

Intelligent Design Environment for Second-Order Positioning Systems

S. Dale, H. Silaghi, D. Zmaranda, U. Rohde

**Sanda Dale, Helga Silaghi,
Doina Zmaranda***

University of Oradea
Romania, 410087 Oradea, Universitatii, 1
sdale@uoradea.ro, hsilaghi@uoradea.ro
*Corresponding author: dzmaranda@uoradea.ro

Ulrich Rohde

TUniversity of Technology Cottbus
Germany, 03048 Cottbus, Lipezker St. 47
ulrich.rohde@tu-cottbus.de

Abstract: From the designer perspective, in real-time applications as mechatronic systems - the control algorithm has to be easy to understand and to implement, easy to optimize and last but not least, to ensure the best possible behavior for the controlled system. The paper presents a design environment which provides the possibility to choose between a classical control algorithm (as state feedback stabilization represents) and a new approach (interpolative controllers) with the possibility to modify the designed parameters such as to obtain the optimum behavior for the controlled system. This opportunity is given through the inner structure and operating laws of the algorithm and exploited using genetic algorithm-based techniques. Interpolative-type controller category covers the controllers based on fuzzy, neural and pure interpolative algorithms, due to their common capability to make an approximate reasoning. The first ones, fuzzy and neural algorithms, are already well-known in the control area. A pure interpolative controller presumes to contain at least one block, placed no matter where in his structure, in which interpolation as mathematical operation to have place. The approach presented in the present paper uses this kind of blocks to reproduce and to optimize the behavior of an already existing controller (in this case the state feedback one) [1]. This type of controllers meets all the requirements stated at the beginning: they are easy to implement, easy to understand, they need reduced calculus time and gives notable results in specific cases. The above mentioned kind of controllers will be presented with all the necessary details in the paper. The design environment is presented as a collection of MATLAB functions, SIMULINK configurable schemes and a user interface. A case study ends the presentation with some experimental results obtained through a simulation for a specific second order positioning system.

Keywords: interpolative control algorithms, intelligent optimization, design environment, mechatronic positioning system.

1 Introduction

A mechatronic system is generally an electromechanical system equipped with sensors which detect the state of the environment, actuators which modify the state of the environment and a control system which controls the actuators based on the environment, as described by the sensors. Mechatronic systems are implemented in all kind of man-made machines as autonomous robots, industrial robots or flexible manufacturing cells.

In many mechatronic applications, the control functions are implemented through a microcontroller. In the past microprocessors and microcontrollers have been used especially by the

electronic and automation engineers (due to the difficulties in design, programming, testing and usage and due to the fact that the amount of needed information for the applications was very large and the given degree of software and hardware integration), today the microcontroller can be used even by people that have minimum knowledge of electronics and programming. There are many brands that offers developing systems with integrated microcontrollers at good prices, well documented and with numerous applications in different fields. Some of them have an active policy in attracting the non-specialists in realization of applications with microcontrollers. The main advantage of such an evolution is the possibility to realize, in a relative easy way a sum of practical applications in various physical areas. In this context there isnt surprising the increasing amount of applications from the mechatronics field based on the usage of microcontrollers which implements various control techniques.

The actual approaching manner of such applications, focused on miniaturization, has stimulated the efforts in this direction. Nowadays, the microcontrollers have better and better performances (memory, speed etc). There is a need to continuous increase of the performances for those products, leading to a larger volume of software applications. Testing of the applications becomes more important. The usage of microcontrollers in mechatronical applications leads to the appearance of new directions of research in the field. Integrated systems based on microcontrollers are low cost solutions and have the advantage that can be used in applications where miniaturization (from the mass, volume, energy consumption point of view) is an important requirement. As a consequence, the researches on the implementation possibilities for control algorithms in applications from mechatronics as well as motion control, mini-robotics etc., constitutes a necessity. The main disadvantage of using the microcontroller in such applications (comparative with PLC's) its related to the potential difficulties in software development. In this context, development of performant control algorithms easy to use and to implement contributes to reduce and eventually eliminate this inconvenient: among them, some of those knew as intelligent control algorithms.

As it is well known, intelligent control is a class of control techniques, that use various AI computing approaches like Bayesian probability, approximate reasoning techniques based on expert knowledge or interpolative-type algorithms (as neural networks, fuzzy logic, pure interpolative algorithms), machine learning, evolutionary computation and genetic algorithms. In this branch of control engineering, the control techniques are extended beyond classical mathematical methods based on integral-differential calculus. Difficulties that appear in complex non-linear systems control with uncertain models can be eliminated by new approaches based on an operator experience. Hence, one way to develop control algorithms is to generate them based on expert knowledge. In that way, they are easier to understand for humans, and this can be considered as the second major requirement for the control algorithms used in mechatronic devices design. Intelligent control systems must treat information referring their own state and the environment state and that implies both heuristic and algorithmic programming methods. Intelligent control implies also a decisional strategy based on synthetic thesis from system theory, advanced mathematical modeling, computers and knowledge based linguistic methods, as in [2] and [7].

From both perspectives exposed above - which can be expressed as two major desiderates or requirements for the control algorithms implemented on microchips and designated to mechatronic devices there are a few categories of appropriate intelligent algorithms. Fuzzy and neural ones, due to their complexity, are not one of them. In [3] and [4] some comparative experimental studies was made from the implementation on microcontroller point of view, between PID, fuzzy and interpolative algorithms. The results allow to consider pure interpolative algorithms implemented as interpolative blocks and used in control structures as candidates for implementation on microcontrollers. One of the advantages of using such blocks are, mainly, based on simplifying the solutions (easier implementation and reduced calculus time, meaning the first requirement

is fulfilled) and the possibility to ensure for the control systems, in a relative easy way, some robustness properties [5]. In the same time, as in [4], [6] and [9], the controllers based on interpolation table, on the same form as in Drechsels RIP (Regelbasierte Interpolation) controllers introduced in [12], can be developed in much more situations than those based on linguistic rules. They can be conceived starting from solutions deducted in different ways, solutions that, finally, can be improved. From this perspective, interpolative algorithms meet the second requirement also, the possibility to be understood and designed by automation engineers or non-specialists with no loses from the control systems performances point of view.

The results achieved in this field in the last years authorise the answer at two major questions. Why an interpolative algorithm? Because these algorithms belong to the intelligent control techniques category, being one of the new-generation performant algorithms. On the other hand, due to the internal structure based on interpolation they are improvable, very simple to understand, use and implement, making them a good choice for the implementation on microcontrollers by specialist either non-specialist. In this way, the interpolative algorithms can present interes for the mechatronic applications field. Why a GA-based optimization technique? Because it seems to be very appropriate for optimizing the performances of the control structure with interpolative controllers, due to the fact that they represent themselves improvement methods based on natural and genetical mechanisms, which combine artificial surviving of best individuals with specific natural operators. Some procedural aspects of an intelligent design environment which combines interpolative control with GA-based optimization are presented in [1].

For the intelligibility of the presentation, in the second chapter interpolative algorithms will be presented from a large perspective, in a descriptive-conceptual manner. The functionality and the structure of the proposed design environment is described in the third chapter that ends with a case study made on a second order positioning system. Some relevant conclusions resume the presentation.

2 Design procedures for controllers with interpolative blocks optimized through GA-based algorithms

The controllers with interpolative blocks as introduced in [5] and [9] represents an alternative to the controllers obtained by various design procedures, that are replaced with controllers containing interpolation tables and algorithms and participate to the partial or integral control law elaboration. The idea of using interpolative blocks, named also table-type blocks its not a new one. They are frequently used in real time programs in order to reduce the calculus time in case of operating with multivariable complex functions. The interpolation technique is usual in robotic control systems.

The concept of controller with interpolative blocks brings as a new element the elaboration method of such a controller which can be implemented quite easy using digital techniques. These techniques allow the real time operation with complex tables. The concept maintains actual the design methods for the linear or non-linear control algorithms, whose results constitute the starting point for interpolative controller synthesis and at the same time improves their performances.

Those controllers are synthesized based on a very well known principle in the quality systems domain: continuous improvement of the activities flux of an already existing system, designated to ensure a specific quality function by which are expressed the system performances [10]. For the interpolative controllers this principle is translated as follows: to obtain, for a certain control system, a better controller than the initial one, designed through usual methods. The synthesis of an interpolation block (implemented by an interpolation table) is accomplished in two stages

(as in Figure 1):

- Primary synthesis stage necessitates other two steps: initial control algorithm synthesis and interpolation table construction based on the command hyper-surface of the initial control algorithm
- Correction stage consists in one or more steps of on-line or off-line correction, operated on the interpolation table

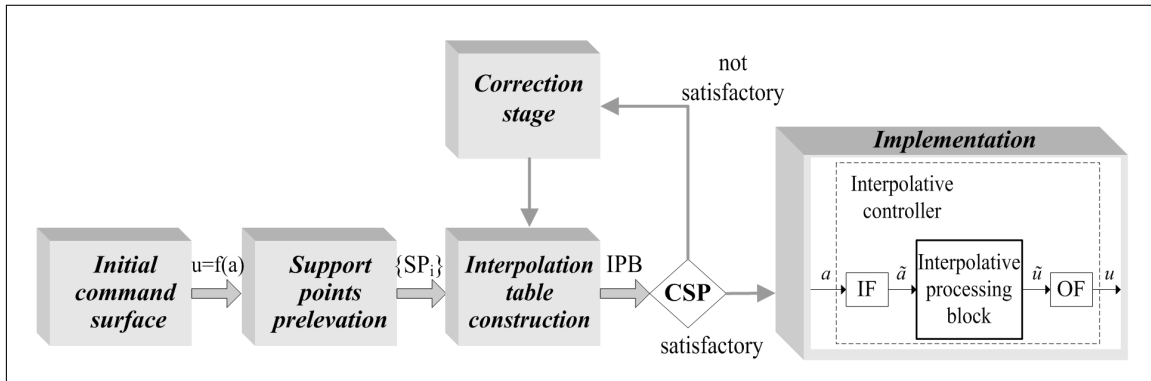


Figure 1: Interpolative controller synthesis

The interpolation table is determined through support points extraction from the initial controller hyper-surface. Support points set must be structured as a Cartesian grid and has to cover all variations domain for the input variables. The support points ensemble has to embed, from a qualitative point of view, all the inflection points of the command hyper-surface. Support points generation is made point by point or based on an analytical function, when such a function is available. If the locations number on the table is too large, the support points extraction can be incomplete; the empty spaces will be filled by Shepard interpolation.

Once the interpolation table was obtained describing approximatively the behavior of the initial block it will replace the initial block, followed by a correction stage. The correction is realized by successive modifications operated on interpolation table with a specific aim: to improve the performance of the control system. The effective correction can be made by modification on the values in table locations and/or by granularity modification on the inputs and/or output and it is an iterative process.

At principle level, an interpolative controller contains an interpolative processing block (IPB), usually a non-dynamic non-linear block, implementing the relation in (1).

$$u = Int(a_1, \dots, a_n) \quad (1)$$

The input and output filters (IF and OF respectively) ensure together the controller dynamics and they adapt the inputs at the IPB and the outputs to the execution element connected after the controller. Control system performances (CSP) can be improved when an improvement criterion is chosen as in [10] or they can be optimized if an optimization, well defined criterion is considered, as in [6] and [8], where GA are used for a so-called *intelligent optimization*.

An IPB implementation is a digital one and its realized through n-dimensional interpolation tables, made in soft manner (programs capable to execute the operations corresponding to the association of table locations for input a and the calculus of command c based on an interpolation algorithm between the table values) or hard (using PLCs, microcontrollers or dSPACE modules). All implementation modalities offer the possibility of real-time operating.

Interpolative controller design has, starting from the correction stage, a heuristic character, mostly empirical. This fact represents in many cases an advantage: brings to the light the creative, innovative, intuitive side of the designer, without forcing him to manipulate a complex mathematical instrument, on models affected by uncertainties.

In the present paper the interpolative blocks will be synthesized through the method exposed above, as in [5] and [9], followed by off-line GA-based correction, as in [6] and [4]. GA-based techniques seems very appropriate in such cases due to the fact that GAs represent themselves improvement methods based on natural and genetic selection, compare and sorting mechanisms, which combine artificial surviving of best individuals with specific natural operators and form a searching mechanism, usable in various purposes.

For this kind of application optimization of an interpolation control table - a GA modifies a table population T_i which implements the control block for the interpolative structure (look-up table block). The starting point represents an initial set made from interpolation tables $T_{initial}$ capable to ensure command characteristics. During the synthesis operation the performances of the interpolative control systems are permanently evaluated by simulations made with the SIMULINK control schemes, for preset scenarios (eg. unit step response). A fitness function is used to provide a measure of performances related to the time requirements imposed to the control system. The adopted objective function induces the desired performances by using appropriate restrictions associated to the control system's response.

Through techniques which are specific to GA, at every generation are preserved a rising number of performant individuals, where the individual performance is associated with a rise in the system performance. At the time when the convergence criterion (CC) - consisting on minimization of the error between an imposed trajectory and the current trajectory of the system - is accomplished, the best individual is extracted and will be used in the effective implementation, being considered the optimal solution, regarded from the used criterion point of view. The entire process can be synthesized by the schema in Figure 2.

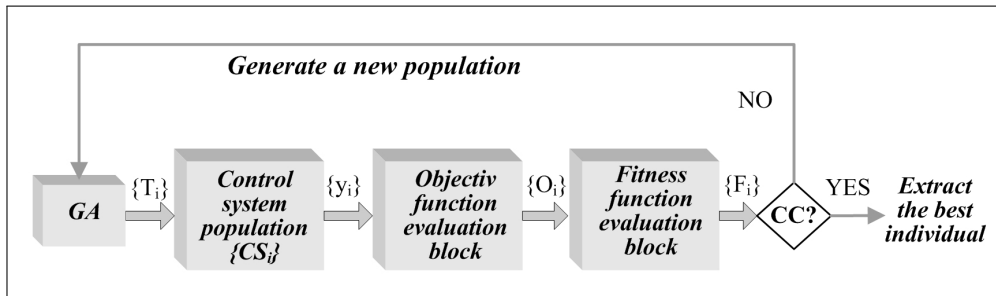


Figure 2: GA-based optimization

The main limitations of the interpolative controllers are represented by the fact that they cannot guarantee the robustness and stability for control structures during the operating regimes not provided (included) in their synthesis. This issue represents a common problem for all the systems with non-linear controllers [11].

3 Structure presentation of the intelligent design environment

The proposed design environment consists in a collection of MATLAB function and SIMULINK schemes managed through a user interface. SIMULINK schemes are configurable by user in order to correct and to justify some values or even to reconfigure some blocks if it is needed.

The environment addresses to second order positioning systems, as simple mechatronic ele-

ments, and offers the possibility to design an interpolative control structure in three stages: 1 - State feedback control structure design, 2 - Interpolative control structure design, 3 - Optimized GA-based interpolative structure design. Every stage, if the user considers it, can be viewed as a final and sufficient stage or it can be seen only as a step on a complex design issue, having as a final result the optimized interpolative structure. The functionality of the environment can be followed through the organigrame presented in Figure 3.

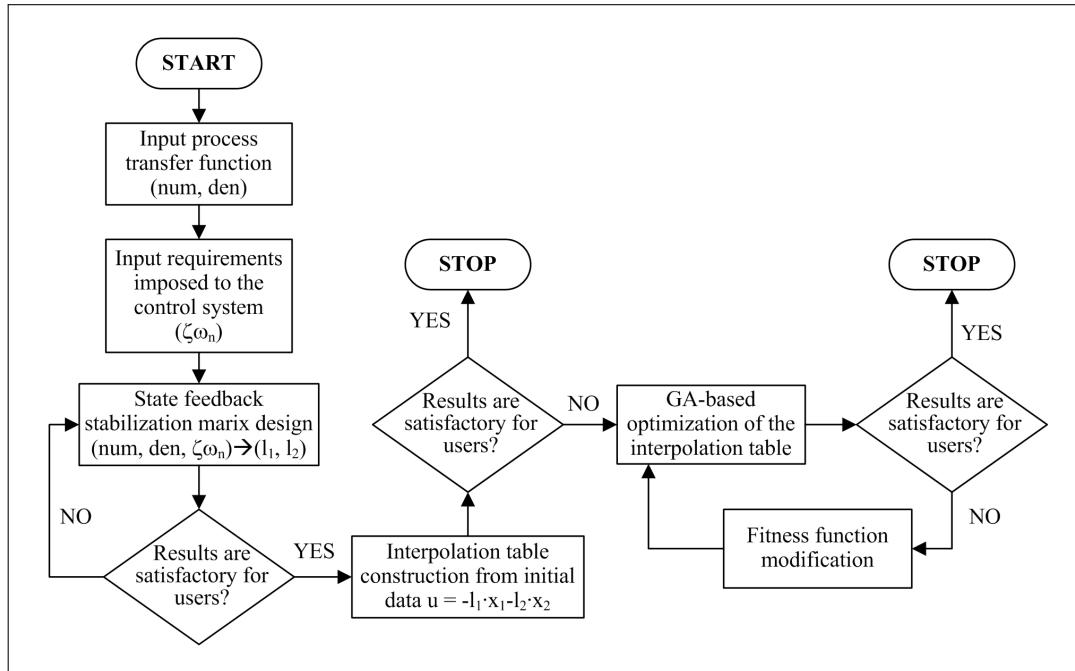


Figure 3: Intelligent design procedure

As depicted in the organigrame, the usage of the design environment presumes a sum of steps, described in detail in the following paragraphs.

3.1 Prelevation of process data

Transfer function of the positioning system and the requirements for the control systems, in terms of damp and rising time are introduced by the user. Based on that information, a preset SIMULINK scheme for the MM-ISI model of the second order system is updated.

3.2 Design of state feedback control structure

Consider the feedback state stabilization law with the following trajectory:

$$u(t) = L_c \cdot u_c + u_f(t) = L_c \cdot u_c - l_1 \cdot x_1(t) - l_2 \cdot x_2(t) \quad (2)$$

The calculus implied in state feedback control law design is implemented by a MATLAB function that use the Ackermann formula (for l_1 and l_2) and transmits the results to the SIMULINK scheme (see Figure 4) of the general state feedback control structure. With some manual corrections made on L_c parameter by trial-error efforts the results are evaluated through a step response.

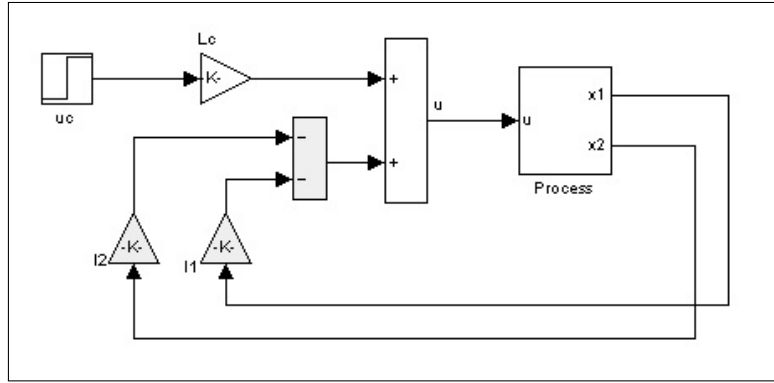


Figure 4: Initial state-feedback control loop (for second order processes)

3.3 Interpolative controller design

Based on initial data, significant information are extracted from the command characteristic of the state feedback control structure, as support points. A table which consists in a number of row and columns chosen by the user - with linguistic terms meaning for the variables $x_1(t)$ and $x_2(t)$ - is filled with the information organized as a Cartesian grid. The command law remains the same, only expressed through an interpolation table, as in Table 1.

Table 1: Interpolation table approximating the control law

x_1/x_2	$LT2_{-n}$	$LT2_{-n+1}$...	$LT2_0$...	$LT2_{n-1}$	$LT2_n$
$LT1_{-n}$	$u_f = -l_1 x_1 - l_2 x_2$						
$LT1_{-n+1}$							
...							
$LT1_0$							
...							
$LT1_{n-1}$							
$LT1_n$							

The interpolation table is implemented as a 2D look-up table in the SIMULINK scheme of the interpolative control structure, replacing the summation element with the irrespective coefficients (which implements the effective state feedback law) in control structure in Figure 4. The interpolative control structure can be seen in Figure 5. After some corrections made on L_c parameter and on table locations made by trial-error efforts the results are evaluated through a step response. If the user is satisfied with the results, the process can be stopped here.

3.4 Optimization of the interpolative control algorithm implemented as an interpolation table through GA-based techniques

As described in section 2 of the present paper, a GA is used to optimize the performances of the interpolative control structure, starting from a population of initial tables $\{T_i\}$ with the restriction: $T_i \in [-150\% \cdot T_{initial} + 150\% \cdot T_{initial}]$, i.e. all the support values corresponding to the support points should not surpass, point by point, the $\pm 150\%$ centered on the initial support values obtained at step 3 from the state feedback command surface. During the evolution of the algorithm (realized by MATLAB functions) the best individuals are stored as tables kept in especially created directories, out of which, in the end, one will replace the interpolation table $T_{initial}$. In this case, a target function was formed, by which was imposed that the system

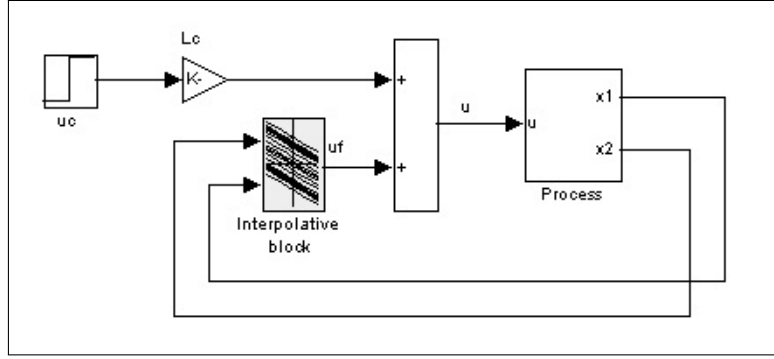


Figure 5: Interpolative control loop (for second order processes)

response to be ideal, meaning to be identical to the standard (unit step). From here appears the necessity of introducing the following criterion, which in fact is a minimizing one: the smaller the difference between ideal response and the tested system, the better the tested system will be.

$$J_i(T_i) = \text{abs} \left[\int_0^{+t_{max}} [y_i(t) - y_{ideal}] dt \right] \quad (3)$$

, where y_{ideal} is the standard. The genetic algorithm uses a fitness function that is a sum of conditions related to the error polarity, speed of the response, monotony a.s.o. The fitness function is implemented in MATLAB-SIMULINK, such that it allows the imposing of the above-mentioned conditions on the whole considered time, or only on certain intervals. At the end of the procedure, the results evaluation is made and if they are satisfactory to the user, the process stops here or. If the results are considered unsatisfactory, the fitness function implemented in the SIMULINK optimization scheme can be modified and the process can be resumed from the beginning of the 4th step.

The interpolation table considered as the best individual at step 3.3 will replace the look-up table in the interpolative control structure. With this action, the entire design process ends.

As it can be observed from the description above, the usage of the proposed design environment presumes that the user has some minimal knowledge about control systems and their performances evaluation (for reconfiguration of the schemes and parameters correction) without implying detailed knowledge regarding control system design. The user interface provides a friendly and easy communication and usage of the environment. A text containing instructions for the user is available, describing in detail minimum requirements, design steps, MATLAB functions and SIMULINK schemes involved in the designing process.

4 Case study: second order positioning system

As a study case a second order positioning system was chosen, given by the transfer function in (4) and the state-space model in (5).

$$H(s) = \frac{0.5}{s^2} \quad (4)$$

$$\begin{cases} x_1(t) = -0.5 \cdot x_2(t) \\ x_2(t) = -u(t) \\ y(t) = x_1(t) \end{cases} \quad (5)$$

The performance requirements for the control system were set at $\zeta = 0.9$ damping and $t_r = 0.8$ sec rising time. After the first step data introduction the design procedure begins with the first stage: state feedback stabilization structure design. For L matrix of the state controller calculus, the following results were provided: $L=[l_1 \ l_2] = [0.8889 \ 19.7531]$. The value for the L_c coefficient of the trajectory following component was manually set on the SIMULINK scheme at $L_c = 39.5$. The simulation results of the step response in (6) are presented in Figure 6 (continuous line) and the command surface in Figure 7.

$$u_c(t) = 1,2 \cdot \sigma(t) \quad (6)$$

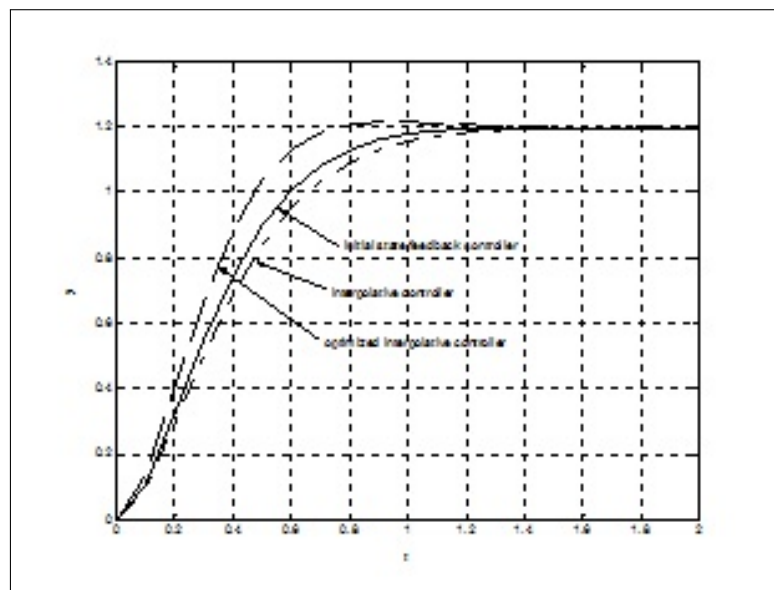


Figure 6: Step response

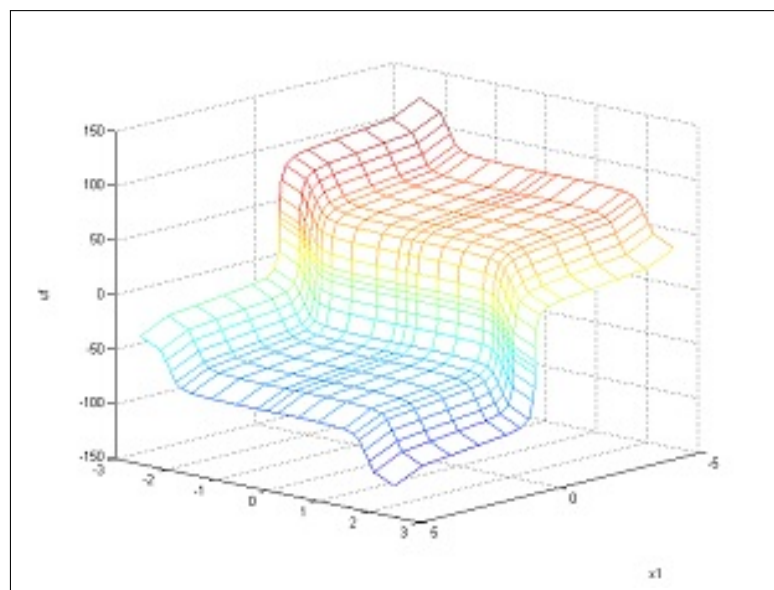


Figure 7: Command surface

In the next stage, significant support points from the command surface of the state feedback

controller were chosen and a 2D look-up table replaces the initial state feedback controller. The simulation results for the step response of the interpolative control system are depicted also in figure 6 (point-dashed line).

Finally, the GA-based optimization was applied to the interpolative controller. A simple GA was used tournament selection, simple crossing and uniform mutation. The size of the population was 50. The probability of crossing and mutation were considered 0.45 and 0.01 respectively, using the Grefenstelte references. The initial populations evolved for 50 generations. The simulation results for the step response of the optimized interpolative control system are depicted also in figure 6 (dashed line).

The results obtained through simulation for the three control structures, for the same scenario, present sensible differences between them, with a small advance for the optimized interpolative controller, that justify the demarche made by developing the design environment.

5 Conclusions and Future Works

An Intelligent design environment development can appear as a daring purpose. The result is a collection of MATLAB functions and SIMULINK schemes, unified by a quite explicit user interface that offers the possibility to design an optimized interpolative structure for any second order positioning system. The procedure is semi-automatic there are operations which have place in a non-transparent manner (they are done by the functions embeded in the environment without interacting with the user) and also other actions that involve the user in a more or less specialized interaction with the environment.

The design environment proposed in the article offers the possibility, for engineers and also for non-specialists, to design in three steps control systems for second order positioning systems and also to simulate their behavior. At the beginning, a state feedback stabilization algorithm can be designed which will constitute the base for a second step, the procedure to develop an interpolative structure. This one will be, in a third step, optimized through a GA-based technique. The environment consists in MATLAB functions, configurable SIMULINK schemes united by a user interface. The user can study and compare the behaviors of the system in each design stage, having hence the possibility to provide some conclusions related to feasibility, implementability and resources for the involved algorithms.

Some observations can be made based on the results obtained in previous work and on the study case from the present article: the combination interpolative algorithm plus GA-based optimization techniques offers easy to implement optimized controllers without loses in control performances and in specific cases even better from the performances point of view. Encouraged by the results obtained through simulation and dSPACE experiments, in future work, assisted with the proposed design environment, real-time applications on more complex systems will be developed.

Bibliography

- [1] S. Dale, H. Silaghi, D. Zmaranda, U. Rohde (2014), Procedural Aspects Concerning Intelligent Design Environment for Real-Time Applications, *Abstracts of ICCCC Papers*, 5th International Conference on Computers, Communications & Control, ICCCC2014, Baile Felix, Romania, ISSN 1844-4334, 4:41.
- [2] F.G. Filip (2012), A Decision-Making Perspective for Designing and Building Information Systems, *International Journal of Computers Communication & Control*, ISSN 1841-9836, 7(2):264-272.

- [3] S. Dale, G. Gabor, C. Gyrodi, D. Zmaranda (2010), Interpolative Control Algorithm Applied on a 3D-Mechatronic System, *Proceedings of 4th International Workshop of Soft Computing Applications*, Arad, Romania, 2010, 229-233.
- [4] A. Bara, S. Dale, T.Z. Nagy (2009), Comparative Real Time Experimental Study Case for Control Algorithms from the Implementation Point of View, *Proceedings of the 8th WSEAS International Conference on System Science and Simulation in Engineering (ICOSSSE09)*, Genova, Italy, 302-306.
- [5] S. Dale, T.L. Dragomir (2009), Interpolative-type Control Solutions, *Studies in Computational Intelligence*, Berlin, Springer Verlag, 196: 169-203.
- [6] S. Dale (2009), Interpolative Control for a d.c. Motor Drive with Genetic Algorithm-based Tuning, *Proceedings of EMES09 in Journal of Computer Science and Control Systems*, Oradea, 93-96.
- [7] A. Bara (2006), An Algorithm for Fuzzy Relation Identification, *Proceedings IEEE-TTTC 2006*, Tome1, AQTR 2006, Cluj-Napoca, 96-102.
- [8] T.L. Dragomir, E. Vladu, S. Dale (2003), Interpolative-type Controller Synthesis Using Genetic Algorithms, *Proceedings of CSCS14*, Bucuresti, 70-75.
- [9] T.L. Dragomir, I. Silea, S. Nanu (2002), Control performances improving by interpolator controllers, *Proceedings of The 6th World Multiconference on Systemics, Cybernetics and Informatics*, July 14-18 2002 Orlando, Florida, 6: 208-213.
- [10] G.F. Kamische, J.P.Brauer (1999), *Qualittsmanagement von A bis Z*, Hanser.
- [11] M.M. Seron, et all (1997), *Fundamental limitation in filtering and control*, Springer-Verlag, 1997.
- [12] D. Drechsel (1996), *Regelbasierte Interpolation und Fuzzy Control*, Vieweg, 1996.