$$

$$\bar{S} = \begin{bmatrix} S & 0 & \cdots & 0 \\ 0 & S & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S \end{bmatrix} \in \mathbb{R}^{n_uN_c,n_uN_c}$$

7.3.2 Constraints

State constraints The velocity along the body longitudinal axis is contrained in a range of $\pm 0.2m/s$ with respect to the initial velocity, while for lateral velocity and angular velocity variation it has been taken values congruent with reality. Instead, about the curvilinear we expected a variation between a minimum of $0m$ and a maximum of $0.2m$ with respect the actual curvilinera value, meanwhile for contraints on lateral variation it has been set the lateral track boundaries which represent the obstacle avoidance contraints, and the angular deviation a freedom of $\pm 30°$ with respect to the centerline direction.

$$\delta\xi_c = [\delta\xi_{min} \quad \delta\xi_{max}] = \begin{bmatrix} \delta v_{x,min} & \delta v_{x,max} \\ \delta v_{y,min} & \delta v_{y,max} \\ \delta\omega_{z,min} & \delta\omega_{z,max} \\ \delta s_{min} & \delta s_{max} \\ \delta e_{y,min} & \delta e_{y,max} \\ \delta e_{\psi,min} & \delta e_{\psi,max} \end{bmatrix} = \begin{bmatrix} -0.2m/s & +0.2m/s \\ -0.5m/s & +0.5m/s \\ -50°/s & +50°/s \\ -0m & +0.2m \\ -1m & +1m \\ -30° & +30° \end{bmatrix} \tag{7.13}$$

$$\delta\xi_{min} \leq \delta\xi_{i|k} \leq \delta\xi_{max} \Rightarrow \begin{cases} -\bar{B}_d(p)U_k \leq -\delta\bar{\xi}_{min} + \bar{A}_d(p)\delta\xi_k \\ \bar{B}_d(p)U_k \leq \delta\bar{\xi}_{max} - \bar{A}_d(p)\delta\xi_k \end{cases} \tag{7.14}$$

where: $\delta\bar{\xi}_{min} = \begin{bmatrix}\delta\xi_{min} & \cdots & \delta\xi_{min}\end{bmatrix}^T \in \mathbb{R}^{n_x N_p}$
$\delta\bar{\xi}_{max} = \begin{bmatrix}\delta\xi_{max} & \cdots & \delta\xi_{max}\end{bmatrix}^T \in \mathbb{R}^{n_x N_p}$

Input constraints The steering angle is contrained to a lower range with respect the real UGV, since the model used is under the assumption of a single steering wheel, so we decided to take a common value for this kind of vehicles. About the accellerations, it has been choosen a standard range of $\pm 0.1 m/s^2$.

$$u_c = \begin{bmatrix}u_{min} & u_{max}\end{bmatrix} = \begin{bmatrix}\delta_{f,min} & \delta_{f,max} \\ a_{min} & a_{max}\end{bmatrix} = \begin{bmatrix}-30° & +30° \\ -0.1m/s^2 & +0.1m/s^2\end{bmatrix} \tag{7.15}$$

$$u_{min} \leq u_{i|k} \leq u_{max} \Rightarrow \begin{cases} -U_k \leq U_{min} \\ U_k \leq U_{max} \end{cases} \tag{7.16}$$

where: $U_{min} = \begin{bmatrix}u_{min} & \cdots & u_{min}\end{bmatrix}^T \in \mathbb{R}^{n_u N_p}$
$U_{max} = \begin{bmatrix}u_{max} & \cdots & u_{max}\end{bmatrix}^T \in \mathbb{R}^{n_u N_p}$

7.4 QP formulation

$$\begin{aligned} \min_{U_k} \quad & U_k^T H(p) U_k + 2 f^T(p) U_k + g(p) \\ \text{s.t.} \quad & A_{ineq}(p) U_k \leq b_{ineq}(p) \end{aligned} \tag{7.17}$$

where: $H(p) = \left(\bar{B}_d^T \bar{Q} \bar{B}_d + \bar{R} + \bar{S}\right)$
$f(p) = \left(\left(\bar{A}_d \delta\xi_k\right)^T \bar{Q} \bar{B}_d - U_{k-1}\bar{S}\right)$
$g(p) = \left(\bar{A}_d \delta\xi_k\right)^T \bar{Q}\left(\bar{A}_d \delta\xi_k\right) + (U_{k-1})^T \bar{S} U_{k-1}$
$$A_{ineq}(p) = \begin{bmatrix} -\bar{B}_d \\ \bar{B}_d \\ -I^{n_u N_p} \\ I^{n_u N_p} \end{bmatrix}, \quad b_{ineq}(p) = \begin{bmatrix} -\delta\bar{\xi}_{min} + \bar{A}_d \delta\xi_k \\ \delta\bar{\xi}_{max} - \bar{A}_d \delta\xi_k \\ -\delta U_{min} \\ \delta U_{max} \end{bmatrix}$$

Part III

Conclusions

Chapter 8

First design

8.1 Implementation

The first design has been implemented in Simulink, as shown in figure 8.1 and 8.3, using ”quadprog” function of the Optimization Toolbox. In these pictures are highlited the sampling time of each block:

- black box: continuos time domain
- red box: $T_s = 0.001s$
- green box: $T_s = 0.01s$
- blue box: $T_s = 0.1s$

8.1.1 Guidance

The guidance system is made up by the ”Low-Level MPC” which embeds the ”Obstacle Avoidance” system, then the ”Reference Update” system and the ”Plant”.

Reference Update It takes care of set and hold the proper waypoint till the UGV reached it, and then it is set the next waypoint on the list. The waypoints have to be choosen in order to avoidance potentially infeasibility in the problem, in other words that it must guarantee the local optimality.

Track The track is the serpentine in fig. 8.2, where are visible the obstacles. This map is provided to the MPC block of obstacle avoidance, as a series of rectangular spaces, where the UGV is free to move.

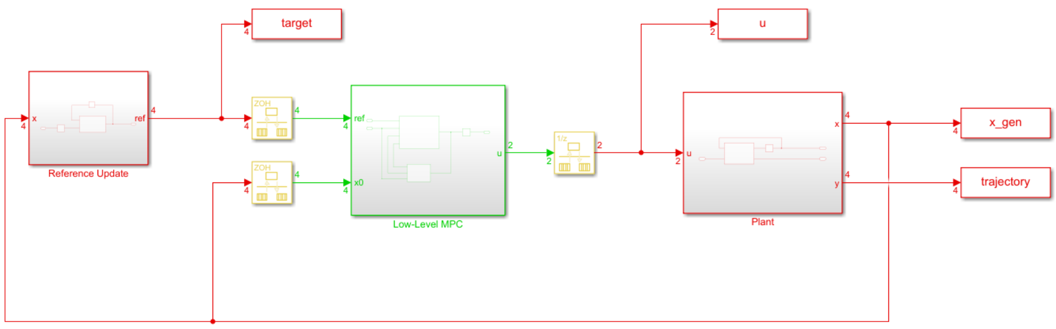


Figure 8.1: High-Level MPC Simulink scheme

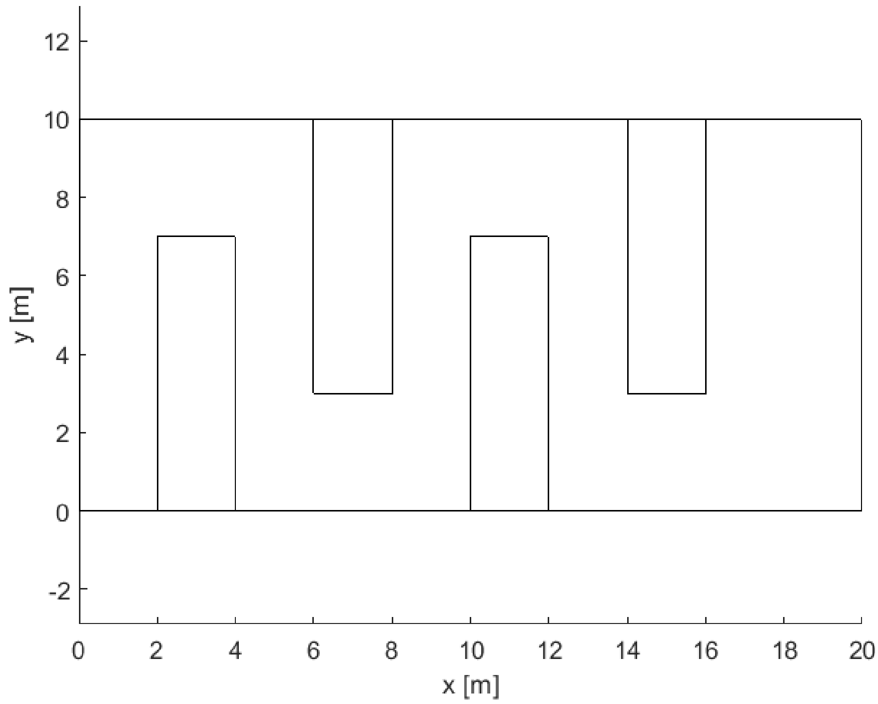


Figure 8.2: First Track

Low-Level MPC The set waypoint is used as fixed reference for the Low-Level MPC, which, according to the constraints given by the "Obstacle Avoidance" system, it compute the optimal input. The obstacle avoidance system takes as input the full description of the map and the pose of the UGV, in order to determine which constraints is active at that instant.

Plant The optimal input is used by the plant to generate the the corresponding trajectory. This trajectory can be computed online or more efficiently offline if there is no need to recomputing the trajectory.

8.1.2 Control

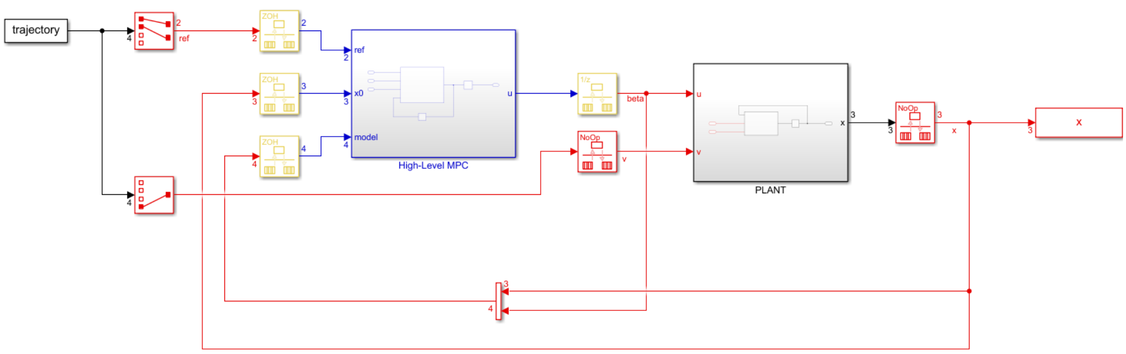


Figure 8.3: Low-Level MPC Simulink scheme

The control system is simply made up by the ”High-Level MPC” which controls the real system. The High-Level MPC takes as refernce the trajectory determined by the guidance system and the current state and input used as operative point by the linearization.

8.2 Simulations

The two systems share each other common information, as the initial state, which is:

State	Value	Description
X	$1m$	xcooridinate in inertial frame
Y	$1m$	ycooridinate in inertial frame
ψ	$90°$	orientation in inertial frame
V_x	$0m/s$	velocity along longitudinal body axis
V_y	$0m/s$	velocity along lateral body axis

Table 8.1: First design: initial state

while the initial tuning of the guidance MPC and control MPC are listed respectively in table 8.2 and table 8.3:

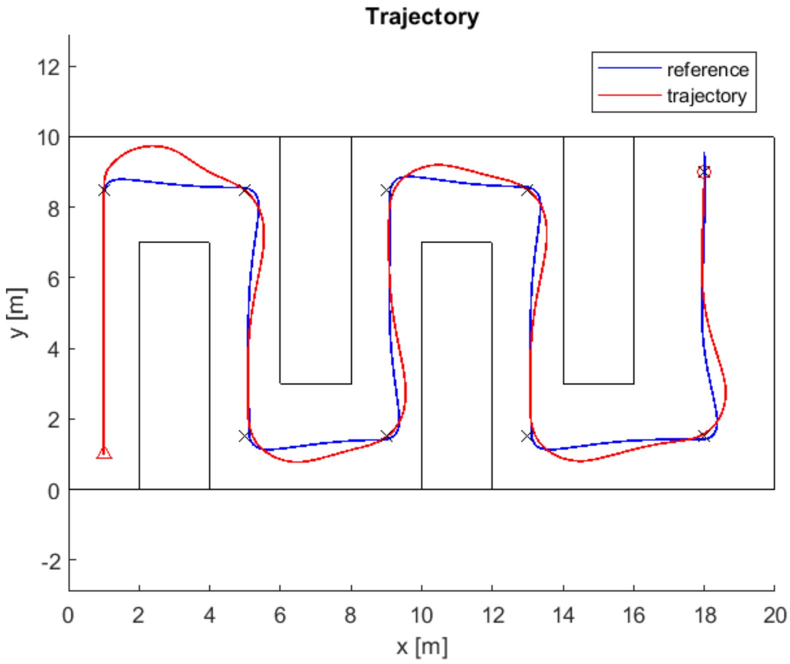
Parameter	Value	Description
N_p	30	prediction horizon
N_c	30	prediction control
Q	$diag([1\ 1\ 1\ 1])$	state matrix weight
R	$diag([10\ 10])$	input matrix weight

Table 8.2: First design: guidance MPC tuning

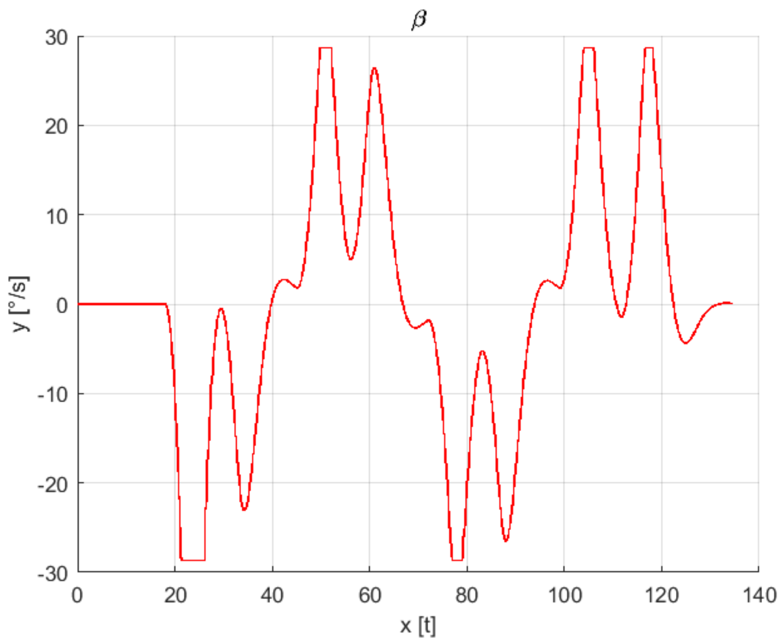
Parameter	Value	Description
N_p	25	prediction horizon
N_c	10	prediction control
Q	$diag([1\ 1])$	state matrix weight
R	$diag([1])$	input matrix weight

Table 8.3: First design: control MPC tuning

With this initial configuration we obtain the results shown in fig. 8.4, in which the planned trajectory generated (in blue), by the point mass model is sharp, making it hard to tightly track it in the control part.



(a)



(b)

Figure 8.4: Results: a) $trajectory$; b) $input$;

8.3 Conclusion

As mentioned before, the trajectory generated appears to be too sharp, due to the point mass model that generates it, which does not have constraints over the heading. Two possible solutions can ba considered: the first consists in implementing a more complex model for the planning MPC, the second instead, in increasing the number of waypoints to make the path generated smoother. Moreover another restriction in the considered system has been in the restricted possibility to tune the MPC, which easily fell in infeasibility conditions.
The advantage of this approach is that it is possible to compute offline the feasible trajectory, while online performing only the control part, reducing drammatically the computation time required and then the amount of resources.

Chapter 9

Second design

9.1 Implementation

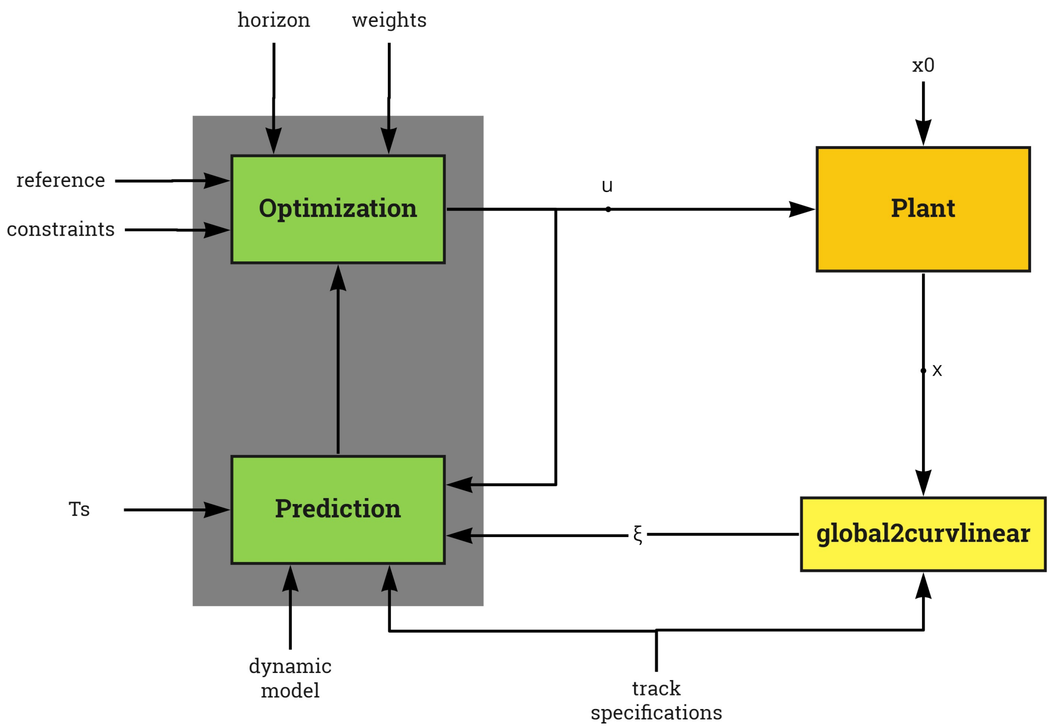


Figure 9.1: MPC in curvilinear frame block scheme

Differently to the first approach, the second has been developed completely on Matlab, with the the scheme shown in figure 9.1, using, as previously done, ”quadprog” function of the Optimization Toolbox.

This system is made up by the ”MPC” (Prediction and Optimization) which embeds the obstacle avoidance directly into constraints, the ”Plant” and the ”gloval2curvilinear” function.

Track The difference with respect the first approach it is that here it has to be fully defined a geometric path. A track can be simple created by specifyng only few elements:

- vector of intial position and orientation of the track
- matrix in which each row defines length and radius of each segment of the track

Then a function fully defines the geometric path, obtaining a matrix, which entries correspond to the position x, y, the orientation ψ, the curvilinear abscissa s, the segment length l and the curvature c of each segment.
The track defined for our tests is a serpentine shown in figure 9.2. With respect to the first track it is sligtly bigger, since it has been required curves with a slightly wider radius of curvature.

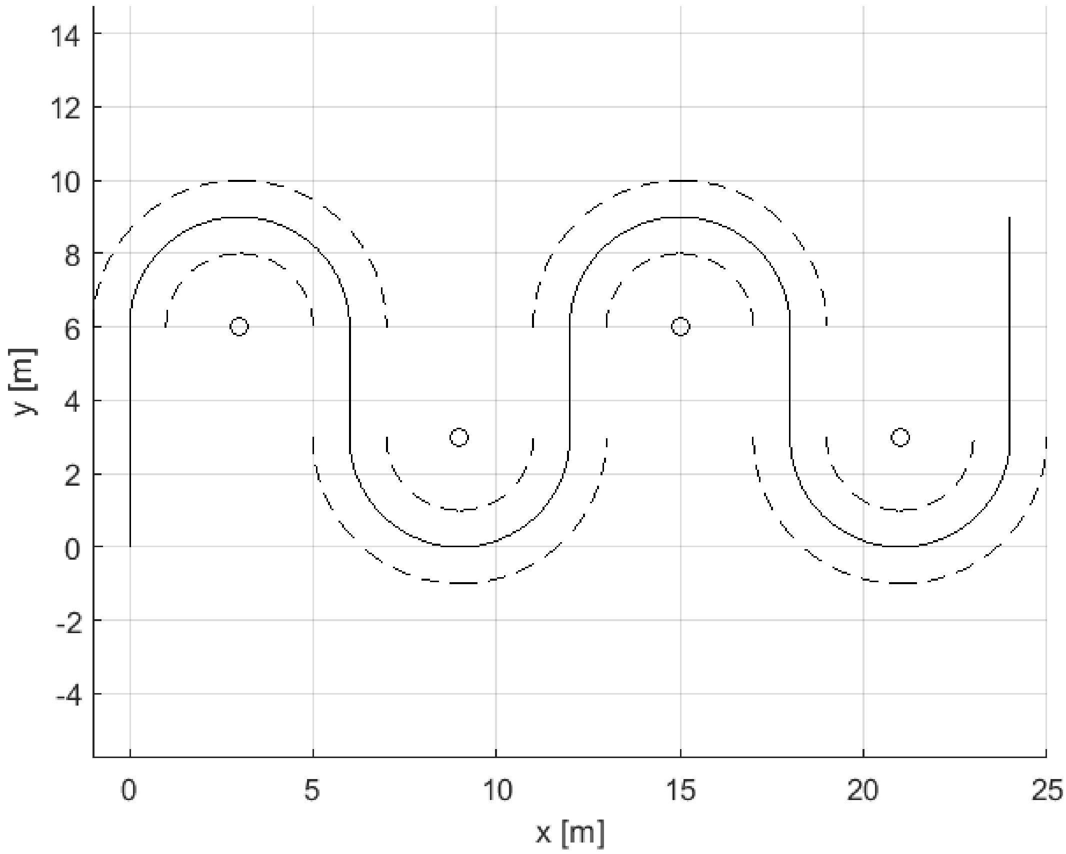


Figure 9.2: Second Track

MPC The MPC uses the prediction model along with the track definition, to make a predicion over a horizon, which is used by the optimization problem to find the optimal input to apply to the system.

Plant In this specific case the system to control is the following nonlinear system obtained by the sets 3.3 and 3.10, which state is:

$$\xi_{sys} = \begin{bmatrix} v_x & v_y & \omega_z & X & Y & \psi \end{bmatrix}^T \tag{9.1}$$

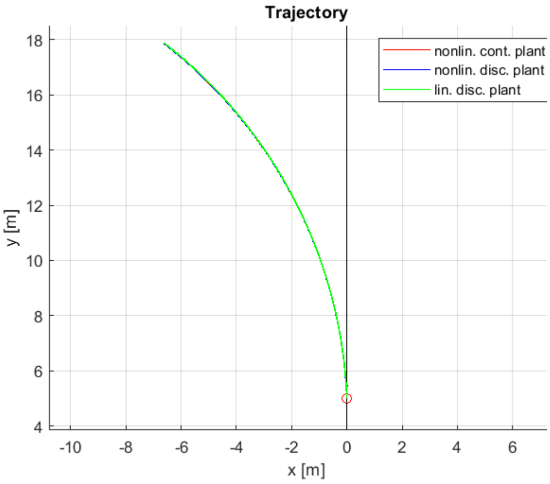
global2curvilinear In general, since the system gives information about the state expressed in another coordinate system, it is required a function that convert that system in the curvilinear one.

9.2 System evolution validation

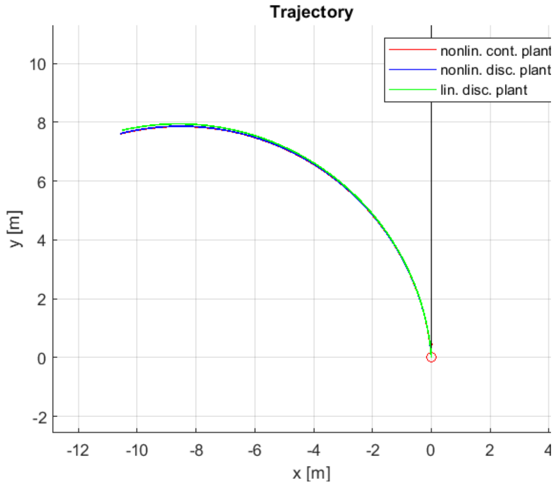
Before using a specific system to control, it has been taken into account three model systems:

- nonlinear continuous bycicle dynamics
- nonlinear discrete bycicle dynamics
- linearized discrete bycicle dynamics

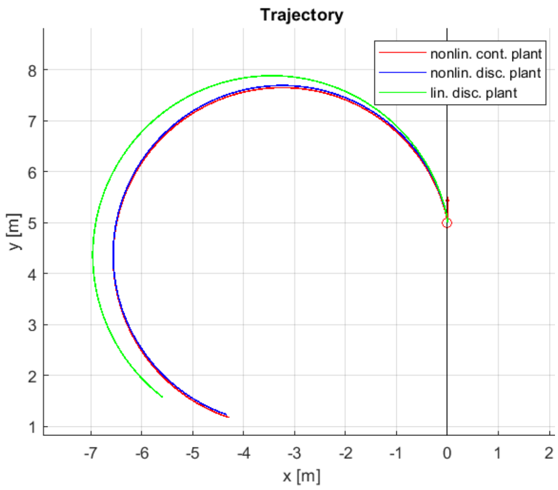
Firstly it has been evaluate the forced response of the systems subjeced to differents steering angles as shown in figure 9.3, where it can be clearly seen that the two nonlinear models coincides perfectly, which means that the discretization accordin Euler approximation is quite good for out purpose, menawhile the linearized model differs from the other two at high steering angles. These results have given us the deviation of the linearized model used for prediction from the nonlinear model from which it was derived.



(a)



(b)



(c)

Figure 9.3: System evolution with initial $v_x = 1,5m/s$: a) $\delta = 5°$; b) $\delta = 10°$; c) $\delta = 25°$.

Then the three models are tested in a simulation with the MPC control, as shown in figure 9.4 giving to us another validation about the behaviour of the systems.

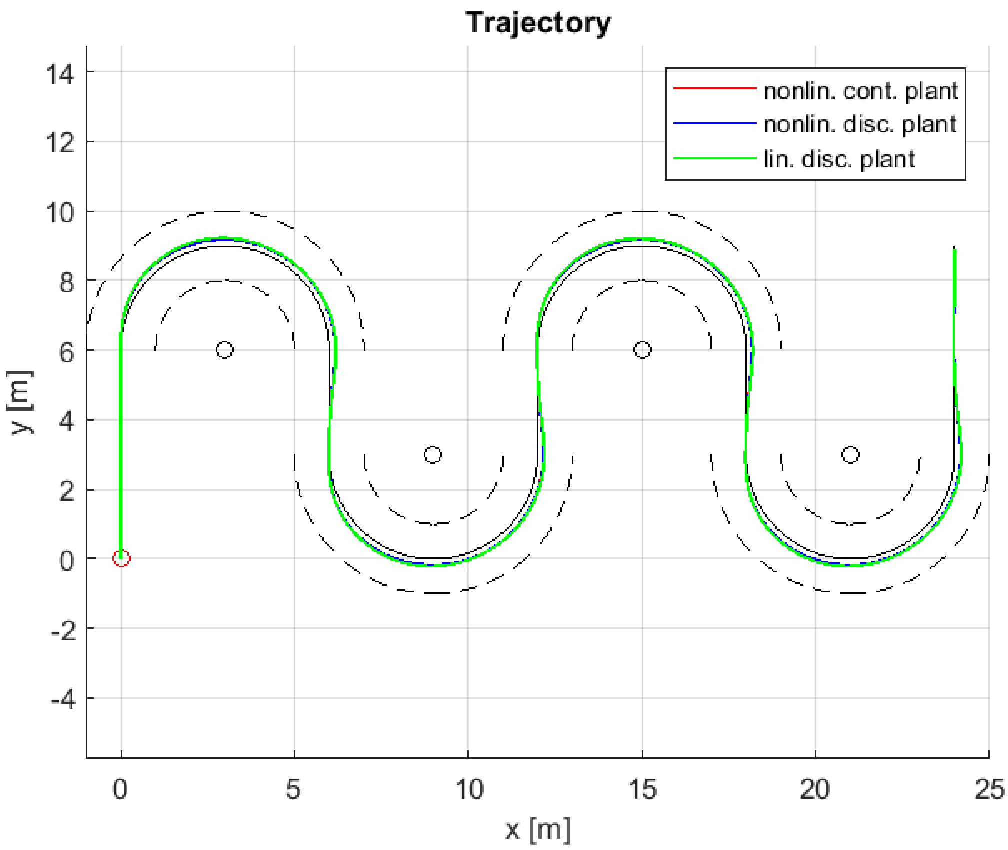


Figure 9.4: Trajectory of the three systems

9.3 Simulations

The following results are obtained from simulations, choosing as plant the nonlinear continuos bycicle dynamic model, with initial state:

State	Value	Description
v_x	$1,5m/s$	velocity along longitudinal body axis
v_y	$0m/s$	velocity along lateral body axis
ω_z	$0°/s$	angular velocity
X	$0m$	position along x inertial frame
Y	$0m$	position along y inertial frame
ψ	$90°$	orientation in inertial frame

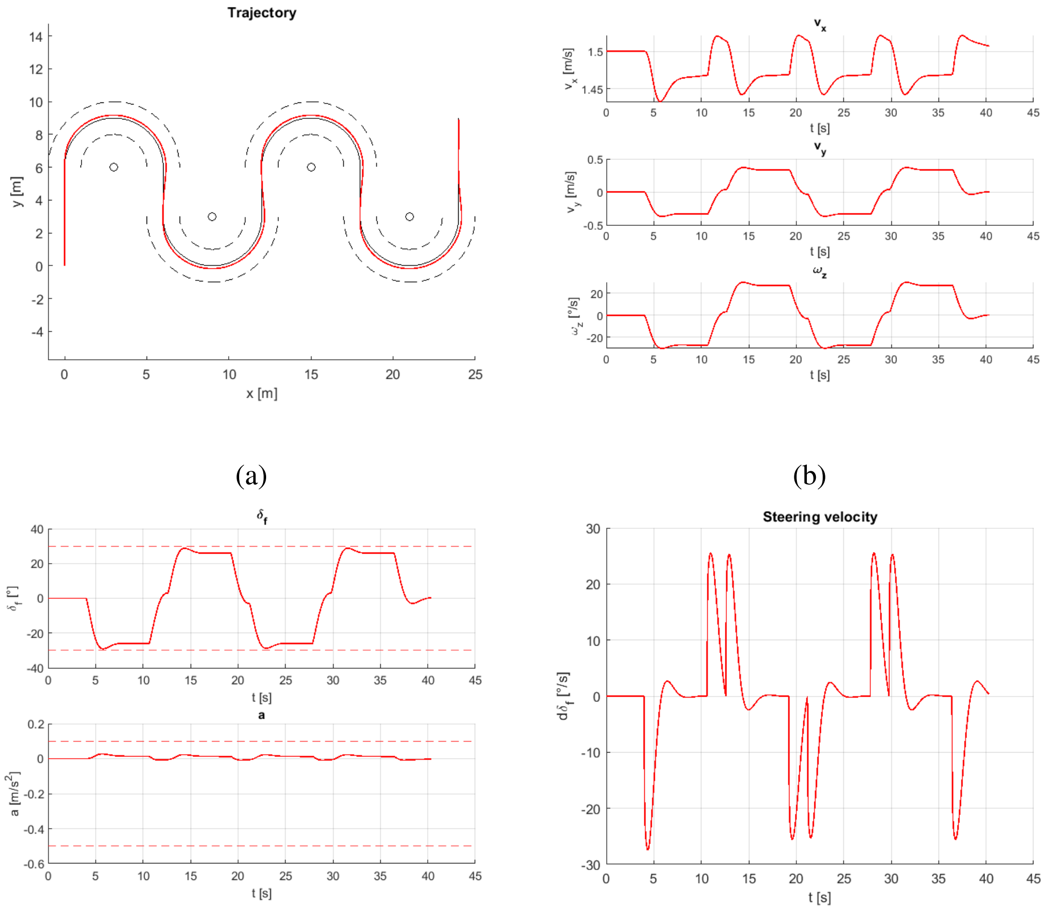
Table 9.1: Initial state

and initial tuning of MPC parameters:

Parameter	Value	Description
N_p	20	prediction horizon
Q	$diag([1\ 1\ 1\ 1\ 10\ 1])$	state matrix weight
R	$diag([1\ 1])$	input matrix weight
S	$diag([1\ 1])$	input rate matrix weight

Table 9.2: MPC tuning parameters

With this configuration, it has been obtain a well defined trajectory, as shown in fig. 9.5, showing quite good results.



(a) (b)

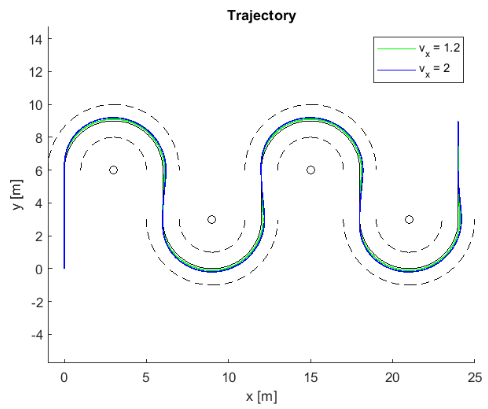
(c) (d)

Figure 9.5: Second design: initial conditions

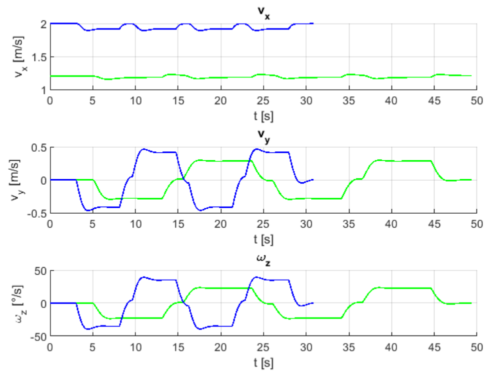
9.3.1 Initial conditions variation

Before trying to look for better performance, we had validated the MPC for working in different initial conditions, trying to obtain a range of operational states.

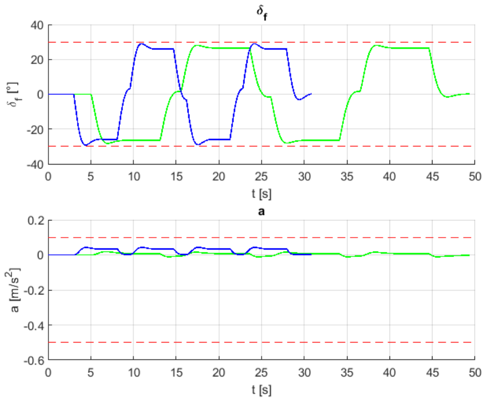
At first it has been taken the extremes of the speed range we consider. Results shown in fig. 9.6 that, as expected the UGV complete the path in smaller time with higher starting velocities. It has been noticed an increasing of the steering rate for the highest velocity of $v_x = 2.0m/s$ with respect the other two that present almost the same behaviour.



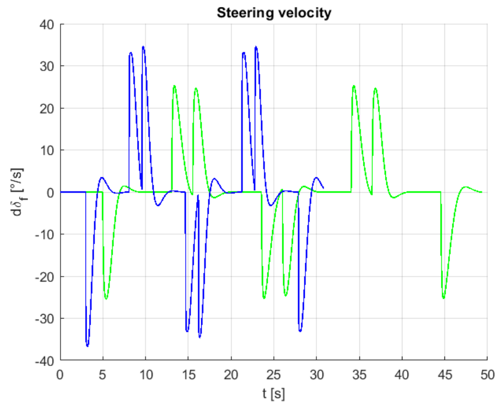
(a)



(b)



(c)



(d)

Figure 9.6: MPC with different initial velocities

Then it has been considered to start from three different initial poses, as shown in fig. 9.7. From these simulation it has been possible to see that there are no restricton on the starting position, as long as it lies within the track boundaries, while for the starting orientation it is obtained a range of $\pm 50°$ with respect to the centerline. For higher deviations, the MPC cannot find a feasible trajectory for that initial condition.

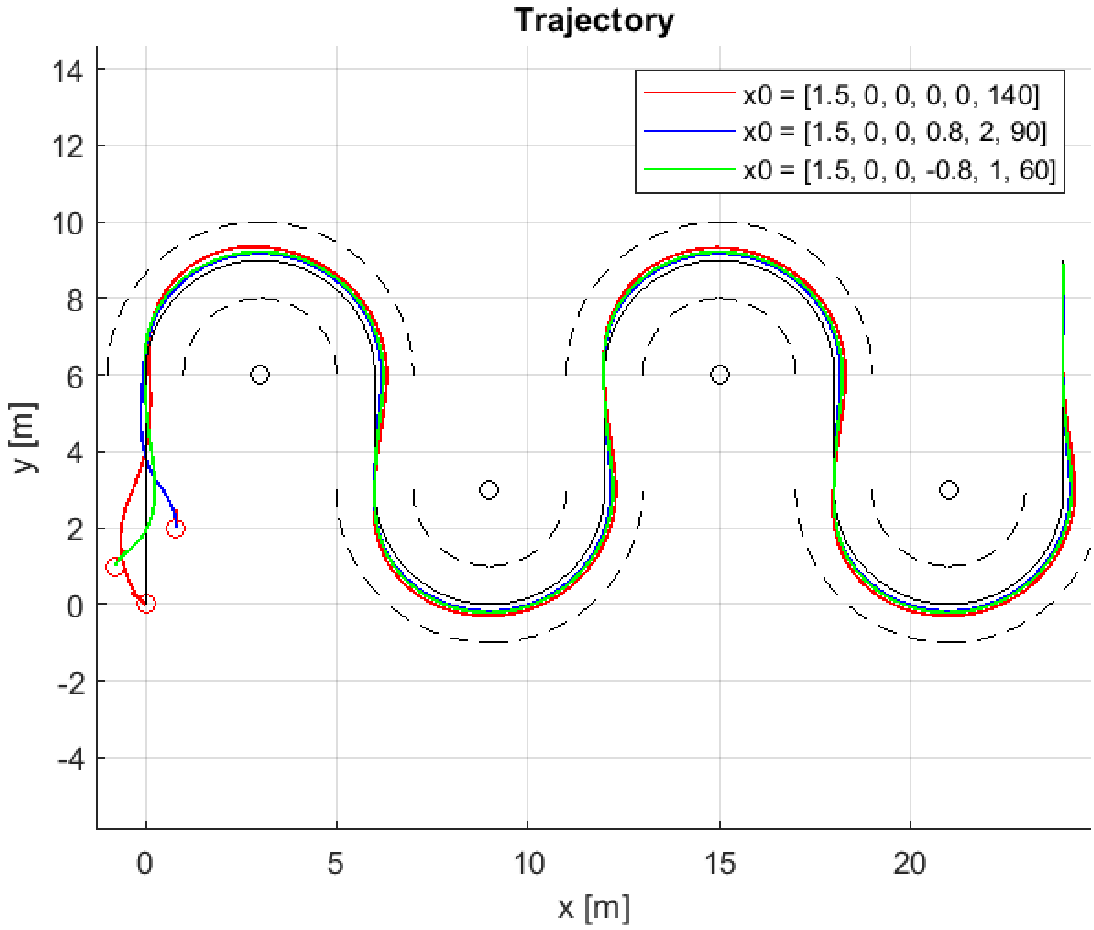


Figure 9.7: MPC with different initial pose

9.3.2 MPC tuning

Firstly it has been changed the prediction horizon N_p in a range from 10 to 50. In general increasing the prediction horizon, the MPC can obtain a better sequence of optimal input to apply in terms of energy minimization, but at the same time it might run into infeasibility, since the constraints must be guaranteed for a longer time.
In fact the horizon at $N_p = 50$ has the trajectory that tightly follow the center-line altought the input never reach saturation. In the opposity way the horizon at $N_p = 10$ has a more relaxed trajectory, but reaching the saturation of the steering angle.

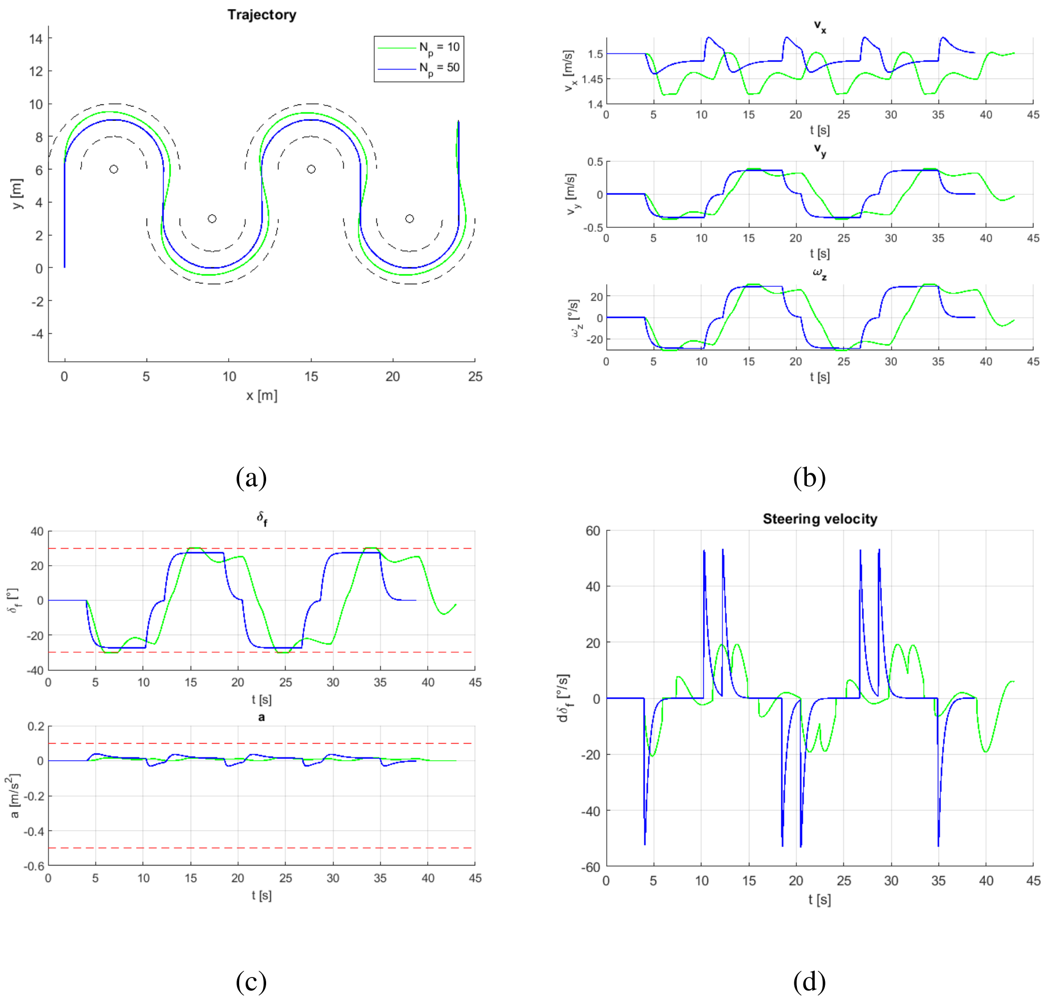


Figure 9.8: MPC with different initial pose

Then we proceed to tune the weighting matrices. In general increasing the value of a diagonal element of a matrix, means penalizing the term related to it more than the other ones, then it can be said that it is important the relative weight and not the absolute value of them.

The aim of this tuning is to obtain a smoother trajectory as well as a smooth behaviour of the steering angle. A first attempt is decreasing the weight related to the deviation from the center-line e_y. As shown in fig. 9.9, we obtain lower values of the steering angle velocities and in general a more uniform behaviour on the steering angle itself.

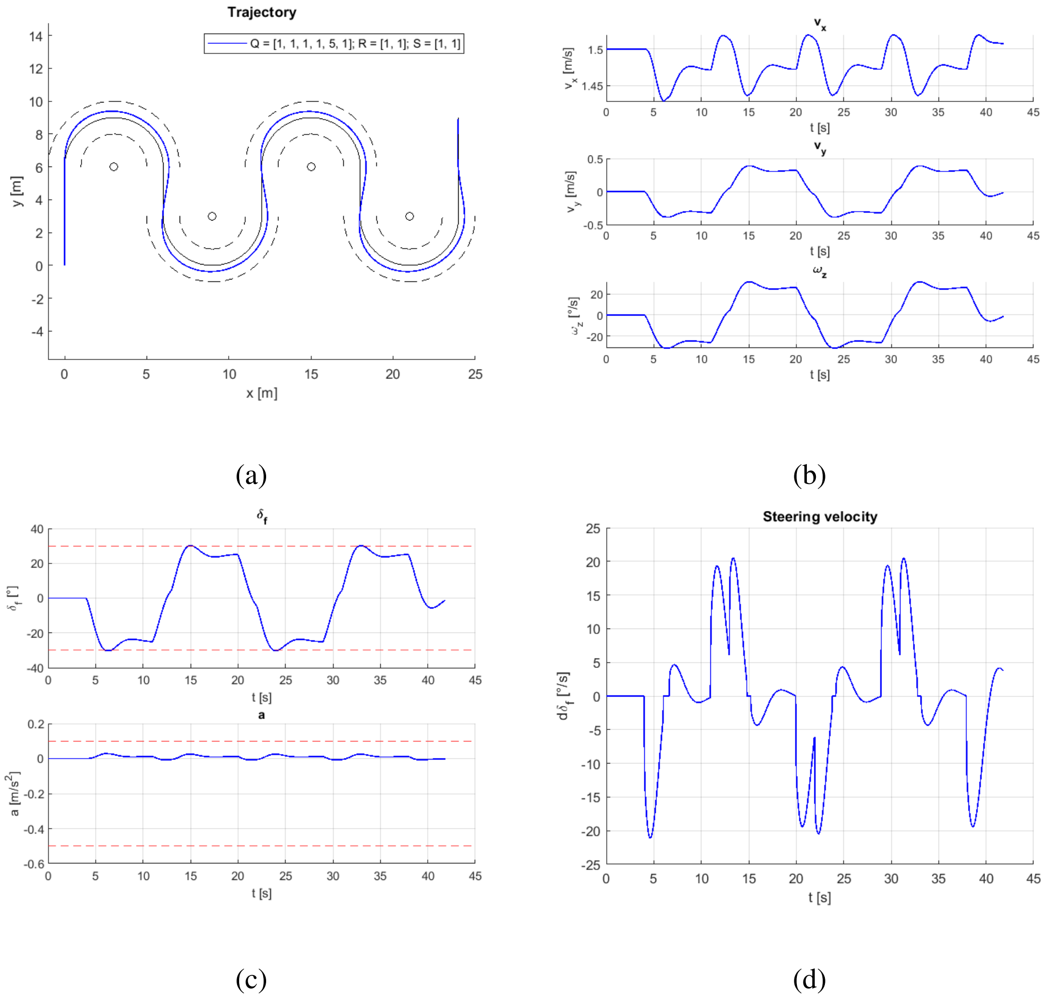
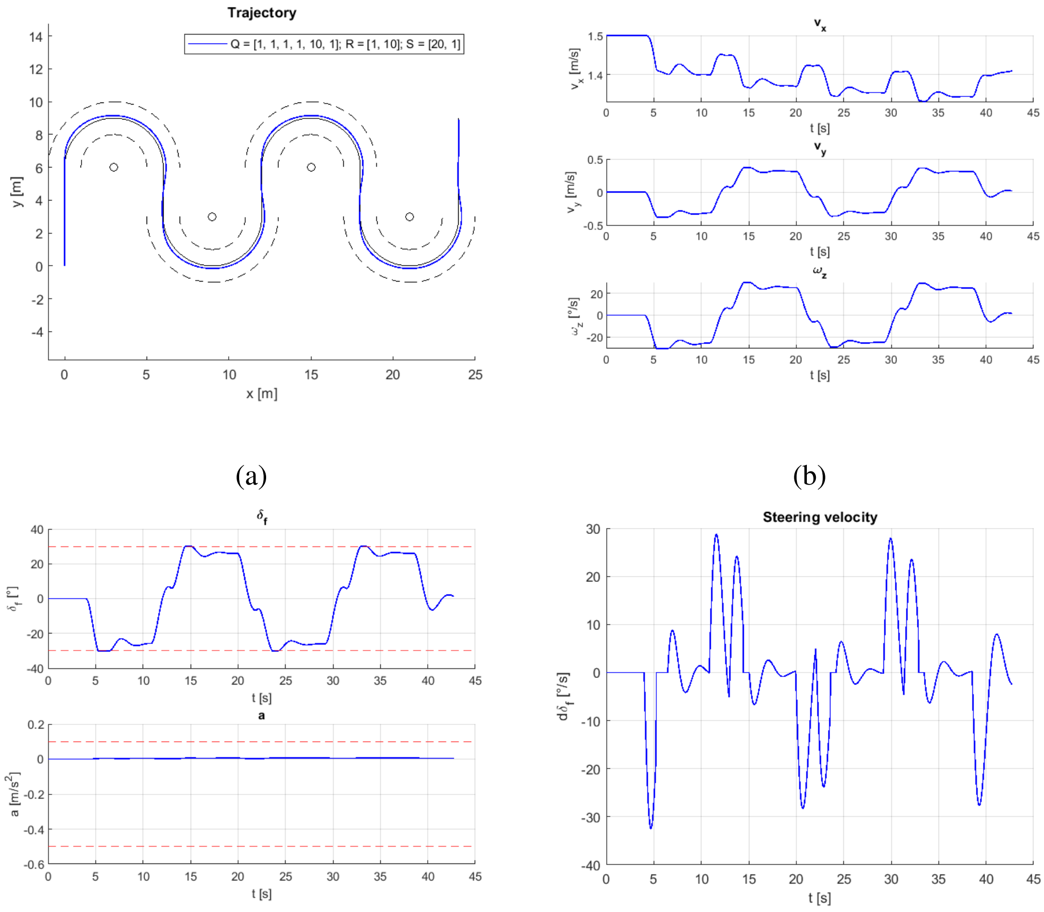


Figure 9.9: MPC weight matrices tuning I

Then we tried to increase the weight on the steering angle and also on the steering angle rate, as shown in fig. 9.10.
In both cases we can notice no significant improvement on the steering angle command, but shown at the end, better performance with respect the first design, with a more complex model, with trajectories that are smoother and more coherent with the given operative environment.



(a) (b)

(c) (d)

Figure 9.10: MPC weight matrices tuning II

9.4 Conclusion

This last approach show bettere performances with respect to the first one. A feasible trajectory is obtained for a wide range of conditions and MPC tuning, despite the tight constraints due to the small curvature radius of the path. Increasing the angle of steering to the our UGV specifications it is posible to obtain feasible trajectories for narrower tracks. Moreover, it is possible to widely tune the MPC parameters in order to have different performance as it is possible to see from simulations. It also whorty to notice the impact ever the steering rate given by introducing it in the cost function of the optimization problem, even if not present directly as state or input of the system, Finally, it is possible to state that the velocity profiles appear to be, in each simulation, smooth and without discontinuities, which are essential requirements.

Part IV

Appendix

Chapter 10

Linearization and Discretization

10.1 Linearization

The linearization allows to make linear a nonlinear system:

$$\dot{x}(t) = f(x(t), u(t)) \tag{10.1}$$

around an operative/equilibrium point (x_{op}, u_{op}). The system can be approximated through the Taylor expansion, stopped to the first derivative:

$$\frac{d}{dt}(\delta x(t)) \approx \frac{\partial f}{\partial x}\delta x(t) + \frac{\partial f}{\partial u}\delta u(t) \tag{10.2}$$

where: $x(t) = x_{op} + \delta x(t)$
$u(t) = u_{op} + \delta u(t)$
which can be replaced:

$$\begin{aligned} \delta\dot{x}(t) &\approx \dot{x}_{op} + \frac{\partial f}{\partial x}\delta x(t) + \frac{\partial f}{\partial u}\delta u(t) \\ &= \dot{x}_{op} + \begin{bmatrix} \frac{\partial f_1}{\partial x} \\ \vdots \\ \frac{\partial f_{n_x}}{\partial x} \end{bmatrix} \delta x(t) + \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \vdots \\ \frac{\partial f_{n_x}}{\partial u} \end{bmatrix} \delta u(t) \end{aligned} \tag{10.3}$$

where: $\frac{\partial f_i}{\partial x} = \begin{bmatrix} \frac{\partial f_i}{\partial x_1} & \cdots & \frac{\partial f_i}{\partial x_{n_x}} \end{bmatrix}$
$\frac{\partial f_i}{\partial u} = \begin{bmatrix} \frac{\partial f_i}{\partial u_1} & \cdots & \frac{\partial f_i}{\partial u_{n_u}} \end{bmatrix}$
The system can be linearized as:

$$\delta\dot{x}(t) = A\delta x(t) + B\delta u(t) \tag{10.4}$$

where: $A = J_f(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_{n_x}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial x_1} & \cdots & \frac{\partial f_{n_x}}{\partial x_{n_x}} \end{bmatrix} \in \mathbb{R}^{n_x, n_x}$

$B = J_f(u) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_{n_u}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial u_1} & \cdots & \frac{\partial f_{n_x}}{\partial u_{n_u}} \end{bmatrix} \in \mathbb{R}^{n_x, n_u}$

Expanding the variables:

$$\begin{aligned} &\frac{d}{dt}(x(t) - x_{op}) = A(x(t) - x_{op}) + B(u(t) - u_{op}) \\ &\Rightarrow \dot{x}(t) = A(p)x(t) + B(p)u(t) + K(p) \end{aligned} \tag{10.5}$$

where: $K(p) = \dot{x}_{op}(t) - A(p)x_{op}(t) - B(p)u_{op}(t)$

10.2 Discretization

Since controllers in general works in discrete time domain, then a continuous state-space continuous model has to be discretized.

10.2.1 Exact integration

$$\begin{cases} A_d = \exp(A_c T_s) \\ B_d = A_c^{-1}(\exp(A_c T_s) - I)B_c \end{cases} \tag{10.6}$$

10.2.2 Euler approximation

$$\begin{cases} A_d = I + A_c T_s \\ B_d = B_c T_s \end{cases} \tag{10.7}$$

Bibliography

[1] LaValle, S.M. ”Rapidly-exploring random trees: a new tool for path planning.” *Tech. Rep (TR 98-11), Computer Science Dept*, Iowa State University (1998)

[2] LaValle, S.M., Kuffner, J.J., ”Randomized kinodynamic planning.” *Proceedings of IEEE International Conference on Robotics and Automation, vol. 20, no. 5, p. 378*, Iowa State University (1999)

[3] ”Agriculture 4.0 – The Future Of Farming Technology” *World Government Summit*, 2018

[4] ”GNC NASA report” *https://www.nasa.gov/sites/default/files/atoms/files/gnc.pdf*

[5] C. Goerzen, Z. Kong, B. Mettler, ”A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance”, 2010

[6] Cerone Vito, ”Modern design of control systems”, *Politecnico di Torino*

[7] Giuseppe Carlo Calafore, ”Convex optimization and engineering applications”, *Politecnico di Torino*

[8] Massimo Canale, ”Digital control technologies and architectures”, *Politecnico di Torino*

[9] J. Nilsson, Y. Gao, A. Carvalho, F. Borrelli, ”Manoeuvre generation and control for automated highway driving” *Proceedings of the 19th World Congress*, 2014

[10] P. Falcone, F. Borrelli, H. E. Tseng, ”A hierarchical model predictive control framework” *2008 American Control Conference*, 2008

[11] T. Schouwenaars, B. DeMoor, E. Feron, J. How, ”Mixed Integer Programming for Multi-Vehicle Path Plannning” *European Control COnforence (ECC)*, 2001

[12] R. Rajamani, *”Vehicle Dynamics and Control”*, Chapter 2, 2012.

[13] A. Micaelli and C. Samson, *Trajectory tracking for unicycle-type and two-steering-wheels mobile robots*, 1993.

[14] C. K. Law, D. Dalal, S. Shearrow, ”Robust Model Predictive Control for Autonomous Vehicle Self-Driving Cars”, 2014

[15] Yiqi Gao. *”Model Predictive Control for Autonomous and Semiautonomous Vehicles”*, PhD Thesis, 2014.

[16] F. Borrelli, ”Learning Model Predictive Control”, in *”Proceedings 56th Conference on Decision and Control”*, 2017.

[17] U. Rosolia, F. Borrelli, ”Learning How to Autonomously Race a Car - A Predictive Control Approach ”, 2020.