HC12: Efficient PID Controller Design

R. Matousek, *Member, IAENG*, P. Minar, S. Lang, M. Seda

Abstract— **The control problems leak out into the most areas of human operations. It includes machine control and business control operations as well. In the areas of technical relevance the concept of PID controller (Proportional-Integral-Derivative controller) is widely known. The tuning process of PID controller can be solved by experimental way (the expert is needed) or determined as optimization task. An optimal PID control design includes more goals of optimal regulation process which are often contradictory. Optimal setting of PID controller is generally a task of nonlinear mathematical optimization which is furthermore done on top of dynamic system. The multi-criterion optimization of PID controller by means of soft computing optimization method HC12 is presented. The paper extends previous work [1] and shows some new conclusions. Presented solutions are compared with classical control design methods Ziegler-Nichols and Modulus Optimum.**

Index Terms—**HC12, PID, Optimal Control Design, PID Tuning, Magnetic Levitation System**

I. INTRODUCTION

SOFT-COMPUTING methods provide very robust tools usable in tasks of mathematical optimization. In this paper shall be presented relatively new optimization method denoted as HC12 [1], [2], [3] and [3]. HC12 algorithm is Soft Computing optimization method, capable of solving not-differentiable, nonlinear and multi-modal objective functions. As optimization task the optimal setting of PID controller for three dynamic systems has been chosen. From the point of view of optimization it is a nontrivial task of setting the optimum parameters of dynamic system. In general it is a task nonlinear and multi-criterion. From point of view of evolutionary algorithms (generally optimization soft-computing algorithms) this type of task can be viewed as challenge which solution has very practical implications [6], and [7]. Apart from automation where PID controllers have very wide tradition it is a new approach which can support or compete with present methods of controller setting.

PID controller can be used to control position, velocity, revolutions, temperature, etc. At present time because of major usage of digital technology it is used mainly the digital variant of PID controller. Commonly the PID algorithm is in the process of control implemented using programmable platform (Programmable Logic Controllers, Distributed Control System) or stand alone controllers.

There are key questions to control process behavior apart from the paradigm of the controller (PID, fuzzy, state-space, etc.). In some applications is prohibited to overshoot of desired value in another case it can be but rise time is most important. Also behavior of control action has to reach some limitations. Some of requirements to control process are in fact contradictory.

The paper presented efficient optimal PID controller design approach based on the HC12 optimization algorithm. The optimal PID parameters design is transformed into corresponding optimization problem which is convenient for HC12 applications. There are also other various soft computing methods such as Differential Evolution (DE), Simulated Annealing (SA), Genetic Algorithms (GA) and many more [4], [10] etc. Presented results and implementation of algorithms have been realized using the tools of Matlab/Simulink environment and Java for implementation of optimization solvers.

Revised and extended version received January 29, 2012. This work was supported by the research projects of MSM 0021630529 "Intelligent Systems in Automation", and IGA FSI-S-11-31 "Application of Artificial Intelligence".

All authors are from Brno University of Technology, Faculty of Mechanical Engineering, Dept. of Applied Computer Science, Technicka 2896/2, 61669 Brno, Czech Republic, corresponding author Radomil Matousek have e-mail matousek@fme.vutbr.cz.

II. PID CONTROLLER

The theory of control deals with methods which leads to change of behavior of controlled dynamic system (further only system). The desired output of a system is called the reference or set point. When one or more outputs of the system need to follow a certain reference over time then a controller modifies the inputs of system to obtain the desired value on the output of the system, Fig. 1.

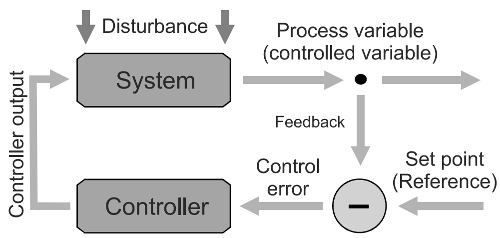

Fig. 1. The general concept of the negative feedback loop to control the dynamic behavior of the system with description of the major parts.

The PID controller has three separate constant parameters: Proportional (P), Integral (I) and Derivative (D). It can be said the P depends on present error, I on accumulation of past errors and D is prediction of future errors based on rate of change. The PID controller calculates an error value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the control error by adjusting the process controller outputs. After corrective action from the controller the system should reach point of stability as the result. Stability means the set point is being held on the output without oscillating around it.

(Advance online publication: 27 February 2012)

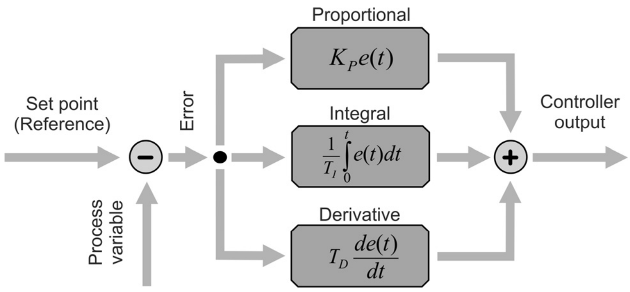

Fig. 2. The block diagram of the PID controller.

The controller in control system given by Fig. 1 is the classical one degree-of-freedom (1-DOF) controller. Description of controller is provided in forms of formulae or algorithms. Basic block diagram of standard PID controller is based on parallel circuit, Fig. 2. The proportional, integral, and derivative terms are summed to calculate the output of the PID controller. Defining $u(t)$ as the controller output, the general ideal form of the PID algorithm is:

$$u(t) = K_P e(t) + \frac{1}{T_I}\int_0^t e(\tau)d t + T_D \frac{de(t)}{dt} \qquad (1)$$

where K_P is the single gain, $K_I = K_P / T_I$, $K_D = K_P \cdot T_D$, T_I is the integral time constant and T_D is the derivative time constant.

In many cases are used the variants of PID controller given in standard form by equation (2). Mutual conversion of controller's constants is obvious.

$$u(t) = K_P \left[e(t) + K_I \int_0^t e(\tau)d t + K_D \frac{de(t)}{dt} \right] \qquad (2)$$

In this standard form, the parameters have a clear physical meaning. In particular, the inner summation produces a new single error value which is compensated for future and past errors. The addition of the proportional and derivative components effectively predicts the error value at T_D seconds (or samples) in the future, assuming that the loop control remains unchanged. The integral component adjusts the error value to compensate for the sum of all past errors, with the intention of completely eliminating them in T_I seconds (or samples). The resulting compensated single error value is scaled by the single gain K_P. Other advanced forms of PID controllers with better real properties can be found in [5].

Using Laplace's transformation the transfer function of PID controller looks like the (3).

$$\begin{aligned} G_C(s) &= K_P + \frac{K_I}{s} + K_D s = \ldots \\ &= K_P \left[1 + \frac{1}{\frac{K_P}{K_I} s} + \frac{K_D}{K_P} s \right] = K_P \left[1 + \frac{1}{T_I s} + T_D s \right] \end{aligned} \qquad (3)$$

It is good to say that in a real PID controller with derivative action is proper to limit the high frequency gain. It can be done by implementing the first order filter with time constant T_D / N, where constant N has typical values 8 to 20.

$$G_C(s) = K_P + \frac{K_I}{s} + \frac{K_D s}{N s + 1} = K_P \left[1 + \frac{1}{T_I s} + \frac{T_D s}{\frac{T_D}{N} s + 1} \right] \qquad (4)$$

In case of the paper we are used mainly standard form PID controller given by (3) instead of (4) [9] because modeling examples. On the other side the expansion to variable or constant N is possible (/w N).

Tuning a control loop is the adjustment of its control parameters (gain/proportional band, integral gain/reset, derivative gain/rate) to the optimum values for the desired control response. Designing and tuning a proportional-integral-derivative (PID) controller appears to be conceptually intuitive, but can be hard in practice, if multiple (and often conflicting) objectives such as short transient and high stability are to be achieved. There are several methods for tuning a PID loop. In our paper we will designate the found parameters as in (2) or (3), i.e. $\{K_P, K_I, K_D\}$.

In order to solve optimal setting of controller's parameters we will use reformulation of the problem. Optimal solution will be searched for by HC12 algorithm as optimum of given objective function. However for comparison there are used common methods for controller parameters tuning, i.e. empirical method Ziegler-Nichols (Ziegler, Nichols 1947) and Modulus Optimum method [12]. Of course many other PID controller tuning methods exist which consider desired properties of control loop.

A. Ziegler-Nichols Tuning Method

Ziegler-Nichols (ZN) tuning rule was the first such effort to provide a practical approach to tune a PID controller. According to the rule, a PID controller is tuned by firstly setting it to the P-only mode but adjusting the gain to make the control system in continuous oscillation (the edge of the stability). The corresponding gain is referred to as the ultimate gain K_u and the oscillation period is denoted as the ultimate period P_u.

The key step of the Ziegler-Nichols tuning approach is to determine the ultimate gain and period. Then, the PID controller parameters are determined from K_u and P_u using the Ziegler-Nichols tuning Table I.

TABLE I
COMMONLY USED ZIEGLER-NICHOLS TUNING CHART*

Controller	K_P	K_I	K_D
P	0.5 K_u	---	---
PI	0.45 K_u	1.2 K_P / P_u	---
PID	0.6 K_u	2 K_P / P_u	$K_P P_u$ / 8
PID $_{\text{Low Overshoot}}$	0.2 K_u	2 K_P / P_u	$K_P P_u$ / 3

* Ziegler, J.G and Nichols, N. B. (1942).
Optimum settings for automatic controllers [14].

B. Modulus Optimum

Modulus Optimum (MO) method is based on the transfer function of set point $G_{ref}(s)$, where this transfer function is ratio of Laplace s-domain of process output variable to set point input variables. In ideal case the transfer function would be $G_{ref}(s) = 1$, i.e. step response of process variable is equal to set point. In frequency domain it corresponds with following condition (5).

$$\begin{aligned} &A_{ref}(\omega) = 1 \\ &\Rightarrow \left|G_{ref}(j\omega)\right| = A_{ref}(\omega) = 1 \end{aligned} \tag{5}$$

This condition can not be satisfied in reality, however it can be proven that control process ends the fastest when amplitude characteristics $A_{ref}(j\omega)$ will be flat at first and then it will monotonically decreasing as we can see in Fig. 3.

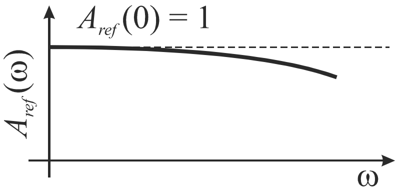

Fig. 3. The block diagram of the PID controller.

Description of this method can be found in v [12]. The setting of PID parameters K_P, T_I and T_D by MO method is sorted in the table for practical use and it depends on the type of controlled plant, Table II.

TABLE II
CALCULATION OF PID CONTROLER'S PARAMTERS BY MO METHOD*

Model of controlled plant	K_P	T_I	T_D
$\dfrac{k}{(T_1 s+1)(T_2 s+1)(T_3 s+1)}$ $T_1 \geq T_2 \geq T_3$	$\dfrac{T_i}{2kT_3}$	$T_1 + T_2$	$\dfrac{T_1 T_2}{T_1 + T_2}$

* Example of calculation of PID controller's parameters by Modulus Optimum method. Given controlled plant corresponds with our test plant which is described by transfer function $S1(s)$ in next part of the paper.

III. EXPERIMENTAL PLANTS

For out tests we have developer three dynamic systems. First and second are artificially created systems given by simple transfer function $S_1(s)$ and $S_2(s)$. On these systems there are shown classic methods of PID controller tuning (Ziegler-Nichols, Modulus Optimum) and multi-criterion tuning using soft computing method HC12. Optimization method HC12 have been compared for objective consideration with classic method of nonlinear optimization Nelder-Mead [13]. Also was tested the PID controller variant denoted as PIDF with correction of derivative part by (4). This is implementation of the first order filter with time constant T_D / N, where constant N was chosen from 1 to 128.

Third dynamic system have a real physical basis and it is denoted as MGL. It is a model of magnetic levitation. In this case there are shown results of multi-criterion optimization of PID controller using HC12 method.

A. Plant S1 (3rd order stable system)

The test system S1 is given in time domain by (6) and in s domain by transfer function (7) or more practically by transfer function in gain-time constant form by (8). This is open loop stable system of third order with one step response by Fig. 4.

$$y_1(t) = \frac{3}{2}e^{-\frac{1}{2}t} - 6e^{\frac{1}{4}t} + \frac{9}{2}e^{-\frac{1}{6}t} \tag{6}$$

$$S_1(s) = \frac{6}{48s^3 + 44s^2 + 12s + 1} \tag{7}$$

$$S_1(s) = \frac{6}{(2s+1)(4s+1)(6s+1)} \tag{8}$$

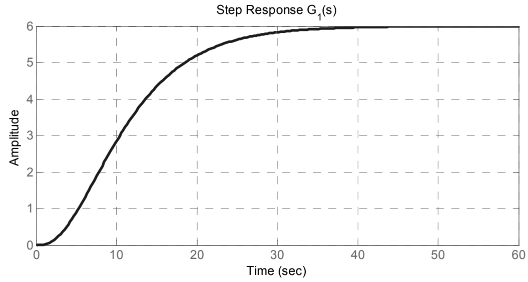

Fig. 4. Unit step response of system $S_1(s)$ with zero initial state.

Parameters required for the use of Ziegler-Nichols method are ultimate gain $K_u = 1.65$ and corresponding period $P_u = 12.6$ (s^{-1}). These values correspond with calculated pa-rameters PID_{ZN} = {0.99, 0.16, 1.575} according to Table I.

For Modulus Optimum method are crucial parameters of plant {k, T_1, T_2, T_3} = {6, 6, 4, 2}. According to Table II. these values correspond wits calculated parameters PID_{MO} = {0.08, 0.16, 1.33}.

B. Plant S2 (3rd order oscillation system)

The test system S2 is given in time domain by (9) and in s-domain by transfer function (10) or more practically by transfer function in gain-time constant form by (11). This is open loop oscillating system of third order with one step response by Fig. 5.

$$y_2(t) = \frac{2}{5}\sin\left(\frac{1}{4}t\right) - \frac{1}{5}\cos\left(\frac{1}{4}\right) + \frac{1}{5}e^{-\frac{1}{2}t} \tag{9}$$

$$S_2(s) = \frac{2}{32s^3 + 16s^2 + 2s + 1} \tag{10}$$

$$S_2(s) = \frac{2}{(2s+1)(4s+i)(4s-i)} \tag{11}$$

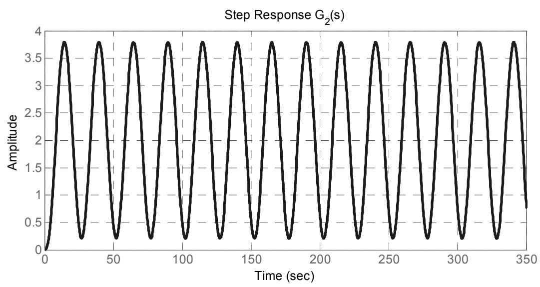

Fig. 5. Unit step response of system $S_2(s)$ with zero initial state.

C. MGL System

The objective of this experiment is to design a controller that levitates the steel ball from the post and makes it track a specified position trajectory. Magnetic Levitation system (MGL) which is used for our experiments is based on the balance of all forces acting on the steel ball. System is driven by two signals which represents desired position of the steel ball in magnetic field. The motion equation is given by (12).

$$F_a = F_m - F_g - F_d \quad (\mathrm{N}) \tag{12}$$

where the forces are

$F_a = m\ddot{x}$	the accelerating force (N),
$F_m = \dfrac{k_c i^2}{(x-x_0)^2}$	the electromagnetic force (N),
$F_g = mg$	the gravity force (N),
$F_d = k_d \dot{x}$	the damping force (N),
k_d	the damping constant (N/m.s),
k_c	the coil constant,
x	the ball position (m),
x_0	coil offset (m)
m	the mass of the ball (kg).

The (12) can be written as (13) for our Matlab/Simulink® realization. And also in real time realization with using Real-Time Toolbox for Simulink and MF624 multifunction I/O card (Humusoft, Ltd.). The matching block model of MGL system is showed in Fig. 6.

$$\ddot{x} = -\frac{k_d}{m}\dot{x} - g + \frac{k_c}{m}\left(\frac{i^2}{(x-x_0)}\right) \tag{13}$$

The power amplifier is designed as a source of constant current i, controlled by the input voltage signal u. The position sensor can be approximated with a linear function between the ball position x and the sensor voltage output y.

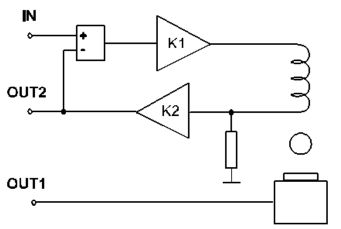

Fig. 6. Sketch diagram of MGL system.

IV. Optimal PID Tuning Using HC12

Goal of this paper is to present optimal tuning of PID controller (2) using relatively novel soft computing optimization method denoted HC12. For every optimization process it is not only crucial the selection of the solver but the design of objective function as well. Further it is worthy to note that in case of soft computing methods the way of coding is very important, i.e. the representation of searched parameters of the task.

Implementation of the HC12 algorithm have been done using Java environment. Plants have been done using Matlab/Simulink® environment. This join have ensured effective realization of simulation model necessary for calculation of objective function and also effective implementation of HC12 solver.

A. Objective Function

In case of optimal PID parameter tuning it can be counted for many demands of resulting control process. For example shortest time of control process, zero steady state error, zero overshoot, no oscillation, etc. Moreover the demands of optimization can be combined. Examples of common performance integral criteria for optimal control design are in Table III.

TABLE III
Common Integral Objective Functions

Label	Caption	Formula*
ISE	*Integral of Squared Error*	$f_{ISE} = \int_0^t e^2(t)dt$
IAE	*Integral of Absolute Error*	$f_{IAE} = \int_0^t \lvert e(t)\rvert dt$
ITSE	*Integral of Time multiply Squared Error*	$f_{ITSE} = \int_0^t te^2(t)dt$
ITAE*	*Integral of Time multiply Absolute Error*	$f_{ITAE} = \int_0^t t\lvert e(t)\rvert dt$

* This control error criteria was used in our experiments.

In our case we have the objective function f_{TOTAL} composed as sum of three objective functions which penalize inconvenient behavior of transitional process. To given type of optimization problem it is convenient to note that it is integral way of creating the objective function. The value of objective function (penalty functions) is obtained after finish of the simulation.

$$f_{TOTAL} = \alpha\log(f_{ITAE}) + \beta\log(f_{OVER}) + \gamma\log(f_{WAVE}) \tag{14}$$

$$[K_P, K_I, K_D]^* = \arg\min_{\mathbf{K}\in\mathbb{R}^3} f_{TOTAL}(\mathbf{K}) \tag{15}$$

Where α, β, γ are weight coefficients. Penalty function f_{OVER} calculates all overshoot where response signal (process variable) exceeds its target (set point). In case of overshoot the function value is increased by one every sample period. Penalty function f_{WAVE} calculates sum of all detected oscillations. Oscillation is detected if the shape of the process variable signal changes from concave to convex. The vector $[K_P, K_I, K_D]^*$ corresponds to find optimum PID parameters.

Principle of penalization is well known variant of evaluation in case of multicriterion optimization. However, determining of the weights of individual objective functions can be difficult. In case of our implementation we have reduced it significantly using logarithm function as is shown in (14). This practice can't be applied generally, it has to be always confronted with the choice of individual penalization functions.

B. HC12 Algorithm

The basic principle of HC12 algorithm is very simple. Shortly, for a given optimization problem, in each iteration step i, a solution ($\mathbf{A}_{kernel,\,i}$) exists to which a neighborhood of further possible solutions is generated using a fixed pattern.

From this neighborhood, the best solution is chosen for iteration step $i+1$, which will again be used to generate a new solution ($\mathbf{A}_{kernel,\ i+1}$). The algorithm stops if no best solution can be found, that is, if (for a minimization problem)

$$\min\left(f\left(\mathbf{A}_{kernel,i}\right)\right) \le \min\left(f\left(\mathbf{A}_{kernel,i+1}\right)\right) \tag{16}$$

where i is the iteration number and f is the objective function. A mathematical description of the algorithm can be found in [1] and [3].

Here the basic ideas are summarized: The solution of a given optimization problem is represented by a binary vector $\mathbf{A}$. This binary vector $\mathbf{A}$ codes k real parameters of the optimization problem, that is, the real input parameters x_i of the objective function. This provides a basis for discretizing the domain of definition of the problem parameters to be found. The degree of discretization depends on the size of the binary string being proportional to 2^s where s is the number of bits per parameter.

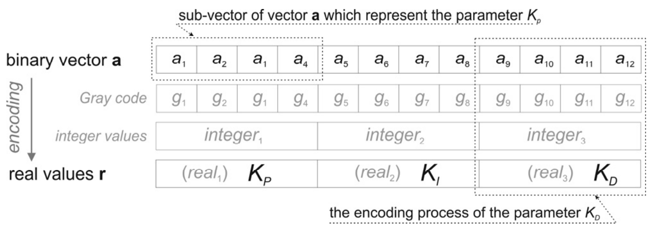

Fig. 7. Example of generating 4-bits parameters encoding scheme (binary and Gray code, integer and real parameters). As can be seen from the figure the number of bits determine the precision of the encoded real parameters. In our experiments we used 10-bit per parameter, i.e. precision approximately 1e-2 for all PID tuning parameters $\{K_P, K_I, K_D\}$.

In the first iteration, a binary vector$\mathbf{A}_{kernel,1}$ and a neighborhood to fit a fixed pattern are randomly generated. With HC12, this is a neighborhood with distances 1 and 2 from vector $\mathbf{A}_{kernel}$ in the sense of the Hamming metric. In each iteration, the best solution is chosen as the new basis. The Hamming distance ρ_H between two binary vectors of equal length is the number of positions for which the corresponding symbols are different.

The principle of decoding a binary string to a vector of real parameters, which are in our case the PID parameters $\{K_P, K_I, K_D\}$ is shown in Fig. 7. It follows from the principle that the cardinality (size) of the neighborhood for a Hamming distance of 1 corresponds to the length n of the binary vector thus growing linearly with the length of the binary vector.

Applying the principle of addition in modular arithmetic (mod 2) and using the special matrixes $\mathbf{M}$ (as a Mask) we can derived neighborhood vectors $\mathbf{A}_{neighborhood}$ by (17)

$$\mathbf{A}_{neighborhood} = \left(\mathbf{A}_{kernel} \oplus \mathbf{M}_1\right) \cup \left(\mathbf{A}_{kernel} \oplus \mathbf{M}_2\right) \tag{17}$$

where $\mathbf{M}$ are the mask matrixes given by (18) and the indexes $i = 1, 2$ are corresponding with $\rho_H = 1$ and $\rho_H = 2$.

Is should be noticed that generalization of the neighborhood generated process is possible ($\rho_H = 3$, $\rho_H = 4$, ... , $\rho_H = n$), but for real life optimization tasks the combinatorial expansion is ruinous.

TABLE IV
COMMON PERFORMANCE CRITERIA* FOR PID TUNING

Label	Description
Overshoot (OVER)	Overshoot occurs when the output signal exceeds its set point. It arises especially in the step response, often followed by ringing.
Peek time (PET)	Peek is the highest value reached by the response before reaching the desired value, therefore peak-time is the time when the peek is reached.
Rise time (RIT)	Rise time is the time required for the response to rise from x% to y% of its final value", with 0%-100% rise time common for overdamped second order systems. (90% in our description was used)
Setting time (SET)	Settling time of output signal is the time elapsed from the application of an ideal step input to the time at which the output is equal to set point within tolerance. (2% is our set point tolerance)

* There are a lot of criteria which can be used for comparison of performance ratio in case of optimal PID tuning.

$$\mathbf{M}_1 = \begin{pmatrix} 1 & 0 & \cdots & 0_{1,n} \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0_{n,1} & & & 1_{n,n} \end{pmatrix}$$
$$\mathbf{M}_2 = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0_{1,n} \\ 1 & 0 & 1 & & 0_{2,n} \\ \vdots & & & \ddots & \\ 0_{\binom{n}{2},1} & & & & 1_{\binom{n}{2},n} \end{pmatrix} \tag{18}$$

The computational time complexity is the amount of steps the algorithm has to do accordingly to the number of inputs. In the big O notation the behavior of function is analyzed when number of inputs is very high. Algorithm HC12 has quadratic complexity $O(n^2)$.

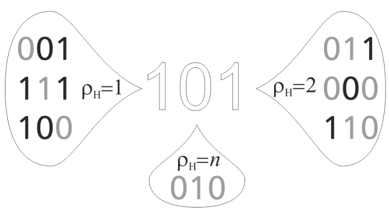

Fig. 8. An example of 3-bits neighborhood generating.

It generates neighborhood using bit masks. Those masks have Hamming distances one and two and rank of matrices $\mathbf{M}$ are n and n choose 2 respective, where n is the length of binary string which represent the problem solution. The short example of HC12 possibility to generate neighborhood with Hamming distances 1 and 2 for 3-bits string is shown above in Fig. 8.

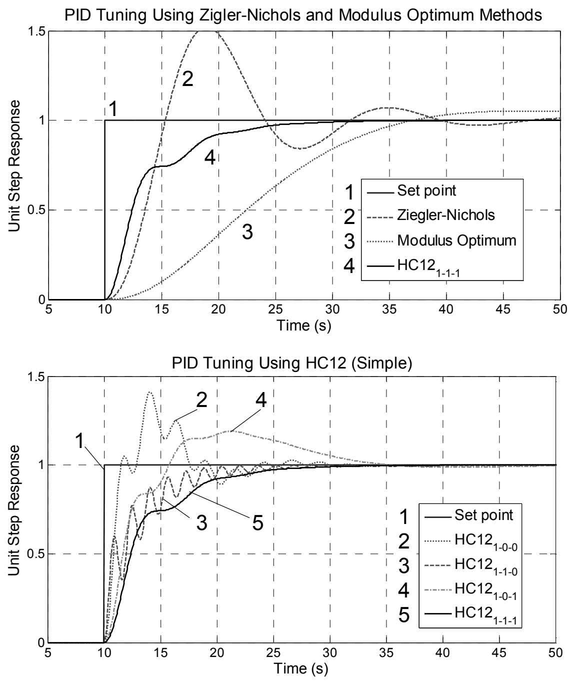

Fig. 9. Unit step responses for the plant S1 using ZN, MO and $HC12_{1\text{-}1\text{-}1}$ PID tuning methods (above). Unit step responses using HC12 PID tuning methods for basic variants of penalty functions (bottom).

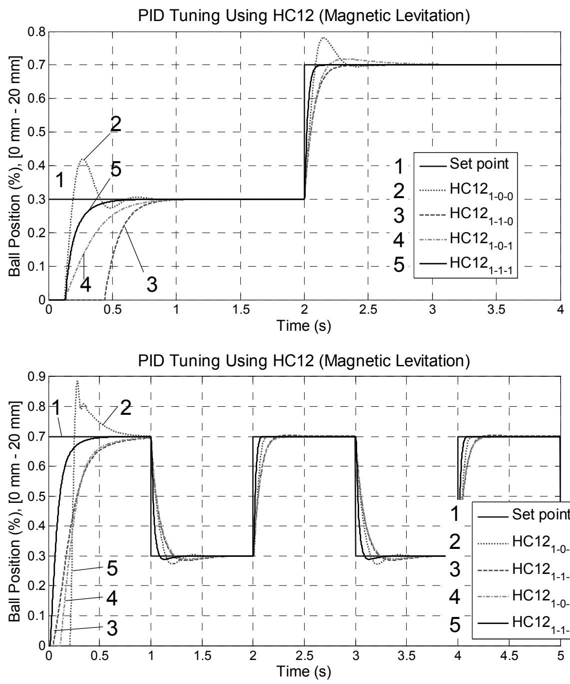

Fig. 10. Double step responses for the plant MGL using HC12 PID tuning methods and basic variants of penalty functions (above). The kind of the test but for PWM input signal reference (bottom).

C. Nelder-Mead Method

The Nelder-Mead algorithm (NM) or simplex search algorithm, originally published in 1965 (Nelder and Mead, 1965), is one of the best known and successful algorithms for multidimensional unconstrained optimization without derivatives. In our case we use this algorithm for the comparison of HC12 performance.

The Nelder-Mead method is simplex-based. A simplex S in $\mathbb{R}^n$ is defined as the convex hull of n+1 vertices $x_0,\ldots,x_n \in \mathbb{R}^n$. For example in our case of PID parameters tuning $\{K_P, K_I, K_D\}$ a simplex is tetrahedron. Nelder–Mead generates a new test position of simplex by extrapolating the behavior of the objective function measured at each test point arranged as the simplex. The algorithm then chooses to replace one of these test points with the new test point and so the technique progresses. Our implementation of the NM was based on Matlab function fminsearch [13]. Furthermore, the presented results are obtained as optimum from 20 runs of the NM algorithm with different start points.

V. Tests and Results

In this section there will be presented the results of optimal tuning of PID controller using presented strategies. For better objectiveness and possibility of comparing our results with others we mention PID tuning parameters and descriptive characteristics (popular performance criteria) of control process in Table IV.

Optimization of parameters of PID controller using HC12 algorithm has been done 20 times for every case of plant and configuration of weight coefficients in (14). The reason is much more better than in case of Z-N tuning and similar as in the case of method NM, certain sensitivity of the solution to first iteration of algorithm.

Naturally, the best idea about results of the control and effect of selected penalization functions (14) to resulting controller's setting (15) are given to us by respective step responses. These characteristics best describe effectiveness of PID tuning by HC12 in comparison with other methods and also show very well the impact of penalization to given control process.

A. Plant S1 and S2

This artificially designed plant of third order has been taken as reference, i.e. plant for comparison of some known methods for tuning of PID controller with our developed approach to PID tuning by HC12. Computed parameters of controller and values of performance criteria are in Table V.

Unit step responses for the ZN and MO methods are in Fig. 9. The Fig. 9 clearly shows influence of penalty functions where values 0/1 means if the given weight coefficient (α, β, γ) enables the use of penalty function and also value 1 means real value of given weight coefficient. In the text we will use notation α-β-γ as index of given HC12 optimization, i.e. $HC12_{\alpha\text{-}\beta\text{-}\gamma}$.

B. MGL System

The plant is represented by (13) and real time realization match chart diagram in Fig. 6. This nonlinear high speed MGL system is real equipment in our laboratory (MGL model CE152 made by Humusoft® Ltd.). Therefore the actuating signal corresponds with real values of desired position of steel ball levitating above the ground. The physical parameters are: $m = 0.008$ (kg), $x_0 = 0.01$ (m), $k_d = 0.02$ (N/m·s), $\tau_a = 0.003$ (s), $k_a = 1.5$ (A/V), $k_x = 200$ (A/V), $y_0 = 0$ (V), $k_c = 2.16\text{e-}6$.

Computed parameters of controller and statistic characteristics of objective function are in Table VI. There are 20 runs per penalization's variant of objective functions.

TABLE V
THE RESULTS OF PID TUNING FOR S1 AND S2 PLANTS*

S1 | S2	PID Parameters			PID Performance Criteria			
Method	K_P	K_I	K_D	OVER	PET	RIT	SET
ZN	0.990	0.159	1.575	0.516	8.9	5.3	40.6
MO	0.083	0.166	1.333	0.052	36.5	26.9	74.3
$HC12_{1\text{-}0\text{-}0}$	24.80	0.646	2.126	0.411	4.0	1.5	16.6
$HC12_{1\text{-}1\text{-}0}$	16.14	0.004	3.071	0	0	11.9	13.6
$HC12_{1\text{-}0\text{-}1}$	4.724	0.157	3.543	0.191	11.3	5.7	30.9
$HC12_{1\text{-}1\text{-}1}$	2.362	0.018	4.016	0	0	12.4	14.1
/w PIDF	2.052	0.145	7.841	0.282	3.25	2.05	14.6
$HC12_{1\text{-}0\text{-}0}$	13.62	1.25	3.81	0.519	5.7	1.9	33.35
$HC12_{1\text{-}1\text{-}0}$	4.103	0.027	5.277	0	0	24.4	19.6
$HC12_{1\text{-}0\text{-}1}$	5.685	0.525	5.250	0.432	7.7	4.55	38.55
$HC12_{1\text{-}1\text{-}1}$	0.420	0.122	9.459	0	0	23.84	12.37
/w PIDF	0	0.269	4.320	0.183	6.9	4.5	21.1

* There is unit step function as set point. The HC12 PID Tuning was realized with following intervals: $K_P, K_D \in [0, 50]$, $K_I \in [0, 2]$ and in case of filtration $N \in [1,128]$. The bit sizes for all K is 11b, N is 7b respectively.

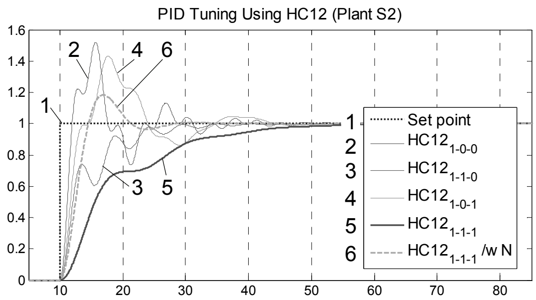

Fig. 11. Unit step responses using HC12 PID tuning methods for basic variants of penalty functions and $HC12_{1\text{-}1\text{-}1}$ variant with N filtration.

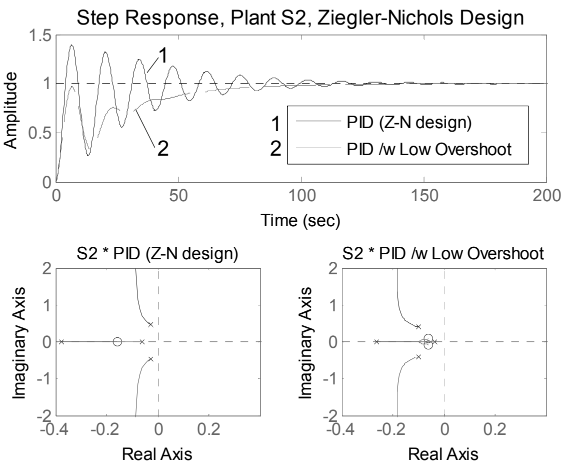

Fig. 12. Unit step responses using Ziegler-Nichols PID tuning method in two variants given by Table I. There are ideal PID controllers by (3).

VI. CONCLUSION

In this paper we introduce novel efficient optimal PID tuning method based on soft computing optimization algorithm HC12 and sophisticated design of objective function as well. The HC12 has been compared with classical PID tuning methods ZN and MO. The comparison with traditional clever optimization methods NM has shown that the HC12 is more stable in case of optimal solution. Efficient results are summarized in Tables V and VI, and also are very clearly displayed by comparing the figures Fig. 9, Fig. 10, Fig. 11 and Fig. 12.

TABLE VI
THE RESULTS OF PID TUNING FOR MGL PLANT

INPUT	OPTIMIZATION	PID Parameters*			Objective function exp. values characteristics		
		K_P	K_I	K_D	MEDIAN**	STD**	MIN
DOUBE STEP	$HC12_{1\text{-}0\text{-}0}$	1.181	19.98	0.078	12.543	1.256	12.72
	$HC12_{1\text{-}1\text{-}0}$	1.299	3.779	0.079	49.911	2.170	15.20
	$HC12_{1\text{-}0\text{-}1}$	1.299	3.779	0.079	18.353	4.953	15.33
	$HC12_{1\text{-}1\text{-}1}$	1.765	7.221	0.031	34.862	4.001	8.844
	$NM_{1\text{-}0\text{-}0}$	3.696	7.643	0.014	69.224	3.075	6.704
	$NM_{1\text{-}1\text{-}1}$	1.930	10.22	0.021	51.919	4.144	7.058
PWM	$HC12_{1\text{-}0\text{-}0}$	0.354	19.98	0.079	31.047	1.887	22.57
	$HC12_{1\text{-}1\text{-}0}$	1.063	5.354	0.079	70.537	1.250	54.48
	$HC12_{1\text{-}0\text{-}1}$	0.827	6.299	0.079	51.379	1.136	50.06
	$HC12_{1\text{-}1\text{-}1}$	1.255	11.04	0.031	90.737	1.101	71.62
	$NM_{1\text{-}0\text{-}0}$	2.171	11.75	0.018	59.045	1.471	33.62
	$NM_{1\text{-}1\text{-}1}$	0.804	12.01	0.036	176.78	2.062	79.87

* The parameters' limitation: $K_P \in [0,10]$, $K_I \in [0,30]$, $K_D \in [0,5]$.
** The median and standard deviation is proper basic characteristics for comparison of stability solution HC12 vs. NM optimizations.

REFERENCES

[1] R. Matousek, P. Minar, S. Lang, P. Pivonka, Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2011, WCECS 2011, 19-21 October, 2011, San Francisco, USA, 2011.

[2] R. Matousek, "GAHC: A Hybrid Genetic Algorithm", in *Proc. of the 10th Fuzzy Colloquium in Zittau*, Zittau, pp. 239-244., 2002.

[3] R. Matousek, "GAHC, Improved Genetic Algorithm", in *the Springer book series* (Eds.: Krasnogor, et al.) *Nature Inspired Cooperative Strategies for Optimization* (NICSO 2007), Volume 129, 2008, XIV, pp. 114-125., ISSN 1860-949X, Springer Berlin, 2008.

[4] R. Matousek, "Grammatical Evolution: STE criterion in Symbolic Regression Task", in *Proc. of the IAENG Int. conference WCECS 2009*, pp.1050-1054, ISBN 978-988-18210-2-7, San Francisco, 2009.

[5] P.Pivonka, V. Veleba, M. Seda, P. Osmera, R. Matousek, "The short Sampling Period in Adaptive Control", in *Proc. of the IAENG Int. conference WCECS 2009*, pp.724-729, San Francisco, 2009.

[6] P. Popela. J. Dupacova, "Melt Control: Charge Optimization via Stochastic Programming". In W. Ziemba. et al. (eds.): *Applications of Stochastic Programming*. Chapter 15. pp. 277–289 , SIAM, 2005.

[7] Z. Oplatkova, R. Senkerik, I. Zelinka, "Synthesis of Control Rule for Synthesized Chaotic System by means of Evolutionary Techniques", in *16th Int. Conference on Soft Computing*, pp.91-98, Brno, 2010.

[8] G. Cazzulani, F. Resta, F. Ripamonti, "Active Modal Tuned Mass Damper for Smart Structures", *Engineering Letters*, vol. 19, no. 4, 2011. ISSN: 1816-0948 (online version); 1816-093X (print version)

[9] P. Olranthichachat, S. Kaitwanidvilai, " Design of Optimal Robust PI Controller for Electro-Hydraulic Servo System", *Engineering Letters*, vol. 19, no. 3, 2011. ISSN: 1816-0948; 1816-093X

[10] J. Roupec, J.:, "Advanced Genetic Algorithms for Engineering Design Problems", Engineering Mechanics, vol. 17, No. 5/6, 2010.

[11] K. J. Astrom, H. Panagopoulos, and T. H¨agglund, "Design of PI ˚controllers based on non-convex optimization", *Automatica*, vol. 34(5), pp. 585–601, 1998

[12] K. J. Astrom and T. Hagglund, "The Future of PID Control," IFAC J. Control Engineering Practice, Vol. 9, pp. 1163-1175, 2001.

[13] M.J.D. Powell, "Direct search algortihms for optimization calculations", in *Acta Numerica 1998*, A. Iserles (Ed.), UK, 1998.

[14] J. G. Ziegler and N. B. Nichols, "Optimum settings for Automatic Controllers," *ASME Transactions*, vol. 64, pp. 759–768, 1942.

[15] J. Stastny, V. Skorpil: Neural Networks Algorithms Analysis in Image Processing Application, GESTS International Transactions on Computer Science and Engineering, Vol.12, No.11, 2008.

[16] G. Cazzulani, F. Resta, F. Ripamonti, " A New Algorithm with Low Complexity for Adaptive Filtering", *Engineering Letters*, vol. 18, no. 3, 2011. ISSN: 1816-0948, 1816-093X (print version)