Stochastic Reconstruction of Loading Histories from a Rainflow Matrix

Klaus Dreßler, TecMath GmbH, Sauerwiesen 2, D-67661 Kaiserslautern
Michael Hack, Univ. Kaiserslautern, P.O.Box 3049, D-67653 Kaiserslautern
Wilhelm Krüger, TecMath GmbH, Sauerwiesen 2, D-67661 Kaiserslautern

Abstract

This paper is devoted to the mathematical description of the solution of the so–called rainflow reconstruction problem, i.e. the problem of constructing a time series with an a priori given rainflow matrix.

The algorithm we present is mathematically exact in the sense that no approximations or heuristics are involved. Furthermore it generates an uniform distribution of all possible reconstructions and thus an optimal randomization of the reconstructed series. The algorithm is a genuine on–line scheme. It is easy adjustable to all variants of rainflow such as symmetric and asymmetric versions and different residue techniques.

1 Introduction

The estimation of the lifetime of a developed part in industry has to include both physical tests and the application of numerical tools. On one hand the results of calculations just based on the theory and algorithms which are known today are not reliable enough if they are used as the only test. On the other hand physical tests for any part in any stage of development are much too expensive.

What is the typical reason for the failure of a part in a car? The parts are usually not destroyed by one large load, but by the accumulation of the energy dissipated by many (typically several millions) hysteresis loops. It is assumed that the damage of a loop is rate–independent. This means the frequency of the oscillations is not taken into account. The damage induced by the individual loops is accumulated using Miner's rule. (see [13])

It is well known, both from practical experience and from theoretical reasoning, that rate–independent, range–oriented "counting methods" are the right approach to fatigue oriented analysis of time series. Such methods capture the relevant aspects of elasto–plastic loading much better than e.g. the common schemes of classical mathematical spectral analysis.

It should be mentioned that the motivation for data reduction schemes in fatigue analysis is not just the reduction of data (in the sense of storage saving) as suggested by the direct meaning of the word. The main point, however, is the concentration on the relevant information by an intelligent filtering, i.e. by omitting the immense mass of data, having no effect on damage accumulation. This allows both, an effective modular use of modern numerical damage evaluation techniques and the reorganization of test drive data for test stand experiments. For the latter it is most desirable that manipulations like superposition and extrapolation can be performed directly on the reduced data–sets, and it is absolutely essential that a stochastic on–line reconstruction of a time series with given reduced data–sets is possible.

The enormous practical impact of such methods, especially in the automotive industry, is discussed in [9] and [8].

From the point of view of damage analysis it is well accepted that the rainflow method is the optimal rate–independent data–reduction scheme for one–dimensional load histories (compare e.g. [14]). This is mainly due to the fact that the rainflow method counts the nested hysteresis loops in the stress–strain diagram. More generally, rainflow captures all the complicated material memory mechanisms of hysteretic material laws (like Masing plus memory hypotheses).

Furthermore there are strong arguments in favor of rainflow from a practical point of view: All important one–parameter counting methods, like range–pair, level crossing and the various peak countings are sub–schemes of rainflow. Thus all the practical experience connected with those schemes can be used in rainflow–based analysis.

In spite of all these arguments, in signal generation and simulation there is still much preference for Markov simulation or spectral schemes. A very popular mixture between both is the Kowalewski (so–called Gauss–) process, a special Markov simulation controlled by a one–dimensional irregularity parameter representing special spectral information (compare [11]). We are convinced that this is just due to the fact, that rainflow based reconstruction or simulation algorithms are not well enough understood; indeed, they are conceptionally harder and look a little bit more complicated than Markov or Gauss simulations.

The rainflow reconstruction algorithm presented here is very easy to use and yet has no disadvantage against Markov simulation, e.g., in performance and reliability. It is mathematically exact in the sense that there are no approximative or heuristic arguments used. Furthermore it is statistically correct in the sense that no artificial systematics are involved in the reconstructed series. Therefore, the reconstructed signals are perfectly randomized and are hardly distinguishable by appearance from original test drive data.

The fact that a method with the described benefits is available since 1985 (see [12]) seems to be not well known in the international community. The algorithm here presented can be seen as both a refinement and a conceptional and didactical reorganization of the classical KSBP–algorithm. By this we hope to make this valuable tool more transparent.

2 Rate–independence and turning points

As mentioned in the introduction a well established assumption to the process of fatigue is that it is rate–independent.

Mathematically spoken, the damage is a functional on the set of loading functions. We assume that a loading function

$$s : [0, T] \longrightarrow \mathbb{R}$$

is piecewise monotone. This means there exists a subdivision $0 = t_0 < t_1 < \cdots < t_n = T$ such that the restriction of s on any interval $[t_i, t_{i+1}]$

is monotone. In this case we call $(t_0, t_1, \ldots, t_N)$ a monotonicity division of $[0, T]$.

Definition 2.1 *(i) Let us denote by $\mathcal{M}_{pm}([0,T])$ the class of all piecewise monotone functions on $[0,T]$*

(ii) Let $s \in \mathcal{M}_{pm}([0,T])$ and $(t_0, t_1, \ldots, t_N)$ be a minimal monotonicity division of $[0,T]$. Then we call $(s(t_0), s(t_1), \ldots, s(t_N))$ the string of turning points of s

(iii) A transformation $\phi : [0,T] \longrightarrow [0,T]$ is called monotonicity preserving $:=$ ϕ is monotone increasing, $\phi(0) = 0$ and $\phi(T) = T$

(iv) A functional $\mathcal{D}$ on $\mathcal{M}_{pm}([0,T])$ is called rate–independent $:=$
For any $s \in \mathcal{M}_{pm}([0,T])$ and any monotonicity preserving ϕ, it holds

$$\mathcal{D}(s) = \mathcal{D}(s \circ \phi)$$

It is an immediate consequence, that:

Lemma 2.1 *Let $s \in \mathcal{M}_{pm}([0,T])$ and $(t_0, t_1, \ldots, t_N)$ be a matching monotonicity division of $[0,T]$. Let D be rate–independent, then*

$$\mathcal{D}(s) = \mathcal{D}(\tilde{s})$$

for any $\tilde{s} \in \mathcal{M}_{pm}([0,T])$, with $\tilde{s}(\tilde{t_i}) = s(t_i)$, for all $i = 0, \ldots, N$ and $(\tilde{t_0}, \tilde{t_1}, \ldots, \tilde{t_N})$ a monotonicity division of $\tilde{s}$

This implies we can define any rate–independent functional as a functional on the set of turning points.

This concept gives us the first part of data reduction. We just need the string of turning points.

3 Rainflow counting – a brief survey

The first definition by Endo 1968 [10] used heuristic rain–flow models and thus gave the name to the method. Independently, in 1969 de Jonge [5] developed the equivalent range–pair–range method. Since then a lot of further variants and points of view on Rainflow have been presented.

Clormann and Seeger [4] suggest an on–line counting, which carefully includes the Masing plus memory hysteresis material behavior. Rychlik's off–line definition [15] is motivated by the stochastic process aspect, and is well suited to handle transition probabilities on the basis of rainflow. The hysteresis loop counting [1],[7], [2], based on the two–parameter distribution of the rainflow density, is especially useful for multi–axial generalizations.

From an algorithmic point of view, a very effective on–line counting is the so–called "4–point"–counting , as presented in [12] and briefly sketched below:

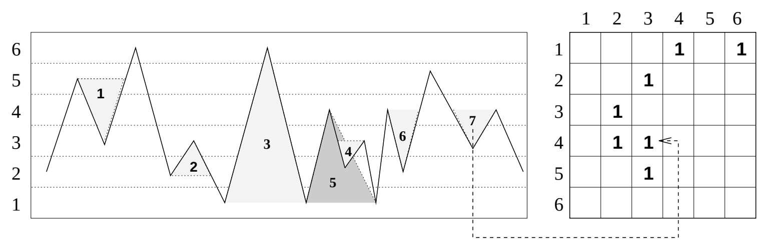


Figure 1: *The 4–point counting algorithm: The loops are found in the ordering of the numbers. In the case of loops 4 and 5 the loops are nested. First the small loop 4 is found, then the large one 5. The loops are classified and counted in the rainflow matrix* RFM *(e.g. loop 7 is counted in (4,3)).*

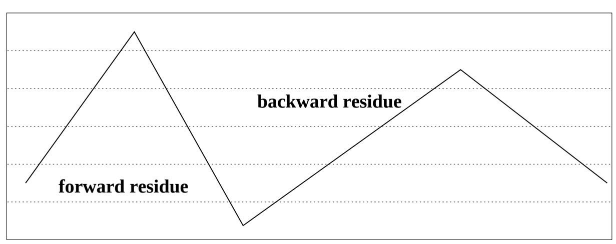


Figure 2: *The resulting residue: The first part represents the forward residue (future determining). The remainder is the backward residue (representing the history).*

Let $(s_0, \ldots, s_M)$ be a string of turning points of a loading signal. M must be larger than 3, such that hysteresis loops can occur. The values s_i are usually classified such that $s_i \in \{1, \ldots, N\}$

The algorithm works with the last four points of an actual residue $(res(\ell), res(\ell-1), res(\ell-2), res(\ell-3))$. We know that the inner points $res(\ell-2), res(\ell-1)$ build up a hysteresis loop if they are contained in the interval spanned by the left and right value $res(\ell-3)$ and $res(\ell)$.

We give the algorithm in pseudo code, the notation $[a, b]$ means the convex hull of a and b, i.e the interval $[a, b]$ if $a \leq b$ and the interval $[b, a]$ if $b < a$.

Initialization
repeat *ongoing loop*
  repeat *backtracking loop*
    if $[res(\ell-2), res(\ell-1)] \subset [res(\ell-3), res(\ell)]$ **then**
      Add loop to rainflow matrix
      Delete loop from actual residue
  until *no loop found* **or** *less than four points in the actual residue*
  if *no loop found* **then**
    take next turning point if there is any and add it to the actual residue
  if *less than four points in the actual residue* **then**
    fill actual residue with turning points if possible
until *there are no more turning points*

At the end of the process we have got both, the rainflow matrix and the residue. This residue consists of a strictly increasing part – the forward residue – and of a decreasing part – the backward residue (see Fig. 2). This splitting of the residue will be of importance for the on–line reconstruction algorithm.

The last approach to be mentioned here is the so–called memory definition [2], which is a dimension–free description from a mathematical/mechanical point of view.

We find the hysteresis loops if we follow the trajectory in the plane of our measured loading and a depending value. For example if we measured the local strain, the depending value may be the stress. From the mechanical point of view we consider the dependency to follow the Masing plus memory laws.

This process can also be considered as an special operator on the set $\mathcal{M}_{pm}([0,T])$ called Preisach hysteresis operator (see [3] for a rigorous definition). Here we have a clear concept of memory:

Let $\mathcal{W} : \mathcal{M}_{pm}([0,T]) \longrightarrow \mathcal{M}_{pm}([0,T])$ be a hysteresis operator of Preisach type and let $\boldsymbol{\sigma} = \mathcal{W}(\boldsymbol{\varepsilon})$. Then usually the value of $\boldsymbol{\sigma}$ at any time $t \in (0,T)$ does not only depend on $\boldsymbol{\varepsilon}(t)$ but also on the history $\boldsymbol{\varepsilon}|_{[0,t)}$. Memory at the time t is all one has to know about $\boldsymbol{\varepsilon}|_{[0,t]}$ to uniquely determine $\boldsymbol{\sigma}|_{[t,T]}$ from $e|_{[t,T]}$.

In other words: The memory is all one has to know about the past to uniquely determine the future output by the future input. It turns out that the backward rainflow residue of $\boldsymbol{\varepsilon}|_{[0,t]}$ is a coding for the memory. We see from the algorithm that whenever a hysteresis loop occurs, it is deleted from the actual residue. In that sense we can say, that rainflow counting is equivalent to the rule: *"Look at the memory (i.e. the backward residue) of $\boldsymbol{\varepsilon}|_{[0,t]}$. As t increases, count whatever the memory forgets."*

All the counting methods mentioned here are basically equivalent up to some slight variations regarding the treatment of the residual part. All methods allow both symmetric and asymmetric versions, i.e. the choice of whether to distinguish between hanging and standing hysteresis loops of the same range and amplitude.

4 Rainflow reconstruction – general scope and results

The basic problem is the following: Given a rainflow matrix and its residue, find a time signal, whose rainflow counting gives exactly the prescribed rainflow data. We saw in the section on rate–independence, that it is equivalent to find a string of turning points leading to this rainflow data. Finding any such time signal is what we call a mathematically exact reconstruction.

The importance of a mathematically sound treatment of this question is emphasized by the examples displayed in Fig. 3. If one develops a reconstruction algorithm without understanding of the combinatorial distribution of the possible strings of turning points (compare the following section) one is very likely to produce signals with undesired systematics like the ones shown here.

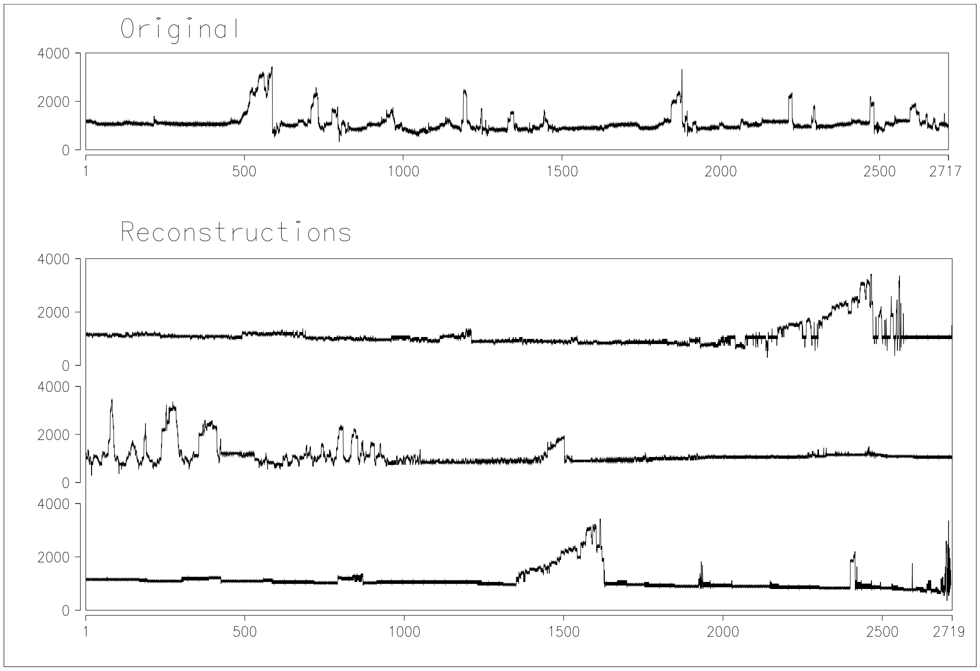


Figure 3: *Examples of possible rainflow reconstructions with undesired systematics*

The reconstruction algorithm presented in this paper involves a randomization procedure, which uses a complete mathematical analysis of the combinatorial distribution of all possible reconstructions. Among many more useful things, this analysis allows to calculate the number of all possible correct reconstructions out of a given rainflow matrix a priori by an explicit factorial formula. Even for matrices with very few (i.e. short signals) this number becomes astronomically large.

The practical consequence of the combinatorial analysis and of the resulting randomization procedure is, that the reconstruction produces perfectly non–systematic sequences as shown in Fig. 4:

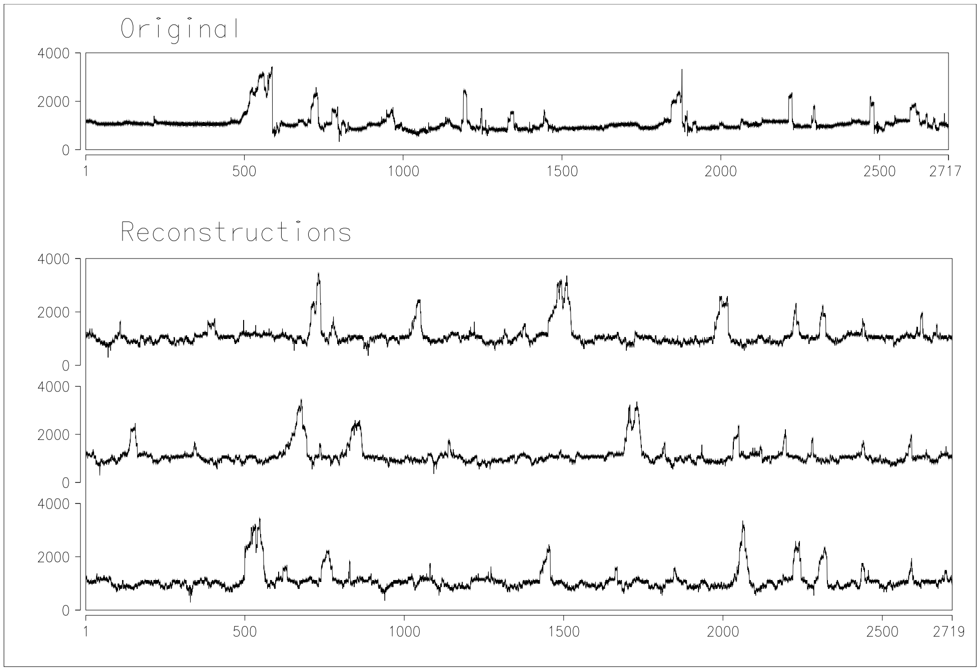


Figure 4: *Randomized rainflow reconstructions*

5 Mathematical description of the reconstruction algorithm

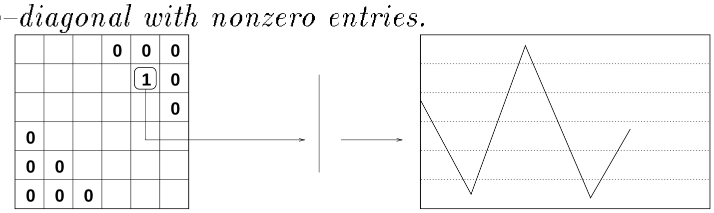
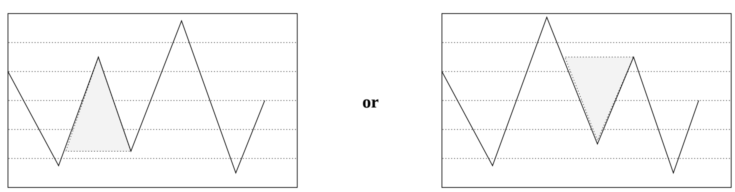
For didactic reasons, we split up our description of the general reconstruction algorithm in three steps: First we explain the principles of the rainflow reconstruction in the form of an off–line algorithm. Then we show how the resulting non–systematic combinatorial cycle distribution can be generated in an on–line algorithm for symmetric rainflow. Some specific problems regarding the asymmetric version are discussed in the final section.

5.1 Rainflow reconstruction – off–line

Suppose an N–class rainflow matrix RFM (symmetric or asymmetric) with a consistent residue RES are given. We want to construct a string of turning points s.

In the discussion of the rain flow method we have seen, that in the case of nested loops first the small loops are found then the larger ones. In the case of reconstruction we have to proceed the other way round. It is clear that large loops can not be nested in small ones but small loops can be nested in appropriate large ones.

A closer look on the structure of the rainflow matrix shows that we can distinguish between larger and smaller loops by their place in the RFM. The modulus of the difference between column and row index gives the amplitude of the loop. Hence we find loops of the same amplitude at the same sub–diagonal of RFM and small loops are found near to the main diagonal while the large one are far away from it.

> *We set s initially to* RES *and start with the original* RFM
> **repeat**
>
> *Insert all loops in* RFM *with maximal amplitude into s. Finding the loops with maximal amplitude means look for the first sub–diagonal with nonzero entries.*
>
> 
>
> ***Figure 5:*** *The loop belongs to the loops with the largest amplitude in this example* RFM*.*
>
> *In general there are various possible resulting strings, such as:*
>
> 
>
>
> ***Figure 6:*** *Here there are two possibilities to insert the loop.*
>
> *Select one of them at random and set s to this one*
> *Delete the loops from* RFM
>
> **until** RFM *is empty*

Why does it work at all? This simple algorithm produces a correct rainflow reconstruction, i.e. a turning point sequence, whose rainflow counting is exactly the initial (RFM, RES). This result is independent of the probability distribution underlying the random choice.

We have seen in the previous section that the rainflow algorithm counts out closed hysteresis loops. These loops may develop directly from the residue–curve, but may also be nested in other loops. Since we reinsert full loops in our algorithm their rainflow count is the same set of loops. The problem that could occur is that there are loops which can not be inserted anymore. This is the reason for first inserting the loops with the larger amplitudes before the loops of smaller amplitude.

Observation: Inserting a rainflow loop into a turning point sequence does not affect the possibility to insert any loop of equal or smaller amplitude afterwards. Hence if there was a possibility to insert it (and there must have been one, since the loop was counted from the original signal) then there is still this possibility after the insertion of the large loops. In contrast, a too early insert of a "small" loop, like

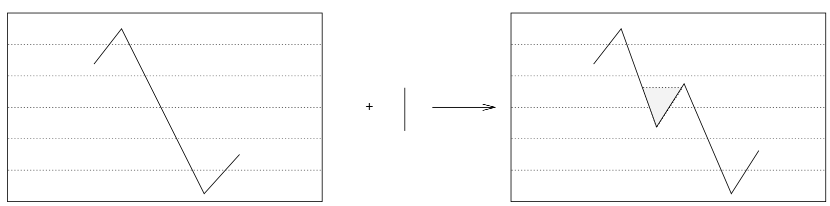


Figure 7: *A small loop inserted*

destroys possibilities to build in larger ones later on:

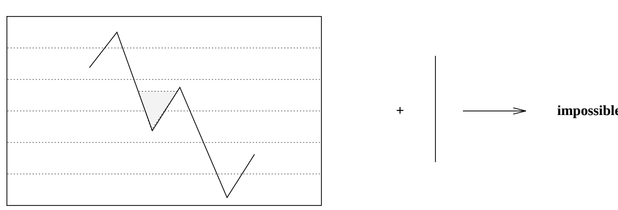


Figure 8: *Now we can't insert this larger loop which could have been inserted before.*

We summarize the result of this observation in

Lemma 5.1 *Strictly proceeding from large to smaller amplitudes in the above off–line algorithm guarantees a correct reconstruction.*

How to choose the sequences? If one is content with the result so far, one has a reconstruction, but one can not avoid to have undesired systematics in the resulting sequences. With just heuristic randomization, one is very likely to produce series like those in Fig. 3. The following lemmata show how the random choice has to be done in order to produce an uniform distribution in the set of possible reconstructions. This guarantees perfectly randomized results as shown in Fig. 4.

Lemma 5.2 *Let* RFM,RES *be a symmetric rainflow count with classes* $1, \ldots, N$ *and* $i < j$ *two classes. and let* s *be any string of turning points that represents the residue enlarged by all loops with amplitude larger than* $|j - i|$

(i) The number of possibilities $\mathbf{P}(i,j)$ *to insert an loop* (i,j) *into the string* s*, is a function of the original* RFM *and* RES *and thus a priori known:*

$$\begin{aligned}\mathbf{P}(i,j) &= 2\sum_{k=1}^{i-1}\sum_{\ell=j+1}^{N}\mathsf{RFM}(k,\ell) \\ &+ \sum_{\ell=j+1}^{N}\mathsf{RFM}(i,\ell) + \sum_{\ell=1}^{i-1}\mathsf{RFM}(j,\ell) \\ &+ \#\{k \,|\, [i,j] \subset [\mathsf{RES}(k), \mathsf{RES}(k+1)]\}\end{aligned}$$

where $[a,b]$ *denotes the interval* $[a,b]$ *if* $a < b$ *resp. the interval* $[b,a]$ *if* $b < a$ *and* $\#M$ *the cardinality of the set* M*.*

(ii) The number $\mathbf{K}(i,j)$ *of different resulting sequences after inserting all oscillations* (i,j) *is a combinatorial function of* $\mathbf{P}(i,j)$ *and* $\mathsf{RFM}(i,j)$ *and thus also known a priori:*

$$\mathbf{K}(i,j) = \binom{\mathbf{P}(i,j) + \mathsf{RFM}(i,j) - 1}{\mathsf{RFM}(i,j)}$$

Proof:

Consider a loop (i, j) with $i < j$. In the symmetric case inserting a loop (k, ℓ) with $k < \ell$ from RFM increases the possible positions for (i, j) by two, if $k < i < j < l$ and by one, if $k = i < j < \ell$ or $k < i < j = \ell$. If we add the possible positions in the residual we achieve the formula for $\mathbf{P}(i, j)$. The formula for $\mathbf{K}(i, j)$ is achieved by standard combinatorics. □

Remark: In the asymmetric case the same type of argument holds. However, the consideration of the orientation of the loops leads to slight modifications of the formula for $\mathbf{P}(i, j)$. We have to distinguish between hanging loops (these are loops that start on an increasing branch) and standing loops. Since hanging loops are decreasing in the beginning and we classify using the opening part of the loop here i is larger than j.

We get for $i > j$:

$$\begin{aligned}\mathbf{P}(i,j) &= \sum_{k=1}^{i-1}\sum_{\ell=j}^{N}\mathsf{RFM}(k,\ell) + \sum_{k=i+1}^{N}\sum_{\ell=1}^{j}\mathsf{RFM}(k,\ell) \\ &+ \#\{\nu \in \{1,\ldots,M-1\} | \mathsf{RES}_\nu \le j, \mathsf{RES}_{\nu+1} \ge i\}\end{aligned}$$

and for $i < j$:

$$\begin{aligned}\mathbf{P}(i,j) &= \sum_{k=1}^{i-1}\sum_{\ell=j}^{N}\mathsf{RFM}(k,\ell) + \sum_{k=j+1}^{N}\sum_{\ell=1}^{i}\mathsf{RFM}(k,\ell) \\ &+ \#\{\nu \in \{1,\ldots,M-1\} | \mathsf{RES}_\nu \ge j, \mathsf{RES}_{\nu+1} \le i\}\end{aligned}$$

The other formulae hold in the asymmetric as in the symmetric case.

Now we know how many possible positions for one loop (i, j) we have. We have to fix one of these positions for each of the $\mathsf{RFM}(i, j)$ loops by our randomization process. To choose it we need the probability distribution for the process.

Lemma 5.3 *Let us consider the same situation as in Lemma (5.2) and let* $k \in \{0, 1, \ldots, \mathsf{RFM}(i,j)\}$. *Let the probability to insert* k *loops* (i,j) *into one of the* $\mathbf{P}$*(i,j) positions be*

$$\mathbf{W}_{ij}(k) = \frac{\binom{\mathbf{P}(i,j) + \mathsf{RFM}(i,j) - k - 2}{\mathsf{RFM}(i,j) - k}}{\binom{\mathbf{P}(i,j) + \mathsf{RFM}(i,j) - 1}{\mathsf{RFM}(i,j)}}$$

Then each possible constellation resulting after inserting all $\mathsf{RFM}(i,j)$ *elements with size* $|j - i|$ *will be created with equal probability.*

Proof:

Let $S > 2$ loops of type (i,j) be given. Furthermore let us consider any turning point sequence, whose rainflow counting contains no elements with amplitude smaller than $|j - i|$. Let $P > 2$ be the number of possible positions for inserting a loop (i,j) into that sequence.

Now the decision to put $k \in \{0, 1, \ldots, S-1\}$ loops into one particular position leaves

$$\binom{(P-1) + (S-k) - 1}{S-k}$$

possibilities to distribute the other loops among the other positions. Thus the distribution

$$\mathbf{W}_{ij}(k) = \frac{\binom{P + S - k - 2}{S-k}}{\binom{P + S - 1}{S}}$$

yields each possibly resulting sequence with equal probability. □

Combining the results so far, we have :

Theorem 5.4 *The number Z of all possible reconstructions factorizes into*

$$Z = \prod_{i,j} \mathbf{K}(i,j)$$

The reconstruction algorithm with random choice according to Lemma (5.3) yields each of them with equal probability.

5.2 Rainflow reconstruction – on–line symmetric

The basic idea of the on–line reconstruction is to invert the rainflow counting according to the memory definition described at the end of section 3. We start with the given forward residue.

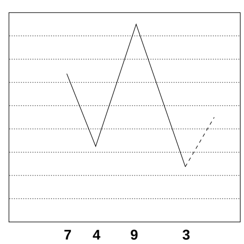


Figure 9: *Our example forward residue contains the classes 7 4 9 and 3.*

This forward residue tells where to go, i.e. which levels have to be reached and in which order, i.e. whether the first (next) visit of level i has to occur before or after that of level j.

In particular the first grid step (in our example: $7 \mapsto 6$) is uniquely determined. So we can save the actual first value (here: 7) as the first (next) value of the signal in construction. Our new actual forward residue now starts with the new grid value (here: 6)

After the first step (here $7 \mapsto 6$), there are various possible new forward residues, which are consistent with the available information, here:

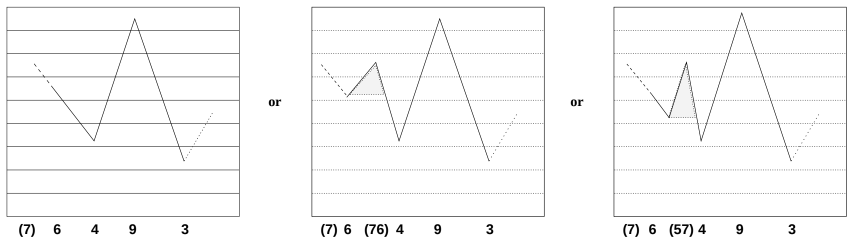


Figure 10: *In this example are sketched some of the possible new forward residues. Of course one possibility is that nothing happens (leftmost picture), In the middle and right picture we inserted the loops (6,7) and (5,7). There are other possibilities with loops going to level 8.*

The second and third version in the example are only possible, if RFM(6,7) and RFM(5,7) are not zero.

How to select a new forward residue? We have to remember the probabilities from Lemma (5.3). If we are in the situation that the first step in our forward residue is from level i to level j.

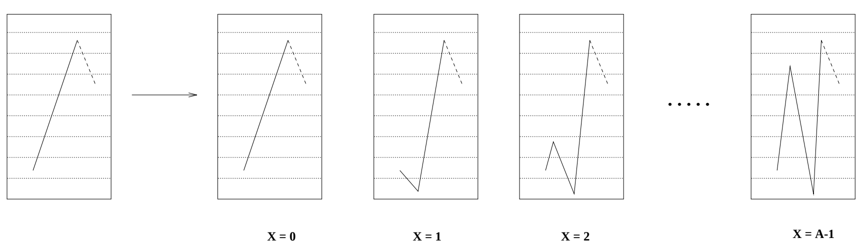


Figure 11: *We sketched all possible insertions. The variable X gives the amplitude of the next inserted loop and $A = |j - i|$ the maximal possible amplitude to be inserted.*

We can calculate the possibility that there is a loop with amplitude k as:

$$\mathbf{w}_k = \sum_{\ell=1}^{RFM(i,i+k)} \mathbf{W}_{i,i+k}(\ell) = 1 - \mathbf{W}_{i,i+k}(0)$$

If we want to calculate the probability that the next step has amplitude k, we have to exclude that there are smaller loops before. Hence the probability the the next step is given by a loop with amplitude k must be calculated as

$$\begin{aligned}
\mathcal{P}(X = k) &= w_k(1 - w_{k-1}) \cdots (1 - w_1) \quad \text{for all } k \in \{2, \ldots, A-1\} \\
\mathcal{P}(X = 1) &= w_1 \qquad \text{and} \\
\mathcal{P}(X = 0) &= (1 - w_{A-1}) \cdots (1 - w_1)
\end{aligned}$$

Here A denote the maximal possible amplitude, i.e. $|j - i|$. The value for $X = 0$ gives the possibility that no loop is inserted in between.

We can summarize in the following theorem

Theorem 5.5 *Let* $\mathsf{RES}(1) = i$, $\mathsf{RES}(2) = j$. *If we choose the "new future" after the step* $i \mapsto i \pm 1$ *among the* $A = |j - i|$ *possibilities with the probabilities P(X), then all possible reconstructions are produced with equal probability.*

This leads to the algorithm:

> *Start with the given forward residue* RES
> **repeat**
> > *do the first step which is induced by the residue:* $\mathsf{RES}(1) + sgn(\mathsf{RES}(2) - \mathsf{RES}(1))$
> > *Choose one of the new possible forward residues using the probability distribution P(X).*
> > *Delete the inserted loop from the* RFM
>
> **until** RFM *empty*

5.3 Rainflow reconstruction – on–line asymmetric

In the off–line situation there is no principle difference between symmetric and asymmetric rainflow reconstruction. In fact, the only technical difference are the formulae in lemma (5.2) and the remark that follows this lemma.

In the on–line situation, however, one has to deal with the following fundamental problem: In contrast to symmetric rainflow, the asymmetric counting depends on the orientation of time. Since the reconstruction procedure described in section 5.2 involves the orientation of time in form of the local forward residues, one runs into the following problem.

The orientation of a loop as it appears in the "forward residue inserting mechanism" of Theorem (5.5)

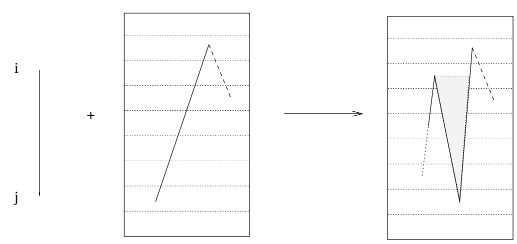


Figure 12: *The loop is inserted as a hanging loop.*

may differ from the resulting (counting) orientation,

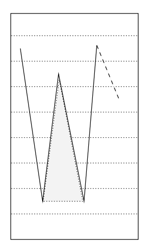


Figure 13: *but is counted as a standing loop.*

depending on the previous time history. Hence in the asymmetric situation we have to consider part of the actual backward residue, too. The situation depends also on the orientation of the inserted loop. In the example in the figures above, we insert a hanging loop (i, j), which is counted with $i > j$. In the example we visited a level larger than i after the last visit of a level smaller than j. If we denote by $S := sgn(i - j)$ the orientation of the loop and by first/last(k) the the time of the first/last visit of the level k, we can characterize this situation (also in the situation that $i < j$) by

$$\text{last}(i + S) > \text{last}(j - S)$$

In the case that a level smaller than j was visited after the last visit of a level larger than j this situation is impossible. The only case we have still to take into account is if none of the enclosing levels has been visited in the actual past. We can characterize this situation by

$$\mathrm{last}(i+S) = \mathrm{last}(j-S) = 0$$

Here we have to consider the levels i and j themselves. In fact the first situation describes the nesting of hysteresis loops now the loop is inserted directly into the forward residue and hence we have to consider the first visits of the levels:

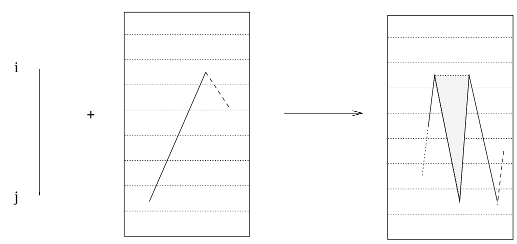


Figure 14: *Here the loop is inserted correctly as a hanging loop.*

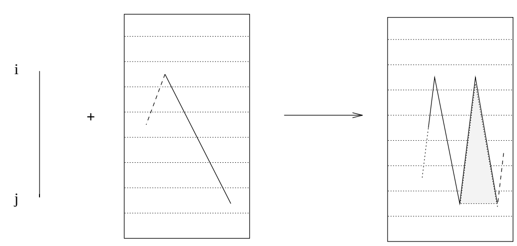


Figure 15: *But here it must be counted as a standing loop.*

We can characterize the latter (wrong) situation in terms of first:

$$0 < \mathrm{first}(i) < \mathrm{first}(j)$$

This leads to the result

Theorem 5.6 *If we omit the situations*

$$\begin{aligned} last(i+S) &> last(j-S) \qquad and \\ last(i+S) &= last(j-S) = 0 \ \ and\ 0 < first(i) < first(j), \end{aligned}$$

then our asymmetric reconstruction using the formulae of the remark to lemma (5.2) has the correct asymmetric rainflow count and yields any possible reconstruction with the same probability

This description allows to take care of the very special needs of the asymmetric situation by a straightforward technical modification of the symmetric on–line reconstruction algorithm.

References

[1] **Beste, A.** et al.: *Multiaxial Rainflow: A Consequent Continuation of Professor Tatsuo Endo's Work.* In: **Y. Murakami** (ed.) *The rainflow method in fatigue*, Oxford 1992.

[2] **Brokate, M., Dreßler, K., Krejčí, P.**: *Rainflow counting and energy dissipation for hysteresis models in elastoplasticity*, submitted.

[3] **Brokate, M., Visintin, A.**: *Properties of the Preisach model for hysteresis*, J. reine angew. Math. **402**, 1–40 (1989).

[4] **Clormann, U. H., Seeger, T.**: *Rainflow–HCM: Ein Zählverf. für Betriebsfestigkeitsnachweise auf werkstoffmech. Grundlage*, Stahlbau **55**, 65–71 (1986).

[5] **de Jonge, J.B.**: *The Analysis of Load–Time–Histories by Means of Counting Methods*, National Aerospace Laboratory NLR. MP 82039U, ICAF Document. **1309**, (1982).

[6] **Dowling, N. E.**: *Fatigue failure predictions for complicated stress–strain histories*, J. Materials, **7**, 71–87 (1972).

[7] **Dreßler, K., Carmine, R., Krüger, W.**: *The multiaxial rainflow method.* In: **K.T.Rie** (ed.) *Low cycle fatigue and elasto–plastic behaviour of materials*, (Elsevier Science Publ., London 1992), pp.325–330.

[8] **Dreßler, K; Köttgen, V.B.; Beste, A.; Kötzle, H.** : *Feasibilities of numerical procedures in fatigue analysis*, VDI Berichte Nr. **1153**, 43–59 (1994).

[9] **Dreßler, K., Krüger, W., Beste, A.**: *Rainflow – das Werkzeug für den Lebensdauernachweis von Fahrzeugen*, 19.Vortagsveranstaltung des DVM–Arbeitskreises "Betriebsfestigkeit", München (1993).

[10] **Endo, T.**: *Papers presented at the Kyushu District Meeting of the Japanese Soc. of Mech. Engineers* (1967) (reprinted in [Mu])

[11] **Haibach, E.**: *Betriebsfestigkeit*, (VDI–Verlag, Düsseldorf, 1989)

[12] **Krüger, W., Scheutzow, M., Beste, A., Petersen, J.**: *Markov– und Rainflowrekonstuktionen stochastischer Beanspruchungszeitfunktionen*, VDI–report, series 18, No **22** (1985).

[13] **Miner, M.A.** : *Cumulative damage in fatigue*, J.Appl.Mech. **12** A159–A164 (1945).

[14] **Murakami, Y.**, (ed.), *The Rainflow Method in Fatigue*, (Butterworth & Heinemann, Oxford, 1992).

[15] **Rychlik, I.**: *A new definition of the rainflow cycle counting method*, Internat. J. Fatigue **9**, 119–121 (1987).