European Consortium for Mathematics and Industry

Lifetime estimation in the car industry

Sixth ECMI Modelling Week
September 4-13, 1993
Grenoble (France)

Supervisor :
Michael HACK (University of Kaiserslautern, Germany)

Students :
Krister BOMAN (Chalmers University of Technology, Göteborg, Sweden)
Pim MEIJ (Eindhoven University of Technology, Netherlands)
Philippe MONTARNAL (University Joseph Fourier, Grenoble, France)
Kerstin SCHUSTER (University of Dresden, Germany)
Darren WALL (University of Strathclyde, Glasgow, Scotland)

Contents

1 Introduction

1.1 Problem description

Whenever new parts of a car have been developed, the manufacturer needs an estimation of the lifetime of this new part. On one hand the construction must not be too weak, so that the part holds long enough to satisfy the customer, but on the other hand, if the construction is too excessive, the part gets too heavy.

One is interested in methods that only need few measured data from the specimen itself, but use data about the material, because constructing and testing of specimen is expensive.

1.2 Data available

A specimen of the developed part is built in a car in its specific position and the car is driven along a test track. In the critical points of the specimen (e.g. notches, edges, etc.) one places strain-measure-stripes (SMS) which are able to measure local strains and record them.

A change in the local strain causes an elastic deformation of the specimen. If the change gets to large then a plastic deformation takes place. After the direction of the change has reversed, then again we have the same process: first elastic then plastic deformation. Hence, if one follows the trajectory in the stress-strain-plane, the unloading does not follow the loading path, but a hysteresis loop will be built up.

One knows, that for the fatigue damage of metals, these closed hysteresis loops in the stress-strain-plane play a major rule. One has performed strain-controlled tests : How many closed hysteresis loops of a given strain-amplitude a given material can sustain. The influence of loops with a different amplitude is accumulated.

1.3 Masing and Memory

We have the following information about the characteristics of the trajectory in the stress-strain-plane ($\sigma - \epsilon -$plane).

There are two type of curves : the cyclic curve and the doubled curve.

Metals and alloys obey the following Masing and Memory laws :

Masing :

- Initially — during the first loading step — the $\sigma - \epsilon-$path follows the so called cyclic $\sigma - \epsilon-$curve : $\epsilon = g(\sigma)$. This means, if we change σ in the time interval $[0, t_1]$ monotonically from $\sigma(0) = 0$ to $\sigma(t_1)$, the strain at an intermediate time $t \in (0, t_1)$ is given as $\epsilon(t) = g(\sigma(t))$
- After a turning-point, i.e. the derivative of σ with respect to t changes the $\sigma - \epsilon-$path follows the doubled $\sigma - \epsilon-$curve : $\Delta\epsilon/2 = g(\Delta\sigma/2)$:

  We get the value of $\epsilon(t)$ as follows: Get the time of the last turning-point t_n. We know that $\sigma(t)$ is monotonic on (t_n, t). Now following the Masing law, we get

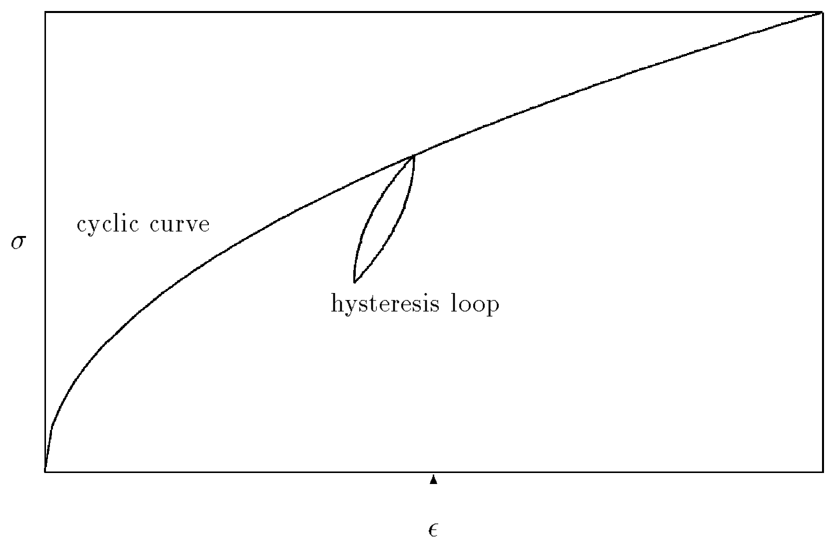
$$\epsilon(t) = \epsilon(t_n) + 2\ \mathrm{sign}(\sigma(t) - \sigma(t_n))\ g\left(\frac{\sigma(t) - \sigma(t_n)}{2}\right)$$

But these rules do not completely describe the behaviour. Not only the turning–points are important, but the closure of loops in the stress–strain–plane and a certain symmetry–condition have influence, too:

Memory :

- **M1 :** After closing a hysteresis loop, which started on the cyclic $\sigma - \epsilon-$curve, the $\sigma - \epsilon-$path again follows the cyclic $\sigma - \epsilon-$curve. See figure 1.

Figure 1: M1



- **M2 :** After closing a hysteresis loop, which started on a hysteresis branch, the $\sigma - \epsilon -$path again follows this hysteresis branch. See figure 2.

- **M3 :** A hysteresis branch, that started on the cyclic $\sigma - \epsilon -$curve ends, if the modulus of strain of its starting point is reached in the opposite quadrand. After that the $\sigma - \epsilon -$path again follows the cyclic $\sigma - \epsilon -$curve. See figure 3.

Figure 2: M2

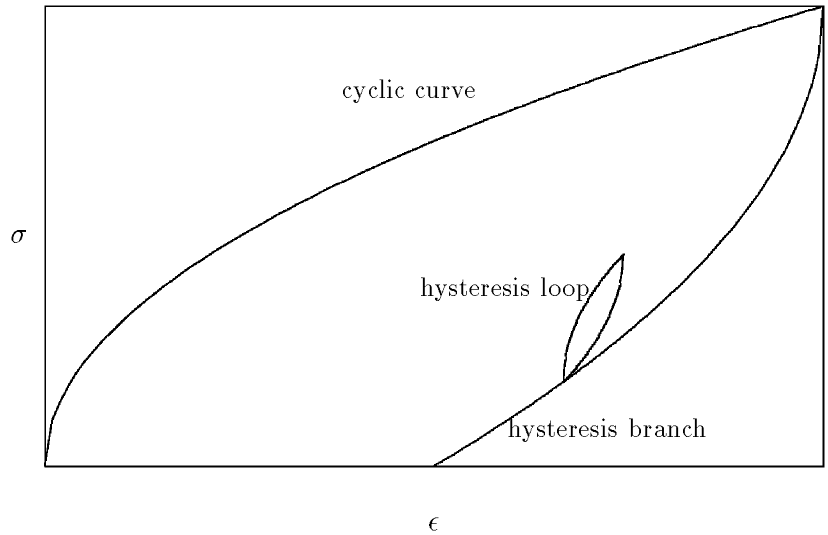
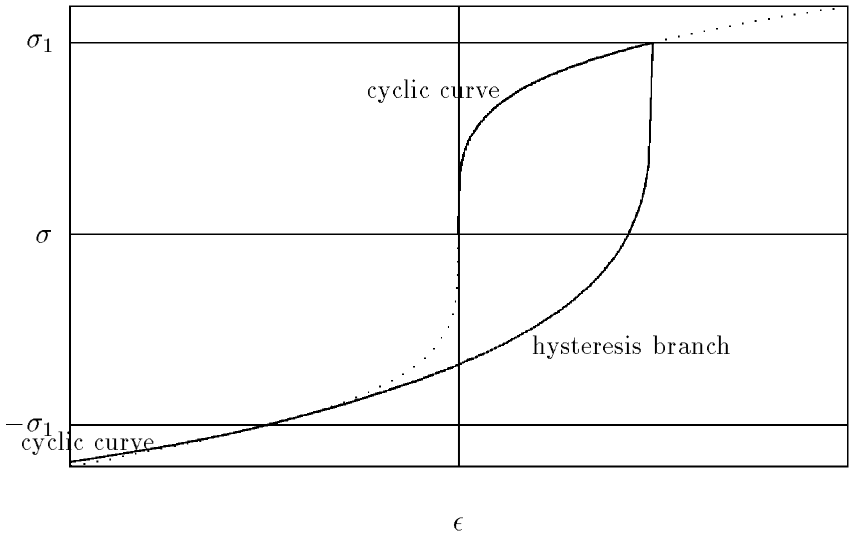


Figure 3: M3



1.4 Goal

We want to transform the stream of measured strain values into the corresponding set of hysteresis loops in the stress-strain-plane. In particular we must determine the turning points corresponding to the starting point and end point of each hysteresis loop.

We must consider all the masing and memory laws to minimize the data.

1.5 Plan of presentation

At first we will study a simple model which will help us for the comprehension of the global model (memory laws, turning points). Then we will start with the real problem which can be considered as a generalization of the first one. Finally we will present a computer program (which is written in MATLAB) and some test results.

2 Simple model - Mechanical play

2.1 Description of the mechanical play

First we examine the mechanical play, a simple model were we can find all the rules and a similar behavior to the original problem.

The mechanical play consists of a stick and a slot with an inside radius r. The stick will be moved to the left or to the right and if the movement of the stick is large enough the slot will move. To describe the position of the slot we use the midpoint. See figure 4 for a sketch.

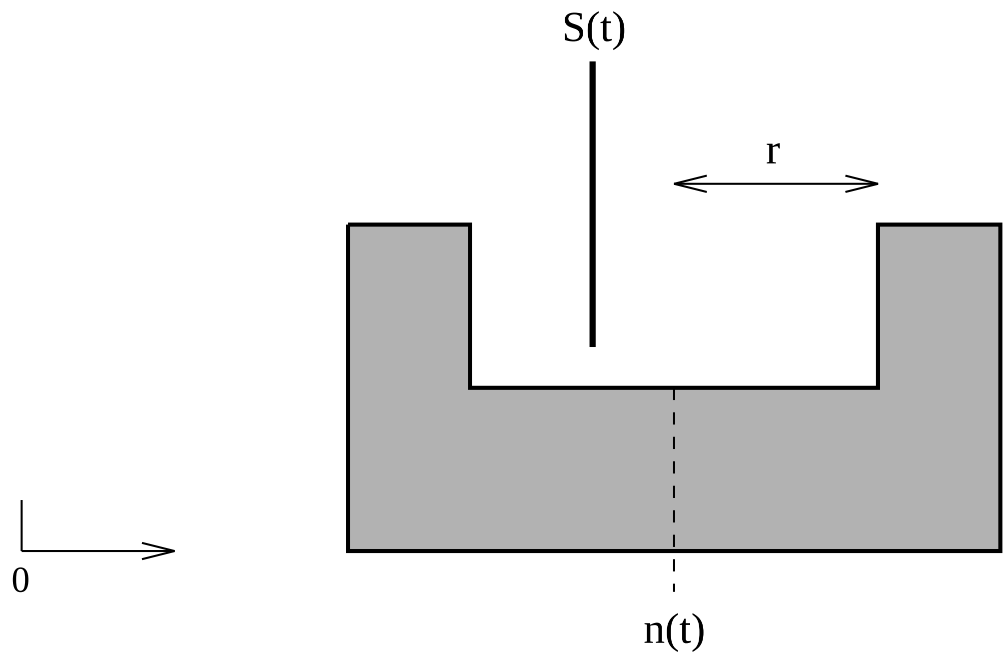


Figure 4:

Now we ask for the new position of the slot after a change in stick position. There are 3 cases possible : the slot will move to the left or to the right or it will stay in the old position. So we get :

$$n(t_{k+1}) = \begin{cases} s(t_{k+1}) - r & \text{if } s(t_{k+1}) - n(t_k) \geq r, \\ s(t_{k+1}) + r & \text{if } n(t_k) - s(t_{k+1}) \geq r, \\ n(t_k) & \text{otherwise.} \end{cases}$$

This is equivalent to :

$$n(t_{k+1}) = \min[\max[s(t_{k+1}) - r, n(t_k)], s(t_{k+1}) + r]$$

If we only have monotone input, we find a simpler formula :

- increasing : $n(t_{k+1}) = \max[s(t_{k+1}) - r, n(t_k)]$
- decreasing : $n(t_{k+1}) = \min[s(t_{k+1}) + r, n(t_k)]$

In the original problem we can find elastical and plastical behavior. How is it in the mechanical play? If we consider the case when the slot stays in the old position after a change in the stick position, we can say, this is an elastical behavior. In the case where the slot has moved, we can speak of plastical behavior.

2.2 Connection between memory, memory laws and deletion rules

The memory of a material for a time t_0 describes the condition of this material in terms of the history of the strain ($t < t_0$), such that we are able to calculate the condition at the time t_1 with the help of the future strain ($t_0 < t < t_1$).

In our mechanical play, the input is the position of the stick $s(t)$ and the output is the position of the slot $n(t)$. In chapter 2.1 we found that we can calculate the new slot position $n(t_1)$ using $n(t_0)$ and $s(t_1)$, i.e. the memory at the time t_0 can be described by $n(t_0)$.

This also means, that for the time $t > t_0$ one can not distinguish between input functions which caused the same value $n(t_0)$. They yield the same memory and one can use the simplest input function. It is the task of the deletion rules to find these functions (i.e. a finite number of values).

In general, the deletion rules transform the given input data to a minimal string only containing the relevant input data for the current memory at t_0.

The deleted data can however be important for the damage calculation. During the run through the hysteresis loops energy will be dissipated so we are interested in the number and size of the hysteresis loops.

The memory laws M1 - M3 describe the behavior of the material phenomenologically (What happens, if ...). Using the deletion rules it is possible to show, that the material has the same phenomenological properties as described by M1 - M3.

2.3 Deletion rules

We can find the following three deletion rules.

Lemma

If we set $n_{-1} = 0$, i.e. initially not disturbed system, then

- D1 : delete t_k if $s(t_k) \in [s(t_{k-1}), s(t_{k+1})]$ (monotone deletion, see figure 5.)
- D2 : delete t_k if $s(t_k) \in [s(t_{k+1}), s(t_{k+2})]$ (Preisach deletion, see figure 6.)
- D3 : delete t_0 if $|s(t_0)| < |s(t_1)|$ (see figure 7.)

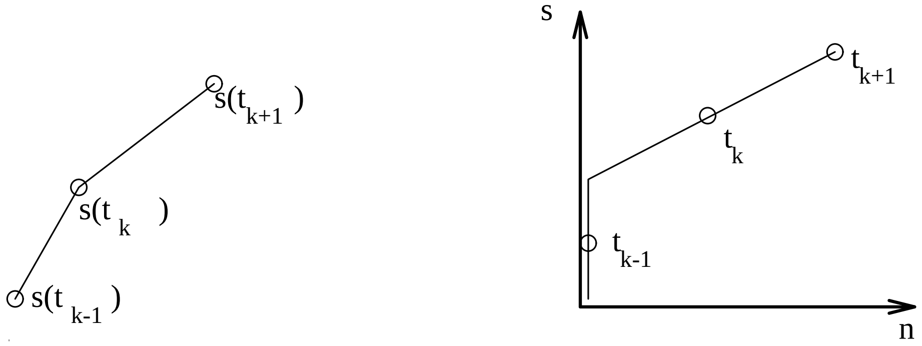


Figure 5:

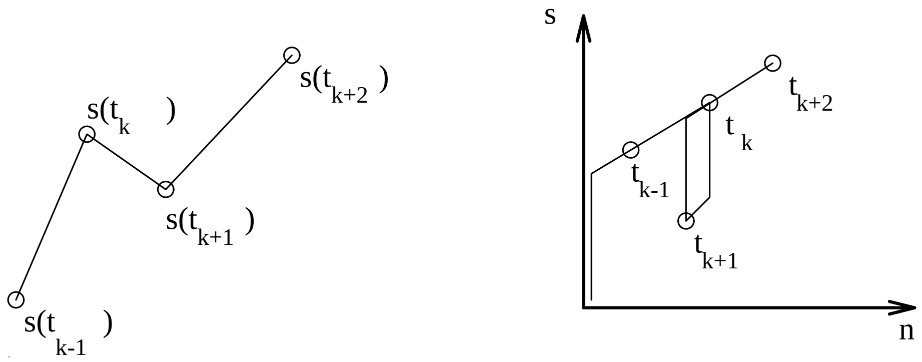


Figure 6:

Explanation

- D1 :

  If the stick will be moved to the right/left until t_k and after that until t_{k+1} still to the right/left, for the new position of the slot only $s(t_{k+1})$ and $s(t_{k-1})$ are important.

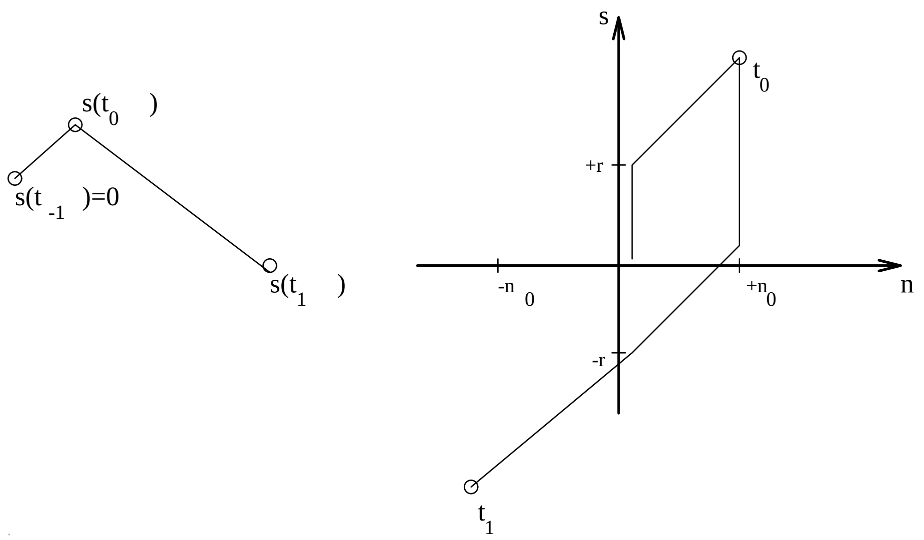


Figure 7:

The memory of $s(t_{k+1})$ does not depend on t_k and we can delete t_k.

(This means, that only the local minima and maxima of the input function $s(t)$ are important for the description of the behavior of the slot.)

- D2 :

  The memory of $s(t_{k+2})$ does not depend on t_k and we can delete t_k.

  This means, that the opening point of a loop can be deleted.

- D3 :

  This rule is equivalent to memory 3. The proof will explain this rule.

Proofs

- D1 :
  The proofs are given in figure 8 and figure 9, for the increasing case and the decreasing case respectively.

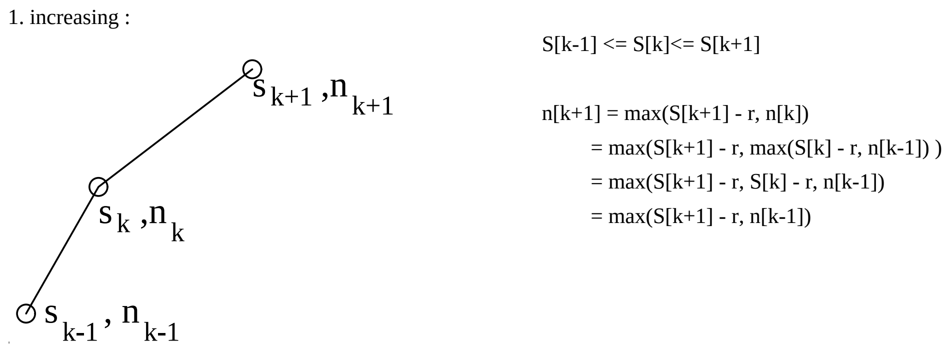


Figure 8:

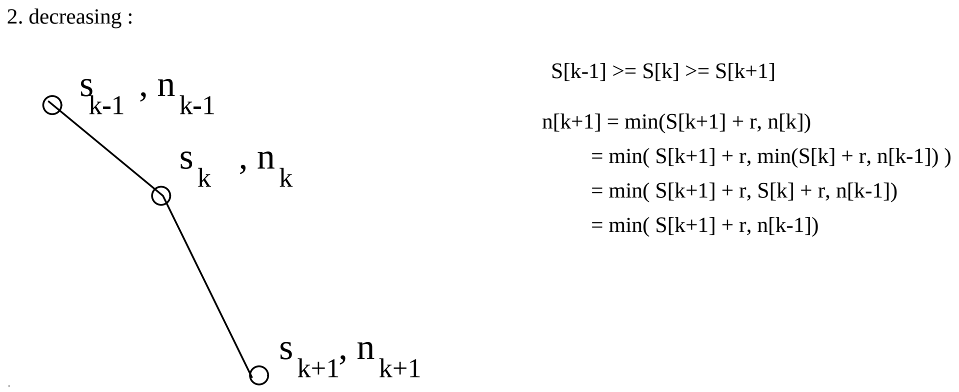


Figure 9:

- D2 : Without loss of generality assume that $k = 1$, we have to show that n_3 does not depend on S_1, see also figure 10. We consider the case $S_2 \le S_1 \le S_3$.

  $\Rightarrow n_3 = \max(S_3 - r, \min(S_2 + r, \max(S_1 - r, n_0)))$

  - if $S_1 - r \ge n_0$ then $n_3 = \max(S_3 - r, \min(S_2 + r, S_1 - r))$
    - if $S_2 + r < S_1 - r$ then $n_3 = \max(S_3 - r, S_2 + r)$
    - if $S_2 + r \ge S_1 - r$ then $n_3 = \max(S_3 - r, S_1 - r) = S_3 - r$
  - if $S_1 - r < n_0$ then $n_3 = \max(S_3 - r, \min(S_2 + r, n_0))$. So for this case we have found that the position n_3 is independent of S_1. The proof is analogously for the case $S_3 \le S_1 \le S_2$.

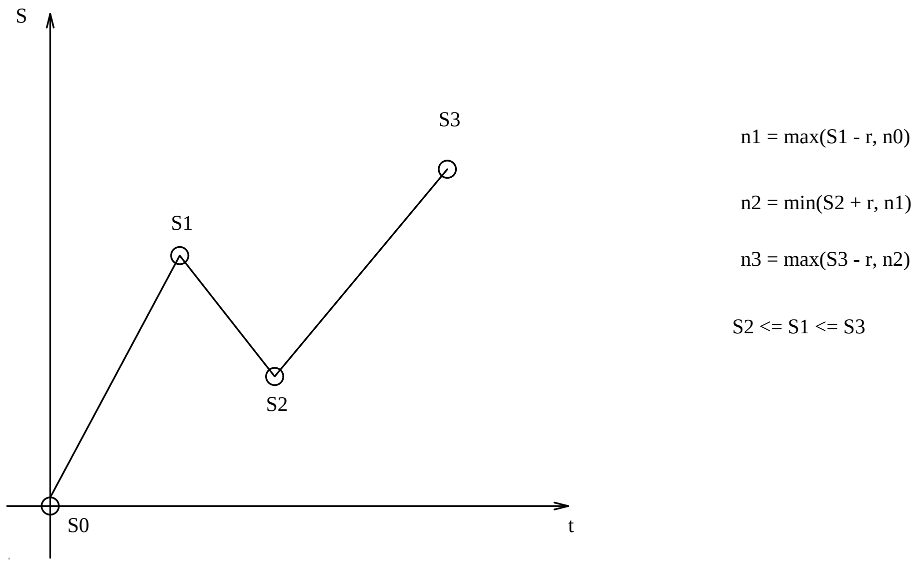


Figure 10:

- D3 :

  1. $|S_1| \leq r \Rightarrow |S_0| \leq r \Rightarrow$ nothing happens
  2. $S_1 < -r \Rightarrow n_1 = \min(S_1 + r, \max(S_0 - r, 0)) = S_1 + r$
  and $n_1 = S_1 - r$ if $S_1 > r$ and since we can use D1 when S_0 and S_1 both have the same sign we are done.

2.4 Memory laws

Using the three deletion rules, it's possible to show that all three kinds of memory laws can be found in the mechanical play.

- M1 : After closing a hysteresis loop, which started on the cyclic s-n-curve the s-n-path again follows the cyclic s-n-curve.

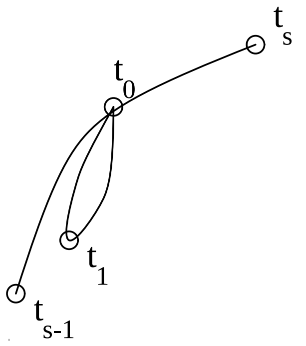
  Proof :

  See figure 11 for an explanation of the symbols. In t_0 the path followed the cyclic curve and therefore the memory is $(s(t_0))$.

  In t_s the memory is $(s(t_0), s(t_1), s(t_s))$.

  With rule D2, we delete t_0 and with D3 the point t_1 and the memory in t_s is $(s(t_s))$, which is what M1 says.

- M2 : After closing a hysteresis loop, which started on a hysteresis branch, the s-n-path again follows this hysteresis branch.



t_0 : begin of the loop
t_1 : turning point of the loop
in $[t_{s-1}, t_s]$ the loop will be closed

Figure 11:

Proof : We consider a hysteresis loop inside another hysteresis loop. We only have to consider two loops. The other cases can be found using recursion.

Now two cases can occur: The inside loop can begin on the opening curve or in the closing cur ve of the outside hysteresis branch.

t_0^o, t_1^o, t_2^o are the opening, turning and closing point of the outside loop. t_0^i, t_1^i, t_2^i are the opening, turning and closing point of the inside loop.

- First case :
  We have to show that the memory in t_1^o is $(..., s(t_0^o), s(t_1^o))$.
  In this case the memory in t_1^o is $(..., s(t_0^o), s(t_0^i), s(t_1^i), s(t_2^i), s(t_1^o))$.
  With rule D1 we delete t_2^i, with rule D2 t_0^i, and with rule D1 t_1^i, and we get the correct result.
- Second case :
  In the same way as in the fist case, we show that the memory in t_2^o is $(..., s(t_0^o), s(t_1^o), s(t_0^i), s(t_1^i), s(t_2^i), s(t_2^o)) = (..., s(t_0^o), s(t_1^o), s(t_2^o))$

- M3 : A hysteresis branch, that started on the cyclic curve ends if the modulus n of i ts starting points is reached in the opposite quadrand. after that the s-n-path again follows the cyclic curve.

  This is equal to deletion rule D3.

3 Multi-slot model - Mechanical play

Now we seek to model the stress-strain Masing behaviour of car parts by using several mechanical play operators. Let us first consider again the example of a single stick operating in a single slot.

3.1 One-Slot Model

We consider the movement of the midpoint, $n(t)$, of a block with a single slot with time, t. We will assume the movement of the stick is piecewise linear between known positions at $t = t_0, t_1, t_2, \ldots$. We wish to find an operator which yields the position of the midpoint of the slot given a change in stick position, $S(t_i)$, and the former position of the midpoint of the slot $n(t_{i-1})$. We have seen that the former stick position $S(t_{i-1})$is not required in this calculation. For example, for a given radius $r > 0$ of the slot, movement of the block to the right (increasing n) will occur if and only if the change in stick position is greater than the amount of 'play' in the slot, that is if and only if

$$S(t_i) - S(t_{i-1}) > n(t_{i-1}) + r - S(t_{i-1}). \tag{1}$$

Given that condition (1) holds, then the change in position of the midpoint of the block, Δn, is given by the change in stick position less the amount of 'play' in the block, that is

$$\Delta n = S(t_i) - S(t_{i-1}) - [n(t_{i-1}) + r - S(t_{i-1})] = S(t_i) - n(t_{i-1}) - r,$$

thus the new position of the block is given by

$$n(t_i) = n(t_{i-1}) + \Delta n = S(t_i) - r.$$

The other cases of movement of the block to the left and no movement may be similarly calculated to yield

$$n(t_i) = \begin{cases} S(t_i) - r & \text{if } S(t_i) - n(t_{i-1}) \geq r \\ S(t_i) + r & \text{if } n(t_{i-1}) - S(t_i) \geq r \\ n(t_{i-1}) & \text{otherwise.} \end{cases} \tag{2}$$

The expression (2) is equivalent to

$$n(t_i) = \min[\max(S(t_i) - r, n(t_{i-1})), S(t_i) + r] \tag{3}$$

A natural progression from this model is to consider the case of two slots, three slots and so on until the case of a distribution of infinitely many slots may be considered.

3.2 Infinitely Many Slots

Considering $W(t)$ as a mean position of all the slots, and if we split up this position into a plastical part, $h(t)$, and an elastical part $f(t)$, then

$$W(t) = h(t) + f(t).$$

The elastical part corresponds to slots of radius zero, and these slots obviously follow the movement of the stick precisely, summing over all such slots we have $f(t) = \gamma S(t)$, where γ is a constant. For the plastical part of the movement we sum over all operators $n(t,r)$ where r denotes the radius of slot that the operator $n(t,r)$ relates to. In the Riemann limit, this sum becomes

$$h(t) = \int_{r>0} \alpha(r)\, n(t,r)\mathrm{d}r,$$

where $\alpha(r)$ denotes the continuous spectrum of radius sizes, which must be nonnegative to make physical sense. Thus we have

$$W(t) = \int_{r>0} \alpha(r)\, n(t,r)\mathrm{d}r + \gamma S(t). \tag{4}$$

In order to obtain any conditions which must be satisfied on the cyclic curve we consider a simple loading with $S(0) = W(0) = 0$ and assume that the stick moves to a position $S(t_1) = a$ where $a > r$ for movement. For ease we will consider W here as a function of stick position rather than time (of course the two are proportional in this simple loading). Now

$$n(t_1, r) = \min\{\max(a-r, 0), a+r\} = a - r,$$

so

$$W(a) = g(a) = \int_{0+}^{a} \alpha(r)\,(a-r)\mathrm{d}r + \gamma a, \tag{5}$$

thus we have

$$g'(a) = \int_{0+}^{a} \alpha(r)\mathrm{d}r + \gamma$$

so

$$g''(a) = \alpha(a). \tag{6}$$

Substituting equation (6) into (5) we have

$$g(a) = \int_{0+}^{a} g''(r)\,(a-r)\mathrm{d}r + \gamma a, \tag{7}$$

using integration by parts twice we have

$$g(a) = g(a) - g(0^+) - a g'(0^+) + \gamma a, \tag{8}$$

Given that $g(0^+) = 0$, then we have $g'(0^+) = \gamma \geq 0$. So in summary, we have the following conditions on a cyclic curve

$$\begin{array}{ll} (1) & \alpha(r) = g''(r) \\ (2) & g(0^+) = 0,\ g'(0^+) = \gamma \geq 0 \\ (3) & g''(r),\ g'(r) \geq 0. \end{array}$$

4 Relating the multi-slot model to the original model

Let us assume the following: $S(t_0) = c < S(t_2) = b < S(t_1) = a$ and $a - b > 2r$ (for movement), see Figure 12.

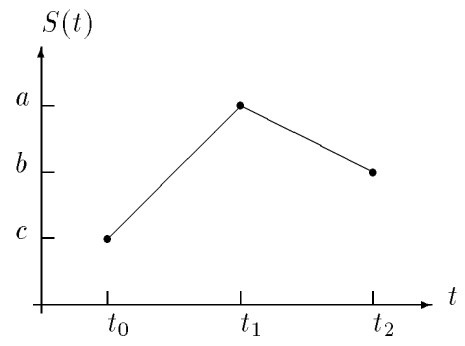


Figure 12: The graph of the load when we have a turning-point.

We get

$$\Delta W = \int_0^\infty \alpha(r)(n(b,r) - n(a,r))dr. \tag{9}$$

So using (3) we have

$$n(b,r) - n(a,r) = 2r + b - a. \tag{10}$$

Hence, using the condition $a - b > 2r$,

$$\Delta W = -2\int_0^{(a-b)/2} \alpha(r)((a-b)/2 - r)dr. \tag{11}$$

Thus, using (5)

$$\Delta W/2 = -g((a-b)/2), \tag{12}$$

which is precisely Masing's rule for a turning point.

So we see that the multi-slot model shows the same kind of Masing behavior as the original function in the $\sigma - \epsilon -$plane. Of course, also all deleting and therefore memory rules will hold for the multi-slot model, because they were already proven for the model with one slot, and the multi-slot model is just an infinite summation of one slot models. Thus we can conclude that the Masing and Memory behavior in the $\sigma - \epsilon -$plane can be completely described and proved using the one slot model.

5 Algorithm and results

In the previous sections we have described the memory laws and deletion rules and given some theoretical prove. Now, with these memory laws M1 - M3 we saw that hysteresis loops, once finished, do not contribute anymore to the memory. Hence, they can be deleted from the original stream of strain data and written to a seperate file containing all these loops. This is exactly what the computer program, presented in Appendix A, does. The program is written in MATLAB and consists of a main program and three subroutines. For details, check the computer code. A comprehensive description of the algorithm will now be given. The flowchart and file description in figure 13 are used as a guideline.

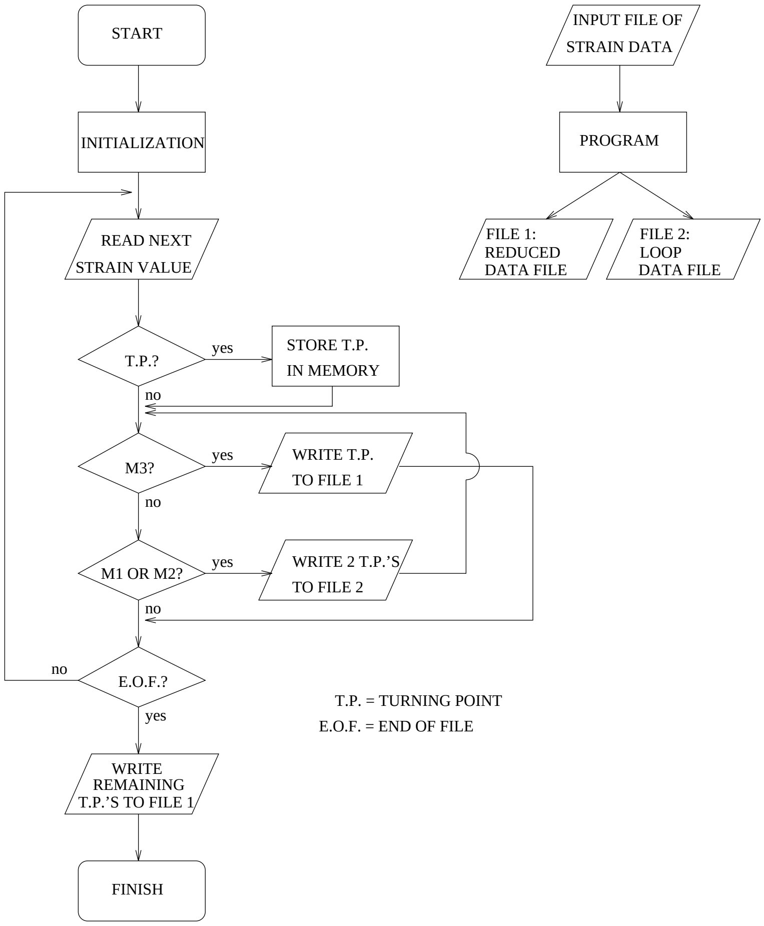


Figure 13: The flowchart and file description of the algorithm

The input file contains the measured stream of strain data or ϵ-values. The program converts this file into two files:

1. File 1 contains the data stream without all the loop data. Also, all intermediate

points which can be deleted with deletion rule D1 are left out. So the points we are left with mark the turning points at which an M3 condition occured. Looking at this condition, we see immediately that this sequence of M3 points must be increasing. The tail of file 1 also contains the turning points of the loops which were not finished at the end of the data stream. This sequence has to be decreasing because we get a loop in a loop in a loop, etc.

2. File 2 contains all the loop data. Again, all intermediate points are left out. This leaves us with two data points for each loop. These are the opening and the turning-point of the loop. This file 2 can afterwards be used for some post-processing. E.g.: how many loops occur, how big are they in size? All such information can be of interest to calculate the energy released by the car part and so estimate the damage.

The algorithm is as follows:

1. Initialization: The file with strain values is read and some program variables are set.
2. Loop: Read the next strain value, say, ϵ_i.
3. We can easily check if a turning-point has occured. This is the case if $\text{sign}(\epsilon_i - \epsilon_{i-1}) \neq \text{sign}(\epsilon_{i-1} - \epsilon_{i-2})$. Then ϵ_{i-1} is a turning-point and is stored in internal memory. Otherwise, we can forget ϵ_{i-1}, using deletion rule D1.
4. We check if the M3 condition is satisfied. We only need to check for this condition if there is only one turning-point left (otherwise we have not yet encountered a turning-point or there are still some unfinished loops). If M3 is satisfied, the turning-point can be deleted from internal memory and written to file 1.
5. If M3 is not satisfied, we can check for the M1 or M2 condition. One of these conditions is satisfied if $\epsilon_i > tp_1 > tp_2$ or $\epsilon_i < tp_1 < tp_2$ where tp_1 and tp_2 are the opening point and turning-point of the last encountered loop. They are stored in internal memory. If either M1 or M2 is satisfied, then tp_1 and tp_2 are deleted from internal memory and written to file 2. We should now again check if ϵ_i satisfies the M3 condition or perhaps is the closure of other loops.
6. When the end of the input file with strain values is reached, we write the remaining turning-points in internal memory to file 1 and finish the program.

Appendix B contains some test data.
On the first page we see the input file containing a stream of strain data $\epsilon_i, 1 \leq i \leq 250$.
The second page shows the points in file 1 after running the program. We see that

the data with the loops "stripped off" consists of an increasing sequence of M3 points. At the end of the file some remaining turning-points of unfinished loops were left in memory. This decreasing sequence is also shown.

The third page contains the loop data in file 2. We see pairs of crosses and circles connected with eachother. Every cross marks the opening (and closing) point of the loop while every circle marks the turning-point of the loop. There are 39 loops in total.

For the loop data we can make a so-called *rainflow matrix*. For this end, the interval $(-1.0, 1.0)$ is divided in 20 parts: $1 : (-1.0, -0.9), 2 : (-0.9, -0.8), \ldots, 18 : (0.7, 0.8), 19 : (0.8, 0.9), 20 : (0.9, 1.0)$. Let A be a 20 by 20 matrix. For every loop the opening point falls in one of these 20 intervals, say, in interval i. Also the turning-point of the loop falls in one of them, say, in interval j. Now let the elements a_{ij} of A be the number of such loops with opening point in interval i and turning-point in interval j. The result for the test-data is shown graphically on the fourth page. The height of each peak on position (i, j) denotes the value of a_{ij}.

We see that most loops are along the main diagonal $\{a_{ij}; i = j\}$ of A. These are the loops for which the opening point and turning-point are fairly close, so they are small loops i.e. not inflicting too much damage.

A Appendix: Computer program

Main program:

```
clear
load testp;
epsold=0;
delold=0;
lenarr=0;
loopnr=0;
output1=0;
output2=0;
flag2=1;
epsnew=testp(2);
delnew=spsign(epsnew-epsold,delold);

for i=3:length(testp)
  epsold=epsnew;
  delold=delnew;
  epsnew=testp(i);
  delnew=spsign(epsnew-epsold,delold);
  if delnew~=delold
    lenarr=lenarr+1;
    array(lenarr)=epsold;
  end

  if lenarr==1
    if abs(epsnew)>abs(array(1))
      output1(length(output1)+1)=array(1);
      epsold=0;
      delold=0;
      lenarr=0;
      delnew=spsign(epsnew-epsold,delold);
    end
  end

  if delnew==1
    while control1(lenarr,epsnew,array)==1
      output2(length(output2)+1-flag2)=array(lenarr-1);
      output2(length(output2)+1)=array(lenarr);
```

```
      lenarr=lenarr-2;
      flag2=0;
    end
  else
    while control2(lenarr,epsnew,array)==1
      output2(length(output2)+1-flag2)=array(lenarr-1);
      output2(length(output2)+1)=array(lenarr);
      lenarr=lenarr-2;
      flag2=0;
    end
  end

 end




 if lenarr>0
   N=length(output1)+1;
   output1(N:N+lenarr-1)=array(1:lenarr);
 end}
```

Subroutines:

control1.m:

```
function val=control1(lenarr,epsnew,array)
val=0;
if lenarr>=2
   if epsnew>=array(lenarr-1)  val=1;
end
end
```

control2.m:

```
function val=control2(lenarr,epsnew,array)
val=0;
if lenarr>=2
   if epsnew<=array(lenarr-1)  val=1;
end
end
```

spsign.m:

```
function val=spsign(x,delold)
if x==0
  val=delold;
elseif x>0
  val=1;
else
  val=-1;
end
end
```

B Appendix: Test results