Choosing an Objective Function Type When Adjusting Mechanical Components' Macromodel Parameters

Volodymyr Ladogubets, Oleksandr Beznosyk, Oleksii Finogenov

CAD Department, NTUU "KPI", UKRAINE, Kyiv, P. Myrnogo street 19, E-mail: sasha@cad.ntu-kpi.kiev.ua

Abstract – **In the paper the problems of choosing an objective function type when adjusting mechanical components' macromodel parameters are considered. The most used objective function types which cover the majority of practical optimization tasks are specified, and their analysis is conducted. The recommendations on choosing an objective function type depending on the task's features are proposed.**

Keywords – **Simulation, Optimization, MEMS.**

I. Introduction

Choosing an objective function type is one of the tasks to be solved to adjust mechanical components' macromodel parameters [1]. According to the automatized technique of obtaining macromodels from finite-element descriptions [2], in order to simulate heterogeneous devices it is advisable to bring separate subsystems to a common basis, for example, to replace a mechanical part by some equivalent electrical circuit.

As a circuit design software it is reasonable to use NetALLTED package [3] due to a variety of the optimization methods and a wide set of the objective functions (Table 1).

Objective functions could be quite different by their appearance, however all of them are similar by a form. All the objective functions are grouped into the special library and could be used to simplify preparing an optimization task. In fact, this set of the objective functions cover the majority of practical optimization tasks. Whether there is no a proper objective function for a task in question, there is a possibility to add a user-defined objective function by means of the specially designed USOBJ subroutine [4].

In the Table 1, x_1, x_2, ..., x_n are the objective function's arguments, which could be characteristics defined by the FIX, INT, FUNC directives [4]. PV_1, PV_2, ..., PV_n stand for the objective function's parameters and in some cases (for example, for the *F1*, *F2* functions) could be interpreted as desired optimization results. *D* defines a tolerance; K_1, K_2, ..., K_n are the weighting factors; Z_{min}, Z_{max} are the lower and upper limits of the optimization function's allowable values range at the linear search.

TABLE 1

NetALLTED objective function set

№	Function	Description
1	$\sum_{i=1}^{n}(PV_i - x_i)^2$	$F1(PV_1,...,PV_n / x_1,...,x_n)$
2	$\sum_{i=1}^{n}\lvert PV_i - x_i\rvert$	$F2(PV_1,...,PV_n / x_1,...,x_n)$
3	$\sum_{i=1}^{n}K_i\left(D - \lvert PV_i - x_i\rvert\right)^2$	$F3(D, PV_1,...,PV_n / x_1,...,x_n)$
4	$\sum_{i=1}^{n}K_i\left(PV_i - x_i\right)^2$, $K_i = \begin{cases} 0 - if\ \lvert x_i\rvert \le PV_i \\ 1 - if\ \lvert x_i\rvert > PV_i \end{cases}$	$F4(PV_1,...,PV_n / x_1,...,x_n)$
5	$\sum_{i=1}^{n}K_i x_i$	$F5(K_1,...,K_n / x_1,...,x_n)$
6	$\max_i (PV_i - x_i)^2$, $i = \overline{1,N}$	$F6(PV_1,...,PV_n / x_1,...,x_n)$
7	$\max_i \lvert PV_i - x_i\rvert$, $i = \overline{1,N}$	$F7(PV_1,...,PV_n / x_1,...,x_n)$
8	$\sum_{i=1}^{N}\left(\frac{PV_i - x_i}{PV_i}\right)^2$	$F8(PV_1,...,PV_n / x_1,...,x_n)$
9	$\max_i \frac{PV_i - x_i}{PV_i}$, $i = \overline{1,n}$	$F9(PV_1,...,PV_n / x_1,...,x_n)$
10	$\sum_{i=1}^{n}K_i x_i$	$F10(Z_{\min}, Z_{\max}, K_1,...,K_n / x_1,...,x_n)$
11	$\sum_{i=1}^{n}K_i(PV_i - x_i)^2$	$F11(K_1PV_1,...,K_nPV_n / x_1,...,x_n)$
12	$\sum_{i=1}^{n}K_i\lvert PV_i - x_i\rvert$	$F12(K_1PV_1,...,K_nPV_n / x_1,...,x_n)$
13	$\max_i\left[K_i(PV_i - x_i)^2\right]$	$F13(K_1PV_1,...,K_nPV_n / x_1,...,x_n)$
14	$\max_i \lvert PV_i - x_i\rvert$, $i = \overline{1,N}$	$F14(K_1PV_1,...,K_nPV_n / x_1,...,x_n)$
15	$K\sum_{i=1}^{n}(PV_i - x_i)^2$	$F15(KPV_1,...,PV_n / x_1,...,x_n)$
16	$K\sum_{i=1}^{n}\lvert PV_i - x_i\rvert$	$F16(KPV_1,...,PV_n / x_1,...,x_n)$

II. Choosing an objective function

Although the requirements to the optimization stage remain the main criteria to choose an objective function type, however even in this case it is possible to use several objective functions meeting these requirements. Let's consider typical requirements posed to the macromodel parameter optimization stage. In a common case, when formulating requirements, one or several macromodel parameters are being defined, which have to be adjusted with respect to:

- analytical results (manual calculations);
- calculation results obtained by a third-party software;
- experimental results.

By that, depending on the object analysis type, the task formulation could be:

- to get the required displacements/angles X_i at the points B_i under the forces $Y_i(t)$ applying at the points A_i;
- to get the required values of amplitudes A_i at the specified frequencies f_i;
- to get the required signal transfer delay Δt between the macromodel's input and output.

The task formulations listed above may not cover an entire spectrum of the demanded tasks, however an improvement of either a set of discrete values (for example, eigenfrequencies or displacements of the object's specific points), or difference characteristics (such as a signal transfer delay and so on) is typical for all the task types. Solving such tasks, obviously, the parameter optimization will require getting a minimal error between macromodel parameters and results the optimization is going with respect to. So, for such tasks the objective functions of types

- *F1*: sum of squares of absolute divergences;
- *F2*: sum of absolute divergences;
- *F6*: square of maximal divergence;
- *F7*: maximal divergence;
- *F8*: sum of squares of relative divergences;
- *F9*: maximal relative divergence

are more suitable.

For some tasks the functions of types *F11* and *F12*, which use weighting factors, could be applied too, for instance, for a problem of finding displacements of the object's points when applying forces in a case of the weak connection between the object's freedom degrees. In this case, the probable displacements could vary in degrees at different points of the object complicating obtaining a macromodel that will provide the same accuracy everywhere.

The functions of types *F1*, *F2*, *F6*, *F7* are suitable when the dispersion between the objective function's parameters is not very big, for example, to adjust eigenfrequencies. In this case, it is desirable that the following ratio to be satisfied:

$$f_n \leq 10*f_1,$$

where f_1 and f_n are the last left and right eigenfrequencies being optimized. As far as the parameter dispersion increases, the accuracy of finding separate eigenfrequencies differs significantly due to using the absolute values, so the accuracy of an entire macromodel will correspond to the greatest divergence.

The functions of types *F6* and *F7*, which use maximal values of divergence, is suitable to provide macromodel parameter divergences in the ranges not exceeding the specified ones.

The functions of types *F8* and *F9* are the most universal ones as they provide the accuracy of the macromodel parameters at the same level (or not exceeding the common level).

The optimization of the parameters to the results got by means of the analytical calculations should be noted apart. In some cases, a result can be obtained as a function, however the most of the circuit design software does not allow getting a symbolic solution. Accordingly, in this case the discrete set of values to be adjusted should be specified for the optimization stage reducing the task to the ones considered above.

The optimization of the parameters to the experimental results is similar to the optimization to the third-party software's results, with the exception of some features:

- when defining the required parameter values it is needed to take into account the accuracy of defining the experimental data (there is no sense to fulfil optimization with the accuracy of 1% for the data obtained with the accuracy of 10%);
- since a macromodel came out of interpretation and reduction of a finite-element model, the initial values and the possibility to lead to the experimental results will depend on the adequacy of the source finite-element model.

References

[1] Ladogubets V. Features of the Optimization of Mechanical Component Macromodels / Vladimir Ladogubets, Oleksii Finogenov, Oleksandr Beznosyk, Tatiana Ladogubets // The Experience of Designing and Application of CAD Systems in Microelectronics : 11-th Anniversary Intern. Conf. «CADSM'2011», 23-25 February 2011, Lviv-Polyana (Zakarpattya), Ukraine : proc. – Lviv, 2011. – P. 410–412.

[2] Ладогубец В.В. Методика построения моделей механических компонентов МЭМС для пакетов схемотехнического проектирования / Ладогубец В.В., Чкалов А.В., Безносик А.Ю., Финогенов А.Д. // Электроника и связь : тематический выпуск «Электроника и нанотехнологии». – 2009. – Ч. 1, № 2/3. – С. 298–305.

[3] Ладогубец В.В. Состояние и перспективы развития автоматизтрованного схемотехнического проектирования / Ладогубец В.В. // Электроника и связь : тематический выпуск «Проблемы электроники». – 2005. – Ч. 1. – С.121–127.

[4] Petrenko A. ALLTED – a computer-aided engineering system for electronic circuit design / Petrenko A., Ladogubets V., Tchkalov V., Pudlowski Z. – Melbourne: UICEE, 1997. – 205 p.

III. Conclusion

Based on the experiments fulfiled on a test set of the mechanical components when solving typical tasks of the macromodel parameter optimization stage, it is possible to recommend applying the *F8* and *F9* objective function as most universal ones, which use relative values of divergences allowing carrying out parameter optimization in a wide value range. Using the *F1*, *F2*, *F6* and *F7* functions is limited to the tasks with a small parameter dispersion. The *F11* and *F12* functions, which use weighting factors, in the most cases could be successfully replaced by the *F8* and *F9* functions, which do not require calculation of weighting factors.