

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

SCHOOL OF ENGINEERING

Department of
Electrical, Electronic and Information Engineering
“Guglielmo Marconi”
DEI

Master’s Degree in Automation Engineering

Master Thesis
in
Optimal Control

**Control of an autonomous aerodynamic
airshield for training Olympic 100m
sprint athletes**

Candidate:

Giulia Cutini

Supervisor:

Prof. Giuseppe Notarstefano

Co-supervisor:

Prof. Melanie Zeilinger
Dr. Andrea Carron
Ing. Lorenzo Sforni

Academic Year
2022–2023

Session
IV

Abstract

Contents

Introduction	4
Motivations	4
Literature	5
Contributions	7
1 Theoretical Background	9
1.1 Linear Quadratic Regulator	9
1.2 Linear Model Predictive Control	12
1.3 Offset-free tracking MPC and disturbance observer	14
1.3.1 Observer design	16
2 Control design	17
2.1 Model	17
2.2 Linear approximation of the system dynamics	19
2.3 LQR design	21
2.3.1 Catch-up maneuver	21
2.4 MPC design	23
2.5 Offset-free MPC and disturbance observer design	26
3 Simulation results and Hardware-in-the-loop validation	29
3.1 Simulation results	29
3.2 System structure and hardware description	29
3.3 Perception	31
3.3.1 Controller design changes for hardware integration	34
LQR	34
Bibliography	38

Introduction

Motivations

In contemporary times, the integration of advanced technologies has become an integral facet of sport, ushering in a new era of innovation and performance enhancement [1]. In sports characterized by high-speed competition, where victories are often determined by mere thousandths of a second, the advancements in technology have revolutionized performance across various disciplines.

The development of streamlined swimsuits with advanced materials reduces drag and enhances buoyancy, leading to faster times in competitive swimming events. Ski manufacturers utilize materials like carbon fiber and titanium in ski construction, resulting in lighter yet more stable skis that offer better control and responsiveness on the slopes. The incorporation of carbon fiber also into athletic footwear has not only reduced weight but also enhanced energy return and propulsion, while concurrently mitigating the risk of injury.

The exploitation of technology for improving performances is not only used for competitions but also during the training phase. In the track and field context, several training techniques have been used for the enhancement of running speed. Among several possibilities overspeed training involves performing exercises or movements at a velocity higher than what the athlete can achieve through voluntary effort. Many methods for experiencing supramaximal velocities during trainings have been proposed: the usage of assistance mechanisms such as bungee cords or sleds is analyzed in [2] and is found to be potentially dangerous for sprinters, requiring body contact with an external tool. Performing sprints on a downward slope allows to accelerate beyond typical maximal speed due to assistance provided by gravity force [3].

In the context of gaining a training edge in sprint events the concept of aerodynamic drag resistance assumes a paramount importance. Accordingly sprinters

meticulously refine their body position, posture, and apparel to minimize aerodynamic drag. For this reason CONI (Italian National Olympic Committee) Institute of Sports Science presented in 2021 an aerodynamic shield to drastically reduce the resistance to forward movement during sprinters' training [4]. This shield allows athletes to run in the slipstream behind a car pulling the shield, experiencing speeds higher than those of the competition but with the same power output.

Advancement in automated driving technology has created opportunities for many fields and autonomous vehicles (AVs) have become increasingly prevalent. Given their versatility and potential to enhance efficiency and convenience, it's natural to consider their application in the realm of sports. Similarly to the user-cooperative robots, designed to stay in direct contact with humans and assist them in several situations, an autonomous vehicle has been designed for driving the aerodynamic airshield. Furthermore, intelligent control algorithms have shown huge advantages in autonomous system control, taking into account factors such as control error, bound constraints on system actuators, disturbance rejection and safety guarantees. Additionally, control algorithms can be designed to continuously monitor and analyze data from various sensors in real-time, enabling them to detect and respond to potential hazards or deviations from safety protocols more effectively than human operators.

This thesis is the result of an abroad internship in ETH (Eidgenössische Technische Hochschule) Zürich, with the purpose of designing a proper controller to regulate an autonomous go-kart pulling a plexiglass airshield, according to the runner behaviour.

Literature

Numerous control techniques have been used in the last decades to enhance the efficiency of AVs in several different contexts. Longitudinal control of automated vehicles has received attention since the 1960s, and possibly even earlier. [5] gives an historical review of advanced vehicle control systems and longitudinal control of automated vehicles. The control of vehicle acceleration to achieve a desired speed profile, essential keypoint of longitudinal control, is the main purpose of this thesis.

The initial and simplest approach adopted to control the go-kart, which must maintain a constant reference position and velocity, relative to the runner behind it, involved a cascade of two Proportional-Integral-Derivative (PID) controllers. The

first PID controller regulates the kart's position and generates a reference velocity for the second controller, a velocity PID.

While PID controllers are widely used in industrial applications due to their straightforward implementation, they were not the optimal choice for this application. The main challenges of this control scheme includes difficulty in parameter tuning, in handling multi-variable processing and in addressing the initial phase of motion, where the runner's acceleration exceeds the kart's maximum capability, together with the impossibility of predicting the future motion of vehicle.

The task for which the controller has to be designed is quite similar to adaptive cruise control (ACC), where the goal is to maintain a safe distance from the vehicle ahead while also adjusting the speed to match changing traffic conditions. Similarly, in this scenario, the controller aims to ensure that the go-kart maintains a consistent position and velocity relative to the runner, analogous to how an adaptive cruise control adjusts the speed of a vehicle to maintain a safe following distance.

Thanks to the capability of optimal controllers to cover multiple objectives and to predict future vehicle behaviour, Linear Quadratic Regulator (LQR) and Model Predictive Control (MPC) have been widely used for vehicle control in interacting contexts. In [6] distance and relative velocity between preceding and controlled vehicles are used as states to design an LQR controller that gives acceleration of the controlled vehicle as input, in such a way to minimize a proper cost function. Model Predictive Control is also largely used thanks to its capability of controlling a multi-variable process while satisfying a set of state and input constraints. According to [7] the essential elements of this controller are a prediction model that allows future predictions, the definition of an objective function reflecting the desired system behaviour, and constraints on both state and input. In the framework of MPC for adaptive cruise control, in [8] the same state and input variables than in [6] are used but constraints can be also incorporated. Differently, in [9] the controlled vehicle velocity is added as a third state and the acceleration of the preceding vehicle is included in the prediction model, regarded as an unknown external disturbance.

The MPC approach is strongly model-based in the sense that uses a model of the system to produce a prediction of system behaviour. The presence of uncertainties on this model may lead to unsatisfying results of the control task. For this reason [10], [11] show offset-free MPC formulations able to achieve output tracking of reference signals despite the presence of Model-Plant Mismatch (MPM). For this purpose an augmented predictive model added with an artificial constant disturbance that represents modeling uncertainties, is used. To estimate both state

ad disturbance from output measurements a disturbance observer must be properly designed and added in conjunction with feedback control law. Accordingly to [12] by appropriately including a disturbance estimate in a control law, MPM effects can be approximately removed in steady-state and offset-free tracking of constant references can be achieved.

Contributions

The main contribution of this thesis lies in the analysis and implementation of linear controllers for a pioneering autonomous driving go-kart, which is utilized for towing the airshield during the training of Olympic athletes participating in the 100m event. Specifically, the thesis proposes and compares two different control schemes in terms of performances and control architecture.

Initially, a non linear model that accurately captures the dynamics of the kart with the airshield attached is identified. Later a good linear approximation is found and used as basis for the design of the predictive controllers. The first step in control development involves the design of a Linear Quadratic Regulator (LQR) with a gain scheduling approach, where Q and R parameters vary depending on the motion phase. This gain scheduling approach addresses the challenges associated with the different maximum accelerations of the kart and of the runner in the first phase of the motion. Subsequently, a Model Predictive Control (MPC) is designed with the purpose of incorporate information on the runner future expected behaviour into the linear prediction model. Furthermore, since MPC is highly model based a disturbance observer and Offset-free MPC has been implemented with the purpose of increasing the accuracy of the predictive model and to obtain a zero-offset in tracking peace-wise affine (PWA) reference trajectories.

The depicted workflow highlights how the control scheme architecture has been simplified starting from the two control loops governing the cascade of PID controllers. The gain scheduling LQR, which switches between two different control gains depending on the motion phase, just makes use of two different regulators. Ultimately, this thesis culminates in a single MPC controller capable of governing the system under all possible operating conditions, integrating information about the future evolution of the runner's velocity profile into the prediction model to further enhance tracking performance.

Both controllers have been developed and tested in simulation using Python and

on the real system using ROS2.

Organization

This section provides to the reader a guide to the organization of the thesis.

In Chapter 1 a theoretical background and an overview of the fundamental methodologies used in the control design are presented.

Chapter 2 is focused on discussing the problem set-up, the formulation of the optimal control techniques developed to accomplish the desired task, in particular the Linear Quadratic Regulator (LQR) and the Model Predictive Control (MPC), in the specific context of this application.

Chapter 3 gives a description of the system mechanical structure, hardware and sensors. Then it shows the numerical results of the controllers application obtained both in Python simulation and on the real hardware. A comparison of the performance in different scenarios is also presented. This demonstrate the effectiveness of the proposed methodologies.

Chapter 1

Theoretical Background

In this chapter the aim is to provide the necessary theoretical background for thoroughly understand the design process and the main peculiarities and limitations of the designed controllers: the Linear Quadratic Regulator (LQR) and the Model Predictive Control (MPC).

Optimal control theory provides a mathematical framework for addressing control theory problems by optimizing control laws with respect to given cost functions. The primary goal is to find a control strategy for a given system while penalizing deviations of the system's states or inputs from desired values. This approach enables achieving optimal system regulation, maximizing control effectiveness, and minimizing the adverse effects of variations in operating conditions or external disturbances.

Such theoretical analysis is essential for fully comprehending the features and performance of the LQR and MPC controllers discussed in Chapter 2 within the specific context of the considered application.

1.1 Linear Quadratic Regulator

In 1960, Kalman pioneered with the field of linear-quadratic feedback control, laying the groundwork for what later became known as the Linear Quadratic Regulator (LQR) [13] in the continuous-time context. The LQR is a feedback control algorithm designed to stabilize and optimize the performance in case where the dynamical system can be described by linear equations and the cost function is quadratic in terms of the states and inputs.

Assuming the process to be controlled to have a good discrete-time LTI descrip-

tion like the following state-space form:

$$\begin{aligned} x^+ &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1.1}$$

where $x \in \mathbb{R}^n$ is the actual system state, $u \in \mathbb{R}^m$ is the system input and $y \in \mathbb{R}^p$ is the measured output, all considered at a certain discrete time instant t . The state at the subsequent time instant $t + 1$ is referred to as x^+ .

$A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are the state transition and input distribution matrices, respectively. In the context of this thesis is possible to assume the full state information available, thus $C \in \mathbb{R}^{p \times n}$ is the identity.

The primary goal of the control task is to achieve optimal regulation of the system's state towards the origin. The notion of optimality, in this context, is framed within a quadratic objective function. The LQR problem, therefore, can be viewed as a multi-objective optimization task, where the goal is to simultaneously minimize competing objectives. For example, minimizing the deviation of the system's states from desired values may require higher control effort, and vice versa. In the context of multi-objective optimization, the challenge lies in finding a balance between these conflicting objectives and this always involves making trade-offs to achieve a satisfactory solution.

The Infinite Horizon Linear Quadratic Regulator problem is defined as follows: find the optimal control input $u_t, \forall t \in [0, \infty)$ that makes the following quadratic criteria as small as possible

$$J(\mathbf{u}, \mathbf{u}) = \sum_{t=0}^{+\infty} x_t^T Q x_t + u_t^T R u_t \tag{1.2}$$

where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are symmetric positive-definite weight matrices.

The vector \mathbf{u} contains the input trajectory, i.e. $\mathbf{u} = \{u_0, u_1, \dots\}$ and \mathbf{x} is the one containing the state trajectory, i.e $\mathbf{x} = \{x_1, x_2, \dots\}$. The cost function in 1.2 can be seen as the mathematical formulation of the trade-off between the two objectives to be minimized. Specifically the term $x_t^T Q x_t$ penalizes deviations of the system's states from desired values, while $u_t^T R u_t$ penalizes excessive control effort. The weights of the Q and R , typically chosen as diagonal matrices, can be adjusted to find solutions that strike an appropriate balance between minimizing state deviations and control effort.

Assuming the pair (A, B) to be controllable and the pair (A, C) with $Q = C^T C$ to be observable, holds:

- there exists a unique positive definite P_∞ solution of the following equation called Algebraic Riccati Equation (ARE)

$$P_\infty = Q + A^T P_\infty A - A^T P_\infty B (R + B^T P_\infty B)^{-1} B^T P_\infty A. \quad (1.3)$$

- exploiting the first-order necessary and sufficient conditions for optimality, the optimal control law is a closed form solution, feedback of the state

$$u_t^* = Kx_t^* \quad (1.4)$$

with

$$K = -(B^T P_\infty B + R)^{-1} (B^T P_\infty A) \quad (1.5)$$

and it is able to asymptotically stabilize the system.

The pair (x_t^*, u_t^*) is the optimal state and input at discrete time instant t .

Despite the fact that one of the main applications of LQ optimal control is regulation, this strategy can be used to track a desired system behaviour as closely as possible, in the framework of Linear Quadratic Tracking (LQT). To incorporate the desired behaviour to be tracked in the objective, the problem can be written as:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & J = \sum_{t=0}^{+\infty} (x_t - x_{t,\text{des}})^T Q (x_t - x_{t,\text{des}}) + u_t^T R u_t \\ \text{subj. to} \quad & x_{t+1} = Ax_t + Bu_t \quad t = 0, 1, \dots \\ & x_0 = x_{\text{init}} \end{aligned} \quad (1.6)$$

where x_{des} is the reference signal we want our system's states to track over time.

Cost function quantifies the controlled system's performance, defined as a weighted sum of the deviations of the system's states from desired values and control effort.

In the context of tracking piece-wise affine desired state behaviour the optimal feedback control law is modified in the following way:

$$u_t^* = K(x_t - x_{t,\text{des}}) \quad (1.7)$$

where the term $x_t - x_{t,\text{des}}$ is the tracking error at time t .

The effectiveness of LQR lies in its ability to leverage the full state information of the system, enabling precise and efficient control.

1.2 Linear Model Predictive Control

Model Predictive Control (MPC) is a modern control strategy renowned for its capacity to provide optimized responses while accounting for state and input constraints, such as saturation limits. These constraints can be explicitly integrated into the control design, enhancing the robustness and performance of the system. The origin of MPC trace back to the mid-seventies to mid-eighties, initially emerging in industrial applications. However, it was not until the nineties that MPC theory underwent significant advancements, leading to its widespread adoption and refinement.

MPC relies on the provided dynamic model to predict the behaviour of the system and determine the optimal control action based on the chosen performance criteria. A key aspect of MPC design is selecting a suitable model, one that adequately describes the system's dynamics while maintaining simplicity to ensure tractability and real-time solvability of the optimization problem.

The description of the linear discrete-time system prediction model is the same shown in 1.1.

For what regards the states and inputs constraints they can be expressed as follows

$$x_t \in \mathcal{X} \quad (1.8)$$

$$u_t \in \mathcal{U} \quad (1.9)$$

the state set \mathcal{X} is closed and the input set \mathcal{U} is compact and non-empty. Usually the sets are described with linear inequalities and both sets are convex and contain the origin.

Similarly to the LQR case, Model Predictive Control aims to minimize a cost function by defining positive definite matrices Q and R . The goal is to find the optimal control input that minimizes a cost function that takes the following form:

$$J(x_t, \mathbf{u}, t) = \sum_{i=t}^{+\infty} x_{i|t}^T Q x_{i|t} + u_{i|t}^T R u_{i|t} \quad (1.10)$$

In this formulation $x_{i|t}$ represents the predicted state at time i , given the actual state $x_{t|t}$. Similarly, $u_{i|t}$ is the predicted control sequence at current time t found considering the entire future predicted trajectory for the system.

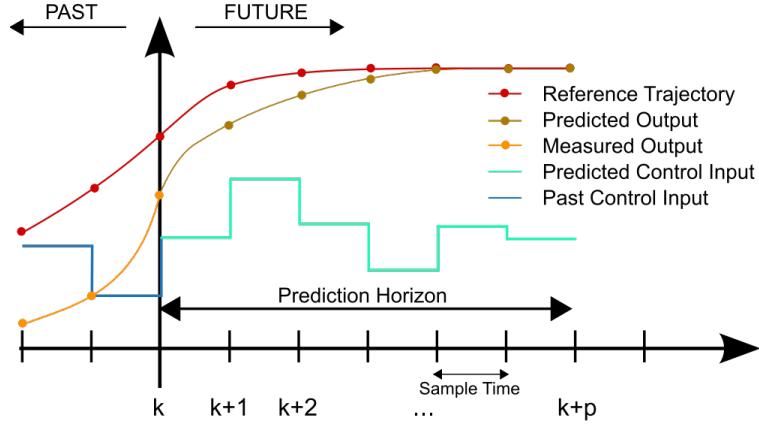


Figure 1.1: Model Predictive Control receiding horizon scheme

The aforementioned formulation represents the infinite horizon MPC problem, that in many practical cases results challenging to be implemented or requires highly demanding computations. For this reason usually MPC formulations optimize only on a finite horizon N , thus the problem of tracking x_{des} with the system's states becomes:

$$\begin{aligned}
 \min_{\mathbf{x}, \mathbf{u}} \quad & J = \sum_{i=t}^{t+N-1} (x_{i|t} - x_{t,\text{des}})^T Q (x_{i|t} - x_{t,\text{des}}) + u_{i|t}^T R u_{i|t} \\
 \text{subj. to} \quad & x_{i+1|t} = Ax_{i|t} + Bu_{i|t} \quad t = 0, 1, \dots \\
 & x_{i|t} \in \mathcal{X} \\
 & u_{i|t} \in \mathcal{U} \\
 & x_{t|t} = x_{\text{meas}}
 \end{aligned} \tag{1.11}$$

The control law is applied based on the receding horizon principle, which is a key feature of MPC. This principle dictates that at each time instance t , an open-loop optimal control problem is solved using the current state of the plant x_{meas} as the initial state, in order to obtain the control action. All predicted states and inputs within a prediction horizon N are optimized to find an optimal input sequence. However, only the first input of the optimal control sequence generated by the optimization is injected into the plant. This input is used to control the system over the next time step. This procedure is repeated, as illustrated in figure 1.1, at each time instance and by continuously updating the control action based on the most recent information, MPC ensures that the control strategy adapts to changes in the system dynamics and external conditions over time.

1.3 Offset-free tracking MPC and disturbance observer

Even if MPC is able to react to small unmodeled dynamics effects through the receding-horizon fashion, it is not able to consider them inside the predictions, thus to optimize upon them. MPC solves at each time an optimal control problem using the nominal model of the controlled process and, only when the actual plant behaviour closely matches the nominal model, the feedback control scheme can be guaranteed to stabilize the system, and achieve precise tracking of desired setpoints without any offset.

Offset-free Model Predictive Control represents an advanced control strategy designed to address this limitation. It aims to achieve zero steady-state tracking errors in the controlled system, even in presence of disturbances or model uncertainties that cause model-plant mismatch. For instance, in the context of regulating nonlinear systems using a linear predictive control framework, offset-free MPC seeks to regulate the system to its desired setpoint without steady-state error, ensuring precise and accurate control performance.

Consider a discrete-time, time-invariant system describing the plant:

$$\begin{aligned} x_p^+ &= f_p(x_p, u) \\ y_p^+ &= h_p(x_p) \end{aligned} \tag{1.12}$$

where $x_p \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y_p \in \mathbb{R}^p$ are the state, input and plant measured output respectively.

The aim of the offset-free model predictive controller is to have y_p tracking a certain reference signal $y_{\text{des}} \in \mathbb{R}^p$ that is assumed to asymptotically converge to a constant.

Let $w \in \mathbb{R}^n$ and $v \in \mathbb{R}^p$ be defined as:

$$\begin{aligned} w &:= f_p(x_p, u_t) - (Ax + Bu) \\ v &:= h_p(x_p) - Cx \end{aligned} \tag{1.13}$$

where the A , B and C matrices are the ones mentioned in the linear model 1.1. In particular the w term express the model mismatch between the non-linear plant model and the linear predictive model.

The key feature of this control strategy lies in the incorporation of a disturbance model in the prediction model 1.1. This is used to estimate and predict the model-plant mismatch between 1.1 and 1.12 in steady state, thus the w term.

Accordingly, an augmented model is used for computing the system's predictions in the offset-free MPC framework. It can be, in general, written in the following form:

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + B_d d_t \\ d_{t+1} &= d_t \\ y_t &= Cx_t + C_d d_t \end{aligned} \tag{1.14}$$

where the additional state $d_t \in \mathbb{R}^{n_d}$ is known as disturbance state or simply disturbance and follows an integral dynamics, as proposed by [14]. The maximum dimension of the disturbance state has to be equal to the number of measured outputs, i.e. $n_d \leq p$. The pair (B_d, C_d) can be seen as the disturbance model.

Assuming observability of the augmented system 1.14, an estimator that continuously estimates the augmented state (x, d) can be designed. An example could be a linear disturbance observer, that uses the prediction error between the real plant output and the predicted one, to estimate the augmented state as described by the following dynamical system:

$$\begin{aligned} \begin{bmatrix} \hat{x}_{t+1} \\ \hat{d}_{t+1} \end{bmatrix} &= \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}_t \\ \hat{d}_t \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} L_x \\ L_d \end{bmatrix} (-y_{p,t} + \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} \hat{x}_t \\ \hat{d}_t \end{bmatrix}) \\ &= A_e \begin{bmatrix} \hat{x}_t \\ \hat{d}_t \end{bmatrix} + B_e u + L(-y_{p,t} + C_e \begin{bmatrix} \hat{x}_t \\ \hat{d}_t \end{bmatrix}) \end{aligned} \tag{1.15}$$

where \hat{x} and \hat{d} are the state and disturbance estimates.

Given the current estimates of the augmented state (\hat{x}_t, \hat{d}_t) the MPC classical formulation is modified into an offset-free MPC algorithm as follows:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad J &= \sum_{i=t}^{t+N-1} (x_{i|t} - \bar{x}_t)^T Q (x_{i|t} - \bar{x}_t) + (u_{i|t} - \bar{u}_t)^T R (u_{i|t} - \bar{u}_t) \\ \text{subj. to} \quad x_{i+1|t} &= Ax_{i|t} + Bu_{i|t} + B_d d_{i|t} \quad t = 0, 1, \dots \\ d_{i+1|t} &= d_{i|t} \\ x_{i|t} &\in \mathcal{X} \\ u_{i|t} &\in \mathcal{U} \\ x_{t|t} &= \hat{x}_t \\ d_{t|t} &= \hat{d}_t \end{aligned} \tag{1.16}$$

where \bar{x}_t and \bar{u}_t are solution of the following system:

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_t \\ \bar{u}_t \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}_t \\ y_{\text{des}} - C_d \hat{d}_t \end{bmatrix} \quad (1.17)$$

This optimization cancels the estimated disturbances \hat{d}_t and thanks to this action is able to track the given feasible setpoint with the controlled variables.

1.3.1 Observer design

In this formulation the nominal augmented state (x, d) needs to be estimated at each time instant, given the plant output measurement y_p .

The $L_x \in \mathbb{R}^{n \times p}$ and $L_d \in \mathbb{R}^{n \times n_d}$ observer gain matrices in 1.15 has to be designed in such a way the estimator to be stable, while the method doesn't matter for our purposes.

Two possible choices of state estimator could be Luenberger observer or the Kalman filter. Specifically in the context of this thesis the Luenberger observer has been used. The Luenberger observer updates the state prediction computed at time $t - 1$ using the current measured plant output y_p at time t as in the following:

$$\hat{x}_t = A\hat{x}_{t-1} + Bu_{t-1} + L(y_{p,t} - \hat{y}_t) \quad (1.18)$$

where $\hat{y}_t := C(A\hat{x}_{t-1} + Bu_{t-1})$. The difference $(y_{p,t} - \hat{y}_t)$ is called estimation error or correction term and $L \in \mathbb{R}^{n \times p}$ is the observer gain.

If the pair (A_e, C_e) in 1.15 is observable, than the eigenvalues of $(A_e + LC_e)$ can be placed arbitrarily in such a way the estimator dynamical system to be stable. Usually we want the estimator to be faster with respect to the controller, for this reason the desired eigenvalues for $(A_e + LC_e)$, thus the poles of the estimator dynamical systems, are chosen smaller with respect to the ones of the controlled stabilized system.

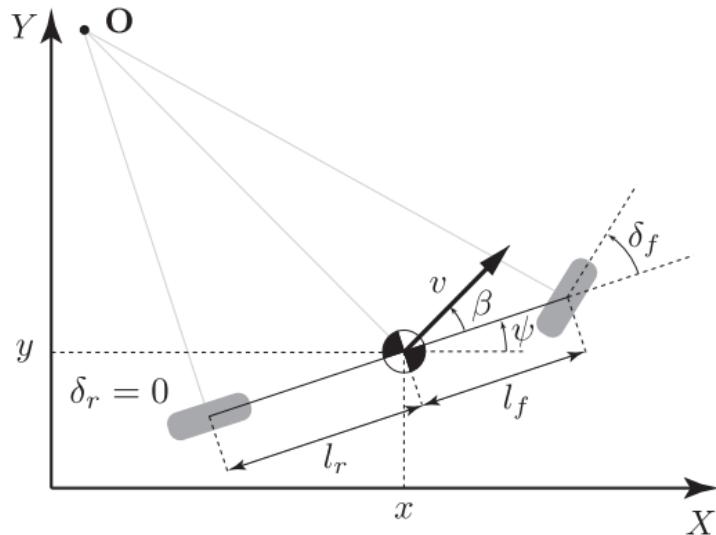
Chapter 2

Control design

This chapter will provide first of all a description of the plant model and of the linear approximation used to design the predictive controllers.

Later on a proper description of the design choices and the formulations of both the controllers in the context of the sport application: the Linear Quadratic Regulator (LQR) and the Model Predictive Control (MPC), nominal and with Offset-free version.

2.1 Model



Having a good model plays a central role in the controller design framework. The non linear model used to describe the kart dynamics is a bicycle model. 2D

bicycle model can be seen as a simplified vehicle model that approximates very well the motion of a 4-wheel vehicle in normal driving conditions.

The mathematical equations are the following:

$$\begin{cases} \dot{x}(t) = v(t) \cos(\psi(t) + \beta(t)) \\ \dot{y}(t) = v(t) \sin(\psi(t) + \beta(t)) \\ \dot{\psi}(t) = \frac{v(t)}{l_r} \sin(\beta(t)) \\ \dot{v}(t) = \frac{F_x(t)}{m} \end{cases} \quad (2.1)$$

where $x = [x, y, \psi, v]$ is the state vector where the components are respectively the x and y world frame coordinates, ψ the heading angle with respect to the reference frame, and v the total speed of the vehicle.

The longitudinal force F_x acting on the vehicle is modeled as a single force applied to the center of gravity of the vehicle and is computed as a combination of the drive-train command and the velocity.

$$F_x(t) = (C_{m_1} - C_{m_2}v(t))a(t) - C_f v(t) - C_d v^2(t) - C_{roll} \quad (2.2)$$

$$\beta(t) = \arctan\left(\tan\left(\frac{l_r}{l_f + l_r}\delta_f(t)\right)\right). \quad (2.3)$$

The inputs of this system are $u = [a, \delta_f]$ where a is the drive-train acceleration and δ_f the front steering angle. The variable β is the slip angle at the center of mass.

The model has two parameters, l_f and l_r , which represent the distances from the centre of mass to the front axle and rear axle, respectively.

The mass $m = 300$ Kg includes both the kart and the trailer weight, while $l_f = l_r = 0.8$ m. The other parameters of this model have been identified as reported in table 2.1.

The identified coefficients represent:

- C_{roll} : rolling coefficient modeling the wheel friction on the road surface
- C_d : coefficient modeling aerodynamic resistance or drag
- C_f : coefficient modeling the dynamic friction resistance
- C_{m_1} : coefficient associated with the force component due to the driving torque of the vehicle

C_{m1}	930
C_{m2}	0
C_f	10
C_d	1.5
C_{roll}	73

Table 2.1: Identified parameters for bicycle model

- C_{m2} : coefficient associated with the force component due to both the torque and the velocity

According to the estimated values the 2.2 can be rewritten as follows:

$$F_x(t) = C_{m1}a(t) - C_f v(t) - C_d v^2(t) - C_{roll} \quad (2.4)$$

The physical actuations limits on the input variable a can be expressed as box constraints of the following type:

$$a_{\min} \leq a_t \leq a_{\max} \quad (2.5)$$

The coefficients reported in table 2.1 are identified in such a way to have $a_{\min} = 0$ and $a_{\max} = 1$ as actuation limits for the motor torque.

2.2 Linear approximation of the system dynamics

The application for which the system is designed is for training Olympics runner in overspeed training. Due to the high velocities involved in this kind of training technique, the athlete is interested to execute only distances of about 50/60 m. For this reason it is reasonable to consider only the longitudinal motion of the vehicle along one direction and not taking anymore into account the steering degree of freedom. According this assumption the plant model can be simplified as in the following:

$$\begin{cases} \dot{p}(t) = v(t) \\ \dot{v}(t) = \frac{F_x(t)}{m} = \frac{1}{m}(C_{m1}a(t) - C_f v(t) - C_d v^2(t) - C_{roll}) \end{cases} \quad (2.6)$$

where p indicates the position of the vehicle along an arbitrarily oriented direction.

It's very easy to notice that a good linear approximation of 2.6 can be found and used for the controller design. The only non linear term is in fact the one related to

the drag coefficient in the equation describing the longitudinal force. Assuming that in the context of this application, the velocities the kart exploits during the motion are quite small, this term will be small as well.

Accordingly, continuous time dynamics equations of the linear approximation model can be obtained by just removing that non linear term:

$$\begin{cases} \dot{p}(t) = v(t) \\ \dot{v}(t) = \frac{F(t)}{m} = \frac{1}{m}(C_{m1}a(t) - C_f v(t)) \end{cases} \quad (2.7)$$

The Forward Euler method is used for discretize the given continuous-time dynamical system:

$$y_{t+1} = y_t + dt \cdot f(y_t, t) \quad (2.8)$$

where f is the continuous time function and dt is the discrete-time step size, distance between two consecutive discretization steps.

It is possible to write the following linear discrete-time, time-invariant system:

$$\begin{cases} p_{k,t+1} = p_{k,t} + dt v_{k,t} \\ v_{k,t+1} = v_{k,t} + dt \left(\frac{C_{m1}}{m} u_t - \frac{C_f}{m} v_{k,t} \right) \end{cases} \quad (2.9)$$

And moving into the state space form it can be rewritten as:

$$\begin{aligned} x_{k,t+1} &= \begin{bmatrix} p_{k,t+1} \\ v_{k,t+1} \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 - dt \frac{C_f}{m} \end{bmatrix} x_{k,t} + \begin{bmatrix} 0 \\ dt \frac{C_{m1}}{m} \end{bmatrix} u_t \\ &= Ax_{k,t} + Bu_t \end{aligned} \quad (2.10)$$

where $x_k = [p_k, v_k]^T$ are the kart state variables expressing respectively the kart position and the kart velocity and $u = a$ is the control input of the system, while dt is the sampling step used for Euler forward integration.

This linear model is the one used as basic for the controllers design in the following.

2.3 LQR design

Consider that at each time instant the information about the runner state $x_r = [p_r, v_r]^T$ is known, where p_r represents the absolute runner position and v_r indicates the absolute runner velocity.

At each time instant t the desired state for the go-kart is given by:

$$x_{k,\text{des}} = \begin{bmatrix} p_r + d_{\text{des}} \\ v_r \end{bmatrix} \quad (2.11)$$

where d_{des} denotes the reference distance between the kart and the runner. Using this reference, the closed-form control law in 1.7 is applied to obtain the optimal unconstrained solution u^* , at each time instant.

While MPC explicitly incorporates constraints into the optimization problem, LQR does not provide a direct mechanism for handling constraints. They can be indirectly addressed by penalizing their violation in the cost function, but it's still possible to deal with control actions that violate certain physical limits, such as those described in 2.5. For this reason the unconstrained control actions generated by the LQR controller has been "clipped" as follows:

$$\tilde{u}^* = \text{sat}_{[a_{\min}, a_{\max}]}(u^*) \quad (2.12)$$

The value \tilde{u}^* can be referred to as the clipped-unconstrained optimal LQR control action. While clipping allows the controller to handle constraints in a straightforward manner, it can lead to suboptimal performances, as the controller may be forced to operate at the constraint boundaries. Therefore, alternative approaches such as MPC may be preferred for systems with stringent constraint requirements.

2.3.1 Catch-up maneuver

The most difficult part of the motion is the first couple of seconds where the runner has an acceleration bigger than the maximum one the kart is able to develop. This phase has been called catch-up maneuver. For this reason, in order to avoid a crash between the airshield and the runner, this one is asked to start the sprint performance some meters farther from the kart with respect to the steady state desired distance d_{des} .

For this reason the initial condition for the kart state is set to be the following:

$$x_{k,\text{init}} = \begin{bmatrix} d_{\text{des}} + \bar{d} \\ 0 \end{bmatrix} \quad (2.13)$$

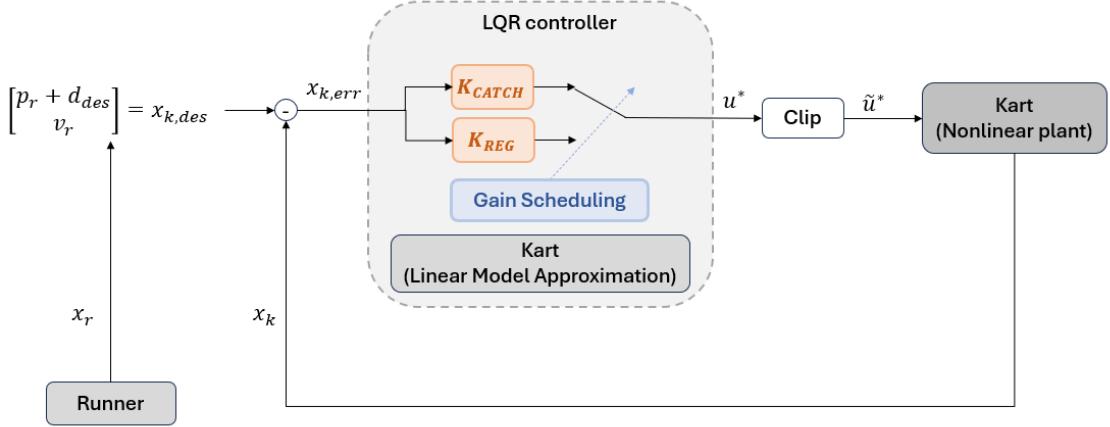


Figure 2.1: Linear Quadratic Regulator control scheme

where \bar{d} depends on how fast the runner is going to accelerate during his performance. Usually, for professional runners velocity profiles, this value is between 7 m and 10 m.

From the control architecture point of view a gain scheduling approach has been introduced in the control scheme. The catch-up is regulated with a controller gain K_{catch} computed using the following weights:

$$Q_{\text{catch}} = \begin{bmatrix} 40 & 0 \\ 0 & 1000 \end{bmatrix}, \quad R_{\text{catch}} = [2] \quad (2.14)$$

As clearly understandable from the weights, during the catch-up maneuver, the primary interest is to match the go-kart velocity with the runner one. Once they are matched at 80% the controller switches to K_{reg} that gives almost the same importance to distance and velocity mismatch.

$$Q_{\text{reg}} = \begin{bmatrix} 1000 & 0 \\ 0 & 4000 \end{bmatrix}, \quad R_{\text{reg}} = [0.002] \quad (2.15)$$

The overall control scheme is the one shown in 2.1, while algorithm 1 describes the points followed by the LQR algorithm while controlling the system.

Algorithm 1 LQR implementation

```

1: caught = false;
2: Set initial runner position with a proper distance  $\bar{d}$  from kart;
3: for  $t = 0, 1, 2, \dots$  do;
4:   Take kart state information  $x_{k,t} = [p_{k,t}, v_{k,t}]^T$ ;
5:   Take runner state information  $x_{r,t} = [p_{r,t}, v_{r,t}]^T$ ;
6:   Compute corresponding desired state for kart  $x_{k,t,des} = [p_{r,t} + d_{des}, v_{r,t}]^T$ ;
7:   if  $v_{k,t} \geq 0.8v_{r,t}$  and caught = false then
8:     caught = true;
9:   end if
10:  if caught = false then
11:     $u^* = -K_{\text{catch}}(x_{k,t} - x_{k,t,des})$ ;
12:  else
13:     $u^* = -K_{\text{reg}}(x_{k,t} - x_{k,t,des})$ ;
14:  end if
15:  Clip input according to limit saturation limits  $\tilde{u}^* = \text{sat}_{[a_{\min}, a_{\max}]}(u^*)$ ;
16:  Inject  $\tilde{u}^*$  in the plant;
17: end for

```

2.4 MPC design

The main purpose of the Model Predictive Control scheme design lies in the possibility of including constraints, in the incorporation of a runner model inside the prediction one and in the simplification of the overall control scheme. The gain scheduling approach switching between two controllers depending on the operation conditions can be replaced with a single MPC controller. Including explicitly constraints in the MPC formulation allows also to encode safety guarantees in the formulation: some example should be a minimum safety distance or a bound on the maximum velocity in order to avoid injuries.

Consider the linear model approximation 2.10 for the go-kart dynamics, and introduce the following integrator model for the runner, considered as an autonomous system:

$$\begin{cases} p_{r,t+1} = p_{r,t} + dt v_{r,t} \\ v_{r,t+1} = v_{r,t} + dt a_{r,t} \end{cases} \quad (2.16)$$

The runner is considered to be well described by this constant acceleration mathematical model.

According to this assumption, the following augmented state has been used for the prediction model of the MPC:

$$x_p = \begin{bmatrix} \Delta p \\ \Delta v \\ v_k \end{bmatrix} \quad (2.17)$$

where $\Delta p := p_k - p_r$ and $\Delta v := v_k - v_r$, thus the relative distance and velocity between kart and runner at a given time, respectively.

The overall prediction model, in discrete-time, state-space formulation can be written as follows:

$$\begin{aligned} x_{p,t+1} &= \begin{bmatrix} \Delta p_{t+1} \\ \Delta v_{t+1} \\ v_{k,t+1} \end{bmatrix} = \begin{bmatrix} 1 & dt & 0 \\ 0 & 1 & -dt\frac{C_f}{m} \\ 0 & 0 & 1 - dt\frac{C_f}{m} \end{bmatrix} x_{p,t} + \begin{bmatrix} 0 \\ dt\frac{C_{m1}}{m} \\ dt\frac{C_{m1}}{m} \end{bmatrix} u_t + \begin{bmatrix} 0 \\ -dta_{r,t} \\ 0 \end{bmatrix} \\ &= A_p x_{p,t} + B_p u_t + a_t \end{aligned} \quad (2.18)$$

The runner actual acceleration is considered to be known and introduced in the prediction model as a linear affine term that allows the predictions to better fit with the real future runner behaviour. This term relative to the actual runner acceleration can be seen as the introduction of a known disturbance in the system.

The optimal control problem solved at each discrete time instant, in a receding horizon fashion, has the following formulation:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & J = \sum_{i=t}^{t+N-1} (x_{p,i|t} - x_{p,\text{des}})^T Q (x_{p,i|t} - x_{p,\text{des}}) + u_{i|t}^T R u_{i|t} \\ \text{subj. to} \quad & x_{p,i+1|t} = A_p x_{p,i|t} + B_p u_{i|t} + a_t \\ & u_{\min} \leq u_{i|t} \leq u_{\max} \\ & d_{\text{safe}} \leq \Delta p_{i|t} \\ & x_{p,t|t} = x_{p,\text{meas}} \end{aligned} \quad (2.19)$$

where the reference values for the prediction states are the following

$$x_{p,\text{des}} = \begin{bmatrix} d_{\text{des}} \\ 0 \\ 0 \end{bmatrix} \quad (2.20)$$

and d_{safe} is a minimum distance between go-kart and runner, ensuring a safe situation. N is the prediction horizon thus the number of control intervals the controller

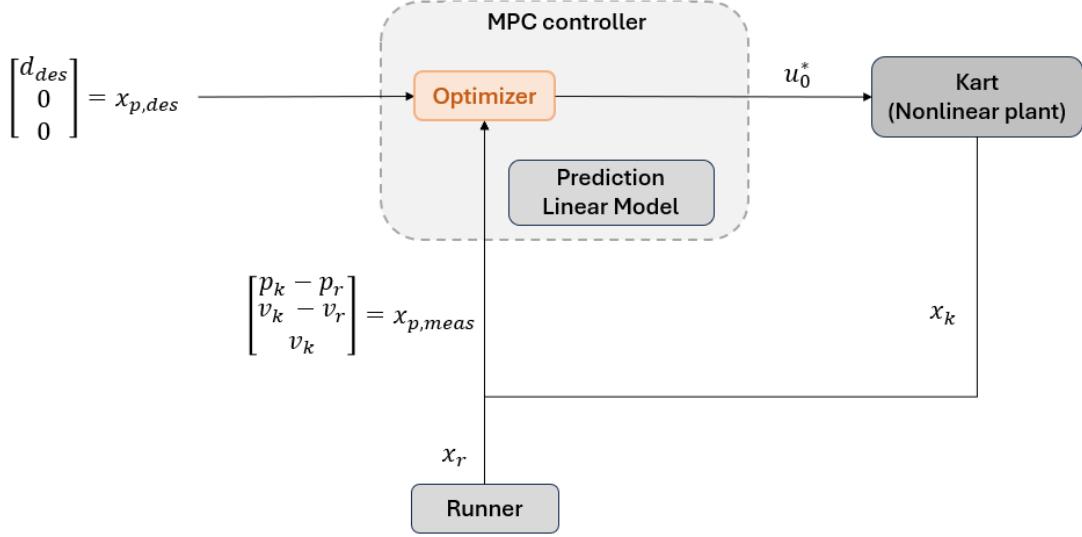


Figure 2.2: Model Predictive Control scheme

must evaluate by prediction. This parameter must be chosen carefully since the complexity of solving the optimization problem increases when the prediction horizon increases. This problem could have important consequences in systems having fast dynamics, where the number of prediction steps is limited by the processor power. At the same time the prediction horizon has to be chosen long enough to adequately capture the system dynamics. The optimal input sequence, found by repeatedly solving at each time step the optimizaton problem, is such that $\mathbf{u}^* \in \mathbb{R}^{m \times N}$.

The weights in the cost function are much less sensible to changes with respect to the ones used in the LQR formulation and are chosen like:

$$Q = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad R = [0.02] \quad (2.21)$$

The overall MPC control scheme is shown in 2.2. It is easy to understand the semplification in the architecture achieved with the MPC, with respect to the LQR case shown in 2.1. Algorithm 2 describes the points of the MPC implementation.

Algorithm 2 MPC implementation

-
- 1: Define desired prediction state $x_{p,\text{des}} = [d_{\text{des}}, 0, 0]^T$;
 - 2: Set initial runner position with a proper distance \bar{d} from kart;
 - 3: **for** $t = 0, 1, 2, \dots, t_{\text{end}}$ **do**
 - 4: Take prediction state information $x_{p,t} = [\Delta p_t, \Delta v_t, v_{k,t}]^T$;
 - 5: Solve the constrained QP and get $\mathbf{u}^* = \{u_0, u_1, \dots, u_N\}$;
 - 6: Inject the first element of the optimal input sequence u_0^* in the plant;
 - 7: **end for**
-

2.5 Offset-free MPC and disturbance observer design

The MPC problem formulated in 2.19 uses as prediction model a linear approximation of the kart model (see 2.10) incorporating the runner model in 2.16.

This prediction model does not take into account the non linear term due to the drag coefficient present in the identified kart dynamic equations in 2.6. In particular the true non linear prediction model for the MPC should be the following one:

$$\begin{bmatrix} \Delta p_{t+1} \\ \Delta v_{t+1} \\ v_{k,t+1} \end{bmatrix} = \begin{bmatrix} 1 & dt & 0 \\ 0 & 1 & -dt \frac{C_f}{m} \\ 0 & 0 & 1 - dt \frac{C_f}{m} \end{bmatrix} x_{p,t} + \begin{bmatrix} 0 \\ dt \frac{C_{m1}}{m} \\ dt \frac{C_{m1}}{m} \end{bmatrix} u_t + \begin{bmatrix} 0 \\ -dta_{r,t} \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ dt \left(\frac{C_d}{m} v_{k,t}^2 \right) \\ dt \left(\frac{C_d}{m} v_{k,t}^2 \right) \end{bmatrix} \quad (2.22)$$

where the last term includes the non linear, non modelled drag effect, thus the w in 1.13 representing the model-plant mismatch term in the offset-free MPC context.

Since MPC derives its solution based on the linear model in 2.18, the performance of the controller are directly related to the magnitude of the non modeled effects and in general the MPC cannot achieve optimal closed-loop performances. On the contrary, data-driven control strategies, do not require models and can implicitly manage uncertainties in processes. However, pure data-driven approaches, without any prior assumptions about process require a huge amount of data and slow exploratory techniques. Some recently explored control techniques combine model-based techniques with data-drive approaches.

The idea used in this thesis to compensate in the MPC formulation for the model-plant mismatch term, combines a predictive control based on the linear prediction model but it's able to estimate at run time, the non-modeled, non-linear term from the data and compensate for it using an offset-free approach. Even if the MPC in

offset-free formulation can compensate for the model-plant mismatch, the estimation of the disturbance is obtained from the measurement error. This introduces a delay in the estimation of the disturbance value and in its compensation. As a consequence, the disturbance term can be properly compensated only at steady-state and not during the transition state when that term varies at each time instant.

Considering the problem statement presented in 1.16, the affine term that takes into account the runner acceleration, has been added as done for the nominal MPC case.

The prediction model in 1.14 is applied and it results to be:

$$\begin{aligned} x_{p,t+1} &= \begin{bmatrix} 1 & dt & 0 \\ 0 & 1 & -dt\frac{C_f}{m} \\ 0 & 0 & 1 - dt\frac{C_f}{m} \end{bmatrix} x_{p,t} + \begin{bmatrix} 0 \\ dt\frac{C_{m1}}{m} \\ dt\frac{C_{m1}}{m} \end{bmatrix} u_t + \begin{bmatrix} 0 \\ -dta_{r,t} \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} d_t \\ d_{t+1} &= d_t \end{aligned} \quad (2.23)$$

Where $B_d = [0, 1, 1]^T$ means that the disturbance enters only in the second and third states while the prediction model for the relative position has no mismatch with the plant. For the purposes of this thesis the state is considered to be fully observable (i.e. $C = I_3$) and, as a consequence, the reference for the measured plant output is translated in reference for the plant states, thus $y_{des} = x_{p,des} = [d_{des}, 0, 0]^T$. Accordingly also $C_d = [0, 0, 0]^T$.

The observer design is very important for enhancing the robustness of the proposed control scheme. The L_x and L_d observer gain matrices they have been designed in such a way the poles of $(A_e + LC_e)$ are placed in $\lambda = [0.5, 0.51, 0.52, 0.53]^T$ and the matrix results to be Hurwitz. As weights the same Q and R matrices presented in the nominal MPC context, are used.

In image 2.3 the block diagram of the proposed offset-free MPC implementation is represented.

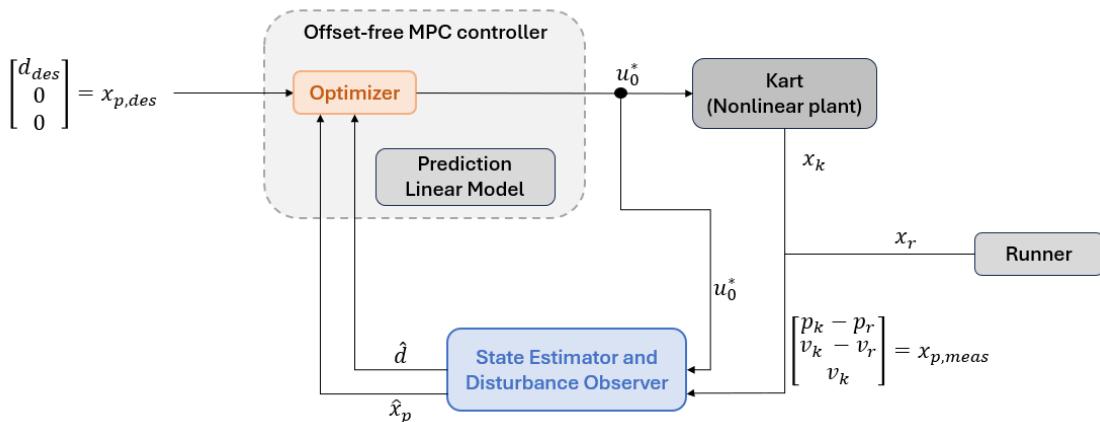


Figure 2.3: Offset-free Model Predictive Control scheme with disturbance observer

Chapter 3

Simulation results and Hardware-in-the-loop validation

The first purpose of this chapter is to present and analyze the results of Python simulations.

Later an overview of the hardware structure and sensors equipment of the system is provided together with the changes in the controllers formulations with respect to the simulation environment.

Finally, some test developed on the real hardware are presented and their results are compared with the simulation framework.

3.1 Simulation results

This section presents the results obtained carrying out a Python simulation of the go-kart non linear dynamical model. Two different velocity profiles for the runner are used as data and represented in figure (Plot solo profili velocità). The three controllers: Linear Quadratic Regulator, Model Predictive Control and Offset-Free Model Predictive Control have been tested and their performances are evaluated and compared.

3.2 System structure and hardware description

This part discuss the hardware components present in the system.

The system comprises a fully electric-powered go-kart with an attached airshield structure via a fixed joint. According to the CIK-FIA (International Karting Commission Federation International Automobile), a go-kart is a land vehicle with four



Figure 3.1: System structure

non-aligned wheels in contact with the ground, two of which control steering while the others transmit power. Go-karts emerged after the post-war period of the 1950s, originally created by airmen as a pastime. Typically, go-kart chassis are constructed from steel pipes that provide both stiffness and flexibility to compensate for the lack of a suspension system and a differential.

Electric-powered go-kart

The kart is a SinusIon purchased from the German company RiMO (originally Richter + Mohn).

The battery on board is a lithium-iron-manganese-4-phosphate (LiFeMnPO₄), used in order to reduce the battery pack weight.

For what regards the motor it's a Heinzmann PMS 100, with air-cooled cooling technology. This DC motor is a permanent magnet brush-less motor with a link voltage of 28V, a nominal speed of 6500 rpm and a rated torque of 3.82 Nm, representing respectively the rotational speed at which the motor is designed to operate optimally and the maximum rotational force that the motor can exert. Talking about efficiency this motor is characterized with high-efficiency and energy-saving capabilities, the rated power output is 2.6kW. This parameter reflects the motor's ability to convert electrical energy into mechanical power. The gearbox has a gear ratio of 1:7.

The computer on board is a AMD Ryzen MinisForum EliteMini X500 with an AMD Ryzen 7 5700G processor (8 cores with nominal frequency of 3.8GHz) coupled with a 16GB DDR4 RAM.

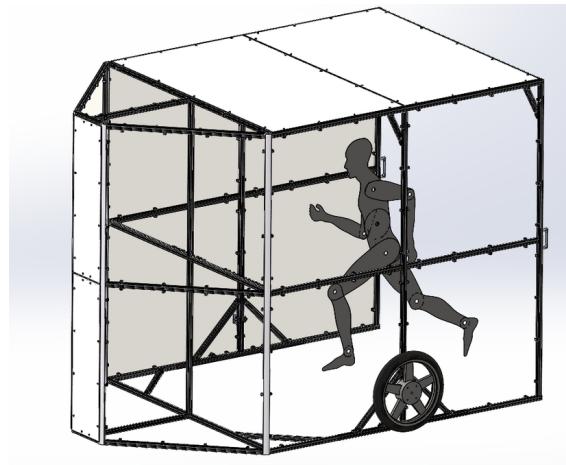


Figure 3.2: Airshield structure

Airshield

The shield has the structure shown in 3.2. All the side and front panels are of plexiglass while the top part is an aluminium composite.

The dimensions are reported in table 3.1.

Height	2.15 m
Width	1.86 m
Lenght	2.95 m

Table 3.1: Aishield dimensions

The height can be adjusted inside a range of circa 13 cm by simply moving a wheel bracket up or down, in order to better fit the height of the runner.

3.3 Perception

Autonomous vehicles sense and perceive the surrounding environment using measurements coming from sensors and then process the information in order to make informed decisions. This is the fundamental principle behind closed loop control that relies on measurements of controlled variables, provided by an appropriate sensor. The perception systems used in autonomous vehicles are usually categorized in two groups:

- Proprioceptive sensors (or internal state sensors) sensing the vehicle own state like wheel encoders or inertial measurement unit, location sensors like GPS,

etc.

- Exteroceptive sensors (or external state sensors) gathering informations about the surrounding environment like cameras, LiDAR, RADARs, etc.

In this application the system is equipped with a LiDAR (Light Detection and Ranging), a camera, an inertial measurement unit and wheel encoders.

LiDAR

LiDAR (Light Detected and Ranging) is a remote sensing technology that measures distances to objects and surfaces using laser pulses. The sensor consists of three main components: a laser emitter, a scanning mechanism, and a receiver. The laser emitter sends out short pulses of laser light, which travel towards surrounding objects. When these pulses encounter an object they reflect off its surface and return to the LiDAR sensor. The receiver then detects the reflected light and measures the time it takes for the pulses to return, allowing to calculate the distance of the object based on the speed of light. Moreover, LiDAR sensors are capable of operating in various environmental conditions including darkness, rain, fog or low visibility, making them perfect for the application and highly reliable.

The LiDAR mounted on the airshield is a Velodyne VLP-16, a compact sensor that features 16 laser channels arranged in a circular configuration to have an important field of view.

The field of view (angular extension of a scene that is projected by the laser beam) is of 360° horizontal and 30° vertical. The range of the laser is up to 100 m. Measures have an accuracy of +/- 3 cm and are acquired with a frequency of 20 Hz.

Camera

Cameras provide rich information about the surrounding environment and can be used in a variety of ways. On the airshield a ZED 2i Stereo Camera with polarized lens of 4mm is mounted.

The field of view is of 120° and the camera is able to sense ranges between 1.5 m and 35 m. The depth accuracy is smaller of the 2% of the distance up to 10 m while it increases up to the 7% for distances up to 30m distance.

The camera is equipped with a Stereolab ZED Box Xavier NX 8GB, an industrial compact and powerful mini PC used for computing and processing camera data. It's powered with an NVIDIA Jetson embedded GPU.

At the moment LiDAR measurement are prefered with respect to camera ones thanks to the higher accuracy. Camera remains responsible for the detection of the gesture to identify the moment in which the runner is able to start the performance.

Wheel Encoder

Rotary encoders are a type of sensor that measures the rotation of a mechanical shaft. Two wheel encoders, one for each wheel, are integrated in the PMS 100 motor as speed sensors providing the motor rotational speed. Then by knowing that the gear ratio is 1:7 and that the wheel diameter is 27.89 cm the absolute go-kart speed can be evaluated as follows:

$$v_{\text{kart}} = \frac{\pi \cdot d \cdot \text{avg rate}}{60 \cdot \text{gear ratio}} \quad (3.1)$$

where avg rate is the average motor rate between left and right sides.

Inertial Measurement Unit

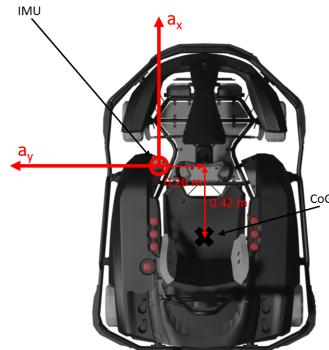


Figure 3.3: IMU position on the go-kart

Typically an IMU or Inertial Measurement Unit consists of a combination of accelerometers and a gyroscopes, and sometimes also magnetometers in order to measure the orientation, velocity and acceleration of an object. Thanks to the combination of accelerometers and gyroscopes the IMU is able to sense the object orientation or angular motion and acceleration with respect to some predefined axes. The IMU used in the go-kart is the IRIMU-V2 from Izze-Racing that outputs data at 200Hz via CAN protocol and has an accuracy of less than 1% of full scale for the acceleration and less than 1.5% for the angular acceleration. The position with respect to the go-kart centre of gravity is showed in 3.3.

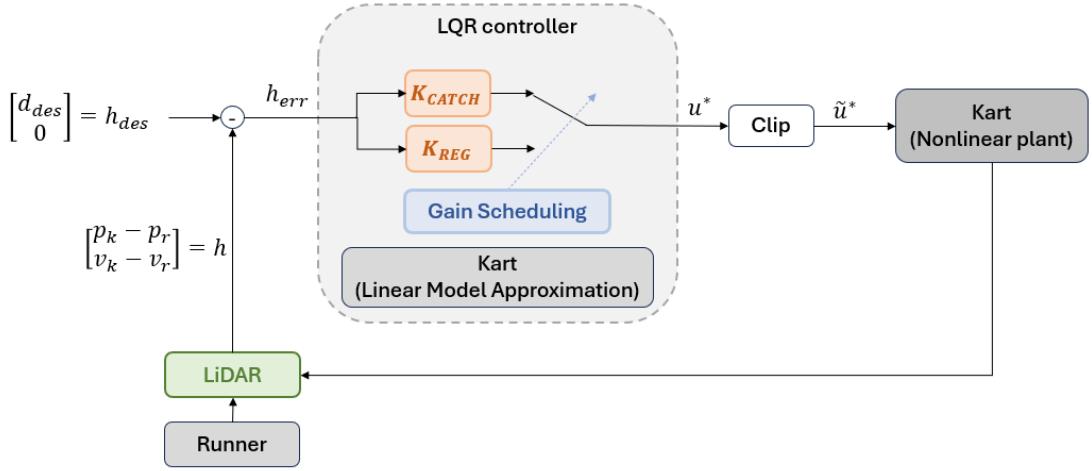


Figure 3.4: Linear Quadratic Regulator scheme with sensors

3.3.1 Controller design changes for hardware integration

In the context of the hardware control implementation the available informations are the one coming from sensors and we cannot rely on a full state knowledge for both runner and kart as hypotized in Chapter 2. For this reason we have to take into account the sensor present in the system and the measurement provided by them to control the system properly.

LQR

Differently from the simulation context, where all the state variables both for runner and kart have been considered to be fully known, in the hardware-in-the-loop context, absolute position of both kart and runner, as well as the runner absolute velocity, result to be unknown.

The LiDAR, as a range sensor, provides directly the measurement of the relative distance between runner and kart Δp , while the relative velocity can be estimated via finite differences:

$$\Delta v_t = \frac{\Delta p_t - \Delta p_{t-1}}{dt} \quad (3.2)$$

where $\Delta p_t := p_{k,t} - p_{r,t}$ and $\Delta v_t := v_{k,t} - v_{r,t}$, and dt is the sampling time.

The third component of the prediction state, the kart absolute velocity, is obtained from the wheel encoders.

The closed loop control law in 1.7 can be modified as follows to directly use the

Algorithm 3 LQR implementation on hardware system

```

1: caught = false;
2: Set initial runner position with a proper distance  $\bar{d}$  from kart;
3: Define desired system behaviour  $h_{\text{des}} = [d_{\text{des}}, 0]^T$ ;
4: for  $t = 0, 1, 2, \dots$  do;
5:   Take LiDAR information  $h_t = [\Delta p_t, \Delta v_t]^T$ ;
6:   Take absolute kart velocity  $v_{k,t}$  from wheel encoders;
7:   Evaluate absolute runner velocity  $v_{r,t} = v_{k,t} - \Delta v_t$ ;
8:   if  $v_{k,t} \geq 0.8 v_{r,t}$  and caught = false then
9:     caught = true;
10:    end if
11:    if caught = false then
12:       $u^* = -K_{\text{catch}}(h_t - h_{\text{des}})$ ;
13:    else
14:       $u^* = -K_{\text{reg}}(h_t - h_{\text{des}})$ ;
15:    end if
16:    Clip input according to limit saturation limits  $\tilde{u}^* = \text{sat}_{[a_{\min}, a_{\max}]}(u^*)$ ;
17:    Inject  $\tilde{u}^*$  in the plant;
18: end for

```

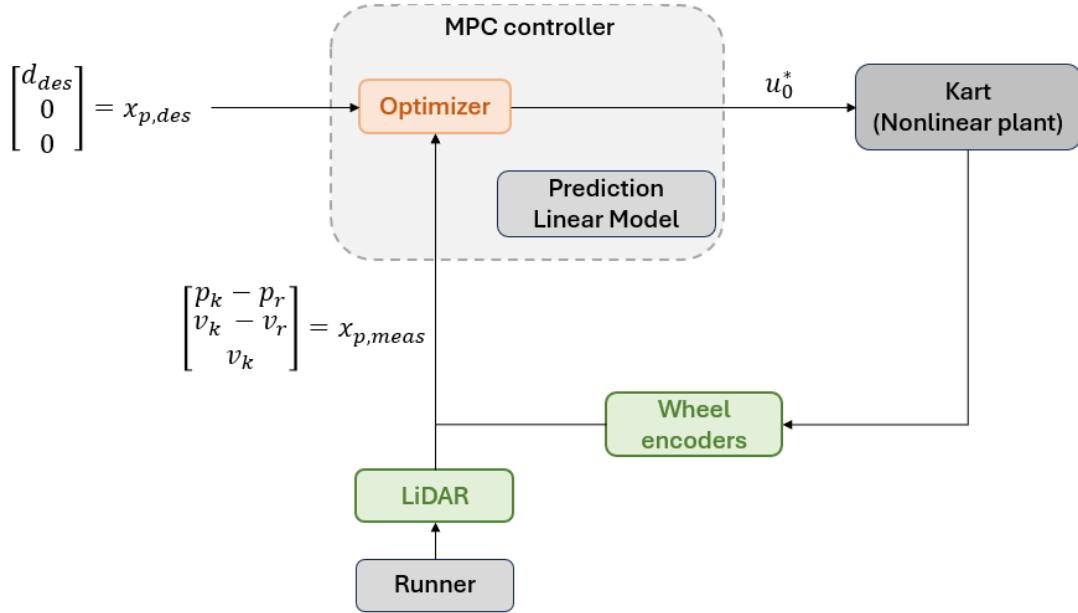


Figure 3.5: Model Predictive Control scheme with sensors

measurements provided by the LiDAR to evaluate the feedback error and compute the current control action:

$$u_t^* = K(h_t - h_{des}) \quad (3.3)$$

where

$$h_t = \begin{bmatrix} \Delta p_t \\ \Delta v_t \end{bmatrix} = \begin{bmatrix} p_{k,t} - p_{r,t} \\ v_{k,t} - v_{r,t} \end{bmatrix}, \quad h_{des} = \begin{bmatrix} d_{des} \\ 0 \end{bmatrix}. \quad (3.4)$$

MPC

The overall implementation of the LQR in the context of the real system is described by algorithm 3, while the control architecture using the sensors for taking the measurements is shown in figure 3.5.

It has to be noticed that the actual absolute runner acceleration is estimated. First of all the relative acceleration between kart and runner Δa is estimated, as well as the kart absolute acceleration a_k . The absolute runner acceleration used is equal to $a_r = a_k - \Delta a$. For this purpose a Kalman filter has been used.

Tutto il discorso sul noise

Bibliography

- [1] H. Müller and B. Glad, “Technology in athletics,” *Journal of Sports Science*, 2014.
- [2] R. J. Corn and D. Knudson, “Effect of elastic-cord towing on the kinematics of the acceleration phase of sprinting,” *Journal of Strength and Conditioning Research*, vol. 17, no. 1, pp. 72–75, 2003.
- [3] W. P. Ebben, J. A. Davies, and R. W. Clewien, “Effect of the degree of hill slope on acute downhill running velocity and acceleration,” *Journal of Strength and Conditioning Research*, vol. 22, no. 3, pp. 898–902, 2008.
- [4] CONI. “Presentato lo scudo aerodinamico, l’arma in piú al servizio dell’atletica.” (2021), [Online]. Available: <https://www.coni.it/it/news/18045-presentato-lo-scudo-aerodinamico,-l-arma-in-piu-al-servizio-dell-atletica.html>.
- [5] S. E. S. D. Director, “Review of the state of development of advanced vehicle control systems (avcs),” *Vehicle System Dynamics*, vol. 24, no. 6-7, pp. 551–595, 1995.
- [6] K. Yi, J Hong, and Y. Kwon, “A vehicle control algorithm for stop-and-go cruise control,” *School of Mechanical Engineering, Hanyang University, Seoul, Korea; Hyundai Motor Company, Kyonggi-Do, Korea*, 2001.
- [7] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000, ISSN: 0005-1098.
- [8] W. Lim, S. Lee, J. Yang, M. Sunwoo, Y. Na, and K. Jo, “Automatic weight determination in model predictive control for personalized car-following control,” *IEEE Access*, vol. 10, pp. 19 812–19 824, 2022.
- [9] G. Naus, R. van den Bleek, J. Ploeg, B. Scheepers, R. van de Molengraft, and M. Steinbuch, “Explicit mpc design and performance evaluation of an acc stop-&-go,” in *2008 American Control Conference*, 2008, pp. 224–229.

- [10] G. Pannocchia, “Offset-free tracking mpc: A tutorial review and comparison of different formulations,” in *2015 European Control Conference (ECC)*, 2015, pp. 527–532.
- [11] U. Maeder, F. Borrelli, and M. Morari, “Linear offset-free model predictive control,” *Automatica*, vol. 45, no. 10, pp. 2214–2222, 2009, ISSN: 0005-1098.
- [12] A. Radke and Z. Gao, “A survey of state and disturbance observers for practitioners,” in *2006 American Control Conference*, 2006, 6 pp.–.
- [13] R. E. Kalman *et al.*, “Contributions to the theory of optimal control,” *Bol. soc. mat. mexicana*, vol. 5, no. 2, pp. 102–119, 1960.
- [14] G. Pannocchia and J. B. Rawlings, “Disturbance models for offset-free model predictive control,” *AIChe J.*, vol. 49, pp. 426–437, 2003.