2nd International Conference "From Scientific Computing to Computational Engineering"
2nd IC-SCCE
Athens, 5-8 July, 2006

COMPARISON OF ALGORITHM COMPUTATIONAL REQUIREMENTS FOR REAL-TIME MONITORING OF STRUCTURAL COMPONENTS

Vassilios C. Moussas

Laboratory of Applied Informatics
Department of Civil Works Technology
School of Technological Applications (S.T.EF.),
Technological Educational Institution (T.E.I.) of Athens
Ag. Spyridonos Str., Egaleo(Athens) GR-12210, Greece,
Email: vmouss@teiath.gr, Web page: www.vmoussas.com

Keywords: Failure Prediction, Fatigue Crack Growth, NDT, NDE, Acoustic Emission, Extended Kalman Filter, Multi-Model Partitioning Algorithms, ALF

Abstract. *Critical structures often require extensive monitoring of their components to ensure proper condition and operation. Especially when operating under severe loads or extreme conditions, real time monitoring can detect evolving defects and prevent future failures. A combination of Non-Destructive Testing, Detection, Identification and Prediction techniques is required. Typically the data collection is done on line and the rest of the techniques are applied off line, due to time, space and computational requirements. In this work we investigate and compare the requirements of several advanced stochastic algorithms for detection, identification or prediction, such as EKF or MMPA, in order to apply them on line during data collection. Depending on the data collection rate of the NDT method and the speed of the selected algorithm, it is possible to combine real-time monitoring of structure components with real-time identification and prediction using even the more demanding stochastic algorithms.*

1 INTRODUCTION

Critical (i.e. rare, expensive, dangerous, etc.) structures often require extensive monitoring of their components to ensure proper condition and operation. Lifetime predictions and in-service inspections of each component are used to update the reliability analysis of the overall structure. Especially when operating under severe loads or extreme conditions, real-time health monitoring can detect evolving defects and prevent future failures in large structures (airplanes, nuclear reactors, bridges, etc.) [1-4].

Fatigue is a major factor of component failure, therefore, Fatigue Crack Growth (FCG) monitoring and prediction is an essential tool for life prediction and maintenance. When studying structural components that are subjected to cyclic stresses over a prolonged period of time, a combination of Non-Destructive Testing, Detection, Identification and Prediction techniques is required. The models applied for detection and prediction depend on the selected FCG laws and NDT/NDE methods (e.g. Acoustic Emission, Thermal Emission, Potential Drop, etc.). Today's FCG and NDT models become more and more complex and therefore advanced nonlinear and adaptive methods of increased complexity are required to process them [5-7].

Typically, due to time, space and/or computational requirements, the data collection is done on site and the identification and prediction techniques are applied later off line. For example, data collected during a flight are processed later, when the aircraft is landed, using a powerful mainframe. Today, on board computers become smaller and more powerful and they are able to perform more complex processing tasks. Our aim is to improve structural reliability by reacting early and prevent catastrophic disasters. Early warning and reaction can be improved by performing all monitoring and prediction processing on-line, during the data collection.

In this work we focus on the problem of FCG estimation, identification and prediction of the remaining time to failure, using NDE data collected by Acoustic Emission (AE) methods. We investigate the possibility to perform the identification and predictions on-line. We compare the requirements of several advanced stochastic algorithms for detection, identification or prediction, such as the Extended Kalman Filter (EKF) and the Multi-Model Partitioning Algorithm (MMPA) proposed by Lainiotis [8-9]. Finally, we compare the speed and time requirements to the data collection rates in order to select the appropriate algorithm for on-line implementation.

2 THE ACOUSTIC EMISSION NDT/NDE METHOD

The Acoustic Emission (AE) method uses several resonant sensors on the surface of a component to detect acoustic emissions originating from growing defects inside the component. The data collection system usually consists of a set of acoustic sensors, a preprocessing module and a data storage module. The preprocessing module separates potential AE signals from other acoustic signals or noise. When three ro more sensors are available, preprocessing also detects the exact position of the AE source. All potential AE signals are stored for later processing by more sophisticated algorithms. This processing step is computationally intensive and includes pattern recognition techniques, as well as identification, estimation and prediction algorithms.

The collection of acoustic emissions is a demanding task and requires large and fast storage capabilities. In addition, sensitive sensors and fast hardware are required sampling at a higher rate. Most of the AE signals are irrelevant to the FCG. Other sources such as friction, heat, displacement, other components, etc., generate hundreds of AE signals every second.

In order to provide the necessary processing time between successive inputs, it is necessary to eliminate as soon as possible any noise and any other irrelevant signal. This is performed on-line using filters and thresholds on the acquisition devices. For instance, the AE signals collected during FCG experiments [11] on a steel vessel, had specific characteristics such as duration (e.g. 182-283 μsec), rise time (e.g. 10-30 μsec), amplitude (e.g. 125-280 mV), they appeared after a certain pressure level (e.g. 220-240 Bar), etc. By adjusting the device sensitivity, thresholds, buffers, etc. it was possible to reduce the amount of data from several hundreds per minute to just 10 signals per minute.

Further reduction was possible by calculating several signal characteristics either scalar (such as frequency, energy, maximum amplitude, RMS value, etc.), or array (e.g. using functions such as autocorrelation, FFT, etc.). Classification based on those characteristics can separate potential AE signals from irrelevant emissions. This step of AE pattern recognition can reduce further the amount of data arriving for the identification and prediction stage, down to an average of 1/min or even 1 AE every 5min.

The statistics of the interarrival time, resulting after all preprocessing and cleaning of the AE signals, are the key values for selecting a processing algorithm and applying it on-line. Obviously each processing cycle should be completed before any new data is received.

3 THE IDENTIFICATION/PREDICTION ALGORITHM REQUIREMENTS

In order to use the advanced methods under consideration, the FCG/NDE models should be in a state-space form. Therefore they are written as a recursive State-Space model of the general form:

$$\begin{aligned} x(k+1) &= f\left[k, x(k)\right] + g\left[k, x(k)\right] w(k) \\ z(k) &= h\left[k, x(k)\right] + v(k) \end{aligned} \tag{1}$$

This model is suitable for all advanced algorithms, such as Kalman Filter (KF) or Extended Kalman Filter (EKF), and the state equation is compatible with all FCG semi-empirical laws. [7]. Depending on the NDT/NDE method applied (e.g. Acoustic Emission, Thermal Emission, Potential Drop, etc.), the measurement equation contains the relationship between monitoring data and crack growth.

For the more complex 'parameter identification' problem, we let θ be the vector containing all unknown or varying parameters, then the state is replaced by the augmented state $x_\theta(k) = \left[x(k) \mid \theta\right]^T$.

Another implementation is to consider the model with unknown parameters but without augmenting the state by the vector θ. In this model (16), θ is a random variable with known a-priori probability density $p(\theta/0)$.

$$\begin{aligned} x(k+1) &= f\left[k, x(k); \theta\right] + g\left[k, x(k)\right] \cdot w(k) \\ z(k) &= h\left[k, x(k); \theta\right] + v(k) \end{aligned} \tag{2}$$

The KF and EKF implementations are quite similar. Both the first order EKF algorithm for a nonlinear model (f, h) and the basic Kalman algorithm for linear model (F, H instead of f, h) are given by the following set equations:

$$
\begin{aligned}
&P(k/k-1) = F(k)\cdot P(k-1/k-1)\cdot F^T(k) + G(k)\cdot Q(k)\cdot G^T(k)\\
&K(k) = P(k/k-1)\cdot H^T(k)\cdot\left[H(k)\cdot P(k/k-1)\cdot H^T(k) + R(k)\right]^{-1}\\
&P(k/k) = \left[I - K(k)\cdot H(k)\right]\cdot P(k/k-1)\\
&\hat{x}(k/k-1) = f\left[k-1, \hat{x}(k-1/k-1)\right]\\
&\tilde{z}(k/k-1) = z(k) - h\left[k, \hat{x}(k/k-1)\right]\\
&\hat{x}(k/k) = \hat{x}(k/k-1) + K(k)\cdot\tilde{z}(k/k-1)
\end{aligned}
\tag{3}
$$

On the other hand, MMPA equations for the optimal state estimation and its variance are:

$$
\begin{aligned}
&\hat{x}(k/k) = \int \hat{x}(k/k;\theta)\, p(\theta/k)\, d\theta\\
&P(k/k) = \int\left[P(k/k;\theta) + \left\|\hat{x}(k/k) - \hat{x}(k/k;\theta)\right\|^2\right]\cdot p(\theta/k)\, d\theta\\
&p(\theta/k) = \frac{L(k/k;\theta)}{\int L(k/k;\theta)\cdot p(\theta/k-1)\, d\theta}\cdot p(\theta/k-1)\\
&L(k/k;\theta) = \left|P_{\tilde{z}}(k/k-1;\theta)\right|^{-1/2}\, \mathrm{e}^{-1/2\left\|\tilde{z}(k/k-1;\theta)\right\|^2\cdot P_{\tilde{z}}^{-1}(k/k-1;\theta)}
\end{aligned}
\tag{4}
$$

where, $x(k/k\,;\theta)$ και $P(k/k\,;\theta)$ can be calculated using a Kalman filter designed for each model with parameter θ. For discrete or discretised parameters, the integrals are replaced by sums. It is clear that the MMPA requires for its operation L sub-filters either Kalman or similar algorithms, each representing a different candidate model.

The above presented MMPA possesses several interesting properties:

1. Its structure is a natural parallel distributed processing architecture and hence it is more suitable to current computers clusters.
2. By breaking a large non-linear model to smaller sub-cases the algorithm has a much smaller dimensionality and hence much less architectural complexity.
3. Although computationally intensive, it works faster due to parallelism and hence it is much more appropriate for real-time applications.
4. It is more robust than any single filter as it is capable to isolate any diverging sub-filter. Numerous applications and simulations in the literature also show this.
5. The algorithm is well structured and modular and it is easy to implement and modify on any standard programming environment (e.g. MATLAB).

4 COMPUTATIONAL REQUIREMENTS

For compatibility reasons to other researchers' work, the computational requirements are calculated using the techniques presented in [10]. We calculate storage requirements and operations requirements based on the filtering algorithm equations and on the model dimensionality (n = state dimension, p = input dim., m = measurements dim.). The following table summarizes the required array operations (A,B,C: general, S,T,U: symmetric, w,y,z: vector)

Array Operation	Result	ADD	MULT	DIV	SQRT
A(n,m)+B(n, m)	C(n, m)	$n \cdot m$	-	-	-
S(n, n)+T(n, n)	U(n, n)	$0.5\cdot(n^2+n)$	-	-	-
w(n,1)+y(n,1)	z(n,1)	n	-	-	-
A(n, m)·B(m,p)	C(n, p)	$n\cdot(m-1)\cdot p$	$n\cdot m\cdot p$	-	-
A(n, m)·B(m, n)	S(n, n)	$0.5\cdot(n^2\cdot m + n\cdot m - n^2 - n)$	$0.5\cdot n\cdot m\cdot(n+1)$	-	-
A(n, m)·w(m,1)	z(n,1)	$n\cdot(m-1)$	$n\cdot m$	-	-
A(n,n)$^{-1}$	B(n,n)	$0.5\cdot(n^3+3n^2-2n)$	$0.5\cdot(n^3-n)$	2·n-1	n

Table 1 : Array Operations

The KF requirements are calculated from equations (3) and depend only on n, p & m. The EKF requirements in the simplest case are equal to the KF, but when dealing with augmented models (parametric), the model matrices are larger and more complex. A new quantity, the size (q) of parameter vector θ, is included in dimension n. The MMPA is also using an extra parameter, the number of the candidate models (S).

	Storage	Operations
K.F.	$0.5n^2 + 2.5n + 0.5m^2 + 1.5m + 2\max[n^2, nm] + \max[\frac{1}{2}(m^2+m), \frac{1}{2}(n^2+n)]$	MUL: $1.5n^3+1.5n^2+0.5m^3+1.5m^2-m+3nm+1.5n^2m+1.5nm^2+np^2+0.5n^2p+0.5np$ ADD: $1.5n^3-1.5n+0.5m^3-0.5m+nm+1.5n^2m+1.5nm^2+np^2+0.5n^2p-0.5np$ DIV : $2m-1$ SQRT: m
E.K.F. *(n=n+q)*	$0.5n^2 + 2.5n + 0.5m^2 + 1.5m + 2\max[n^2, nm] + \max[\frac{1}{2}(m^2+m), \frac{1}{2}(n^2+n)]$	MUL: $1.5n^3+0.5n^2+0.5m^3+1.5m^2-m+2nm+1.5n^2m+1.5nm^2+np^2+0.5n^2p+0.5np$ ADD: $1.5n^3-n^2-0.5n+0.5m^3+0.5m+1.5n^2m+1.5nm^2+np^2+0.5n^2p-0.5np$ DIV : $2m-1$ SQRT: m
M.M.P.A. *(S=models)*	$n^2+3n+0.5m^2+2.5m+S[2+n+\frac{1}{2}(n^2+n)+KF_{Stor}]$ (* KF_X = *corresponding KF/EKF quantity*)	MUL: $S[n^2+2n+0.5m^3+2.5m^2+7+KF_{Mul}]$ ADD: $S[n^2+3n+0.5m^3+m^2-0.5m+3+KF_{Add}]-n-1$ DIV : $S[2m+3+KF_{Div}]$ SQRT: $S[m+1+KF_{Sqrt}]$

Table 2 : Algorithm Computational Requirements

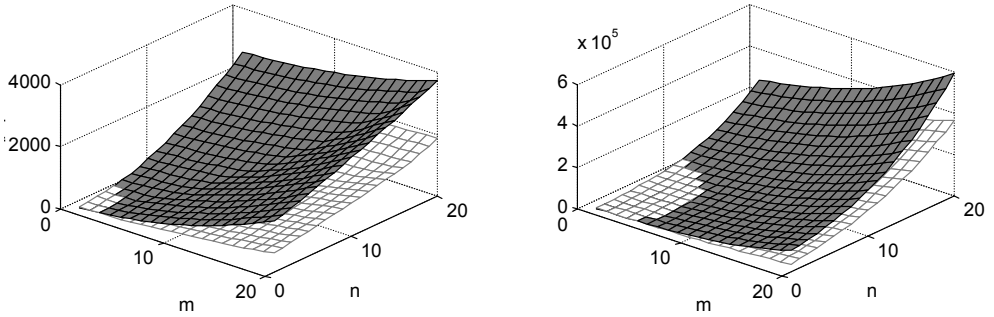

Figure 1. Comparison of memory (left) & operations (right) requirements, for EKF (grid) & MMPA (dark) algorithms for S= 2, q=3, n=1…20 & m=1…20.

The advanced MMPA algorithm is more complex, requires more operations and, normally, it is more time consuming. Detailed analysis of the computational requirements for EKF and MMPA show that the complexity and the time consumption depend on the problem dimensionality and on the partitioning capabilities of the algorithm. It is also clear that parallel implementation is essential for the MMPA algorithm. The internal structure of MMPA is suitable for parallel implementation as all sub-filters are decoupled and can be implemented by separate processors. As a result the total number of operations may be higher but the time requirements are dramatically reduced to those of a single EKF filter plus a small overhead for the MMPA equations (4).

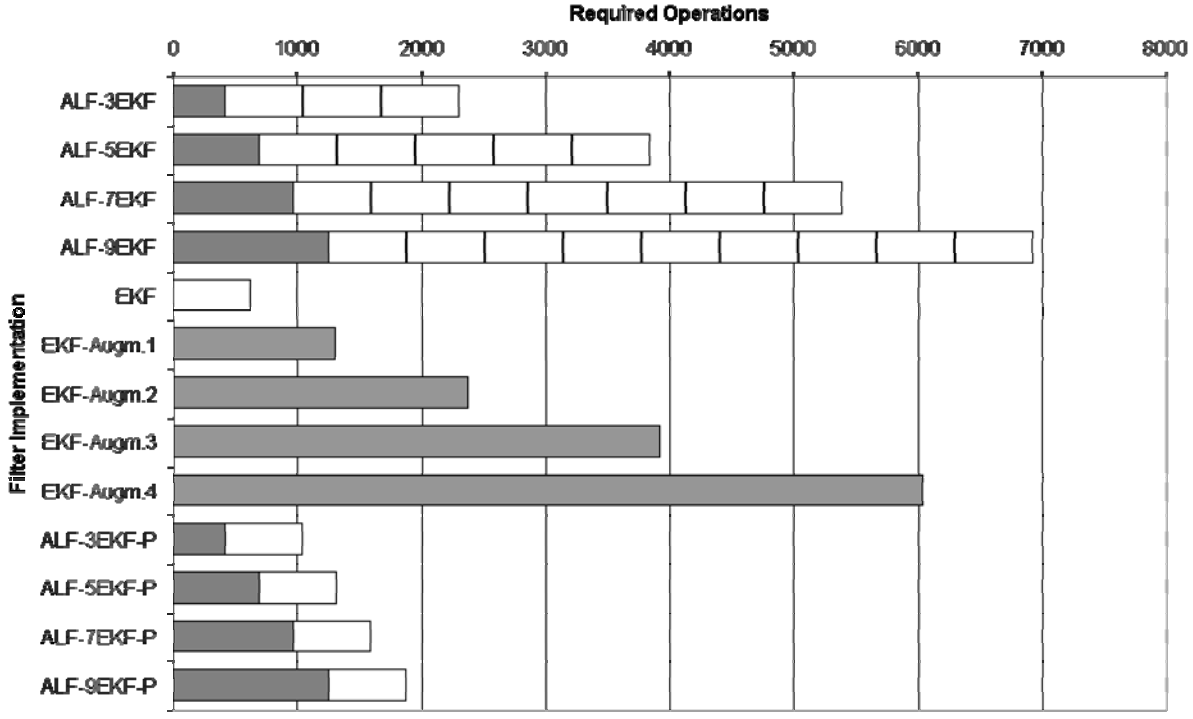

Figure 2. Computational requirements for MMPA(ALF), simple EKF/KF, EKF(with augmented state, q=1-4) and MMPA(ALF-P) in parallel implementation (S=3-7).

Based on this analysis, a number of EKF & MMPA implementations were analyzed for the state-space model of Eq. 1 & 2. For each case studied, the required operations per iteration are calculated and shown in Figure 2. The EKF time requirements increase exponentially when its state is augmented by 1 to 4 parameters. The MMPA time requirements also increase with the number of EKF sub-filters (*S*), but only when the filter is implemented sequentially. If MMPA is implemented in parallel, the time requirements are minimal and the filter overhead is increasing linearly. The requirements of a simple KF/EKF (without augmentation) are also shown for reference.

The real time implementation of the above algorithms is done in the MATLAB environment on a Pentium 4. The MMPA with 9 sub-filters completed its iteration in a few seconds. The algorithm can perform even faster using compiled C of Fortran code. In addition, by implementing the algorithm in parallel the above speed is multiplied even further. This speed is more than adequate for the AE signals data rate presented earlier.

5 RESULTS

The adaptive nonlinear algorithm MMPA is applied to the FCG problem for estimation and residual lifetime prediction. The MMPA is tested against a classic EKF with augmented state. From the point of view of Accuracy and efficiency, the results show that although both predictors do converge to the actual time to live, MMPA can do it more accurately and much sooner than the EKF. More precisely, MMPA converges sooner to the correct predictions requiring fewer measurements, and leaving more time for reaction than the EKF. This is even more important when the algorithms are working on-line. In addition MMPA is more robust than a single EKF as it incorporates the mechanism to isolate any diverging sub-filter, an attractive feature for on-line systems. From the point of view of computational requirements, MMPA is generally a more demanding algorithm than EKF, but, due to its partitioned structure, MMPA is suitable for parallel implementation. Parallel MMPA overcomes easily its complexity as it performs faster than the corresponding EKF. Its speed in a simple sequential implementation shows that it can be easily implemented on-line for the problem of AE signals data and for higher data rates.

REFERENCES

[1] Keller, E., Ray, A. (2003), "Real-time Health Monitoring of Mechanical Structures", Structural Health Monitoring SHM, Vol. 2(3): 191-203, Sage Publications 2003.

[2] Hunt, J. Victor (2000), "Nondestructive Evaluation and Health Monitoring of Highway Bridges", Ph.D. Dissertation, Dept. of Electrical and Computer Engineering and Computer Science, University of Cincinnati, USA.

[3] Lucia, A. C. (1985), "Probabilistic Structural Reliability of PWR Pressure Vessels". Nucl. Eng. Design, *87*, 35-49.

[4] Lainiotis, D. G. (1984), "Adaptive Dynamic Modeling, Risk Assessment, Failure Detection and Correction", In *Advances in Structural Reliability*, Proceedings of the Advanced Seminar on Structural Reliability, JRC-Ispra (VA) Italy, 4-8 Jun 1984; Lucia, A. C., Ed.; D. Reidel Pub Co: Dordrecht, 1984.

[5] Kaminski, M. (2002), "On Probabilistic Fatigue Models for Composite Materials", Int. Journal of Fatigue, *24*, 477-495.

[6] Moussas, V. C., Solomos, G., Lucia, A. C. (1994), "A General Method for Raw Fatigue Crack Growth Data Processing and Structural Reliability Assessment", *In Computational Stochastic Mechanics, Proceedings of the 2nd International Conference on Computational Stochastic Mechanics, Athens, Greece, June 12-15, 1994; Spanos, P.D., Ed.; Balkema: Rotterdam*.

[7] Moussas, V.C., Katsikas, S.K., Lainiotis, D.G. (2005), "Adaptive Estimation of FCG using Nonlinear State Space Models", *Stochastic Analysis and Applications,* 23, 1-18.

[8] Lainiotis, D. G. (1974), "Partitioned Estimation Algorithms, I: Nonlinear estimation", Information Sciences 1974, *7*, 203-235.

[9] Lainiotis, D. G. (1976), "Partitioning: A Unifying Framework for Adaptive Systems, I: Estimation", IEEE Proceedings 1976, *64* (8), 1126-1143.

[10] Katsikas, S. K., Likothanassis, S. D., Lainiotis, D. G. (1991), "On the Parallel Implementations of the Kalman and the Lainiotis Filters and their Efficiency", Signal Processing, 18, 289-305.

[11] Moussas, V.C. (1989), "Acquisition and Analysis of Acoustic Emission Waveforms from PWR Pressure Vessel Fatigue Tests", JRC Report TN I.89.72-1709/89, Jun. 1989.