Causality and Design of Dynamical Systems

S. Hirche, I. Santibanez-Koref, I. Boblan
TU-Berlin; FG Bionik u. Evolutionstechnik; Ackerstr. 71-76;13355 Berlin;GERMANY

Abstract. A new kind of fitness functions for controller optimization is presented. This new fitness functions are postulated to be strong causal. Thus a better behaviour during the optimization process can be achieved.

1 Introduction

The design of dynamical systems for industrial applications can be divided in to two parts: first the identification of a plant or system and second the design of an appropriate controller.

This paper presents an approach to the controller design problem. The paper focus on the problem of parameter identification. We use an Evolution Strategy (ES) to identify the parameters [12, 14].

We implemented this approach in the SCADS system (Strong CAusality driven generation of Dynamical Systems, [13]). The main purpose of SCADS is to give a framework for automatic plant and controller design. One property of SCADS is that one basic concept behind the implementation is the use of “Strong Causality” [12] through all stages of the search process. “Strong Causality” means that small variations of the individual, i.e. dynamical system, should produce small variations of the quality.

Genetic Programming (GP) [7] is used for the structure identification of the dynamical system. The GP is adapted by implementing strong causality in the representation and in the genetic operators [11]. The ES is used to adapt the parameters of the dynamical systems. Strong causality in this algorithm means that the quality assignment to each system should be strong causal.

Another aspect of SCADS is the practical usability. That’s why we build it up using Mathworks Inc. MATLAB/SIMULINK. MATLAB offers wide variety of tools for the design and identification of dynamical systems. Specially a number of toolboxes offers well proven applications and tailored methods for the simulation of dynamical systems.

In order to use SCADS, we have to formulate an appropriate description of dynamical systems. That means that the description should be evolvable and be able to describe a wide variety of dynamical systems. We choose a presentation as a directed graph [11]. This special representation as a graph guarantees that it is possible to create closed loops in the dynamical systems, thus it is possible to generate the most common structures for dynamical systems. The nodes represent the standard building blocks for dynamical systems (e.g. P-,PD- ,PID-blocks etc. [2]). The vertices represents the signal flow between this blocks. One individual is a dynamical system, represented as a graph. That means, that we are evolving graphs.

As an important feature it is necessary to assign to each individual a quality or fitness. In our case quality means: How well does the dynamical system fulfill the required task. In case of system identification, how well does it reproduce a given behavior. In the case of controller design: how well does the system control a given plant. In order to assign a suited quality the user has to provide a calculation procedure, which involves the simulation of the dynamical system by a integration algorithm. As we use standard representation for functional blocks, we have to take in to account that the blocks are parametrized. Thus not only the structure of the dynamical system is important, also the parameters of the blocks are significant for the resultant system behavior. We have to choose the appropriate parameter settings in order to decide what is a good system. Thus it is necessary to optimize the parameters in order that the quality criterion is fulfilled as best as possible. That means that besides a simple simulation of the dynamical system is it necessary to optimize the parameters of this system in order to determine the quality of the system.

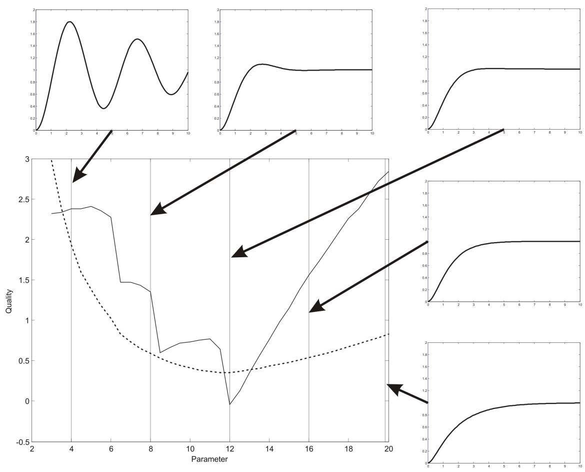


Figure 1: Comparison between a Strong Causal (doted line) and a non Strong Causal quality function for simple control system.

Providing the quality information it is easy to select the best individuals. The first choice would be to use truncation selection or tournament selection. But taking in to account that for each quality assignation a simulations are needed, we decide to choose another selection structure. For selection we take a certain number of individuals from the population, select the best from this group and replace the worst individuals by new generated individuals (called partial tournament selection). Thus we can guarantee that we have not a generational EA. That guarantees a better usage of available parallel resources. Using this approach for selection we can expect an efficient implementation of the EA. That is why we implement the quality assignment on independent, asynchronous computational slaves, which guarantee a linear speed up of the execution time.

The next problem is how to define a strong causal quality function. This quality is decisive for the efficiency of the whole algorithm. Figure 1 presents two functions for quality assignment for controlled system, in which the controller depends on one parameter. The continuous line represent a quality function that tends to have big value changes with small parameter changes. The function is multimodal and hard to optimize for a wide variety of optimization algorithms. The second function depicted in figure 1 is a so called strong causal function (dotted line). We can see that the function is smoother, thus small changes in the parameter gives also small changes in the quality. The second property of this function is that it is unimodal, thus it can be optimized very easy.

In this paper we describe one variant of the quality assignment for dynamical systems in SCADS.

The structure of the paper is as follows: In the next section we give an overview over the classical quality criteria for controllers. In section 3 we present the new controller criteria. Some results obtained with this criteria are described in section 4. Conclusions are drawn in section 5.

2 Quality Criteria for Controller Optimization

Most quality criterions known from the practice of control system design, e.g. the characteristic value criterion [15] and the integral criterion value the transient response characteristic to a test input signal in the time domain as shown in Figure 2. Here $r(t)$ stands for the reference input, $e(t)$ is the error signal, $u(t)$ symbolises the correcting variable and $y(t)$ is the output signal of the system.

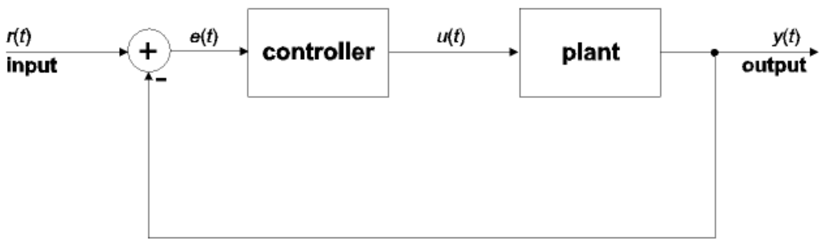


Figure 2: Basic control system with unity feedback

Fortunately the performance indices defined in time-domain [4] can easily be measured. They result directly from the application and can be considered as standard. Usually a step input is used for reference input [2].

- The swiftness of the response is defined by the rise time t_r and the peak time t_p. For an over damped system the peak time is not defined and the 10%-90% rise time t_r is normally used (see [2]).
- The similarity of the response $y(t)$ to the step input $r(t)$ is represented by the peak value or overshoot M_p and the settling time t_s. The settling time t_s is defined as the time required for the system to settle within a certain percentage of the input amplitude (see [2]).

In the characteristic value criterion the interesting performance indices are summarized. This method provides a transparent correlation between the transient response characteristic and the fitness function. But the fitness landscape turns out to have ridges. That reduces the convergence speed or even worse, the optimisation algorithm can get stuck there. Therefore this criterion is not suitable for numerical optimisation [10].

The integral criterion provides smooth fitness landscapes by valuating the quality using the weighted squared error that is evaluated between a reference function and the transient response to a test input signal.

$$Q_I = \int_0^T f\left(e(T), u(t), y(t), t\right) \mathrm{d}t$$

So in this criterion the location of the optimum can be influenced (moved) by means of the reference function choice, the time weighting and the weighting of the correcting variable. As reference signal a unit step input is often used. A widely applied formulation of the general integral criterion is the integral of time multiplied by the squared error criterion (ITSE):

$$Q_I = \int_0^T t^n \left(e^2(T) + \alpha u(t)\right) \mathrm{d}t$$

The time weighting causes a higher penalty the later a deviation from the reference signal occurs. The effect is that the optimisation leads to more stable systems, at least within the time horizon. The squared correcting variable $u(t)$ is multiplied with a scalar weight α In theoretical developments the problem of the correcting variable is often ignored and without importance. In real-world problems though the limitation of the correcting variable must be taken into consideration.

This approach has two main disadvantages. Firstly, the solutions are often not definite [10] and meet the performance requirements often inaccurately, since the aspired aim is not to optimise towards a desired characteristic value but to minimise the squared error. Additional penalty terms for certain characteristic values do not provide appropriate results either [1]. A simple way to transfer the performance indices into the criterion is by comparing the output signal with a reference signal that meets the desired performance requirements, e.g. with PT2-behaviour. Consequently the optimal control system in terms of such a performance measure never behaves better than the compared PT2-system. This is an unacceptable restriction.

Secondly, as the correcting variable is minimised in relation to the signal error, thereby the solutions are often conservative. In most cases this formulation does not reflect the situation of real-world problems, where the correcting variable is bounded and therefore set to the maximum in order to speed up the transient response.

Subsequently a quality criterion is designed that combines the advantages of both performance measures: the implementation of the time-domain constraints and a satisfying smooth fitness landscape without ridges.

3 Alternative quality criterion

The basic idea is, to divide the general form of the integral criterion into signal value orientated time sections [1]. We call it piecewise integral of time multiplied by the squared error criterion or short PITSE. By using the characteristic values an indirect avoidance of parameter ambiguity can be reached. Each section represents one characteristic value of the step response with the magnitude r. The more values need to be considered the more sections have to be created. In most cases four characteristic values thus sections prove to be sufficient. Some of them are signal value related (M_p,M_U,e_{ss}) others time related (t_r, t_s). In order to exert a comparable selection pressure by each of the time sections the dimensions of them must match. All sections are based on the integration of a squared error. The differences can be found in the measurement of the error and the integration boundaries. Another point regarding the optimisation needs to be taken into account. The start vector of the control parameter is often set to 1 and accordingly leads to a system response that never lies in the near surrounding of the desired one. In order to achieve a flow transition from far to near optimisation the objective function must meet the demand for Strong Causality as mentioned above. Therefore high values are assigned to each section if the solution is still far from the optimum, so in terms of control systems if it is unstable. The objective function is not changed in the course of approaching the optimal solution. The fitness function consists of the following sections:

Undershoot section: Preventing an undershoot is often impossible but minimising is required. The difference $e_{mu}(t)$ between the signal and the boundary value for undershoot M_{u_req} is squared and integrated over the entire time horizon $[0, T]$:

$$q_1 = \int_0^T e_{mu}^2(t)\mathrm{d}t \qquad e_{mu}(t) = \begin{cases} 0 & \text{for } y(t) > M_{u\_\mathrm{req}} \\ y(t) - M_{u\_\mathrm{req}} & \text{otherwise} \end{cases}$$

As soon as the system gets unstable, the part q_1 can reach high value. Thereby this section supports the movement towards stable systems.

Rise time section: In control system design the requirements for the rise time and the overshoot often turn out to be the most important aspects after the demand for robust stability. Again a squared error $e_{tr}(t)$ is integrated within the boundaries $(t_{r\_\mathrm{req}}, t_1)$ where t_{r_req} is the required rise time and the value of $t_1$depends on the step response characteristic:

$$q_2 = \int_{t_{r\_\mathrm{req}}}^{t_1} e_{tr}^2(t)\mathrm{d}t \quad e_{tr}(t) = \begin{cases} 0 & \text{for } t\,|_{y=r} \le t_{u\_\mathrm{req}} \\ r(t) - y(t) & \text{otherwise} \end{cases}$$
$$t_1 = \begin{cases} t\,|_{y=r} & \text{for } y(t) \ge r \\ T & \text{otherwise} \end{cases}$$

The error $e_{tr}(t)$ is defined by the difference between the amplitude of the step input $r(t)$ and the signal $y(t)$. If the signal does not cross the zero line within the time horizon, a rise time is not defined at all. By double counting the undershoot area a solution far from the optimum is highly penalized in order to increase the selection pressure and therefore speed up the searching. Additionally a smooth transition from far to near optimisation is reached. In Figure 3 several scenarios for the error evaluation of the rise time section are shown.

Overshoot section: In correspondence to the undershoot section the overshoot section is designed. The squared error $e_{M_p}(t)$ is defined as the difference between the required maximum of overshoot M_{p_req} and the signal $y(t)$) within the time horizon $[0, T]$ is integrated:

$$q_3 = \int_0^T e_{M_p}^2(t)\mathrm{d}t \qquad e_{M_p}(t) = \begin{cases} 0 & \text{for } y(t) < M_{p\_\mathrm{req}} \\ M_{p\_\mathrm{req}} - y(t) & \text{otherwise} \end{cases}$$

Steady state error section: The steady state error and therefore the static system behaviour is valued by this part of the fitness function. If the signal leaves the ε-tube after the required settling time t_s then the difference $e_{ts}(t)$ between the ε-tube and the step response

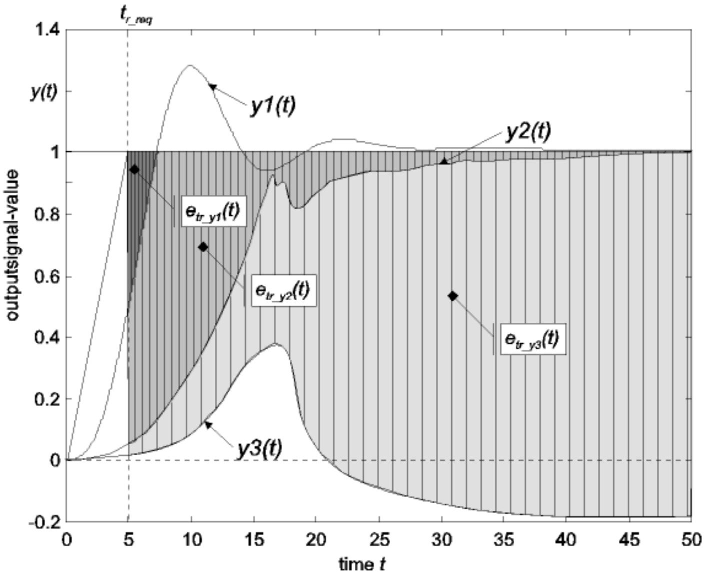


Figure 3: Scenarios for the evaluation of the rise time error $e_{tr}(t)$

is squared, time weighted and integrated:

$$q_4 = \int_0^T e_{ts}^2(t)\mathrm{d}t \qquad e_{ts}(t) = \begin{cases} 0 & \text{for } -\epsilon \leq \frac{y(t)}{r(t)} - 1 \leq \epsilon \\ y(t) - t(t) \cdot (1+\varepsilon) & \text{for } y(t) > r(t) \cdot (1+\epsilon) \\ y(t) - t(t) \cdot (1-\varepsilon) & \text{otherwise} \end{cases}$$

Systems with an unstable step response are highly penalized. The time weighting indirectly leads to stable systems. By means of the time weighting this part of the fitness function becomes the main power exerting selection pressure far from the optimum. Additionally the triple evaluation of under-, overshoot and steady state error in case of unstable solutions accelerates the far optimisation.

Correcting variable term: In real-world problems the value of the correcting variable is physically bounded within a lower value u_{low} and an upper value u_{up}. As long as the correcting variable remains within these boundaries this part maintains the value 0. Whenever the correcting variable crosses a boundary, an error function $e_u(t)$ is defined. Consistent with the evaluation of the other sections the error is squared and integrated:

$$q_5 = \int_0^T e_u^2(t)\mathrm{d}t \qquad e_u(t) = \begin{cases} 0 & \text{for } u_{low} \leq y(t) \leq u_{up} \\ u(t) - u_{up} & \text{for } u(t) > u_{up} \\ u_{low} - u(t) & \text{otherwise} \end{cases}$$

Unstable systems tend to have high values of the correcting variables; therefore unstable systems get penalized by this term, too.

There exist several possibilities to combine these sections to a scalar fitness value [3]. In the following thequality is presented by the sum of the weighted section values q_i:

$$Q_{PITSE} = \sum_{i=1}^{5} \alpha_i \cdot q_i$$

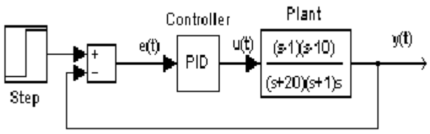
The advantage of this method compared to others is its efficiency. As weakness appears the difficulty to find the appropriate weights α_i in order to reach a desired solution. Therefore an iterative adjustment process of the weights in the course of optimisation turns out to be necessary. Some of these sections are in concurrence to each other, e.g. the rise time section versus the overshoot section.In order to gain a lower rise time an increase of overshoot must be accepted . A compromise must be obtained which leads to the problem of vector or multi-objective optimisation [3]. The weighting can influence the balance of the trade-off. Unlucky weighting can distort the relationship between the sections and thereby it can cause the getting stuck of the ES in a local optimum. In most cases satisfying results are achieved by setting all weights to 1 in the new fitness function. In case that all performance requirements are met, the fitness function obtains the value 0, which is a terminating, condition for the ES. In order to continue the optimisation the required characteristic values must be set to a lower value. Even if a well balanced selection pressure concerning each penalty term thus characteristic value was aspired, several simulations have shown that the rise time term exerts the highest selection pressure followed by overshoot-, undershoot term, the correcting variable term and last the steady state error term. The optimum step response tends to meet the required rise time rather then the required overshoot or the required settling time with all weights set to 1. But this depends on the system and the length of simulation, too. In order to lay the emphasis of optimisation to these characteristic values, the weighting must be changed. The optimisation of a system with unknown behaviour is an iterative process. The new criterion guarantees a steady selection pressure over the whole variety of output signals. This is absolutely necessary, because if the ES finds a local gap, the convergence speed decreases or the algorithm even gets stuck in multimodality.

4 Simulation results

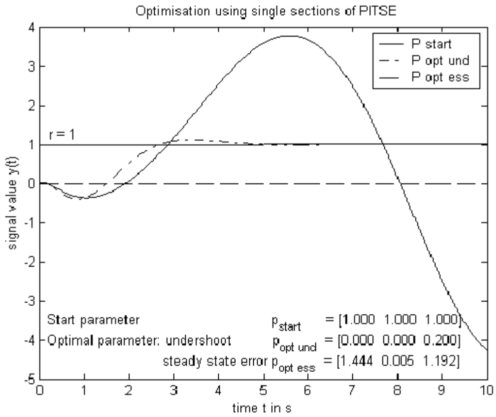
In the following simulations the behaviour of new quality criterion is studied. The results of the optimisation with ITSE and PITSE are compared. All simulations are accomplished in the same environment under MATLAB/SIMULINK with the stiff variable step solver ‘ode23tb’ [9]. For the optimisation the ES-CMA (3,10) [5] with the start parameter and the start step size of 1 is used. First of all the behaviour of the single sections in optimisation is studied separately on a test system shown in Figure 4 a). Therefore all weights are set to zero except the one for the studied part of the quality function, which is set to one. With the start parameter settings the unit step response has unstable behaviour with undershoot as shown in Figure 4 b) (solid line). The parameters of the PID-Controller are given in the K-normal form [4]. Different test cases are regarded:

Undershoot section: Now the unit step response is required to have no undershoot, $M_{U\_\mathrm{req}} = 0$. In the test system there is no reasonable possibility of preventing an undershoot. The only solution to obtain the desired performance is the theoretical one by setting all parameters to zero causing a system output of zero magnitude. Surely this is of no practical use, but shows the effect of the undershoot section. After 60 generations the terminating condition for the quality function value of $Q = 10^{-10}$ is reached. The optimal parameter setting leads to an output that is close to zero shown in Figure 4 b) (dashed line). The signal value remains within the interval of $[-10^{-4}, 10^{-4}]$.

Steady state error section: Now the unit step response is required to remain within the 2%-band of the desired value $r = 1$ after the required settling time $t_{s\_\mathrm{req}} = 5s$. The terminating



(a) Test system with undershoot.



(b) Step responses with optimal parameters (undershoot (–), steady state error (-.)).

Figure 4: Test system with undershoot

condition for the fitness value is reached after 60 generations. The optimal control system step response shown in Figure 4 b) (dash-dotted line) does not leave the 2%-band after the desired settling time.

Rise time section: The optimal parameter setting for the required rise time of t_{r_req}=2s is reached after 5 generations. The generated system is unstable, but the rise ids below the requirementsdesired rise time .

Overshoot section: After 6 generations the optimisation is terminated with a fitness value $Q < 10^{-10}$. The step response with the optimal parameter setting has an overshoot smaller than the required $M_{p\_\mathrm{req}} = 1.2$.

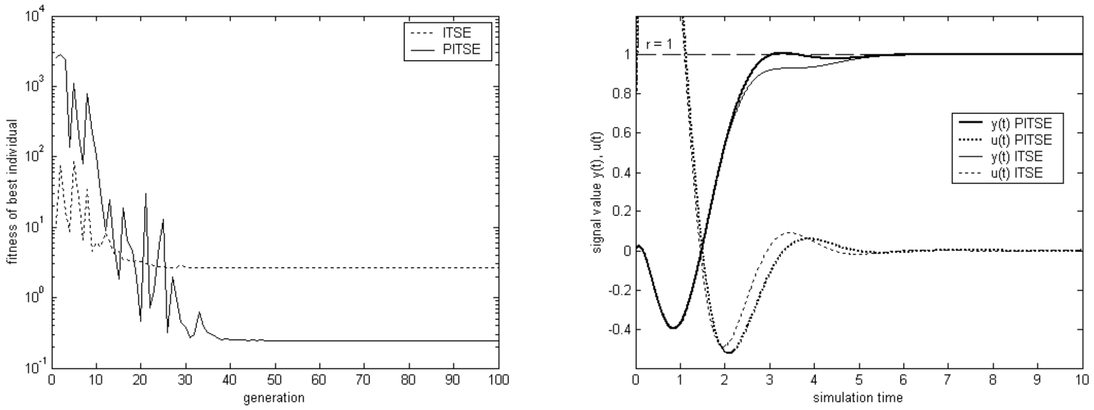
Now it is shown that each error section optimises towards the required characteristic value. In the following simulations the single error sections are combined to a fitness value as proposed in the previous section. The results of the optimisation with PITSE and ITSE are compared. The goal is to obtain an optimal step response with acceptable undershoot $M_{u\_\mathrm{req}} = -0.2$, rise time $t_{r\_\mathrm{req}} = 2s$, no overshoot $M_{p\_\mathrm{req}} = 1$ and settling within the 1%-band after the settling time $t_{s\_\mathrm{req}} = 5s$. The correcting variable value should remain within the boundaries of [-5, +5]. In Table 1 the results for the best of 10 runs are presented. In the third column the optimal parameters are listed, in the forth one the fitness value evaluated by the PITSE-criterion. The subsequent columns show the performance indices of the step response. In the last column the number of generations is listed when the fitness value enters the +%5-band of the fitness value in the optimum. This parameter is used to indicate the convergence speed of the optimisation algorithm.

The optimisation by ITSE with the correcting variable weight set to $\alpha = 0.1$ creates a system with a rather conservative step response expressed by a high rise time, little overshoot and a low settling time. The emphasis lays on reducing the error of the latter part of the signal. The first runs with the PITSE-criterion with all weights set to one show very low rise time, caused by a higher correcting variable value. The performance concerning the undershoot is unacceptable though (2. row). In the following runs the weight for the undershoot section is set to 10 in order to gain lower undershoot (3. row). The undershoot is reduced to the level of optimisation with ITSE, but the signal does not settle within the simulation time. Therefore in the next optimisation (4. row), the steady state error section is weighted by $\alpha_4 = 100$. Now the performance of the system seems acceptable, even if the undershoot is still pretty high. The undershoot cannot be reduced without a trade-off concerning the rise time:

Table 1: ITSE and PITSE with variable weighting

	Weights	PID	Q	M_u	t_r	M_p	t_s	$\lvert u_{max} \rvert$	Gen
ITSE	α_1= 0.1	[1.334 0.003 1.333]	0.161	-0.391	6.040	1.005	5.248	2.689	29
PITSE	α_1= 1	[1.474 -0.003 1.504]	0.046	-0.480	2.400	1.022	7.067	3.307	57
PITSE	α_1= 10	[1.394 -0.006 1.293]	0.109	-0.402	3.098	1.002	> 10	2.760	38
PITSE	$\alpha_1 = 10$, $\alpha_4 = 100$	[1.388 0.001 1.286]	0.113	-0.400	3.028	1.009	5.140	2.743	39

The last column of Table 1 and the evolution of the fitness presented in Figure 5 a) indicate faster optimisation by means of the old integral criterion. The ITSE-criterion provides a smooth fitness landscape, whereas multi-objective fitness functions usually create multi-modal fitness landscapes that lead to lower convergence speed. From the step responses in Figure 5 b) it can be seen, that the faster response of the PITSE-optimal system results from the higher level of $|u(t)|$ in the rise time section. Here the conservative character of optimisation with the ITSE-criterion becomes obvious.



(a) Optimisation with ITSE and PITSE.

(b) Optimal Step responses of $y(t)$ and $u(t)$.

Figure 5: Optimization results of the different criteria.

5 Conclusions

The principal contribution of this paper is the introduction of an alternative quality function for parameter optimisation with Evolution Strategies, combining the advantages of the ITSE criterion and the characteristic value criterion. The integral criterion is divided into error sections that are adjusted on the characteristic values of the signal in the time domain. A smooth fitness landscape is achieved. Time-domain based, user-specified performance requirements are implemented and determine the location of the optimum in the parameter space. By means of weighting the priority of reaching certain performance specifications in the optimum can be defined. The new criterion is applicable to all control system design problems with time domain based performance requirements.

References

[1] I. Boblan, I. Santibáñez Koref, A. Schütte: ”A New Integral Criterion for Parameter Optimisation of Dynamic Systems with Evolution Strategy”. VDI Berichte Nr. 1526 pp. 143 - 150. 2000

[2] R.C. Dorf, R.H. Bishop: “Modern Control Systems”. Prentice Hall. 2000

[3] C.A. Coello Coello: “A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques”. 1998

[4] O. Föllinger; F. Dörrscheidt, et al.: “Regelungstechnik: Einf. in d. Methoden u. ihre Anwendung.” Heidelberg, Hüthig-Verlag. 1994

[5] N. Hansen and A. Ostermeier: “Convergence Properties of Evolution Strategies with the Derandomized Covariance Matrix Adaptation: The $(\mu/\mu_I, \lambda)$-CMA-ES”. 5th Europ. Congr.on Intelligent Techniques and Soft Computing**:** S. 650 - 654. 1997

[6] S. Hirche: “Strong Causal Quality Functions for automatic Controller Design”, Beiträge zur Bionik und Evolutionstechnik 02/2001, Technical University Berlin, ISSN 1618-8675, Berlin. 2001

[7] J.R.Koza: “Genetic Programming: On the Programming of Computers by Means of Natural Selection”. MIT Press. 1992

[8] J.R. Koza et al.: “Automatic creation of human-competitive programs and controllers by means of genetic programming”. Genetic Programming and Evolvable Machines. 1 (1 -2) pp. 121 - 164. 2000

[9] MathWorks Inc.: “SIMULINK : dynamic system simulation software”. Natick, Mass., MathWorks Inc. . 1994

[10] M. Meyer: “Parameteroptimierung dynamischer Systeme mit der Evolutionsstrategie”. Berlin, TU. Berlin. 1998

[11] J. Niehaus, W. Banzhaf: “Adaption of operator probabilities in genetic programming”. In Julian F. Miller, Marco Tomassini, Pier Luca Lanzi, Conor Ryan, Andrea G. B. Tettamanzi, and William B. Langdon, ed., Genetic Programming, Proceedings of EuroGP’2001, volume 2038 of LNCS, pp. 325–336. Springer-Verlag. 2000

[12] I. Rechenberg: “Evolutionsstrategie 1994”. Fromann Holzbog. Stuttgart. 1994

[13] F. Lohnert, A. Schütte, J. Sprave, I. Rechenberg, I. Boblan, U. Raab; I. Santibanez Koref, W. Banzhaf; R. Keller; J. Niehaus, H. Rauhe: “Genetisches Programmieren für Modellierung und Regelung dynamischer Systeme - GEPROG -”. Beiträge zur Bionik und Evolutionstechnik 01/2001, Technical University Berlin, ISSN 1618-8675, Berlin. 2001

[14] H.-P. Schwefel: “Evolution and optimum seeking”. New York, Wiley. 1995

[15] H. Unbehauen: “Regelungstechnik I, Klassische Verfahren zur Analyse und Synthese linearer kontinuierlicher Regelsysteme”. Vieweg Verlag. 1997