Int J Interact Des Manuf (2017) 11:859–879
DOI 10.1007/s12008-015-0265-4

TECHNICAL PAPER

Simulation NMPC in 2-HIL to design ECU

José Albeiro Valencia Chica · Adalberto Gabriel Díaz Torres

Received: 2 December 2014 / Accepted: 11 February 2015 / Published online: 25 February 2015

Abstract The development, testing and tuning of control systems for complex plants found hardware in the loop (HIL) technical, a perfect ally to reduce risks, costs and times redesign. HIL allows for example, in the automotive field contain an embedded system complexity nonlinear dynamics modeling the internal combustion engine, including the processes of discrete events and continuous. With the goal to represent as closely as possible the behavior of the engine, dynamics is simulated by the embedded system in real time, sensors including. On the other hand, the electronic control unit (ECU), by construction also constitutes an embedded system that the plant operates properly. Is of wide interest to optimize engine operation, and a valid opportunity, is to design ECU's that are running optimal control algorithms, such as nonlineal model predictive control (NMPC). This document is a report of the practical reliability of the implementation of a HIL simulation scheme for the design of NMPC controllers for internal combustion engine. In a RT hardware element implements the component models the nonlinear plant, in another such element is implemented and tested two NMPC optimal control algorithms: model predictive control with linearization on-line and NMPC based on sequential quadratic programming, included in the simulation loop the real actuator elements. Combined with an interface to the designer that allows actively interact with the system, evaluating an expanded field operating conditions and even bordering operating limits.

J. A. V. Chica (✉)
Servicio Nacional de Aprendizaje (SENA), Dg 104 # 69-120, Medellín, Colombia
e-mail: albeirovalencia@misena.edu.co

A. G. D. Torres
Universidad EAFIT, Cra. 49 #7sur-50, Medellín, Colombia
e-mail: gdiaz@eafit.edu.co

Keywords Simulation · Hardware in the loop · Automotive · Optimal control nonlinear · Real time control

1 Preface

Since environmental regulations given in the United States by the Environmental Protection Agency [1], especially at the end of the 70s, combustion engines have an associated electronic system that controls its operation, especially talking about the amount of fuel injected and ignition timing of the air/fuel mixture, seeking to reduce the toxic emissions, fuel consumption and get the best energy performance. To this one element is called electronic control unit (ECU) [2–4]. Efforts to improve the techniques that these devices to use are still working today, because environmental and economic requirements become more stringent. One possibility, which arises with the development of increased ability of the programmable electronic computing, allows developing complex optimization algorithms and exploits the inherent non-linearity, has the mathematical representation of the motor. In the current paper, the implementation of a simulation scheme hardware in the loop (HIL) using hardware for both the model plant and the control system is reported. A real-time processor working in parallel with an FPGA implements the nonlinear model of combustion engine, and a similar element implements a nonlinear model predictive control (MPC) controller, hence the term 2-HIL. Together with an interface operation and design, the end result is a contribution in the design stage to the development cycle engineering, specifically in combustion engines. Through the operator interface designer interacts with the system 2-HIL for adjusting optimal controllers, limit performance testing, experiment with changes in modeling parameters and optimization goals. This will incur the risk and costs of experience in the

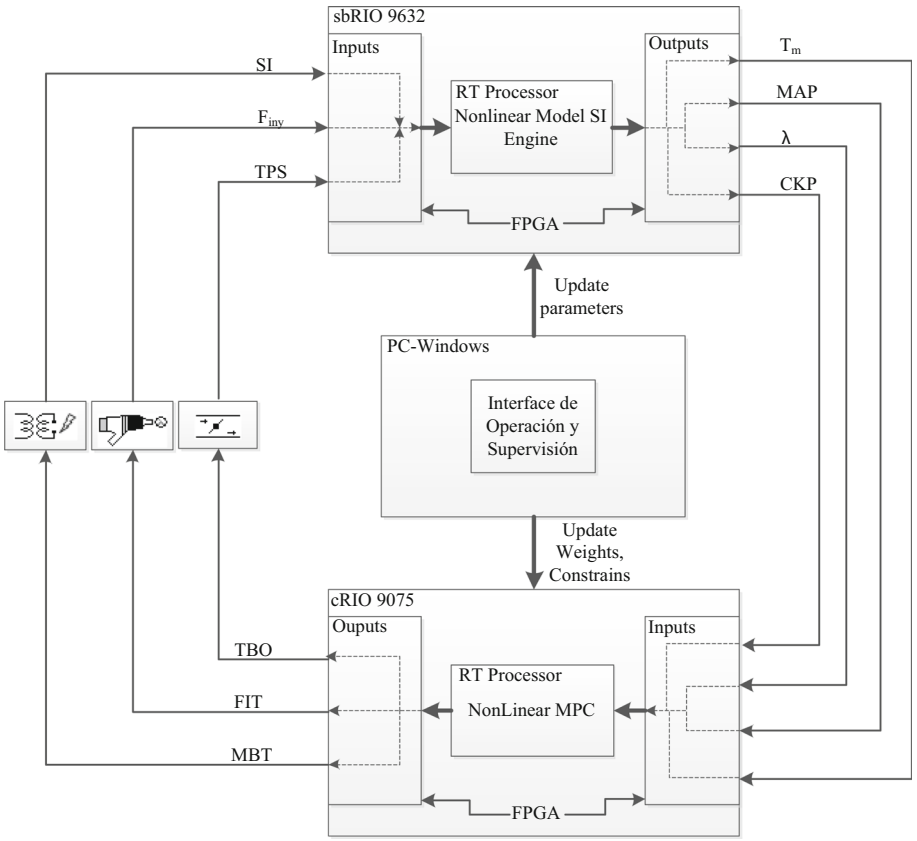


Fig. 1 System architecture

real machine, given the virtual prototype built on embedded real-time systems. The HIL tools, the FPGA, the real-time processors have allowed to create systems of experimentation and interactive design, within the area of mechatronics such that the design stage incubates the implementation itself. Products designed, ECUs for the case, can be achieved with ease of installation of a device "plug and play", connected to the engine and start it.

2 Materials and methods

2.1 System architecture

Figure 1 depicts the scheme implemented. The block named sbRIO 9632, is a real-time processor 400 Mhz with VxWorks operating system, 2 Mb FPGA logic gates and 128 Mb of direct access (DMA). A nonlinear model of the combustion engine involving the pressure and temperature in the intake manifold, the revolutions in the crankshaft and the normalized stoichiometric combustion ratio, are components implemented in this block. The solver state equations of the model are implemented in the processor, whereas others elements such as sensor models, scaling and updating analog outputs are designed with gates FPGA. The model inputs are signals from the actuators involved in the hardware loop: throttle position opening measurement or throttle position sensor (TPS), measured flow of fuel injected through the injectors F_{iny}, and the instant of ignition spark from the coil-spark plug assembly.

The cRIO 9075 block, with slightly higher performance that block described above, is used to implement the nonlinear MPC controller. The data lines from the block sbRIO to block cRIO, are physical connections of voltage signals, that cRIO expects in the usual ranges of the sensors used in automotive technology. So that the block sbRIO taking into account these requirements to normalize properly model signals, and so maintain compatibility with the connectivity of the real plant.

The MPC generates optimal control actions solving an optimization problem for the current operating point, based on a model replica also programmed the RT processor and information available of the current state obtained by the analog inputs through the correct scaling of the FPGA. The FPGA of this block also implements event handling discrete motor, i.e. synchronization signal from crankshaft position (CKP) the fuel injection time and ignition of the mixture. Same time implements a control loop the throttle opening, according to the reference commanded by the optimal controller. The power drivers are embedded in cRIO outputs for actuation of the injectors, ignition coil and motor butterfly valve. A flow sensor disposed in the injector rail reports the injected flow to model inputs, proper conditioning installed

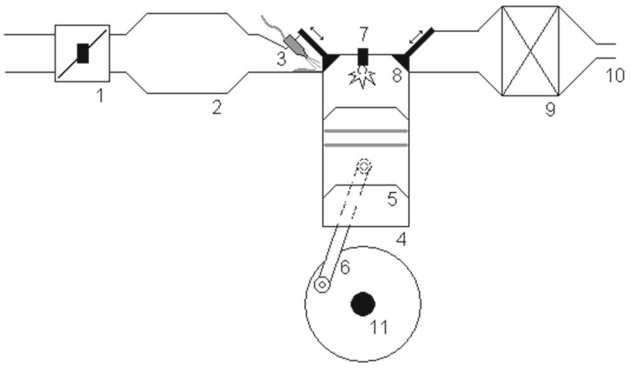


Fig. 2 Simplified schematic of the machine

in the spark plug electrode reports the ignition timing and with the TPS the opening angle.

There is provided an interface operation and supervision on a Windows based computer, to provide a means of modifying model parameters and weight matrices control system, and visualization of the variables of interest.

2.2 Model implemented

The model used in this report is coined as mean value engine model (MVEM) by Hendricks and Sorenson [5]. The model represents the average value of the variable rather than its instantaneous value at each machine cycle. We distinguish three main dynamics in MVEM (see Fig. 2 for clarity): filling dynamics air intake manifold, fuel dynamics on the walls of the valve intake and crankshaft speed dynamic.

In the sketch of a machine distinguished combustion internal: (1) butterfly valve, (2) intake manifold, (3) injector, (4) combustion chamber, (5) cylinder, (6) rod-crank, (7) spark plug, (8) valves intake and exhaust valves, (9) the catalytic converter, (10) exhaust and (11) output shaft. The first-mentioned dynamic occurs in areas 1 and 2, fuel dynamics involves the elements 3 and 8, and crankshaft dynamics including combustion itself with the elements 7, 4, 5, 6, and 11.

It is considered that a major dynamics involved in MVEM [5] is which occurred at the intake manifold, which is described by expression 1.

$$d\frac{P_m(t)}{dt}\left(\frac{V_m}{RT_m}\right) = \dot{m}_\alpha(t) - \dot{m}_\beta(t) \tag{1}$$

The intake manifold dynamics is described from the application of the isentropic equation of air and the ideal gas law in the throat of the butterfly. A development of this can be extended in [6]. P_m, refers to the instantaneous pressure in the manifold and forms the first reservoir or state variable included in the model. V_m and T_m, refer respectively to the volume and temperature in the manifold. The volume is a constructive feature dependent on engine capacity, and although the temperature is considered constant also, has used an approximate model for this report. R, refers to the gas constant. The interaction between the mass inflow $\dot{m}_\alpha(t)$ by the butterfly valve and flow $\dot{m}_\beta(t)$ entering the cylinder (zones 1 and 3 of Fig. 2), generates a reservoir for P_m (zone 2).

The opening angle of the butterfly is an input to the system and becomes important in its effect within the model. Such angle, together with atmospheric and manifold pressures then determine the flow $\dot{m}_\alpha(t)$ through the throat of the throttle, here comes the equation of isentropic expansion of air, given in expression 2.

$$\dot{m}_\alpha(t) = \begin{cases} A(\alpha)\frac{P_{atm}}{\sqrt{RT_{amb}}}\frac{1}{\sqrt{2}} & si \quad \frac{P_m(t)}{P_{atm}} < 0.5 \\ A(\alpha)\frac{P_{atm}}{\sqrt{RT_{amb}}}\sqrt{2\frac{P_m(t)}{P_{atm}}\left[1-\frac{P_m(t)}{P_{atm}}\right]} & \end{cases} \tag{2}$$

$A(\alpha)$, represents the effective throat area of the butterfly, which is a function of the opening angle α, determined from the body constructively butterfly, although experimentally extraction of a polynomial relationship is often easier and more accurate. In the model implemented, development found in [5] was used in the form of 3.

$$A(\alpha) = 1 - \cos(\alpha) - \frac{\alpha_o^2}{2!} \tag{3}$$

α_o, is the minimum angle constant used for idling and involves an aggregate by summing $\dot{m}_{\alpha o}(t)$ in 2. P_{atm} and T_{amb}, correspond to the atmospheric pressure and the ambient air temperature.

Air dynamics that enters the cylinder $\dot{m}_\beta(t)$, can be modeled as a positive displacement pump, where the intake air is dependent on the angular speed and manifold pressure (see Eq. 4).

$$\dot{m}_\beta(t) = \frac{V_d}{4\pi RT_m}\eta_v\left(\omega(t), P_m(t)\right)\omega(t) \tag{4}$$

η_v, is the volumetric efficiency of the engine and is found experimentally. The expressions used to model it, usually are polynomial forms, functions of speed and manifold pressure. For the implemented model, has been used the expression 5 taken from model Hendricks:

$$\eta_v = s_i(\omega(t))P_m + y_i(\omega(t)) \tag{5}$$

s_i and y_i, are explicitly experimental features depending on the angular velocity whose preparation is in [5].

To search for a better model accuracy, expression 1 is modified modeling the dynamics of the manifold temperature using the law of conservation of energy, Eq. 6.

$$\frac{dT_m}{dt} = \frac{RT_m}{P_m V_m}\left[m_\alpha(\kappa T_{amb} - T_m) - m_\beta(\kappa - 1)T_m\right] \quad (6)$$

Thus the expression 1 using the adiabatic model changes to Eq. 7. With κ, identifying the specific heat of air and using the value of 1.4 in the model. T_m, becomes another state variable model.

$$d\frac{P_m(t)}{dt}\left(\frac{V_m}{R\kappa}\right) = \dot{m}_\alpha\,(t)\,T_{amb} - \dot{m}_\beta(t)T_m \quad (7)$$

When the injector delivers fuel into the intake port (item 3 in Fig. 2), not all the fuel is vaporized directly within the cylinder, but rather part of it remains on the walls of the intake valve and in areas close to the intake port. Also depending on the current time of the machine cycle, the intake valve for a given cylinder is closed at the moment of injection, and will not enter the fuel injected into the cylinder until the moment of admission. The above considerations determine a dynamic delay between the injected fuel and who actually enters the cylinder, which is defining the future air/fuel ratio AFR. This dynamic model described in the set of Eq. 8.

$$\begin{aligned}\dot{m}_\varphi &= \dot{m}_{fv} + \dot{m}_{fl}\\ \dot{m}_{fv} &= (1 - X)\dot{m}_\psi\\ \frac{d}{dt}\dot{m}_{fl} &= (1/\tau_{fl})(-\dot{m}_{fl} + X\dot{m}_\psi)\end{aligned} \quad (8)$$

Than 8 is distinguished $\dot{m}_\varphi$ as the mass of fuel which reacts in the cylinder and produces work, $\dot{m}_{fv}$ is the amount of fuel vaporized, $\dot{m}_{fl}$ is the amount that permeates the surrounding walls, which gradually evaporates with time constant t_{fl}, $\dot{m}_\psi$ corresponds to the injection quantity and X is the proportion of the impregnated part. X and t_{fl}, variable elements are dependent mainly on the pressure in the manifold and the current regime. Expressions used for both polynomials are documented in [5].

Movement dynamics in the crankshaft is expressed in 9, indicating the relation between the produced torque and the load torque.

$$\frac{dn}{dt} = \frac{1}{I}H_u\eta_t(P_m \cdot n, \lambda)m_\varphi(t - \Delta t_d) - \frac{1}{I}\tau_l \quad (9)$$

The torque produced is mainly dependent on the calorific value of fuel H_u, the amount of fuel that reacts m_φ, and an experimental feature known as thermal efficiency η_t [5,6]. The $(t-\Delta t)$ factor indicates the dead time or delay in the production of torque with respect to the fuel injection moment and is a function of speed and engine design conditions as expressed by 10.

$$\Delta t_d = \frac{60}{n}\left(1 + \frac{1}{\#_{cyl}}\right) \quad (10)$$

Ignition timing of the mixture is introduced by Hendricks as an exponential relationship of the form 11. This expression influences the torque produced and introduced into the function of the thermal efficiency.

$$\eta_{t_s} = e^{-\frac{\theta^2}{\theta_s^2}} \quad (11)$$

η_{t_s}, is named as a factor in thermal efficiency dependent spark t_s retard, θ y θ_s, the angle being measured before TDC [7].

In addition to the state equations that model these dynamics, in the model used, is added the expressions to calculate the stoichiometric ratio, which is a measurement that is taken in the catalytic converter and exhaust, elements 9 y 10 in the Fig. 2.

Figure 3 shows a block diagram that outlines the component implementation in block sbRIO. The AFR and fuel dynamics are programmed as sub models of combustion model. The variables are computed in engineering units, and a component in the FPGA is responsible for translating the scaled variables in sensors ranges and generating the typical periodic sensor signal CKP.

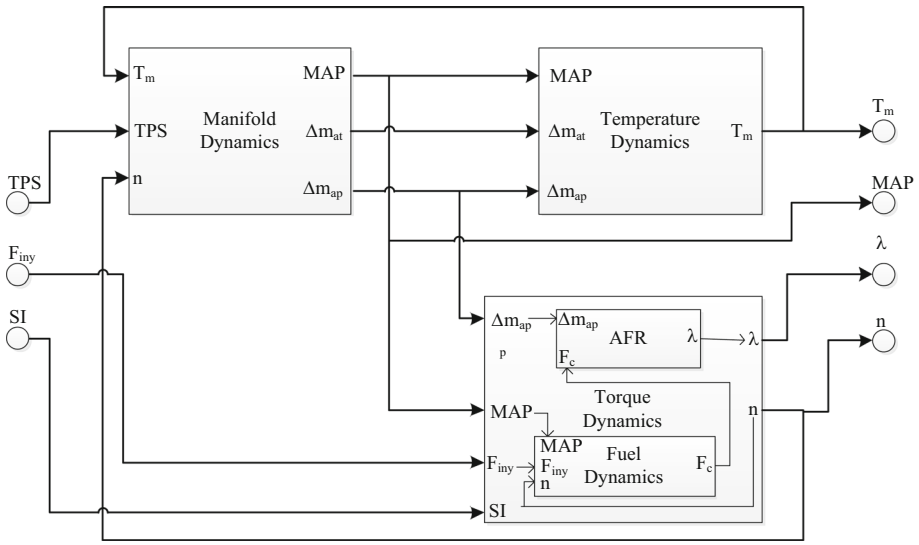
National Instruments Corp technology was used for the implementation of this HIL system. The platform engineering and programming *LabVIEW*, has a module control system design and simulation of dynamic systems, the model of Fig. 3 is simulated by the solver ordinary differential equations (ODE) of this module in the processor within the platform sbRIO on real-time operating system VxWorks [8]. As mentioned later the same model is implemented in the cRIO platform for relevant tasks with the control system. Additional aspects of implementation and execution of the model is reported in the results section.

The physical outputs of sbRIO platform marked in Fig. 1 as T_m, MAP, λ and CKP, correspond to voltage signals that simulate the corresponding temperature sensors, manifold pressure, oxygen in the catalytic converter and Hall Effect sensor for the CKP. Numerical sources for these signals in the model corresponds to the variables T_m, P_m, and n, for T_m, MAP y CKP, and for λ the ratio $\frac{1}{14.7}\frac{\dot{m}_\beta}{\dot{m}_\psi}$. At the FPGA device in the embedded platform is programmed physical relationship between the measured variable and the usual output voltage corresponding sensor, numerical values generated by the model are used as the measured value.

The temperature in the test bench is measured using a glass encapsulated thermistor, with measuring range from −60 to 300 °C [9], accuracy 0.5 °C and response times of 8 s. The relationship between temperature and the sensor resistance is given by the expression 12:

$$\frac{R_T}{R_{T_0}} = e^{\beta\left(\frac{1}{T} - \frac{1}{T_0}\right)} \quad (12)$$

Fig. 3 Model implemented



R_T, is the resistance at temperature T, and R_{T_0} resistance at reference temperature 25 °C. β, is the temperature coefficient of the material. A simple voltage divider allows a reading of the change in resistance as a function of temperature, such that 13:

$$T_m(v) = \frac{R_{T_0} e^{\beta\left(\frac{1}{T_m} - \frac{1}{T_0}\right)} \times V_{DC}}{R_C + R_{T_0} e^{\beta\left(\frac{1}{T_m} - \frac{1}{T_0}\right)}}. \tag{13}$$

To simulate the behavior of the thermistor, Eq. 13 is implemented in FPGA section delaying the output response to emulate the dynamics of the thermistor. As parameters has been used $\beta = 3528$, $R_{T_0} = 200\,\text{k}\Omega$, the resistor of calibration R_C adjusted at 100 Ω and 5v to V_{DC}.

The common sensor manifold absolute pressure (MAP) has been modeled to provide a simulation of voltage output versus pressure calculated in the motor model. The sensor mounted on the test bench corresponds to the MAP sensor described in the technical reference [10]. The characteristic response of this device is shown in Fig. 4, taken from this reference. The linear function 14 defines its behavior.

$$P_m\,(v) = V_{DC}\,[0.0001\,P_m - 0.12] \tag{14}$$

Although the figure presents units in kPa, expression 14 provides unit conversion from bars, to engage the model output units.

The other output of the model is necessary to represent physical voltage for proper interpretation by an ECU or control system is implemented in this work, is the signal corresponding to the lambda probe. Emulated operation corresponds to a lambda probe type oxygen sensor Universal Exhaust Gas Oxygen also called oxygen sensor wideband. Specifically the emulated sensor is NTK L1H1, technical

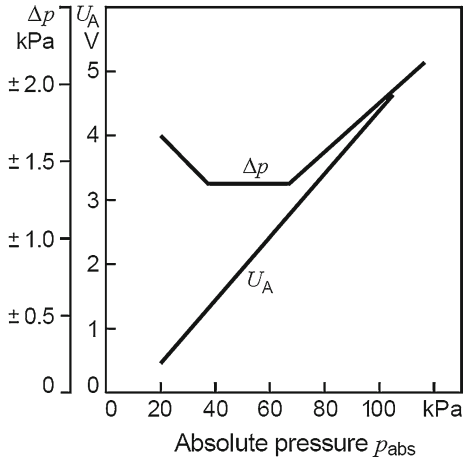


Fig. 4 Transfer function MAP sensor (taken from [10])

information can be found at [11], from which we extract Fig. 5, showing the transfer function between the voltage information and lambda sensor output.

To generate the output voltage according to information AFR which produces the model, a lookup table was implemented with the marked points on the graph and with linear interpolation the intermediate points are obtained.

The last emulated sensor corresponds to the speed information n. Speed in internal combustion engines is determined by the ECU from the information generated by the sensor CKP and a toothed steering wheel solidly joined to crankshaft. Figure 6 schematically this operation.

The toothed wheel has 60 points for the location of the teeth uniformly distributed, but a couple of them are not machined. This feature is used to synchronize the moment of injection and ignition of the mixture. A Hall effect sensor is positioned to detect the presence of teeth when the motor rotates. The voltage signal ranging from −4 to 4 V whenever

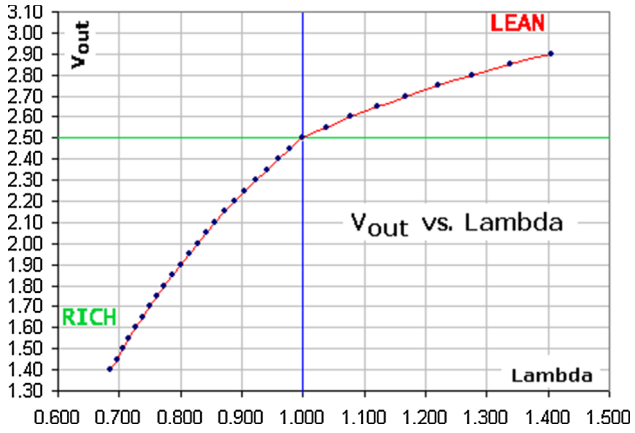


Fig. 5 Transfer function λ sensor (taken from [11])

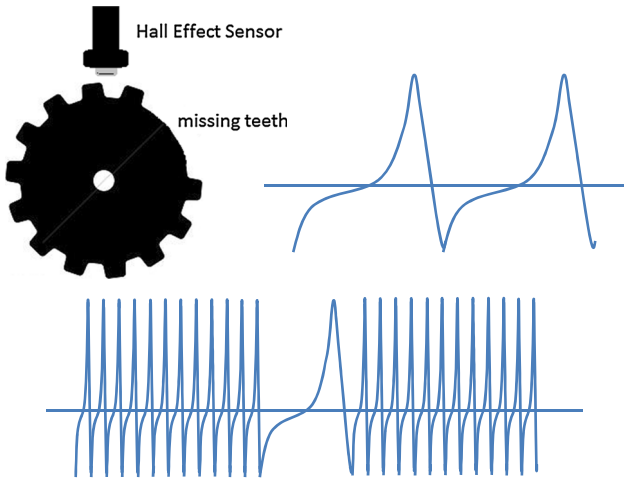


Fig. 6 CKP sensor operation

it detects the passage of a tooth shown in Fig. 6 (top, right side, showing the passage of two teeth). The signal from the bottom shows an oscillogram with the detection of somewhat more than one revolution by going through missing teeth.

The detection pattern of a tooth is stored in memory and a timing mechanism transfers each point of the pattern to an analog output of the FPGA. The update period is modified by the value of ***n*** reported by the model, also a counter monitors the number of teeth emulated to generate an additional update delay emulate missing teeth.

The model input signals are processed according to the real signals of the actuators interposed between the platforms sbRIO and cRIO. In Fig. 1, blocks that connects to the physical inputs of the FPGA device in the sbRIO, which are SI, Finy and TPS. These blocks are test bench real actuators, such that the connected block to spark ignition (SI) constitutes the ignition coil which is connected to the spark plug; in the secondary circuit, together with the spark plug, has been added a spark detector circuit that allows leading a logic signal to a digital input of the FPGA and determining the moment when the spark occurred. By the side of the block connected to Finy, the block symbolizes a flow meter installed between the fuel pump and injector rail delivering the actual measurement of the instantaneous amount of fuel injected. The relationship between the measured flow and the voltage output is linear and conversion is programmed FPGA level, such that the information provided to the model is given in units of flow (kg/s). Finally the position sensor TPS of the motorized throttle is used to provide the real information on the opening angle α. Angle information is linear with respect to the voltage signal delivered by the sensor; this signal is connected to an analog channel of the FPGA and converted by a linear relationship to geometric angle information to be used by the model.

2.3 Optimization problem

In engineering it has become inseparable optimization requirement in almost all aspects of this: by lack of resources, by quality, by utility, etc. In the operational phase, optimization requirements of internal combustion engines can distinguish three aspects or variables: obtaining the maximum amount of energy, resulting in maximum torque available at the crankshaft; minimization of fuel used to operate the machine, resulting in an economic question; and environmental awareness and government regulations, reducing emissions of noxious gases from the combustion [12].

Energy optimization is a desired benefit of the power delivered by the machine aspect, implying capitalizes on work capacity. When used for example in a power plant, such a need may arise from an increase in demand for amperage, or if the engine is coupled to a mechanical load, usually you want the motor to move the heaviest load. In transport vehicles, driving feeling, it is responsible for this requirement optimization; usually the driver expects the engine to bring the rest of the vehicle to the desired speed quickly, what is usually done with a strong increase in torque. As determined by the expression 15, the torque produced by the machine, is dependent besides the constructive aspects of the machine, of the amount of fuel $\dot{m}_\varphi$ reacting:

$$\tau_c(t) = \frac{H_\mu \cdot \eta_t\,(t - \tau_d) \cdot \dot{m}_\varphi}{I_e}. \tag{15}$$

So you might think that to maximize the torque reaction more fuel is added. Well, then there are restrictions to that amount of fuel, determined by building capacities and limits of amount of mass to be given proper combustion [7]. On the other hand, note that this action is punishing economic optimization, and to some extent environmental optimization.

The efficiency is related to the operation of the machine using the minimum resources. In the internal combustion engine consumable resources are the air and fuel (energy

expenditure by the pulse of electrical current to produce the ignition spark is not considered in this report). Air at atmospheric pressure is not a resource that requires a business transaction to be obtained, so talking about consumable resources, the opportunity to be limited to the efficient use of fuel. It is possible to consider the minimization of fuel consumption using cost functions, optimization algorithms within, that contemplate the accumulation of spent fuel through a driving cycle [7], however for optimum control algorithms minimize usual increase of fuel injected in each control period.

Independent minimization of fuel consumption, the injected fuel must ensure stable operation of the machine, since there is a lower limit of speed for which the machine is maintained in operation; therefore the injected fuel cannot be zero [13]. However, the operation of the machine usually requires additional operating regimes at minimum speed of stable operation (usually 750 rpm), then the minimum values of fuel injection will vary depending on the operating point. In Fig. 7, speed shown in steady state and vacuum, a function of the throttle opening α and the injected flow $\dot{m}_\psi$, built with the engine model (Fig. 3) using an ignition advance 27.5°, constant atmospheric pressure and ambient temperature. The limits of the surface on the graph are restrictions on the variability of function arguments speed. The minimum value of TPS to keep the motor is 16.5° and an injection of fuel flow 0.00015 kg/s. The upper limits are according to the above regimen used in modeling experiments. The plane parallel to the base shown in its cutting surface corresponding to a speed isoline 2000 rpm with three points on it demarcating different values α and $\dot{m}_\psi$. The point at the left end shows that a higher fuel consumption (0.0006 kg/s) it is necessary with the minimum opening, however with a slight additional opening 10° it reaches the minimum fuel consumption, which remains relatively unchanged with larger apertures, as shown by the two other points to the right (0.00051 kg/s).

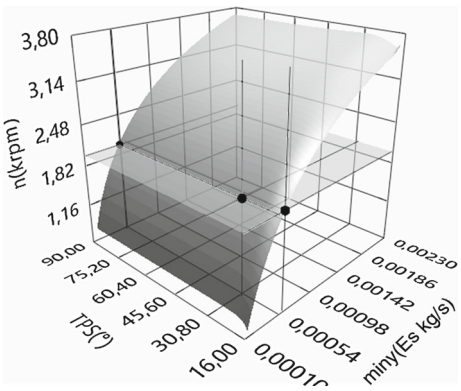


Fig. 7 Map of free response of the model compared to the range of possible inputs

The environmental aspect is addressed with additional constituents that have been added to the combustion engine as solutions to environmental requirements. This element is the catalytic converter in the exhaust path of the flue gases, together with the element oxygen sensor provides information about the reactions in this (refer to Fig. 2).

In the three reactions occur catalyst that reduce nitrogen oxides to nitrogen and oxygen, oxidizes carbon monoxide to carbon dioxide and oxidizing unburnt hydrocarbons to carbon dioxide and water [7], so that the harmful combustion product gases are converted into harmless to the environment. Two of these reactions require an oxidizing atmosphere is obtained with gasoline lean mixtures, while the remainder requires little oxygen, achievable with a rich fuel mixture. For the three can find a balance simultaneously is necessary to control the amount of oxygen within a very narrow range. The amount of oxygen present in the exhaust gases depends on the ratio between the quantities of air mass and fuel mass deposited in the combustion chamber, what is commonly called air/fuel ratio. For gasoline, combustion is theoretically complete when the AFR is 14.7, and this data is called the stoichiometric ratio. Is customary to normalize the measured AFR as set forth in 16:

$$AFR = \frac{m_\propto}{m_\varphi} \rightarrow \lambda = \frac{AFR}{AFR_e}. \tag{16}$$

The mixture is rich if $\lambda < 1$, and is poor if $\lambda > 1$, with rich mixtures $\lambda \approx 0.9$ high power can be obtained operating and lean mixtures $\lambda \approx 1.5$ fuel consumption is decreased [14]. But if the mixture is rich in fuel unburned hydrocarbons are produced, if it is poor, then nitrogen oxides are produced.

Figure 8 represents the conversion efficiency of pollutants according to λ. The narrow window displays where the three reactions are simultaneously and therefore the operation of the catalytic converter is more efficient. Environmental optimization is located in this condition, in getting a mixture as close to the stoichiometric point. The lambda sensor or oxygen sensor arranged in the exhaust pipe feedback allows the richness of combustion.

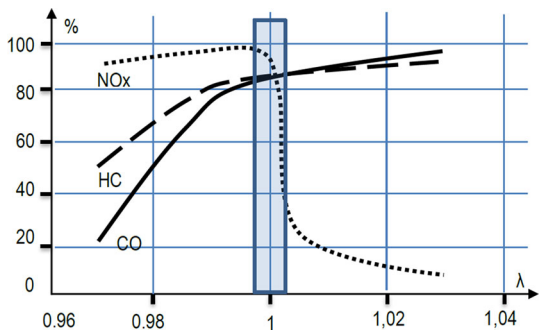


Fig. 8 Working window of the catalytic converter

As mentioned, goals can be incompatible watching all three fronts, although it is possible to find a compromise and watch multiobjective systems [15]. Another way may be to switch targets according to the dynamic changes of machine operation, for example, in transient instants, the optimization seeks to maximize the power delivered will be activated temporarily sacrificing minimizing harmful gases produced and fuel consumption. Both schemes can be implemented with relative ease considering optimization schemes proposed by the optimal control theory, addressing the control of the machine as a dynamic optimization problem solved at each control period.

In the field of optimal control, the optimization problem is addressed considering an objective function that relates the trajectory of state and control effort over a time horizon:

$$\min J = S\left(x(t_f), tf\right) + \int_{t_0}^{t_f} V\left(x(t), u\left(t\right), t\right) dt. \tag{17}$$

The first term of this expression 17 is called the terminal or final cost, and involving the integral expression is a functional that can measure the optimal time, the minimum expenditure and the minimum energy required to bring the dynamic system from an initial state $x\left(t_0\right)$ to a final state $x\left(t_f\right)$.

The functions S and V, can be particularized in the matrix form of the expression 18.

$$\begin{aligned}\min \quad J &= x_d\left(t_f\right)^T \boldsymbol{F} x_d\left(t_f\right) \\ &+ \int_{t_0}^{t_f} \left[x_d(t)^T \boldsymbol{Q} x_d(t) + u^T\left(t\right) \boldsymbol{R} u(t)\right] dt \end{aligned} \tag{18}$$

With $x_d\left(t_f\right)$, defined as $x_d\left(t_f\right) = Ref_x - x(t_f)$ and $x_d\left(t\right) = Ref_x(t) - x(t)$. Ref_x, is the final position for the state variable, whilst $Ref_x(t)$ denotes the trajectory in the time horizon. The matrices are used to weight the deviations of state variables compared to the references given [16].

The space of the decision variables $x(t)$ and $u\left(t\right)$ is restricted by the system dynamics control, described by the model equations, and boundaries imposed by physical security issues and operating ranges of the actuators. The model imposes equality constraints and boundaries, inequality constraints, as described hereinafter.

In the experiment conducted in [6] to validate the model, the minimum speed that keeps the engine operation is 0.6 krpm, and maximum security obtained by operating the engine is 6 krpm. The maximum opening angle of set for the butterfly valve model used is 90° and the minimum to maintain operation has been defined by the possible minimum aperture control engine idle in 16°. Boundaries for injection flows are taken from the experimental data at idle speed and 6 krpm, then the resulting flow range is 0.00015–0.005 kg/s. Referencing literature AFR control [17], it has been estimated that trips the stoichiometric ratio of 10 % are permissible. The usual practice is a window of ±3 % [14], in these range alternative excursions between rich and lean not considered harmful. However permissiveness defined in this aspect allows extending freedom in the control action to track the speed reference and does not touch the limits of toxic emissions. Boundaries for the signal λ are given between 0.90 and 1.1. The remaining state variables, temperature T_m and manifold pressure P_m are naturally bounded engine operation, dependent as described by the model, of ambient temperature, TPS and atmospheric pressure, so no are added boundaries to the problem formulation.

The complete optimization problem the objective function of the form 18, the mathematical model introduced equality constraints and bounds for the decision variables are set at 19, 20 y 21. The equality constraints or set of differential equations are defined for all t in the time interval $[t_0\ t_f]$ with an initial value, which in practice is made by a set of measurements.

$$\begin{aligned}\min J = &\begin{bmatrix} Refn - n & 1-\lambda \end{bmatrix} \begin{bmatrix} F_{11} & 0 \\ 0 & F_{22} \end{bmatrix} \begin{bmatrix} Refn - n \\ 1-\lambda \end{bmatrix} \\ &+ \int_{t_0}^{t_f} \begin{bmatrix} Refn - n & 1-\lambda \end{bmatrix} \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{22} \end{bmatrix} \\ &\times \begin{bmatrix} Refn - n \\ 1-\lambda \end{bmatrix} \\ &+ \begin{bmatrix} \alpha & m_\psi \end{bmatrix} \begin{bmatrix} R_{11} & 0 \\ 0 & R_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ m_\psi \end{bmatrix} dt \end{aligned} \tag{19}$$

s.a.

$$\begin{aligned} & d\frac{P_m(t)}{dt}\left(\frac{V_m}{R\kappa}\right) - \dot{m}_\alpha\left(t\right) T_{amb} - \dot{m}_\beta(t) T_m = 0 \\ & \frac{dT_m}{dt} - \frac{RT_m}{P_m V_m}\left[m_\alpha(\kappa T_{amb} - T_m) - m_\beta(\kappa - 1) T_m\right] = 0 \\ & \frac{dn}{dt} - \frac{1}{I} H_u \eta_t(P_m \cdot n, \lambda) m_\varphi(t - \Delta t_d) - \frac{1}{I}\tau_l = 0 \\ & \dot{m}_\varphi = \dot{m}_{fv} + \dot{m}_{fl} \\ & \dot{m}_{fv} = (1 - X)\dot{m}_\psi \\ & \frac{d}{dt}\dot{m}_{fl} - (1/\tau_{fl})(-\dot{m}_{fl} + X\dot{m}_\psi) = 0 \\ & 0.00015 < m_\psi < 0.003 \\ & 16 < \propto < 90 \\ & 0.6 < n < 6 \\ & 0.9 < \lambda < 1.1 \end{aligned} \tag{20, 21}$$

The performance index used, then consider the error against an externally supplied reference constant speed, and against a reference unitary for the normalized stoichiometric ratio λ (have been intentionally suppressed time dependencies), coupled with minimizing the waste of resources in the actions control the throttle opening and the flow of

fuel injected. Scalar set F_{11}, F_{22}, Q_{11} and Q_{22}, let give relevance to the solution of the optimization problem permissiveness regarding the speed error and emissions, so that more weighting F_{11}, Q_{11} becomes more demanding to achieve the speed reference, perhaps with some issues, and otherwise, emphasizes a clear environmental goal, a little neglected the power output of the machine. Similarly, R_{11} gives weight to the opening of the air inlet valve, is to say the mass flow of air into the manifold, and considering a usual commercial situation, this flow does not involve a direct cost, so this element can be weighted sufficiently low relative to R_{22} to tilt the outcome toward the minimum possible fuel injected.

The optimization problem described by 19–21, is the problem that the cRIO platform must solve at each sampling period to grab control of the drive machine and its operation as required. Within the optimization theory [18], is a nonlinear programming problem NLP, since the equality constraints, that is the model are nonlinear differential equations. There are several methods proposed to address the problem: stochastic methods, as adaptive random search, conjugate direction, genetic algorithms [19,20]; and deterministic methods, where the convergence to the optimum can be estimated. The latter methods are suitable for their applicability in control systems, in a way that allows the solution of the optimization problem in a time interval determined. Deterministic reclassified into direct and indirect methods. Direct methods such as simplex and complex to use in general, a search pattern [21,22], and indirect methods which are supported in the calculation of derivatives, is required so that the objective function is smooth with at least three continuous derivatives. For reasons including the shortest time calculation of the objective function and that converge faster [23], besides allowing applicability to the problem given, the latter type of methods developed for the optimization problem of engine operation.

An appropriate scheme to solve the optimization problem defined by 19–21 is the possibility of a reformulation as a problem of quadratic programming QP. For this the form of the objective function 19, is suitable because it is the quadratic form of the generalized form 17. However, requires that the restrictions are linear. The alternative is to linearize the model, such that the constraints are transformed to a set of linear constraints. To do this is then required to run a model linearization procedure for each operating point required and solve the problem at that point.

The solution of the problem QP can be addressed, by two different algorithms Newtonian type: active set methods and interior point methods. The last few have been developed, large problems focusing, while the former are widely used and verified effectiveness in problems of medium and small scale [18], then suitable for the optimization problem of combustion engine.

Active set method under consideration is defined only for the convex case. In the case of the objective function of the quadratic programming problem, this is established by the definiteness of the weighting matrices, they must be positive definite or positive semidefinite. In the Newtonian methods, in an iterative process, is sought a vector of decision variables x^*, such that the objective function evaluated at this vector is less than any other neighbor vector x of decision variables $f(x^*) < f(x)$. Starting from an initial value of this vector x_0, each iteration of the algorithm updates the new vector in the generalized form $x_{k+1} = x_k + d_k$. The term d_k, corresponds to a function of search direction. In indirect methods calculate the search direction is done with the use of information of the gradient and the Hessian of the objective function, looking convergence within a given accuracy level. The active set algorithm, also divides the restrictions in a set of active constraints and a set of constraints inactive for a current point x_k. A restriction, is activated when evaluating x_k in restricting equality holds, and is inactive when the inequality holds. The set of active constraints is called the working set $\mathcal{W}_k$. Starting from an initial set $\mathcal{W}_0$ and an initial value feasible x_0, an iterative process continuously solves a sub-problem QP whose decision variable is a vector step p_k towards optimum, with a number of equality constraints given by the active constraints in x_k. This sub-problem can be solved directly or by some method of factorization [23]. In the solution of p_k if a new search direction is not found, it is possible that the current point is optimal, then restricted optimality conditions are checked, associated Lagrange multipliers are obtained in active constraints and if all are positive this indicates that x_k is the optimum. If it is determined that the current point is not optimal, then the working set $\mathcal{W}_k$ is modified by removing the constraint with the most negative Lagrangian and forming a new sub-problem QP. If the result is a downward direction search, then be calculated for that length vector step and execute step by updating the current position x_k, if another restriction is activated then it is added to the active working set $\mathcal{W}_k$.

Deal linearization for each operating point and solve the optimal control problem it is a possibility if you have sufficient computational resources, since the two processes are demanding processing, and if the operating point continuously changes the call to the linearization process is continuous. Another scheme to solve the optimization problem given by 19–21, is dealing directly with the nonlinearity of the engine model. I.e. solve the NLP problem, and not reformulated as a QP problem. One option explored for system 2-HIL, is derived from the same problem QP, called sequential quadratic programming (SQP).

The SQP method resembles the foregoing scheme, in which an iterative method is that, at each iteration solves a sub-problem QP defined by a linearization of the restrictions. The minimum of the problem is used to construct the QP of

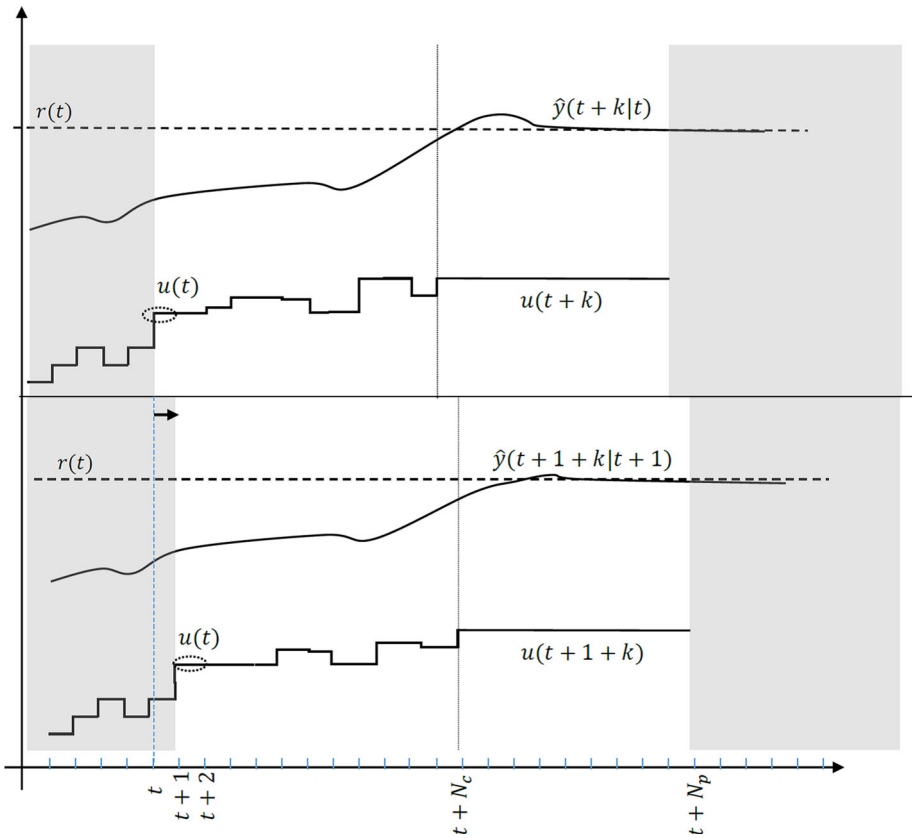


Fig. 9 Operation MPC

the next iteration whose solution is a better approximation to the optimum, such that the sequence $x_k, k = 0, 1, 2, \ldots$ converges to the optimum. Each sub-problem QP is constructed using a quadratic approximation of the objective function and restrictions on local affine x_k approximations. Some schemes SQP, are supported on the Newton method, and apply the terms and theorems developed for this, such that the objective function and constraints must be smooth (three times differentiable), addition techniques can be considered as the active set solution which was used in the above scheme. The development of the algorithm can be found in [18,23].

The active set and SQP methods are evaluated in this report to determine the ability of the cRIO platform optimally controlling the operation of the internal combustion engine. Results of preliminary tests on the solution of an instance of the optimization problem are discussed in the results section.

The active set and SQP methods are used to implement control strategies nonlineal model predictive control (NMPC), to be checked on the cRIO platform. The NMPC derives its name from the MPC control strategy using a linear model, while the NMPC using a nonlinear model. However, was coined in this report to the abbreviation NPMC a pair of nonlinear control schemes: one based on MPC controllers, updated by a linearized model at the current operating point and another, using the internal approximation of the objective function and the constraints, of the solution method SQP.

2.4 NMPC controllers

With regard to industrial control systems which must be solved in real time, mechanisms and solution algorithms running on computing systems discreet and online, such that the information of the plant and optimization are treated in discrete form. MPC scheme is then formulated based on discrete or discretized model of the plant or process being controlled. Considering the problem of optimizing the performance index or objective function 18, and $t_f = \infty$, consider implementing your solution as an infinite horizon controller. The design of this controller can be made by repeatedly solving optimal control problems of finite time so that the horizon is moving to meet the infinite horizon. At each sampling time a problem of optimal open-loop control with a finite horizon is resolved, using the measurements to define the present state. In Fig. 9, an outline of the process is shown, unshaded area shows a time window provided by the MPC controller. The upper part shows the process given in the current sampling period; the lower part illustrates a subsequent sampling period.

In the current time t, the controller conforms, based on the model and measurements of the state, problem of optimal open-loop control with a final time for states equal to a prediction horizon N_p, and for the input variables equal to a control period N_c. N_p and N_c they are given in terms of

sampling instants k and are tuning parameters of the controller. Formed optimization problem is solved by a method and solution, which has the form of a piecewise constant trajectory, only the first piece is applied to the actuators of the process, as highlighted in the graph with $u(t)$, that is in the range $[t, t+1]$. A sampling period after, with current status information, a new problem is formulated with the same size in the horizons but taken from $t+1$, that is with a shift of a sampling period. Note of the graph that the trajectories of the output variable differ from one sampling instant to the next in the control window, this shows the controller operation, where for a sampling time of the plant behavior is predicted for a prediction horizon, taking into account current measurements, using the process model and the computed path control, in the next instant, as the prediction of the real behavior differs, prediction will be adjusted again. So $\hat{y}(t+k|t)$, in figure, does not refer to the process measures, as it is either $u(t+k)$ regarding the process inputs. The notation $(t+k|t)$ denotes the value of the variable that accompanies, in the future instant $t+k$ predicted or calculated in time t. It is valid to consider that the time t, always used as a reference point, such that as the horizons are fixed for a given parameter, we can do without loss of generality $t=0$, and with this in mind we establish a widespread practice MPC controller formulation [24,25]. MPC discrete formulation is presented in Eqs. 22, 23 y 24:

$$\min_u J = x_N^T F x_N + \sum_{k=0}^{N_p} \left[\hat{y}(k) - r(k)\right]^T Q \left[\hat{y}(k) - r(k)\right] + \sum_{k=0}^{N_c} u(k)^T R u(k) \tag{22}$$

s.a.

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) \quad k = 0, 1, 2 \ldots N_p \tag{23}$$

$$y(k) = C\hat{x}(k)$$

$$x(0) = x(k)$$

$$\left.\begin{array}{l} X_{min} \le \hat{x}(k) \le X_{max} \\ U_{min} \le u(k) \le U_{max} \\ Y_{min} \le \hat{y}(k) \le Y_{max} \end{array}\right] \forall k$$

$$x_N \in X_F. \tag{24}$$

The objective function is rewritten 19 directly in the form of 22 and MPC scheme solves for each interval one LQ problem described by a linear quadratic objective function [26] (note that LQ is a generalization of the QP), including the error in the control outputs, crankshaft speed and AFR, a weigh for inputs, position of the throttle and fuel injection, including the restriction imposed by the motor model and operating restrictions, for both the control variables and the manipulated, at one operating point, where the plant dynamics is defined by the model 23.

The MPC controller is based on a linear representation of the plant model to predict their behavior, so this represents a drawback for applications in nonlinear plants. The literature offers a vast number of possibilities already developed [27], of which are specifically experienced in this report two: linearization technique on line, and another with the use of an explicit algorithm for nonlinear programming solution called SQP, well face in predictive control problem based on the nonlinear model NMPC.

2.4.1 NMPC with linearization

The formulation of this MPC corresponds to the objective function 22 with $\hat{y} = [n\ \lambda]$, $r = [Refn\ 1]$ y $u = [\alpha\ m_\psi]$, with the restrictions given in 21 for all sampling instant k, but with 25, as prediction model:

$$\begin{aligned} \frac{dx}{dt} &= f(x(t), u(t))\Big|_{\substack{x=x_{op} \\ u=u_{op}}} \\ \hat{y}(t) &= f(x) \\ x(0) &= x_{op}{}^{\wedge} u(0) = u_{op}. \end{aligned} \tag{25}$$

The subscript *op* means operating point and $f(x(t), u(t))$ refers to non-linear engine model. The 25 expression symbolizes the operation linearization equations of the model in the operating point *op* and measurements at time k define the operating point.

The dynamic model 23 is a usual way to describe a linear system in the state space, continuous representation of this model is given in expression 26 [16].

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{26}$$

The state matrices A, B the input matrix and output matrix C are constant matrices in general, but in the dynamics of the plant in question, these matrices are functions of time and other state variables [6]. To address the feasibility of implementation of the algorithm, several schemes are proposed in the literature, such as the incorporation of dynamics into linear parameters varying model and a policy of programmable gains [28] or parameterization of various MPC algorithms from a family of linear models along the path or region of operation and switching the MPC under the supervision of the operating point [29]. In take advantage of this last strategy, the non-linear system describing the combustion engine, which in general is given by 27, can be linearized, that is calculating the state matrices, input and output, from Taylor series expansion [30] in a neighborhood close to an operating point and achieve a nominal control u_n to generate a nominal state trajectory x_n.

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), t) \\ y(t) &= g(x(t), u(t), t) \end{aligned} \tag{27}$$

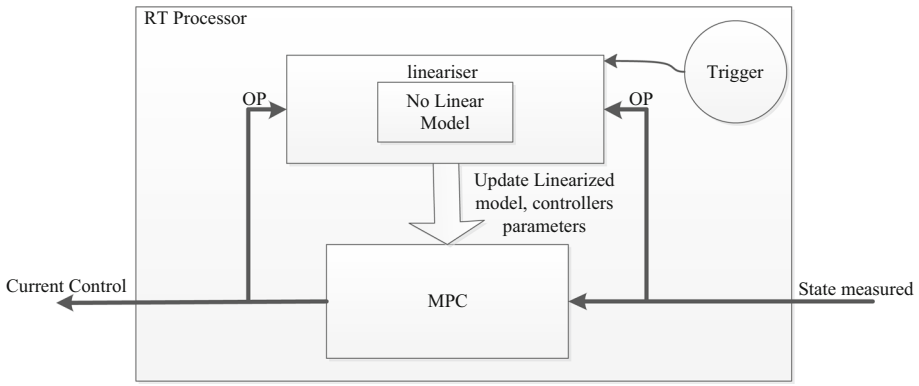


Fig. 10 NMPC with linearizer

Rewriting the variables according to the neighborhood of the operating point, are introduced into 27, the set of expressions 28.

$$\begin{aligned} \delta x(t) &= x(t) - x_n \\ \delta u(t) &= u(t) - u_n \\ \delta y(t) &= y(t) - y_n \end{aligned} \tag{28}$$

In terms of the new variables restricted to a neighborhood of the operating point, 27 become 29.

$$\begin{aligned} \delta \dot{x}(t) &= A\delta x(t) + B\delta u(t) \\ \delta y(t) &= C\delta x(t) \end{aligned} \tag{29}$$

The constant matrices A, B y C, are calculated according to 30, as Jacobians of system 27 [30].

$$A = \left.\frac{\partial f}{\partial x}\right|_{x_n,u_n} \quad B = \left.\frac{\partial f}{\partial u}\right|_{x_n,u_n} \quad A = \left.\frac{\partial g}{\partial x}\right|_{x_n,u_n} \tag{30}$$

The linearization function is introduced to run in parallel and asynchronously with respect to the MPC controller.

Figure 10 shows a diagram with the components and relationships between them to the MPC linearized as has been implemented in the cRIO. A linear MPC algorithm runs continuously using a linearized model of the engine for the current operating point together with the formulation of the problem for this model. In parallel a component called linearizer, which contains a nonlinear engine model, runs to provide an updated linearized model and reformulation of the problem.

Four trigger schemes to run the linearizer have been checked: at regular time intervals, based on the steady state error, linearity observing changes from operation point to another and by estimating a change in the operating point with observation of the trend of the reference. The circle in Fig. 9 denotes this element.

2.4.2 NMPC with SQP

Note that despite working in the area of nonlinear process control, control scheme described above, remains a linear MPC solution, then at a given instant, the controller uses the optimal resolving a linear model. The term is not widely NMPC coined such schemes; on the contrary, it is usual to use nonlinear programming algorithms. SQP algorithm is an adequate option within the NLP, that has adjusted well in solving optimal control problems [27,31]. Same as standard MPC, in that the linear model is used to predict the response of the system from an initial state to the prediction horizon, the nonlinear model is used for the same purpose for the standard NMPC, the difference is that the solution of the future states with the linear model is obtained with established methods for solving systems of linear algebraic equations, and for the nonlinear case using differential equations solvers with mature algorithms in the numerical solution of differential equations as Euler or Runge–Kutta. The joint operations of ODE and NLP solver may involve a higher processing load, even more so considering the infinite dimension of the design of optimal control problem, then for a tractable solution to this, algorithms for NLP problems, the optimal control problem is converted into a nonlinear programming problem finite dimensional [25], parameterizing the inputs and the states by a finite number of parameters and approximating differential equations during the optimization. In the literature three methods or strategies are reported for this, called direct methods [25,27]: single shooting, also called sequential approach or feasible route, multiple shooting y collocation, the latter also referred to as simultaneous methods.

In the sequential method only the control variables are discretized over the time horizon and the model equations are solved in each iteration of the NLP solver by solver ODE, such that only the control paths are considered as variables of optimization or degrees of freedom. For each evaluation, differential equations and the cost are numerically solved using the current estimate of the parameter vector of the inputs coming from the optimizer, this is the reason by which the sequential method named, since the optimization and simulation steps are executed one after the other leading to a feasible state trajectory.

Sequential schemes have been being efficiently used in chemical process applications [32], due to ease of implemen-

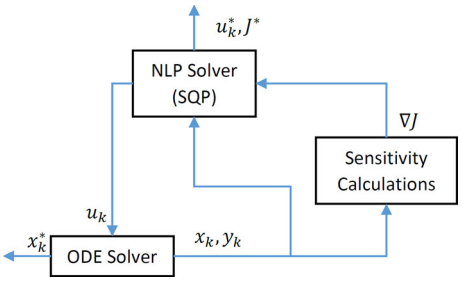


Fig. 11 Single shooting for NMPC

tation and reliable by linking and NLP efficient ODE solvers, with the obvious disadvantage of repeated numerical integration of the model and do not guarantee the handling of naturally unstable systems. The multiple shooting method, was developed to address precisely this last drawback, of unstable systems, performing a partition prediction horizon and formulating a problem for each subinterval, then the solutions of the models in each partition allow to introduce more information of the state variables in the formulation and capture unstable modes, as the mesh is finer. NLP problem formed usually increases its size, and also requires ODE solvers efficient enough to solve the multiplicity of models produced. With regard to scheme implemented for optimal combustion engine controller, it seems that a sequential scheme is sufficient, since most of the area of functional operational engine behavior has a naturally stable, boundary regions only have some degree of instability. Besides the possibility of using available ODE solvers and NLP for embedded systems, accelerate the design and implementation of real-time optimal controller.

The single shooting working scheme is shown in Fig. 11. In each iteration of the NLP one simulation is performed from the initial conditions to the prediction horizon in the ODE, state profile is analyzed for convergence and again the NLP block calculates a new vector control. After convergence the optimal control vector is obtained from the NLP block and the optimal state vector from the ODE block.

The accuracy of the solution depends mostly on the model solutions in block ODE, and the correct estimation of the initial values. The calculation of sensitivities block determines the assessment criteria of convergence of SQP algorithm, obtains the gradient of the objective function with respect to changes in the control vector, $\nabla_{u_k} J, k = 0, 1, \ldots, N-1$. For this, sensitivity functions are used state x_k and the sensitivity control vector as in [33].

The use of nonlinear optimization algorithm SQP in the NLP Solver, involves the explicit use of the nonlinear model of the plant in the formulation of the problem, thus constitutes a problem of nonlinear programming NLP. The generality of this method is similar to the previous with the operation of linearization embedded in the algorithm. In the SQP, a sequence of sub-problems is solved, where for every problem the model is linearized and a quadratic model of the Hessian of the Lagrangian is formed at the point of operation [26]. This requires discretize the optimization problem. In each iteration, the solution is a search direction, and SQP algorithm searches in this direction to a new operating point such decrement a merit function that determines the convergence of the algorithm.

2.5 Implementation concepts

An important added value, to be pointing in this work is the contribution of a quick and intuitive tool for designing optimal ECU algorithms for SI engines. Two technological trends support this: HIL platform and approach to interactive design. Briefly discuss these issues and their representation or application in the work of this report.

Hardware in the loop test platforms use a technique for testing embedded systems level, where regularly embedded controller system functions as a plant or physical process, but before the duo embedded-plant work together, embedded hardware is subjected to testing using a process description based software. The hardware is in a "Loop" with a simulator of the plant to be controlled. Embedded generated through interfaces I/O, the governing electrical signals for the actuators which should be interpreted by the simulation model; is usual also find schemes with embedded actuators in the loop to circumvent their modeling; at the same time the embedded controller, expects to receive the electrical signals from the sensors of the process, so the simulator should possibly perform a translation of the numerical values of the variables of the model towards scaled values of electrical signals.

The usefulness of HIL testing lies in that testing of control algorithms with complex plants can be costly and/or dangerous. A test session can result in unstable systems, overshoots, and inadmissible behavior; unproductive expenditure of energy and raw materials; or simply the plant is still nonexistent.

The reliability of the results obtained in the HIL tests depend on how faithful reproduction of the process is within the simulation system and the rigor of the tests carried out. It is obvious that a good model of the process is necessary; covering dynamics, the effects of uncertainty, noise measurement systems, discrete events in addition to continuous processes; with ODE solvers that solve the models in real time to ensure determinism and parallel processing for simultaneous dynamics that may occur in a process. The inclusion of the actuators within the loop, if possible, is a point that adds realism. Consider every aspect seriously contributes to the creation of a HIL platform with a highly realistic virtual model of the process.

With a realistic model and a human–machine interface (HMI) that allows a deep and extended interaction model and

Table 1 Parameters used for the model

Par	Meaning	Value
V_m	Volume in the intake manifold	0.0001723 m^3
V_d	Volume displacement of the machine	0.001149 m^3
R	Universal gas constant	286.9 J/kg °K
P_{atm}	Barometric pressure	85,113 Pa
T_{amb}	Ambient temperature	298.15 °K
α_0	Minimum aperture of the throttle	16°
κ	Isentropic coefficient	1.4
s_i	Volumetric efficiency coefficient	0.961
y_i	Volumetric efficiency coefficient	−0.07
H_u	Calorific value of fuel	43,000 kJ/kg
I	Inertia of the motor shaft	5.2638 kg m^2
$\#_{cyl}$	Number of cylinders	4

embedded, embedded system designer can trust the design results are directly applicable to the real plant, ie you can rely on immediate implementation, "ready to plug and play". Then the HMI or operator interface should be conceived for an outcome with interactive design principles: continuous visual information, full duplex interaction, prevent and recover erroneous actions, oriented fast and intuitive learning. This is achievable from the construction process itself, i.e. from interaction design and software architecture of the application. In the case, the recommended architectures used development tools (LabVIEW) contemplated and the following requirements associated with the operation of the system from the hardware platform PC-Windows depicted in Fig. 1 are:

- The general characteristics of each user interface are: clarity, ease of navigation, concreteness, focused on the user and functional.
- Changing optimization weighting matrices with online update and in real time.
- Ease of selection of targets optimization individually and simultaneously.
- Graphical presentation of trends in each state variable, each output and each input.
- Various forms of stimulus to the input signals.
- Availability of numerical tools, and 3D graphics for data analysis simulation.
- Creation of test scenarios with possibilities of operation in the limit, insertion of disturbances and noise models, and dynamic load changes.
- Update of model parameters online, and aggregation of uncertainty.
- Inserting errors NMPC controllers and control processes discrete events.
- Simulation of failures of sensors and actuators.

In the next section some views of software architecture used for meaningful interaction experience for designer ECU's with the system are reported.

3 Results and discussion

The aim of the research, encompassing this report, is to build HIL platform, to implement optimal control units for the internal combustion engine. In hardware unit is implemented model, and in another control unit is implemented. Once the control unit is implemented, tested and verified in performance in controlling the hardware simulated plant, should do with the real engine. The measure of success is the control unit that works with the model to work with the real engine in a plug and play manner.

The first results of this purpose are reported here: model implementation, model behavior and some validations with real plant; implementation of the ECU, software architecture, preliminary tests the ability of the control hardware to solve the optimal control problem, and performance testing of complete scheme running two different types NMPC speed controllers.

3.1 Implementing the model

Table 1 corresponds to the set of constants and initial values for the model captured in Fig. 3, which correspond to the constructive parameters of bench tests as described in [6]. Some functions and parameters as those for the volumetric and thermal efficiencies were taken from the literature, specifically [5], but also in the results of [6], use has been validated with experimental data combustion engine with the corresponding error study. It has taken that job validating two state variables in Figs. 12 and 13.

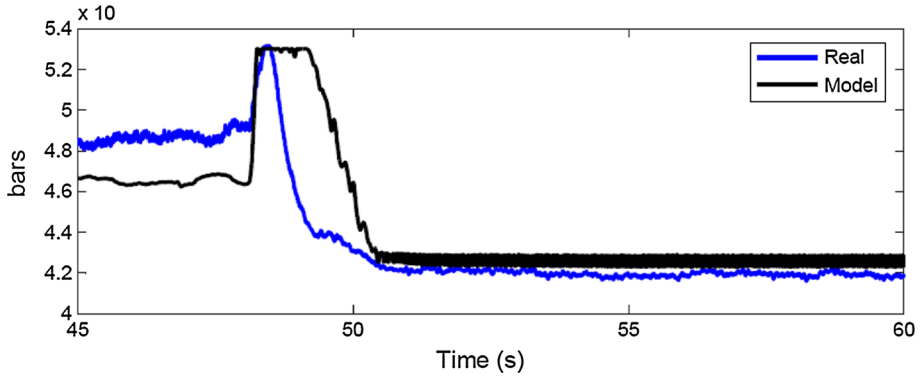


Fig. 12 Validation for P_m (from [6])

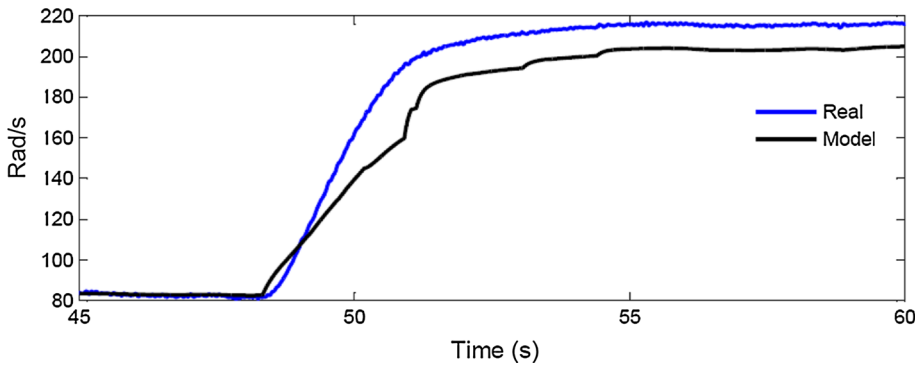


Fig. 13 Validation for n (from [6])

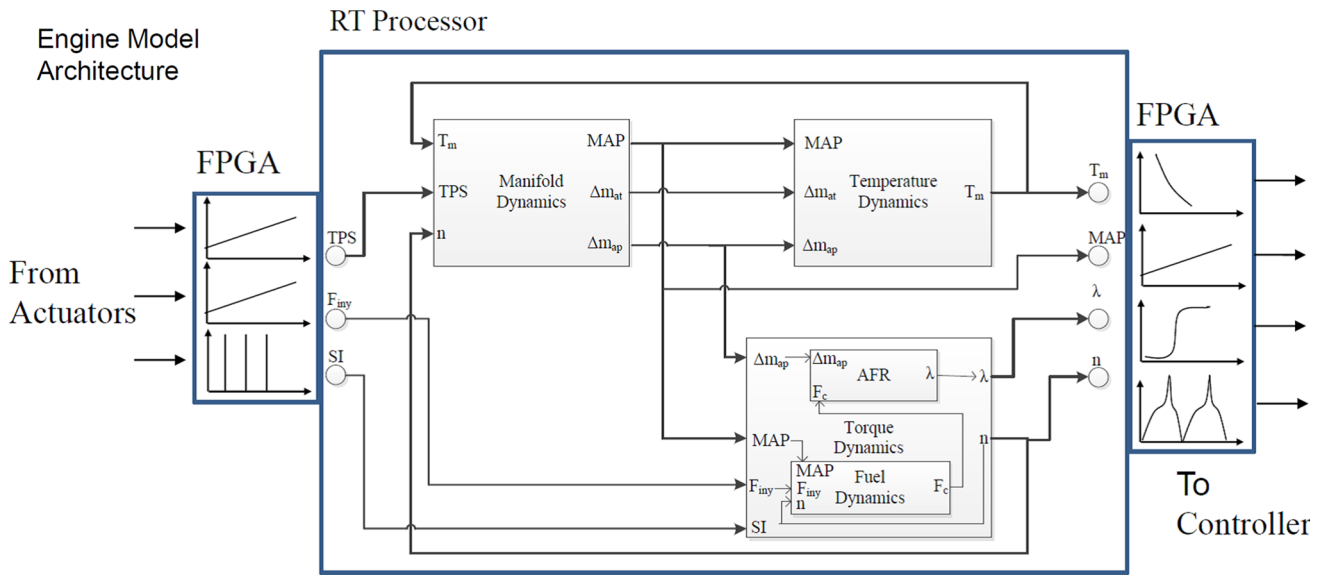


Fig. 14 Final model unit architecture

As mentioned in the materials section model implemented in the platform sbRIO running on the microprocessor device under ODE solver. Furthermore FPGA hardware realizations were added for simulating the physical behavior of the sensors, therefore, sbRIO platform on the full set was simulated engine-sensors. Real actuators have been used in the control loop, and the signals that identify the actuator drive function as inputs to the model. A conversion scale units and treatment through the FPGA hardware was implemented to translating electrical signals into numerical values in engineering units. A representation of the resulting scheme is shown in Fig. 14.

In Figs. 15 and 16 captures model simulation tests are shown in their free response to changes in α and m_ψ inputs. ODE solver parameters and initial conditions are presented in Table 2.

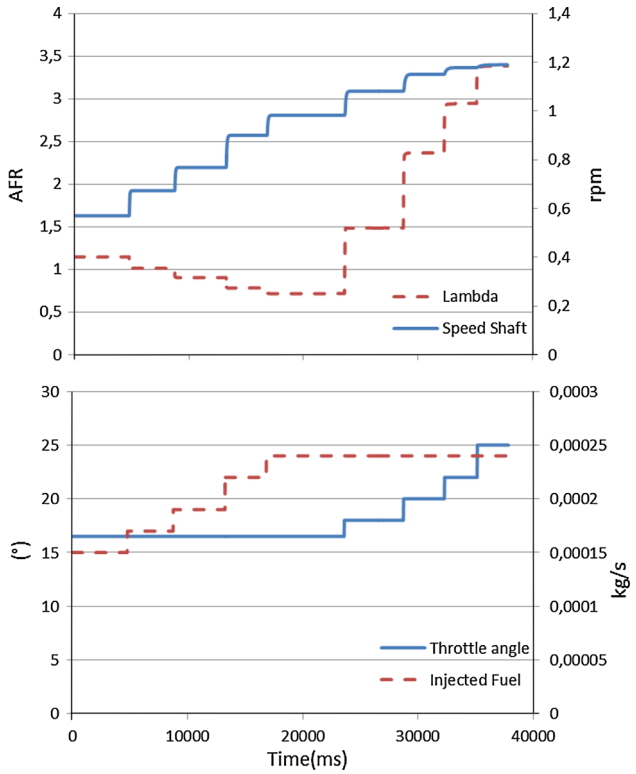


Fig. 15 Model behavior simulation (low speed)

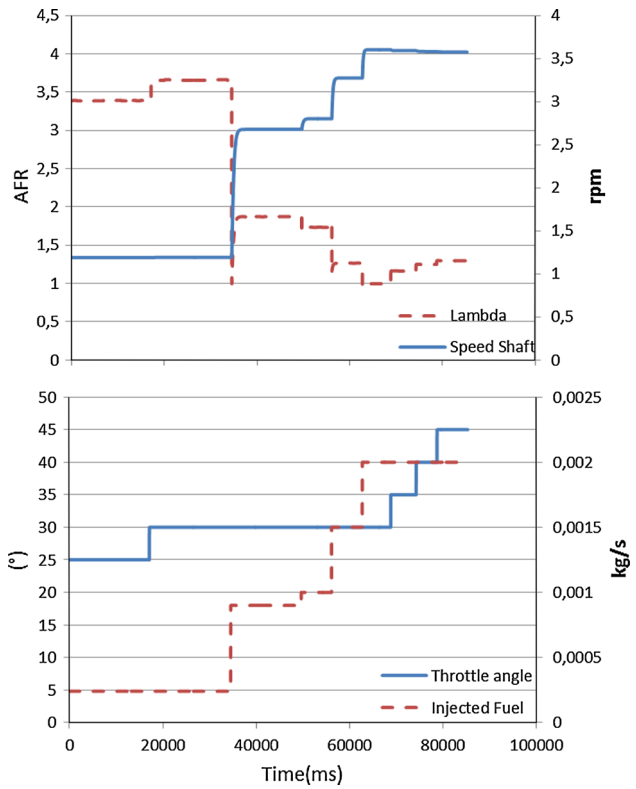


Fig. 16 Model behavior simulation (medium speed)

Table 2 Parameters used for model simulation

Parameter	Value
Solver method	Runge–Kutta23 (variable size)
Initial step size	0.01s
Minimun step size	100 ps
Relative tolerance	0.001
Absolute tolerance	1E−7
CI n	0.8 krpm
CI P_m	0.2393 bar
CI T_m	293 °K

The upper and lower parts of the figures are synchronized in time; in Fig. 15 stimulations low speed zones are shown. Changes successive step types are applied on the input m_ψ leaving the opening angle of the butterfly constant, immediate response showing the shaft speed and enriching the air/fuel mixture. Then, the flow of injected fuel is left constant and stimulation is performed on the angle α. The responses are strong changes in the AFR impoverishing mixture, and even slight noticeable changes occur in the speed of the engine. Temporal responses are comparable to the results of validation of [6], should take into account differences in the time scales.

In Fig. 16, the behavior of the model led to a scheme shown more accelerated greater 3 krpm, shown that changes in the quantity of fuel injected is primarily responsible for the dynamic changes of the rotational energy of the machine. Notice how a change in openness does not generate noticeable fluctuations in the dynamics. Rising fuel immediately reflects changes in AFR.

The results presented in the graphs are given in real time taken for data acquisition systems from the third component of the platform named "Interface Operation and Monitoring" running on a Windows desktop operating system. Table 2 provides the rigors model for implementation with relatively small time steps and low error tolerances to ensure sufficiently high accuracy and numerical resolution.

3.2 Implementation of control unit

The second element of the 2-HIL architecture corresponds to the ECU. The technology platform to build is similar to that used to implement the model, but with the main feature of increased robustness technology, so much in their enclosure like protection from electromagnetic interference, with degree of protection IP 40. This feature allows use with confidence in an industrial environment and its use in the automotive field; this supports the goal of providing connectivity plug and play mode.

The platform modules have power outputs suitable for handling injectors and ignition coils installed, power output as the motor driver to butterfly, besides conditioning systems TPS signals, CKP, MAP, temperature sensor and lambda probe.

As preliminary tests to verify the ability of the platform processor for executing operation on the NMPC controller combustion engine, the Active Set and SQP algorithms were checked, using commercially available software components for use in real-time platforms.

Functions as "Quadratic Programming.vi", "Constrained Nonlinear Optimization.vi" and "SIM Linearized.vi" are software units coded in LabVIEW and copyrighted from National Instruments used for testing in solving the optimization problem of engine operation, as well as other programming tools and solvers of differential equations.

To solve the discrete problem formed from 19 to 21 must be rewritten in the form of a general quadratic programming problem. To do this, the two investigated formulations are called the spread out formulation and formulation condensed [34]. In the second, the states are algebraically operated for the decision variables correspond only process inputs, whereas the former considers the whole set as decision variables. Both methods somehow increase the weight coefficient matrix of the objective function and the coefficients of the model to set up a problem that covers the prediction horizon. This treatment can be consulted in the literature referenced.

Table 3 shows the times taken by the RT processor to solve an optimization problem. Processing time taken to solve the spread out and condensed linearized formulations solved with the algorithm Active Set for different prediction horizons are compared, and the time taken for the SQP algorithm for nonlinear solution for three prediction horizons.

Table 3 Processing times in the solution of the problem

Np	Scattered QP (ms)	QP condensed (ms)	SQP (ms)
0	1	NA	–
1	2	3	–
2	9	8	–
3	19	14	–
4	39	24	–
5	180	91	58–115
6	4302	165	–
7	–	328	–
8	–	342	–
9	–	154	–
10	–	192	67–190
15	–	384	–
20		–	105–284

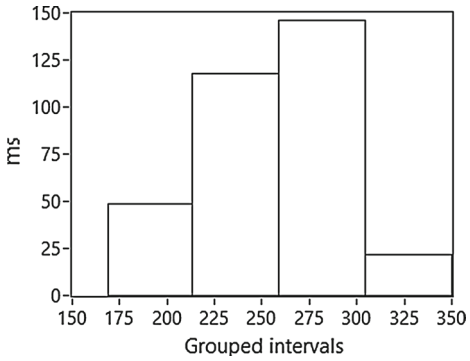


Fig. 17 Linearization processing time

According to the time taken for the transition period of the motor response 2 s controlled factory ECU, taken from the experiments for model validation (see Fig. 13), the controller must solve the problem in less time than 200 ms. The table shows that the formulation scattered lower times were achieved at 200 ms for prediction horizons of five sampling periods, meaning that higher prediction horizons are not practical. Is desirable to use small sampling periods, but this may mean that the model does not estimate the full transient behavior to reach steady state; and large sampling periods subtract accurately the solution implications of instability and steady state error within a control loop. The condensed formulation allows defining prediction horizons wider, but it should be noted that also takes significant linearization processing times. Figure 17 shows a histogram 350 of a series of processing time measurements for different points of operation. It can be inferred that the time it takes the linearization is 225–300 ms. The variability is mainly due to the linearization algorithm must find that the operating point is a point of steady state (Fig. 17).

The SQP algorithm shows great superiority, apart from not needing to use the linearization, can be defined prediction horizon of up to 20 sampling periods. A drawback can be seen, when very short horizons are used, since the ODE solver would cut off simulation before reaching steady state and the solution found may not be enough to bring the state to the end point due to information biased of the solution. One possible solution, as specified in the literature of SQP method, meets the requirement that the initial vector is sufficiently close to the optimum.

Having verified the feasibility to solve the optimization problem in the time required, is implemented and checked the NMPC controllers with the necessary components to support engine operation:

- PI controller for the throttle position
- Knocking compensator
- Synchronizer spark

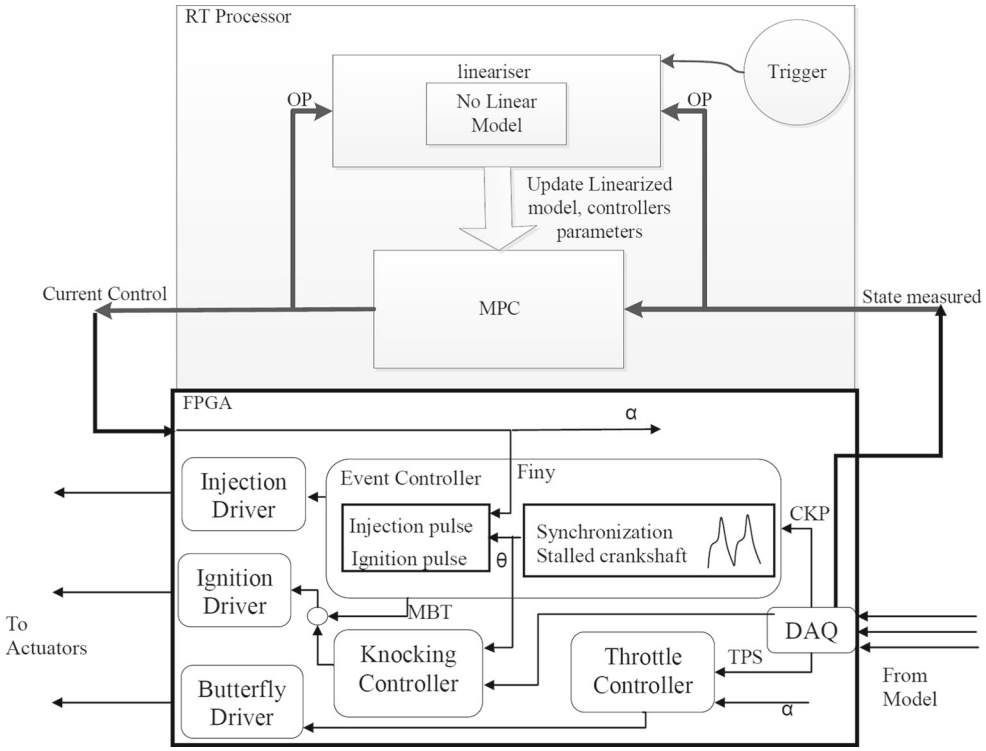


Fig. 18 Final control unit architecture

- Synchronizer injection
- Scaling the sensor units

The implementation details of these components are not reported here, but generalizations about it can be found at [7]. In Fig. 18, the architecture of the implementation of the control unit presents. These elements are implemented at the hardware level FPGA. The figure shows the representation linearization NMPC, however remains the same architecture with the NMPC SQP based algorithm.

The remaining figures show the initial performance test system control 2-HIL with both strategies NMPC control.

Figure 19 shows the behavior of NMPC based on a standard MPC controller with linearization in the current operating point. MPC standard solves the problem of optimization in each period using a quadratic programming algorithm QP.

Various schemes for updating the linear model were checked:

(a) MPC controller action: after starting the engine and a few seconds of open loop control is activated.
(b) MPC controller action: Before starting the engine controller is active. Setpoint changes are shown in lower operating areas. The reference is achieved in 200 ms with zero offset.
(c) Linearization offset: in error detection, linearization is triggered. Middle performance operation area in shown.
(d) Performance at higher operating areas: uncorrectable error, active constraints.
(e) Timed linearization: is linearized every 100 ms and MPC parameters are updated a varying overshoot, occurs in most low ranges.
(f) Linearization change reference: linearized whenever the reference changes. Offset and overshoot is corrected, but has a small delay in the response

In Fig. 20, the simulation results of the control unit executing a NMPC based on an algorithm presented SQP. Tests on various areas of operations have been checked: in a. behavior shown in lower operating zones, a settling time of 400 ms is observed, zero steady-state error; as shown in b. solution time affects loop response inputting a dead time and in areas of higher operation as shown in c. offset becomes more critical.

3.3 Implementation of the operator interface

Sustained architecture for implementing each component and complete system 2-HIL is recommended for CompactRIO systems the manufacturer for embedded applications running in real time. Figure 21 shows part of the view of implementation adapted to the case. The breakdown of the components of the lower platform already described. The upper platform, which as stated is a desktop with Windows operative system, contains a set of software components framed in an appli-

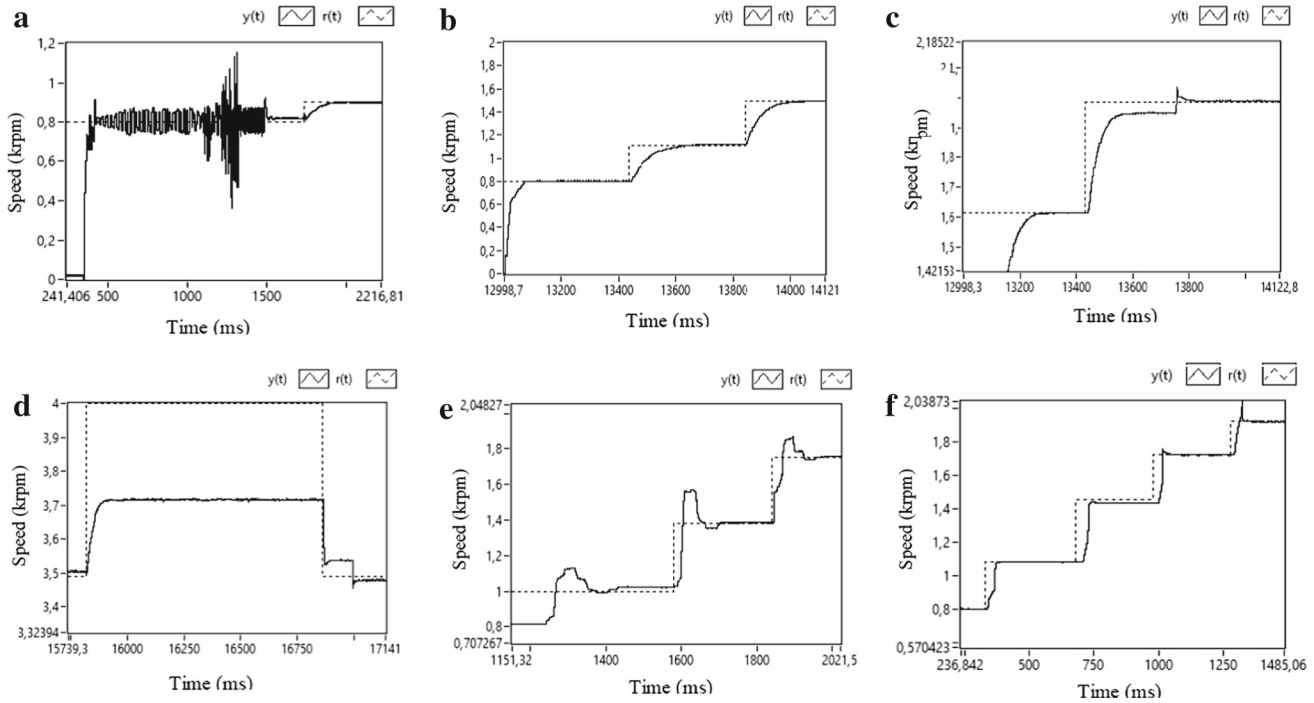


Fig. 19 NMPC with SQP

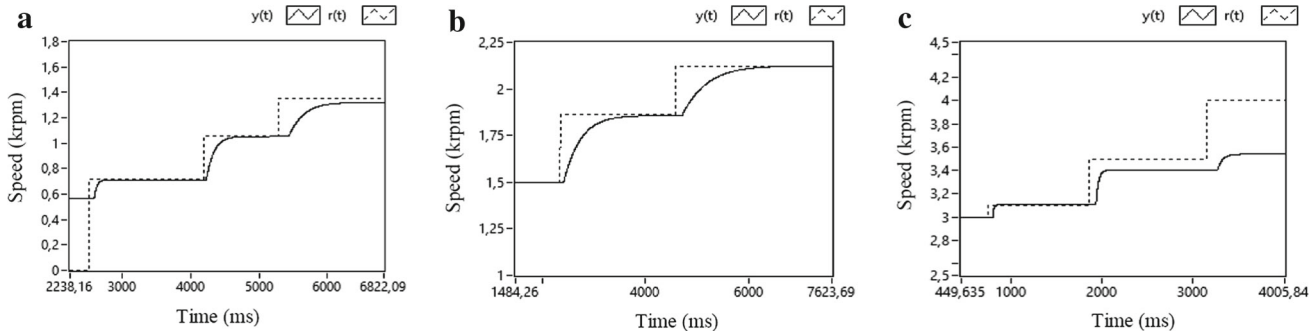


Fig. 20 NMPC with QP

cation denoted as "USER INTERFACE VI". VI, referred to by the acronym "Virtual Instruments", as it is commonly referred to applications built with the software tools that have been used. This set of components expose a user interface that allowed strong interactive experimentation with components of the other two platforms for the proper design of NMPC controllers which reduces to the choice of weight matrices. The components frame the requirements set forth in Sect. 2.4, e.g. the component "UI Tuning controllers" has mechanisms to update the matrices F, Q y R, and store them along with the responses obtained, that become input in the "Analysis tools" component; in the component "UI Scenario Building" paths are defined for both speed n to λ, the bounds are set for restrictions, and trigger methods are switched for linearization processes for linearization based NMPC, disturbances are inserted, and is simulated in the loaded motor shaft. The construction of the components and their interrelationships with other components and other components in the other platforms was supported on software architecture recommendations, and its detailed description is beyond the scope of this report.

3.4 Final discussions and next steps

The NMPC controllers with QP show acceptable behavior to follow the reference from different model updating schemes, usually changes speed reference are not presented in the form of step, but as a ramp (assume that the reference is given by a user pressing the accelerator pedal) this necessarily reduces overshoot shown in e. and operating point changes are gradual, so error correction due to a bias in the point of operation is milder than in c.

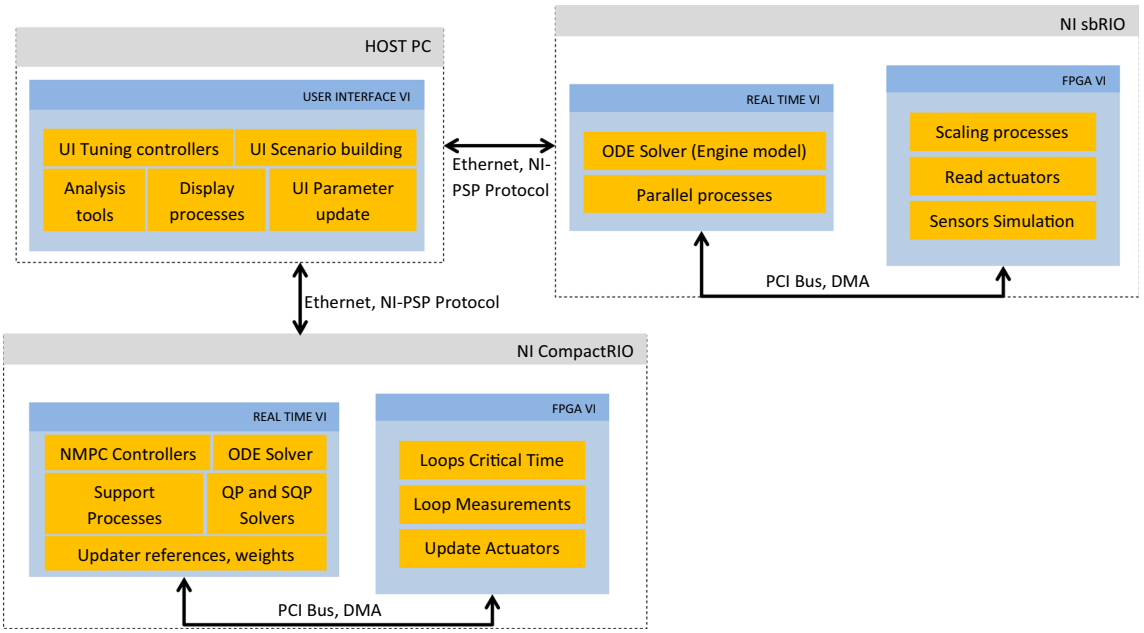


Fig. 21 Implementation view

The operation with the second NMPC with SQP, is also satisfactory carrying the engine speed the reference in very short times, susceptible of improvement, taking lower prediction horizons. As mentioned, this can lead to a false optimum because of biased information, but this can be corrected by implementing mechanisms that generate feasible initial points much closer to the optimum, this also would correct the steady state error present in c (Fig. 19).

With the above results it is and validates the proposed platform as a possible means for the design and testing of ECUs for internal combustion engine ignition. Another check is that the computational power of the equipment proposed as control elements allows the introduction of NMPC control algorithms the plant in question and equivalent engines. This will result in the ECU's ability to design for optimization on all three fronts: environmental, economic and energy, either simultaneously or dynamic objective functions.

The component corresponding to the model, is likely to be improved from different points of view, among them: corrected by adaptive algorithms experimental polynomial functions used for the thermal and volumetric efficiencies, as well as models of the time constants for dynamic fuel presented in 8; subjecting it further validate the component to be controlled by the ECU factory.

Reduction in time can be achieved from different points of view: linearisation process can be downloaded to FPGA hardware level with corresponding increase in computational speed that entails, Active Set and SQP algorithms specially tailored to solve the problem at hand can leverage the structure of the problem and improve the matrix operations involved.

The results in the operation of the algorithms show a possibility for the feasibility of practical embodiment for the optimal control of internal combustion engines without use of cartographic injection.

References

1. Penney, X.J.: Managing the implementation of automotive emissions control technologies using system engineering principles. Master thesis, System Design and Management Program, Massachusetts Institute of Technology, Boston (2007). http://hdl.handle.net/1721.1/34737. Accessed 10 Nov 2013
2. Krupp, D.M., Hood, R.R.: Electronic control system. Patent 3,893,432, 1975
3. Hartford, T.W. et al.: Microprocessor-based electronic engine control system. Patent 4,255,789, 10 March 1981
4. Cox, R.W.: GM Emission Control Project Center—I was There. Generations of GM History (2011). http://history.gmheritagecenter.com. Accessed 10 Nov 2013
5. Hendricks, E., Sorenson, S.: Mean value modelling of spark ignition engines. SAE Technical Paper 900616 (1990). doi:10.4271/900616
6. Hernández, L.M.: Modelo Matemático de Tipo Valor Medio de un Motor de Combustión Interna y Estimación de Estados usando Técnicas Bayesianas (Master Tesis). Universidad Eafit, Medellín, Colombia (2012). http://hdl.handle.net/10784/1291. Accessed 10 Nov 2013
7. Guzzella, L., Onder, C.H.: Introduction to Modeling and Control of Internal Combustion Engine Systems, 2nd edn. Springer, Germany (2010)
8. National Instruments Corp.: What operating system is my real-time controller running and why? (2008). http://digital.ni.com/public.nsf/allkb/35F1FD98520D6E0E8625783A005AF557. Accessed 8 July 2014

9. Measumerent Specialities TM.: Glass Encapsulated Style. St. Marys, Pensilvania (2013)
10. Robert Bosch GmbH: Piezoresistive absolute-pressure sensors in thick-film technology. Stuttgard, Germany (2013)
11. Tech Edge (Pty) Ltd.: NTK L1H1 Sensor Information (2003). http://wbo2.com/~techedge.com.au/vehicle/wbo2/wbntk.htm. Accessed 15 May 2013
12. Europea Unión: Euro 5 and Euro 6 standards, Regulation of the European Parliament and of the Council on type approval of motor vehicles with respect to emissions from light passenger and commercial vehicles (Euro 5 and Euro 6) and on access to information relating to the repair and maintenance of vehicles. Legislative acts and other instruments, PE-CONS 3602/2/07 REV 2, 1–35. Bruselas (2007)
13. Nash, F.C.: Mecánica Automotriz: teoría, mantenimiento y reparación, vol. 2. McGraw-Hill, Mexico (1991)
14. Nevot, C.J.: Diseño de un Controlador Avanzado Basado en Redes Neuronales para la Gestión de la Mezcla Aire-Gasolina en un Motor Alternativo. PhD thesis, Universidad Politécnica de Cataluña, Barcelona (1999)
15. del Re, Luigi, Ornert, P., Alberer, D.: Chances and challeges in automotive predictive control. In: del Re, L., et al. (eds.) Automotive Model Predictive Control, LNCIS 402, pp. 1–22. Springer, Heidelberg (2010)
16. Subbaram, N.D.: Optimal Control Systems. CRC Press, Idaho State University, USA (2002)
17. Zhai, Y.J., Yu, D.L., Tafreshi, R., Al-Hamidi, Y.: Fast predictive control for air–fuel ratio of SI engines using an nonlinear internal model. Int. J. Eng. Sci. Technol. **3**(6), 1–17 (2011)
18. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, USA (2006)
19. Powell, M.J.D.: An efficient method for finding the minimum of a function of several variables without calculating derivates. Comput. J. **7**, 155–162 (1964)
20. Holland, J.H.: Adaptations in Natural and Artificial Systems. University of Michigan Press, Michigan (1975)
21. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**, 308–320 (1965)
22. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statical problems. J. ACM (JACM) **8**(2), 212–219 (1961)
23. Biegler Lorentz, T.: Nonlinear Programming. Concepts, Algorithms and Applications to Chemical Proccess. Carniege Mellon University, Pittsburgh, SIAM, Pennsylvania (2010)
24. Lie, B., Dueñas, D.M., Hauge, T.A.: A Comparison of implementation strategies for MPC. In: Proceedings of International Symposium on Advanced Control of Industrial Processes, Kumamoto (2002)
25. Allgöwer, F., Findensein, R., Nagy, Z.K.: Nonlinear model predictive control: from theory to application. J. Chin. Inst. Chem. Eng. **35**(3), 299–315 (2004)
26. Findeisen, R., Algower, F.: An introduction to nonlinear model predictive control. In: 21st Benelux Meeting on Systems and Control (2002)
27. Diehl, M., Ferreau, H.J., Haverbeke, N.: Efficient numerical methods for nonlinear MPC and moving horizon estimation. Lect. Notes Control Inf. Sci. **384**, 391–417 (2009)
28. Chisci, L., Falugi, P., Zappa, G.: Gain-scheduling MPC of nonlinear systems. Int. J. Robust Nonlinear Control **13**, 295–308 (2003)
29. The MathWorks Inc: Gain-scheduled MPC (2013). http://www.mathworks.com/help/mpc/ug/gain-scheduling-mpc.html. Accessed 30 May 2013
30. Ogata, K.: Ingeniería de Control Moderna. Pearson Prentice Hall, Madrid (2003)
31. Martinsen, F., Biegler, L.T., Foss, B.A.: A new optimization algorithm with application to nonlinear MPC. J. Process Control **14**(18), 853–865 (2004)
32. Balsa, C.E.: Algoritmos Eficientes para la Optimización Dinámica de Procesos Distribuidos. PhD thesis, Universidad de Vigo, España (2001)
33. Boiroux, D.: Nonlinear Model Predictive Control for an Artificial Pancreas. Master thesis, Technical University of Denmark, Lyngby (2009)
34. Ferreau, H.J.: An online active set strategy for fast solution of parametric quadratic programs with applications to predictive engine control. Diploma thesis, University of Heidelberg (2006)