HARDWARE IMPLEMENTATION OF A NONLINEAR PARAMETER ESTIMATOR FOR PROCESS MONITORING

Uwe Marschner and Wolf-Joachim Fischer

Dresden University of Technology,
Semiconductor and Microsystems Technology Laboratory
Mommsenstraße 13, 01062 Dresden, Germany
email: marschne@ehmws1.et.tu-dresden.de

ABSTRACT

In this paper the mapping of a nonlinear parameter estimator onto a DSP-architecture under use of an Electronic Design Automation (EDA) tool is presented. The work concentrates on the implementation of static nonlinear process models with SISO-structure and block processing. It is shown that the synthesized processor is able to perform such a parameter estimation in less than 30 μs at a reasonable chip size.

1. INTRODUCTION

In control engineering the monitoring of technical processes by parameter estimation methods has been recommended [1], [2]. It can increase the precision, resolution and the stability from a statistical point of view, allows the observation of process quantities which are not directly measurable and small faults of the process operation can be detected at early stages.

One problem is the high computational power required by this method. In order to monitor a complex technical system, for example a power plant, the signals measured at each subprocess should be processed separately. One possibility is the usage of industrial PC's or DSP-Boards but they are relatively expensive and heavy to protect against environmental influences.

In this paper the design, simulation and application of a Parameter Estimation Processor (PEP) is presented. The on-chip combination of this Function and Application Specific Integrated Circuit (FASIC) together with a sensor creates a microsystem which can be located close to the process and compress the information obtained by the measured signals.

The algorithm is written in Data Flow Language (DFL), a *Mentor Graphics* version of *Silage* language from UC Berkeley.

2. ON-LINE PARAMETER ESTIMATION

Fig. 1 shows the underlying scheme for parametric system identification by output error correction with initial point adjustment.

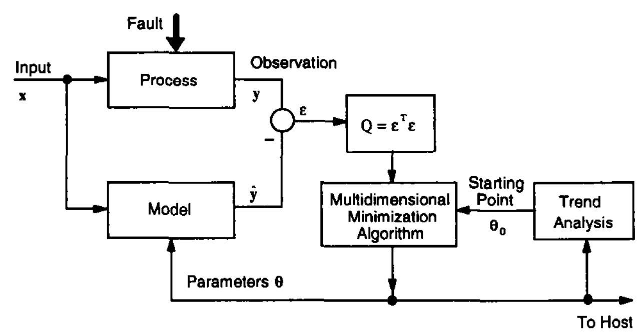

Figure 1: On-line parameter estimation with initial point adjustment

The input variables $x^N = (x(1), x(2), .., x(N))$ in $\Re^N$ and output variables $y^N = (y(1), y(2), .., y(N))$ in $\Re^N$ of the process are measured simultaneously at equidistant sampling intervals and stored to block size N in the vectors x and y during a process operation cycle, for example when a valve is closed. Every previous measured block is deleted or forgotten. The parameters are viewed as quasi constant or time invariant during block measurement.

2.1. Model and Criterion Function Calculation

The process is described by the static nonlinear function f between the arguments x and the results $\hat{y}$ by using parameters θ

$$\begin{aligned} f : \mathbf{X} \times \Theta \rightarrow \mathbf{Y}, \quad \hat{\mathbf{y}} = f(\mathbf{x}, \theta) \\ \theta \in \Theta \subset \Re^{\mathrm{M}}, \quad \mathbf{x} \in \mathbf{X}, \hat{\mathbf{y}} \in \mathbf{Y} \subset \Re^{\mathrm{N}} \end{aligned} \tag{1}$$

or

$$\begin{aligned} \hat{y_1} &= f(x_1, \theta) \\ &\cdot \\ &\cdot \\ \hat{y_N} &= f(x_N, \theta) \end{aligned} \tag{2}$$

The above mentioned information compression aspect is thereby described by the mapping of the measurement space into the parameter space.

It is assumed that the model is physical-based and includes the fault cases, thereby the faults have unique features, i.e. the characteristic curves are independent of each other. The model is already verified by real measurements and the fault cases can be related to one parameter or a tuple of parameters, i.e. the model is sensitive in these parameters.

At this stage a quadratic criterion function is calculated. A more robust function w.r.t. the measurement disturbances like a Huber function [3] could be implemented but has not been necessary for the investigated examples. In this content it is assumed that the minimization reaches only the (desired) minimum which is located together with the initialization point in the same basin of attraction.

This assumption requires the initialization of the first parameter estimation at a known process operation case, usually the normal operation case. The parameter estimation scheme will hold for such "slow" parameter changes that the model still fits the observations at reasonable criterion function values.

2.2. Multidimensional Minimization Algorithm

Iterative numerical minimization algorithms can be divided into groups depending on the usage of the function value, its gradient and its Hessian [4].

A comparison of the implementation of the simplex method and the Gradient and Hessian calculating Sequential Quadratic Programming (SQP) algorithm showed that the simplex method is suitable for the implementation in an microsystem for the intended usage. The main reason is the 7 times smaller chip size [5].

The simplex algorithm requires only $M + 1$ function evaluations in M parameter dimensions which form a geometric figure in the parameter space [6] and is known as slow but robust [7]. The simplex is able to reflect, to reflect and expand, to contract along one dimension and also to contract along all dimensions and so to reach the minimum.

Together with a moving parameter trend analysis which is used to predict the next minimization initial point the number of iterations can be decreased strongly compared to an unknown initial point.

2.3. Moving Trend Analysis

The moving trend analysis is based on a second order polynomial:

$$\theta_m(k) = K_0 + K_1 \cdot u(k) + K_2 \cdot u(k)^2 \tag{3}$$

where K_0 is constant at the k-th parameter estimation. This function should be sufficient for the description of the assumed slow parameter drift.

The LS-estimation of the linear parameters K_1 and K_2 is done by generalized correlation functions as described in [8]. The next initial point is predicted by an extrapolation of the polynomial. The implemented trend analysis uses the parameters from the last six estimations.

2.4. Parameter Estimation viewed as Discrete Time System

In order to use the automatic scheduling capabilities of the EDA-tool (see section 4) the parameter estimation scheme had been rewritten as discrete time system (Fig. 2). Here θ is the parameter vector, k the discrete time variable and Q a map of the parameter space into itself.

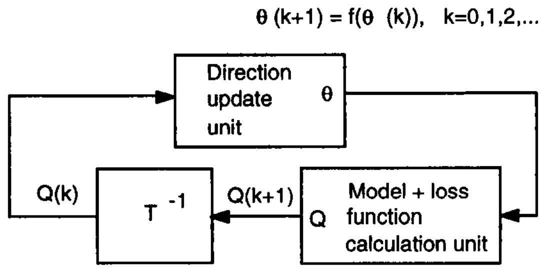

Figure 2: Discrete time system

Globally spoken one can think on a direction update at one iteration step depending on the criterion function value. The result of the behavioural simulation of the parameters generated by this system is shown in Fig. 7.

3. EXAMPLE PROCESSES

The characteristic function of a process which can be described by static nonlinear model is shown in Fig. 3. Here the potential and the electrical current of a microplating process are measured and among others the concentration of the electrolyte is estimated [9].

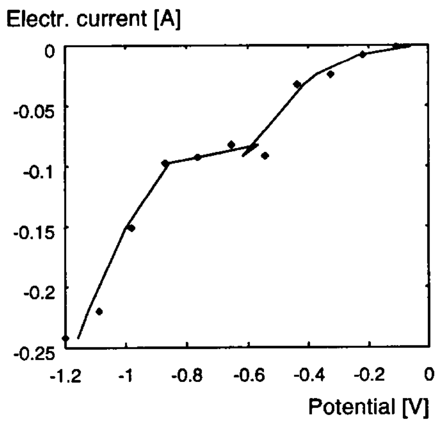

Figure 3: Microplating process with 5 parameters

A second example is the monitoring of a Motor-Operated Valve (MOV) using the relation between the stem position and the operator torque [10]. The characteristic curve shown in Fig. 4 is represented by a constant normal operation vector influenced among others by a function representing several on-going faults, for example a high stem friction due to a poor lubrication. The model is still under investigation.

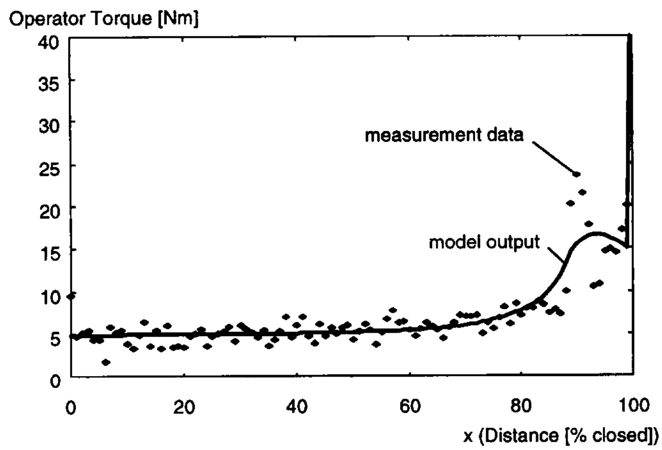

Figure 4: Model-based diagnosis of a motor-operated valve with 6 parameters

The microsystem including the PEP is intended to be fixed very close to a process and so to concentrate computational power locally.

4. DSP ARCHITECTURE

The presented PEP is designed as core. To complete the microsystem at least a sensor and an A/D converter have to be added (Fig. 5). The reliability of the data transfer should be increased by a field bus interface, for example the CAN-bus interface.

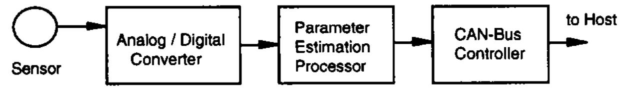

Figure 5: Microsystem parts

Design data for the A/D-converter and the CAN-bus controller are available when a single chip microsystem is desired as well as chips for each unit so that only the architecture of the PEP is explained.

The implementation of the parameter estimation scheme has been done under use of the MISTRAL2 synthesis tool by Mentor Graphics which is suited for third-generation DSP-algorithms. These DSPs are designed for the application of complex decision making algorithms to large blocks of sampled data.

The size and speed estimation for the bitparallel PEP were performed at logic level by MISTRAL2 whereas the scheme was defined at the algorithmic level. At the register transfer level a description of the processor architecture in terms of execution units (EXUs) and interconnections is available as well as the description of the processor controller. At present the EXUs shown in Fig. 6 are used.

Thereby the sampled data are read from the Input RAM (ipb_1) and processed by the Arithmetic Logic Unit (alu_1) and the Multiplier (mult_1). To the background RAM (ram_1) belong data input register files, address input register files and other registers. The foreground ROM (romctrl_1) is used for constant signals with compile-time constant address. Address computation and loop counting are performed by the Address Computation Unit (acu_1). Finally the output signals are stored in the Output RAM (opb_1). The controller of the processor is microcode-based and of multi-branch type.

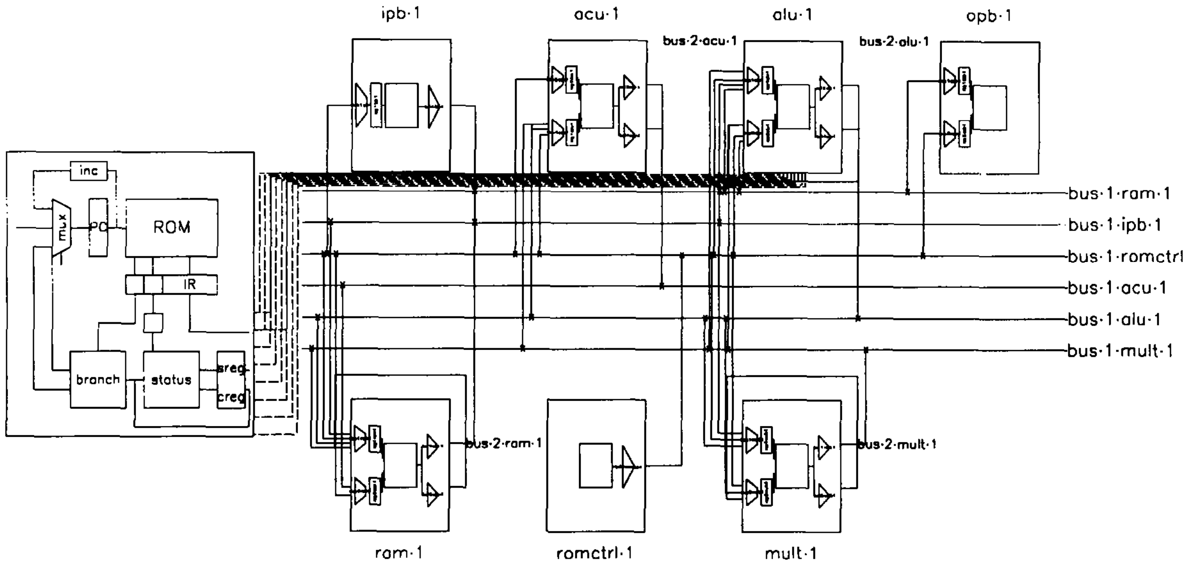

Figure 6: EXUs of the PEP

5. BEHAVIOURAL SIMULATION AND IMPLEMENTATION RESULTS

In Fig. 7 the results of a behavioural simulation of three parameters during an estimation are depicted. Since the geometric vertices of the first simplex are calculated serially one can observe the initial simplex during the first three steps. The initialization is followed by a reflection and then a contraction. Close to the minimum the simplex contracts further so that the parameter "signal amplitude" decreases. When the termination tolerance is matched the estimation stops.

In Fig. 8 the results of a timing analysis are presented. After the trend analysis the simplex module is activated delivering a parameter tuple which serves as input for the model and criterion function calculation. After M+1 iteration steps the first simplex is calculated and evaluated by the simplex module and the next iteration cycle starts.

For the simulation of a trend analysis of a parameter drift shown in Fig. 9 the parameters of the last six estimations are used.

Assuming 40 iteration cycles the total parameter estimation time amounts to 26.2 μs for the example process shown in Fig. 4.

The area of the PEP containing the simplex algorithm was determined to 87000 gate equivalents. That can be called "normal" for a DSP and can certainly further improved by accuracy investigations and architectural optimizations.

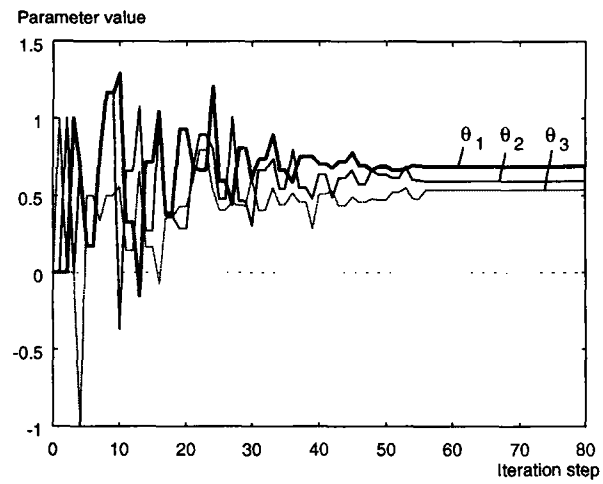

Figure 7: Parameter "signals"

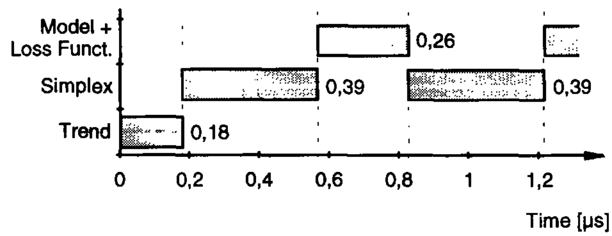

Figure 8: Timing analysis of the simplex-PEP

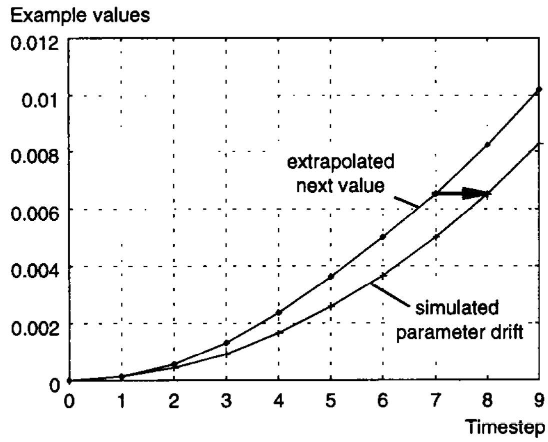

Figure 9: Trend analysis by second order polynomial

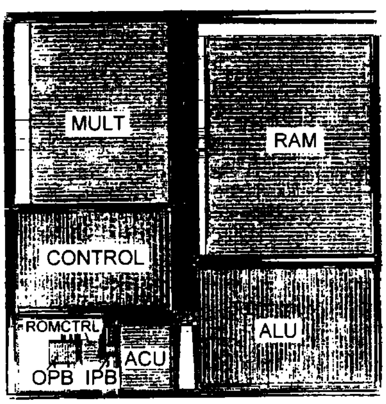

Figure 10: Layout of the simplex-PEP

6. CONCLUSION

The practical investigations presented in this paper show that it is possible to design a DSP suitable for nonlinear parameter estimation at reasonable costs (chip size). The implementation of a usually as slow known gradient free method combined with a parameter trend analysis to predict the next initial point allows the repetition of the parameter estimation approximately every 30 μs for the given example, i.e. the process parameters might change within this dynamic.

The authors wish to thank HS-Doz.Dr. H. Buchta from the Dresden University of Technology for stimulating discussions on model-based diagnosis as well as H. Grätz (Fraunhofer Institut für Mikroelektronische Schaltungen und Systeme) for his help during the DSP synthesis.

7. REFERENCES

[1] A. van den Bos. Parametric statistical model-based measurement. *Measurement*, (14):55–61, 1994.

[2] Isermann, R. On the Applicability of Model Based Fault Detection for Technical Processes. In *12th IAFC World Congress*, Sydney, Australia, July 18-23 1993.

[3] H. Schwetlick. Nonlinear parameter estimation: Models, criteria, and algorithms. *Numerical Analysis*, page 164, 1991.

[4] Ljung, L. *System Identification, Theory for the User*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[5] Marschner, U. and Fischer, W.-J. *Local Parameter Estimation Performed by Tailor-Made Microsystems*. SAFEPROCESS'97, Kingston Upon Hull.

[6] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge New York Oakleigh, 1992.

[7] J. Brignell and N. White. *Intelligent Sensor Systems*. Institute of Physics Publishing, Bristol, 1994.

[8] Isermann, R. *Idendifikation dynamischer Systeme 1*. Springer Verlag, Berlin, 1988.

[9] Marschner, U. and Hoffmann, J. *Modelbased Diagnosis of Microplating Processes*. in preparation, 1997.

[10] Marschner, U. *A New Wedge Gate Valve Model for Model Based Diagnosis*. Fourth Annual Review Meeting of the MIT International Program on Enhanced Nuclear Power Plant Safety, Cambridge, 1993.