Jürgen Broede, Hugo Pfoertner

Advanced Algorithms Design and Implementation in On-board Microprocessor Systems for Engine Life Usage Monitoring

Entwicklung fortschrittlicher Rechenverfahren zur Lebensdauerüberwachung von Triebwerken und deren Implementierung in bordgestützte Mikroprozessorsysteme

Summary

Advanced engine life usage monitoring requires close equivalence between mechanical design and monitoring algorithms in order to achieve sufficient accuracy. Usual and abnormal engine operation must be covered as well as input signal failure and other disturbance factors. Thus, algorithms have been developed which are sufficiently accurate but simple enough to satisfy the constraints due to processor speed and storage size. Those algorithms are embedded in a frame controlling the functional sequence of the monitoring procedure. Algorithm development also includes the conversion into a suitable form for processing, where accuracy is maintained when using integer arithmetic, and last not least the verification and validation of the implemented software.

Zusammenfassung

Eine fortschrittliche, hinreichend genaue Uberwachung des Lebensdauerverbrauchs fur Triebwerkskomponenten kann nur durch eine weitgehende Äquivalenz der Auslegungs- und Überwachungsalgorithmen erreicht werden. Normaler und gestörter Triebwerksbetrieb, aber auch Fehler in den Eingangssignalen und andere Störfaktoren müssen korrekt behandelt werden. Ziel der Entwicklung ist eine an die verfügbare Rechengeschwindigkeit und Speichergröße angepasste Vereinfachung unter Beibehaltung der erforderlichen Genauigkeit. Die Lebensdauerverbrauchsrechnung ist in eine geeignete Ablaufsteuerung einzubinden. Die Entwicklung umfasst ausserdem die Umsetzung in eine zur Verarbeitung geeignete Form, wobei trotz der Verwendung von Integer-Arithmetik kein Genauigkeitsverlust entsteht, und nicht zuletzt den Nachweis der korrekten Funktion der eingebauten Software.

1. Introduction

Safe and economic service of modern aircraft engines requires life usage monitoring systems based on advanced algorithms converted into suitable forms for realtime processing in on-board microprocessor systems.
Parts to be monitored are parts with limited service lives which are classified as “fracture critical parts”. These parts experience basically centrifugal stresses due to high rotational speeds and additionally remarkable amounts of thermal stresses due to steady state and transient temperature gradients. Consequently, proper monitoring requires not only the consideration of actual speeds, but also actual transient temperature distributions.

2. Equivalence between mechanical design and monitoring

The most accurate procedures for the estimation of performance parameters, temperature and stress histories are the design procedures. Since these are based on highly complex computer programs running on mainframe computers, they cannot be used directly in on-board microprocessor systems. Thus, algorithms have to be developed which represent optimal approaches to the design procedures, but satisfy the constraints due to processor speed and storage capacity.
The best way to arrange such algorithms is to achieve a close equivalence between mechanical design procedures and monitoring algorithms, which means that for each major step or module of the design procedure an equivalent monitoring module should be built.
An additional requirement is that these algorithms must be able to cover normal and abnormal engine operating conditions. That means that engine start, usual manoeuvring, engine shut down at arbitrary operating conditions and cooling down of the stopped engine (in order to get a suitable estimate of the initial conditions for an engine restart) must be covered as well as overtemperature, overspeed and windmilling conditions.

3. Algorithm design route

The MTU design route for life usage algorithms starts from so called ”Basis Data”. These " Basis Data” are performance parameters, temperature distributions within the considered components, stresses and strains at the critical locations of the monitored parts and damage calculation and accumulation rules, which are produced by the design procedures when applied to a number of mission profiles or parts of them.
The general structure of the algorithms is based on the physical behaviour which is described by

mathematical expressions, sets of differential equations, etc. This structure is formed by general analytical or numerical solutions to the physical behaviour, but tailored to the actual real components.

This tailored algorithms structure now contains a number of unknown parameters. In the following step, these parameters are optimized so to reproduce the “Basis Data”. The optimization criteria are

- ☐ to minimize the deviations of the monitoring data from the “Basis Data”
- ☐ finally to minimize the error of the life consumption result.

In general, the optimization procedure is not limited to the parameter optimization but also includes variations of the formulation and tailoring of the numerial results as well as the dependency of the parameters on engine operating conditions.

4. Accuracy of the algorithms

The life usage monitoring algorithms shall be closely equivalent to the design procedures and therefore they shall be judged against data derived from the design procedure. Thus, the accuracy as given in this presentation is not an absolute accuracy, but only the relative accuracy when comparing the monitoring data with the “Basis Data”. Inaccuracies of the “Basis Data” (e.g. accuracy of the finite element models) are not considered.

Performing the monitoring tasks, the algorithms use measured input signals.Inaccuracyresulting from inaccurate input parameters or sensor failure is not considered here. Nevertheless, provisions are made

- ☐ to check the input signals and to correct them, if logically possible
- ☐ to check the results and to correct them, if possible
- ☐ to interpret faults.

The procedures applied are built in such a way that conservative life usage quotation is ensured, and while doing so the processor systems is protected against running in undefined conditions possibly resulting from invalid input data.

5. Functional sequence of the monitoring procedure

The functional sequence of the monitoring procedure is based on an engine running history. Three modes are identified which are separated by special criteria. The initial mode commences with monitoring unit power-on and lasts till the start criterion (engine start) is sensed. Between start criterion and end criterion (engine shut down on ground or full stop landing with a stopped engine) the main mode is active. The end criterion initiates the final mode.

6. Algorithm design

6.1 Performance algorithms

Performance algorithms are used to derive performance parameters necessary for subsequent calculations (such as gas temperatures and pressures for the blade passages and the cooling air flows, torques) from the measured flight and engine parameters. These algorithms cover steady state and transient engine operation. The simplified calculation of the thermodynamic cycle in the engine uses compressor and turbine maps, gas properties and thermodynamic relationships. Methods used are mainly table interpolation and the evaluation of mathematical formulas.

The accuracy of the algorithms depends on the data sources, what means

- ☐ assumptions for analytical evaluation,
- ☐ scatter of measured data, interpolation method,

but in principle the algorithms are identical with those used in the design process and therefore can be considered as exact.

6.2 Temperature algorithms

The temperature algorithms serve to calculate the transient metal temperature distribution of components containing parts with areas to be monitored. An example for temperature profiles is given in Fig. 1. The algorithms are different for the three modes described above.

In the initial mode the initial temperature distribution is calculated. For this purpose the temperature distribution of the last engine shut down and the time elapsed since then is used.

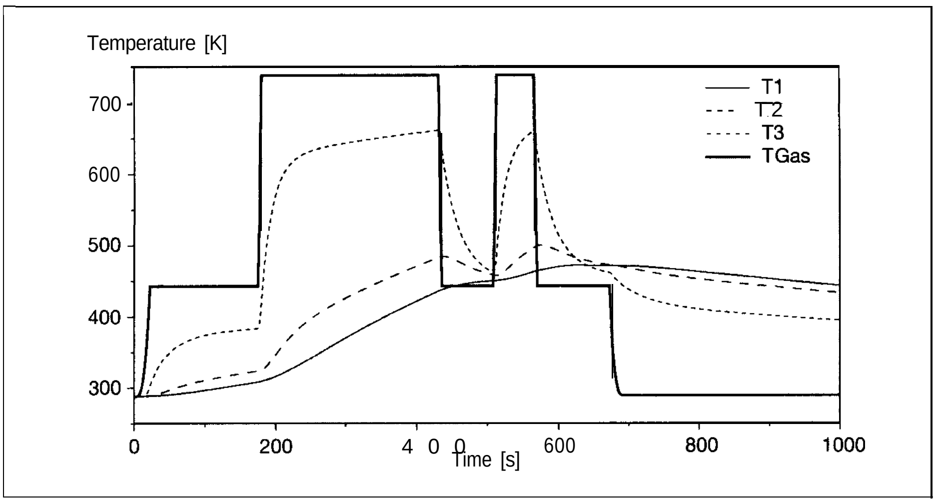
In the main mode the temperature distribution of each time step is calculated. The temperatures of the current time step are derived from the temperature distribution of the previous time step and coefficients describing heat transfer and conduction calculated from the operating conditions of the actual time step.

In the final mode the temperatures of the shut down stress peak are calculated using the temperature distribution and the performance data and rotor speeds of the last main mode time step.

The algorithm used to compute the transient temperatures for all three modes is a transfer matrix method. The algorithms are built in such a way that the steady state temperatures are exact whilst the transient temperatures are more accurate than 10 degrees C.

The entries of the temperature transfer matrix describe heat transfer between hot gas, cooling air and surfaces of components, heat conduc-

Fig. 1: Calculated temperature *profile*
Bild 1: Berechnete Temperaturverläufe



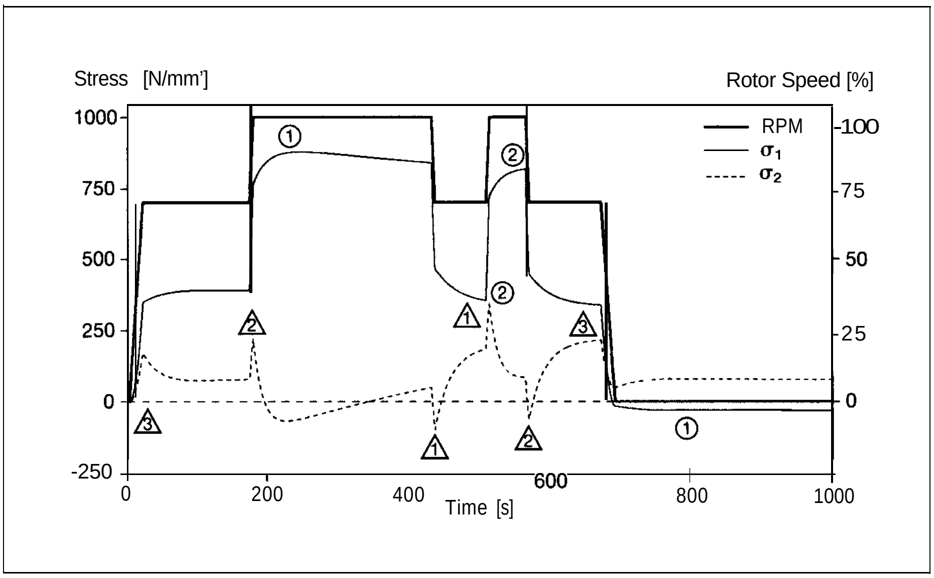


Fig. 2: *Calculated stress profiles*
Bild 2: *Berechnete Spannungsverläufe*

tion and radiation. Due to the strong dependence on operating conditions, which usually is described by some sort of power law, the matrix entries may vary by an order of magnitude for different operating conditions.
In contrast to the design calculations, where variable time steps are used to optimize accuracy and to reduce computation costs, fixed time steps are used for the current MTU monitoring algorithms. This results in large variations of the temperature changes per time step between transient and steady state operation of the engine. The range of changes per time step is from some 10 degrees after slam accelerations to 1/100 degrees in the asymptotic phase.

Both the representation of matrix elements and the large range of temperature changes place serious demands on the accurate transformation of the method into a suitable form for on-board processing.

6.3 Stress algorithms

The total stress at each critical area is calculated as the sum of thermal, centrifugal and additional stresses. The thermal stress is a function of the current temperature distribution within the component. The centrifugal stress is related to the squared rotor speed and the additional stresses depend on gas pressures, bolt clamping forces, etc.
Examples of total stress profiles are shown in Fig. 2. The fraction of thermal stress depends strongly on the current point in the time history and on the location within the component (Fig. 3).
The algorithms can be fitted in such a way that the stresses for particular conditions (e.g. maximum and minimum stress) are exact and for other conditions more accurate than 2 % of the min-max stress range.

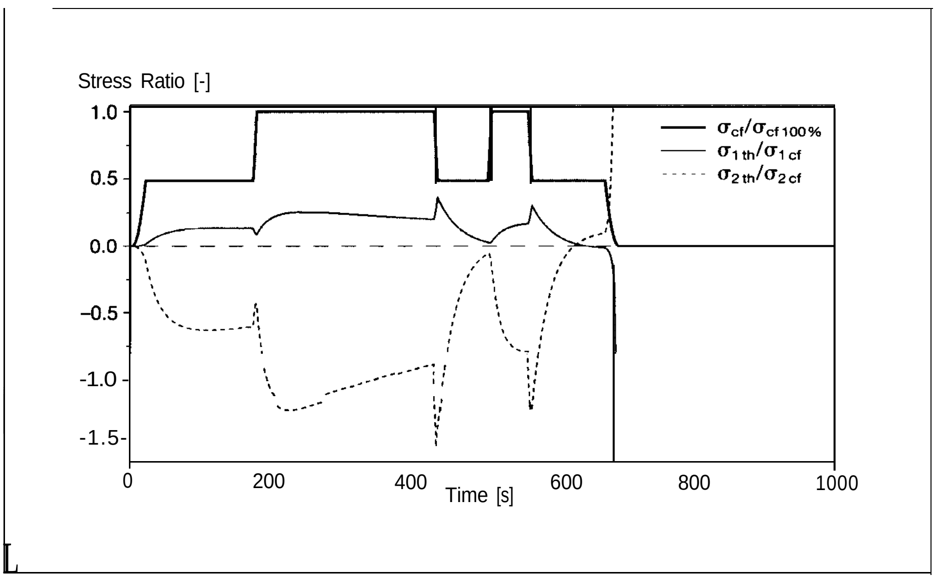
6.4 Damage algorithms

A number of damage mechanisms are known, but monitoring is limited to those mechanisms which have been identified as life limiting. For current engine design, these mechanisms are LCF and creep.
For LCF-monitoring the stress histories are broken down into stress cycles using a Rainflow Algorithm. Maxima and minima of the cycles found are marked in Fig. 2. Only those cycles are marked, which have a non-vanishing contribution to LCF-damage. The corresponding temperatures are also acquired. The fatigue per cycle is calculated with respect to the stress range and mean stress as well as the temperature and material strength. The fatigue increments of all cycles are accumulated to the overall fatigue life consumption.
For creep monitoring the creep increments of each time step are calculated using characteristic stresses and temperatures for this time step. All creep increments are accumulated to the overall creep life consumption.
Since the damage assessment procedures for monitoring purposes are the same as for engine design, these algorithms are considered as exact.

7. Conversion into integer arithmetic

The need for the conversion of monitoring algorithms comes from the fact, that the sophisticated floating point processing capabilities of mainframes are not available in the standard microprocessor systems currently used for engine monitoring units.
The reasons, why the current MTU monitoring systems do not use floating point processors are manifold: Non-availability of MIL-Standard hardware at the time the current systems were planned, financial reasons, assumed low complexity of algorithms, problems to integrate a monitoring processor into an existing processor system with older techno-

Fig, 3: Ratio of thermal stresses to centrifugal stresses
Bild 3: Verhältnis der thermischen Spannung zur Fliehkraftspannung



logy, unawareness that software life cycle costs can be dramatically reduced by avoiding floating-point to integer conversion.

If there is no floating point processor available, all the algorithms have to be reformulated to accomplish the desired functions by only using 16-bit and 32-bit integer numbers. Using standard software emulated floating point arithmetic (well known to PC-users) slows down the speed by a factor of about 10 compared with hardware arithmetic and therefore is not a viable way to implement numerical algorithms in on-board computers. The process of converting given algorithms into the required form is not standard and depends strongly on the complexity of the calculation method. So the first step to facilitate the conversion is a clear and simple formulation of the monitoring algorithms. However, there are some common principles and basic procedures, which are worthwhile to be mentioned here to throw some light on the conversion process.

7.1 Representation of functional relationships

Since there is no mathematical library available, a few mathematical standard functions necessary for the monitoring algorithms have to be provided in a suitable form. LOG, EXP and SQRT have been implemented with proper scalings using similar techniques as in mainframe libraries /3/ (e.g. reduction to a standard interval, table interpolation, approximation by polynomials, Newton iterates). To optimize run times, parts of these routines are written in assembler language.

Other functions or relationships between physical parameters, the computation of which is too expensive or cumbersome in integer arithmetic may be represented in two different ways.

One way is to find simplified expressions by series expansions or by least squares fits. Preferred forms are low order polynomials or simple power laws. The other approach is using piecewise linear interpolation. Storage requirements can be minimized by selecting the breakpoints according to the local curvature of the original function under the constraint of a maximum admissible approximation error.

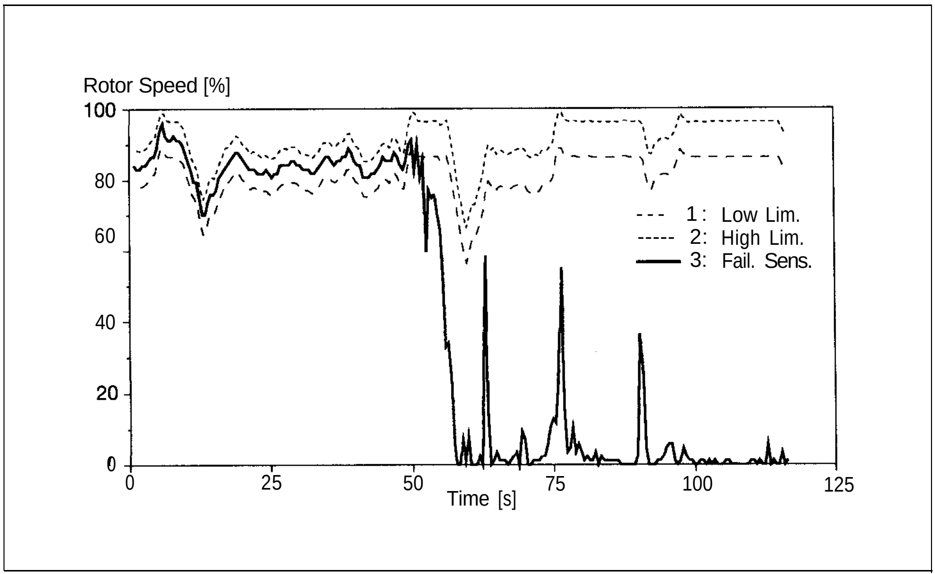


Fig. 4: Parameter outside tolerance band
Bild 4: Parameter außerhalb des Toleranzbandes

Since linear table interpolations are involved in many parts of the monitoring algorithms, especially in the calculation of gas temperatures and damage increments, the optimization of the interpolation routines with respect to run times /4/ has a non-neglible effect on the overall performance of the complete monitoring algorithm.

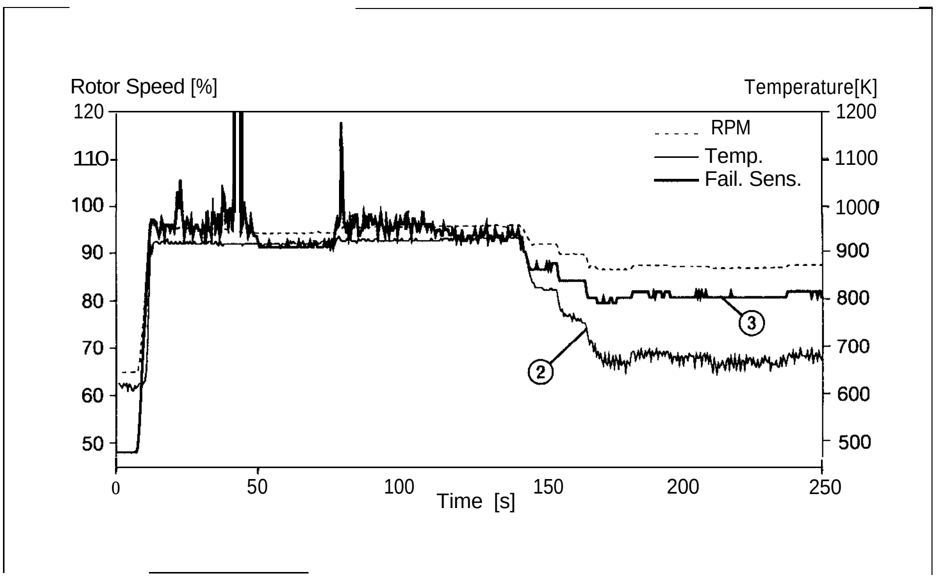
Table interpolations are also used to avoid iterations normally found in some of the life estimation procedures. Generally, iterative computations, particularly those with questionable convergence should be avoided for reasons of testability.

7.2 Scaling of input parameters and results

The measured input parameters can normally be represented as 16-bit values with no loss of accuracy. After the physical conversion done by counters or A/D converters a scaling transformation is applied to the input values to convert them into program variables with a unified internal scaling. Some examples of reasonable scalings are 16/K for temperatures, 256/% for rotor speeds or 128/kPa for inlet total pressures. Powers of 2 are preferred scaling factors, because a recourse to the original physical values (e.g. for display purposes) can be achieved by inexpensive shift operations.

Results of monitoring programs can be roughly divided into two cate-

Fig. 5: Comparison of 3 signals
Bild 5: Vergleich von 3 Signalverläufen



gories: Results, which are overwritten, when a new flight or engine run begins and accumulating accounts. Diagnostic information for one engine run can normally be cast into 16-bit results, whereas the accumulation of LCF cycles over long periods without producing overflow requires 32-bit output values. If LCF damage is measured in units of "reference cycles", a typical scaling for one reference cycle is 2048.

7.3 Scaling of internal variables

The engine monitoring systems currently in service /2/ are equipped with 16-bit microprocessors (e.g. Motorola 68000). To avoid expensive 32-bit operations the scaling of internal variables has to be chosen according to the range and the required accuracy of intermediate results. Overflows, which would produce completely confused results have to be avoided in any case. Checking each operation for overflow in the microprocessor program would produce an intolerable overhead. Therefore this task has to be accomplished during the development phase by careful examination and extensive testing of the program modules on the mainframe computer with synthesized and recorded data.

The need for avoiding overflows places a rigorous constraint on the selection of internal scaling factors. In certain situations no fixed scaling factors can be found which fulfil simultaneously the requirements for accuracy and range. Variable shifts with later reversal or a temporary change to 32-bit arithmetic is used to circumvent this bottleneck.

7.4 Signal checks

To obtain "good" scaling factors and to avoid overflows rigorous assumptions have to be made for the behaviour of input signals. Accepting erroneous input data without proper correction could not only lead to wrong results, but also to undefined states in the microprocessor and the whole monitoring system. Standard signal checks /1,2/ (range, rate-of-change, parameter interrelationships) are applied to the parameters entering the calculation routines. A situation caused by a loose connector which is handled correctly by these checks is shown in Fig. 4.

The discrimination between faulty and correct signals is sometimes rather diffcult. Fig. 5 shows the comparison

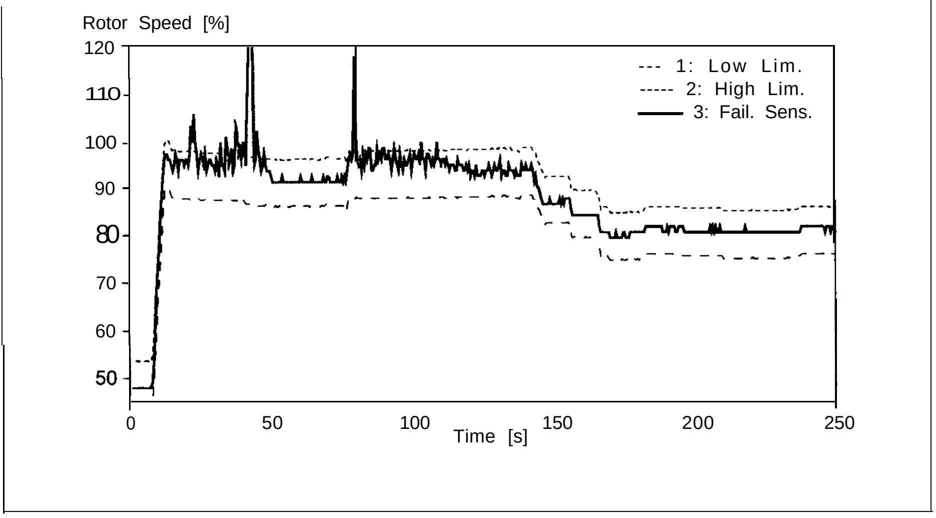


Fig. 6: Insufficient *check for sensor failure*
Bild 6: Prüfung, die zum Erkennen von Sensorfehlern nicht ausreicht

Fig. 7: Method 1, without rounding, 16 bit-accumulation
Bild 7: Methode 1, ohne Rundung, 16 Bit-Akkumulation

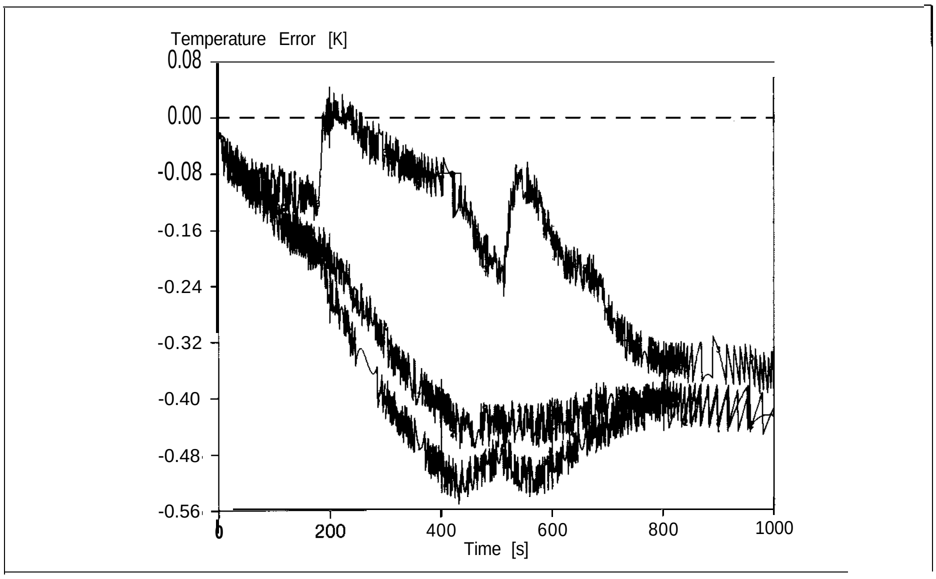
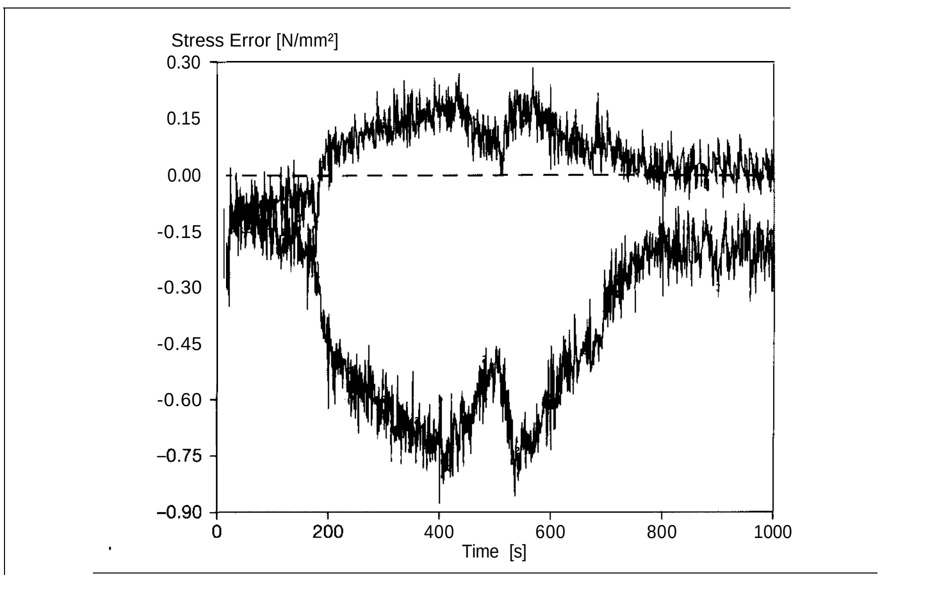


Fig. 8: Stress error of method 1
Bild 8: Spannungsabweichung bei Methode 1



of 3 signals, which are expected to have a closely similar dynamic behaviour. But only the peaks and the scatter of signal 3 are a consequence of a faulty sensor, whereas the noise in signal 2 in the right part of the figure comes from a variable resolution over the measuring range. As shown in Fig. 6 only the most prominent peaks are eliminated by a tolerance check. The inclusion of the signal peaks lying within the tolerance band into the calculation of stresses can lead to a dramatic increase in computed life consumption. The correctness of the results can only be judged at the end of the mission by comparing measures for the dynamics of closely correlated signals or by checking the calculated damage against tolerance bands deduced from independent data sources.

7.5 Accuracy of converted algorithms

The results shown in Figs. 1 and 2 were produced by a mainframe computer program using 32-bit floating point arithmetic. The temperature and stress histories are typical for thermally highly loaded engine disks. This program was the basis for 4 different versions of integer programs. Only the algorithms for the calculation of metal temperatures and stresses were investigated, all other modules remained the same.

In Method 1 the temperature changes are accumulated using 16-bit arithmetic. Temperature residuals below 1/16 K were accumulated over several time steps. The elements of the temperature transfer matrix are computed in a straightforward manner by summing the influences of conduction and temperature transfer. Fig. 7 shows a time history of the differences between the temperatures computed by the floating point program and those computed by the integer program, Fig. 8 shows the corresponding stress differences. In an attempt to find the reason for the increasing errors in the first 400 seconds, it was concluded that the low resolution of the temperature variables should be increased.

For Method 2 it was decided to use a temperature scaling of 1048576/K and 32-bit temperature variables and increments. The results of this action are shown in Fig. 9. Surprisingly, the trend of the temperature errors remained the same as in Fig. 7 and the expected dramatic reduction of the errors did not take place. The maximum absolute error was only reduced from 0.54 K to 0.35 K which is a poor result for a double precision calculation.

By an analysis of intermediate results it was found, that the major influence on the accumulation of errors did not come from an insufficient scaling of the temperatures, but from rounding errors in the computations of the transfer matrix. The corresponding program module was revised by introducing rounding operations and changing the order of execution for some sensitive operations.

The result is shown in Fig. 10. The maximum absolute error until engine shut down is reduced to 0.10 K. After shut down the error of all 3 temperatures are constantly increasing. This is caused by very small entries of the transfer matrix, with larger relative errors in their integer representation.

Since the influence of the precision of the matrix elements was found to be decisive for the approximation quality, it was decided to return to a 16-bit accumulation of the temperature increments with the results of Figs. 11 and 12. The temperature errors are no longer smooth, but their time history has the same form as in Fig. 10 with superimposed fluctu-

Fig. 9: Method 2, without rounding, 32 bit-accumulation
Bild 9: Methode 2, ohne Rundung, 32-Bit-Akkumulation

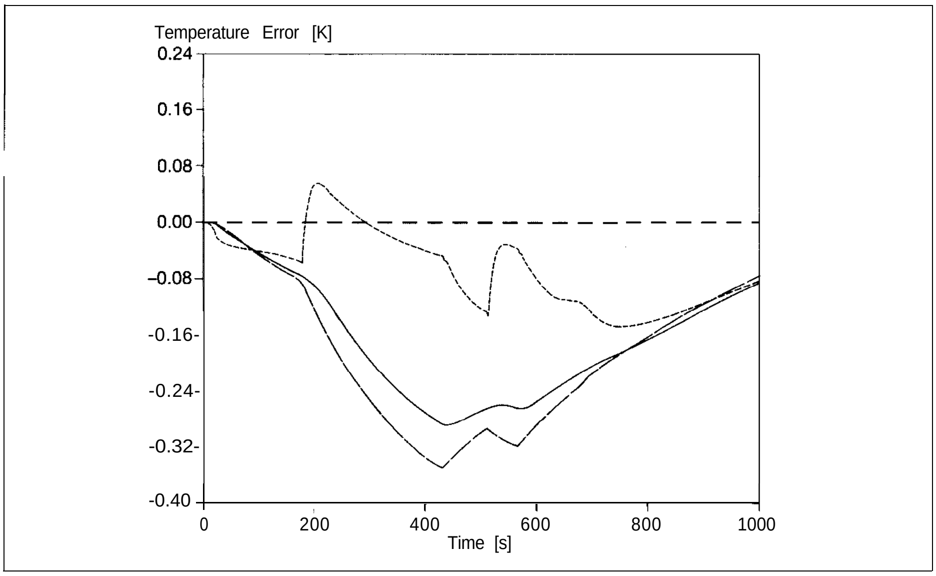
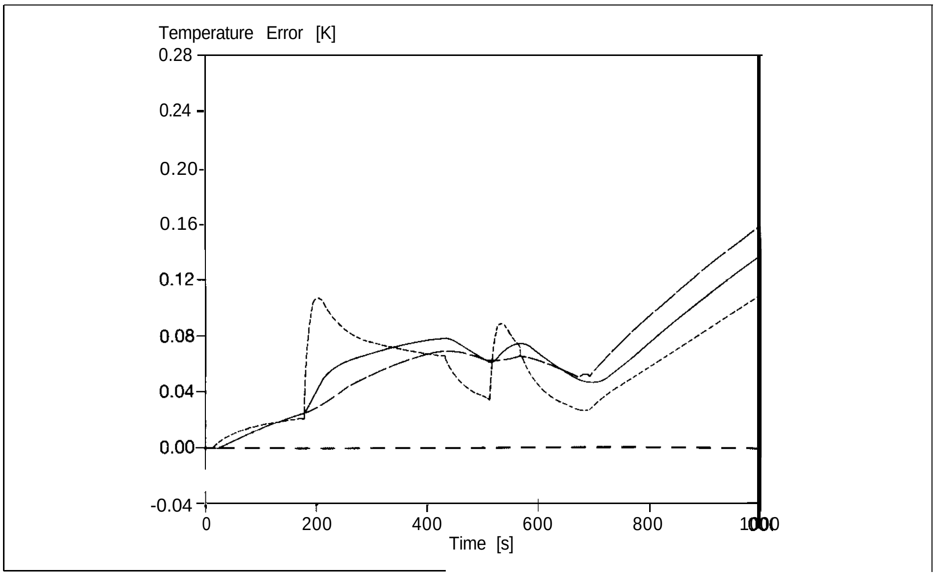


fig. 10: Method 3, with rounding, 32 bit-accumulation
Bild 10: Methode 3, mit Rundung, 32-Bit-Akkumulation



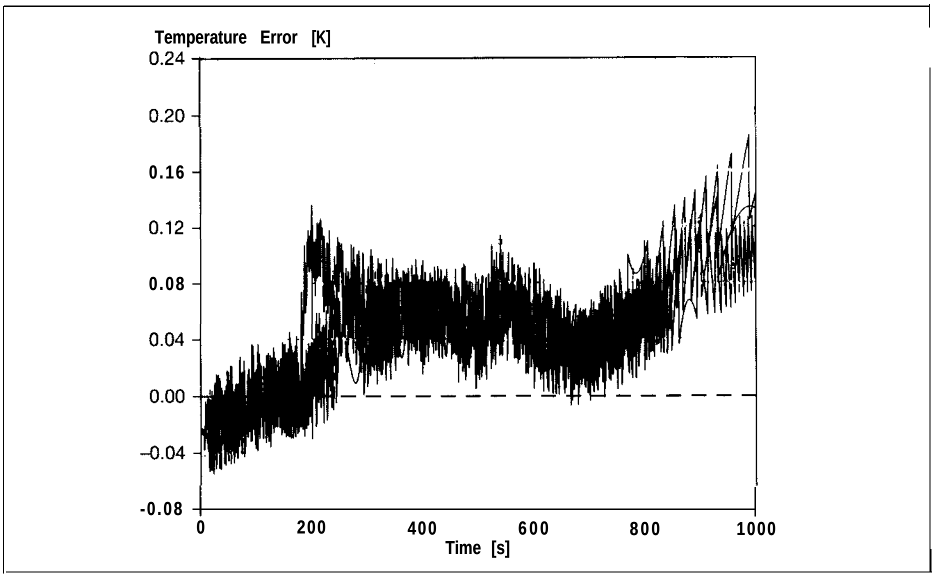


Fig. 11: Method 4, with rounding, 16 bit-accumulation
Bild 11: Methode 4, mit Rundung 16 Bit-Akkumulation

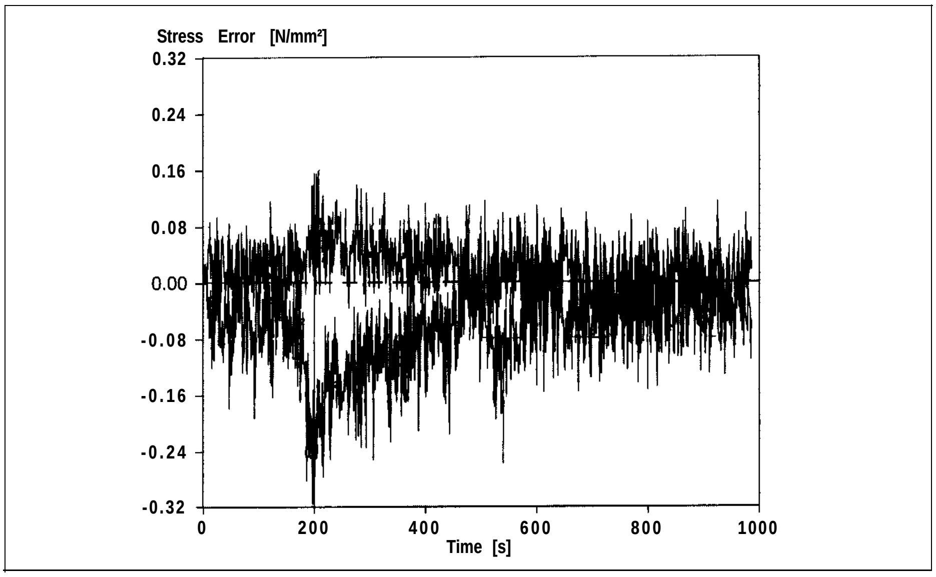


Fig. 72: Stress error of method 4
Bild 12: Spannungsabweichung bei Methode 4

ations, which are a consequence of the 16/K-scaling of the output temperatures. Still stronger fluctuations are found in the plot of stress differences. These fluctuations in the stress histories are also present, if the smooth temperatures of methods 2 and 3 are fed into the stress computation module.
To avoid buffer overflows in the cycle extraction modules, modified rainflow algorithms are used, which guarantee the timely elimination of cycles resulting from bit fluctuations in the input signals or in the stress computation.
Comparing the LCF-damage calculated with the original floating point method and the results of the integer methods, the maximum difference for all 4 methods is below 0.25%.

8. Conclusions

The accuracy of life consumption monitoring for critical engine components is limited by the approximations made in the conversion of design procedures into monitoring algorithms and by the quality of the input signals. Even with today's 16-bit microprocessors and with integer arithmetic the inaccuracies introduced by the conversion into a microprocessor program are by one order of magnitude lower than those introduced by the mathematical models.

In service experience proves that algorithm design and conversion methods fulfil the requirements properly. No problems have been raised concerning algorithm development and implementation.

9. References

/1/
SAE Committee E-32
Guide to Life Usage Monitoring and Parts Management for Aircraft Gas Turbine Engines. Aerospace Information Report AIR 1827
Warrendale, PA, SAE, 1988

/2/
Broede, J.
Engine Life Consumption Monitoring Program for RB199 Integrated in the On-board Life Monitoring System.
AGARD PEP 71th Symposium, Engine Condition Monitoring - Technology and Experience, 1988, Quebec, CP No. 488, pp. 9.1-9.10

/3/
Anlauff, H.
Beschleunigung der Berechnung mathematischer Funktionen durch Mikroprogrammierung. Informationstechnik it 31 (1989) No. 4, S. 291-296

/4/
Bentley, J.
Programming Pearls: Code Tuning.
Comm. of the ACM 27 (1984) No. 2 pp. 91-96.

Acknowledgement

The different versions of the integer algorithms were prepared by Dr. W. Möhres. His help is gratefully acknowledged by the authors.

This paper was presented at the 15 th Symposium on "Aircraft Integrated Monitoring Systems" on September 12-14, 1989 in Aachen.

Dieser Beitrag wurde auf dem 15. Symposium über "Aircraft Integrated Monitoring Systems" vom 12.-14. September 1989 in Aachen veröffentlicht.