Int J Interact Des Manuf (2018) 12:297–318
https://doi.org/10.1007/s12008-016-0343-2

TECHNICAL PAPER

NMPC controller applied to the operation of an internal combustion engine: formulation and solution of the optimization problem in real time

José Albeiro Valencia Chica[1] · **Adalberto Gabriel Díaz Torres**[2] · **Diego Andrés Acosta**[2]

Received: 17 June 2016 / Accepted: 13 July 2016 / Published online: 22 July 2016

Abstract Numerical optimization solve problems efficiently where such efficiency is focused on the speed with which the optimal $\boldsymbol{x}^*$ is achieved, is open line of research and strong work in the scientific community in order to achieve control systems in dynamic processes with response times of the order of milliseconds. A clear example of this, is the implementation of optimal controller's combustion engines. For subsequent approach to the design and implementation of nonlinear model predictive control controllers, it has made a comparison of yields algorithms quadratic programming by active set with linearization restrictions, and sequential quadratic programming with single shooting technique to solve quadratic optimization problem formulation referred to a dynamic internal combustion engine of spark ignition, in embedded systems with real-time processing.

Keywords Simulation · Hardware in the loop · Automotive · Optimal control nonlinear · Real time control · SQP and QP algorithms

✉ José Albeiro Valencia Chica
albeirovalencia@misena.edu.co

Adalberto Gabriel Díaz Torres
gdiaz@eafit.edu.co

Diego Andrés Acosta
dacostam@eafit.edu.co

1 Servicio Nacional de Aprendizaje SENA, Dg 104 # 69-120, Medellín, Colombia

2 Universidad EAFIT, Cra. 49 #7sur-50, Medellín, Colombia

1 Preface

Internal combustion engines have allowed boost industrial growth in the world with its development lead to the emergence of the automobile and the airplane and private transport industry took a significant place in the world occurred. Passenger cars, especially demand the use of internal combustion engines with spark ignition (SI), making them responsible for a high percentage of pollutant emissions. The goal of better and more efficient internal combustion engines SI is justified by the above and the cost resulting from the use of fossil fuels. Several lines of investigation were directed on that route, some from the looks of new designs, and others from the operation itself of the motor. As in this report, part of the effort to find a more optimal operation of SI combustion engines, it has been supported on modern experimental schemes such as simulation hardware-in-the-loop (HIL). This technique allows you to check the engine control system without actually connecting the motor to its controller, rather the controller is connected to a motor model such that eliminates complexity, reduces costs experimentation, and security risks. Has been implemented scheme with two embedded systems, one containing a model of the mean value of the engine and the other as ECU, including in the loop operation the actuators involved: ignition coils, injectors and throttle valve, and an operator interface, monitoring and parameter setting. The scheme then encompasses a system that allows you to design electronic control units for combustion engines SI, allowing interactively make changes parameters, and evaluating destructive limits for the machine, to design efficient control strategies. In [1] the results and overall conclusions interactive simulation system are presented, in [41] an assessment of the control systems applied NMPC strategies presented, in the current report the formulation of the optimization problem and solution strategies applied in real time is presented.

2 The problem

In considering the optimal operation of the combustion engine as part of the research process have been set up three fronts of high interest and of use: an energy front, an economic front and an environmental front. Every front is susceptible to attacking many forms, as changes in the design of the mechanisms of the engine, using alternative fuels, implementing new catalyst systems, however, this work is aims at improving the operation of an engine test bench as it is, only managing resources and the operation itself with an optimal policy. It is usual in combustion engines with spark ignition, using an embedded processing system to regulate the management of such resources. Then, the end is to use an embedded processing system to find and implement the optimal policy, and that policy contemplates the three fronts of interest.

A concrete and apply optimal policy option is in the implementation of numerical optimization algorithms. The variables involved serve as indicators of optimality of engine operation. To involve all three fronts of interest have been selected variables and/or indicators that directly or indirectly affect every front. For the energy front, the use of the maximum possible amount of energy, resulting in the maximum torque available at the crankshaft. Torque is also responsible for the movement of the motor shaft, is who accelerated or maintained at a reference speed $Refn$, the speed n of the motor shaft, then it can be used as indicator of energy front. As shown in expression 1, the torque produced by the machine, is dependent apart from the constructive aspects of the machine, the amount of fuel that reacts $\dot{m}_\varphi$:

$$\tau_c(t) = \frac{H_\mu.\eta_t(t-\tau_d).\dot{m}_\varphi}{I_e} \quad (1)$$

Other terms are related to the heat capacity of the fuel, the inertia of the machine and an experimental function energy efficiency, these parameters have been expanded in an earlier report [1]. The term immediate and influential selection is the fuel $\dot{m}_\varphi$, that somehow relates to the fuel injected m_{iny}, since increasing the fuel injected under usual conditions, the torque produced increases. However this would punish the front of economic optimization, and there are also restrictions on the amount of fuel to be injected, so that does not involve a drowning from the ignition in the chamber by excess fuel, ie limits the amount of mass needed for combustion [2].

Clearly the economic front is related to the expenditure of resources, and resources related to the operation of the engine are air and fuel, and perhaps very low electric power as the ignition spark. It becomes clear that this indicator is against the injected fuel, and as in the previous case, an immediate solution as arbitrarily decrease the injected flow is not applicable, it must be noted that the injected fuel is not directly reactive fuel, a relationship with the inlet air flow, which in turn depends on atmospheric pressure and temperature, and further there are minimum values of fuel to keep the machine in operation (Idle speed around 750 rpm [3]). Also, change the operating point of the machine, defined by the speed reference, necessarily require change in the minimum fuel injection, according to expression 1.

When referring to a front environmental optimization it is done in the operation of combustion engines, mainly points to index toxic emissions combustion products to the environment. As a mechanism to comply with environmental regulations [4] combustion engines are equipped with a device called a catalytic converter installed in the path of evacuation of combustion gases, and dissolved oxygen sensor. Inside the converter, or catalyst three reactions occur, reduce nitrogen oxides to nitrogen and oxygen, oxidizing carbon monoxide to carbon dioxide and oxidizes unburned hydrocarbons to carbon dioxide and water [2], these products result innocuous for environmental environment. The quality of the reactions depend on the relationship between the amounts of fuel and air react in the cylinders, the aim is that the combustion is complete, consuming all the fuel. For this, theoretically air/fuel ratio (air-fuel ratio AFR) is required to be 14.7 (called stoichiometric ratio), or normalized $\lambda = 1$, determined by measuring the ratio AFR, and stoichiometric. Thus, the deviation to this reference constitutes an appropriate parameter for the front of environmental optimization. The variable measurement is taken by the oxygen sensor, commonly named lambda λ sensor, and it can be regulated by the amounts injected fuel and air flow estimated entering the cylinder according to the ratio 2.

$$AFR = \frac{m_\propto}{m_\varphi} \rightarrow \lambda = \frac{AFR}{AFR_e} \quad (2)$$

Figure 1 represents the efficiency of the chemical process catalyst. Only when the normalized stoichiometric ratio is in a neighborhood about 1, maximum efficiency of the converter is presented in the three reactions simultaneously. Although the efficiency of conversion of carbon monoxide and hydrocarbons improve as the ratio air/fuel increases, the conversion of nitrous oxide is lost drastically negatively affecting the environmental front, and the opposite is also true. A narrow window of λ variable related to environmental regulations applied, can define constraints decision variable environmental front.

As mentioned, the objectives can come into conflict contemplating the three fronts, although it is usual to find a compromise and look multi-objective systems [5]. Another

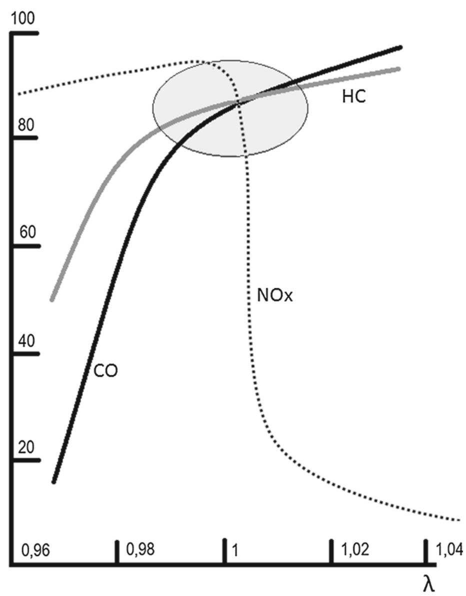


Fig. 1 Working window of the catalytic converter

way might be to switch the objectives according to the dynamic changes of operation of the machine, for example, in the transitional enable optimization that seeks the maximum power delivered, temporarily sacrificing produced minimizing harmful gases and fuel consumption. Both schemes can be implemented with relative ease optimization considering structures proposed by the optimal control theory, addressing the control of the machine as a dynamic optimization problem solved at each control period.

3 Materials and methods

3.1 Dynamic optimization problem

Engineering, most running processes and the machines are susceptible of improvement, many schemes to find the optimum point of operation have been established in the literature, and a good overview is given in [6]. The motor optimization problem is tackled in this paper as a problem of non-linear programming (NLP), especially due to the nonlinear behavior of the plant described in [7]. From the viewpoint of NLP, in an optimization problem distinguish three elements: a function of the decision variables that maps to a scalar whose value is to be maximized or minimized, this expression is typically called objective function; a predictive model that describes the behavior of the system to optimize, consists of a set of equations that define equality and inequality constraints, in case the model is formed by the equations of state and the limits on state variables and input; and a set of decision variables which form a subspace of the state variables appear constraints and the objective function and, usually in control processes, are the manipulated variables or inputs to the plant. A general formulation of the problem of non-linear programming is presented in 3.

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} f(x) \\ s.a. & \\ & h(x) = 0 \\ & g(x) \leq 0 \end{aligned} \tag{3}$$

$f(x)$, is the objective function, while $h(x)$, relates the set of equality constraints, particularly formed by the equations of dynamics model, and $g(x)$ represents a set of inequality constraints that delimit the feasible solution region. Under steady-state engine operation can be conducted under formulation 3 and their methods of solution [6,8], TPS and $\dot{m}_{iny}$ get to the desired operating point, and use compensators for dealing with disturbances. Furthermore, with NLP can reformulate the optimization problem such that dynamic considerations involved, facing disturbances and transients from the solution itself. Specifically, this reformulation and solution methods relate to problems of dynamic optimization and within the optimal control theory. The formulation of the problem of nonlinear programming for a dynamic system described by a set DAE (differential-algebraic equations) usually made in a general way as in [9]:

$$\begin{aligned} & \min f(x(t), u(t), t) \\ s.a. \quad & \frac{dx}{dt} = f(x(t), y(t), u(t), t); x(t_0) = x_0 \\ & g_I(x(t), y(t), u(t), t) \leq 0 \\ & g_E(x(t), y(t), u(t), t) = 0 \end{aligned} \tag{4}$$

The performance index is then a function of the state of the excitation or inputs to the system and time. In turn, the dynamics of state equality constraints introduced the feasible set of solutions to an initial value determined, and g_I and g_E represent the set of expressions with equality constraint and additional inequality may have. Problems arising in optimal control, the objective function or performance index involves the final time the problem is observed, and an integral of a functional over time [8]:

$$\min J = S(x(t_f), t_f) + \int_{t_0}^{t_f} V(x(t), u(t), t)dt \tag{5}$$

The first term of the expression is called the terminal or final cost and the term involving the integral is a functional that can measure the optimal time, minimum cost and minimum energy required to bring the dynamic system from an initial state $x(t_0)$ to a final state $x(t_f)$. It is usual in modern control problems [10] that the dynamic system under control reaches a constant reference or follow a path to an exit or state variable chosen, and speaking of optimization should be kept to a minimum resource and energy used for that purpose. Minimize energy expenditure with a k rate of resource consumption in the control time it can be established by:

$$\min J = \int_{t_0}^{t_f} ku(t)^2 dt \tag{6}$$

Generally speaking, in state variables,

$$\min J = \int_{t_0}^{t_f} u^T \boldsymbol{R} u(t) dt \tag{7}$$

In this case u, represents the set of inputs to the dynamic system and $\boldsymbol{R}$, is a weighting matrix defining the influence of each input on the result of the objective function. Similarly, you can establish a performance index punishing the integral of error for a final reference and/or for a given path, such that 5 is redefined as:

$$\min J = x_d(t_f)^T \boldsymbol{F} x_d(t_f) + \int_{t_0}^{t_f} \left[x_d(t)^T \boldsymbol{Q} x_d(t) + u^T(t) \boldsymbol{R} u(t) \right] dt \tag{8}$$

With $x_d(t_f)$, defined as $x_d(t_f) = Ref_x - x(t_f)$ and $x_d(t) = Ref_x(t) - x(t)$. Ref_x is the final position for the state variable, while $Ref_x(t)$ denotes the path in the control range. Similarly, matrices are elements weighing deviations from the state variables against the given references.

With the decision variables n and λ, then the performance index is defined by the error to the desired speed reference and the standard deviation of stoichiometric λ, which is 1. The speed reference $Refn$, is an unknown continuous function, usually set by the operator of the machine within the allowable limits. The input variables are defined by the fuel flow injected m_{iny} and opening the butterfly TPS that relates to the air mass in the combustion, then 8 can be specified as:

$$\begin{aligned} \min J = & \begin{bmatrix} Refn - n & 1 - \lambda \end{bmatrix} \begin{bmatrix} F_{11} & 0 \\ 0 & F_{22} \end{bmatrix} \begin{bmatrix} Refn - n \\ 1 - \lambda \end{bmatrix} \\ & + \int_{t_0}^{t_f} \begin{bmatrix} Refn - n & 1 - \lambda \end{bmatrix} \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{22} \end{bmatrix} \begin{bmatrix} Refn - n \\ 1 - \lambda \end{bmatrix} \\ & + \begin{bmatrix} TPS & m_{iny} \end{bmatrix} \begin{bmatrix} R_{11} & 0 \\ 0 & R_{22} \end{bmatrix} \begin{bmatrix} TPS \\ m_{iny} \end{bmatrix} dt \end{aligned} \tag{9}$$

The elements of F, Q and R have meaning adjustment trend forward desired optimization, albeit it is particularly relevant for the performance of optimal control loop, for evaluation of algorithms, it is sufficient to comply with and be properly defined, this for the convergence of the algorithms. In [1] they settled considerations regarding the performance of the control.

With the performance index defined by 9, and the equality constraints by model, only it needs to define the bounds for the state variables, outputs and control signals. The previous report of this project [1] the definitions of the dimensions are taken up again: in the experiment obtained in [7] to validate the model, the minimum speed that keeps the engine operation is 0.6 krpm, and the maximum is obtained from the technical manual for engine operation is 6 krpm. The maximum opening angle set for the butterfly valve is 90° geometry, and the minimum to maintain the operation defined by the minimum possible engine idling opening 16° geometry. The coordinates for injection flows are taken from the experimental data at idle speed and 6 krpm, but because these experiments are not free response (original ECU has an estimator according injection manifold pressure, besides being regulated by measuring the oxygen sensor) have been tested by extending the range of variation by 15 %, then the flow rate is 0.00015–0.005 kg/ms. Referencing literature AFR control [11], allowable excursions is estimated 10 % of the stoichiometric ratio. The usual, for this window is ± 3 % [12], in this range trips to rich and/or lean not considered harmful. However the permissiveness that is determined in this aspect can extend freedom in action control for tracking the speed reference and no borders and toxic emissions limits. The bounds for the signal λ are given from 0.90 to 1.1. The remaining state variables temperature T_m and manifold pressure P_m (MAP) are naturally bounded in engine operation, dependent as described by the model, ambient temperature, *TPS* and atmospheric pressure, so no bounds for finding the solution are added. These restrictions are grouped in the set of expressions 10.

$$\begin{aligned} 0.00015 < m_{iny} < 0.003 \\ 16 < TPS < 90 \\ 0.6 < n < 6 \\ 0.9 < \lambda < 1.1 \end{aligned} \tag{10}$$

The full optimization problem with the objective function, the mathematical model that introduces equality constraints and dimensions to the decision variables is established in 11. 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 taken by a set of measurements. The relationships of the variables are treated in detail in [1,7].

$$\begin{aligned} \min J = & \begin{bmatrix} Refn - n & 1 - \lambda \end{bmatrix} \begin{bmatrix} F_{11} & 0 \\ 0 & F_{22} \end{bmatrix} \begin{bmatrix} Refn - n \\ 1 - \lambda \end{bmatrix} \\ & + \int_{t_0}^{t_f} \begin{bmatrix} Refn - n & 1 - \lambda \end{bmatrix} \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{22} \end{bmatrix} \begin{bmatrix} Refn - n \\ 1 - \lambda \end{bmatrix} \\ & + \begin{bmatrix} \propto (TPS) & m_{iny} \end{bmatrix} \begin{bmatrix} R_{11} & 0 \\ 0 & R_{22} \end{bmatrix} \begin{bmatrix} \propto (TPS) \\ m_{iny} \end{bmatrix} dt \end{aligned} \tag{11.1}$$

$$
\begin{aligned}
&s.a.\\
&d\frac{P_m(t)}{dt}\left(\frac{V_m}{R\kappa}\right)-\dot{m}_\alpha\,(t)\,T_{amb}-\dot{m}_\beta(t)T_m=0\\
&\frac{dT_m}{dt}-\frac{RT_m}{P_mV_m}\left[m_\alpha(\kappa T_{amb}-T_m)-m_\beta(\kappa-1)T_m\right]=0\\
&\frac{dn}{dt}-\frac{1}{I}H_u\eta_t(P_m.n,\lambda)m_\varphi(t-\Delta t_d)-\frac{1}{I}\tau_l=0 \qquad (11.2)\\
&\dot{m}_\varphi=\dot{m}_{fv}+\dot{m}_{fl}\\
&\dot{m}_{fv}=(1-X)\dot{m}_\psi\\
&\frac{d}{dt}\dot{m}_{fl}-(1/\tau_{fl})(-\dot{m}_{fl}+X\dot{m}_\psi)=0
\end{aligned}
$$

$$
\begin{aligned}
&0.00015<m_{iny}<0.003\\
&16<\propto(TPS)<90\\
&0.6<n<6 \qquad (11.3)\\
&0.9<\lambda<1.1
\end{aligned}
$$

In the following possible solutions 11 are explored, that applies or are coupled to a control scheme in real time.

3.2 Solution of the problem of optimization

Methods to solve 11 have been analyzed in the literature [6,13] first as problems not restricted, and from there they have made modifications to implement the methods developed solutions restricted space. The optimum point of J may be out of feasible solutions due to the siege imposed by the restrictions, though not restricted optimal provides a path in which optimal path is restricted. J Function can have a global optimum and several local optima. If the optimum is a minimum, any local minimum is larger than the global optimum. Algorithms for finding global optima are commonly higher complexity than focus on finding local optima and the latter with a wide applicability in engineering [6]. Also pointing to the operation of the combustion engine, find the global optimum does not correspond to a practical stage. If the search is to minimize input signals (air and fuel), in [1] show that the overall engine leads to a point of minimum speed operation or idling, and if it is to maximize power/speed shown us that the result is the upper speed limit, fuel flow with the highest possible. The practical thing is to determine an operating point of the engine defined by a speed reference and in the vicinity of this range to find the optimum solution, what constitutes a local solution. The neighborhood would be determined by the points on the surface conforming to the contour of each operating speed, even if the tracking reference points that curve neighbors relaxes also part of the feasible solutions.

To find local optima, have been developed stochastic methods, such as random search adaptive conjugated address, genetic algorithms [14,15], and deterministic methods where the optimum convergence can be estimated. These latter methods are made convenient for applicability control systems, so allowing the solution of the optimization problem in a certain period of sampling. Deterministic are reclassified into direct and indirect methods: direct methods like simplex and complex using a general search pattern [16,17]; and indirect methods which are supported in the calculation of derivatives, so requires that the objective function is smooth with at least three continuous derivatives. For reasons which includes the less calculation times of the objective function and that converge faster, besides allowing applicability given problem develops latter type of methods for the optimization problem of engine operation.

Conveniently, the target 11.1 function is a smooth function, if m_{iny} and λ had a smooth behavior. Now if the formulation of the optimization problem is such that the objective function is quadratic form and linear constraints as shown in 12 [13], then we have a problem of quadratic programming with specific methods to solve it. In fact quadratic forms are beginning the analysis to the search algorithms used in this work.

$$
\begin{aligned}
&\min_{x\in\mathbb{R}^n} f(x)=\frac{1}{2}x^TBx-x^Tb\\
&s.a. \qquad (12)\\
&A_1x=c_1\\
&A_2x\le c_2
\end{aligned}
$$

In Eq. 12, A_1 and A_2 are matrices formed by the coefficients of the variables in the constraints whose dimensions are based on the number of decision variables and the number of restrictions, and B, is called the Hessian matrix, which if positive semidefinite then the quadratic problem is called a convex QP problem. 11 can be made, or the original problem being treated, it can be reformulated as a problem of quadratic programming if the set of Eq. 11.2 is linear, and this implies linearize the model describing the engine. If once linearized model, the control zone is operating close to the point where the model is linearized, solving the quadratic problem corresponds to say optimum operating point [10]. As stated in other sections, this is one of the techniques evaluated in the current report. It will continue to develop computational methods for optimal for a quadratic programming problem, and then in the process of linearization of the model, not without first making the formalization of the characterization of the optimal points on indirect methods of optimization.

3.2.1 Fundamentals

In the iterative methods, problem solving non-constrained optimization, seeking optimal vector x^*, such that the objective function evaluated at said vector is less than any other neighbor vector x of the decision variables $f(x^*)<f(x)$. Starting from an initial value for said vector x_0, each iteration of the algorithm updates the new vector in the generalized form $x_{k+1}=x_k+p_k$. The term p_k corresponds to an

address search function. In indirect deterministic methods that address calculation it is made using the information of the gradient and the Hessian of the objective function, seeking convergence within a given level of precision, ensuring that is solution according to the verification of the optimality conditions. Optimality conditions verify that an extreme point is a local optimum, Using the information yielded by the derivatives in the space $\mathbb{R}^n$, specifically gradient and Hessian. In the literature [6,8,13], the optimality conditions are derived from the Taylor expansion of $f(x)$ around the optimal point x^*, thus:

$$f(x) = f(x^*) + \nabla f(x^*)^T(x - x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*)(x - x^*) + \cdots \quad (13)$$

In the local x^* optimum, verified that $f(x) - f(x^*) \geq 0$, and with the expansion of the first order with a neighborhood close enough, when $f(x) \rightarrow f(x^*)$ then $\nabla f(x^*) = 0$. However this condition can also be a saddle point, which is not optimal. To fully characterize the optimal information can be used curvature of $f(x^*)$. Using the condition of gradient and the expansion of second order Taylor, is obtained $\nabla^2 f(x^*) > 0$.

Hence the conditions for unrestricted Local optimality happen, as set out in [6]:

- If $f(x)$ is twice continuously differentiable and there exists a point x^*, which is a local minimum, then the gradient at that point is zero, and its Hessian matrix is positive semidefinite. Necessary condition.

$$\nabla f(x^*) = 0 \text{ y } {}^T\nabla^2 f(x^*)y \geq 0 \quad (14)$$

- If $f(x)$ is twice continuously differentiable and there exists a point x^*, where the gradient at that point is zero, and its Hessian matrix is positive definite, then that point is a local minimum strictly isolated. With y defined as the vector of eigenvalues of the Hessian.

Optimality conditions define the optimal and quadratic programming methods used to find are based on Newton's method. Such method is an iterative search to find the zeros of the first derivative of $f(x)$ using a convergence value to terminate the iterations. How local method, the initial value has great impact on the convergence of the algorithm. At each step, called step Newton, $f(x)$ is linearized and the new point corresponds to the linear function equal to zero. Developing $f(x)$ in Taylor series, at a point x and p direction, we have:

$$f(x + p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x) p + O(\|p\|^3). \quad (15)$$

Differentiating with respect to the p direction:

$$\nabla f(x + p) = \nabla f(x) + \nabla^2 f(x) p + O(\|p\|^2). \quad (16)$$

With $\nabla f(x + p) = 0$ and $\|p\|$ small, the third term is negligible and obtain the vector p direction as follows:

$$0 = \nabla f(x) + \nabla^2 f(x) p \Rightarrow p = -(\nabla^2 f(x))^{-1} \nabla f(x) \quad (17)$$

For the convergence of the method to the local solution it is important that the Newton step is a downward step, that is to say verification that $\nabla f(x)^T p < 0$, such that the evaluation of the objective function at each new point decreases [6]. Basic steps Newton algorithm is then:

1. Select a start point x_0.
2. Select the limits of convergence ϵ_1 y ϵ_2, for direction and gradient.
3. While $\|p_k\| > \epsilon_1$ y $\|\nabla f(x_k)\| > \epsilon_2$
   a. Evaluate $\nabla f(x_k)$ and $\nabla^2 f(x_k)$. If $\nabla^2 f(x_k)$ is singular, then stop cycle.
   b. Solve the linear system: $\nabla^2 f(x_k) p_k = -\nabla f(x_k)$.
   c. Update to the next iteration $x_{k+1} = x_k + p_k$ and $k = k + 1$.

It can be shown that the rate of convergence for x_k and $\nabla f(x_k)$ is quadratic. However, problems can arise when the objective function is not smooth enough, as each Newton step requires first and second derivatives and, to keep bounded Lipschitz continuity is required in the second derivatives. Furthermore, if the step does not generate a downward direction then it means that the Hessian is positive definite or singular. And in general, initial point too far, and problems of high dimension, resulting in algorithms that do not converge.

The quadratic function within 11 is soft enough, three times derivable, if the restrictions are linearized. With nonlinear constraints, SQP method can be used, as it develops further. The discussion of the starting point, is faced with the problem under analysis, using information from its solution optimization problem in an earlier event. That is, in the context of optimal control of the motor at each instant control algorithm solves an optimization problem, the solution of the problem of the current time can use initial values containing information from the solution of the previous moment. Besides the size of the problem is considered low because the number of decision variables are two, facing equality constraints 5 and 4 inequality constraints. They considered problems of high dimension when you have more than 100 decision variables [6]. Finally, addressing the shortcomings of the basic method of Newton, the Hessian matrix can be modified to ensure you have a limited number of defined condition and remain positive, or avoid using calculation of the

Hessian of the objective function differences in successive iterations.

If the second derivatives of the objective function are available the modified Hessian can be constructed as:

$$B_k = \nabla^2 f(x_k) + E_k \tag{18}$$

The B_k matrix is then modified Hessian, which is used in step b the basic Newton algorithm. The term E_k is the Hessian correction element, which can be determined by any of the methods described in [13], as modified Cholesky factorization. An alternative unused second derivatives to calculate the matrix B_k, using gradient information calculated in previous steps, make up the so-called quasi-Newtonian methods. The concept emerges secant relationship given by:

$$\begin{aligned} & B_{k+1}s = y \\ \text{With :} \quad & y = \nabla f(x_{k+1}) - \nabla f(x_k) \\ & s = (x_{k+1} - x_k) \end{aligned} \tag{19}$$

And a simple way to update B_k is:

$$B_{k+1} = B_k + \frac{(y - B_k s)(y - B_k s)^T}{(y - B_k s)^T s} \tag{20}$$

This formula is developed postulating an update of rank 1, hence is referenced as SR1 (symetric Rank 1). An improvement over this, is updated using a range 2, which has been known as formula BFGS (Broyden–Fletcher–Goldfarb–Shanno) [18–21]:

$$B_{k+1} = B_k + \frac{yy^T}{s^T y} - \frac{B_k s s^T B_k}{s^T B_k s} \tag{21}$$

In addition, to address the problem of optimizing the internal combustion engine, should be restricted to use optimality conditions, and consider significant changes to the optimal search methods, since restrictions imposed modifications in realizing the gradient and Hessian. In what follows this is considered.

In the general problem 3, the vector x^*, must satisfy the relations denoted by $h(x)$ and $g(x)$, this can compress defining the feasible region as $\mathcal{F} = \{x | g(x) \leq 0, h(x) = 0\}$ such that the point x^* is a local minimum if $f(x^*) \leq f(x)$ for all $x \in \mathcal{N}(x^*) \cap \mathcal{F}$ with $\mathcal{N}(x^*) = \|x - x^*\| < \epsilon, \epsilon > 0$. The optimum point can no longer be given the minimum possible value of $f(x)$ unrestricted, the feasible region can not include that point, but still will be directed towards that point $\nabla f(x^*)$. For example with $x \in \mathbb{R}^2$, inequality constraints define a possible area on the surface of $f(x)$, and equality constraints, a line. The optimum point must satisfy the surface $f(x)$, the area set by the set $g(x) \leq 0$, and lines

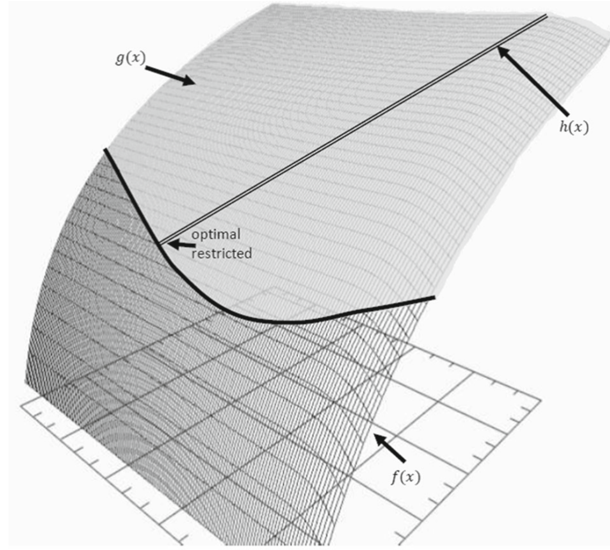


Fig. 2 Outlining restricted optimality

$h(x) = 0$ (See Fig. 2). Still, the narrow sweet spot, a point addressed by the curvature of $f(x)$.

The conditions that characterize the restricted optimal point is set in restricted named optimality conditions Karush–Kuhn–Tucker (KKT) [22], as follows:

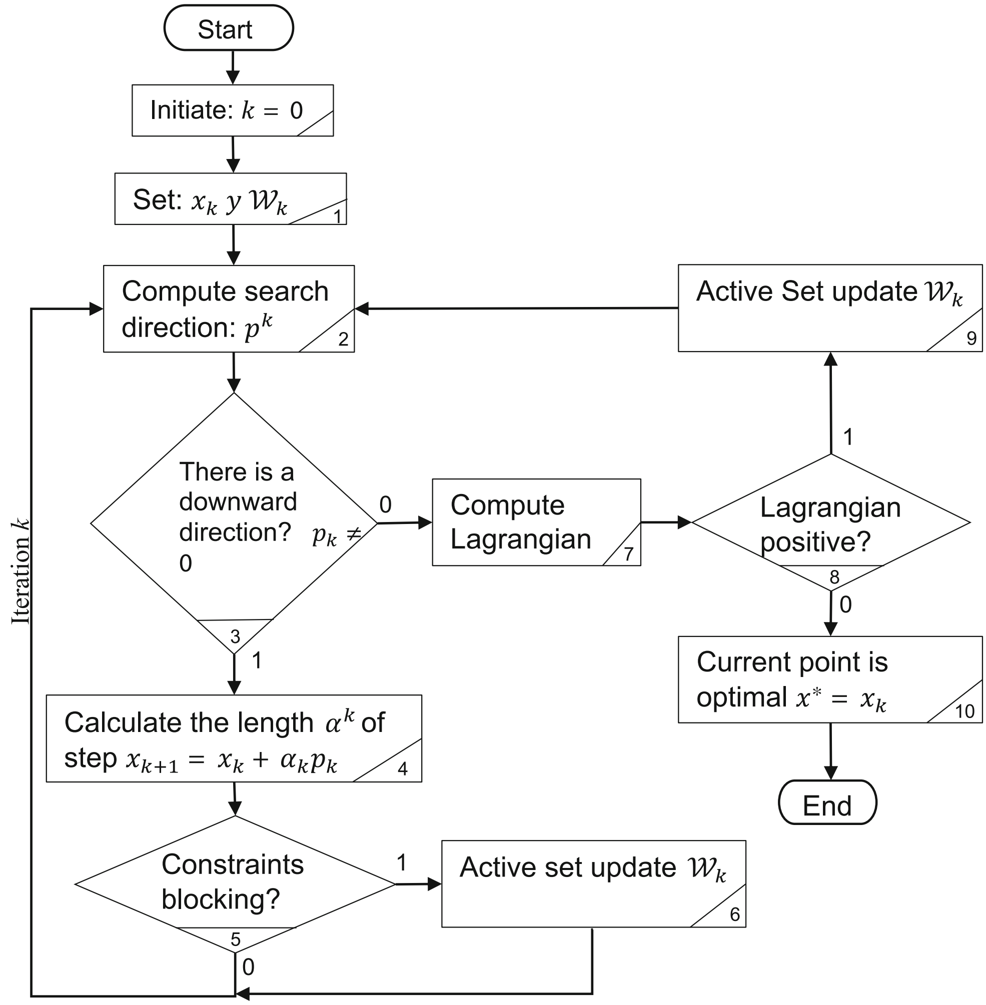
- Defining the function Lagrangian $L(x, u, v) = f(x) + g(x)^T u + h(x)^T v$ with u and v defined as weighting elements or multipliers of the inequality constraints and equity respectively, in solution, in the restricted optimal point is satisfied that:

$$\begin{aligned} \nabla_x L(x^*, u^*, v^*) = \nabla f(x^*) + \nabla g(x^*)^T u^* \\ + \nabla h(x^*)^T v^* = 0 \end{aligned} \tag{22}$$

- The restrictions are satisfied: $g(x^*) \leq 0$ and $h(x^*) = 0$.
- Inequality constraints must be strictly satisfied or active $g(x^*) = 0$, can also be inactive $g(x^*) < 0$, and in this case the restriction is irrelevant to the solution and the associated multiplier must be zero: $g(x^*)^T u^* = 0$ and $u^* \geq 0$. This is known as complementary slackness condition.
- Restrictions must be qualified so as to ensure that the gradients are sufficient to characterize the local optimum. A typical qualification among several, is that the gradients of the active constraints in $x^*$are linearly independent [23].
- The condition of positive curvature is necessary for all the feasible region. So the addresses of curvature p, different from zero determine the conditions necessary second order:

$$p^T \nabla^2 L(x^*, u^*, v^*) p \geq 0 \text{ Para todo } p \neq 0 \tag{23}$$

Fig. 3 Active set algorithm



3.2.2 Active set algorithm

The solution of the problem 11 linearizing the dynamic model around an operating point, can be approached by two different algorithms Newtonian type: methods of active set and interior point methods. The latter have been developed focusing on large problems, while the former are widely used and with an effectiveness verified in problems of medium and small scale [13], then suitable to the problem of optimization of the combustion engine.

The active set method under consideration is defined for the convex only case. In the case of the objective function quadratic programming problem, it is almost defined by explicitly defining the weighting matrix or gains (matrix B in 12), this must be positive definite or positive semidefinite. The method divides the constraints into a set of active constraints and a set of inactive for a current constraints x_k point. 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 $\mathcal{W}_k$ assembly work. Starting from an initial set $\mathcal{W}_0$ and a feasible initial value x_0, an iterative process continuously solves a QP sub-problem whose decision variable is a vector of p_k step towards the optimum and a set of equality constraints given by the active constraints in x_k. This sub-problem can be solved directly or by some method of factorization [24]. In solving p_k if a new search direction is not found, it is possible that the current point is optimal, then the conditions of restricted optimality are checked, the Lagrange multipliers associated with active constraints are obtained, and if all are this indicates positive x_k is optimal. If it is determined that the current point is not optimal then the set of $\mathcal{W}_k$ work is modified by removing the restriction with the most negative Lagrangian and forming a new sub-QP problem. If an address search is found downward, then you should calculate a vector length for this step and execute step updating the current position x_k, if another restriction is then activated to the active set of work $\mathcal{W}_k$ it is added. Figure 3 shows the flowchart of the algorithm, and follow it with some details.

When the process starts with $k = 0$ as iteration variable, in process 1 the search starting point is set and thus the first active working set. A good selection of that point implies convergence to the optimum, as mentioned in developing methods generally Newtonian. To address a correct selection of x_0, it has to rely on the arbitrariness of a random selection or find a systematic method according to the nature of the problem. It is usual in MPC strategies, problem solving using QP of previous sampling instant problem solving QP of the current moment. Specifically, since the solution space of the problem is determined by proximity to point of operation, as discussed earlier, the steady state obtained from a free response of the system, is a good starting point to search for the optimal.

The process 2 determines whether there is a downward direction p_k objective function for solving a sub-problem of equality constrained QP in which the restrictions correspond to the set $\mathcal{W}_k$ are treated as equalities and the rest are neglected. To do this based on the objective function of the problem 12 defining variable direction generally p as $p = x - x_k$, the linear term associated $g_k = Bx_k + b$, and by substituting $f(x)$ are:

$$\min_{x \in \mathbb{R}^n} f(x_k + p) = \frac{1}{2} p^T Bp - (g_k)^T p \quad (24)$$

In the above substitution, has been eliminated a term depending on x_k, independent of p, which may be omitted in the solution process [13] to the eliminate x_k variable. Then the sub-problem to solve for the search direction p, is

$$\begin{aligned} &\min_p f(p) = \frac{1}{2} p^T Bp - (g_k)^T p \\ &s.a. \\ &A_1 p = 0 \\ &a_i^T p = 0, i \in \mathcal{W}_k \end{aligned} \quad (25)$$

With A_2 in 12, denoted here as components a_i the subscript number i as restriction, such that the active set is $\mathcal{W}_k = \{i | a_i(x_k) = 0\}$. Formulation 25 is a quadratic problem with equality constraints, which in general can be established written as:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} x^T Bx - x^T b \\ &s.a. \\ &Ax = c \end{aligned} \quad (26)$$

Then, to solve 26 the KKT optimality conditions apply and the resulting linear system is solved. The conditions of the first order states that for x^* be the solution, there is a Lagrange multiplier vector λ^* satisfies the following linear system:

$$\begin{bmatrix} B & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -b \\ c \end{bmatrix} \quad (27)$$

For reasons of computational solution 27 is usually done in a change of variables so [13]:

$$x^* = x + p \rightarrow p = x^* - x \quad h = Ax - c \quad g = b + Bx \quad (28)$$

$$\begin{bmatrix} B & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda^* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix} \quad (29)$$

To solve the system 29 there are several methods, from direct solution of the system, and other LU factorization and iterative methods. Is selected for the problem, the method of null space, because suitable for reported problems with few degrees of freedom. Reproduced here, the development of Nocedal [13] of the method.

Assuming that A has full row rank, a subset of m columns of A is linearly independent such that it can build the submatrix $H \in \mathbb{R}^{mxm}$, a submatrix N with the remaining columns, and a permutation matrix P, to position H in the first columns, such that $AP = [H|N]$. The corresponding vector of variables as x_H part and x_N whereby $\begin{bmatrix} x_H \\ x_N \end{bmatrix} = P^T x$ and restrictions equation is rewritten as:

$$\begin{aligned} &c = Ax = AP(P^T x) = Hx_H + Nx_N \\ &x_H = H^{-1}c - H^{-1}NX_N \end{aligned}$$

In addition, any feasible point can be written as:

$$\begin{aligned} &\begin{bmatrix} x_H \\ x_N \end{bmatrix} = x = Yc + Zx_N \\ &Y = \begin{bmatrix} H^{-1} \\ 0 \end{bmatrix}, Z = \begin{bmatrix} -H^{-1}N \\ I \end{bmatrix} \end{aligned}$$

Z is such that $AZ = 0$, this constitutes to Z as a basis for the null space of A. Now returning to 29, p is split into two components $p = Yp_Y + Zp_Z$, with Z again as the null space of A with dimensions $n \times (n - m)$, and Y is formed, such that $[Y|Z]$ is nonsingular. Yp_Y, then a particular solution of $Ax = c$ system, and Zp_Z is a displacement along restrictions. Reformulation replacing the previous p in $A(-p) = h$ (second equation 29), resulting in:

$$\begin{aligned} &A[-(Yp_Y + Zp_Z)] = h \\ &AYp_Y + AZp_Z = -h, AZ = 0 \\ &(AY)p_Y = -h \end{aligned}$$

And doing such, in the first equation of 29, we have:

$$
\begin{aligned}
&-BYp_Y - BZp_Z + A^T\lambda^* = g\\
&(-BYp_Y - BZp_Z + A^T\lambda^* = g)*Z^T\\
&(Z^TBZ)p_Z = Z^TBYp_Y - Z^Tg
\end{aligned}
$$

From the above expressions, you can get P_Y, P_Z and components. To P_Z, the term Z^TBZ, called the reduced Hessian must be positive definite, and can be applied Cholesky factorization for the solution. In general this is true if B is strictly positive definite. Once the solution of the p direction, we obtain the Lagrangian vector:

$$
\begin{aligned}
&(B(-p) + A^T\lambda^* = g)*Y^T\\
&(A^T\lambda^* = g + Bp)*Y^T\\
&A^TY^T\lambda^* = Y^T(g + Bp)
\end{aligned}
$$

Once obtained the current search address p_k, there are two possibilities on the route algorithm, shown in decision block 3: if $p_k \neq 0$ means that there is a downward direction which further reduces the objective function of the original problem, if $p_k = 0$, there is a possibility that the current point is a local optimum, which for a convex QP problem is proved to be a global minimum [6].

In the process 4 after determining the search direction is necessary to determine both the length of α step forward in that direction, that is, such that the new point is $x_{k+1} = x_k + \alpha_k p_k$. This is necessary because in such movement should be calculated maximum path length that does not involve a violation of any restrictions that may affect their active or inactive carácter, so the upper limit of this length is the distance to the nearest restriction on p_k address. The maximum length is 1, the next point to take the minimum of the objective function within a space equal to the null space of A, or step may be to the nearest constraint is less than unity. To check if a length of step 1 violate the restrictions, then the evaluation of $a_i^T p_k$ product is negative, otherwise step is feasible in the active set. In case you find a limitation in step due to a restriction, then the distance to the restriction is such that:

$$
\begin{aligned}
&a_i^T(x_k + \alpha_{ik}p_k) = c_i\\
&\alpha_{ik} = \frac{c_i - a_i^Tx_k}{a_i^Tp_k}
\end{aligned}
$$

Various restrictions that limit the advance may exist, or may not be some, then the length of α_k step taken, it will correspond to the lowest, this is set as follows:

$$
\alpha_k = \left\{ \begin{array}{l} min\{\alpha_{ik}\}\, sia_i^Tx_k < 0, \quad i \notin \mathcal{W}_k \\ 1siia_i^Tx_k \geq 0, \quad i \notin \mathcal{W}_k \end{array} \right\}
$$

In block 5, the decision to update the set of $\mathcal{W}_k$ work is taken. If no blocking restrictions, then the nearest restriction, which was defined for α_k, becomes part of the working set $\mathcal{W}_k$, otherwise the current working set $\mathcal{W}_{k+1} = \mathcal{W}_k$ remains. It continues to minimize $f(x)$, in the new space or the current space as appropriate.

The process 7 runs when found a working set and such a point that has a downward direction $p_k = 0$. Then optimality conditions for problems with inequality constraints in the active set, we have:

$$
\begin{aligned}
&Bx^* + b - \sum_{i\in\mathcal{W}^*} a_i\lambda_i^* = 0\\
&a_i^Tx^* = c_i, \quad i \in \mathcal{W}^*\\
&a_i^Tx^* \geq c_i, \quad i \in I\backslash\mathcal{W}^*\\
&\lambda_i^* \geq 0, \quad i \in I \cap \mathcal{W}^*
\end{aligned}
$$

From 12 and $p = 0$, in process 7 we have:

$$
A^T\lambda^* = \sum_{i\in\mathcal{W}_k} a_i\lambda_i = Bx_k + b \tag{30}
$$

Since all the points x_k are always feasible on all restrictions, restrictions remain equal and inequality that are not in the active set also. From 30, the Lagrange multipliers corresponding to the active constraints are calculated, and zero for the corresponding multiplier is done not active. From this criterion it is determined whether the current point is optimal according to the optimality conditions, if all Lagrangian are not negative, then the current point is a local optimum for the original problem $x_k = x^*$. If one or more of the multiplier is negative, the latter condition is not maintained and the objective function may decrease by changing the working set. Usual then remove the $\mathcal{W}_k$, corresponding to more negative multiplier restriction, because it demonstrates that the rate of decrease when a restriction is removed is proportional to its Lagrange multiplier [13], and again is computed by a new p_k.

3.2.3 Linearization

The method of active set exhibited in the previous section then solves a QP problem involving the set of constraints are linear functions of the decision variables. To solve the problem of optimization of the operation of the combustion engine with said algorithm, the linearization of the model for the current operating point is posed and establish a QP problem at each operating point.

The process of linearization model of a nonlinear dynamical system [10,25], arises from the use of Taylor series to approximate a nonlinear function. So, if $f(x_1, x_2, x_{3,} \ldots x_n)$ is a nonlinear function of n variables state, an approximation of f, to an equilibrium point x_0 it is:

$$f(x_0 + \delta x) = f(x_0) + \left.\frac{df}{dx}\right|_{x_0} \delta x + \frac{1}{2!}\left.\frac{d^2 f}{dx^2}\right|_{x_0} dx^2 + \cdots .$$

Now considering the overall dynamic system described by:

$$\begin{aligned}&\dot{x}(t) = f(x(t), u(t)), x(t_0) = x_0\\&y(t) = h(x(t))\end{aligned}$$

Integrating the model has:

$$\begin{aligned}&x(t) = x_0 + \int_{t_0}^{t} f(x(t), u(t))dt\\&y(t) = h\left[x_0 + \int_{t_0}^{t} f(x(t), u(t))dt\right]\end{aligned}$$

Defining an operating point in the form of $x(t_0) = X, u(t_0) = U, y(t_0) = h(X) = Y$ and setting that is a point of steady state operation, then$X$, U and Y are constant. Faced with disturbances δx and δu, $x(t_0) = X + \delta x_0$ and $u(t) = U + \delta u(t)$. These disturbances affect the model as follows:

$$\begin{aligned}&x(t) = X + \delta x_0 \int_{t_0}^{t} f(X + \delta x_0(t), U + \delta u(t))dt\\&y(t) = h(X + \delta x_0(t))\end{aligned}$$

The disturbance model is obtained with $\delta x(t) = x(t) - X$ and $\delta y(t) = y(t) - Y$:

$$\begin{aligned}&\delta x(t) = \delta x_0 + \int_{t_0}^{t} f(X + \delta x_0(t), U + \delta u(t))dt\\&\delta y(t) = h(X + \delta x_0(t)) - h(X)\end{aligned}$$

Applying a Taylor series approximation for the operating point X, U is obtained:

$$\begin{aligned}f(X + \delta x(t), U + \delta u(t)) &= f(X, U) + \left.\frac{\partial f}{\partial x}\right|_{X,U} \delta x(t)\\&\quad + \left.\frac{\partial f}{\partial u}\right|_{X,U} \delta u(t)\\h(X + \delta x_0(t)) &= h(X) + \left.\frac{\partial h}{\partial x}\right|_{X} \delta x(t)\end{aligned}$$

Replacing on the model of the disturbance:

$$\begin{aligned}&\delta x(t) = \delta x_0 + \int_{t_0}^{t} \left[\left.\frac{\partial f}{\partial x}\right|_{X,U} \delta x(t) + \left.\frac{\partial f}{\partial u}\right|_{X,U} \delta u(t)\right] dt\\&\delta y(t) = \left.\frac{\partial h}{\partial x}\right|_{X} \delta x(t)\end{aligned}$$

Terms are omitted upper and equal to 2 order, because only interested in the event the linear terms. Renaming $A = \frac{\partial f}{\partial x}|_{X,U}$, $B = \frac{\partial f}{\partial u}|_{X,U}$ and $C = \frac{\partial h}{\partial x}|_{X}$ then we can rewrite the previous system, thus:

$$\begin{aligned}&\delta x(t) = \delta x_0 + \int_{t_0}^{t} [A\delta x(t) + B\delta u(t)] dt\\&\delta y(t) = C\delta x(t)\end{aligned}$$

Differentiating the perturbed state:

$$\begin{aligned}&\delta \dot{x}(t) = A\delta x(t) + B\delta u(t)dt, x(t_0) = \delta x_0\\&\delta y(t) = C\delta x(t)\end{aligned} \tag{31}$$

Then the operating point determining and evaluating A, B and C, called the Jacobian matrices, at said point of operation, we find an equivalent linear system 31 which is shown in state variables. From here then the relevant restrictions are replaced in the formulation of the optimization problem.

3.2.4 Sequential quadratic programming

The other scheme assessed in the problem of optimal control of combustion engine, is part of a set of widely applicable methods [9,26,27] to solve problems NLP. The basic method is commonly referenced with the SQP (Sequential Quadratic Program) acronym, and resembles the scheme outlined above, which is an iterative method in each iteration solves a QP sub-problem defined by a linearization of the restrictions. The minimum of the problem is used to construct the QP of the next iteration whose solution is a better approximation to the optimal, such that the $x_k, k = 0, 1, 2, \ldots$ c sequence converge to the optimum. Each QP sub-problem is constructed with a quadratic approximation of the objective function and restrictions on x_k local affine approximations. Some SQP schemes, as set forth herein, are supported on Newton's method, and apply the conditions and theorems developed for it, such that the objective function and constraints should be soft (three times differentiable), also can be considered techniques solution as the active set.

To continue and outline the SQP algorithm, we return to the general formulation for NLP:

$$\begin{aligned}&\min_{x\in\mathbb{R}^n} f(x)\\s.a.\quad &h(x) = 0\\&g(x) \le 0\end{aligned} \tag{32}$$

The goal is to formulate a QP problem, approximately 32 in each iteration given x_k. Direct option is to apply an approximation using Taylor series and taking up the quadratic term for $f(x)$ and up to the linear term $h(x)$ and $g(x)$:

$$\begin{aligned} f(x) &\approx f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k) \\ h(x) &\approx h(x_k) + \nabla h(x_k)(x - x_k) \\ g(x) &\approx g(x_k) + \nabla g(x_k)(x - x_k) \end{aligned} \quad (33)$$

However, the shape of $f(x)$ function in Eq. 33, the objective function does not take into account nonlinearity of constraints, and is shown in Boggs and Tolle [28] that the quadratic program algorithm failure to a simple problem. On the other hand, applying the KKT optimality conditions restricted to the problem 34, we note that the solution of 32 is also a local minimum of problem 34, and then there is an equivalence of both problems in their solution. Also, using the Lagrangian objective function involves restrictions within the problem, then valid as presented at 34, to develop the SQP algorithm.

$$\begin{aligned} &\min_x L(x, u^*, v^*) \\ s.a. \quad & h(x) = 0 \\ & g(x) \le 0 \end{aligned} \quad (34)$$

Recalling the definition of the Lagrangian $L(x, u, v) = f(x)+h(x)^T v+g(x)^T u$, again seeks a quadratic approximation of the objective function and linear approximation of the restrictions, the quadratic approximation of the Lagrangian Taylor in a given iteration x_k is:

$$\begin{aligned} &L(x_k, u_k, v_k) + \nabla L(x_k, u_k, v_k)^T p_x + \frac{1}{2} p_x^T \nabla^2 L(x_k, u_k, v_k) p_x \\ &p_x = (x - x_k) \end{aligned}$$

By taking up the quadratic term and neglecting the constant term, rebuild 33 with a new objective function equivalent:

$$\begin{aligned} &\min_{p_x} \nabla L(x_k, u_k, v_k)^T p_x + \frac{1}{2} p_x^T \nabla^2 L(x_k, u_k, v_k) p_x \\ &s.a \\ &h(x_k) + \nabla h(x_k)^T p_x = 0 \\ &g(x_k) + \nabla g(x_k)^T p_x \le 0 \end{aligned} \quad (35)$$

The p_x term of 35 QP subproblem solution can be found with the method of the active set, and is used to formulate the following QP subproblem. Explicitly, the p_x is the direction vector to generate the new x_{k+1} step. The estimation of the direction for the Lagrange multipliers using the optimal values of the current solution results in p_u and p_v such that $p_u = u^* - u_k$ and $p_v = v^* - v_k$. The evaluation of the convergence parameters, determines the end QP subproblem and therefore the relationship between p_x solution, and the solution of 32. Nevertheless, considerations regarding the conditioning of each subproblem QP and the convergence of the algorithm in general should be highlighted as any problems QP.

Each QP subproblem formed, must have a solution, then you must ensure that the system is consistent linearized constraints and the objective function is bounded on the feasible set; It must also drive the convergence of the sequence of solutions. These considerations can be addressed by assuming a local SQP, ie the condition of consistency can be guaranteed when x_k is in a neighborhood of x^*, and the convergence of the method of Newton is ensured when the initial iteration x_0 is sufficiently close the solution x^* [28]. To maintain the condition number of the Hessian of bounded Langrangiano a B_k approximation is used, as it outlined in paragraph substantiation, whether or SR1 using the BFGS method and an initial approximation to the Hessian B_0 optimal Lagrangian. Using an approximation, the QP subproblem 35 is reformulated as:

$$\begin{aligned} &\min_{p_x} \nabla L(x_k, u_k, v_k)^T p_x + \frac{1}{2} p_x^T \mathrm{B}_k p_x \\ &s.a \\ &h(x_k) + \nabla h(x_k)^T p_x = 0 \\ &g(x_k) + \nabla g(x_k)^T p_x \le 0 \end{aligned} \quad (36)$$

It is common to find in the literature the linear term $\nabla f(x_k) p_x$ instead of the Lagrangian gradient [13] and assume equivalence between two formulations if you have a problem restricted equality. However in the original NLP problem, they can rewrite the inequality constraints as adding slack variables equal and limiting these variables, so replacement is validated mentioned.

The other aspect to address is the definition of the correct initial conditions to ensure optimum proximity to and then maintaining the convergence properties of the same. The SQP method has properties of linear, quadratic and higher convergence with the B_k approximations properly. To do this, set to Boggs and Tolle [28] the following conditions, assuming the optimal Lagrangian Hessian is not unique: the B_k matrices are uniformly positive definite in the null space of $\nabla h(x_k)$; $\{B_k\}$ sequence is uniformly bounded; and each B_k has uniformly bounded inverse. On the other hand, considering that in proximity of the optimal QP sub-problem in x_k has the same set of active constraints in the NLP x^*, and are known, then the inactive constraints can be ignored and assume that the sub-problem QP is a problem of restricted equity, with associated multipliers, eliminated from the Lagrangian. The application of the optimality conditions of first order this leads to the following system:

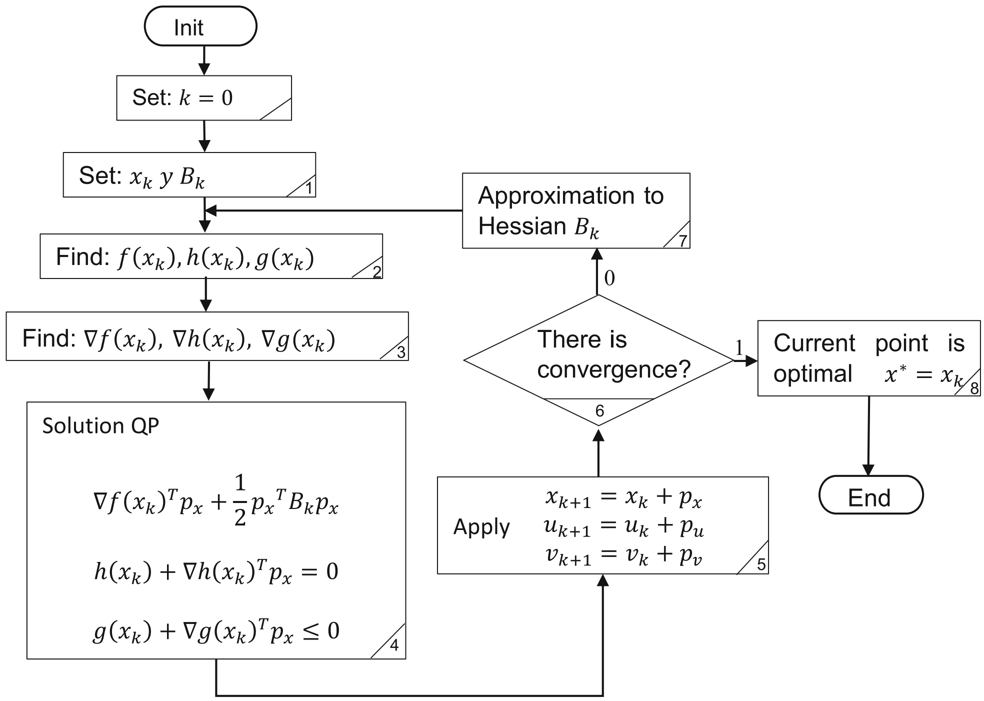


Fig. 4 SQP algorithm

$$
\begin{aligned}
B_k p_x + \nabla h(x_k) v^* &= -\nabla_x f(x_k) \\
\nabla h(x_k) p_x &= -h(x_k) \\
v^* = v_k + p_v &= v_{k+1}
\end{aligned}
$$

Which it is identical to the system obtained after applying a step of Newton unitary system of equations obtained from the first order conditions of the problem NLP:

$$
\begin{bmatrix} \nabla_x^2 L(x_k, v_k) & \nabla h(x_k) \\ \nabla h(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_v \end{bmatrix} = \begin{bmatrix} \nabla_x L(x_k, v_k) \\ h(x_k) \end{bmatrix}
$$

Consequently iterations for vector variables and equalities multipliers are the same. Under these considerations, it can be set upon a correct initial parameter estimation, a sequence of solutions formed by unit steps in directions whose solution addresses the problem of restricted equality, if $||x_0 - x^*||$ is sufficiently small, the sequence of iterations is well defined and quadratically converges to the optimum, ie there is a positive constant ξ such that:

$$
\|x_{k+1} - x^*\| \leq \xi \|x_k - x^*\|
$$

However, finding a starting point is not a trivial task, and hence the local SQP method can lose practical interest. To globalizar the method, the step length to the next sub-QP problem is modified as the algorithm in the active set by a parameter α which is determined in each iteration using merit functions. The treatment is far from so simple, and despite the difficulty in finding correct starting points sufficiently close to the solution, the specific study of the problem can allow to find them find proper strategies. In this work we have built experimental functions to define appropriate near points of operation of the combustion engine, which in the next section will specify initial solutions. In Fig. 4 SQP algorithm based on the reports presented Boggs and Tolle.

In process 1, an estimate of the initial solution is defined as just mentioned, experimental functions or using a previous result of the algorithm and further assessing the Hessian approximation to set the initial estimate. Steps 2 and 3 are simply evaluations of the original objective function and constraints in the current step x_k, and calculating gradients for the Lagrangian approximations and restrictions used to formulate the QP sub-problem. Step 4 solves the problem formulated in a QP solver, in this case, an active set algorithm. In step 5, the solution of the problem and the result of optimal Lagrange multipliers are used to define the next iteration x_{k+1}, with the new estimates for multipliers. In step 6, the convergence is analyzed to determine if it has reached the optimum, as in the basic formulation of Newton, to stop the algorithm parameters are used as standard Euclidean step in k such that $\|p_x\| \leq \epsilon_1$ and the gradient of the Lagrangian $\|\nabla L(x_k, u_k, v_k)\| \leq \epsilon_2$. The values are based $\epsilon_{1,2}$ numerical experimentation and possibly the accuracy required in the solution. By accepting the current solution it will then x^* NLP. Otherwise it is then calculated approximation to the Hessian of the Lagrangian, denoted by B_k, using the BFGS formula, and as an alternative the SR1, which were mentioned in the general development of Newton method.

4 Implementations, results and analysis

Both algorithms “Active Set” and “Sequential Quadratic Programing SQP” have been evaluated and executed in embedded processing systems for real-time solutions, specifically used the NI cRIO-9075 National Instruments running VxWorks operating system with real-time processor of 400 Mhz and integrating a 2M gate array Spartan-6 LX25 [29]. These elements are an alternative available on the market that is well suited as a replacement to the electronic control unit ECU in a motor factor y [30,31]. Functions as “Quadratic Programming.vi”, “Constrained Nonlinear Optimization.vi” and “SIM Linearized.vi” is a unit of software encoded in LabVIEW and National Instruments proprietary tests used for solving the optimization problem engine operation and other programming tools and solvers of differential equations.

4.1 Linearization and active set

As described, a possibility that is evaluated in this report to address the problem of optimal control of motor operation is the linearization of the model at the point of actual operation and use of the algorithm “Active Set” to solve the problem quadratic programming made with the proposed objective function and linear equations of the model. They are to be considered especially those times used to linearize the model equations and the time taken by the algorithm to find the optimal in order to determine the feasibility of their use in the times required by engine operation.

To solve a problem linearized formulation through the active set has been used “Quadratic Programming.vi” function in your choice “Active Set”. This function takes as input arguments the Q matrix and weight matrix for the decision variables x, and c vector for the linear term [32] such that the objective function minimizes $\frac{1}{2}x^TQx + cx$, similar to the quadratic changing the literal formulation 12. Also the matrix A and vector b to establish the equality constraints $Ax = b$, and elements for the inequality constraints D, $lmin$ and $lmax$ such that $lmin \leq Dx \leq lmax$ and other termination criteria, starting point, and output arguments as minimum, minimum value of the objective and multipliers active LaGrange function. Additionally address the problem of dynamic optimization involves rethinking the problem of infinite dimension in a finite-dimensional problem by rewriting the problem for some points in time. To do this based on the general optimization problem with a quadratic function and a linearized or linear dynamic system:

$$\begin{aligned} \min J &= x_d(t_f)^T \boldsymbol{F} x_d(t_f) + \int_{t_0}^{t_f} \left[x_d(t)^T \boldsymbol{Q} x_d(t) + u^T(t) \boldsymbol{R} u(t) \right] dt \\ s.a. \quad & x(t_0) = x_m \\ & \dot{x}(t) = Ax(t) + Bu(t) \\ & y(t) = Cx(t) \\ & \underline{x} \leq x(t) \leq \bar{x} \\ & \underline{u} \leq u(t) \leq \bar{u} \\ & \underline{y} \leq y(t) \leq \bar{y} \end{aligned} \tag{37}$$

As In 8, $x_d(t_f)$ it is defined as $x_d(t_f) = Ref_x - x(t_f)$ and $x_d(t) = Ref_x(t) - x(t)$, x_m denotes initial conditions linear model described by the matrices A, B, C. $\bar{x}$ and $\underline{x}$ denote the bounds for the state variables and literals in such a way to bar the input variables and output. Time $[t_0 - t_f]$ is subdivided into N_p ranges, implying a loss of degrees of freedom, but greatly simplifies the solution by direct methods as the active set. This loss does not affect the performance of control systems in practice. The subdivision transforms the problem into an optimization problem of discrete time, with a discrete prediction horizon $Tp = \{k_0, \ldots, k_0 + N_p - 1\}$ whose task is to find a sequence of entries as a piecewise constant function. Rewriting the problem 37 in discrete form are:

$$\begin{aligned} \min J &= x_k^T \boldsymbol{F} x_k + \sum_{k=k_0}^{k_0+N_p-1} x_k^T \boldsymbol{Q} x_k + u_k^T \boldsymbol{R} u_k \\ s.a. \quad & x_{k_0} = x_m \\ & \left. \begin{array}{l} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \\ \underline{x} \leq x_k \leq \bar{x} \\ \underline{u} \leq u_k \leq \bar{u} \\ \underline{y} \leq y_k \leq \bar{y} \end{array} \right\} \forall k \in T_p \end{aligned} \tag{38}$$

For this formulation the matrices A, B and C are the discrete equivalent of the original, depending on the sampling period chosen for the discretization problem, the discretization process is well documented in [33]. To this report they are assuming sampling periods small sufficiently for an reliable representation of the continuous dynamic process, however the numerical values of these matrices are irrelevant at this time because the current interest is to evaluate the ability of solvers for solve the problem in a sampling period. In addition to the discretization it is necessary to reformulate the dynamic optimization problem specifically a QP problem as in 12, so it can be solved by the algorithm. For this, two formulations have sufficient maturity reported in the literature: the formulation is not condensed and condensed formulation. Proper formulation has a large impact on the size and structure of the problem; processing requirements and memory; and bad numerical conditioning [34].

The condensed formulation makes use of the dynamics of the plant to eliminate the states of the decision variables, expressing them as an explicit function of measuring the current status and estimated inputs control, this leads to a compact and dense problem. To develop this formulation, the problem is rewritten QP, according to the above, changing the decision variables generalized by the input variables or as decision variables of the problem QP. Besides eliminating the states involves removing explicit restrictions equality, then we have:

$$\begin{aligned} &\min J = \frac{1}{2}u^T Qu - u^T f(x_m) \\ &s.a. \\ &\underline{g(x_m)} \leq Du \leq \overline{g(x_m)} \end{aligned} \tag{39}$$

The formulation is based on future states of the measurement process instantly k_0 can be expressed in terms of the initial state and the sequence of entries $u_{k_0}, \ldots, u_{k_0+N_p-1}$[35], as follows:

$$\begin{aligned} x_{k_0+1} &= Ax_{k_0} + Bu_{k_0} \\ x_{k_0+2} &= A(Ax_{k_0} + Bu_{k_0}) + Bu_{k_0+1} \\ x_{k_0+3} &= A(A(Ax_{k_0} + Bu_{k_0}) + Bu_{k_0+1}) + Bu_{k_0+2} \\ x_{k_0+4} &= A(A(A(Ax_{k_0} + Bu_{k_0}) + Bu_{k_0+1}) + Bu_{k_0+2}) + Bu_{k_0+3} \\ &\vdots \end{aligned}$$

All state equations to $x_{k_0+N_p}$ can be compacted in matrix form and obtain an increased matrix system with the following sets:

$$\boldsymbol{x} = \begin{bmatrix} x_{k_0} \\ x_{k_0+1} \\ \vdots \\ x_{k_0+N_p} \end{bmatrix} \quad \boldsymbol{u} = \begin{bmatrix} u_{k_0} \\ u\ k_0+1 \\ \vdots \\ u_{k_0+N_p-1} \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} \mathbb{I} \\ A \\ A^2 \\ \vdots \\ A^{N_p-1} \\ A^{N_p} \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 0 & & & \\ B & & & \\ AB & B & & \\ \vdots & \vdots & \vdots & \\ A^{N_p-2}B & \cdots & AB & B \\ A^{N_p-1}B & A^{N_p-1}B & \ldots & AB\ B \end{bmatrix}$$

According to the augmented system, weight matrices of the objective function are also increased, as the horizon N_p:

$$\boldsymbol{Q} = \begin{bmatrix} Q & & & & \\ & Q & & & \\ & & \ddots & & \\ & & & Q & \\ & & & & F \end{bmatrix} \quad \boldsymbol{R} = \begin{bmatrix} R & & & \\ & R & & \\ & & \ddots & \\ & & & R \end{bmatrix}$$

In the case of this report matrix C is the identity matrix indicating that the state variables are the outputs of the system directly. On the other hand, the inequality constraints can be increased by writing the restrictions of each variable in matrix form: $\underline{x} \leq Mx \leq \bar{x}$ and $\underline{u} \leq Nu \leq \bar{u}$. M and N are matrices containing the coefficients in the linear inequality equations, and can be combined, increasing them, so:

$$M = \begin{bmatrix} M \\ \mathbb{O} \end{bmatrix} N = \begin{bmatrix} \mathbb{O} \\ N \end{bmatrix} \Rightarrow \underline{l} \leq Mx + Nu \leq \bar{l}$$

l denotes vectors levels of state variables and inputs conveniently attached $\underline{l} = [\underline{x}\ \underline{u}], \bar{l} = [\bar{x} \quad \bar{u}]$. Now with increased state vector the matrices of inequality constraints are increased again N_p for:

$$\boldsymbol{M} = \begin{bmatrix} M & & & \mathbb{O} \\ & M & & \\ & & \ddots & \\ & & M & \mathbb{O} \end{bmatrix} \boldsymbol{N} = \begin{bmatrix} N & & & \\ & N & & \\ & & \ddots & \\ & & & N \end{bmatrix} \underline{\boldsymbol{l}} = \begin{bmatrix} \underline{l} \\ \underline{l} \\ \underline{l} \\ \underline{l} \end{bmatrix} \quad \bar{\boldsymbol{l}} = \begin{bmatrix} \bar{l} \\ \bar{l} \\ \bar{l} \\ \bar{l} \end{bmatrix}$$

The problem 38 is written in augmented form:

$$\begin{aligned} &\min J = \frac{1}{2}x^T Qx + u^T Ru \\ &s.a. \\ &x_{k_0} = x_m \\ &x = Ax_{k_0} + Bu \\ &\underline{l} \leq Mx + Nu \leq \bar{l} \end{aligned} \tag{40}$$

Then remove the states of this formulation replaces the initial state and the equation comprising all states in the objective function and constraints equation:

$$\begin{aligned} &\min J = \frac{1}{2}u^T(B^T QB + R)u + u^T(B^T QA)x_m \\ &s.a. \\ &\underline{l} - MAx_m \leq (MB + N)u \leq \bar{l} - MAx_m \end{aligned} \tag{41}$$

As shown, 41 is the proposed form 39 with the linear term as a function of the measured x_m state as well as the limits of the decision variables, which are only inputs to the system, and renaming weight matrices and coefficients, with the relationships shown in augmented matrices. In the objective function is constant terms have been eliminated, and restrictions have been organized the boundaries after the algebraic manipulations, the continuity with the convexity of the problem remains [35]. The formulation does not condensed or spread 42, maintains the states of the plant as decision variables along with the inputs, and considers the dynamics of the system implicitly imposing it as equality constraints, this leads to a larger problem but banded matrices, which solver can take advantage [36,37]:

$$\begin{aligned} &\min_z J = \frac{1}{2} z^T Q z \\ s.a.\quad &Hz = f(x_m) \\ &\underline{l} \le Dz \le \bar{l} \end{aligned} \tag{42}$$

Note that the decision variables are the vector $z = [x\,u]$, formed by the state variables and inputs, the linear term does not appear in the formulation, and the equality constraints depend on the measured state, however levels remain. Such a formulation condensed solution to the horizon dynamics N_p, can be written in matrix form:

$$\begin{bmatrix} I & & & & & & \\ -A & -B & I & & & & \\ & & -A & -B & & & \\ & & & & \ddots & & \\ & & & & & I & \\ & & & & -A & -B & I \end{bmatrix} \begin{bmatrix} x_{k_0} \\ u_{k_0} \\ x_{k_1} \\ u_{k_1} \\ \vdots \\ x_{k_{N_p-1}} \\ u_{k_{N_p-1}} \\ x_{k_{N_p}} \end{bmatrix} = \begin{bmatrix} x_m \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

The above equation is the equality constraints of this coefficient matrix H is defined and the decision variable vector z is obtained. The objective function is written uniting pesos Q and R matrices and increasing this union in N_p. The weight of the terminal cost, the augmented matrix is added:

$$\min J = \frac{1}{2} \begin{bmatrix} x_{k_0} \\ u_{k_0} \\ x_{k_1} \\ u_{k_1} \\ \vdots \\ x_{k_{N_p-1}} \\ u_{k_{N_p-1}} \\ x_{k_{N_p}} \end{bmatrix}^T \begin{bmatrix} Q & & & & & & & \\ & R & & & & & & \\ & & Q & & & & & \\ & & & R & & & & \\ & & & & \ddots & & & \\ & & & & & Q & & \\ & & & & & & R & \\ & & & & & & & P \end{bmatrix} \begin{bmatrix} x_{k_0} \\ u_{k_0} \\ x_{k_1} \\ u_{k_1} \\ \vdots \\ x_{k_{N_p-1}} \\ u_{k_{N_p-1}} \\ x_{k_{N_p}} \end{bmatrix}$$

The inequality constraints are kept in the combined form $\underline{l} \le Mx + Nu \le \bar{l}$, making $= [M\,N]$, $\underline{l} = [\underline{x}\;\underline{u}]$ and $\bar{l} = [\bar{x}\;\bar{u}]$, increased N_p times:

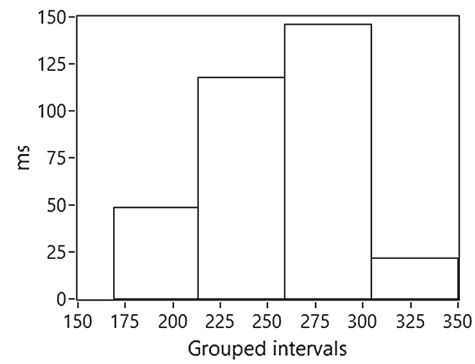


Fig. 5 Processing time for task linearization

$$\begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{x} \\ \underline{u} \\ \vdots \\ \underline{x} \\ \underline{u} \end{bmatrix} \le \begin{bmatrix} D & & & \\ & D & & \\ & & \ddots & \\ & & & D \end{bmatrix} \le \begin{bmatrix} \bar{x} \\ \bar{u} \\ \bar{x} \\ \bar{u} \\ \vdots \\ \bar{x} \\ \bar{u} \end{bmatrix}$$

Both formulations have been evaluated with the active set algorithm for solving an optimization problem, noting the number of iterations, time spent processing and analysis of the solution at different points of operation to determine usability in an MPC controller implemented in embedded hardware, and find areas for optimum initial values.

Figure 5 shows the first element of analysis: the processing time for linearization used in each operating point. This graph shows a histogram processing times used 350 different points of operation, the variability is due, the algorithm should seek stable operating point for the given inputs and simulation model used for this. Some operating points are more stable than others and therefore are found more quickly. However, one can predict with 80 % certainty that the time linearization will be in a range of 225–300 ms. This process is more demanding and blocking, especially with horizons of short prediction when the sampling period is very small compared with the response time of the plant, then the sequential execution linearization—formulation—solver QP is not practicable, and it is convenient to parallelize the process of linearization, beginning execution from monitoring mechanisms of the operation of the engine.

Tables 1 and 2 show the processing times used by the embedded to solve the problem with different prediction horizons system. Have been applied bounds on the variables of speed, throttle opening and injection flow. Experimenting with a stabilization problem, ie references to the outputs are zero, this does not lead to changes in the formulation and solution of the problem [35]. Weighing only the

Table 1 Experimenting with scattered formulation

Np	Time CPU (ms)	Iterations	Variables	Equality constraints
0	1	1	7	7
1	2	3	14	14
2	9	5	21	21
3	19	8	28	28
4	39	6	35	35
5	180	50	42	42
6	4302	1405	49	49

Table 2 Experimentation with condensed formulation

Np	Time CPU (ms)	Iterations	Variables	Inequality constraints
1	3	2	2	4
2	8	5	4	10
3	14	4	6	16
4	24	4	8	20
5	91	5	10	24
6	165	8	12	30
7	328	8	14	34
8	342	10	16	40
9	154	8	18	44
10	192	3	20	50
15	384	3	30	88

speed and the injected flow for the decision variables for both Boltza as part of Mayer in the objective function. As will be shown below, not all operating points, have the same difficulty of solution, so it is necessary to clarify that the operation speed for data tables is $n = 0.8345$ and $n = 2.2540$ respectively. The structure formed in the condensed formulation $N_p = 0$ is not solvable, so that's why the results are presented from $N_p = 1$ in Table 2. The tables show the possible experiments within a range of less than 200 ms, ie solutions within this range. The spread formulation shows that it is possible to solve the optimization problem of the engine, even for a prediction horizon equal to 5 sampling periods, to a higher horizon the time taken for the CPU to solve the problem is more than 4 s.

Compared with condensed formulation required less than 200 ms for problems with prediction horizons up to 10 sampling periods, however due to the non-deterministic characteristic of the method, some experimental tests used 200 ms processing to solve the problem. Note also that the complexity of the problem involves at first sight the need for more processing, unless the QP solver exploits the structural features of the big emerging banded linear systems that use COTS software is not always available. For example, the formulation spread a prediction horizon of 5 sampling periods generates 42 decision variables, 42 equality constraints forming a system of equations large compared with the complexity of the basic system, while the condensed formulation are generated only 10 variables with 24 inequalities.

The literature shows that there is a cubic growth of computing requirements, such as floating point operations and memory storage, with respect to increased N_p in the condensed formulation contrasts with the linear growth of the spread formulation [34], this if a suitable algorithm for banded matrices which arise in the spread formulation, suggesting its use when it is necessary to work with large prediction horizons. Although not directly comparable elements, Tables 1 and 2 show exponential growth in both formulations, observing the processing time and average number of iterations needed to find the solution, but as other advantageous element, it is that the exponential trend showing condensed formulation is less pronounced than the spread formulation (remember that the QP algorithm used is standard). In fact, the number of iterations required to solve the problem regarding the prediction horizon, you can guess the slope of this trend is low and not clearly exponential, compare the number of iterations to N_p equal to 9 and 10 about 6, 7 and 8.

Regarding the validity of the solution, it is plotted the prediction made by the solution found for both formulations with a horizon of 5 and 7 sampling periods respectively. In Figs. 6 and 7 show such solutions. The solution formulation spread shows the prediction error variable speed state, with a tendency to stabilize at zero, even with the horizon set to 5 does not accurately predict that convergence. On the other hand, predictions respect the boundaries, but show wide variability of solutions of the input variables and therefore the state, in each sampling period, it shows a zigzag convergence behavior of the algorithm. The solution of condensed formulation, does not present output variable, because it is removed or hid-

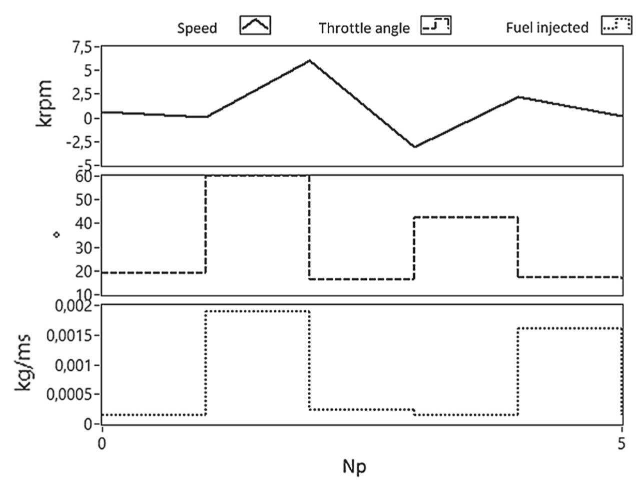


Fig. 6 Scattered solution formulation, with Np = 5

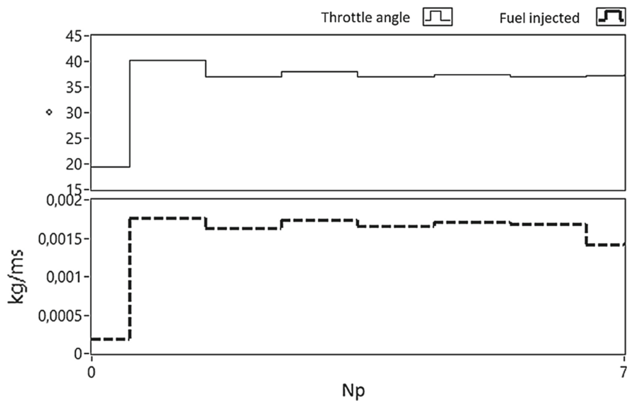


Fig. 7 Condensed solution formulation, with Np = 7

den in the formulation, so the figure on the right shows the solution to a horizon of seven periods corresponding to the input sampling. It can be inferred from on their behavior that the predicted output converges rapidly towards stabilization, even posing a lower horizon. The last aspect to be explored in this report, is the behavior of the active set algorithm for the QP problem formulated through the complete map of possible operating points. This allows to determine regions with curvature characteristics that facilitate the calculation of gradients and Hessians, and then use them as desirable regions to work optimal controller. It has made a sweep through the engine operating map, obtaining the linearized model and solving the optimization problem for some prediction horizons with both formulations. Level graphics show the results of these experiments.

In Fig. 8, the left side, corresponding to a horizon of three samples shows that above 45° opening in a few points resolve the problem in less than 200 ms. The remaining region, shown with better prospects. Remember to Fig. 1, isovelocity lines are almost parallel to the axis of the throttle opening, then it is possible to find areas of operation for any possible system operating speed range of the engine where the problem is resolved efficiently. With lines parallel to the axis of the injected flow, located below 45° can be determine the best initial values for an operating range given, which is the same as saying that can maintain a fixed opening and find a relationship between the operating speed and flow injected to define the starting points of the algorithm. Right side corresponds to a prediction horizon of 5 sampling periods. It is observed that in a few operating points solution in less than 200 ms is achieved.

The map of condensed formulation Fig. 9 shows darker indicating that they are points whose processing time was even less than 100 ms. With the prediction horizon 3 sampling periods, a strong dark band is observed between corresponding parallel lines at 18° to 30° of throttle opening, and a region with low processing times demarcated by an opening of 45° up and 0.0015 kg/ms left. It can be concluded qualitatively better performance of this condensed formulation. Similarly, the right image shows that for experiments with horizons equal to 5 are obtained, best performing regions that spread the formulation.

As a qualitative comparison has been made a map of processing times for both formulations removing restrictions on the parameters, ie relaxing the bounds on the decision variables that are required for the function used COTS "Quadratic Programing.vi". The results are shown in Fig. 10, note that they are representing the levels from zero to 5 ms for the sparse formulation, and up to 2 ms for condensed formulation. The map on the left shows that there is a tendency to improve processing times in zone of relatively low opening of the throttle valve, reiterating the option of keeping the throttle opening lower and varying the flow injected to reach any point of operation. However map condensed formulation, it shows that through the entire area of operation, the algorithm is able to solve the problem in an average of 1 ms, except a region between 30° and 35° of opening.

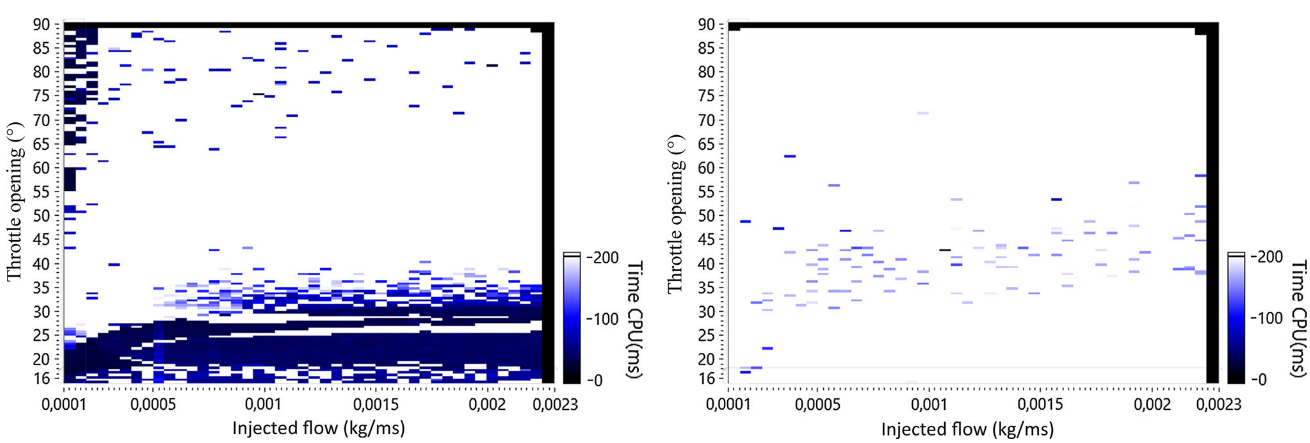


Fig. 8 Operating zones with shorter than 200 ms solution (Formulation spread with Np = 3 *left* and *right* Np = 5)

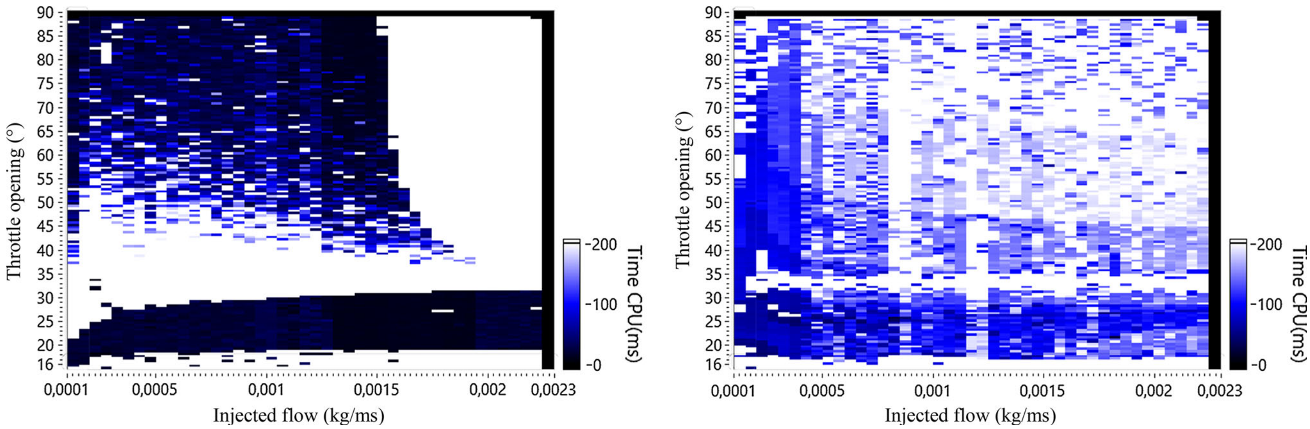


Fig. 9 Operating zones with shorter than 200 ms solution (Formulation condensed with Np = 3 *left* and *right* Np = 5)

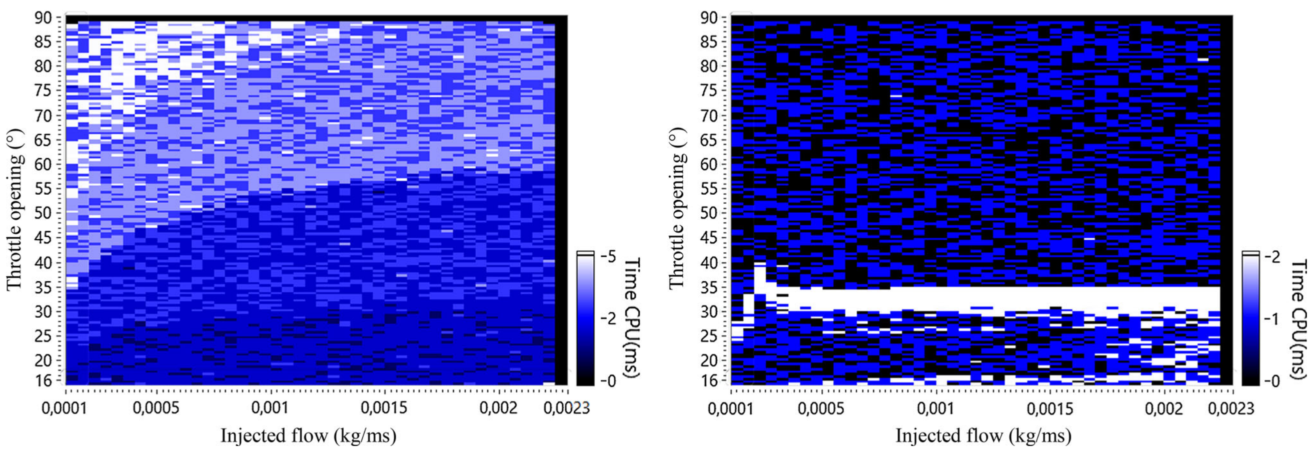


Fig. 10 Unconstrained processing tests for both formulations: scattered *left* and *right* condensed

4.2 Sequential quadratic programming

As another alternative to solve the problem 11 by an embedded system, is the treatment of the constraints imposed by the model in its general form, ie assuming model nonlinearities. As outlined in this report, an alternative for nonlinear optimization algorithm is SQP. In case has been used "Constrained Nonlinear Optimization.vi" function that implements the algorithm SQP in generic form. The documentation of the function is referred to in [38]. As main arguments work, the function expects a continuous-time formulation of the objective function and constraints, limits on the decision variables and values of initial estimation, both in the decision variables and the state of the algorithm (Hessian and Lagrange multipliers). Besides the SQP algorithm, facing the formulation of optimal control problem involves solving the set of equations of the model, and evaluating the solution for the quadratic performance index [39,40]. The model solution to an initial value x_m taken from the sensor readings can be found from a differential equation solver. So it required, cooperation mechanisms between two software: SQP optimizer and EDO solver to solve the problem. There is an inherent discretization when the ODE solver is used, although the model is described in continuous time. In the use of the cooperation of the two mechanisms have been proposed in the literature [6] two possibilities: single shooting and multiple shooting. The second segments the prediction horizon problem and adds several restrictions continuity between them, the first solves a problem only considering the entire prediction horizon. In the case that concerns at this time, it has evaluated only the possibility of single shooting, because multiple shooting is especially recommended for unstable systems.

The interaction between the optimizer SQP, and the solver ODE, in Fig. 11, can be summed up in that for each step of the algorithm SQP, the solver is used to evaluate the prediction horizon and the objective function, using the data obtained the simulation model and then convergence is evaluated. The method used for integration into the solver is Runge–Kutta, and the simulation time defines the prediction horizon, unlike defined for the quadratic programming, here will be treated in units of continuous time (s). Tests were carried algorithm to

Table 3 SQP algorithm testing

Refn	Np = 20 s		Np = 10 s		Np = 5 s	
	Time CPU	Calls ODE	Time CPU	Calls ODE	Time CPU	Calls ODE
0.7	105	24	67	24	58	29
1	148	24	102	29	64	29
1.2	174	24	100	24	79	29
1.4	197	29	134	29	79	29
1.6	227	24	147	29	93	29
1.8	256	24	164	29	101	29
2	284	24	190	29	115	29
2.5	390	29	240	29	137	29
3	396	29	307	29	178	29
3.5	608	29	443	34	204	29
4	709	29	902	49	335	29

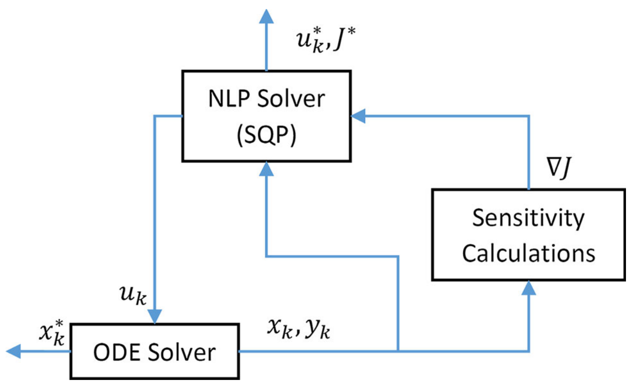


Fig. 11 Single shooting

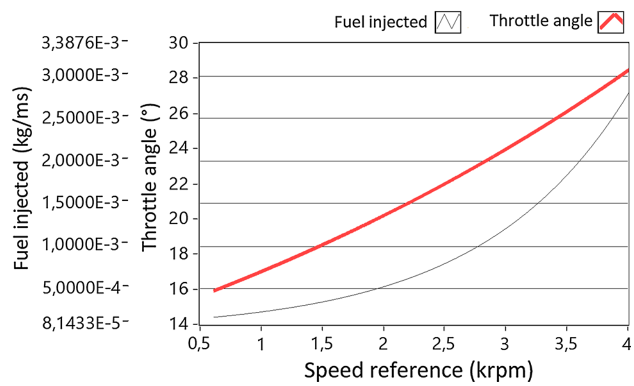


Fig. 12 Experimental mathematical functions to search initial values

a variation range of speeds, to different prediction horizons, analyzing the execution time of the algorithm and the number of simulations taken. The speed variation was imposed *Refn* reference, and the results are summarized in Table 3.

As a first observation, it should be noted that it has been experienced with longer horizons, to the settling time of the plant under the control of the ECU factory, as reported in [7], this would necessarily imply improvements of stability within an MPC framework [39]. Furthermore a horizon of 5 s is comparatively quite long compared to free response times of engine operation, however lower prediction horizons, resulted in greater than 0.1 % of the variable speed errors. The times shown in the table given in milliseconds, are taken by the processor RT to find the optimal running the diagram of Fig. 11, the number of cycles shown in column Calls ODE. The solution of the problem formulation 11 with prediction horizons 5 s, take, from 58 ms to low speed references, and up to 335 ms for high speed references. For relatively very long horizons, 20 s, it varies between 105 and 709 ms. however the times taken in references higher speed, can be improved upon if the model is implemented using adaptive initial conditions, such that it is close to the conditions of the plant to a given operating speed. This is easily solved considering the repeated operation of the algorithm in a control system closed loop. The speed and convergence Local SQP algorithm, also rely heavily the initial values, for it has been built experimentally adjusted functions that determine the initial values of search according to the given reference:

$$m_{iny}\big|_{k=0} = 9.813 \times 10^{-5} e^{0.8390 * Refn}$$
$$\alpha\big|_{k=0} = 14.3339 e^{0.1718 * Refn}$$

Figure 12 shows the practical range of these functions. The initial values of better performance of injected flow for operating speeds below 3 krpm were near areas of low injection. For a change of reference within this range, the initial values close always resulted in convergence of the algorithm. For operating speeds above 3 krpm, the need for more accurate initial values is evident. From there is deducted possible improvement in reducing convergence times obtained in Table 3, if accuracy is increased functions in the areas of operation at high speeds.

5 Conclusions

Three elements have been evaluated in this review with the subsequent aim of implementing optimal controllers for the development of electronic control units for internal combustion engines with spark ignition, detailed here: the type of nonlinear algorithms used for solving the problem dynamic optimization is formulated in each iteration of a loop optimal control; COTS components to solve these algorithms; and embedded hardware for executing the algorithms in real time with interfaces for sensors and actuators. The choice of hardware, which started the study of the application requirements, and is reported in [1] shows satisfactory results in their ability to execute the proposed algorithms, and projected as hardware control schemes NMPC proposed and evaluated in the context of this investigation [41].

Detailing this general conclusion, we point some aspects of the results displayed. In use of the component “Quadratic Programming.vi” a linearization process required in each operating point, which is determined by the “SIM Linearized.vi” component. This process is shown to be the most demanding and not predictive, taking up to 350 ms at times, and regularly between 250 and 300 ms. For free engine response is not optional run in sequence: a linearization algorithm and then “Active set.vi” using as input the model obtained in a sampling period. However the possible solution evaluated in this research is to implement alongside the two COTS “Linearized.vi SIM” and “Quadratic Programming.vi” components under schemes updated model, well established [41]. “Quadratic Programming.vi” is made to solve the formulation of a quadratic programming problem, so the problem has become the engine dynamic optimization of infinite dimension, a finite-dimensional problem for certain moments of time, and adapting as a QP problem by two alternatives: sparse formulation and condensed formulation. The execution of the condensed component formulation has shown the best performance, considering that for horizons of up to 10 sampling periods algorithm converges in 192 ms and growth in CPU time is milder when the prediction horizon increases. The algorithm finds the optimal, in times of less than 200 ms CPU, hardware provided. An analysis of CPU time across the map of engine operating points shows that the delimitation of the throttle opening between 18° and 30° is an area of excellent performance of the algorithm, as such may be selected to redefine feasibility area and/or the initial search points. The operational interrelationship of the components used to solve the problem posed by mediating the method of sequential quadratic programming, resulting in superior performance combined to the possibility of prediction horizons longer than the free response of plant study, which is desirable in terms of stability when the algorithm is embedded in a control system. Horizons of up to 20 s the algorithm converges in less than 100 ms, but this is strongly dependent on initial search point algorithm, so the results have deteriorated as the operating point increases. Building more precise functions can improve the CPU times for the entire operating range.

References

1. Valencia, A., Diaz, A.: Simulation NMPC in 2-HIL to design ECU. Int. J. Interact. Des. Manuf. (IJIDeM). Springer, Paris (2015). doi:10.1007/s12008-015-0265-4
2. Guzzella, L., Onder, C.H.: Introduction to Modeling and Control of Internal Combustion Engine Systems, 2nd edn. Springer, Berlin (2010)
3. Nash, F.C.: Mecánica automotriz: teoría, mantenimiento y reparación. In: Automotive Mechanics: Theory, Maintenance and Repair, vol. 2. McGraw-Hill, Mexico (1991)
4. Unión Europea.: Norma Euro 5 y Euro 6, Bruselas 22 mayo 2007. REGLAMENTO DEL PARLAMENTO EUROPEO Y DEL CONSEJO sobre la homologación de tipo de los vehículos de motor por lo que se refiere a las emisiones procedentes de turismos y vehículos comerciales ligeros (Euro 5 y Euro 6) y sobre el acceso a la información relativa a la reparación y el mantenimiento de los vehículos. In: Standard Euro 5 and Euro 6, Brussels 22 May 2007. REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on type approval of motor vehicles with regard to emissions from passenger cars and light commercial vehicles (Euro 5 and Euro 6) and on access to information on the repair and maintenance of vehicles. Unión Europea, Actos legislativos y otros instrumentos, PE-CONS 3602/2/07 REV 2, pp 1 – 35
5. del Re, L., Ornert, P., Alberer, D.: Chances and challeges in automotive predictive control. In: del Re, L., et al. (eds.) Automotive Model Predictive Control, LNCIS, vol. 402, pp. 1–22. Springer, Berlin (2010)
6. Biegler, L.T.: Nonlinear programming. In: Concepts, Algorithms and Applications to Chemical Proccess. Carniege Mellon University, SIAM, Pittsburgh (2010)
7. Lordui, M., Patricia, M: Modelo Matemático de Tipo Valor Medio de un Motor de Combustión Interna y Estimación de Estados usando Técnicas Bayesianas (Tesis Magistral). In: Mathematical Model of Type Average Value of Internal Combustion Engine and State Estimation Using Bayesian Techniques (Master Thesis). Universidad Eafit, Medellín (2012)
8. Naidu, S.D.: Optimal Control Systems. Idaho State University, CRC Press, Pocatello (2002)
9. Balsa Canto, E.: Algoritmos Eficientes para la Optimización Dinámica de Procesos Distribuidos (Tesis Doctoral) [Efficient Algorithms for Distributed Dynamic Process Optimization (Doctoral Thesis)]. Universidad de Vigo, Vigo (2001)
10. Ogata, K.: Ingeniería de Control Moderna [Modern Control Engineering]. Pearson, Prentice Hall, Madrid, España (2003)
11. Zhai, Y.J., Yu, D.L., Reza, T., Yasser, A.H.: Fast predictive control for air-fuel ratio of SI engines using a nonlinear internal model. Int. J. Eng. Sci. Technol. **3**(6), 1–17 (2011)
12. Cercós, J.N.: Diseño de un Controlador Avanzado Basado en Redes Neuronales para la Gestión de la Mezcla Aire-Gasolina en un Motor Alternativo (Tesis Doctoral) [Design of an Advanced Controller Based on Neural Networks for Management Air-Gasoline Blend

on Alternative Motor (Doctoral Thesis)]. Universidad Politécnica de Cataluña, Barcelona (1999)
13. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006)
14. 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)
15. Holland, J.H.: Adaptations in Natural and Artificial Systems. University of Michigan Press, Michigan (1975)
16. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**, 308–320 (1965)
17. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statistical problems. J. ACM (JACM) **8**(2), 212–219 (1961)
18. Broyden, C.G.: A new double-rank minimization algorithm. Not. Am. Math. Soc. **16**, 670 (1969)
19. Fletcher, R.: A new approach to variable metric methods. Comput. J. **13**, 317–322 (1970)
20. Goldfarb, D.: A family of variable metric methods derived by variational means. Math. Comput. **24**, 23–26 (1970)
21. Shanno, F.: Conditioning of quasi-Newton methods for function minimization. Math. Comput. **24**, 647–657 (1970)
22. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, pp. 481–492. University of California Press, Berkeley (1951). http://projecteuclid.org/euclid.bsmsp/1200500249
23. Andreani, R., Martínez, J.M., Schuverdt, M.L.: On the relation between constant positive linear dependence condition and quasinormality constraint qualification. J. Optim. Theory Appl. **125**(2), 473–485 (2005)
24. De La Fuente O'Connor, J.L.: Técnicas de cálculo para Sistemas de Ecuaciones. Programación Lineal y Programación Entera. Calculation techniques for Systems of Equations. Linear Programming and Integer Programming. Reverté, Barcelona (1998). ISBN 978-84-291-2606-8
25. Nijmeijer, H., Van der Schaft, A.: Nonlinear Dynamical Control Systems. Springer, Berlin (1990)
26. Gill, P.E., Murray, W., Saunders, M.A.: Large-scale SQP methods and their application in trajectory optimization. Comput. Opt. Control ISNM Int. Ser. Numer. Math. **115**, 29–42 (1994)
27. Gould, N.I.M., Toint, P.L.: SQP methods for large-scale nonlinear programming. Syst. Modell. Optim. IFIP Int. Feder. Inf. Process. **46**, 149–178 (2000)
28. Boggs, P.T., Tolle, J.W.: Sequential quadratic programming. Acta Numer. **4**, 1–51 (1995)
29. National Instruments Corp.: Operating instructions and specifications. CompactRIO cRIO-9075/9076. Product Manuals National Instruments # 375650b (2011)
30. National Instruments Corp.: ECU designing and testing using national instruments products. In: White papers of National Instruments #3312 (2009)
31. National Instruments Corp.: Hardware-in-the-loop (HIL) test system architectures. In: White Papers of National Instruments #10343 (2010)
32. National Instruments Corp.: LabVIEW 2014 Help. Quadratic programming VI. In: Documents Supporting National Instruments (2014)
33. Ogata, K., Pérez, J.G.A.: Sistemas de Control en Tiempo Discrete [Control Systems in Discrete Time]. Prentice-Hall Internacional, Mexico (1996)
34. Jerez, J.L.: Custom Optimizations Algorithms for Efficient Hardware Implementations (Tesis Doctoral). Imperial College London, Londres (2013)
35. Ferreau, J.: An Online Active Set Strategy for Fast Solutions of Parametric Quadratic Programs with Applications to predictive engine Control (Tesis Magistral). Universität Heidelberg, Alemania (2006)
36. Dimitrov, D., Sherikov, A., Wieber, P.-B.: A sparse model predictive control formulation for walking motion generation. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2292–2299 (2011). doi:10.1109/IROS.2011.6095035
37. Jerez, J.L., Kerrigan, E.C., Constantinides, G.A.: A condensed and sparse QP formulation for predictive control. In: 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando (2011)
38. National Instruments Corp.: LabVIEW 2014 help. Constrained nonlinear optimization VI. In: Documents Supporting National Instruments (2014)
39. Algöwer, F., Findeisen, R., y Nagi, Z.K.: Nonlinear model predictive control: from theory to application. J. Chin. Inst. Chem. Eng. **35**(3), 299–315 (2004)
40. Albersmeyer, J., Beigel, D., Kirches, C., Wirsching, L., Bock, H.G.y, Schlöder, J.P.: Fast nonlinear model predictive control with an application in automotive engineering. In: Magni, En L., et al. (eds.) Nonlinear Model Predictive Control. Towards New Challenging Applications, pp. 471–480. Springer, Berlin (2009)
41. Valencia, A., Duque, A., Diaz, A.: Evaluation of two strategies NMPC into HIL applied to the operation of an internal combustion engine. In: International Virtual Concept Workshop on INDUSTRIE 4.0. Donostia/San Sebastián, Spain, 26–27, November 2015 (2015). ISBN 978-2-9548927-1-9, VC2014_0025