Increased Performance of a Hybrid Optimizer for Simulation Based Controller Parameterization

Submitted 27th June 2011; accepted 12th October 2011

Reimund Neugebauer, Kevin Hipp, Arvid Hellmich, Holger Schlegel

Abstract:

The controller parameterization is often carried out by applying basic empirical formulas within an integrated automatic design. Hence, the determined settings are often insufficiently verified by the resulting system behavior. In this paper an approach for the controller parameterization by using methods of simulation based optimization is presented. This enables the user to define specific restrictions e.g. the complementary sensitivity function (CSF) to influence the dynamic behavior of the control loop. Furthermore it is possible to choose alternative optimization criteria. A main influence factor for practical offline as well as controller internal optimization methods is the execution time, which can be reduced by applying a hybrid optimization strategy. Thus, the paper presents a performance comparison between the straight global Particle-Swarm-Optimization (PSO) algorithm and the combination of the global PSO with the local optimization algorithm of Nelder-Mead (NM) to a hybrid optimizer (HO) based on examples.

Keywords: *controller parameterization, simulation based optimization, particle swarm optimization, Nelder-Mead*

1. Introduction

In the field of operations research a large number of methods were developed to support decision-making processes. It has been proven, that there is a wide field of application. In this paper a brief introduction using these methods for mechatronic controller parameterization is given with the goal of increased speed using a hybrid optimizer. In section 2 the basics of simulation optimization as well as the used optimization algorithms are stated. Subsequently in section 3 the application for controller parameterization is briefly introduced for two examples. The structure and functionality of the hybrid optimizer is the subject of section 4. A performance evaluation of the hybrid optimizer is done in section 5. The paper closes with a comparison and conclusions given in section 6.

2. Simulation based controller parameterization

Generally, simulation based optimization is a methodology of searching for the global extremum of an objective function by the coupling of a simulator with an optimizer [1]. It results in a cyclic sequence between the optimizer and the simulator.

The optimizer determines a possible solution and passes it to the simulator for evaluation. According to the result of the simulator the optimizer calculates a possible better solution. The core of the simulator is a model of the entire system which is examined. Therefore, an optimization problem (eq. 1) has to be solved [1].

$$F(\theta) \rightarrow \min_{\theta \in \Theta}(F(\theta) \qquad (1)$$

F(), called fitness function [2], is a real-valued function which represents the evaluation of the actual solution. In general, the implementation of constraints is realized by using punishment values. If a constraint is violated, a punishment value is added to the evaluation of the actual solution. Therewith it is depreciated and avoided by the optimizer. The evaluation of a solution is calculated in accordance to equation 2.

$$F(x_n) = \text{Main\_Criterion }(x_n) + \sum_j \text{Punishment\_Constraint}_j(x_n) \qquad (2)$$

Optimization techniques are divided into global and local algorithms [3]. The objective of global optimization is to find the global extremum over the entire function space. In contrast, local methods start from a defined point in the search space and try to determine a better solution. According to [4] simulation based optimization could be used to adjust controller parameters considering definable constraints.

It exits a large number of optimization algorithms for different application fields. Hereafter the PSO and the NM algorithm are described.

1.2. Particle-Swarm-Optimization

PSO is a common heuristic technique [5], which is based on the simulation of the movement of herds or swarms. An individual of a swarm is called particle. The trajectory of each particle depends on the movement of the other individuals of the swarm and random influences. The advantages of the algorithm are among other things its simple structure, no need for gradient information and its performance. The position of every particle in the -th step is described by the vector . The position of each particle in the ()-th step is update according to equations 3 and

$$x_i^{k+1} = x_i^k + \Delta x_i^{k+1} \qquad (3)$$

$$\Delta x_i^{k+1} = \omega \Delta x_i^k + c_1 r_{1,i}^k (x_i^{best,k} - x_i^k) + c_2 r_{2,i}^k (x_{swarm}^{best,k} - x_i^k) \qquad (4)$$

Where c_1, c_2 and ω are positive constants, $r^k_{1,i}, r^k_{2,i}$, and are two random values in the range [0, 1]. The term $x_i^{best,k}$ represents the best previous position of particle i till step k and $x_{swar}^{best,k}$ as the best known position among all particles in the population. Therefore, $x_i^{best,k}$ is called "simple nostalgia" because the individual tends to return to the place that most satisfied it in the past. The term $x_{swar}^{best,k}$ realizes the publicized knowledge, which also every individual tends to [6].

2.2. Nelder-Mead

The NM algorithm (or simplex method), which was originally presented in [7], uses a geometric structure, the simplex, with points in the search space with n+1 the dimension, e.g. for $n = 2$ the simplex is a triangle.

At the beginning the simplex is constructed around a committed start point. The edges of the simplex are called vertex and have to be arranged equidistant from each other. The basic principle of the algorithm is the modification of the simplex towards the extremum. In general this is achieved by replacing the worst vertex by a better one using four functions: reflect(), expand(), contract() and shrink(). A detailed description of the algorithm can be found in [8].

3. Optimization Problem

3.1. Control Loop Dynamic

Assuming the stated closed loop system structure [9], the system behavior is described with a transfer function G_S (eq. 5).

$$G_S(s) = K * \frac{1 + b_1 s + b_2 s^2 + \ldots + b_m s^m}{1 + a_1 s + a_2 s^2 + \ldots + a_n s^n} \quad (5)$$

It is supposed to use a PID controller G_R in the additive structure (eq. 6).

$$G_R(s) = K_R + \frac{K_I}{s} + K_D s \quad (6)$$

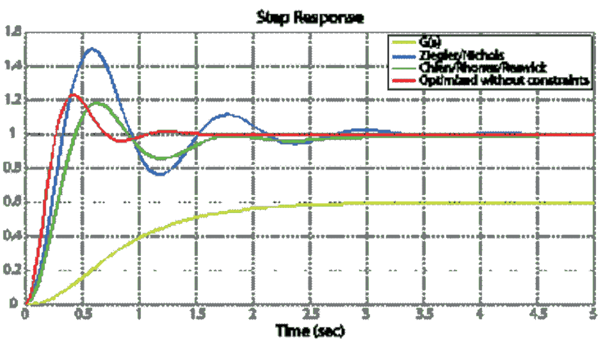

Fig. 1. Comparison of parameter settings

In Figure 1 possible attainable transition functions of a PT_3 plant (K=0.6, a_1= 0.92, a_2= 0.234, a_3=0.018) with a PID controller are shown. The main optimization criterion is the control area. No constraints were defined. The results of the optimization process are K_R= 12.327, K_I= 17.936 and K_D= 2.07.

To reduce the overshoot of the system the CSF T(s) could be used [10]. The mathematical structure is:

$$T(s) = \frac{G_R(s) * G_S(s)}{1 + G_R(s) * G_S(s)} \quad (7)$$

It allows an evaluation of the influence of changes in the command signal. By specifying the CSF it is possible to affect the dynamic of the control loop. Therefore, by setting

$$T(s) = \frac{1.1}{0.1s + 1} \quad (8)$$

the permissible amplification is limited. The new results are K_R=12.327, K_I=17.936 and K_D=2.07. As expected the overshoot is reduced (Figure 2) while the rise time increases.

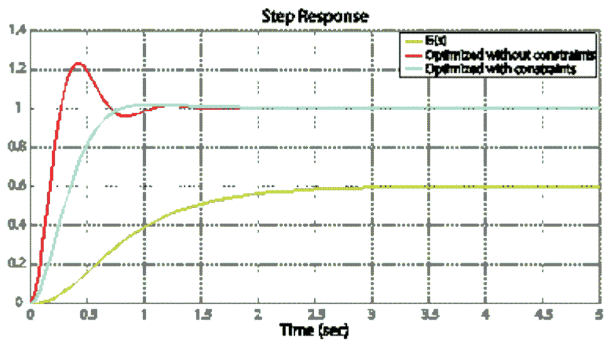

Fig. 2. Comparison of parameter settings

3.2. Energy Consumption

As an alternative to the optimization based on the CSF (section 3.1), the approach can also be used to minimize other criteria, such as the energy consumption of a servo drive. To illustrate this, a PI velocity controller according to equation 9 is used.

$$G_P(s) = K_P \cdot \left(1 + \frac{1}{T_N \cdot s}\right) \quad (9)$$

The model of the controlled system includes the closed current loop and the total moment of inertia. The parameter identification was carried out in [11] and leads to the following first order integral plus dead time system (FOIPD) (eq. 10).

$$G_S(s) = \frac{746}{(1 + 0{,}0004 \cdot s) \cdot s} \cdot e^{-s \cdot 0.00025} \quad (10)$$

To achieve a higher flexibility in formulating the quantification factors (control effort, control area, disturbance area) and the penalty (max. overshoot = 15%), the simulator was realized in MATLAB® Simulink®. Focusing on the energy consumption, the control effort was chosen as main criterion. The resulting controller parameterization as well as the results of the automatic controller tuning system, included in the servo drive system, are listed in Table 1.

Table 1. Controller parameters

Method	K_P [Nm s/rad]	T_N [ms]
Automatic Tuning (SIEMENS)	1.309	8.73
Optimizer	0.9	5.2

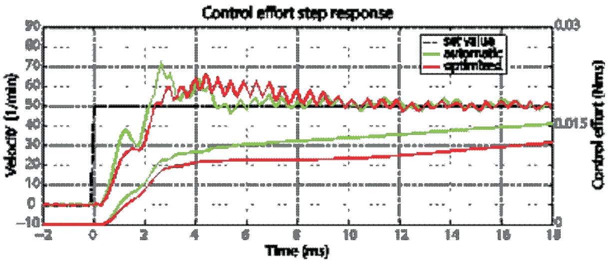

Fig. 3. Control effort step response

Figure 3 shows the resulting step responses and the integrated control effort. Notice that the overshoot reaches the predefined limit without exceeding it. Furthermore the reduction of the energy consumption is visible in the time plot, while the settling times of both variants are comparable.

4. Hybrid optimizer

In this paper the HO is a combination of the global PSO and the local NM algorithm with the objective of better performance in comparison to a standalone optimization approach.

The operation of the algorithm is the following: First the PSO is performed with a small number of calculations and then terminated. Hence, the PSO is only used for global exploration of the search space. Subsequently three instances from NM algorithm start from the best, the 3rd best and the 5th best point examined by the PSO realizing a local search. The solution of the optimization is the best result of the three NM instances (Figure 4).

The reason behind starting the local search from different points is the robustness against local extremes. It has been investigated, that if only one instance is used, the hit rate of the HO to find the extremum is reduced [12].

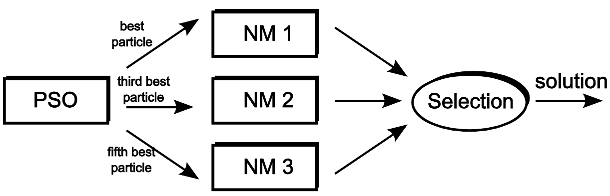

Fig. 4. Structure of hybrid optimizer

5. Performance comparison

The performance tests have been carried out using a self developed modular optimization application written in C# using VisualStudio 2010 supporting different optimization algorithms. The simulation models were implemented in MATLAB®.

Four different transfer functions (table 2) were utilized to compare the required number of calculations to determine the global extremum with a defined tolerance to the best known solution. For every transfer function the optimization was performed thirty times.

As the results in table 1 show the HO only requires 6% to 69% invocations of the simulator in contrast to the standalone PSO. This reduces the execution time of an optimization run tremendous. Furthermore the HO was always able to detect the global extremum. The reason for the high number of necessary calculations for system three can be justified with the complex shape of the search space. The gradients around the global extremum are very high and therefore the location of the extremum is very small.

Table 2. Overview of chosen test functions

Controlled System	Transfer function	Calculations	
		PSO	Hybrid
1 (PT_3)	K=1, $a_1 = 2$, $a_2 = 2$, $a_3 = 1$	1737	342
2 (PT_3)	K=1, $a_1 = 3.1$, $a_2 = 2.3$, $a_3 = 0.2$	2007	459
3 (PT_3)	K=1, $a_1 = 2$, $a_2 = 2$, $a_3 = 3$	23459	1464
4 (PT_2)	K=1, $a_1 = 3$, $a_2 = 2$	542	374

6. Conclusion

The combination of the PSO and NM to a hybrid optimizer increased the performance dramatically in comparison to the standalone PSO algorithm. The advantage of the HO is the switch from the well performing global optimization technique of the PSO to the NM, which is more effective in local exploration. Even with the three instances of the NM the required number of the calculations is still smaller.

This is essential to enable online and real time applications. But furthermore investigations in adjusting the tuning parameter of the algorithms concerning the problem of controller parameterization must be carried out. Moreover even different optimization techniques e.g. the Newton's method, genetic algorithms and different combinations to a hybrid optimizer must be investigated. Furthermore it is conceivable to use the methodology of simulation based optimization for tuning more complex systems like a controller cascade with filters.

Acknowledgements

Funded by the European Union (European Social Fund) and the Free State of Saxony.

AUTHORS
Reimund Neugebauer, Kevin Hipp*, **Arvid Hellmich, Holger Schlegel –** Chemnitz University of Technology, Faculty of Mechanical Engineering, Institute for Machine Tools and Production Processes, Reichenhainer Str. 70, 09126 Chemnitz, Germany;
e-mails: wzm@mb.tu-chemnitz.de;
kevin.hipp@mb.tu-chemnitz.de,
arvid.hellmich@mb.tu-chemnitz.de,
holger.schlegel@mb.tu-chemnitz.de

*Corresponding Author

References

[1] P. Köchel, "Simulation Optimisation: Approaches, Examples and Experiences", TU Chemnitz, 2009, ISSN 0947–5125.
[2] Y. Carson, A. Maria, "Simulation Optimization: Methods and Applications". In: *Proceedings of the 1997 Winter Simulation Conference*, 1997, pp. 118–126.
[3] E. Tekin, I. Sabuncuoglu, "Simulation optimization: A comprehensive review on theory and applications", *IIE - Transactions*, vol. 36, no. 11, 2004, p. 1067.
[4] R. Neugebauer, K. Hipp, S. Hofmann, H. Schlegel, "Application of simulation based optimization methods for the controller parameterization considering definable constraints", *Mechatronik 2011*, 2011, pp. 247–252.
[5] R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory". In: *MHS'95, Proceedings of the Sixth International Symposium*, 1995, pp. 39–43.
[6] R. Eberhart, J. Kennedy, "Particle Swarm Optimization". In: *IEEE International Conference on Neural Networks Proceedings*, 1995, 1942–1948.
[7] J. A. Nelder, R. Mead, "A Simplex Method for Function Minimization", *The Computer Journal*, vol. 7, no. 4, Jan. 1965, pp. 308–313.
[8] H. Schwefel, *Evolution and Optimum Seeking*, Wiley VCH, 1995, ISBN 0471571482.
[9] J. Lunze, *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*, 8th Edition, Springer, Berlin, 2010, ISBN 9783642138072.
[10] K. Aström, T. Hägglund, *Advanced PID Control*, ISA – The Instrumentation, Systems and Automation Society, 2006, ISBN 1556179421.
[11] S. Hofmann, "Time-Based Parameter Identification and Controller Design for Motion Control Systems". In: *Conference Proceedings 55.* IWK Ilmenau, 2010, pp. 404–415
[12] K. Hipp, *Entwurf, Implementierung und Test eines Software-Werkzeuges zur Bestimmung optimaler und robuster Regler mittels Verfahren der simulationsbasierten Optimierung*, Diploma Thesis, Chemnitz, 2010.