RODRIGO DE MARCA FRANÇA

Longitudinal Vehicle Control with Active Braking Using Model-Free Controllers

São Paulo
2023

RODRIGO DE MARCA FRANÇA

Longitudinal Vehicle Control with Active Braking Using Model-Free Controllers

Corrected Version

Dissertation submited to Escola Politécnica of University of São Paulo for obtaining the degree of Master in Science

São Paulo
2023

RODRIGO DE MARCA FRANÇA

Longitudinal Vehicle Control with Active Braking Using Model-Free Controllers

Corrected Version

Dissertation submited to Escola Politécnica of University of São Paulo for obtaining the degree of Master in Science

Concentration area:

3139 - Systems Engineering

Advisor:

Prof. Dr. Bruno Augusto Angélico

São Paulo
2023

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catalogação-na-publicação

França, Rodrigo de Marca
Longitudinal Vehicle Control with Active Braking Using Model-Free Controllers / R. M. França -- versão corr. -- São Paulo, 2023.
89 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Telecomunicações e Controle.

1.CONTROLE (TEORIA DE SISTEMAS E CONTROLE) 2.SISTEMAS DINÂMICOS I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Telecomunicações e Controle II.t.

ACKNOWLEDGMENTS

To Prof. Dr. Bruno Augusto Angélico, for the academic guidance, friendship, patience, trust, and support during the whole master's period.

To my parents Mario Guilherme Valente França and Thais Lima de Marca, and my sister Priscila de Marca França, who always encouraged and supported me since the beginning of the master's course.

To Prof. Dr. Armando Antônio Maria Laganá, for academic guidance and support in parts of the work and for participating on the qualification, comments, and suggestions.

To Prof. Dr. Diego Colón, for participating on the qualification, comments, and suggestions.

To Prof. Dr. Gabriel Pereira das Neves, for guidance, friendship, and support and for participating on the dissertation defense, comments, and suggestions.

To Prof. Dr. Sergio Ribeiro Augusto, for guidance with the controllers used, friendship and for participating on the dissertation defense, comments, and suggestions.

To Prof. Dr. Vanderlei Cunha Parro and Me. Tiago Sanches da Silva, for encouraging me to initiate the master's course, and Prof. Dr. Rodrigo Alvite Romano, for indicating Prof. Dr. Bruno Augusto Angélico as my advisor.

To Me. Paulo Alexandre Pizara Hayashida, Me. André Vinícius Oliveira Maggio and Me. Rennan Santos de Araujo, for guidance regarding vehicles parts and simulators, and friendship.

To LAC (Laboratório de Automação e Controle), from Escola Politécnica of USP, for physical resources availability and all its members for the acquaintanceship.

To FATEC Santo André, for physical resources availability and all its members for the acquaintanceship.

To all my relatives, friends and all who collaborated, directly or indirectly, for the execution of this dissertation.

To the Fundação de Desenvolvimento da Pesquisa – Fundep Rota 2030 – Linha V, regarding the SegurAuto project – Integrated Development of Driver-Assisted Safety and Environmental Functions for Autonomous Vehicles, for financing and project structure.

"The world ends with you. If you want to enjoy life, expand your world. You gotta push your horizons out as far as they'll go."

– Tetsuya Nomura

RESUMO

Com o avanço da tecnologia automotiva a utilização de Sistemas Avançados de Assistência ao Condutor (ADAS) se tornam cada vez mais comuns. Esses sistemas permitem aumentar o conforto, segurança, confiabilidade e eficiência geral de veículos, auxiliando o condutor em diversas tarefas relacionadas a condução. Um desses sistemas é o Controle de Cruzeiro Adaptativo (ACC) que é capaz de controlar longitudinalmente veículos, permitindo manter velocidades fixas ou então desacelerar veículos em situações de emergência. Neste trabalho, são utilizadas técnicas de controle sem modelo (*model-free*) do tipo PID inteligente (iPID) para realizar o controle longitudinal de veículos sem que seja necessário realizar o levantamento de modelos matemáticos complexos. Em particular, esse trabalho utiliza um controle Proporcional inteligente (iP) para controlar longitudinalmente um veículo simulado, substituindo o condutor e controlando a aceleração e frenagem simultaneamente. As características mais interessantes desse controlador são discutidas, em especial sua capacidade de resistir a ruídos. Testes adicionais com um controlador Proporcional Integral (PI) clássico equivalente são realizados para propósitos de comparação. O trabalho também propõe um estimador de tempo real baseado em mínimos quadrados para o parâmetro α do Controle Sem Modelo (MFC) utilizado, criando uma família de Controle Sem Modelo Adaptativo (AMFC) denominada PID inteligente com estimador α (iPIDα). Ambos os controladores iP e iPα foram usados para controlar longitudinalmente veículos simulados em um simulador veicular comercial capaz de simulações dinâmicas de veículos altamente complexas. Vários testes do Programa Europeu de Avaliação de Carros Novos (Euro NCAP) foram realizados para testar a resposta do controlador a emergências, com resultados muito bons.

Palavras-Chave – Controle sem modelo, Controladores inteligentes, Controladores PID inteligentes, Controle longitudinal veicular, Implementação Discreta.

ABSTRACT

With the advancement of automotive technology, the use of Advanced Driver-Assistance Systems (ADAS) has become increasingly common. These systems allow an increase in comfort, safety, reliability, and overall efficiency of vehicles, helping the driver in various driving-related tasks. One of these systems is the Adaptive Cruise Control (ACC) which is capable of longitudinally controlling vehicles, allowing them to maintain fixed speeds or decelerate vehicles in emergency situations. In this work, model-free intelligent PID (iPID) control techniques are used for longitudinal control of vehicles without the need to complex mathematical models. In particular, this work uses an intelligent Proportional (iP) control to longitudinally control a simulated vehicle, replacing the driver and controlling the acceleration and braking simultaneously. The most interesting features of this controller are discussed, especially its ability to resist noise. Additional tests with an equivalent classical Proportional Integral (PI) controller are performed for comparison purposes. The work also proposes a real-time least square-based estimator for the used Model-Free Control (MFC) α parameter, creating an Adaptive Model-Free Control (AMFC) family named intelligent PID with α estimator (iPIDα). Both an iP and iPα controllers are used to longitudinally control vehicles simulated in a commercial vehicle simulator capable of highly complex vehicle dynamics simulations. Several tests from the European New Car Assessment Programme (Euro NCAP) were performed to test the controller's response to emergencies, with very good results.

Keywords – Model-Free control, Intelligent controllers, Intelligent PID controllers, Longitudinal vehicular control, Discrete Implementation.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACC	Adaptive Cruise Control
ADAS	Advanced Driver-Assistance System
AEB	Autonomous Emergency Braking
AMFC	Adaptive Model-Free Control
ASV	Active Vehicle System
BBW	Brake-by-Wire
CC	Cruise Control
CMS	Collision Mitigation Braking
ECU	Electronic Control Unit
EHB	Electro-Hydraulic Brake
ESC	Electronic Stability Control
Euro NCAP	European New Car Assessment Programme
FCW	Forward Collision Warning
FIR	Finite Impulse Response
HIL	Hardware-in-the-loop
IIR	Infinite Impulse Response
iP	intelligent Proportional
iPD	intelligent Proportional Derivative
iPI	intelligent Proportional Integral
iPID	intelligent Proportional Integral Derivative
iPα	intelligent Proportional with α estimator
iPDα	intelligent Proportional Derivative with α estimator
iPIα	intelligent Proportional Integral with α estimator
iPIDα	intelligent Proportional Integral Derivative with α estimator
LKAS	Lane Keeping Assist System
MFC	Model-Free Control

MPC	Model Predictive Control
P	Proportional
PFE	Pedal Force Emulator
PI	Proportional Integral
PID	Proportional Integral Derivative
SG	Stop-and-Go
SISO	Single-Input Single-Output
TMC	Tandem Master Cylinder
VUT	Vehicle Under Test

CONTENTS

1 INTRODUCTION

Among the great number of control problems associated with automotive research, automatic vehicle speed control is a topic of great interest as it allows to increase both the safety and comfort of passengers. In particular, Advanced Driver-Assistance System (ADAS) and Active Vehicle System (ASV) have been topics of research and development since the 1990s. Several commercial systems already include important functions such as Cruise Control (CC), Adaptive Cruise Control (ACC), Stop-and-Go (SG), Lane Keeping Assist System (LKAS), assisted lane change, automatic parking assistant, and Collision Mitigation Braking (CMS). Ultimately, a cruise control system such as CC, ACC, or SG seeks to partially automate the longitudinal control of a vehicle and reduce driver effort in low-speed urban traffic and high-speed highways.

One of the challenges of performing longitudinal control on a moving vehicle is the mechanical complexity involved in its operation. For example, just to control the vehicle acceleration several different mechanical and electronic subsystems need to be engaged, each with their characteristics. Creating a simplified mathematical model is possible, but this complexity makes the usage of control techniques that do not need a plant mathematical model particularly attractive for these applications.

There are several ways to obtain a control without knowing the plant's mathematical model. One of the most used is to tune a classical Proportional Integral Derivative (PID) controller with empirical adjustments (ÅSTRÖM; HÄGGLUND, 1995). Widely used in industrial applications, PID controls are reliable and easy to implement in analog and digital systems. Another less-used method to control unknown plants is to use Model-Free Controllers (MFC) techniques, which are usually capable of adapting themselves to different plants. Fliess and Join (2013) introduced a new approach to MFC that extends the original framework of PID controller, creating intelligent Proportional Integral Derivative (iPID) controllers that aim to be more robust and efficient than traditional ones. This is done with the introduction of an additional adjustment to the PID controller based on an ultra-local model, which must be constantly updated according to the input-output

behavior of the plant, and helps in tracking the desired reference. The iPID controllers have been shown to be highly interesting because they can be implemented at a low computational cost (JOIN; CHAXEL; FLIESS, 2013) and even solve complex non-linear problems (FLIESS, 2009). They have been successfully applied in various systems, such as magnetic levitation systems (MORAES; SILVA, 2015), building photovoltaic energy management (BARTH et al., 2019), hybrid unmanned autonomous vehicles (BARA et al., 2017), and active magnetic bearing (MIRAS et al., 2013). It is important to note that there is a wide literature on several other MFC techniques, iPID being just one of them.

Several works have already studied the application of MFC for the longitudinal control of vehicles (NOVEL, 2018; D'ANDRÉA-NOVEL et al., 2016; MENHOUR et al., 2015; MENHOUR et al., 2013). And some of them managed to apply these structures in actual vehicles with very interesting results (MILANÉS et al., 2012b; MILANÉS et al., 2012a; POLACK; DELPRAT; NOVEL, 2019).

In this work, a longitudinal control system was developed and tested for a simulated vehicle using iPID controllers, in particular, the iP controller. A highly adaptable and low computational cost original digital implementation for the ultra-local model estimator was devised and tested. Several tests were done to verify how the implemented controller behaves when subjected to variations in the vehicle model and measurement noise. An equivalent traditional PID controller, more specifically a PI controller, was submitted to the same tests to verify the differences and possible advantages and disadvantages of using an iPID controller for the application. An analysis to verify how the controllers were affected by variations in the plant was performed by modifying the road incline angle. The brake system behavior is of special interest because of the security implications of the controller failing to properly act on it.

A new form of the iPID controller coupled with a real-time estimator for its α parameter is proposed using the name iPIDα. This controller can be considered an Adaptive Model-Free Control (AMFC), capable of adapting itself even more than the traditional MFC. Mechanical Simulation's CarSim software, a commercial vehicle simulator capable of better simulating vehicle dynamics (CARSIM. . . , 2022) is used for testing both an iP and iPα controllers in CC and ACC applications. MFC and AMFC are interesting for this application since they allow abstracting the highly complex dynamics of the vehicle, such as the engine control performed by the Electronic Control Unit (ECU) and the low-level dynamics of the brake system. Several European New Car Assessment Programme (Euro NCAP) tests are performed with both controllers to test how effective the controllers are in emergency situations, with very good results.

The main objectives of this research are:

- Compare the iPID and PID controllers when applied to CC.
- Study, design and digitally implement iPID MFC.
- Prosope, digitally implement and test an iPID MFC.
- Study the different types of vehicular longitudinal control and their uses.
- Obtain a simplified simulator for a vehicle and use it to evaluate control loops;
- Propose, digitally implement, and test a real-time estimator for the α parameter of the used MFC.
- Propose, digitally implement, and test a iPIDα AMFC, a new form of the iPID controller with a real-time α estimator.
- Test the implemented iPID MFC and the iPIDα AMFC in CarSim vehicle simulator.
- Perform several Euro NCAP tests with the implemented MFC and AMFC controllers to test how they behavior in emergency situations.

The rest of the work is organized as follows: initially chapter 2 presents the research theoretical background, divided into types of vehicle longitudinal control (section 2.1), the Euro NCAP concept and general objectives (section 2.2), electronics braking system and vehicle modeling (section 2.3), on the chosen MFC theory and discrete implementation (section 2.4), and on the proposed α parameter and the AMFC created from it (section 2.5). Then the chapter 3 presents the results of the initial simulations, including the developed simulator (section 3.1), controller topology and tuning (section 3.2), and controllers' comparison (section 3.3). Chapter 4 presents the results of the final tests with CarSim software, with an overview of the simulator usage and configuration (section 4.1), iP controller tuning (section 4.2), iPα controller tuning (section 4.3), and the results of several Euro CNAP tests with them (section 4.4). Lastly, chapter 5 presents the general conclusions of the current work and some possibilities for future works.

2 THEORETICAL BACKGROUND

2.1 Vehicle Longitudinal Control

Offering passengers and drivers the highest level of safety, comfort, and efficiency by partially or completely removing driving duties from humans has been one of the most important goals in the automotive industry in recent years. One example of this is ADAS, which allows an improvement in road capacity (HU et al., 2020b) while also ensuring the safety of drivers and vulnerable road users to some extent (HU et al., 2020a; PENG et al., 2019).

2.1.1 Conventional Cruise Control

Within this scope, conventional CC systems were developed as part of ADAS to perform longitudinal control on vehicles, automatically tracking the desired cruise speed. While very useful for long road travels or no-traffic conditions, ordinary CC systems are becoming less and less meaningful because of the increase in traffic density, which makes driving at a constant pre-selected speed almost impossible.

This problem led to the development of more advanced CC systems, capable of controlling both speed and distance to preceding vehicles, increasing drivers' comfort, and reducing the danger of rear-end collision, especially in heavy traffic conditions. This is possible with the help of new vision technologies, such as long-range radars, enabling a new adaptive CC system, known as ACC system, capable of measuring the distance to obstacles and adapting its cruise speed reference as needed.

2.1.2 Adaptive Cruise Control

Since the 1990s, significant progress in sensors, actuators, and other enabling technologies for ADAS have been made (VENHOVENS; NAAB; ADIPRASITO, 2000; FANCHER; BAREKET; ERVIN, 2001; FANCHER et al., 2004; SHLADOVER, 1978; CHIEN; IOAN-

NOU; LAI, 1994; ISERMANN et al., 1995; YI et al., 2002), which allowed the creation of ACC systems. The most basic feature of the ACC system is the longitudinal control of a vehicle, where the control technology is used to achieve constant cruise speed, vehicle-to-vehicle distance control, identification and tracking of the curve of the vehicle ahead, automatic braking, traffic-flow, and string stability, environmental performance and fuel economy, and other security or comfort functions. Per the working conditions, ACC systems can be divided into cruise mode, following mode, and overtaking mode (GOÑI-ROS et al., 2019). The quality of the longitudinal control effect has a direct impact on the safety and comfort of the system.

Since the ADAS is designed to work with a co-existing human driver, driver acceptance is particularly important. For example, vehicle acceleration control algorithm and spacing policy help determine user acceptance in real operation conditions (MOON; YI, 2008):

- Too large or too small vehicle acceleration can cause the driver to feel uncomfortable or make the vehicle unsafe. Same for systems that constantly accelerate and brake needlessly;
- Too large spacing between vehicles could induce other vehicles to suddenly change lanes, but too small spacing can cause the driver to feel unsafe in heavy traffic conditions.

Ultimately, ACC systems must be useful to the driver and the system's operation characteristics need to be similar enough to the normal driving operation of a human driver. Studies have shown that the ACC considerably helps to reduce the driver's workload (WU et al., 2020).

There are several different ways to execute an ACC system. One interesting strategy is to use two separate control loops (SHAKOURI; CZECZOT; ORDYS, 2015):

1. An outer loop capable of performing obstacles distance calculations and defining a changing cruise speed reference, depending on the user's desired speed, the vehicle speed, and radar/environment data;
2. An inner loop, capable of processing the cruise speed reference generated by the outer loop and effectively controlling the vehicle speed by acting on the engine throttle or the braking system.

Several different control theories have been applied in ACC systems, such as PID feedback control (HE et al., 2019), fuzzy logic (BENALIE et al., 2009), control barriers (AMES;

GRIZZLE; TABUADA, 2014), sliding mode (GANJI et al., 2014), Model Predictive Control (MPC) (BRUGNOLLI, 2018) and MFC (NOVEL, 2018; D'ANDRÉA-NOVEL et al., 2016; MENHOUR et al., 2015; MENHOUR et al., 2013).

2.2 European New Car Assessment Programme

The Euro NCAP was established in 1997 and aims to provide consumers with a safety performance assessment for cars, especially popular consumer cars. Because of its rigorous crash testing, it became an important driver safety improvement to new cars and nowadays operates with an overall rating system for the most important aspects of vehicle safety using a single-star rating (RATINGEN et al., 2016).

Over the last decade, Euro NCAP became synonymous with crash testing and safety ratings and, in the same period, a reduction of roughly a quarter of the total road death toll in EU-28 has been seen despite significant growth in road traffic volumes (JOST et al., 2010).

Studies from early 2000 indicate that the Euro NCAP consumer tests are capable of predicting the outcome in real-world crashes, but only considering a part in all possible accident scenarios. In 2009, Euro NCAP added rear-end crash tests to the test protocol and since 2012 it has gradually revised the rating protocol for the better. Several studies also have presented a correlation between Euro NCAP results and injury risk based on real-world crashes (LIE; KULLGREN; TINGVALL, 2001; LIE; TINGVALL, 2002; KULLGREN; LIE; TINGVALL, 2010; KULLGREN et al., 2019). In general, vehicle crashworthiness has improved steadily over the years, with the proportion of serious injuries being greatly reduced when comparing car models launched 1980-1984 with those launched 2015-2018 (KULLGREN et al., 2019), and a big part of the reason is the implementation of safety tests such as the Euro NCAP during development.

When considering the most important factors of an ADAS system, safety must come first and the Euro NCAP tests are extremely useful to verify the safety of an ACC system. It is not uncommon for commercial vehicle simulators, such as Mechanical Simulation's CarSim software (CARSIM. . . , 2022), to already have Euro NCAP safety test scenarios implemented.

Among the several different Euro NCAP tests, some that can be considered particularly important for the development of ADAS system such as ACC and Autonomous Emergency Braking (AEB) are:

- Car-to-Car Rear brake test, which puts the Vehicle Under Test (VUT) in a predetermined distance and speed from a stopped vehicle.
- Car-to-Car Cut-In test, in which the VUT must deal with another vehicle suddenly entering its front at a slower speed.
- Car-to-Car Cut-Out test, where the VUT is following a vehicle that suddenly changes lanes because of a stopped car on the road.

All these test scenarios analyze if the VUT can safely avoid an accident by stopping or slowing down the vehicle before an imminent crash. They can test the safety measures of an ACC system in an emergency.

2.3 Electronics Braking System and Vehicle Modeling

Several active vehicle safety systems can considerably reduce the driver's workload and the probability of accidents when used. One of the most important of these safety systems is the brake system. In recent years, the automotive industry has been looking for new solutions for Brake-by-Wire (BBW) systems (SCHENK; WELLS; MILLER, 1995; JONNER et al., 1996).

2.3.1 Brake-by-Wire Technology

It is possible to define BBW technology as the ability to control brakes through electrical means. BBW systems are of great interest because they can produce a faster response and obtain better performance than conventional brake systems (HAN; XIONG; YU, 2019).

Different solutions for these systems have been proposed and the Electro-Hydraulic Brake (EHB) solution, based on a hydraulic system driven by a piston pump controlled by an electronic unit or an electric motor, is one of the most used. The most used EHB system nowadays is driven by a piston pump controlled by an electronic unit (D'ALFIO; MORGANDO; SORNIOTTI, 2006; SORNIOTTI; REPICI, 2005; AOKI et al., 2007; MILANÉS et al., 2010). Less commonly used, the electric motor-driven EHB adopts an electromechanical actuator instead of a piston pump, high-pressure accumulator, and various controlled solenoid valves(LEIBER; KÖGLSPERGER; UNTERFRAUNER, 2012; XIONG et al., 2014; WANG et al., 2013; YU et al., 2016).

2.3.2 Electro-Hydraulic Brake Systems

Vehicle manufacturers have proposed in the last 50 years several possible solutions for brake systems capable of overcoming the drawbacks of conventional braking systems based on vacuum boosters. One of the most common disadvantages of these conventional braking systems is the slow response time, which causes a noticeable delay between the application of force on the brake pedal and the useful braking effect caused by the increase in pressure on the brake caliper, especially for emergency conditions (SORNIOTTI, 2004).

One possible solution is the introduction of hydraulic boosters that can reduce response times and, in some cases (HIROTA et al., 2004), can differentiate the pedal feeling according to the braking maneuver. For progressive brake activation, the boost ratio is kept low. While for an emergency braking maneuver, the boost ratio is increased.

EHB systems allow for complete separation between the brake pedal and the wheels' brake calipers, solving problems related to the pedal feeling dependent on the physical conditions of the brake systems (such as temperature and wear). In these systems, the brake calipers remain hydraulically actuated and can be reconnected to the pedal unit in the event of a failure (D'ALFIO; MORGANDO; SORNIOTTI, 2006). In addition, the EHB also makes it possible to completely differentiate the pressure levels in each of the four tires according to the requirements of vehicle dynamics, without the limitations and inconveniences commonly associated with conventional hydraulic units for Electronic Stability Control (ESC).

The characteristics of EHB systems directly influence various aspects of the vehicle, such as fuel consumption, emissions, brake safety, and driving comfort (YANG; LI; ZHANG, 2018). In particular, hybrid vehicles with wheels powered by electric motors can benefit from EHB systems since they allow for efficient and safe energy recovery during braking. They are the only way to have a regenerative brake without interfering with the dissipative brake system, which remains indispensable.

The EHB consists of two main subsystems:

1. Pedal unit, which performs the transfer of the desired force feedback to the driver;
2. Hydraulic unit, necessary to generate the desired pressure in the brake caliper.

The pedal unit is formed by a system called Pedal Force Emulator (PFE), formed by elastic bodies connected in series with a conventional Tandem Master Cylinder (TMC).

The elastic bodies can be coil springs, air springs, rubber elements, or a combination of these and are used to generate the desired feel on the brake pedal.

The TMC closes the output ports during normal system operation and, in this condition, determines the pedal feeling only for the idle travel corresponding to the closure of the orifices that put the chambers in communication with the oil tank. In the event of a failure in the EHB system, detected by the control algorithm, failsafe valves located in the hydraulic unit put the TMC chambers in communication with the wheel brake calipers.

This condition is similar to a conventional brake system without a booster. For this case, the final pedal travel is given by the combination of the deformation of the elastic elements of the PFE and the travel of the pistons inside the TMC, due to the volume displacement of the wheels' brake calipers. In some cases, an additional failsafe valve is placed between the PFE and TMC so that in the event of a failure of the EHB the pedal travel is not too consistent. A system of displacement sensors located in the pedal unit measures the travel of the brake pedal and sends it to the system's electronic control unit. With this information, the electronic control unit decides the pressure level for each brake caliper, which is generated by a hydraulic unit formed by an accumulator and some low-cost digital solenoid valves, similar to common ESC systems. The accumulator is fed by a radial piston pump, also similar to that of an ESC.

Two different valves for each wheel can communicate the hydraulic accumulator with the brake calipers when their pressure needs to be increased, or else communicate the brake calipers with the tank when their pressure needs to be reduced. Pressure control is performed by comparing the desired pressure level and the actual pressure level in the brake caliper, measured by a sensor located in the hydraulic unit that measures the pressure level at the exit of the hydraulic unit towards it. As the brake calipers and hydraulic units can be built by different manufacturers, the pressure sensors cannot be mounted directly to them.

The EHB hydraulic unit is characterized by a total number of six pressure sensors: one for each brake caliper, one on the accumulator, and one on the pedal unit (used mainly by the failsafe algorithm).

EHB hydraulic circuits can be non-isolated or isolated. The non-isolated scheme consists of a single hydraulic circuit for the entire system, which has simplicity as its main advantage. However, as the main disadvantage, a failure in the accumulator causes a dangerous drop in the performance of the brake system, even if the failsafe valves are activated, as the nitrogen from the accumulator can migrate to the other components of

the brake system. This causes a reduction in the brake fluid bulk modulus, making it impossible to build up pressure in the brake caliper.

On the other hand, an isolated circuit has a dedicated hydraulic piston that gives origin to the separation between the hydraulic circuits. A pre-charged spring keeps the piston in contact with the proper endstop when the brake pedal is not pressed. If a failure occurs with the hydraulic accumulator, the circuit connecting the TMC with the brake caliper (through the failsafe valve) is not involved.

A reliable alternative to isolated circuits is to use non-isolated circuits with specific accumulators equipped, for example, with multi-layer or metallic membranes between the gas and the brake fluid. This mechanism makes it possible to limit brake system failures due to gas permeability.

There is also another valve in the EHB system, different from the pressure control and failsafe valves, that allows the hydraulic connection between the two outputs of the hydraulic unit with the brake calipers of the same axle. This guarantees a perfect match between the pressure levels in these calipers in case of braking maneuvers that do not require an asymmetrical distribution of pressure levels in the brake calipers. Without this connection, an error in the measurement of the pressure sensors (caused by temperature variation, for example) could cause a lateral slip of the vehicle, or during a braking maneuver with a null steering wheel angle and no lateral disturbances on the vehicle motion (HAC, 2005).

2.3.3 Simplified Vehicle Reverse Longitudinal Dynamics Equations

For an ACC system to work properly, a host controller defines the desired vehicle acceleration or, less commonly, speed as a control command which must be converted into desired throttle opening and brake pressure by the vehicle reverse longitudinal dynamic model. After this conversion, the new commands are transmitted to the vehicle longitudinal dynamics model in order to control the vehicle acceleration, deceleration, or uniform motion, whichever is necessary to achieve the desired car ACC (LIU; CHE; CAO, 2019).

From a vehicle dynamics standpoint, acceleration and braking are isolated movements (WU et al., 2020). During braking, first, the acceleration is halted (by releasing the accelerator pedal, for example) and then the vehicle is left to decelerate using engine drag, wind resistance, rolling resistance, and any other mechanical drag intrinsic to the vehicle

construction. If those are not capable of achieving the desired deceleration, then the brake system must be engaged to increase the vehicle deceleration (for example, depressing the brake pedal).

When taking into consideration the driving comfort and the reliability of the corresponding parts of the vehicle, is desirable to minimize any frequent switching between acceleration and braking control. A threshold acceleration $a_{threshold}$, usually taken from experience, can be defined to perform this switch in a controlled manner. Defining the expected acceleration of the vehicle as a_{des}, when $a_{des} \geq a_{threshold}$, the acceleration control is engaged. Otherwise, when $a_{des} \leq a_{threshold}$, braking control is engaged.

If the acceleration control is engaged, the desired acceleration is used to calculate the desired engine torque, which can be converted into desired throttle opening through the engine mapping. Not considering the conversion quality of the rotating parts, the vehicle longitudinal dynamics model is given by (2.1) and (2.2) (WU et al., 2020).

$$ma_{des} = F_t - F_{xb} - \sum F\left(v_{spd}\right) \tag{2.1}$$

$$\sum F\left(v_{spd}\right) = \frac{1}{2} C_D A \rho v_{spd}^2 + mgf \tag{2.2}$$

where a_{des} is the desired acceleration, m is the vehicle mass, F_t is the driving force, F_{xb} is the braking force, $\sum F\left(v_{spd}\right)$ is the sum of the resistances, C_D is the air resistance coefficient, A is the frontal area, ρ is the air density, v_{spd} is the car speed, g is the gravitational acceleration, and f is the rolling resistance coefficient.

Regardless of the elastic deformation of the transmission system, the driving force can be calculated as (2.3) (WU et al., 2020).

$$F_t = \frac{\eta \tau_{conv}\left(\omega_t / \omega_e\right) i_g i_0}{r} T_e = K_{drv} T_e \tag{2.3}$$

where η is the mechanical efficiency, T_e is the engine torque, ω_t is the torque converter turbine speed, ω_e is the engine speed, i_g is the transmission gear ratio, i_0 is the main gear ratio, $\tau_{conv}\left(\omega_t / \omega_e\right)$ is a torque converter characteristic function, r is the wheel rolling radius.

During acceleration, $F_{xb} = 0$ and the expected engine torque output T_{des} can be calculated using the transmission gear ration and speed ratio, allowing us to write (2.4).

$$T_{des} = \frac{ma_{des} + \sum F\left(v_{spd}\right)}{K_{drv}} \tag{2.4}$$

Based on the engine mapping, it is easy to get the throttle opening required, β_{des}, to output different torques at different speeds, expressed as (2.5).

$$\beta_{des} = f\left(T_{des}, \omega_e\right) \tag{2.5}$$

On the other hand, if the braking control is engaged the vehicle must comply with the desired deceleration. From this deceleration, the necessary braking force can be calculated and used to obtain the necessary braking pressure, in the brake calipers, through the braking reverse model (ZHENG; ZHAO, 2016). During braking, the engine output torque is terminated, $T_e = 0$, and equation (2.3) shows us that in this case $F_t = 0$. The vehicle longitudinal force can be shown as (2.6).

$$ma_{des} = -F_{xb} - \sum F\left(v_{spd}\right) \tag{2.6}$$

It is possible to approximate the braking force, F_{bdes}, and braking pressure, P_{des}, with a linear relationship according to (2.7).

$$F_{bdes} = K_{brk} P_{des} \tag{2.7}$$

where K_{brk} is a constant.

From (2.6) and (2.7), the necessary braking pressure can be calculated as (2.8).

$$P_{des} = \frac{\left|-ma_{des} - \frac{1}{2} C_D A \rho v_{spd}^2 - mgf\right|}{K_{brk}} \tag{2.8}$$

Therefore, it is possible to define the necessary engine throttle opening or the necessary braking pressure as (2.5) and (2.8), respectively, to achieve the desired acceleration commanded by the host controller.

2.4 Intelligent PID Control (iPID)

The iPID control is a form of MFC that uses an ultra-local model of the plant to expand the capabilities of the classic PID controller. In theory, it enables classical PID

controllers to adapt themselves to changes in the plant behavior or environment. This MFC is also interesting for being able to produce an asymptotically stable trajectory tracking even for some nonlinear systems (DELALEAU, 2014), without any knowledge about the plant mathematical model.

2.4.1 Ultra-Local Model

Originally introduced by (FLIESS; JOIN, 2008), the MFC in question is based on the idea of replacing the complex and unknown model of a plant with an ultra-local model. Assuming a Single-Input Single-Output (SISO) control system, with control variable u and output variable y, we can write its ultra-local model as (2.9).

$$y^{(\nu)} = F + \alpha u \tag{2.9}$$

where $y^{(\nu)}$ is the derivative of order $\nu \geq 1$ of y, α is a scaling constant and F represents unknown parts of the plant and possible disturbances. The value of ν must be an integer and chosen by the designer, and usually assumes low values. The constant $\alpha \in \mathbb{R}$ has the objective of making $y^{(\nu)}$ and αu have the same magnitude and need to be defined by the designer as part of the final controller tuning.

From this ultra-local model, we can close a control loop with an intelligent controller given by (2.10) (FLIESS; JOIN, 2013).

$$u = -\frac{F - y_r^{(\nu)} + \mathfrak{C}(e)}{\alpha} \tag{2.10}$$

where y_r is the reference path of the output, $y_r^{(\nu)}$ is the derivative of order $\nu \geq 1$ of y_r, $e = y_r - y$ is the tracking error, and $\mathfrak{C}(e)$ is a function of the error. By defining $\mathfrak{C}(e)$ as the PID controller function, we can write the iPID.

For $\nu = 2$, we can write the ultra-local model and the intelligent controller, respectively, as (2.11) and (2.12). The model-free intelligent controller block diagram is presented in Figure 1.

$$\ddot{y} = F + \alpha u \tag{2.11}$$

$$u = -\frac{F - \ddot{y}_r + \mathfrak{C}(e)}{\alpha} \tag{2.12}$$

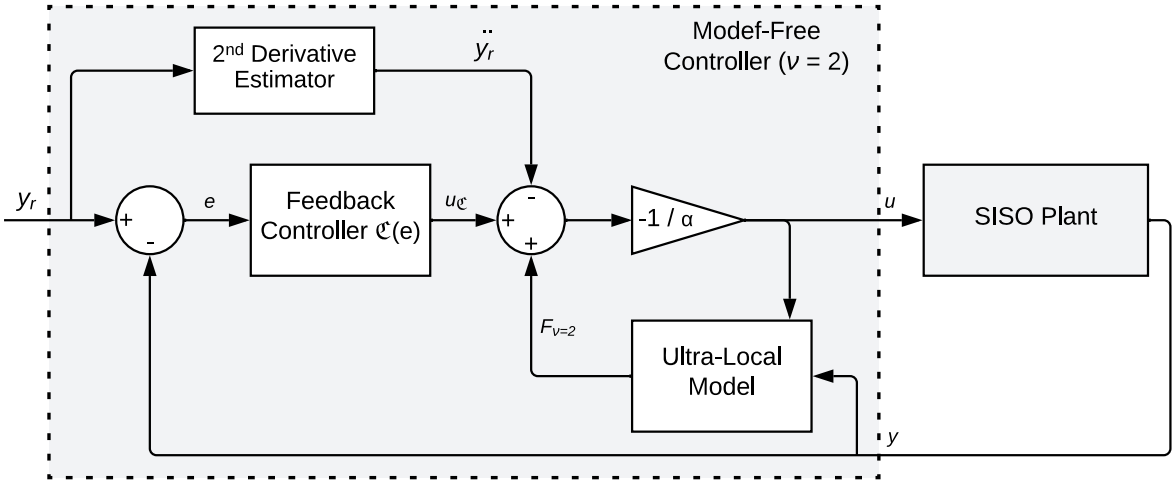


Figure 1: Block diagram for the model-free intelligent controller, for $\nu = 2$.

Source: Author.

Making $\mathfrak{C}(e)$ in (2.12) a PID controller, as in (2.13), we obtain in (2.14) an iPID.

$$\mathfrak{C}(e) = K_P e + K_I \int e + K_D \dot{e} \tag{2.13}$$

$$u = -\frac{F - \ddot{y}_r + K_P e + K_I \int e + K_D \dot{e}}{\alpha} \tag{2.14}$$

Using $K_I = 0$ allows us to write an intelligent Proportional Derivative (iPD) controller, as in (2.15).

$$u = -\frac{F - \ddot{y}_r + K_P e + K_D \dot{e}}{\alpha} \tag{2.15}$$

And for $\nu = 1$, we can write the ultra-local model and the intelligent controller, respectively, as (2.16) and (2.17). The model-free intelligent controller block diagram is presented in Figure 2.

$$\dot{y} = F + \alpha u \tag{2.16}$$

$$u = -\frac{F - \dot{y}_r + \mathfrak{C}(e)}{\alpha} \tag{2.17}$$

In this case, $\mathfrak{C}(e)$ in (2.17) can be a PI controller, as in (2.18). We obtain in (2.19) an intelligent Proportional Integral (iPI).

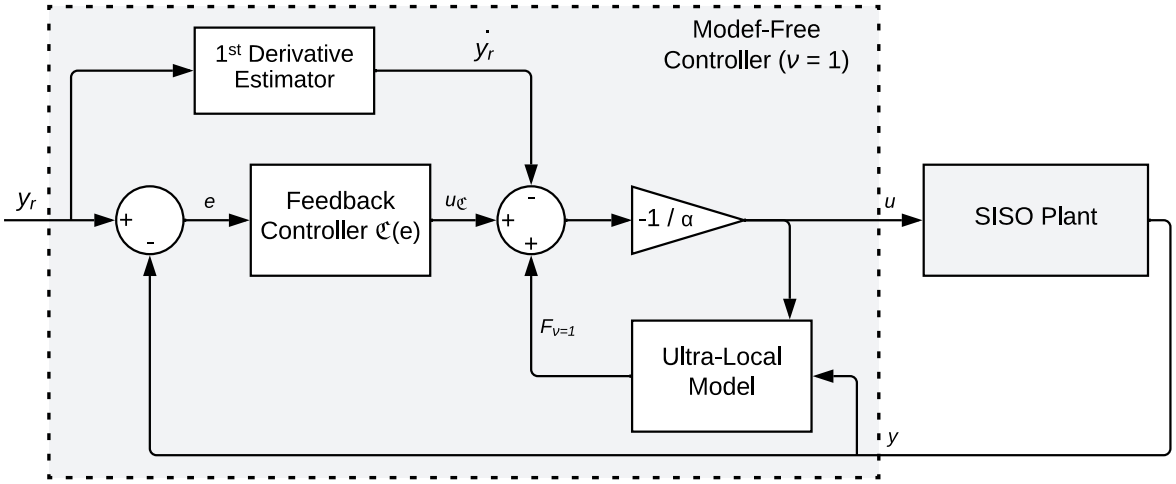


Figure 2: Block diagram for the model-free intelligent controller, for $\nu = 1$.

Source: Author.

$$\mathfrak{C}(e) = K_P e + K_I \int e \tag{2.18}$$

$$u = -\frac{F - \dot{y}_r + K_P e + K_I \int e}{\alpha} \tag{2.19}$$

In many cases we can use $K_I = 0$ in (2.19) , obtaining in (2.20) an iP controller.

$$u = -\frac{F - \dot{y}_r + K_P e}{\alpha} \tag{2.20}$$

A method to adjust the parameters of an iP controller is presented in (POLACK; DELPRAT; NOVEL, 2019) and essentially consists of three steps:

1. The gain α must be chosen as a value high enough so that the controller output is almost zero and K_P must be set to zero;

2. Run the experiment or simulation and reduce the value of α until the average value of $|\dot{y} - \dot{y}_r|$ is as small as possible.

3. If the output oscillates around the reference, α needs to be increased. With α set, progressively increase K_P until tracking error $|y - y_r|$ is low, disturbances are rejected, and the system has a desired dynamic response.

An interesting aspect of iP controllers is that they can achieve equivalent performance to a PI controller without some of the implementation problems presented by them,

such as anti-windup mechanisms (FLIESS; JOIN, 2013). Fliess and Join (2013) also demonstrates that for an implementation sampled with time T_s, we can make a direct association between the gains of an iP controller given by (2.20) and a PI controller given by (2.18), according to (2.21) and (2.22).

$$K_{P_{PI}} = 1/(\alpha_{iP} T_s) \tag{2.21}$$

$$K_{I_{PI}} = K_{P_{iP}}/(\alpha_{iP} T_s) \tag{2.22}$$

where $K_{P_{PI}}$ and $K_{I_{PI}}$ are discrete PI gains and α_{iP} and $K_{P_{iP}}$ are discrete iP gains. It is important to note that this direct association only makes sense for sampled systems and that it depends on the controller's implementation. In this work, this equivalence will be used for comparison purposes between the two controllers. Similar gain equivalences between other forms of PID and iPID controllers are presented in (FLIESS; JOIN, 2013).

2.4.2 Estimation of F

The parameter F needs to be constantly updated and can be estimated by a piecewise constant function using operational calculus (FLIESS; JOIN, 2013). Assuming F constant for a short period of time and $\nu = 1$, we can write the Laplace transform of the ultra-local model, of (2.16), as (2.23).

$$sY(s) - y(0) = \frac{F_{\nu=1}}{s} + \alpha U(s) \tag{2.23}$$

where $y(0)$ is the initial condition of the system. We can eliminate $y(0)$ by differentiating (2.23) as a function of s, obtaining (2.24).

$$Y(s) + s\frac{\mathrm{d}\left(\mathrm{Y(s)}\right)}{\mathrm{d}s} = -\frac{F_{\nu=1}}{s^2} + \alpha\frac{\mathrm{d}\left(\mathrm{U(s)}\right)}{\mathrm{d}s} \tag{2.24}$$

Multiplying (2.24) by $1/s^2$ we can eliminate all time derivatives and integrate all components at least once, effectively filtering the system with low-pass filters and attenuating possible measurement noise, obtaining (2.25).

$$\frac{1}{s^2}Y(s) + \frac{1}{s}\frac{\mathrm{d}\left(\mathrm{Y(s)}\right)}{\mathrm{d}s} = -\frac{F_{\nu=1}}{s^4} + \frac{\alpha}{s^2}\frac{\mathrm{d}\left(\mathrm{U(s)}\right)}{\mathrm{d}s} \tag{2.25}$$

Using the inverse Laplace transform and operational calculus rules, we can apply

(2.26) and (2.27) in (2.25) to obtain (2.28).

$$\mathcal{L}^{-1}\left(\frac{c}{s^a}\right) = c\frac{t^{a-1}}{(a-1)!}, a \geq 1, c \in \mathbb{C}, t > 0 \tag{2.26}$$

$$\mathcal{L}^{-1}\left(\frac{c}{s^a}\frac{\mathrm{d^n\,(Y)}}{\mathrm{d}s^n}\right) = c\frac{(-1)^n}{(a-1)!}\int_0^t (t-\tau)^{a-1}\,\tau^n y\,(\tau)\,d\tau, a \geq 1, t > 0 \tag{2.27}$$

$$\int_0^t (t-\tau)y(\tau)d\tau - \int_0^t \tau y(\tau)d\tau = -F_{\nu=1}\frac{t^3}{6} - \alpha\int_0^t (t-\tau)\tau u(\tau)d\tau \tag{2.28}$$

Isolating the parameter $F_{\nu=1}$ in (2.28) we get its the temporal estimation in (2.29).

$$F_{\nu=1} = -\frac{6}{t^3}\int_0^t \left((t-2\tau)\,y\,(\tau) + \alpha\tau\,(t-\tau)\,u\,(\tau)\right)d\tau \tag{2.29}$$

where t is small and represents the integration interval, which must be defined according to the sampling period and noise intensity present in the system. Increasing the t allows the iPID to reject noise better but worsens the controller's dynamic response.

Considering the same assumptions as before and $\nu = 2$, we can write the Laplace transform of the ultra-local model, of (2.11), as (2.30).

$$s^2Y(s) - sy(0) - \dot{y}(0) = \frac{F_{\nu=2}}{s} + \alpha U(s) \tag{2.30}$$

where $y(0)$ and $\dot{y}(0)$ are the initials conditions of the system. We can eliminate $\dot{y}(0)$ by differentiating (2.30) as a function of s, obtaining (2.31).

$$2sY(s) + s^2\frac{\mathrm{d\,(Y(s))}}{\mathrm{d}s} - y(0) = -\frac{F_{\nu=2}}{s^2} + \alpha\frac{\mathrm{d\,(U(s))}}{\mathrm{d}s} \tag{2.31}$$

Differentiating (2.31) as a function of s one more time allows us to eliminate $y(0)$ and obtain (2.32).

$$2Y(s) + 4s\frac{\mathrm{d\,(Y(s))}}{\mathrm{d}s} + s^2\frac{\mathrm{d^2\,(Y(s))}}{\mathrm{d}s^2} = \frac{2F_{\nu=2}}{s^3} + \alpha\frac{\mathrm{d^2\,(U(s))}}{\mathrm{d}s^2} \tag{2.32}$$

Multiplying (2.31) by $1/s^3$ we can eliminate all time derivatives and integrate all components at least once, obtaining (2.33).

$$\frac{2}{s^3}Y(s) + \frac{4}{s^2}\frac{\mathrm{d\,(Y(s))}}{\mathrm{d}s} + \frac{1}{s}\frac{\mathrm{d^2\,(Y(s))}}{\mathrm{d}s^2} = \frac{2F_{\nu=2}}{s^6} + \frac{\alpha}{s^3}\frac{\mathrm{d^2\,(U(s))}}{\mathrm{d}s^2} \tag{2.33}$$

Using the inverse Laplace transform and operational calculus rules, we can apply (2.26) and (2.27) in (2.33) to obtain (2.34).

$$\int_0^t (t-\tau)^2 y(\tau)d\tau - 4\int_0^t (t-\tau)\tau y(\tau)d\tau + \int_0^t \tau^2 y(\tau)d\tau = F_{\nu=2}\frac{t^5}{60} - \frac{\alpha}{2}\int_0^t (t-\tau)^2\tau^2 u(\tau)d\tau \quad (2.34)$$

Isolating the parameter $F_{\nu=2}$ in (2.34) we get its the temporal estimation in (2.35).

$$F_{\nu=2} = \frac{60}{t^5}\int_0^t \left(\left(\tau^2 - 4(t-\tau)\tau + (t-\tau)^2\right)y(\tau) + \frac{\alpha}{2}\tau^2(t-\tau)^2 u(\tau)\right)d\tau \quad (2.35)$$

2.4.3 Discrete Implementation of the F Estimator

In order to use the iPID controller family in simulation environments or embedded systems, we need to have a discrete implementation of the controller in (2.19). For $\nu = 1$, the iPI controller is given by (2.36).

$$u(k) = -\frac{\left(\hat{F}_{\nu=1}(k) - \hat{\dot{y}}_r(k) + K_P e(k) + K_I \sum e(k)\right)}{\hat{\alpha}} \quad (2.36)$$

where k is the current sample, $e(k) = y_r(k) - y(k)$ is the tracking error, $u(k)$ is the system control output, $\hat{F}_{\nu=1}(k)$ is an estimate of $F_{\nu=1}$, $\hat{\dot{y}}_r(k)$ is an estimate of $\dot{y}_r$, and $\hat{\alpha}$ is a choice of α. The value chosen for $\hat{\alpha}$ admits errors but is indicated to be of the same order of magnitude as the actual value.

The estimation of $F_{\nu=1}$ is complex and must use numerical integration methods. For a fixed integration interval of nT_s, where T_s is the sampling time and n is the number of sampling intervals, we can write (2.29) for a fixed interval as (2.37).

$$\hat{F}_{\nu=1}(k) = -\frac{6}{(nT_s)^3}\int_0^{nT_s} \left((nT_s - 2\tau)\, y(\tau) + \alpha\tau\,(nT_s - \tau)\, u(\tau)\right)d\tau \quad (2.37)$$

According to Polack, Delprat and Novel (2019) to obtain a perfect steady-state estimation, its numerical integration must use a polynomial of at least second order. Because of this, the Simpson's 1/3 Rule was chosen instead of the more traditional Trapezoidal Rule for the numerical integration. The Simpson's 1/3 Rule composite form for n equal intervals (using $n+1$ samples), with n integer and even, is given by (2.38).

$$\int_0^{nT_s} f(t)\,dt \approx \frac{(nT_s)}{3n}\left(f(0) + 4\sum_{j=1}^{\frac{n}{2}} f\left((2j-1)\,T_s\right) + 2\sum_{j=1}^{\frac{n}{2}-1} f\left(2jT_s\right) + f(nT_s)\right) \quad (2.38)$$

It is possible to rewrite (2.38) as (2.39).

$$\int_0^{nT_s} f(t)\,dt \approx \frac{(nT_s)}{3n}\left(f(0) + \sum_{j=1}^{n-1}\left(3-(-1)^j\right) f\left(jT_s\right) + f(nT_s)\right) \quad (2.39)$$

Applying (2.39) to the estimator given by (2.37), we obtain an estimator given by (2.40).

$$\hat{F}_{\nu=1}(nT_s) = -\frac{2}{n^3T_s}\Bigg(ny(0)+ \sum_{j=1}^{n-1}\left(3-(-1)^j\right)\left((n-2j)\,y(jT_s) + \alpha T_s j\,(n-j)\,u(jT_s)\right) - ny(nT_s)\Bigg) \quad (2.40)$$

The estimator in (2.40) calculates the estimation for the time nT_s using data from an initial sample at time 0 to the last sample at time nT_s. For digitally controlled systems, it is more interesting to work directly with samples instead of sample time and to integrate regarding the past samples, where the sample k (sample time kT_s) is the current sample and the integration interval is from sample k to $k-n$ (sample time kT_s to $(k-n)T_s$). The estimator (2.40) can be modified to work with past samples by delaying all its sample times by nT_s and representing the sample times directly by their sample number, obtaining (2.41).

$$\hat{F}_{\nu=1}(k) = -\frac{2}{n^3T_s}\Bigg(ny(k-n)+ \sum_{j=1}^{n-1}\left(3-(-1)^j\right)\Big((n-2j)\,y(k-n-j)+ \alpha T_s j\,(n-j)\,u(k-n-j)\Big) - ny(k)\Bigg) \quad (2.41)$$

Analyzing the estimator of (2.41), there are two very interesting characteristics:

1. When $j=\frac{n}{2}$, sample $k-n-j=k-\frac{n}{2}$, the coefficient $(n-2j)$ for $y\left(k-\frac{n}{2}\right)$ becomes 0. Therefore, at sample $k-\frac{n}{2}$ we only need the sample from $u\left(k-\frac{n}{2}\right)$.

2. There is a symmetry of sorts around the coefficients for y and u around time $j = \frac{n}{2}$. Taking advantage of this symmetry, it is possible to halve the number of coefficient calculations necessary for the estimator. For $1 < j < \frac{n}{2}$ we can write:

   (a) $(n-2j)\,y\,(k-n-j) = -\,(n-2j)\,y\,(k-j)$;

   (b) $\alpha T_s j\,(n-j)\,u\,(k-n-j) = \alpha T_s j\,(n-j)\,u\,(k-j)$.

With these two characteristics, it is possible to rewrite (2.41) as (2.42).

$$\hat{F}_{\nu=1}(k) = -\frac{2}{n^3 T_s}\Bigg(n\left(y\left(k-n\right)-y\left(k\right)\right)+\left(3-(-1)^{\frac{n}{2}}\right)\frac{\alpha T_s n^2}{4}u\left(k-\frac{n}{2}\right)+ \sum_{j=1}^{\frac{n-2}{2}}\left(3-(-1)^{j}\right)\Big(\left(n-2j\right)\left(y\left(k-n-j\right)-y\left(k-j\right)\right)+ \alpha T_s k\left(n-j\right)\left(u\left(k-n-j\right)+u\left(k-j\right)\right)\Big)\Bigg) \quad (2.42)$$

The estimator presented in (2.42) is flexible and efficient for digital systems. An interesting particular form for it occurs when $n = 2$ and it becomes the smallest possible estimator for $\hat{F}_{\nu=1}$, presented in (2.43).

$$\hat{F}_{\nu=1}(k) = -\frac{1}{2T_s}\left(-y\left(k\right)+\alpha T_s u\left(k-1\right)+y\left(k-2\right)\right) \quad (2.43)$$

On the other hand, for $\nu = 2$, the discrete version of the iPID controller (2.14) is given by (2.44).

$$u(k) = -\frac{\left(\hat{F}_{\nu=2}(k) - \hat{\ddot{y}}_r(k) + K_P e(k) + K_I \sum e(k) + K_D \dot{e}(k)\right)}{\hat{\alpha}} \quad (2.44)$$

where $\dot{e}(k)$ is the first derivative of the tracking error, $\hat{F}_{\nu=2}(k)$ is an estimate of $F_{\nu=2}$, and $\hat{\ddot{y}}_r(k)$ is an estimate of $\ddot{y}_r$.

The estimation of $F_{\nu=2}$ can be discretized in the same fashion as the estimation of $F_{\nu=1}$. For a fixed integration interval of nT_s, we can write (2.35) as (2.45).

$$\hat{F}_{\nu=2}(k) = \frac{60}{\left(nT_s\right)^5} \int_0^{nT_s}\left(\left(\tau^2 - 4\left(nT_s-\tau\right)\tau+\left(nT_s-\tau\right)^2\right)y\left(\tau\right)+\frac{\alpha}{2}\tau^2\left(nT_s-\tau\right)^2 u\left(\tau\right)\right)d\tau \quad (2.45)$$

Applying (2.39) to the estimator given by (2.45), we obtain an estimator given by (2.46).

$$\hat{F}_{\nu=2}(nT_s) = \frac{20}{n^5 T_s^2}\Bigg(n^2 y(0) + \sum_{k=1}^{n-1}\left(3-(-1)^k\right)\Bigg(\left(k^2-4k\left(n-k\right)+\left(n-k\right)^2\right)y(kT_s) + \frac{\alpha T_s}{2}k^2\left(n-k\right)^2 u(kT_s)\Bigg) + n^2 y(nT_s)\Bigg) \quad (2.46)$$

Changing the estimator (2.46) time reference by delaying all its samples time by nT_s and writing it using samples instead of samples time, we obtain (2.47).

$$\hat{F}_{\nu=2}\left(k\right) = \frac{20}{n^5 T_s^2}\Bigg(n^2 y\left(k-n\right) + \sum_{j=1}^{n-1}\left(3-(-1)^j\right)\Bigg(\left(j^2-4j\left(n-j\right)+\left(n-j\right)^2\right)y\left(k-n-j\right) + \frac{\alpha T_s}{2}j^2\left(n-j\right)^2 u\left(k-n-j\right)\Bigg) + n^2 y\left(k\right)\Bigg) \quad (2.47)$$

The estimator of (2.47) also presents a similar symmetry as the one from (2.42). For $1 < j < \frac{n}{2}$ we can write:

1. $\left(j^2-4j\left(n-j\right)+\left(n-j\right)^2\right)y\left(k-n-j\right) = \left(j^2-4j\left(n-j\right)+\left(n-j\right)^2\right)y\left(k-j\right)$;
2. $\alpha T_s j\left(n-j\right)u\left(k-n-j\right) = \alpha T_s j\left(n-j\right)u\left(k-j\right)$

Using these, it is possible to rewrite (2.47) as (2.48).

$$\hat{F}_{\nu=2}\left(k\right) = \frac{20}{n^5 T_s^2}\Bigg(n^2\left(y\left(k-n\right)+y\left(k\right)\right) + \frac{\left(3-(-1)^{\frac{n}{2}}\right)n^2}{2}\left(-y\left(k-\frac{n}{2}\right)+\frac{\alpha T_s n^2}{16}u\left(k-\frac{n}{2}\right)\right) + \sum_{j=1}^{\frac{n-2}{2}}\left(3-(-1)^j\right)\Bigg(\left(j^2-4j\left(n-j\right)+\left(n-j\right)^2\right)\left(y\left(k-n-j\right)+y\left(k-j\right)\right) + \frac{\alpha T_s}{2}j^2\left(n-j\right)^2\left(u\left(k-n-j\right)+u\left(k-j\right)\right)\Bigg)\Bigg) \quad (2.48)$$

The estimator presented in (2.48) is also flexible and efficient for digital systems. An interesting particular form for it occurs when $n = 2$ and it becomes the smallest possible estimator for $\hat{F}_{\nu=2}$, presented in (2.49).

$$\hat{F}_{\nu=2}(k) = -\frac{5}{2T_s^2}\left(y(k) - 2y(k-1) + y(k-2) + \frac{\alpha T_s}{2}u(k-1)\right) \tag{2.49}$$

The estimators presented in (2.42) and (2.48) are particularly interesting for digital implementations:

1. They are not dependent on the current control input, allowing them to be calculated before the control input;
2. They take the form of Infinite Impulse Response (IIR) filters with gains that can be calculated in advance. This makes calculating the estimators relatively simple and of low computational complexity;
3. The integration time can be modified just by changing the parameter n and recalculating all the gains of the generated IIR filters;
4. They do not have any kind of accumulator to calculate the integral, avoiding the accumulation of rounding errors.

2.4.4 Tracking Reference Derivative Estimation

To calculate the chosen MFC it is necessary to work with the ν order derivative of the tracking reference $y_r^{(\nu)}$. Calculating the derivative of a signal in real-time can causes several different issues, especially if the signal is noisy (KASAC; MAJETIC; BREZAK, 2018). There are several ways to mitigate these problems, including using the same concept of the F parameter estimator to estimate the derivative in real time (WANG; WANG, 2020).

For $\nu = 1$, if the trajectory reference y_r is generated internally and noise-free, we can estimate $\dot{y}_r$ at time i by a simple numerical derivative, given by (2.50).

$$\hat{\dot{y}}_r(k) = \frac{(y_r(k) - y_r(k-1))}{T_s} \tag{2.50}$$

If the trajectory reference is generated externally or noisy, a more interesting approach is to use an algebraic estimator for its derivative. Assuming $\dot{y}_r$ constant for a short period of time, we can write the estimative as (2.51) and its Laplace transform as (2.52).

$$\hat{\dot{y}}_r = sy_r(t) \tag{2.51}$$

$$\frac{\hat{\dot{y}}_r}{s} = sY_r(s) - y_r(0) \tag{2.52}$$

where $y_r(0)$ is the initial condition of the tracking reference. Differentiating (2.52) as a function of s and multiplying it by $1/s^2$ allow us to eliminate $y_r(0)$, eliminate the time derivative, and filter the signal by integrating it, obtaining (2.53).

$$-\frac{\hat{\dot{y}}_r}{s^4} = \frac{1}{s^2}Y_r(s) + \frac{1}{s}\frac{\mathrm{d}\left(\mathrm{Y_r(s)}\right)}{\mathrm{d}s} \tag{2.53}$$

Using the inverse Laplace transform and operational calculus rules, we can apply (2.26) and (2.27) in (2.53) to obtain (2.54).

$$-\hat{\dot{y}}_r\frac{t^3}{6} = \int_0^t (t-\tau)y_r(\tau)d\tau - \int_0^t \tau y_r(\tau)d\tau \tag{2.54}$$

Isolating $\hat{\dot{y}}_r$ we rewrite (2.54) as (2.55), the final estimator for $\dot{y}_r$.

$$\hat{\dot{y}}_r = -\frac{6}{t^3}\int_0^t (t-2\tau)\, y_r(\tau)\, d\tau \tag{2.55}$$

where t is small and represents the integration interval, which must be defined according to the sampling period and noise intensity present in the tracking reference signal.

Applying the same discretization steps to (2.55) as applied to (2.29), a discrete estimator for $\hat{\dot{y}}_r$ can be written as (2.56). This estimator is interesting for already providing some filtration to the reference tracking signal.

$$\hat{\dot{y}}_r(k) = -\frac{2}{n^3 T_s}\Bigg(n\left(y_r(k-n) - y_r(k)\right) + \sum_{j=1}^{\frac{n-2}{2}}\left(3-(-1)^j\right)(n-2j)\left(y_r(k-n-j) - y_r(k-j)\right)\Bigg) \tag{2.56}$$

On the other hand, for $\nu = 2$, if the trajectory reference y_r is generated internally and noise-free, we can estimate $\ddot{y}_r$ at time k by a second-order numerical derivative, given by (2.57).

$$\hat{\ddot{y}}_r(k) = \frac{y_r(k) - 2y_r(k-1) + y_r(k-2)}{{T_s}^2} \tag{2.57}$$

For an externally generated or noisy trajectory reference, an algebraic estimator is an more interesting approach to calculate the derivative. Assuming $\dddot{y}_r$ constant for a short period of time, we can write the estimative as (2.58) and its the Laplace transform as (2.59).

$$\hat{\ddot{y}}_r = s^2 y_r(t) \tag{2.58}$$

$$\frac{\hat{\ddot{y}}_r}{s} = s^2 Y_r(s) - s y_r(0) - \dot{y}_r(0) \tag{2.59}$$

where $y_r(0)$ and $\dot{y}_r(0)$ are the initials conditions of the system. To obtain (2.60) we can differentiate (2.59) as a function of s twice and multiply the result by $1/s^3$ to eliminate all initial conditions, time derivatives and integrate all components at least once, filtering the signal.

$$\frac{2\hat{\ddot{y}}_r}{s^6} = \frac{2}{s^3} Y_r(s) + \frac{4}{s^2}\frac{\mathrm{d}\left(\mathrm{Y_r(s)}\right)}{\mathrm{d}s} + \frac{1}{s}\frac{\mathrm{d}^2\left(\mathrm{Y_r(s)}\right)}{\mathrm{d}s^2} \tag{2.60}$$

Using the inverse Laplace transform and operational calculus rules, we can apply (2.26) and (2.27) in (2.60) to obtain (2.61).

$$\hat{\ddot{y}}_r \frac{t^5}{60} = \int_0^t (t-\tau)^2 y_r(\tau) d\tau - 4\int_0^t (t-\tau)\tau y_r(\tau) d\tau + \int_0^t \tau^2 y_r(\tau) d\tau \tag{2.61}$$

Isolating the parameter $\hat{\ddot{y}}_r$ in (2.61) we get the final for of the $\dddot{y}_r$ estimation as (2.62).

$$\hat{\ddot{y}}_r = \frac{60}{t^5}\int_0^t \left(\tau^2 - 4\left(t-\tau\right)\tau + \left(t-\tau\right)^2\right) y_r(\tau) d\tau \tag{2.62}$$

Applying the same discretization steps to (2.62) as applied to (2.35), a discrete estimator for $\hat{\ddot{y}}_r$ can be written as (2.63). As in (2.56), this estimator is interesting for already providing some filtration to the reference tracking signal.

$$\hat{\dot{y}}_r(k) = \frac{20}{n^5 T_s^2}\left(n^2\left(y_r(k-n) + y_r(k)\right) + \frac{\left(3-(-1)^{\frac{n}{2}}\right)n^2}{2}\left(-y_r\left(k-\frac{n}{2}\right)\right) + \sum_{j=1}^{\frac{n-2}{2}}\left(3-(-1)^j\right)\left(j^2 - 4j(n-j) + (n-j)^2\right)\left(y_r(k-n-j) + y_r(k-j)\right)\right) \quad (2.63)$$

The estimators presented in (2.63) and (2.63) are particularly interesting for digital implementations for mainly two reasons:

1. They take the form of Finite Impulse Response (FIR) filters with gains that can be calculated in advance. This makes calculating the estimators relatively simple and of low computational complexity;

2. The integration time can be modified just by changing the parameter n and recalculating all the gains of the generated FIR filters.

2.5 Intelligent PID Control with α Estimator (iPIDα)

One of the main reasons for the use of iPID MFC is the ease of use in different systems and its adaptability, using the F estimator to adapt itself to several models' uncertainties. Ideally, the controller would also be able to calculate a good $\hat{\alpha}$, an estimate of α, in real-time as the controller is executed, updating the initial parameter chosen by the controller designer. Not only this would reduce tuning efforts but would allow the MFC to adapt itself even better to different plants and models, creating an AMFC.

This work proposes a least square method based aproach to estimate an control optimum $\hat{\alpha}$ in real-time, that coupled with the iPID MFC controller creates the iPIDα AMFC controller.

2.5.1 Least Square α Estimator

Several different methods can be used to estimate an unknown parameter, with some of them already being used for the estimation of the α parameter. Wang et al. (2020) shows that having an α estimator can considerably improve the iPID response by using a recursive least squares to estimate α, while not using an estimator for the F parameter.

Similar to (WANG et al., 2020), the proposed estimator also works with least squares estimation but uses an entirely different approach and algorithm, working with the concept

of trying to find a $\hat{\alpha}$ that minimizes $\varepsilon^{(\nu)} = y_r^{(\nu)} - y^{(\nu)}$, based on the tuning method proposed by (POLACK; DELPRAT; NOVEL, 2019). It is very important to note that the proposed estimator does not try to estimate the real value of the system's ultra-local model α, but an α that minimizes the derivative of the control error. This means that the estimated α can only be considered optimum from a control standpoint.

Since its based on a least square method, the estimator is expected to have a good convergence in any situation, but no proof of convergency was created.

Starting from the ultra-local model of (2.9), we can write (2.64).

$$\varepsilon^{(\nu)} = y_r^{(\nu)} - y^{(\nu)} = y_r^{(\nu)} - (F + \hat{\alpha}u) \tag{2.64}$$

To apply the least square method we need to sum the squared errors of (2.64) for n different samples, as in (2.65).

$$\sum_{k=1}^{n}\left(\left(\varepsilon_k^{(\nu)}\right)^2\right) = \left(y_{r_1}^{(\nu)} - (F_1 + \hat{\alpha}u_1)\right)^2 + \left(y_{r_2}^{(\nu)} - (F_2 + \hat{\alpha}u_2)\right)^2 + \cdots + \left(y_{r_n}^{(\nu)} - (F_n + \hat{\alpha}u_n)\right)^2 \tag{2.65}$$

Expanding and reorganizing (2.65) we can write the sum of the squared errors as (2.66).

$$\sum_{k=1}^{n}\left(\left(\varepsilon_k^{(\nu)}\right)^2\right) = \sum_{k=1}^{n}\left(y_{r_k}^{(\nu)} - F_k\right)^2 + 2\hat{\alpha}\sum_{k=1}^{n}\left(F_k - y_{r_k}^{(\nu)}\right)u_k + \hat{\alpha}^2\sum_{k=1}^{n}u_k^2 \tag{2.66}$$

To minimize the squared error in function of $\hat{\alpha}$ we need to derivate (2.66) with respect to $\hat{\alpha}$ and equal it to zero, getting (2.67).

$$\frac{d\left(\sum_{k=1}^{n}\left(\left(\varepsilon_k^{(\nu)}\right)^2\right)\right)}{d\left(\hat{\alpha}\right)} = 2\sum_{k=1}^{n}\left(F_k - y_{r_k}^{(\nu)}\right)u_k + 2\hat{\alpha}\sum_{k=1}^{n}u_k^2 = 0 \tag{2.67}$$

Isolating $\hat{\alpha}$ in (2.67) and simplifying the equation allows us to write (2.68), the final least square estimator for $\hat{\alpha}$.

$$\hat{\alpha} = \frac{\sum_{k=1}^{n}\left(y_{r_k}^{(\nu)} - F_k\right)u_k}{\sum_{k=1}^{n}u_k^2} \tag{2.68}$$

It is also possible to write (2.68) using averages instead of sums. In this case, the

estimator is given by (2.69).

$$\hat{\alpha} = \frac{\overline{\left(y_r^{(\nu)} - F\right) u}}{\overline{u^2}} \tag{2.69}$$

The interesting aspect of the proposed estimator is the complete independence of the controller structure itself and the value of ν of the ultra-local model. This allows it to be used with any form of the MFC controller (iPID, iPD, iP, etc.).

2.5.2 Recursive Form of the α Estimator

The least-square estimator proposed is very simple to implement, but its dependency on accumulators makes it non-ideal to be used in long control runs or embedded environments with memory constraints. A recursive approach, where the new $\hat{\alpha}$ can be constantly updated without the need for accumulators, is essential to allow the estimator to be used in any situation. The estimator ideally must be in the form of (2.70).

$$\hat{\alpha}_n = \hat{\alpha}_{n-1} + \Delta_n \tag{2.70}$$

To make notation easier, the numerator of (2.68) can be written as (2.71). Allowing to write (2.68) as (2.72).

$$K_{\alpha_n} = \left(y_{r_n}^{(\nu)} - F_n\right) u_n \tag{2.71}$$

$$\hat{\alpha}_n = \frac{\sum_{k=1}^{n} K_{\alpha_k}}{\sum_{k=1}^{n} u_k^2} \tag{2.72}$$

From (2.72), it is possible to write (2.73).

$$\sum_{k=1}^{n} K_{\alpha_k} = \hat{\alpha}_n \sum_{k=1}^{n} u_k^2 \tag{2.73}$$

With (2.73) we can write (2.72) for $\hat{\alpha}_{n-1}$ as (2.74) and $\hat{\alpha}_n$ as (2.75).

$$\hat{\alpha}_{n-1} = \frac{\hat{\alpha}_{n-1} \sum_{k=1}^{n-1} K_{\alpha_k}}{\sum_{k=1}^{n-1} u_k^2} \tag{2.74}$$

$$\hat{\alpha}_n = \frac{K_{\alpha_n} + \hat{\alpha}_{n-1} \sum_{k=1}^{n-1} K_{\alpha_k}}{u_n^2 + \sum_{k=1}^{n-1} u_k^2} \tag{2.75}$$

From (2.70), (2.74) and (2.75), Δ_n can be written as (2.76).

$$\begin{gathered} \Delta_n = \hat{\alpha}_n - \hat{\alpha}_{n-1} \\ \Delta_n = \frac{K_{\alpha_n} + \hat{\alpha}_{n-1} \sum_{k=1}^{n-1} K_{\alpha_k}}{u_n^2 + \sum_{k=1}^{n-1} u_k^2} - \frac{\hat{\alpha}_{n-1} \sum_{k=1}^{n-1} K_{\alpha_k}}{\sum_{k=1}^{n-1} u_k^2} \\ \Delta_n = \frac{K_{\alpha_n} - \hat{\alpha}_{n-1} u_n^2}{\sum_{k=1}^{n} u_k^2} \end{gathered} \tag{2.76}$$

The last part of equation (2.76) allows us to write $\sum_{k=1}^{n-1} u_k^2$ as (2.77).

$$\sum_{k=1}^{n-1} u_k^2 = \frac{K_{\alpha_{n-1}} - \hat{\alpha}_{n-2} u_{n-1}^2}{\Delta_{n-1}} \tag{2.77}$$

Substituting (2.77) in the last part of (2.76) allow us to write (2.78).

$$\Delta_n = \Delta_{n-1} \frac{K_{\alpha_n} - \hat{\alpha}_{n-1} u_n^2}{K_{\alpha_{n-1}} - \hat{\alpha}_{n-2} u_{n-1}^2 + \Delta_{n-1} u_n^2} \tag{2.78}$$

Lastly, we can use (2.70) to write $\hat{\alpha}_{n-2}$ as (2.79).

$$\hat{\alpha}_{n-2} = \hat{\alpha}_{n-1} - \Delta_{n-1} \tag{2.79}$$

Replacing (2.79) in (2.78) and simplifying the equation arrives in (2.80), the final form for the update term of the estimator. With (2.70) and (2.80) we can finally write the full form of the recursive α estimator as (2.81).

$$\Delta_n = \Delta_{n-1} \frac{K_{\alpha_n} - \hat{\alpha}_{n-1} u_n^2}{K_{\alpha_{n-1}} - \hat{\alpha}_{n-1} u_{n-1}^2 + \Delta_{n-1} \left(u_n^2 + u_{n-1}^2\right)} \tag{2.80}$$

$$\hat{\alpha}_n = \hat{\alpha}_{n-1} + \Delta_{n-1} \frac{K_{\alpha_n} - \hat{\alpha}_{n-1} u_n^2}{K_{\alpha_{n-1}} - \hat{\alpha}_{n-1} u_{n-1}^2 + \Delta_{n-1} \left(u_n^2 + u_{n-1}^2\right)} \tag{2.81}$$

The α estimator in (2.81) is interesting for not using the accumulators necessary for the (2.68) estimator, being appropriate to be used in memory constraints systems or long control system runs.

2.5.3 Recursive Form of the α Estimator with a Forgetting Factor

The recursive form (2.81) of the proposed estimator (2.68) solves some of its issues but keeps a major one regarding the "data saturation" phenomenon (WANG et al., 2020). Since it's based on a least square method, a long estimating time can cause $\hat{\alpha}_n$ to lose its ability to regulate. When the estimator has a lot of old data accumulated, new data can end up being drowned in it and, essentially, ignored.

To improve the situation, a forgetting factor μ (varying from 0 to 1) can be added to the estimator. This forgetting factor μ allows the estimator to partially forget the accumulated data, causing new data to not be drowned in it.

A forgetting factor $\mu = 1$ causes the estimator to not forget anything and have the same behavior as before. The use of the forgetting factor can improve the estimator's performance but can cause instability with the estimation if it's too small. In general, values of μ in the range of 0.95 to 1 are considered good (WANG et al., 2020).

To add the forgetting factor to the estimator, we need to first rewrite (2.72) as (2.82), including the μ forgetting factor in the accumulators $\sum_{k=1}^{n-1} K_{\alpha_k}$ and $\sum_{k=1}^{n-1} u_k^2$.

$$\hat{\alpha}_n = \frac{K_{\alpha_n} + \mu \sum_{k=1}^{n-1} K_{\alpha_k}}{u_n^2 + \mu \sum_{k=1}^{n-1} u_k^2} \tag{2.82}$$

From (2.82), it is possible to write $\mu \sum_{k=1}^{n-1} K_{\alpha_k}$ as (2.83).

$$\mu \sum_{k=1}^{n-1} K_{\alpha_k} = \hat{\alpha}_n \left(u_n^2 + \mu \sum_{k=1}^{n-1} u_k^2 \right) - K_{\alpha_n} \tag{2.83}$$

With (2.83) we can write (2.82) for $\hat{\alpha}_{n-1}$ as (2.84) and $\hat{\alpha}_n$ as (2.85).

$$\hat{\alpha}_{n-1} = \frac{\hat{\alpha}_{n-1} \left(u_{n-1}^2 + \mu \sum_{k=1}^{n-2} u_k^2 \right) - K_{\alpha_{n-1}}}{u_{n-1}^2 + \mu \sum_{k=1}^{n-2} u_k^2} \tag{2.84}$$

$$\hat{\alpha}_n = \frac{K_{\alpha_n} + K_{\alpha_{n-1}} (1 - \mu) + \hat{\alpha}_{n-1} \left(u_{n-1}^2 + \mu \sum_{k=1}^{n-2} u_k^2 \right)}{u_n^2 + \mu \left(u_{n-1}^2 + \sum_{k=1}^{n-2} u_k^2 \right)} \tag{2.85}$$

From (2.70), (2.84) and (2.85), Δ_n can be written as (2.86).

$$\begin{gathered}\Delta_n = \hat{\alpha}_n - \hat{\alpha}_{n-1} \\ \Delta_n = \frac{K_{\alpha_n} + K_{\alpha_{n-1}}(1-\mu) + \hat{\alpha}_{n-1}\left(u_{n-1}^2 + \mu\sum_{k=1}^{n-2} u_k^2\right)}{u_n^2 + \mu\left(u_{n-1}^2 + \sum_{k=1}^{n-2} u_k^2\right)} - \frac{\hat{\alpha}_{n-1}\left(u_{n-1}^2 + \mu\sum_{k=1}^{n-2} u_k^2\right) - K_{\alpha_{n-1}}}{u_{n-1}^2 + \mu\sum_{k=1}^{n-2} u_k^2} \\ \Delta_n = \frac{K_{\alpha_n} - K_{\alpha_{n-1}}(1-\mu) - \hat{\alpha}_{n-1}\left(u_n^2 - u_{n-1}^2(1-\mu)\right)}{u_n^2 + \mu\sum_{k=1}^{n-1} u_k^2}\end{gathered} \tag{2.86}$$

The last part of equation (2.86) allows us to write $\mu\sum_{k=1}^{n-2} u_k^2$ as (2.87).

$$\mu\sum_{k=1}^{n-2} u_k^2 = \frac{K_{\alpha_{n-1}} - K_{\alpha_{n-2}}(1-\mu) - \hat{\alpha}_{n-2}\left(u_{n-1}^2 - u_{n-2}^2(1-\mu)\right)}{\Delta_{n-1}} - u_{n-1}^2 \tag{2.87}$$

Substituting (2.87) in the last part of (2.86) arrives at (2.88).

$$\Delta_n = \frac{\Delta_{n-1}\left(K_{\alpha_n} - K_{\alpha_{n-1}}(1-\mu) - \hat{\alpha}_{n-1}\left(u_n^2 - u_{n-1}^2(1-\mu)\right)\right)}{K_{\alpha_{n-1}} - K_{\alpha_{n-2}}(1-\mu) - \hat{\alpha}_{n-2}\left(u_{n-1}^2 - u_{n-2}^2(1-\mu)\right) + \Delta_{n-1}\left(u_n^2 - u_{n-1}^2(1-\mu)\right)} \tag{2.88}$$

Replacing (2.79) in (2.88) and simplifying the equation arrives in (2.89), the final form for the update term of the estimator.

$$\Delta_n = \Delta_{n-1}\frac{K_{\alpha_n} - K_{\alpha_{n-1}}(1-\mu) - \hat{\alpha}_{n-1}\left(u_n^2 - u_{n-1}^2(1-\mu)\right)}{\Gamma_n} \tag{2.89}$$

where Γ_n is written as (2.90).

$$\begin{aligned}\Gamma_n = K_{\alpha_{n-1}} - K_{\alpha_{n-2}}(1-\mu) - \hat{\alpha}_{n-2}\left(u_{n-1}^2 - u_{n-2}^2(1-\mu)\right) + \\ \Delta_{n-1}\left(u_n^2 + \mu u_{n-1}^2 - u_{n-2}^2(1-\mu)\right)\end{aligned} \tag{2.90}$$

With (2.70) and (2.89) we can finally write the full form of the recursive α estimator as (2.91).

$$\hat{\alpha}_n = \hat{\alpha}_{n-1} + \Delta_{n-1}\frac{K_{\alpha_n} - K_{\alpha_{n-1}}(1-\mu) - \hat{\alpha}_{n-1}\left(u_n^2 - u_{n-1}^2(1-\mu)\right)}{\Gamma_n} \tag{2.91}$$

The α estimator in (2.91) does not use any accumulator and has the forgetting factor,

allowing it to be fully integrated into the iPID controller to form the iPIDα.

2.5.4 iPIDα Controller Structure and Tuning

To properly integrate the α estimator with the iPID controller and arrive at the iPIDα, we can essentially do:

1. An initial value for $\hat{\alpha}_{init}$ is tuned to the system and used to initialize the controller ($\hat{\alpha}_0 = \hat{\alpha}_{init}$).
2. At every iteration:
   (a) The iPID controller, including the $\hat{F}_n$ estimator, is computed normally using the current $\hat{\alpha}_n$.
   (b) The estimated parameter $\hat{F}_n$, the control output u_n and the ν derivative of the desired trajectory y_r are used to update the $\hat{\alpha}_n$ for the next iteration.

The tuning of the iPIDα controller is somewhat simpler than the tuning of the traditional iPID. Since the iPIDα tries to automatically seek an optimal α it can be used for a kind of self-tuning. The proposed steps for the controller tuning are:

1. Initialize the controller with the initial estimate for $\hat{\alpha}_{init}$ as a very large value with a reasonable forgetting factor ($\mu = 0.95$ is usually a good first value).
2. Choose small, different from zero, controller gains for the iPID (K_p gain for the iP controller, for example).
3. Run the experiment with the desired trajectory, $\hat{\alpha}$ should try to automatically converge to an optimal value.
4. Update the initial estimate for $\hat{\alpha}_{init}$ with the last value of the $\hat{\alpha}$ generated from the last experiment ($\hat{\alpha}_{init} = \hat{\alpha}_{final}$).
5. Run steps 3 and 4 as many times as needed for $\hat{\alpha}$ to converge to about the same value as the initial estimate $\hat{\alpha}_{init}$. For example, the changing rate of $\hat{\alpha}_{init}$ between new iterations should be smaller than 0.1%.
6. After $\hat{\alpha}$ converges, adjust the controller gain to achieve the desired system dynamics. This should change the final convergence value for $\hat{\alpha}$.
7. Repeat steps 3 and 4, as needed, to converge $\hat{\alpha}$ again.

8. If the new $\hat{\alpha}$ affected the desired dynamic response, readjust the controller gains and repeat the process to converge $\hat{\alpha}$. Repeat the steps as many times as needed to achieve the desired system dynamics.

The described steps appear complex, but ultimately it only repeats some basic steps several times. It has the advantage of the controller designer only needing to adjust the controller gains (letting the estimated α parameter adjust itself to each new gain). Also, since an optimal $\hat{\alpha}$ keeps being estimated in real time the initial estimate $\hat{\alpha}_{init}$ for it does not need to be particularly precise.

Since the optimal $\hat{\alpha}$ of the iPIDα is calculated after the F parameter, the $\hat{\alpha}$ estimator acts as a sort of fine tuning of the controller. Just like in the aciPID, the estimated parameter F is still the main responsible for adjusting the controller to variations in the system. Therefore, $\hat{\alpha}$ is expected to vary relatively little in value when compared to F.

3 SIMPLIFIED VEHICLE SIMULATOR RESULTS

3.1 Test Simplified Vehicle Simulator

To initially validate the usage of iPID controllers for longitudinal vehicle control, a simulated test vehicle was created with the help of MATLAB/Simulink software and used to compare an iPID controller to its equivalent PID controller. These initial tests seek to compare the performance of both controllers in order to verify if it is indeed possible to substitute a PID controller with an iPID one and if there is any real advantage to it. Since a direct comparison of an iPID and an PID with equivalent gains is desired, the proposed iPIDα controller was not considered for the tests.

The simplified vehicle model was created by the author based on resources provided by MathWorks itself (MATHWORKS, 2022a; MATHWORKS, 2022b). The created model allows the simulation of all major dynamics of a vehicle, including the engine, power train, four-speed gearbox, tires, and longitudinal vehicle dynamics.

This simulated vehicle is controlled through two inputs that represent acceleration and braking signals, ranging from 0 to 1, and are equivalent, respectively, to acceleration and brake pedals from 0 to 100%. The output of the system is the vehicle speed in mph, which was converted externally to km/h for controller development and testing.

Works such as (POLACK; DELPRAT; NOVEL, 2019) showed that an iP controller, with $\nu = 1$ and $K_I = 0$, is enough to be able to control a real vehicle. Using an iP controller also have the advantage of simplifying the controller design while making tuning easier, so it was chosen to control the simulated vehicle. To allow a single SISO controller to act on these two inputs, the controller output is saturated in the range of -1 to 1 and divided between the acceleration and braking commands:

- If the controller output is greater than 0, the car needs to accelerate and only the accelerator pedal is pressed;

- If the controller output is less than or equal to 0, the car needs to brake, and only the brake pedal is pressed (controller signal is converted from range -1 to 0 to range 0 to 1.

The controller input is the error between the car's current speed and the desired speed, in km/h. This way of controlling the vehicle mimics the way a human would control it, so it could be transported to a real vehicle with relative ease. Furthermore, it forces a single controller to deal with two different dynamics when it needs to accelerate or brake, a situation where the use of MFC is highly interesting. The iP controller was implemented using the MATLAB function feature and needed to be coded in a MATLAB script language. This means the controller can easily be ported to embedded applications. In Figure 3 one can see the controller block used in the simulations. The MATLAB code developed to execute the iP controller is presented in appendix A.

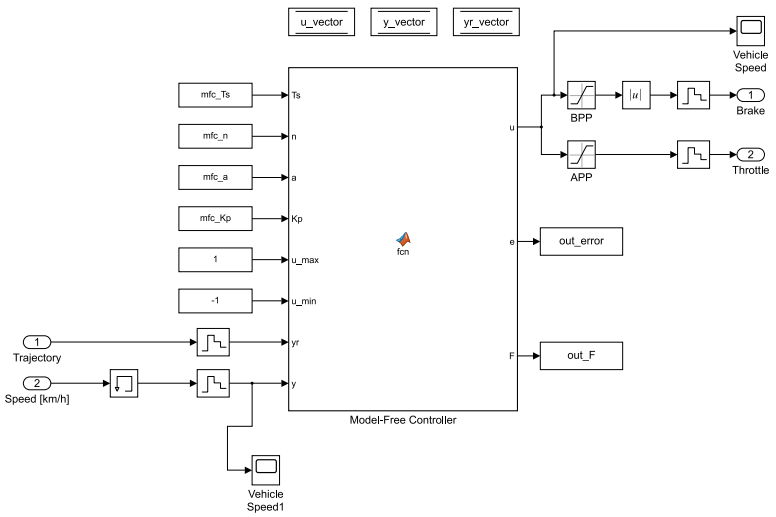


Figure 3: MFC for vehicle control, in Simulink.

Source: Author.

The complete system used for all tests and simulations is shown in Figure 4.

3.2 Controller Tuning

To tune the iP controller, several tests were performed with a speed reference of a sequence of steps, starting at 40 km/h and increasing in steps of 20 km/h to a maximum speed of 120 km/h, and then decreasing in steps of 20 km/h to a final speed of 40 km/h. The reference remained constant for 100 seconds after each step. This sequence of steps

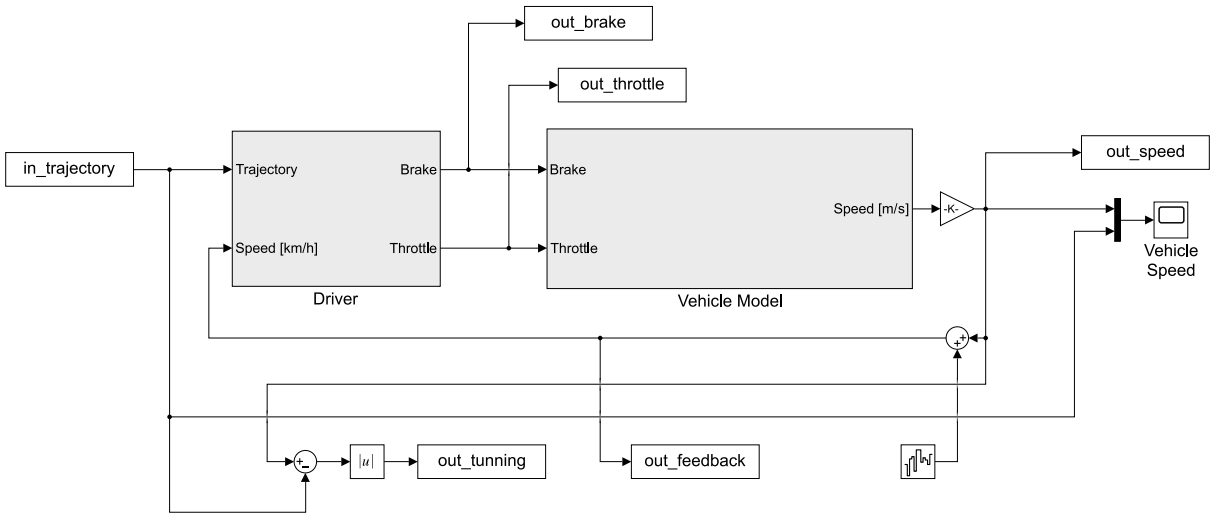


Figure 4: Complete simulation system, in Simulink.

Source: Author.

allows the controller to act over the full range of possible vehicle speeds and control both acceleration and braking.

The sample time for the controller and the integration window for the estimator $\hat{F}_{\nu=1}$ were chosen to be, respectively, $T_s = 0.1$ seconds and $n = 2$ (integration time of 0.2 seconds). Several integration window sizes were tested and a size of $n = 2$ was already able to properly control the simulated vehicle. Using the method described by (POLACK; DELPRAT; NOVEL, 2019), $\alpha = 400$ and $K_p = 0.085$ were obtained as the gains of the iP controller. These gains were chosen seeking a quick response with little overshoot, increasing the comfort and safety of potential passengers. All relevant parameters for the tuned iP controller are shown in Table 1.

Table 1: Tuned iP controller parameters for the simplified vehicle simulation

Parameter	Description	Value	Unit
T_s	Sampling time	0.1	s
n	Integration window size	2	-
α	Ultra-local α parameter	400	-
K_p	iP controller gain	0.085	-
u_{max}	Maximum controller output	1	-
u_{min}	Minimum controller output	-1	-

Source: Author.

For comparison purposes, the simulated vehicle control was also performed using a traditional PI controller. Their gains were defined according to (2.21) and (2.22). Thus,

we obtain, respectively, the gains $K_{P_{PI}} = 0.0250$ and $K_{I_{PI}} = 0.0021$, which should make the PI capable of performing similarly to the designed iP controller.

The response of the controlled system with the described speed reference, both with the iP and PI, is shown in Figure 5. We can see the controllers have very similar responses, practically overlapping, and managed to obtain good performance even with the vehicle having different dynamics for acceleration and braking. Some disturbances caused by the automatic gear shifting were also correctly compensated for.

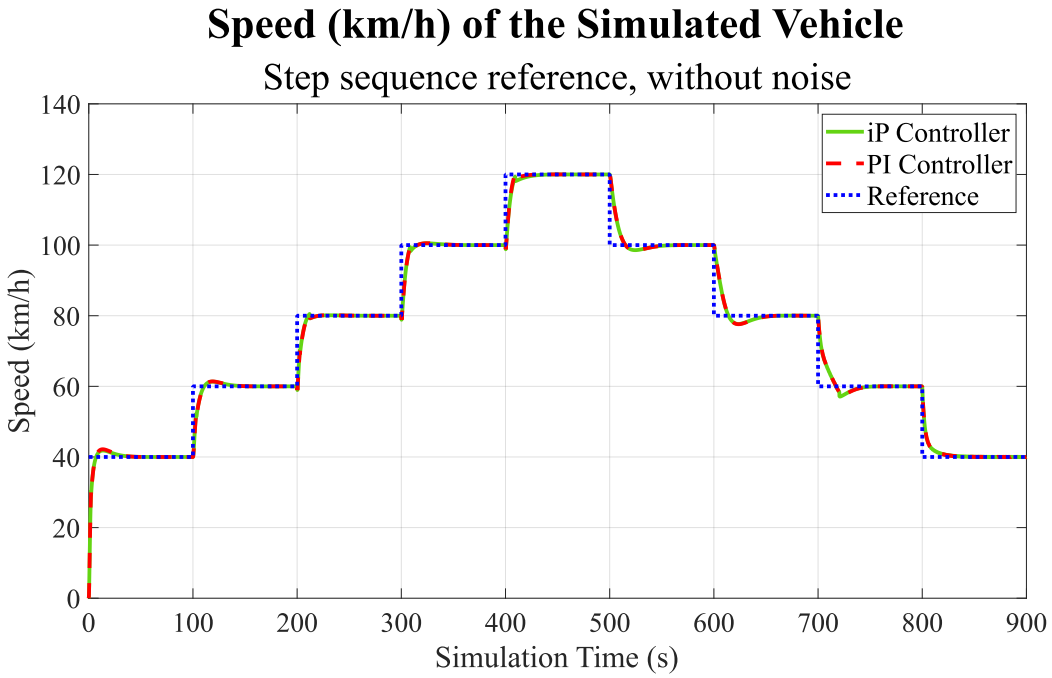


Figure 5: Speed, in km/h, of the simulated vehicle for the iP (green curve) and PI (red curve) controllers, together with the speed reference (blue curve).

Source: Author.

In Figure 6 we can see the result of the estimation of the parameter F of the ultra-local model. Both the throttle and brake pedals efforts are presented in Figures 7a and 7b, respectively.

An interesting aspect of iPID controllers is that, due to the intrinsic filtering of the ultra-local model estimation process, it tends to have a high noise rejection capability. To compare the projected iP with the equivalent PI in this context, measurement noise was inserted at the feedback speed signal, which can be seen in Figure 8. The results of the controllers for this situation are presented in Figure 9. The noise generator can be seen in Figure 4 and is a band-limited white noise generator configured with noise power equal to 0.1, directly connected to the speed feedback signal as additive noise.

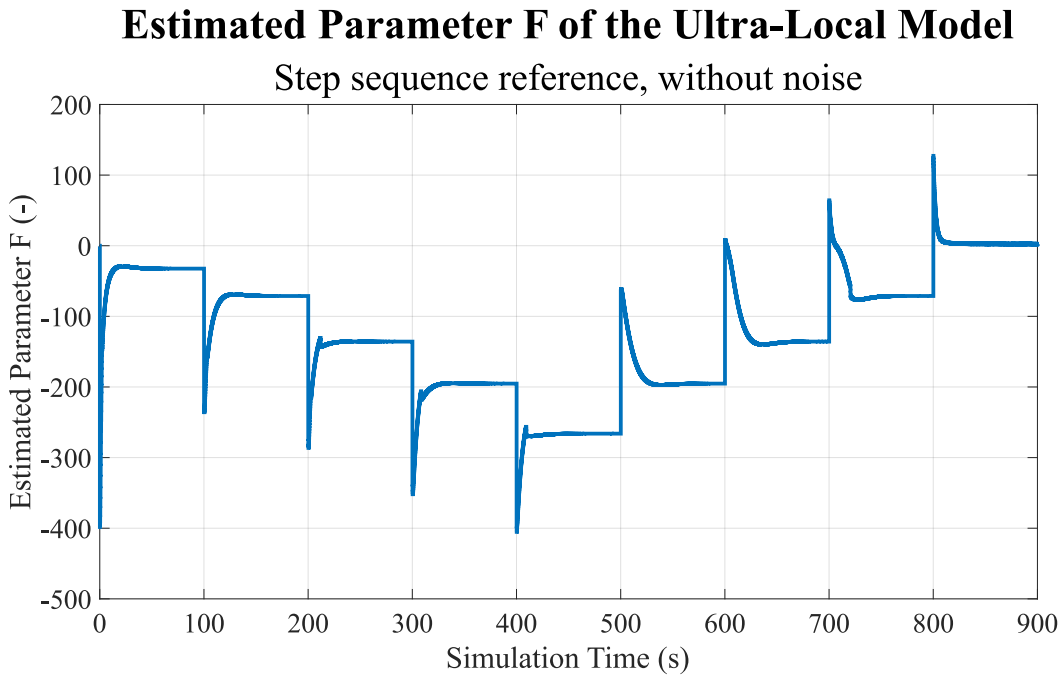
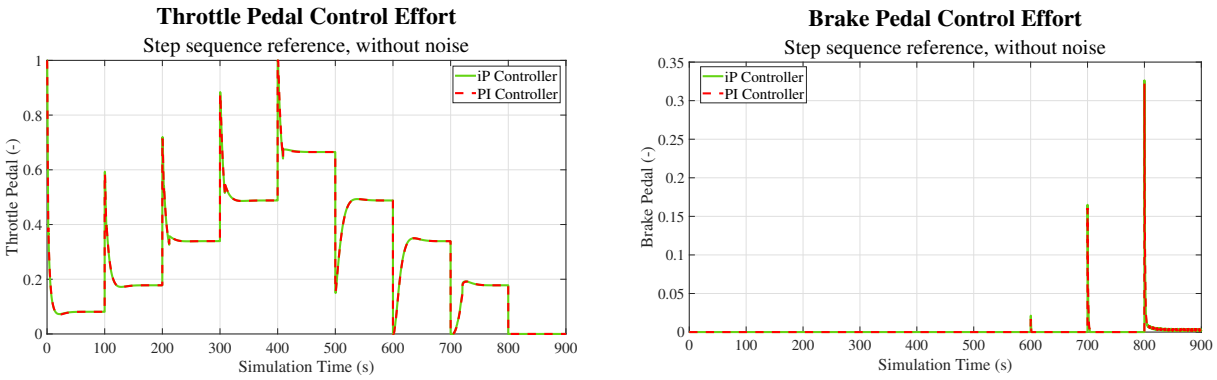


Figure 6: Estimation of the F parameter of the ultra-local model for the multi-step reference.

Source: Author.



(a) Throttle pedal effort.

(b) Brake pedal effort.

Figure 7: Throttle and brake pedals efforts for the reference of several steps, without measurement noise, for the controllers iP (green curve) and PI (red curve).

Source: Author.

Even with the noise, both controllers managed good responses, having similar results to the test without noise. The estimation of the ultra-local model of this case is presented in Figure 10, and we can see that it is trying to include the measurement noise as part of the unknown dynamics of the system. Even with the noise, it manages to maintain a behavior very similar to the noiseless case. Both the throttle and brake pedals efforts are presented in Figures 11a and 11b, respectively.

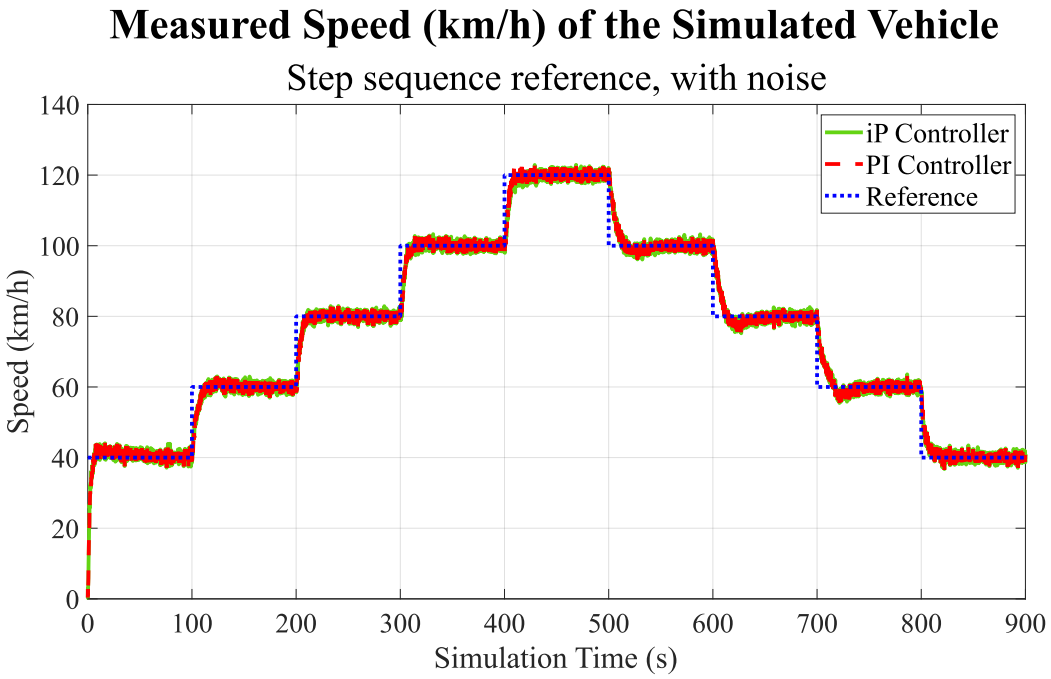


Figure 8: Measured speed, in km/h, of the simulated vehicle with measurement noise, for the iP (green curve) and PI (red curve) controllers, together with the speed reference (blue curve).

Source: Author.

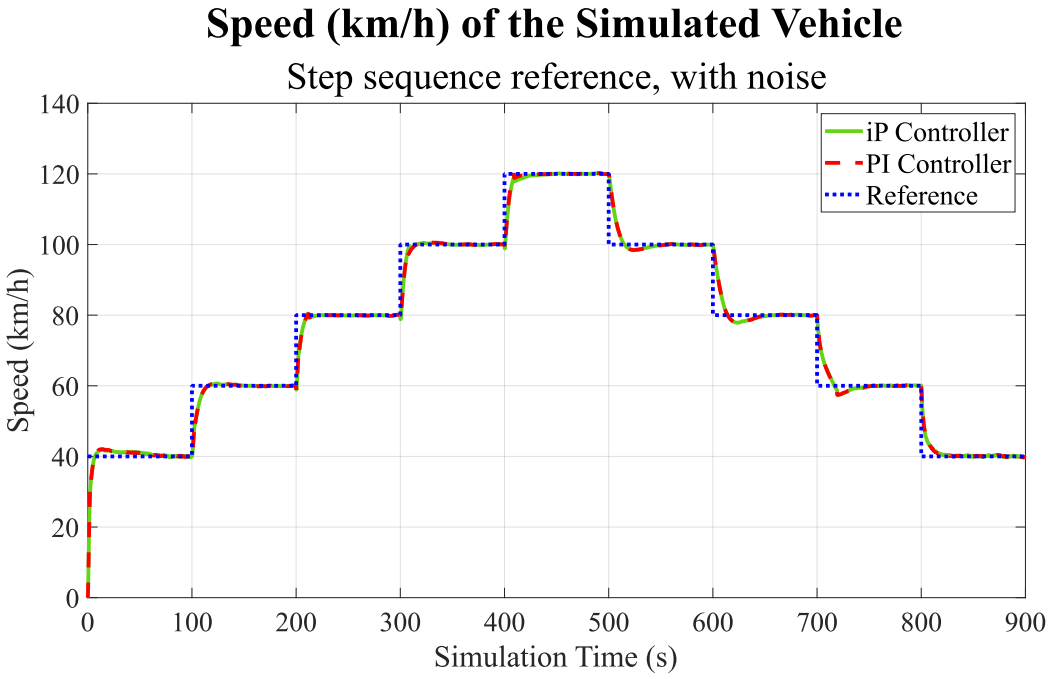


Figure 9: Actual speed, in km/h, of the simulated vehicle, for the controllers iP (green curve) and PI (red curve), together with the speed reference (blue curve).

Source: Author.

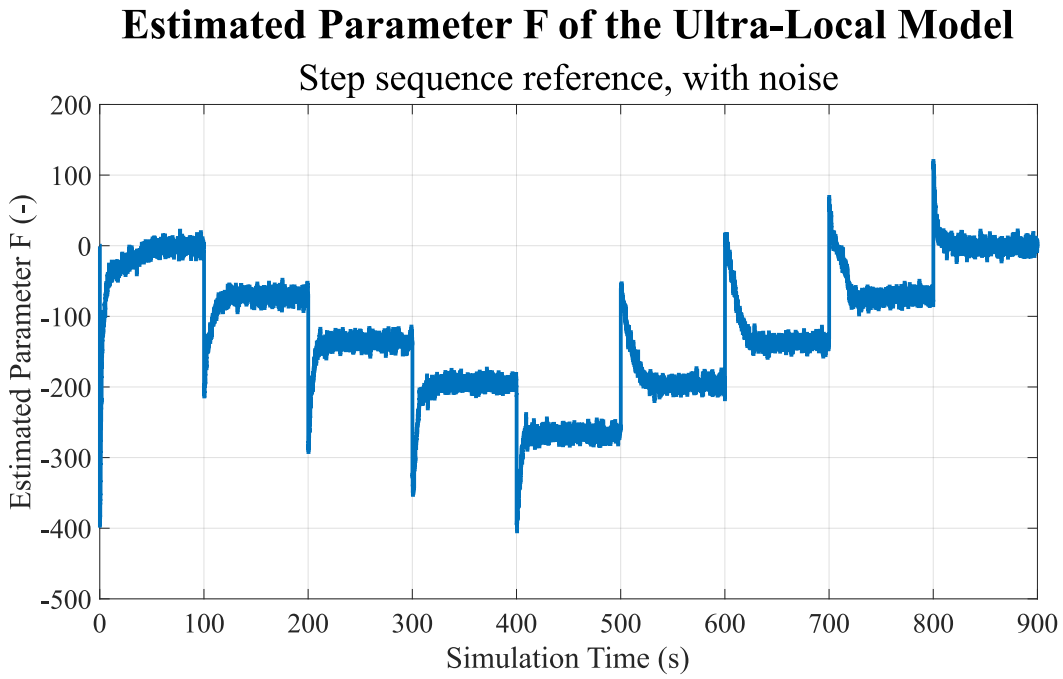
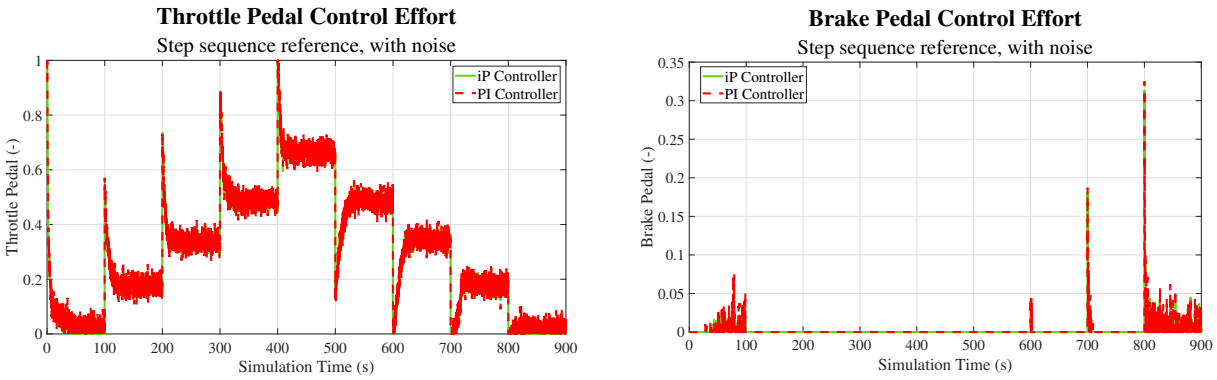


Figure 10: Estimation of the F parameter of the ultra-local model for the reference of several steps, with measurement noise.

Source: Author.



(a) Throttle pedal effort.

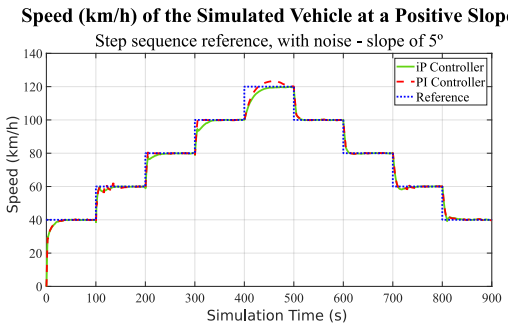
(b) Brake pedal effort.

Figure 11: Throttle and brake pedals efforts for the reference of several steps, with measurement noise, for the controllers iP (green curve) and PI (red curve).

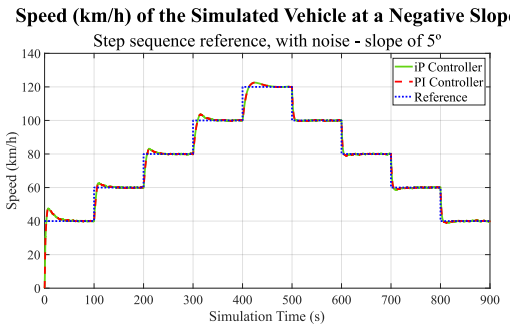
Source: Author.

Another promising aspect of the iPID is, as an MFC, the ability to adapt itself to varying environmental conditions or plant modifications. To test this aspect, the road incline of the simulated vehicle was changed which is equivalent to the simulated vehicle being put on a slope. Figure 12a shows the result of the controllers when the simulated vehicle is positioned on a positive slope and needs to go up a slope of 5° of incline. On the other hand, Figure 12b shows the result of the controllers when the simulated vehicle

is positioned on a negative slope and needs to go down a slope of 5° of incline.



(a) Positive slope of 5°.



(b) Negative slope of 5°.

Figure 12: Actual speed, in km/h, of the simulated vehicle on different slopes with measurement noise, for the controllers iP (green curve) and PI (red curve), together with the speed reference (blue curve).

Source: Author.

In both cases (positive and negative slopes), the iP and PI controllers had very similar results, with the PI having a higher overshoot in some of the speed steps.

3.3 Emergency Braking Test Procedure and Results

One of the main interests of the tests performed is to find out how an iP controller behaves when subjected to variations in the system, especially for the braking dynamics which are important to ensure the safety of potential passengers. While all of the tests performed so far already actively controlled the simulated vehicle brakes to achieve the desired speed, none of them simulated an emergency situation where the breaking need to be robust enough to prevent accidents.

To simulate such an emergency situation and verify how the two controllers compare, a new speed reference was created where the vehicle must accelerate up to 120 km/h and then brake hard to 40 km/h. During the tests, the measurement noise previously used in section 3.2 was maintained. The results for these tests with both iP and PI controllers are shown in Figure 13. We can see that both controllers achieved similar results, with the PI being somehow more aggressive. Both the throttle and brake pedals efforts are presented in Figures 14a and 14b, respectively.

Similar to the test performed for Figures 12a and 12b, the road incline of the simulated vehicle was changed to verify how the controllers behave in these new conditions. Figures 15a and 15b show the result of the controllers for a positive slope of 5° and a negative slope of 5°, respectively.

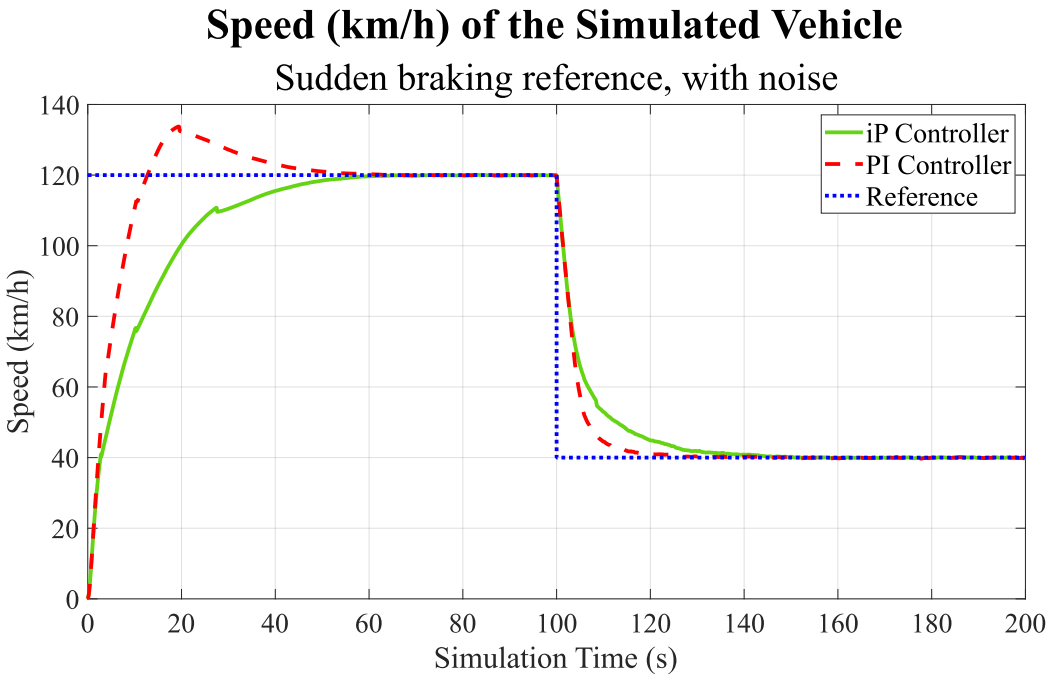
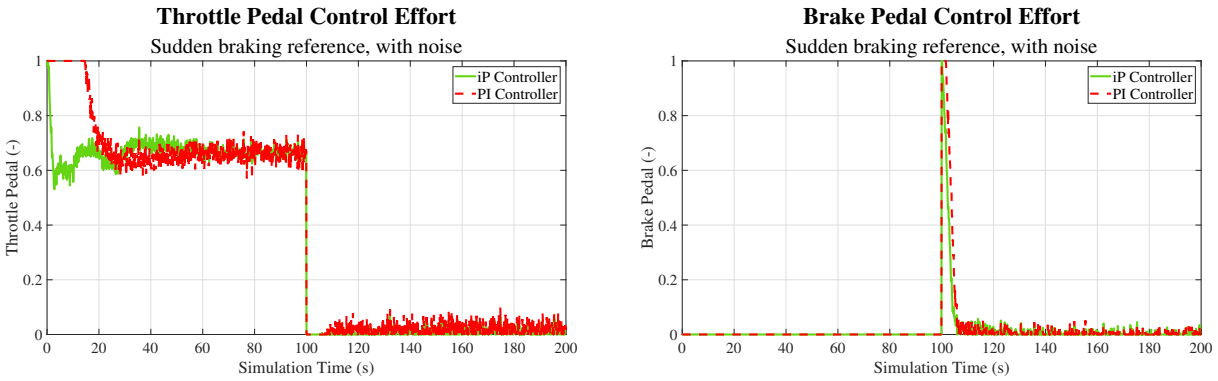


Figure 13: Actual speed, in km/h, for the simulated vehicle braking test with measurement noise, for the iP (green curve) and PI (red curve) controllers, together with the speed reference (blue curve).

Source: Author.



(a) Throttle pedal effort.

(b) Brake pedal effort.

Figure 14: Throttle and brake pedals efforts for the braking test, with measurement noise, for the controllers iP (green curve) and PI (red curve).

Source: Author.

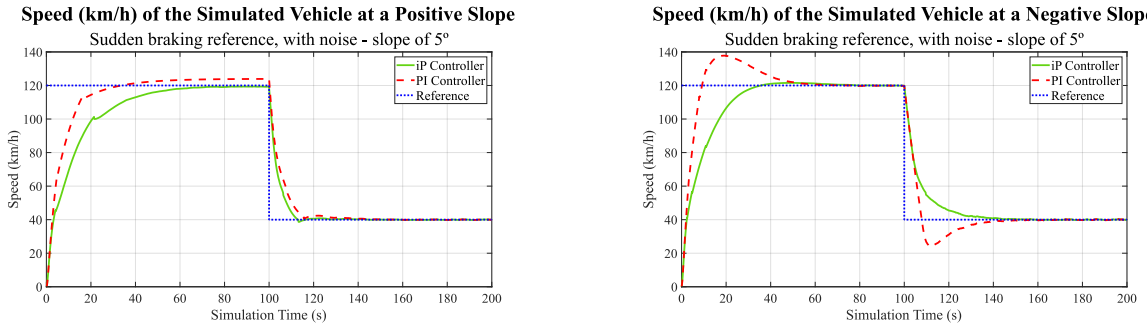
In both of these cases, a big difference can be noticed between the iP and the PI controllers:

- For the positive slope (Figure 15a), the iP controller managed a similar response as before, but the PI controller was not able to achieve the correct reference speed of 120 km/h during the simulated time. It probably could achieve the correct speed if

the vehicle needed to maintain the speed of 120 km/h for a longer time.

- For the negative slope (Figure 15b), the iP controller managed again a similar response as before, while the PI controller caused a considerable overshoot of about 20 km/h both during acceleration and braking.

The iP controller managed to maintain a better result when compared to the PI controller during changes in environmental conditions of the simulated vehicle. This is due to the ultra-local model of the iPID trying to compensate for these disturbances.



(a) Positive slope of 5°.

(b) Negative slope of 5°.

Figure 15: Actual speed, in km/h, for the simulated vehicle braking test on different slopes with measurement noise, for the iP (green curve) and PI (red curve) controllers, together with the speed reference (blue curve).

Source: Author.

In order to better study how much the response of both controllers can differ for different slopes, a sweep of road inclinations was performed from a slope of negative 5° to a slope of positive 5°, in increments of 0.5°. The response for all of these different slopes is presented in Figure 16 for both the iP and PI controllers.

It can be noticed that the PI controller response varied a lot during the sweep and caused an overshoot in most of the tests. On the other hand, the iP had a very consistent response among all the tests, which very little variation across all different slopes. This shows how powerful MFC can be to control plants subject to varying environmental conditions.

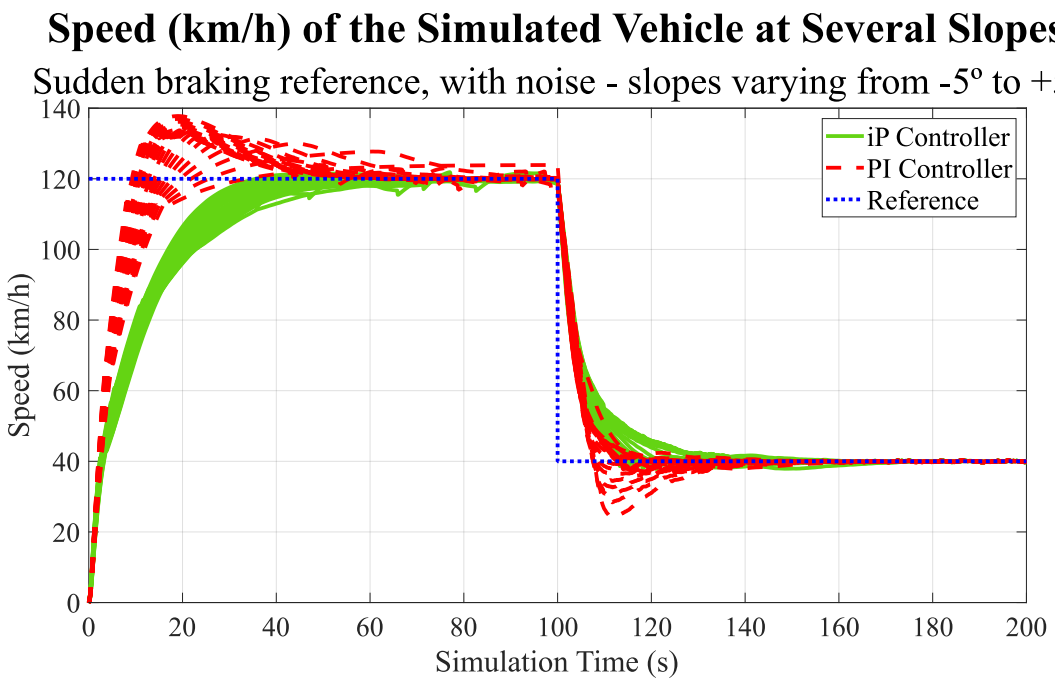


Figure 16: Actual speed, in km/h, for the simulated vehicle braking test on several slopes, from a negative slope of 5° to a positive slope of 5° with measurement noise, for the iP (green curve) and PI (red curve) controllers, together with the speed reference (blue curve).

Source: Author.

4 CARSIM SIMULATOR RESULTS

4.1 CarSim Software

The simulated vehicle presented in section 3.1 is handy to test the applicability of different types of vehicle longitudinal control schemes, but it is still a somewhat simplified model. To perform more realistic tests, Mechanical Simulator‘s CarSim software (version 2022.0), a specialized commercial software used for simulating the performance of passenger vehicles and light-duty trucks (CARSIM. . . , 2022), was also used for testing of the developed controllers. Different from the test with the simplified simulated vehicle, that seek to compare an iPID to an equivalent PID, the tests performed in CarSim are meant to test how effective an iPID controller is for this application. Therefore, no tests were performed using an equivalent PID. Instead, both an iPID and the proposed iPIDα were used to control the CarSim simulated vehicle.

CarSim's interface can be seen in Figure 17 and among several other features and advantages, it can be easily integrated with Simulink/MATLAB, allowing an easy transition from the current simplified vehicle simulation to a CarSim-based one.

An interesting aspect of CarSim software is that although it has numerous options and different configurations, it also has a relatively simple user interface and a lot of pre-configured test scenarios. These scenarios include several tests specific to the development and validation of all kinds of vehicle features, including ADAS and ACC. Particularly, several different Euro NCAP ACC pre-configured test scenarios were used for the validation of the developed controller.

The vehicle model used for all the tests with CarSim is the default vehicle for its Euro NCAP tests scenarios. It is a pre-defined C-Class vehicle with integrated ACC, Forward Collision Warning (FCW) and LKAS. The model has the possibility of having an AEB system that is disabled. The designed controllers will be used to partially substitute the default ACC with the iPID and iPIDα controllers. This is the standar

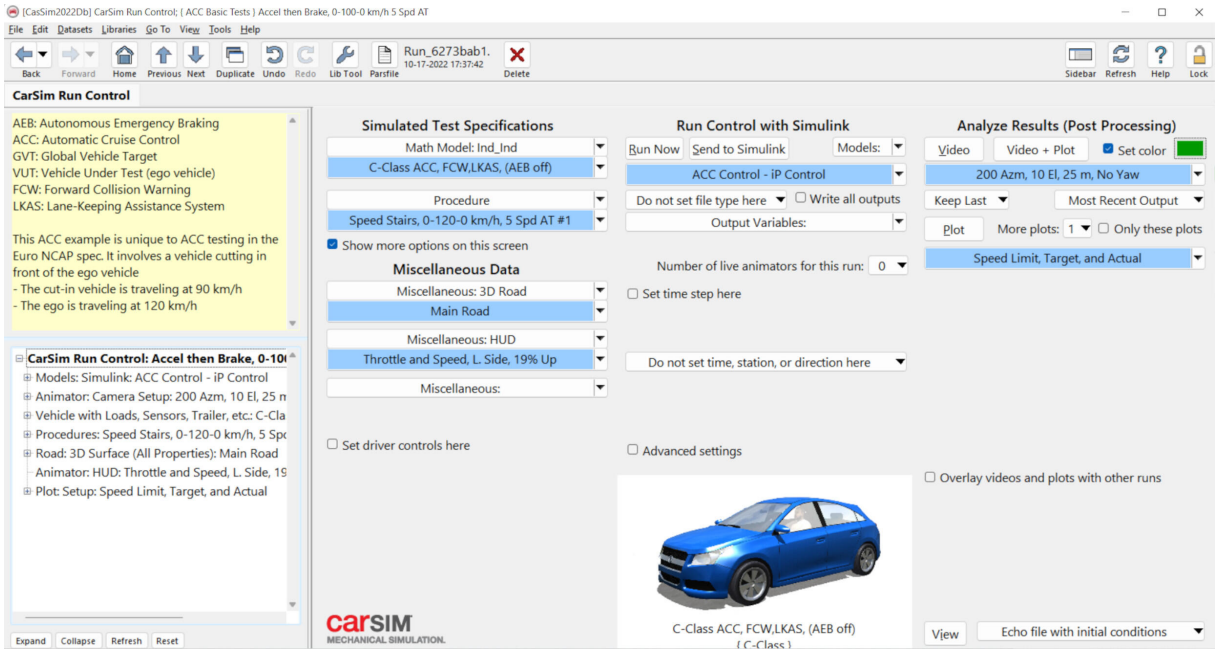


Figure 17: CarSim software basic interface.

Source: Author.

CarSim offers an integration option with Simulink where it automatically creates a Simulink block with configurable inputs and outputs that execute the entire vehicle simulation. To use the developed controllers with this block, it is first necessary to configure it to use a Simulink Model. The model configuration window can be seen in Figure 18 and the simulation basic configurations, such as integration method and step-time, can be left as default, but the Imports and Exports Channels need to be modified.

The Imports Channels configuration window can be seen in Figure 19, and it allows the configuration of the Simulink block inputs. For the developed control, the inputs of the model must be a way to control the throttle and one to control the brakes. The chosen variables to be imported were:

- IMP_THROTTLE_ENGINE, which directly controls the engine throttle with a signal from 0 to 1.
- IMP_PCON_BK, which controls the brake pressure (in MPa) with a signal from 0 to 20.

The Export Channels configuration window can be seen in Figure 20, and it allows the configuration of the Simulink block outputs. For the developed control, the outputs of the model must be the vehicle's target speed and its current speed. The chosen variables to be exported were:

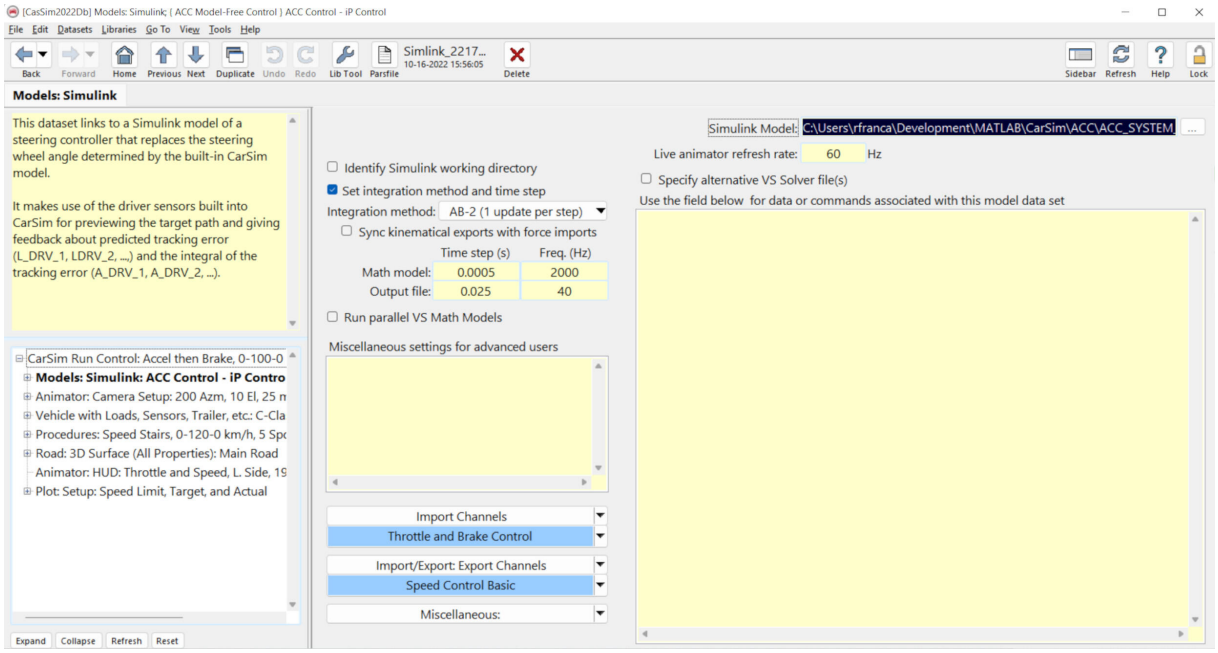


Figure 18: CarSim Simulink model configuration window.

Source: Author.

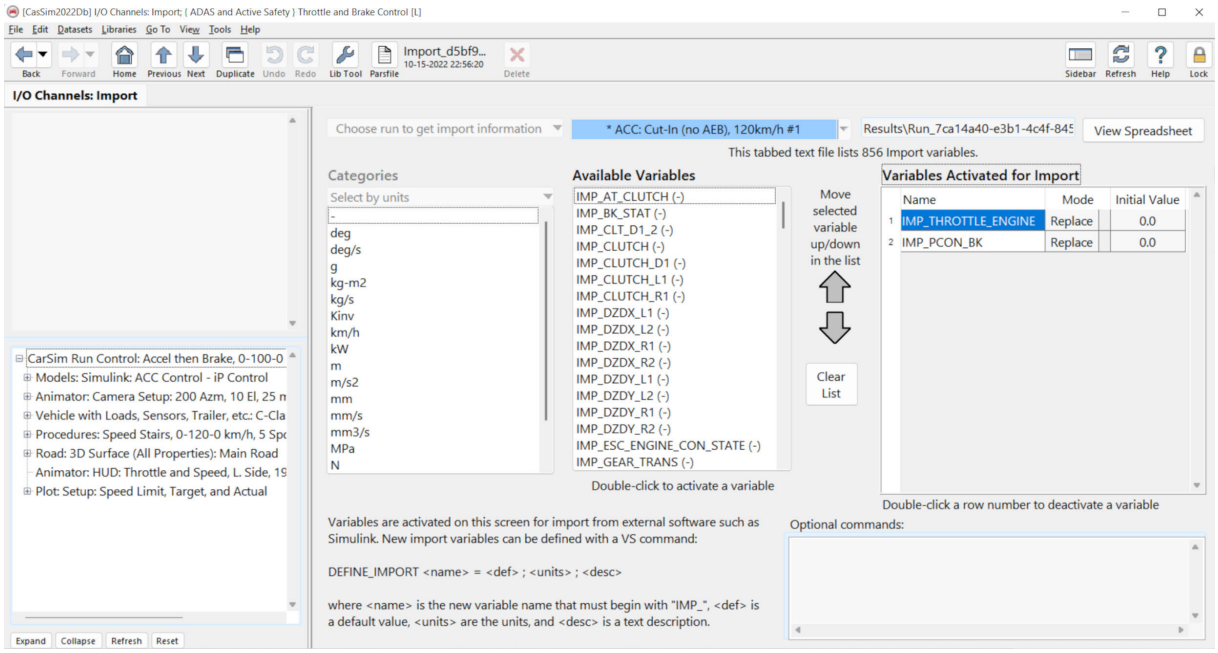


Figure 19: CarSim I/O Channels Import configuration window.

Source: Author.

- VxTarget, which outputs the target speed of the vehicle in km/h.
- Vx, which outputs the current vehicle speed in km/h.

Once the CarSim Simulink model is properly configured, it needs to be integrated

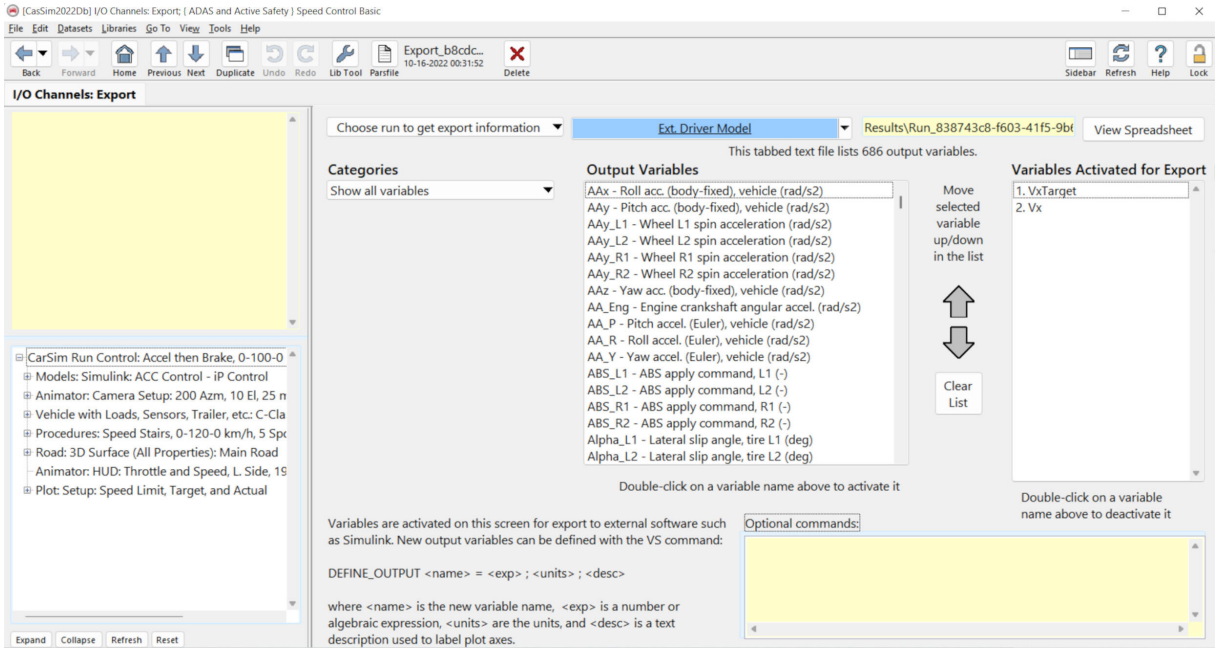


Figure 20: CarSim I/O Channels Export configuration window.

Source: Author.

with the developed controller in Simulink, which can be seen in Figure 21. It is important to notice that the high-level speed reference generation is done by CarSim and the implemented controller only performs low-level throttle and brake control to achieve the desired speed.

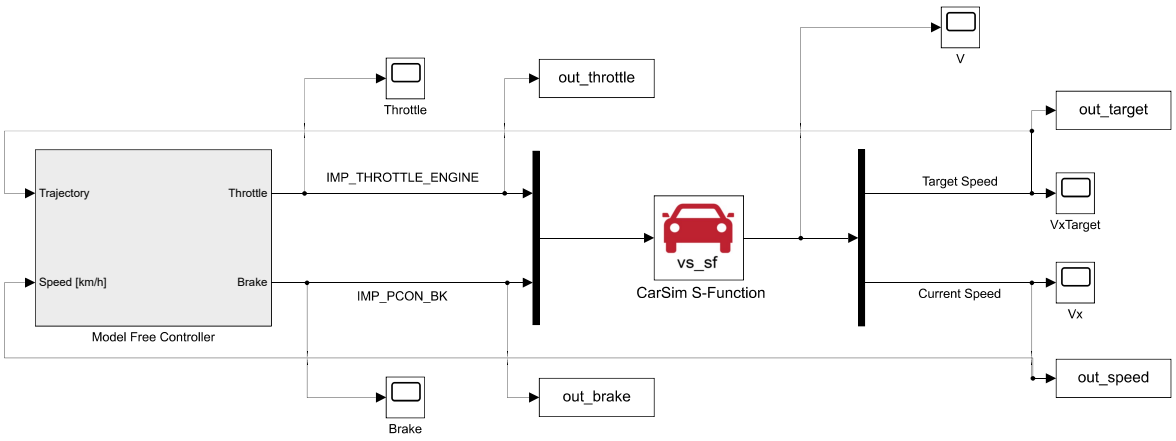


Figure 21: Simulink model with the CarSim block and the developed controller integrated.

Source: Author.

4.2 iP MFC Controller Tuning

Just like in the tests with the simplified simulated vechicle, the MFC choosen to control the CarSim model is an iP controller, with $\nu = 1$ and $K_I = 0$. The internal structure of the Model-Free Controller block presented in Figure 21 can be seen in Figure 22.

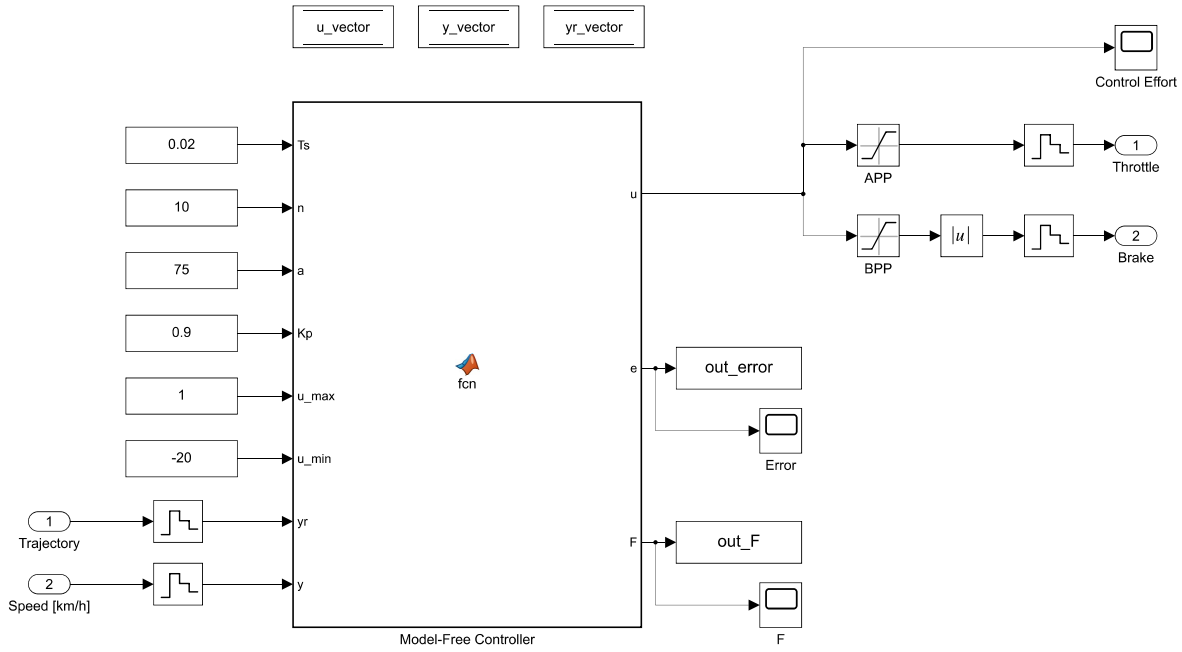


Figure 22: Simulink model with the developed iP controller.

Source: Author.

The output of the controller is a single signal that is divided into positive and negative parts and used for the throttle and brake, respectively. Just like the iP controller used in chapter 3. An interesting aspect of this application of the iP controller is the large asymmetry between positive and negative controller limits:

- The positive limit is 1 because the CarSim engine throttle input varies from 0 to 1.
- The negative limit is -20 because the CarSim brake pressure input varies from 0 to 20 MPa.

The negative limit is 20 times higher than the positive limit, which represents a challenge for the controller. Combined with the fact that the engine throttle and brake pressure inputs have completely different dynamics on the output, it makes the controller configuration even more challenging. The iP controller is expected to be able to adapt itself to these conditions with the help of the F parameter estimation of the ultra-local model.

To tune and test the iP controller, a step reference similar to the one in section 3.2 was used. The reference was created inside CarSim and can be seen in Figure 23a. The speed reference starts at 20 km/h and increases in steps of 20 km/h in speed every 20 seconds until it reached 120 km/h. Then it starts decreasing the target speed in steps of 20 km/h every 20 seconds until the final speed of 20 km/h.

Using the method presented in section 3.2, the iP controller α and K_p were tuned to $\alpha = 75$ and $K_p = 0.9$, with an integration window of size $n = 10$ and sampling time of $T_s = 0.02$ seconds (integration time of 0.2 seconds). Larger integration window size allows the controller to better handle noise at the expense of performance, and the size of $n = 10$ showed in the tests to be a good balance between both. All relevant parameters for the tuned iP controller are shown in Table 2.

Table 2: Tuned iP controller parameters for the CarSim vehicle simulation

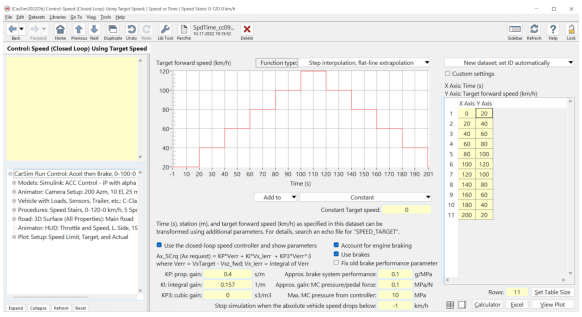
Parameter	Description	Value	Unit
T_s	Sampling time	0.02	s
n	Integration window size	10	-
α	Ultra-local α parameter	75	-
K_p	iP controller gain	0.9	-
u_{max}	Maximum controller output	1	-
u_{min}	Minimum controller output	-20	-

Source: Author.

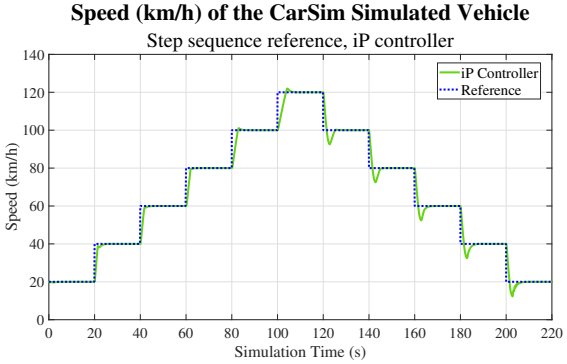
The simulated vehicle speed of these controller gains can be seen in Figure 23b. It is possible to see that the iP controller managed to properly control the vehicle and follow the reference speed with a good dynamic response, having some control issues at low speeds (around 20 km/h). Controlling a vehicle at such low speeds is a challenge in itself, so that's expected.

Since the speed reference is a sequence of steps, it ends up demanding the vehicle to accelerate or decelerate 20 km/h instantly. During acceleration, this demand does not cause much overshoot, but the breaking force necessary for these deaccelerations causes a noticeable undershoot to appear. It would be possible to reduce (or eliminate) this undershoot by reducing the controller gain and slowing the controller response. For safety purposes, the vehicle needs to be able to brake as fast as possible in emergency situations, so the controller gain was chosen to prioritize response, even if it introduces undershoots in these situations.

For this step reference, the estimated F parameter can be seen in Figure 24a, the



(a) CarSim target speed configuration window for the stairs reference.



(b) Speed reference (blue) and simulated vehicle speed (green) for the stairs reference with the CarSim model.

Figure 23: CarSim target speed configuration and iP controller result for the stairs reference with the CarSim model

Source: Author.

speed error can be seen in Figure 24b, the control effort applied to the engine throttle is presented in Figure 24c and, lastly, the control effort applied to the brake pressure is presented in Figure 24d.

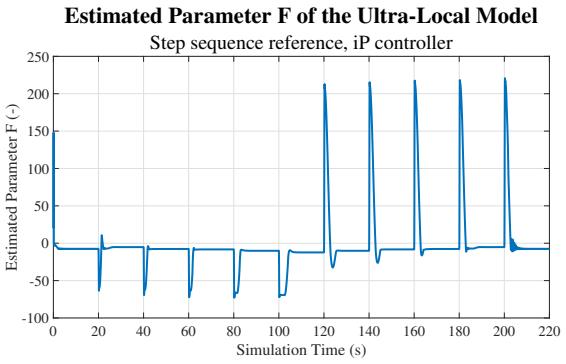
The same iP controller was also applied to the speed reference presented in Figure 25a, where the same target speed levels of Figure 23a are used with spline interpolation, instead of stairs interpolation, to create the reference speed curve. This reference demands a more precise control to maintain the speed close to the target speed, and the iP controller managed to obtain good results. The vehicle speed result of this new scenario can be seen in Figure 25b.

For this spline reference, the estimated F parameter can be seen in Figure 26a, the speed error can be seen in Figure 26b, the control effort applied to the engine throttle is presented in Figure 26c and, lastly, the control effort applied to the brake pressure is presented in Figure 26d.

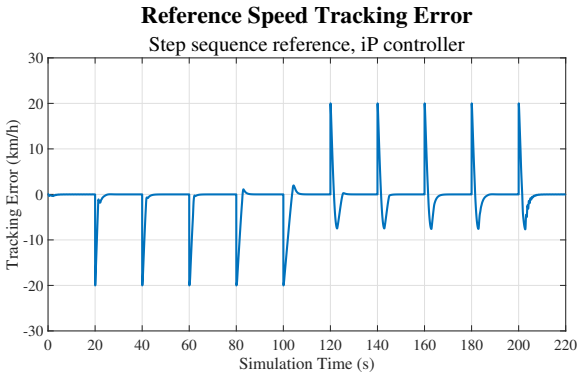
4.3 iPα AMFC Controller Tuning

Using the structure presented in section 2.5, an iPα AMFC was created using the α estimator. The internal structure of the Model-Free Controller block for this case is presented in Figure 27. The MATLAB code developed to execute the iPα controller is presented in appendix B.

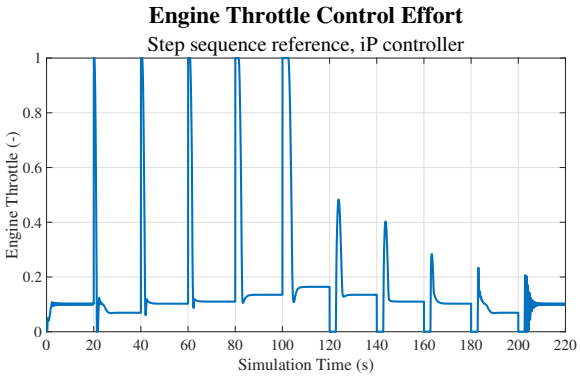
The output of the controller and the controller limits are the same as the iP presented



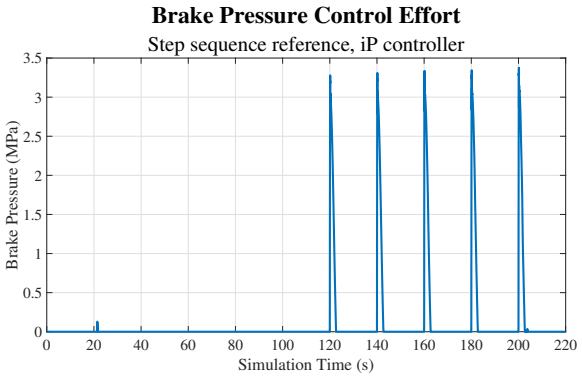
(a) Estimated parameter F (blue).



(b) Speed error (blue).



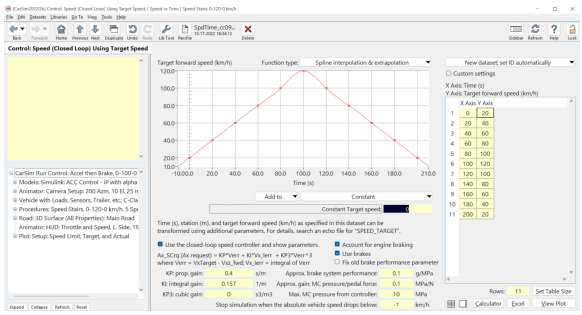
(c) Engine throttle control effort (blue).



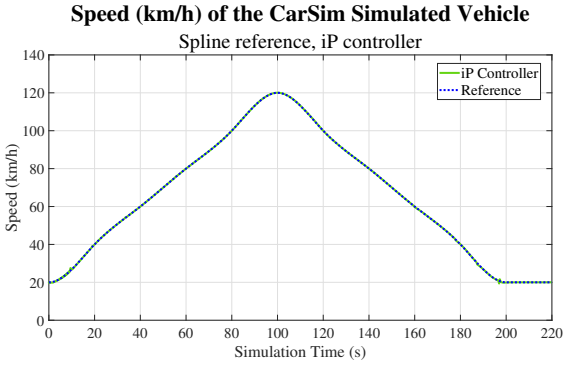
(d) Brake pressure control effort (blue).

Figure 24: Resulting iP controller curves for the stairs reference with the CarSim model.

Source: Author.



(a) CarSim target speed configuration window for the spline reference.

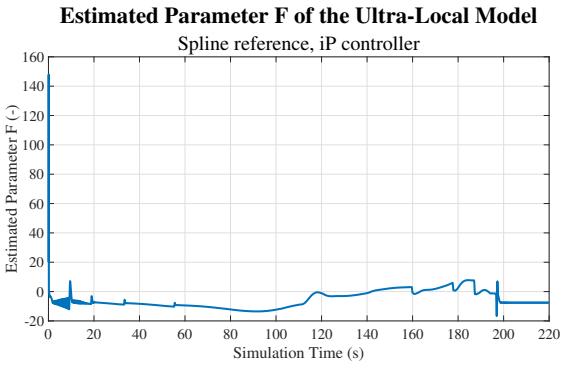


(b) Speed reference (blue) and simulated vehicle speed (green) for the spline reference with the CarSim model.

Figure 25: CarSim target speed configuration and iP controller result for the spline reference with the CarSim model

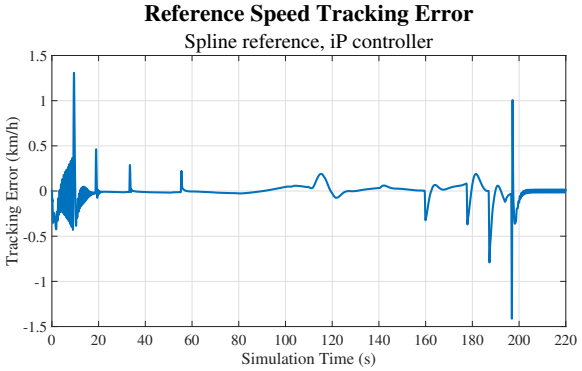
Source: Author.

in section 4.2 and the same step reference presented in Figure 23a was used to tune the iPα controller.



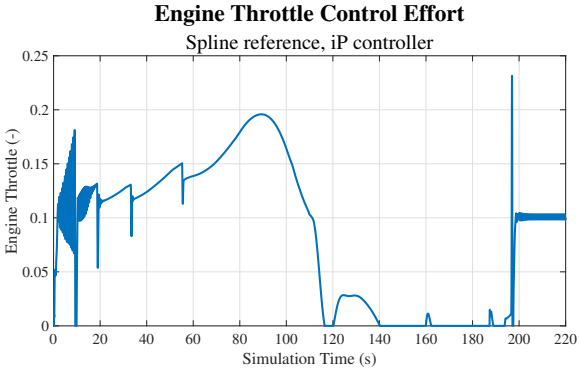
(a) Estimated parameter F (blue).

Source: Author.



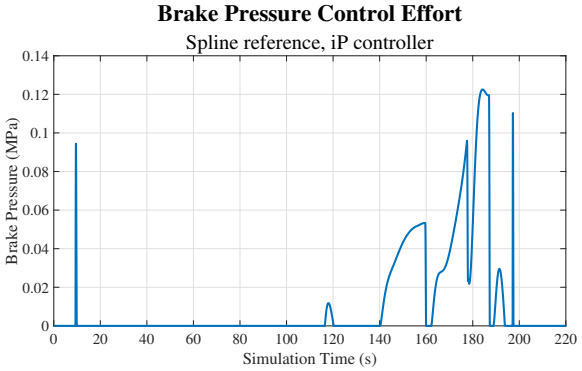
(b) Speed error (blue).

Source: Author.



(c) Engine throttle control effort (blue).

Source: Author.



(d) Brake pressure control effort (blue).

Source: Author.

Figure 26: Resulting iP controller curves for the spline reference with the CarSim model.

Source: Author.

Using the tuning procedure described in section 2.5, the gains of $\hat{\alpha}_{init} = 96.4303$ and $K_p = 0.90$ were achieved, with forgetting factor $\mu = 0.95$, integration windows size $n = 10$ and sampling time of $T_s = 0.02$ seconds. An initial guess of $\hat{\alpha}_{init} = 1000$ was used for the self-tuning process, and it needed 12 iterations to achieve the desired change rate of less than 0.1%. It is possible to see the new initial estimative for the α parameter for each iteration in Figure 28 and all relevant parameters for the tuned iP controller in Table 3.

The simulated vehicle speed for the iPα controller can be seen in Figure 29a, together with the result from the iP controller tuned in section 4.2. It is possible to see that the iPα controller also managed to properly control the vehicle and follow the reference speed with a good dynamic response, having similar issues as the iP at low speed (around 20 km/h). The α parameter estimation can be seen in Figure 29b.

For the step reference and the iPα controller, the estimated F parameter can be seen in Figure 30a, the speed error can be seen in Figure 30b, the control effort applied to

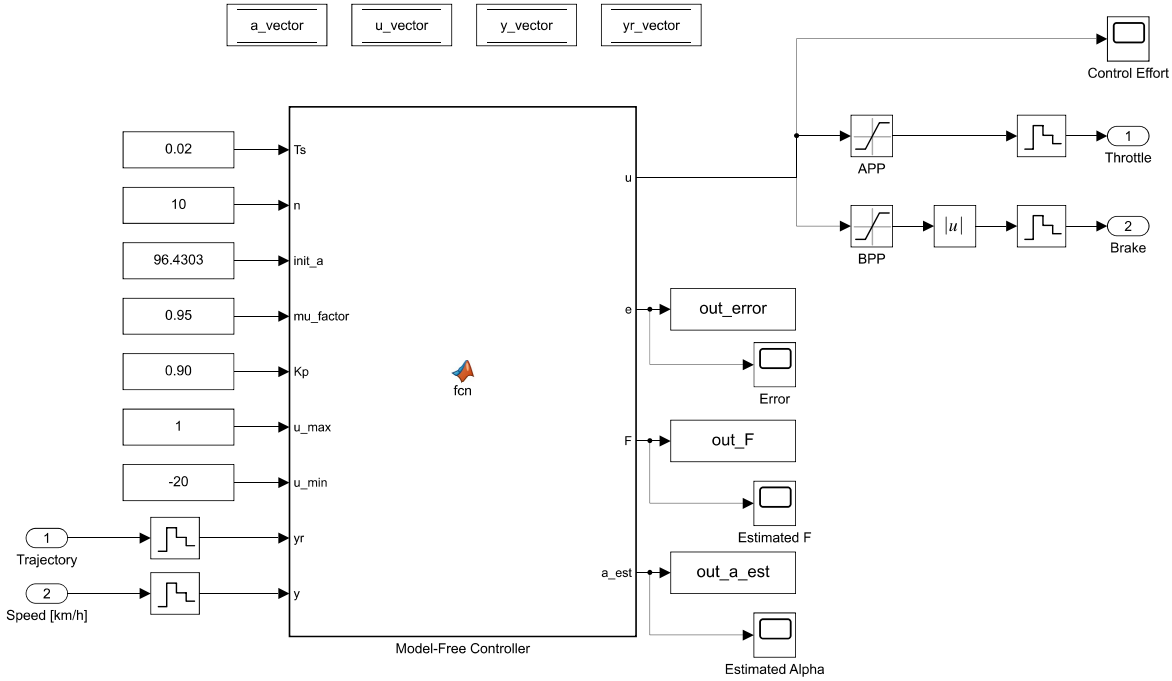


Figure 27: Simulink model with the developed iPα controller.

Source: Author.

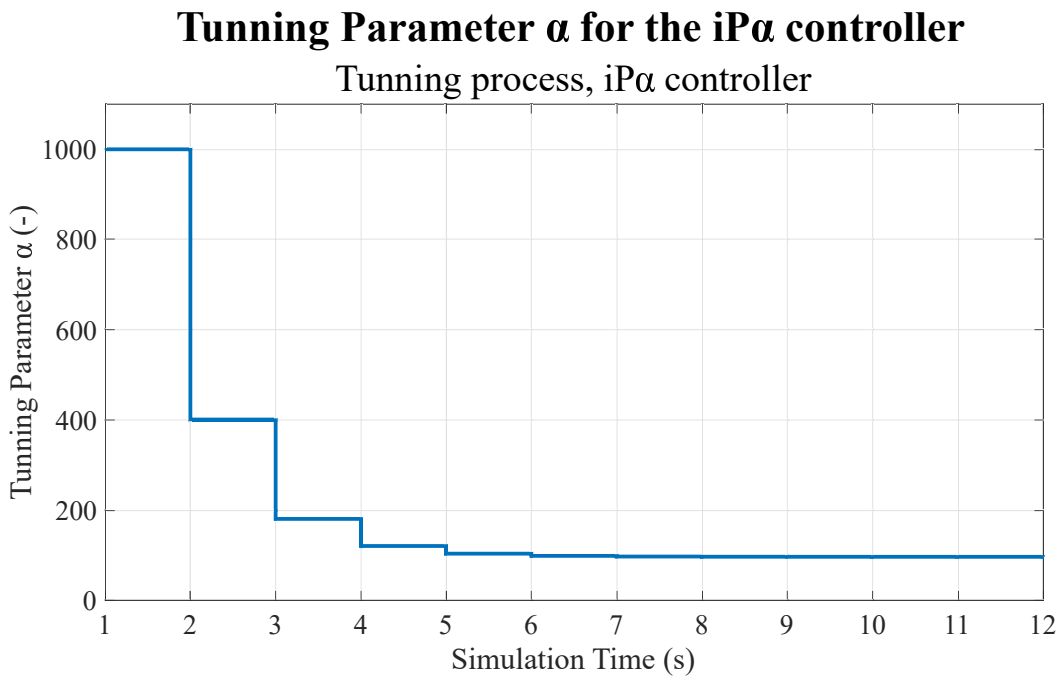


Figure 28: iPα controller estimated initial α during the self-tuning procedure.

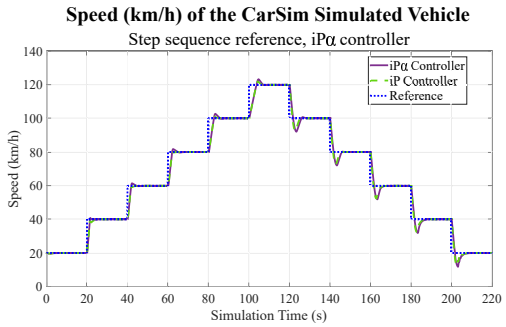
Source: Author.

the engine throttle is presented in Figure 30c and, lastly, the control effort applied to the brake pressure is presented in Figure 30d.

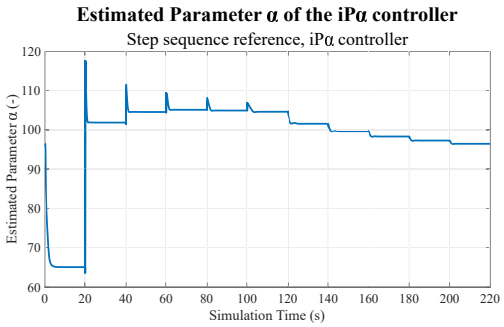
Table 3: Tuned iPα controller parameters for the CarSim vehicle simulation

Parameter	Description	Value	Unit
T_s	Sampling time	0.02	s
n	Integration window size	10	-
$\hat{\alpha}_{init}$	Initial estimative of ultra-local α parameter	96.4303	-
μ	Forgetting factor for α estimator	0.95	-
K_p	iP controller gain	0.90	-
u_{max}	Maximum controller output	1	-
u_{min}	Minimum controller output	-20	-

Source: Author.



(a) Speed reference (blue) and simulated vehicle speed for both iPα controller (purple) and iP controller (green) for the stairs reference with the CarSim model.



(b) Estimation of α parameter (blue) for the stairs reference with the CarSim model.

Figure 29: iPα controller result and α estimation for the stairs reference with the CarSim model.

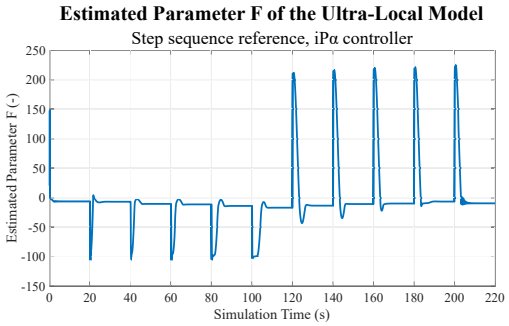
Source: Author.

The same controller was also applied to the speed reference presented in Figure 25a and the vehicle speed result of this scenario can be seen in Figure 31a. The α parameter estimation can be seen in Figure 31b.

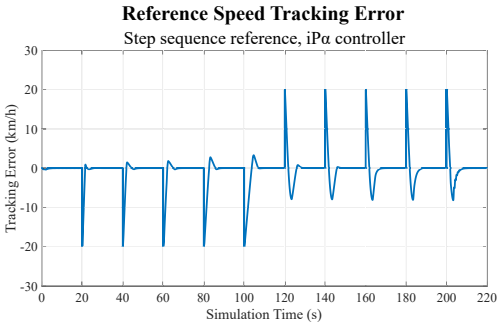
For this spline reference and the iPα controller, the estimated F parameter can be seen in Figure 32a, the speed error can be seen in Figure 32b, the control effort applied to the engine throttle is presented in Figure 32c and, lastly, the control effort applied to the brake pressure is presented in Figure 32d.

4.4 Euro NCAP Tests

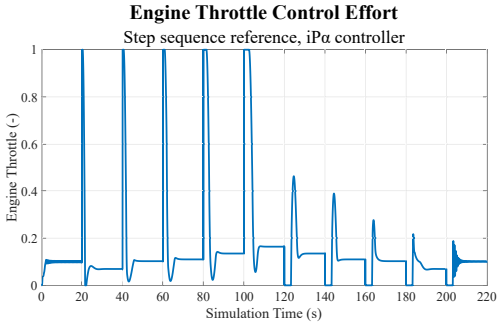
To verify how the iP and iPα controllers projected deals with emergency situations, CarSim preconfigured Euro NCAP test scenarios were used. Each of the controllers was



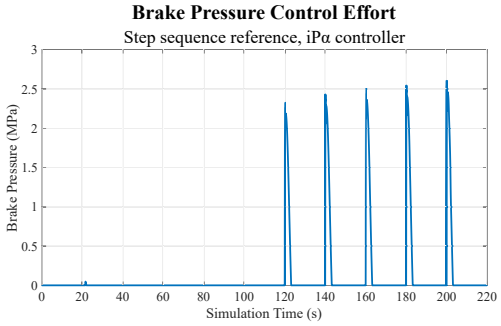
(a) Estimated parameter F (blue).



(b) Speed error (blue).



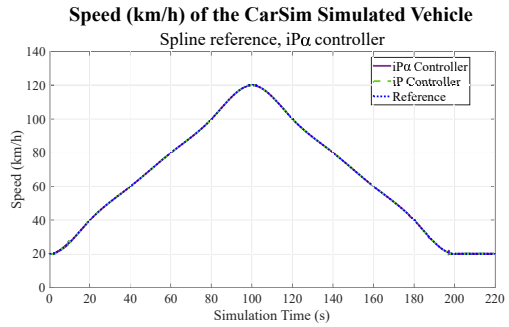
(c) Engine throttle control effort (blue).



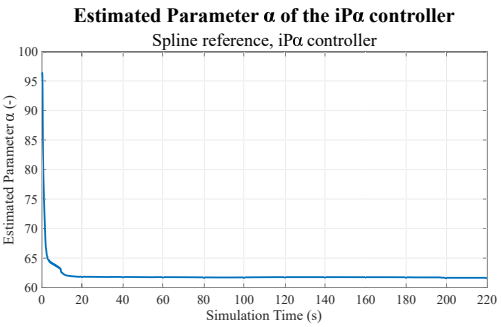
(d) Brake pressure control effort (blue).

Figure 30: Resulting iPα controller curves for the stairs reference with the CarSim model

Source: Author.



(a) Speed reference (blue) and simulated vehicle speed for both iPα controller (purple) and iP controller (green) for the spline reference with the CarSim model.



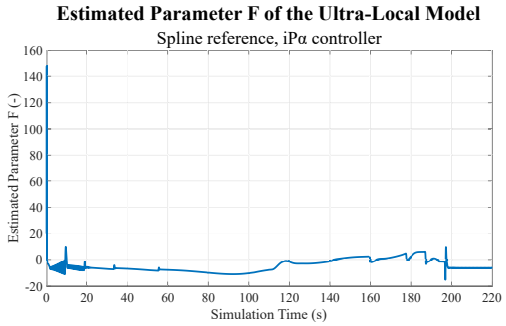
(b) Estimation of α parameter (blue) for the spline reference with the CarSim model.

Figure 31: iPα controller result and α estimation for the spline reference with the CarSim model.

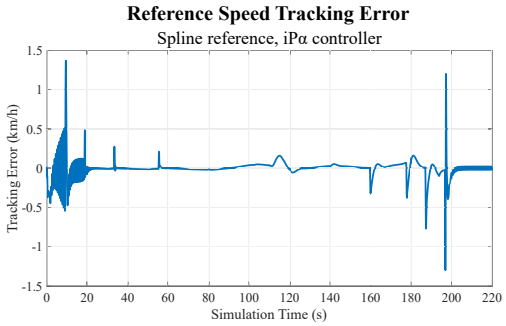
Source: Author.

used in the following test scenarios:

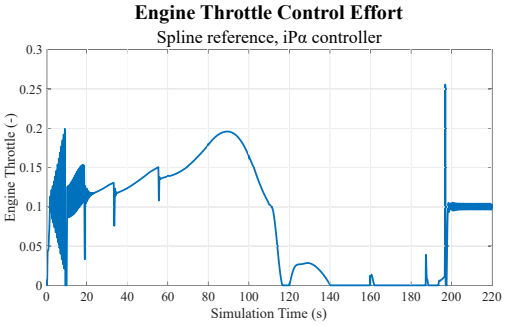
- ACC car-to-car rear braking with a deacceleration of $-2\ m/s^2$ and an initial distance of 40 m.



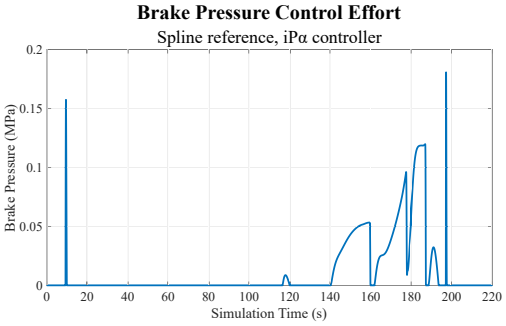
(a) Estimated parameter F (blue).



(b) Speed error (blue).



(c) Engine throttle control effort (blue).



(d) Brake pressure control effort (blue).

Figure 32: Resulting iPα controller curves for the spline reference with the CarSim model

Source: Author.

- ACC car-to-car rear braking with a deacceleration of $-2\ m/s^2$ and an initial distance of 12 m.
- ACC car-to-car rear braking with a deacceleration of $-6\ m/s^2$ and an initial distance of 40 m.
- ACC car-to-car rear braking with a deacceleration of $-6\ m/s^2$ and an initial distance of 12 m.
- ACC car-to-car cut-in with the vehicle under test at 50 km/h.
- ACC car-to-car cut-in with the vehicle under test at 120 km/h.
- ACC car-to-car cut-out with the vehicle under test at 70 km/h.
- ACC car-to-car cut-out with the vehicle under test at 90 km/h.
- ACC car-to-car simple slow-to-stop.
- ACC car-to-car simple slow-down.

At each of these tests, the CarSim simulation tries to detect any collision between the vehicle under test and the leading vehicle and gives a total NCAP score. If no collision happens, the score is 1 and any collision causes this score to be reduced. During the simulation, a short video is generated showing the vehicle under test movements with the most important information of the test scenario, such as the NCAP ACC score, vehicle throttle and speed, time to collision and relative distance to the leading vehicle. The screenshot presented in Figure 33 shows an example of the final frame of one of the Euro NCAP test generated video.

In general, the objective for all the executed tests is to avoid an accident (by either slowing down or completely stopping the vehicle as fast as possible). A curve with the CarSim generated speed reference and the vehicle speed is provided for each test and controller.

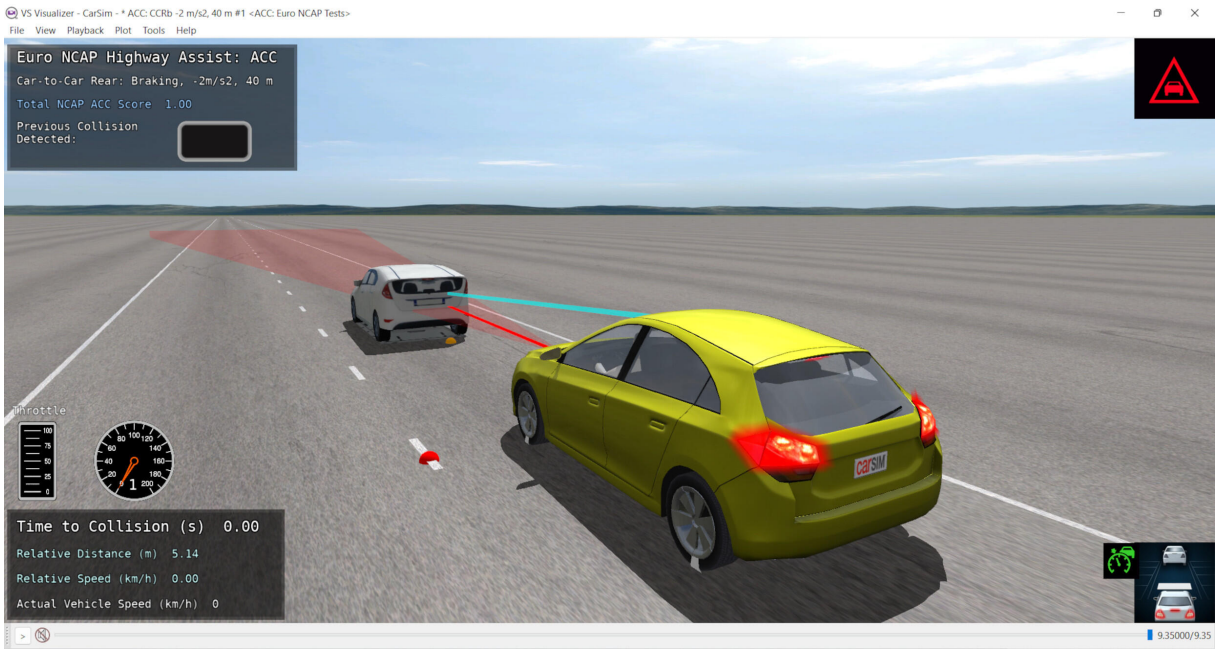


Figure 33: Final frame of the generated video for a CarSim Euro NCAP ACC test.

Source: Author.

4.4.0.1 iP Controller Euro NCAP Test Results

For the iP controller, the test results of the performed Euro NCAP ACC test are presented in Figures 34, 35, 36 and 37.

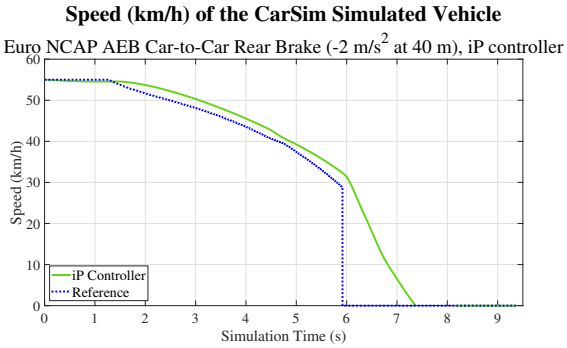
The ACC car-to-car rear braking with a deacceleration of $-2\ m/s^2$ and initial distance of 40 m, deacceleration of $-2\ m/s^2$ and initial distance of 12 m, deacceleration of $-6\ m/s^2$ and initial distance of 40 m and deacceleration of $-6\ m/s^2$ and initial distance of 12 m

test are, respectively, presented in Figures 34a, 34b, 34c and 34d.

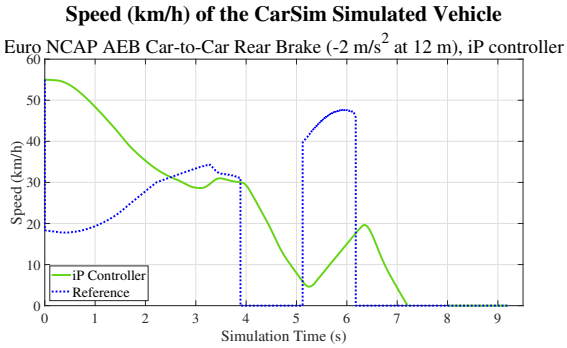
Both ACC car-to-car cut-in with the vehicle under test at 50 km/h and at 120 km/h tests are, respectively, presented in Figures 35a and 35b.

Both ACC car-to-car cut-out with the vehicle under test at 70 km/h and at 90 km/h tests are, respectively, presented in Figures 36a and 36b.

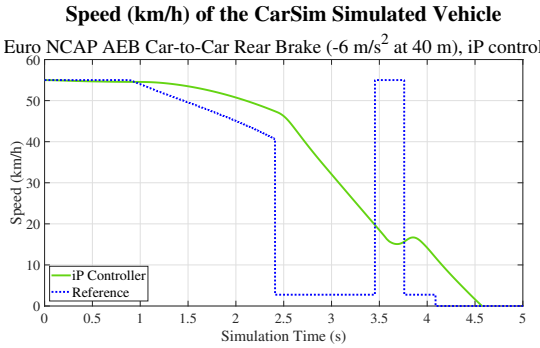
Lastly, the ACC car-to-car simple slow-to-stop and simple slow-down tests are, respectively, presented in Figures 37a and 37b.



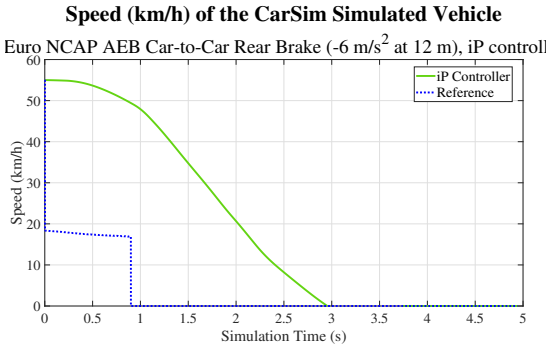
(a) ACC car-to-car rear braking with a deacceleration of $-2\ m/s^2$ and an initial distance of 40 m.



(b) ACC car-to-car rear braking with a deacceleration of $-2\ m/s^2$ and an initial distance of 12 m.



(c) ACC car-to-car rear braking with a deacceleration of $-6\ m/s^2$ and an initial distance of 40 m.



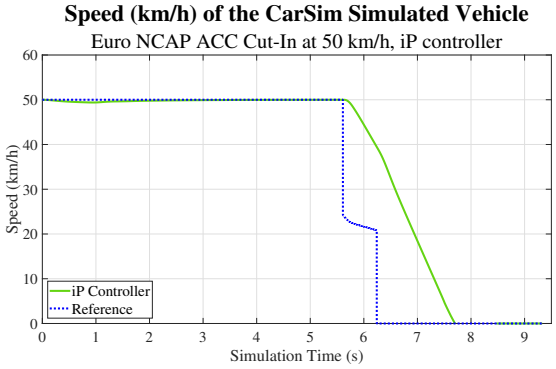
(d) ACC car-to-car rear braking with a deacceleration of $-6\ m/s^2$ and an initial distance of 12 m.

Figure 34: Speed reference (blue) and simulated vehicle speed (green) for the CarSim Euro NCAP ACC car-to-car rear braking tests, for the iP controller.

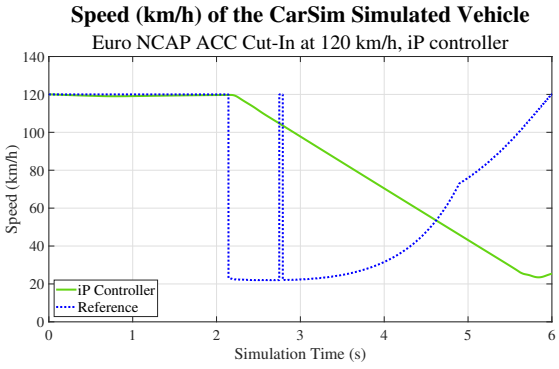
Source: Author.

In all the CarSim Euro NCAP tests performed with the iP controller, the simulated vehicle was able to avoid crashing into the leading car and causing an accident. This shows that the iP controller is perfectly able to maintain the car speed and react to emergencies, being suitable for longitudinal vehicular control.

It is interesting to note that the CarSim internally generated speed reference has



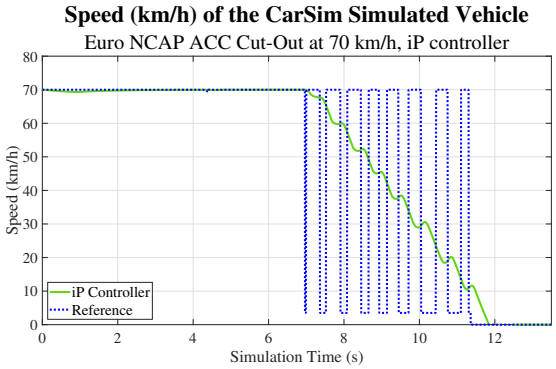
(a) ACC car-to-car cut-in with the vehicle under test at 50 km/h.



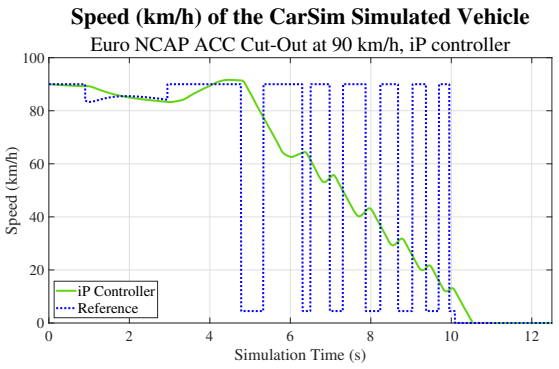
(b) ACC car-to-car cut-in with the vehicle under test at 120 km/h.

Figure 35: Speed reference (blue) and simulated vehicle speed (green) for the CarSim Euro NCAP ACC car-to-car cut-in tests, for the iP controller.

Source: Author.



(a) ACC car-to-car cut-out with the vehicle under test at 70 km/h.



(b) ACC car-to-car cut-out with the vehicle under test at 90 km/h.

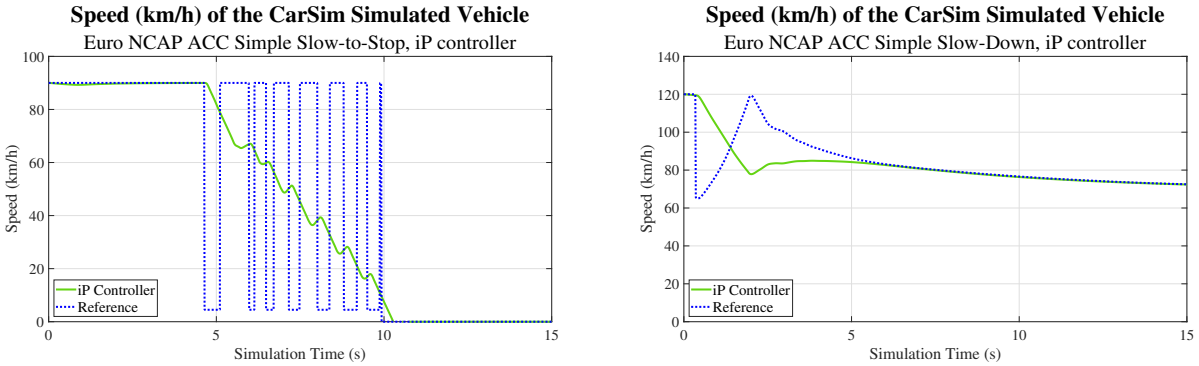
Figure 36: Speed reference (blue) and simulated vehicle speed (green) for the CarSim Euro NCAP ACC car-to-car cut-out tests, for the iP controller.

Source: Author.

a somewhat strange behavior in some of the tests. This is caused by its own control algorithm, and there is no way to modify this behavior without completely substituting the entire ACC in the simulations (which is not in the scope of this work).

4.4.0.2 iP Controller with α Estimator Euro NCAP Test Results

To allow for an easier comparison between the iP and iPα controller, the results of all tests performed with the iPα controller are presented together with the iP ones and the CarSim generated speed reference. It is expected for the iPα to be able to better follow the speed reference and slow down or stop the car sooner than the iP one.



(a) ACC car-to-car simple slow-to-stop. (b) ACC car-to-car simple slow-down.

Figure 37: Speed reference (blue) and simulated vehicle speed (green) for several CarSim Euro NCAP ACC car-to-car simple slow-to-stop and simple slow-down tests, for the iP controller.

Source: Author.

For the iPα controller, the test results of the performed Euro NCAP ACC test are presented in Figures 38, 39, 40 and 41.

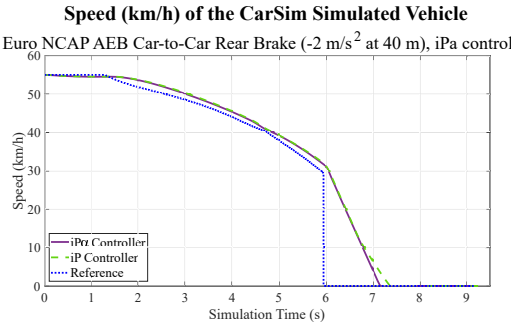
The ACC car-to-car rear braking with a deacceleration of $-2\ m/s^2$ and initial distance of 40 m, deacceleration of $-2\ m/s^2$ and initial distance of 12 m, deacceleration of $-6\ m/s^2$ and initial distance of 40 m and deacceleration of $-6\ m/s^2$ and initial distance of 12 m test are, respectively, presented in Figures 38a, 38b, 38c and 38d.

Both ACC car-to-car cut-in with the vehicle under test at 50 km/h and at 120 km/h tests are, respectively, presented in Figures 39a and 39b.

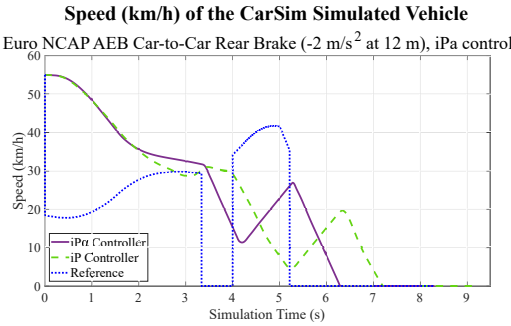
Both ACC car-to-car cut-out with the vehicle under test at 70 km/h and at 90 km/h tests are, respectively, presented in Figures 40a and 40b.

Lastly, the ACC car-to-car simple slow-to-stop and simple slow-down tests are, respectively, presented in Figures 41a and 41b.

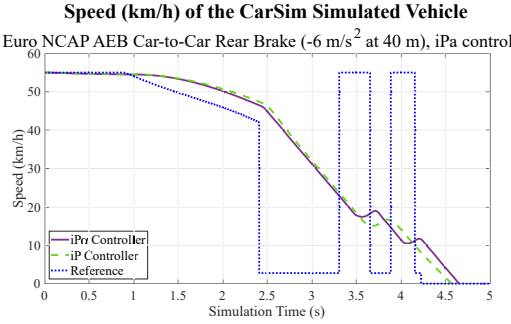
Just like with the iP, the simulated vehicle was able to avoid crashing into the leading car and causing an accident in all the CarSim's Euro NCAP tests performed with the iPα controller. The iPα also appears to have managed to better follow the reference curve in some test scenarios, probably due to the estimation of the α parameter. Unfortunately, since the reference curve is calculated during the simulation, it can change considerably for the same test scenario just by changing the controller, which makes a direct comparison between both controllers results hard to perform.



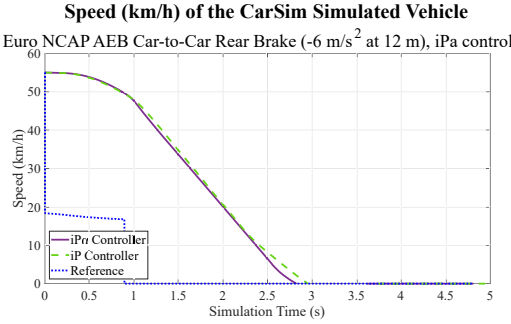
(a) ACC car-to-car rear braking with a deacceleration of $-2\ m/s^2$ and an initial distance of 40 m.



(b) ACC car-to-car rear braking with a deacceleration of $-2\ m/s^2$ and an initial distance of 12 m.



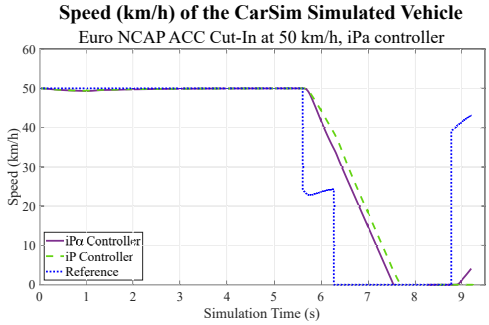
(c) ACC car-to-car rear braking with a deacceleration of $-6\ m/s^2$ and an initial distance of 40 m.



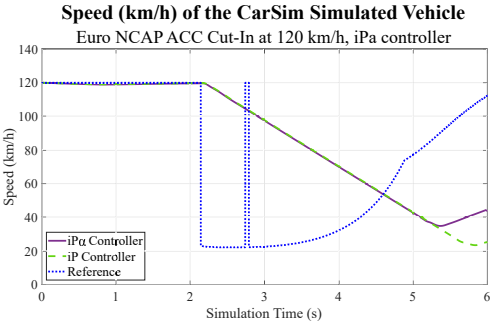
(d) ACC car-to-car rear braking with a deacceleration of $-6\ m/s^2$ and an initial distance of 12 m.

Figure 38: Speed reference (blue) and simulated vehicle speed for both iPα controller (purple) and iP controller (green) for the CarSim Euro NCAP ACC car-to-car rear braking tests, for the iPα controller.

Source: Author.



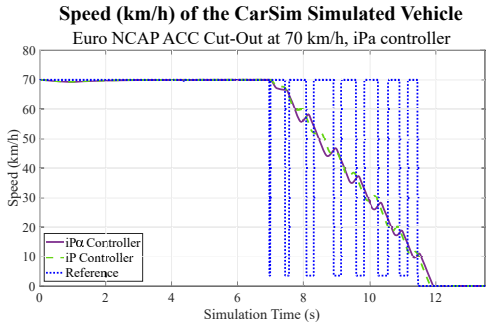
(a) ACC car-to-car cut-in with the vehicle under test at 50 km/h.



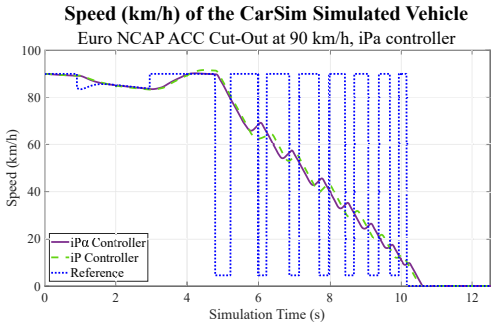
(b) ACC car-to-car cut-in with the vehicle under test at 120 km/h.

Figure 39: Speed reference (blue) and simulated vehicle speed for both iPα controller (purple) and iP controller (green) for the CarSim Euro NCAP ACC car-to-car cut-in tests, for the iPα controller.

Source: Author.



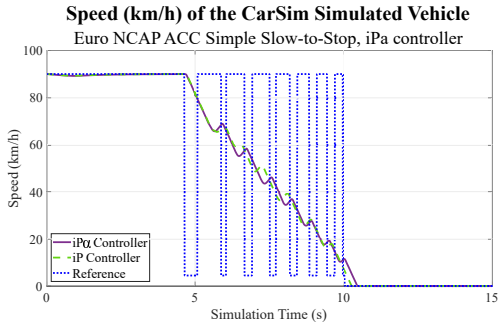
(a) ACC car-to-car cut-out with the vehicle under test at 70 km/h.



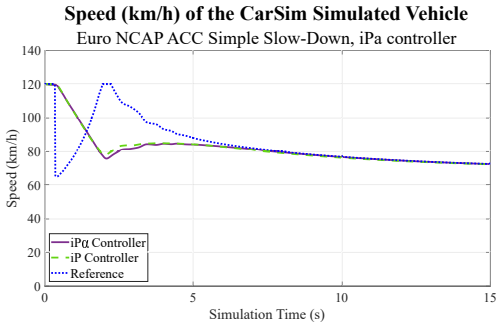
(b) ACC car-to-car cut-out with the vehicle under test at 90 km/h.

Figure 40: Speed reference (blue) and simulated vehicle speed for both iPα controller (purple) and iP controller (green) for the CarSim Euro NCAP ACC car-to-car cut-out tests, for the iPα controller.

Source: Author.



(a) ACC car-to-car simple slow-to-stop.



(b) ACC car-to-car simple slow-down.

Figure 41: Speed reference (blue) and simulated vehicle speed for both iPα controller (purple) and iP controller (green) for the CarSim Euro NCAP ACC car-to-car simple slow-to-stop and simple slow-down tests, for the iPα controller.

Source: Author.

4.4.0.3 Euro NCAP Test Results Summary

Both implemented controllers managed to properly control the simulated vehicle and pass all Euro NCAP tests they were submitted to. But it is difficult to directly compare their final results due to the influence of the high-level speed reference control generated by an unknown CarSim internal controller.

Ultimately, the Euro NCAP tests are meant to test how safe the VUT is in emergency situations. If we consider that for the same emergency situation the better controller is the one that manages to slow down or stops the vehicle the fastest, then it becomes possible to try to perform a more direct comparison between them by comparing the final relative distance of the simulated vehicles in the Euro NCAP tests.

The final relative distances for all executed Euro NCAP tests with the two controllers can be seen in the Table 4. In all the ACC car-to-car braking tests, the iPα controller was able to achieve a slightly higher final relative distance, indicating that the iPα controller can be more powerful than the iP one in these king of emergency situations. For the ACC car-to-car cut-in, car-to-car cut-out, simple slow-to-stop and simple slow-down the results are mixed, not really indicating that one of the controllers performed much better than the other.

In general, these results show that the proposed iPα controller not only can perform at the same level of the iP one, but even better in some situations. Which is a specially interesting result when considering that it has a more simplified and automatic tuning method.

Table 4: Euro NCAP tests final relative distance for the CarSim simulated vehicle, for both iP and iPα controllers

Euro NCAP test description	iP Controller	iPα Controller
ACC car-to-car rear braking, $-2m/s^2$ deceleration at 40 m	5.14 m	5.75 m
ACC car-to-car rear braking, $-2m/s^2$ deceleration at 12 m	7.51 m	8.99 m
ACC car-to-car rear braking, $-6m/s^2$ deceleration at 40 m	9.24 m	9.32 m
ACC car-to-car rear braking, $-6m/s^2$ deceleration at 12 m	3.48 m	4.02 m
ACC car-to-car cut-in, VUT at 50 km/h	9.73 m	11.01 m
ACC car-to-car cut-in, VUT at 120 km/h	29.84 m	27.55 m
ACC car-to-car cut-out, VUT at 70 km/h	9.13 m	9.25 m
ACC car-to-car cut-out, VUT at 90 km/h	9.27 m	9.32 m
ACC car-to-car simple slow-to-stop	9.56 m	9.31 m
ACC car-to-car simple slow-down	35.99 m	36.15 m

Source: Author.

5 CONCLUSIONS AND FUTURE WORKS

All tests with the simulated vehicle showed positive results, demonstrating the applicability of iPID MFC and iPIDα AMFC for vehicular applications.

The discrete implementation presented for the iP controller performed well and can easily be applied to embedded systems. Tuning the PI controllers was relatively simple and intuitive, performing well with little effort. The fact that it can almost automatically convert the gains from a PI to an iP means that switching from one controller to another requires little effort on the part of the designer, making it possible to replace PI controllers with iP ones in noisy environments, potentially achieving better results. Comparison with an equivalent PI shows that it is possible to obtain better results with the iP when dealing with varying environmental conditions (such as road incline for a vehicle). And without the need for integrators or an anti-windup mechanism, which also indicates that the iP could achieve better results than PI on systems with saturation problems.

The proposed α estimator and iPIDα AMFC controller achieved good results, being able to not only auto-tune itself to some extend but to have a comparable performance with the iPID MFC (if not better is some test scenarios). Its digital implementation is relatively simple and light, making it appropriated to be used in embedded or real-time systems. It also has the interesting characteristic of having essentially the same implementation for a ultra-local model with any ν.

Tests executed with a complex vehicle simulator showed that both implemented controllers were able to control the longitudinal speed of the vehicle and protect the driver and passengers in emergency situations. The Euro NCAP can be considered the gold standard for vehicle safety and the controllers managed to pass on all executed tests.

The way the controller's output was used also shows their ability to work with different and changing model dynamics by controlling both the engine throttle and the brake pressure at the same time. The very asymmetrical levels of the possible controller output for them also creates a interesting challenge for speed controllers

Some of the possible future works with the develop controllers are:

- Implementation of the designed controllers in a Hardware-in-the-loop (HIL) that integrates real vehicle parts and the CarSim software for further testing.
- Implementation and testing of the designed controllers in a vehicle test platform.
- More testing and improvements for the proposed α estimator.
- Comparison between the proposed iPIDα and others AMFC presented in literature.
- Development of the ACC high-level controller and integration with the implemented low-level controller.

REFERENCES

AMES, A. D.; GRIZZLE, J. W.; TABUADA, P. Control barrier function based quadratic programs with application to adaptive cruise control. In: IEEE. **53rd IEEE Conference on Decision and Control**. [S.l.], 2014. p. 6271–6278.

AOKI, Y. et al. **Development of hydraulic servo brake system for cooperative control with regenerative brake**. [S.l.], 2007.

ÅSTRÖM, K. J.; HÄGGLUND, T. **PID controllers: Theory, Design, and Tuning**. [S.l.]: ISA - The Instrumentation, Systems and Automation Society, 1995. ISBN 1-55617-516-7.

BARA, O. et al. Model-free load control for high penetration of solar photovoltaic generation. In: IEEE. **2017 North American Power Symposium (NAPS)**. [S.l.], 2017. p. 1–6.

BARTH, J. M. et al. Full model-free control architecture for hybrid uavs. In: IEEE. **2019 American control conference (ACC)**. [S.l.], 2019. p. 71–78.

BENALIE, N. et al. Improvement of adaptive cruise control system based on speed characteristics and time headway. In: IEEE. **2009 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.], 2009. p. 2403–2408.

BRUGNOLLI, M. M. **Predictive adaptive cruise control in an embedded environment.** Tese (Doutorado) — Universidade de São Paulo, 2018.

CARSIM Overview. Mechanical Simulation Corporation, 2022. Accessed on: June, 02 - 2022. Available from: <https://www.carsim.com/products/carsim/index.php>.

CHIEN, C.; IOANNOU, P.; LAI, M. Entrainment and vehicle following controllers design for autonomous intelligent vehicles. In: IEEE. **Proceedings of 1994 American Control Conference-ACC'94**. [S.l.], 1994. v. 1, p. 6–10.

DELALEAU, E. A proof of stability of model-free control. In: IEEE. [S.l.], 2014. p. 1–7.

D'ALFIO, N.; MORGANDO, A.; SORNIOTTI, A. Electro-hydraulic brake systems: design and test through hardware-in-the-loop simulation. **Vehicle System Dynamics**, Taylor & Francis, v. 44, n. sup1, p. 378–392, 2006.

D'ANDRÉA-NOVEL, B. et al. Some remarks on wheeled autonomous vehicles and the evolution of their control design. **IFAC-PapersOnLine**, Elsevier, v. 49, n. 15, p. 199–204, 2016.

FANCHER, P.; BAREKET, Z.; ERVIN, R. Human-centered design of an acc-with-braking and forward-crash-warning system. **Vehicle System Dynamics**, Taylor & Francis, v. 36, n. 2-3, p. 203–223, 2001.

FANCHER, P. et al. Relationships between manual driving and driving with adaptive cruise control. In: **Proceedings of AVEC2004, 7th International Symposium on Advanced Vehicle Control**. [S.l.: s.n.], 2004. p. 23–27.

FLIESS, M. Model-free control and intelligent pid controllers: towards a possible trivialization of nonlinear control? **IFAC Proceedings Volumes**, Elsevier, v. 42, n. 10, p. 1531–1550, 2009.

FLIESS, M.; JOIN, C. Intelligent pid controllers. In: IEEE. **2008 16th Mediterranean Conference on Control and Automation**. [S.l.], 2008. p. 326–331.

FLIESS, M.; JOIN, C. Model-free control. **International Journal of Control**, Taylor & Francis, v. 86, n. 12, p. 2228–2252, 2013.

GANJI, B. et al. Adaptive cruise control of a hev using sliding mode control. **Expert systems with applications**, Elsevier, v. 41, n. 2, p. 607–615, 2014.

GOÑI-ROS, B. et al. Using advanced adaptive cruise control systems to reduce congestion at sags: An evaluation based on microscopic traffic simulation. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 102, p. 411–426, 2019.

HAC, A. **Effects of brake actuator error on vehicle dynamics and stability**. [S.l.], 2005.

HAN, W.; XIONG, L.; YU, Z. Braking pressure control in electro-hydraulic brake system based on pressure estimation with nonlinearities and uncertainties. **Mechanical Systems and Signal Processing**, Elsevier, v. 131, p. 703–727, 2019.

HE, Y. et al. Adaptive cruise control strategies implemented on experimental vehicles: A review. **IFAC-PapersOnLine**, Elsevier, v. 52, n. 5, p. 21–27, 2019.

HIROTA, N. et al. **Development of variable technology in performance of brake booster**. [S.l.], 2004.

HU, L. et al. The injury epidemiology of adult riders in vehicle-two-wheeler crashes in china, ningbo, 2011–2015. **Journal of safety research**, Elsevier, v. 72, p. 21–28, 2020.

HU, L. et al. A review of research on traffic conflicts based on intelligent vehicles. **Ieee Access**, IEEE, v. 8, p. 24471–24483, 2020.

ISERMANN, R. et al. Nonlinear distance and cruise control for passenger cars. In: **Advances in Automotive Control 1995**. [S.l.]: Elsevier, 1995. p. 209–214.

JOIN, C.; CHAXEL, F.; FLIESS, M. "intelligent" controllers on cheap and small programmable devices. In: IEEE. **2013 Conference on Control and Fault-Tolerant Systems (SysTol)**. [S.l.], 2013. p. 554–559.

JONNER, W.-D. et al. Electrohydraulic brake system-the first approach to brake-by-wire technology. **SAE transactions**, JSTOR, p. 1368–1375, 1996.

JOST, G. et al. Road safety target outcome: 100,000 fewer deaths since 2001. **5th Road Safety PIN Report**, 2010.

KASAC, J.; MAJETIC, D.; BREZAK, D. An algebraic approach to on-line signal denoising and derivatives estimation. **Journal of the Franklin Institute**, v. 355, n. 15, p. 7799–7825, 2018. ISSN 0016-0032. Available from: <https://www.sciencedirect.com/science/article/pii/S0016003218305490>.

KULLGREN, A. et al. Developments in car crash safety and comparisons between results from euro ncap tests and real-world crashes. In: . [S.l.: s.n.], 2019.

KULLGREN, A.; LIE, A.; TINGVALL, C. Comparison between euro ncap test results and real-world crash data. **Traffic Injury Prevention - TRAFFIC INJ PREV**, Taylor & Francis, v. 11, p. 587–593, 10 2010.

LEIBER, T.; KÖGLSPERGER, C.; UNTERFRAUNER, V. Modular brake system with integrated functionalities. **Auto Tech Review**, Springer, v. 1, n. 6, p. 24–29, 2012.

LIE, A.; KULLGREN, A.; TINGVALL, C. Comparison of euro ncap test results with folksam car model safety ratings. In: . [S.l.: s.n.], 2001.

LIE, A.; TINGVALL, C. How do euro ncap results correlate with real-life injury risks? a paired comparison study of car-to-car crashes. **Traffic Injury Prevention - TRAFFIC INJ PREV**, Taylor & Francis, v. 3, p. 288–293, 10 2002.

LIU, Y.; CHE, J.; CAO, C. Advanced autonomous underwater vehicles attitude control with l 1 backstepping adaptive control strategy. **Sensors**, MDPI, v. 19, n. 22, p. 4848, 2019.

MATHWORKS. **Complete Vehicle Model**. MathWorks, 2022. Acessado em: 10 de Março de 2022. Available from: <https://www.mathworks.com/help//physmod/sdl/ug/about-the-complete-vehicle-model.html>.

MATHWORKS. **Modeling a Vehicle Dynamics System**. MathWorks, 2022. Acessado em: 10 de Março de 2022. Available from: <https://www.mathworks.com/help/ident/ug/modeling-a-vehicle-dynamics-system.html>.

MENHOUR, L. et al. Multivariable decoupled longitudinal and lateral vehicle control: A model-free design. In: IEEE. **52nd IEEE Conference on Decision and Control**. [S.l.], 2013. p. 2834–2839.

MENHOUR, L. et al. A new model-free design for vehicle control and its validation through an advanced simulation platform. In: IEEE. **2015 European Control Conference (ECC)**. [S.l.], 2015. p. 2114–2119.

MILANÉS, V. et al. Electro-hydraulic braking system for autonomous vehicles. **International Journal of Automotive Technology**, Springer, v. 11, n. 1, p. 89–95, 2010.

MILANÉS, V. et al. Comparing fuzzy and intelligent pi controllers in stop-and-go manoeuvres. **IEEE Transactions on Control Systems Technology**, IEEE, v. 20, n. 3, p. 770–778, 2012.

MILANÉS, V. et al. Low-speed longitudinal controllers for mass-produced cars: A comparative study. **IEEE Transactions on Industrial Electronics**, IEEE, v. 59, n. 1, p. 620–628, 2012.

MIRAS, J. D. et al. Active magnetic bearing: A new step for model-free control. In: IEEE. **52nd IEEE Conference on Decision and Control**. [S.l.], 2013. p. 7449–7454.

MOON, S.; YI, K. Human driving data-based design of a vehicle adaptive cruise control algorithm. **Vehicle System Dynamics**, Taylor & Francis, v. 46, n. 8, p. 661–690, 2008.

MORAES, M. S.; SILVA, P. S. P. da. Model-free control of magnetic levitation systems through algebraic derivative estimation. In: **Proc. 23rd ABCM Int. Congress of Mechanical Engineering, Rio de Janeiro**. [S.l.: s.n.], 2015.

NOVEL, B. d'Andréa. Model-free control of longitudinal and lateral dynamics for automated vehicles. **JTEKT Engin. J.**, v. 1015, p. 2–8, 2018.

PENG, Y. et al. Tunnel driving occupational environment and hearing loss in train drivers in china. **Occupational and environmental medicine**, BMJ Publishing Group Ltd, v. 76, n. 2, p. 97–104, 2019.

POLACK, P.; DELPRAT, S.; NOVEL, B. d'Andréa. Brake and velocity model-free control on an actual vehicle. **Control Engineering Practice**, Elsevier, v. 92, p. 104072, 2019.

RATINGEN, M. V. et al. The european new car assessment programme: A historical review. **Chinese Journal of Traumatology**, Chinese Medical Journals Publishing House Co., Ltd. 42 Dongsi Xidajie ..., v. 19, p. 63–69, 2016. ISSN 1008-1275. Available from: <https://www.sciencedirect.com/science/article/pii/S1008127516000110>.

SCHENK, D. E.; WELLS, R. L.; MILLER, J. E. **Intelligent braking for current and future vehicles**. [S.l.], 1995.

SHAKOURI, P.; CZECZOT, J.; ORDYS, A. Simulation validation of three nonlinear model-based controllers in the adaptive cruise control system. **Journal of Intelligent & Robotic Systems**, Springer, v. 80, n. 2, p. 207–229, 2015.

SHLADOVER, S. E. Longitudinal control of automated guideway transit vehicles within platoons. 1978.

SORNIOTTI, A. **Hardware in the loop for braking systems with anti-lock braking system and electronic stability program**. [S.l.], 2004.

SORNIOTTI, A.; REPICI, G. M. Hardware in the loop with electro-hydraulic brake systems. In: CITESEER. **9th WSEAS International Conference on Systems**. [S.l.], 2005.

VENHOVENS, P.; NAAB, K.; ADIPRASITO, B. Stop and go cruise control. **International journal of automotive technology**, The Korean Society of Automotive Engineers, v. 1, n. 2, p. 61–69, 2000.

WANG, H. et al. α-variable adaptive model free control of irehave upper-limb exoskeleton. **Advances in Engineering Software**, v. 148, p. 102872, 2020. ISSN 0965-9978. Available from: <https://www.sciencedirect.com/science/article/pii/S0965997820302738>.

WANG, Z.; WANG, J. Ultra-local model predictive control: A model-free approach and its application on automated vehicle trajectory tracking. **Control Engineering Practice**, Elsevier Ltd, v. 101, 8 2020. ISSN 09670661.

WANG, Z. et al. **Prototype of distributed electro-hydraulic braking system and its fail-safe control strategy**. [S.l.], 2013.

WU, W. et al. Adaptive cruise control strategy design with optimized active braking control algorithm. **Mathematical Problems in Engineering**, Hindawi, v. 2020, 2020.

XIONG, L. et al. **Analysis and design of dual-motor electro-hydraulic brake system**. [S.l.], 2014.

YANG, Y.; LI, G.; ZHANG, Q. A pressure-coordinated control for vehicle electro-hydraulic braking systems. **Energies**, MDPI, v. 11, n. 9, p. 2336, 2018.

YI, K. et al. Implementation and vehicle tests of a vehicle stop-and-go cruise control system. **Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering**, Sage Publications Sage UK: London, England, v. 216, n. 7, p. 537–544, 2002.

YU, Z. et al. **An integrated-electro-hydraulic brake system for active safety**. [S.l.], 2016.

ZHENG, H.; ZHAO, M. **Development a HIL test bench for electrically controlled steering system**. [S.l.], 2016.

APPENDIX A – IP MFC MATLAB CODE

The iP MFC controller MATLAB code developed and used to execute all simulations is presented fully in Figure 42.

Editor - Block: ACC_SYSTEM_CS9/Model Free Controller/Model-Free Controller

Model Free Controller/Model-Free Controller

```
function [u, e, F] = fcn(Ts, n, a, Kp, u_max, u_min, yr, y)
global u_vector;
global y_vector;
global yr_vector;

% Get Global Values
ctrl_cnt = yr_vector(20);

% Estimation of F using ALIEN filter using Simpson 1/3 rule
Iy = n*(y_vector(n) - y);
Iu = ((3 - ((-1)^(n/2)))/4)*Ts*(n^2)*u_vector(n/2);
if (n > 2)
    for k = 1:((n-2)/2)
        Iy = Iy + (3 - ((-1)^k))*((n - 2*k)*(y_vector(n-k) - y_vector(k)));
        Iu = Iu + (3 - ((-1)^k))*(Ts*k*(n - k)*(u_vector(n-k) + u_vector(k)));
    end
end
Iy = -2/((n^3)*Ts) * Iy;
Iu = -2/((n^3)*Ts) * Iu;
F = Iy + a*Iu;

% Estimation of dyr using ALIEN filter using Simpson 1/3 rule
Iyr = n*(yr_vector(n) - yr);
if (n > 2)
    for k = 1:((n-2)/2)
        Iyr = Iyr + (3 - ((-1)^k))*((n - 2*k)*(yr_vector(n-k) - yr_vector(k)));
    end
end
Iyr = -2/((n^3)*Ts) * Iyr;

% Derivate of Yr
dyr = Iyr;

% Calculate error
e = yr - y;

% Model Free Control calculation
u = (dyr - F)/a + (Kp * e)/a;

% Saturate Output
if (u > u_max)
    u = u_max;
elseif (u < u_min)
    u = u_min;
end

% Block Controller Ouput in indeterminated state
if (ctrl_cnt < (n+1))
    u = 0;
    ctrl_cnt = ctrl_cnt + 1;
end

% Update global variables
for k = length(u_vector):-1:2
    u_vector(k) = u_vector(k-1);
end
u_vector(1) = u;
for k = length(y_vector):-1:2
    y_vector(k) = y_vector(k-1);
end
y_vector(1) = y;
for k = length(yr_vector):-1:2
    yr_vector(k) = yr_vector(k-1);
end
yr_vector(1) = yr;
yr_vector(20) = ctrl_cnt;

```

Figure 42: iP MFC controller MATLAB code.

Source: Author.

APPENDIX B – IPα AMFC MATLAB CODE

The iPα AMFC controller MATLAB code developed and used to execute all simulations is presented fully in Figures 43 and 44.

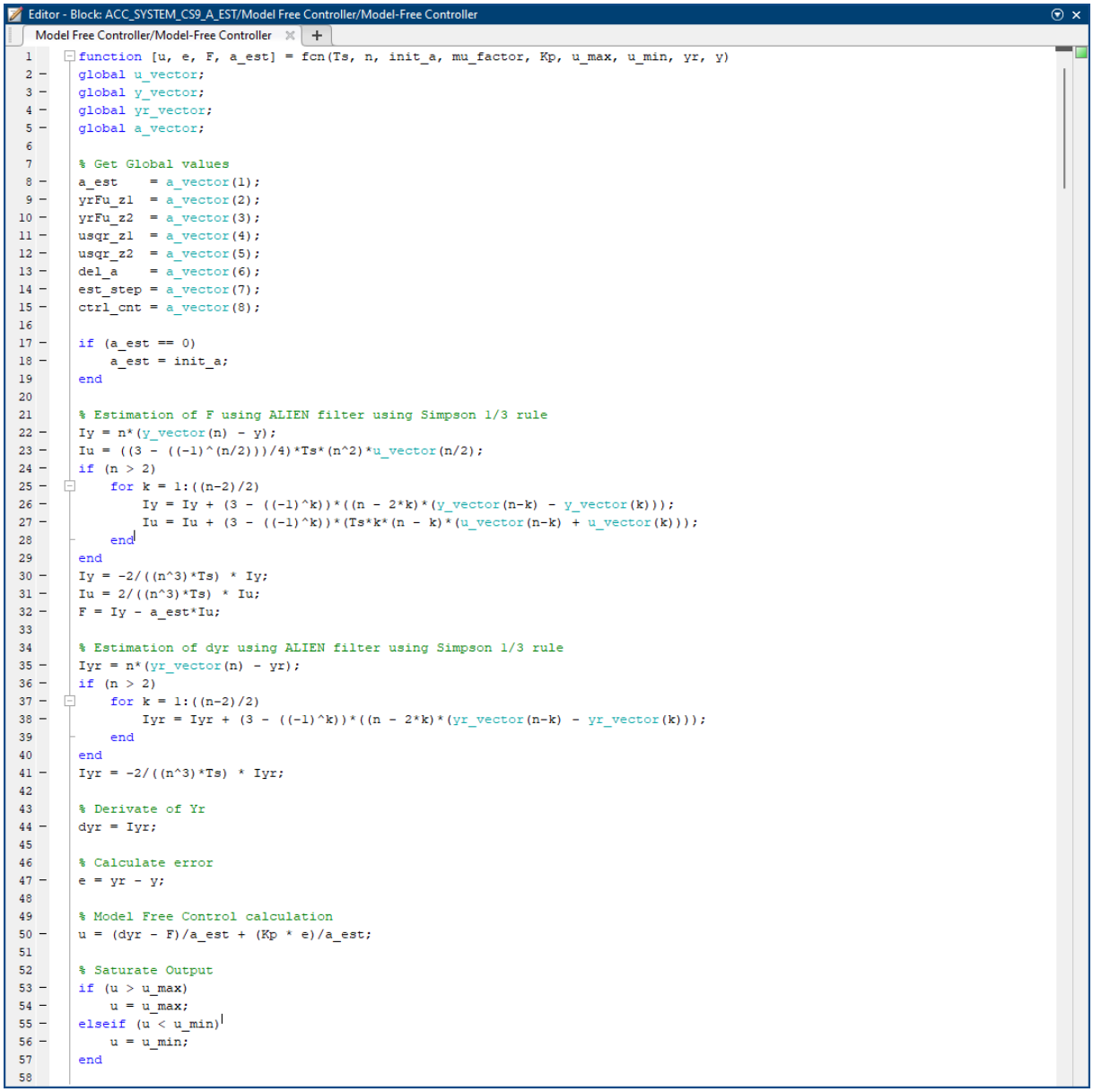
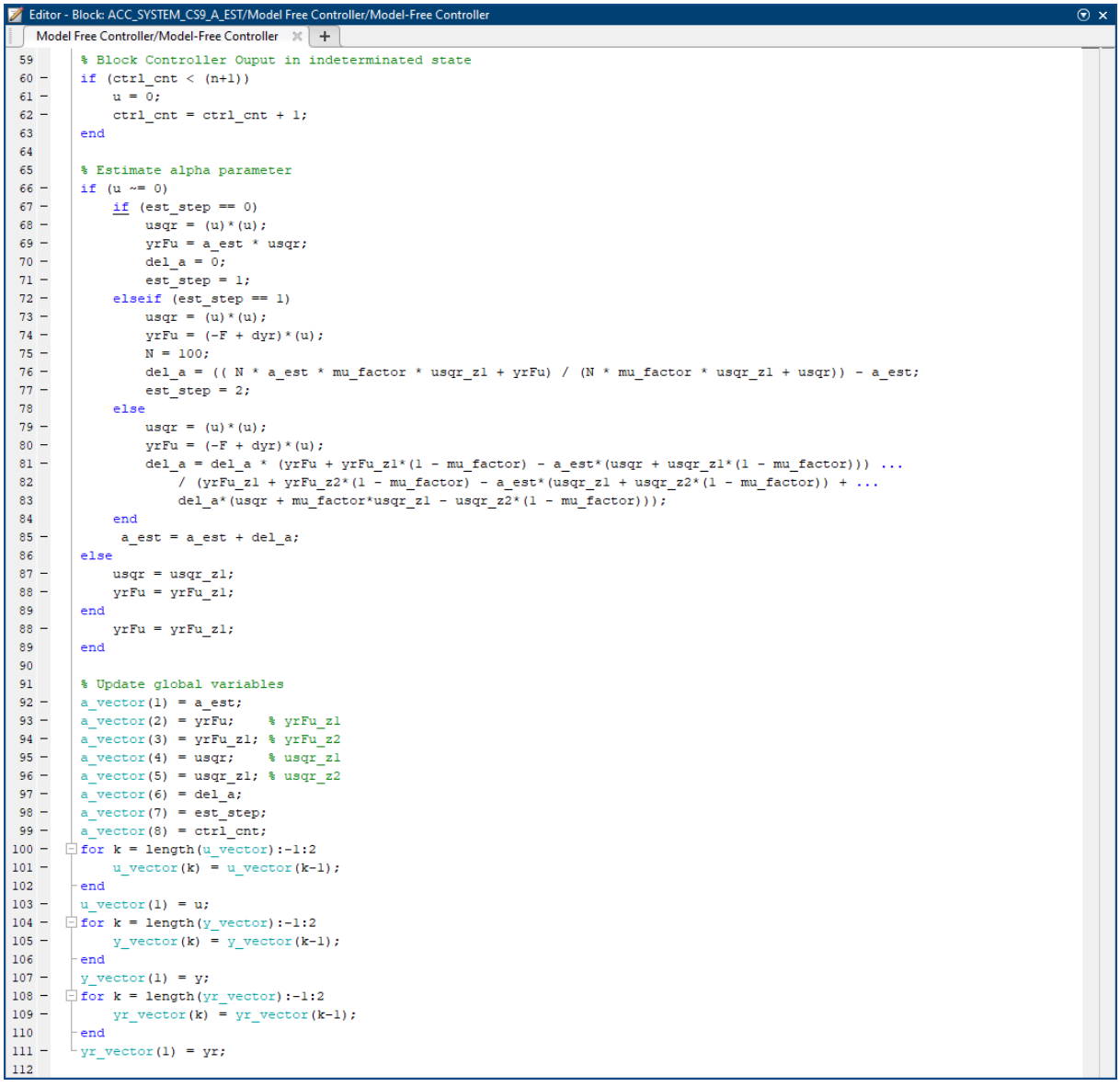


Figure 43: iPα AMFC controller MATLAB code - Part I.

Source: Author.




```
% Block Controller Ouput in indeterminated state
if (ctrl_cnt < (n+1))
    u = 0;
    ctrl_cnt = ctrl_cnt + 1;
end

% Estimate alpha parameter
if (u ~= 0)
    if (est_step == 0)
        usqr = (u)*(u);
        yrFu = a_est * usqr;
        del_a = 0;
        est_step = 1;
    elseif (est_step == 1)
        usqr = (u)*(u);
        yrFu = (-F + dyr)*(u);
        N = 100;
        del_a = (( N * a_est * mu_factor * usqr_z1 + yrFu) / (N * mu_factor * usqr_z1 + usqr)) - a_est;
        est_step = 2;
    else
        usqr = (u)*(u);
        yrFu = (-F + dyr)*(u);
        del_a = del_a * (yrFu + yrFu_z1*(1 - mu_factor) - a_est*(usqr + usqr_z1*(1 - mu_factor))) ...
            / (yrFu_z1 + yrFu_z2*(1 - mu_factor) - a_est*(usqr_z1 + usqr_z2*(1 - mu_factor)) + ...
            del_a*(usqr + mu_factor*usqr_z1 - usqr_z2*(1 - mu_factor)));
    end
     a_est = a_est + del_a;
else
    usqr = usqr_z1;
    yrFu = yrFu_z1;
end
    yrFu = yrFu_z1;
end

% Update global variables
a_vector(1) = a_est;
a_vector(2) = yrFu;    % yrFu_z1
a_vector(3) = yrFu_z1; % yrFu_z2
a_vector(4) = usqr;    % usqr_z1
a_vector(5) = usqr_z1; % usqr_z2
a_vector(6) = del_a;
a_vector(7) = est_step;
a_vector(8) = ctrl_cnt;
for k = length(u_vector):-1:2
    u_vector(k) = u_vector(k-1);
end
u_vector(1) = u;
for k = length(y_vector):-1:2
    y_vector(k) = y_vector(k-1);
end
y_vector(1) = y;
for k = length(yr_vector):-1:2
    yr_vector(k) = yr_vector(k-1);
end
yr_vector(1) = yr;
```

Figure 44: iPα AMFC controller MATLAB code - Part II.

Source: Author.