

Comparison of Parameter Identification Techniques

Rafael Eder¹, Christian Zehetner¹ and Wolfgang Kunze²

¹Linz Center of Mechatronics GmbH, 4040 Linz, Austria

²Salvagnini Maschinenbau GmbH, 4482 Ennsdorf, Austria

Abstract. Model-based control of mechatronic systems requires excellent knowledge about the physical behavior of each component. For several types of components of a system, e.g. mechanical or electrical ones, the dynamic behavior can be described by means of a mathematic model consisting of a set of differential equations, difference equations and/or algebraic constraint equations. The knowledge of a realistic mathematic model and its parameter values is essential to represent the behaviour of a mechatronic system. Frequently it is hard or impossible to obtain all required values of the model parameters from the producer, so an appropriate parameter estimation technique is required to compute missing parameters. A manifold of parameter identification techniques can be found in the literature, but their suitability depends on the mathematic model. Previous work dealt with the automatic assembly of mathematical models of serial and parallel robots with drives and controllers within the dynamic multibody simulation code HOTINT as fully-fledged mechatronic simulation. Several parameters of such robot models were identified successfully by our embedded algorithm. The present work proposes an improved version of the identification algorithm with higher performance. The quality of the identified parameter values and the computation effort are compared with another standard technique.

1 Introduction

The trend to highly accurate products – often with arbitrary lot size and low production time – requires a deep understanding of the production process with highly complex mechatronic machines e.g. automatic panel benders as provided by Salvagnini (www.salvagninigroup.com). In order to satisfy this demand, it is necessary to obtain detailed knowledge of the mechatronic behavior of each machine component. After representing this knowledge as a mathematical model, the methods of model-based control lead to improvements in the performance and accuracy of the production process. Thus, it is important to develop highly accurate mathematical models with accurate parameter values. Usually some important parameters of the mathematic model of a mechatronic machine part are unknown, so it is important to compute those missing parameters with an appropriate parameter identification technique.

A manifold of parameter identification techniques can be found in the literature. Main focus of the present work is to compare several techniques, to find criterions for choosing the appropriate strategy for parameter identification for mathematic models of specific mechatronic components. Those models may also contain nonlinearities and discontinuities, so it is necessary to use a zero-order method, which does not need gradients information. Furthermore, it is important, that those

techniques should also be suitable to automatically assembled models, e.g. [1]. In [2, 3], such models have been obtained by multibody simulation code HOTINT. In a previous work, a novel zero-order parameter identification algorithm has been introduced and applied successfully to parameter identifications of test examples with well-known parameters as well as to real mechatronic systems, e.g. the identification of drive parameters from an industrial robot with parallel kinematics [4]. In a further work and among investigations with this identification algorithm, the performance assessment and the comparison against another heuristic technique, i.e. the Particle Swarm Optimization (PSO) [5, 6], showed higher accuracy and precision in most of the tested cases as well as lower computation times and a lower number of optimization steps [7]. Our experience has shown, that Newton Search is not very suitable for such problems, especially when the initial parameter values are not close to the optimal parameters [8].

Various Genetic Algorithms (GA's) were applied successfully to highly complex mechatronic optimization problems [9, 10, 11]. Increasing computation power as well as the capability to parallelize GA's has lead generally to an upward trend of this way to solve optimization problems, compare Fig. 1.

The present work is focused on the performance and quality of the identification results, especially when the local minima of the cost functions increase due to the

number of unknown parameters. In contrast to previous works, an improved parameter identification algorithm is proposed, in the following labeled as Special Genetic Algorithm (SGA), i.e. a special type of the GA's and described in section 2.1. The results of the parameter identification are compared to a standard GA.

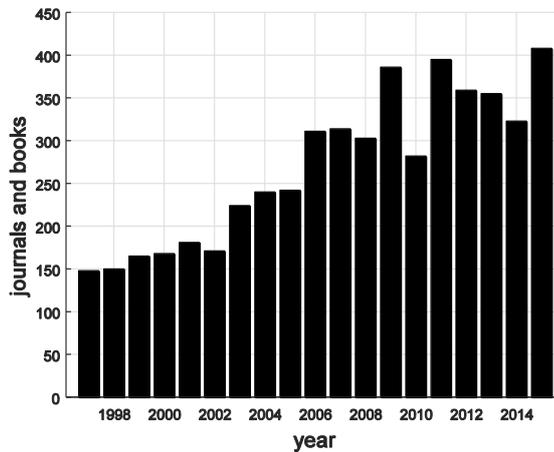


Figure 1. Trend of GA – publications (source: ScienSeDirect).

2 Identification techniques

The proposed SGA is described in section 2.1. Standard parameter identification techniques for comparisons are shortly described in section 2.2 and 2.3.

2.1 SGA

The algorithmic efficiency of the SGA has been increased in order to compete with other heuristic techniques. The improved SGA is described in this section – the previous version of the algorithm has been described in [4].

The SGA is an evolutionary algorithm. Therefore, the fitness of a population of parameter vectors dedicated to a mathematic model is optimized. Each fitness value is computed by an evaluation of the cost function. Note, that in parameter identification problems, the cost function is a measure of the accordance of simulated and measured quantities. The best parameter vector is dedicated to the global minimum. In order to find it, following major steps are processed:

1. Compute generation g of parameter vectors (Initialisation/Mutation) and compute fitness values
2. Choose fittest parameter vectors (Selection)
3. Check stop criterion
4. If not stopped: $g \rightarrow g+1$, then repeat 1.

2.1.1 Step 1: Initialization or mutation

In step 1 of the SGA, the components of the parameter vectors are computed. Note, that each component may be dedicated to an unknown parameter value of a mathematic model.

Due to the fact, that the first generation of parameter vectors has no predecessor, the first generation ($g=1$), is uniformly distributed between the user defined limits of each parameter.

For all further generations, $g \in \{2, 3, \dots\}$, the surviving parameter vectors (also denoted as parent parameter vectors) are well-known after the selection process, as described in section 2.1.2. Thus, the components of those mutated parameter vectors of the further generations depend on the selected surviving parameters. The main idea is that the mutation of the parent parameter vectors leads to a user defined number of children parameter vectors, which are located near the surviving parameter vectors. Thus, each surviving parameter is surrounded by its normally distributed children parameters after the mutation process is finished. Remark, that the children parameters are placed in a normally distributed parameter space, which shrinks each generation, so the SGA is able to estimate the optimal parameter vector very close to the real optimum.

After the computation of a generation of children parameter vectors due to the latter described initialization or mutation, their cost functions are evaluated.

2.1.2 Step 2: Selection

In the selection step, the fitness function of each parameter vector is well-known. Our experience with the SGA showed, that the algorithm for the selection process in the SGA is crucial for the computation speed. Thus, during the selection step, only a user defined number of the fittest children parameter vectors survive, i.e. chosen from the union of the set of surviving parameter vectors of the previous generation and the set of children parameter vectors of the actual generation. Note, that it is important to choose an adequate number of surviving parameter vectors for searching in the whole parameter space in a sufficient way.

2.1.3 Step 3 – Stop criteria

The extended stop criteria of the SGA define the end of the computation:

1. Stop, if a user defined, maximal number of generations g is reached
2. Stop, if the best cost function value is below an user-defined accuracy limit

After stopping the algorithm, the fittest parameter vector with the lowest cost function is returned.

2.1.4 Step 4 – Compute next generation

If the latter described stop criteria are not fulfilled, the number of generations g is increased. Then the algorithm proceeds with the mutation process, as described in section 2.1.1.

2.2 Genetic Algorithm (GA)

An overview over GA's can be found in [12]. In contrast to the latter described SGA, a common GA uses also the principle of crossover of parent parameters to obtain children parameters. The latest released genetic optimization code, the Octave package "ga" was chosen

for further investigations. In section 4, the results of the comparison of the GA with the SGA can be found.

2.3 Nelder-mead simplex method

A highly popular direct search method is the Nelder-Mead Simplex algorithm (NMS). The convergency properties for low dimensions are investigated in [13]. Experience with this technique has shown, that the choice of the start value of the parameter search should be chosen near the global minimum, otherwise the optimization result might be only a local minimum. If the cost function has multiple minima, e.g. in cost function shown in Fig. 2,

$$e(\theta) = [\cos(\theta - 20) + 0.1(\theta - 20) + 1]^2 + 0.1|\theta - 20| \quad (1)$$

In contrast to the GA and SGA, the NMS only can find a local minimum near the starting parameter value, i.e. 50, see Fig. 2. The computation time of NMS is nearly equal to the GA (section 2.2). The SGA, see section 2.1, finds the minimum of eq. 1 nearly four times faster than the NMS.

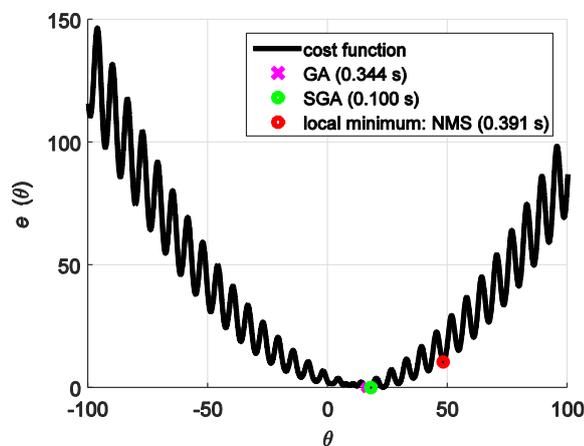


Figure 2. Local minimum found by NMS; GA and SGA find better solutions.

Due to this convergence to a local minimum, the further investigations are more focussed on the comparison of the SGA and GA. Although the GA needs 3.44 times higher computation time for minimizing eq. 1, only the SGA finds the global minimum, see Fig. 3.

These results motivate to perform further comparison of the performance and quality of the SGA and standard heuristic techniques for parameter identification in more detail. Therefore, a set of cost functions has been defined in section 3.

3 Benchmark cost functions

In the following benchmark, the identification algorithms search the optimal N -dimensional parameter vector,

$$\theta_{opt} = [\theta_{1,opt}, \theta_{2,opt}, \dots, \theta_{N,opt}], \quad (2)$$

i.e. minimize the N -dimensional cost functions.

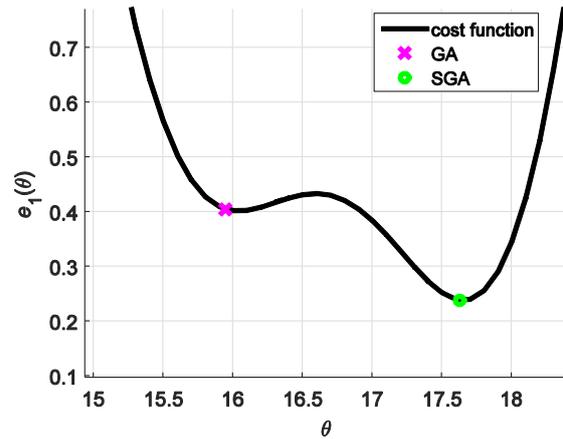


Figure 3. SGA find global minimum of function e , GA only a local minimum.

For the comparison of the GA and the SGA, the cost functions in Table 1 are defined, which are well-established for testing optimization methods.

Table 1. Cost functions for benchmark.

Rosenbrock	$\sum_{k=1}^{N-1} 100(\theta_{k+1} - \theta_k^2)^2 + (1 - \theta_k)^2$
Rastrigrin	$\sum_{k=1}^N \theta_k^2 - 10 \cos(2\pi \theta_k)$
Griewank	$\sum_{k=1}^N \frac{\theta_k^2}{4000} + \prod_{k=1}^N \cos\left(\frac{\theta_k}{\sqrt{k}}\right) + 1$
Sphere	$\sum_{k=1}^N \theta_k^2$

The optimal parameter value of each component of the optimal, N -dimensional parameter vector is searched in the very large interval $[-25, 100]$. This leads to a high number of local minima, especially in the Rosenbrock, Rastrigrin and Griewank function.

4 Results

In the following, the parameter identification algorithms are tested with the cost functions in section 3. The settings of the algorithms are described in section 4.1 and the identification residuals and the computation times are shown in section 4.2.

4.1 Settings of Algorithms

In order to compute comparable results with the GA and the SGA, the auxiliary variable

$$p = \min(5^N, 5^4) \quad (3)$$

is defined. It is a tradeoff between the increasing number of local minima, which strongly depend on the dimension N and the required computation times. Remark, that the algorithms might find the optimum also with a lower number p , but the present work is interested in the comparison of different comparable parameter identification algorithms.

Table 2. Settings of parameter identification algorithms.

Option	SGA	GA
Surviving population size	p	No GA option
Number of children	p	No GA option
Initial population size	p	No GA option
Number of maximal generations	15	p
Accuracy limit	10^{-4}	10^{-4}
Population size	No SGA option	p^2

The options of SGA and GA are summarized in Table 2. Note, that with the default options of the SGA, a number of 15 generations is sufficient, due to the reduction parameter range around the surviving parameter vector, see section 2.1.1.

The SGA and the GA have slightly different options, see Table 2, but the population size is equivalent to the surviving parameter size multiplied with the number of children in the SGA.

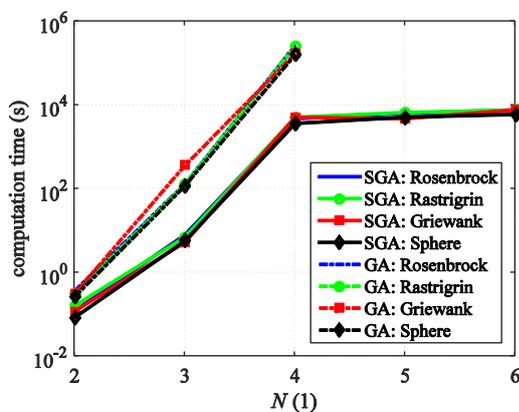


Figure 4. Computation time (s): SGA is faster than GA.

4.2 Computation results of GA and SGA

contrast to previous work, the SGA has been computed against a standard GA for the first time in a successful

The following results are obtained by applying the GA and SGA (section 2), to minimize the benchmark cost functions, as defined in section 3 with rising dimension of the parameter space N . Note, that it was not possible to compute all results with the GA within reasonable computation times.

Table 3. Cost function residuals.

	GA $N=\{2,3,4\}$	SGA $N=\{2,3,\dots,6\}$
Rosenbrock	$[4.55 \times 10^1,$ $1.97 \times 10^{-5},$ $4.96 \times 10^{-5}]$	9.51×10^{-6} 1.32×10^{-5} 9.50×10^{-5} 3.06×10^{-5} 3.72×10^{-4}
Rastrigrin	$[4.66 \times 10^{-1},$ $5.32 \times 10^{-5},$ $5.19 \times 10^{-5}]$	1.48×10^{-5} 6.03×10^{-5} 9.67×10^{-6} 4.20×10^{-5} 1.92×10^{-4}
Griewank	$7.40 \times 10^{-3},$ $7.40 \times 10^{-3},$ $7.98 \times 10^{-5}]$	8.42×10^{-6} 2.00×10^{-5} 7.40×10^{-3} 7.66×10^{-6} 9.86×10^{-3}
Sphere	$[5.59 \times 10^{-5},$ $3.91 \times 10^{-5},$ $4.76 \times 10^{-5}]$	5.33×10^{-5} 2.48×10^{-5} 3.62×10^{-5} 5.85×10^{-5} 3.39×10^{-5}

5 Conclusion

In the present work, the comparison of quality and computation time of a common GA with a proposed parameter identification technique SGA leads to the conclusion, that the SGA solved the minimization problem of the benchmark cost functions faster than the GA, see Figure 4. Furthermore, the cost function residuals were solved more accurately than the standard GA in the majority of the tested cases (Table 3). Even, in cases where no reasonable computation time has been possible with the standard GA, the SGA solved also problems in a six-dimensional parameter space and a high number of multiple local minima. The ability to deal also with a higher number of unknown parameters is needed for parameter identification of mechatronic systems. In way. The proposed method needs no differentiation of the cost function and algebraic manipulations of the

mathematic model of a mechatronic system and is immediately usable to arbitrary identification problems. Further work will focus on possible improvements of the SGA in higher dimensions of the parameter vector.

Acknowledgement

This work has been supported by the Linz Center of Mechatronics (LCM) in the framework of the Austrian COMET-K2 programme.

References

1. R. Ludwig, J. Gerstmayr, C. Augdopler, C. Mittermayr, Flexible Robot with Clearance, Proc. of 4ECSC **2**, (2008)
2. J. Gerstmayr, M. Stangl, High-order implicit Runge-Kutta methods for discontinuous mechatronical systems, Proc. XXXII Summer Sch. AP, St. Petersburg, (2004)
3. J. Gerstmayr et. al., HOTINT-a script language based framework for the simulation of multibody dynamics systems, Proc. of ASME 2013 IDETC&CIE Conf., Oregon, (2013)
4. R. Ludwig, J. Gerstmayr, Automatic Parameter Identification for Generic Robot Models, Proc. of ECCOMAS Conf., Brussels, Belgium, (2011)
5. J. Kennedy, R. Eberhart, Particle Swarm Optimization, Proc. IEEE Int. Conf. on Neural Networks **4** (1995)
6. J. Kennedy, R. C. Eberhard, *Swarm Intelligence* (Morgan Kaufmann Publishers, San Francisco, California, 2001)
7. R. Eder, J. Gerstmayr, Special Genetic Identification Algorithm with Smoothing in the Frequency Domain, Adv. Eng. Softw. **70C** (2014)
8. R. Ludwig, J. Gerstmayr, Identification of vibration-relevant parameters in robotic systems, Proc. of Austrian Robotics Workshop, Linz, (2014)
9. P. Boyraz, M. Gündüz, Dynamic modeling of a horizontal washing machine and optimization of vibration characteristics using Genetic Algorithms, Int. J. Mechatron. **23**, 6 (2013)
10. C. Chan, G. Liu, Hysteresis identification and compensation using a genetic algorithm with adaptive search space, Int. J. Mechatron. **17**, 7 (2007)
11. Z. Affi, B. El-Kribi, L. Romdhane, Advanced mechatronic design using a multi-objective genetic algorithm optimization of a motor-driven four-bar system, Int. J. Mechatron. **17**, 9 (2007)
12. I. Rechenberg, *Evolutionsstrategie 1* (Friedrich Frommann Verlag, Stuttgart, 1994)
13. J. C. Lagarias, J. A. Reeds, M. H. Wright, P. E. Wright, Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions, SIAM J. of Optim. **9**, 1 (1998)