# Shortcuts in multiple dimensions: the multiaxial racetrack filter

M.A. Meggiolaro, J.T.P. Castro
*Pontifical Catholic University of Rio de Janeiro*
*meggi@puc-rio.br, jtcastro@puc-rio.br*

H. Wu
*Tongji University in Shanghai*
*wuhao@tongji.edu.cn*

**ABSTRACT.** Filtering techniques have been proposed for multiaxial load histories, usually aiming to filter out non-reversals, i.e. sampling points that do not constitute a reversal in any of its stress or strain components. However, the path between two reversals is needed to evaluate the equivalent stress or strain associated with each event. Filtering out too many points in such path would almost certainly result in lower equivalent stresses or strains than expected. To avoid such issues, it is important to consider how a measured multiaxial loading path deviates from its course using some metric, such as the von Mises stress or strain. In this work, a multiaxial version of the racetrack filter is proposed, which is able to perform efficient filtering even for 6D non-proportional histories. In the Multiaxial Racetrack algorithm, the stress or strain history is represented in a 6D space, only requiring from the user a desired scalar filtering amplitude r. For uniaxial histories, the proposed algorithm exactly reproduces the classic racetrack filter. The efficiency of the proposed Multiaxial Racetrack filter is qualitatively verified from a tension-torsion history example, showing the reduction in the number of data points for larger filter amplitudes r. The procedure can efficiently filter out non-damaging events but preserving the overall multiaxial path shape and multiaxial reversion points, which usually do not coincide with the reversion points of individual stress or strain components.

*KEYWORDS.* Multiaxial fatigue; Racetrack filter; Non-proportional loading; Variable amplitude loading.

## INTRODUCTION

The racetrack filter, originally proposed by Fuchs et al. in 1973 [1] for uniaxial histories, aims to eliminate small amplitude load events that, although usually plenty, do not induce fatigue damage from a variable amplitude loading history. So, the resulting condensed histories can much accelerate both experiments and computations, focusing only on the few events that cause most or all of the damage. Fig. 1 illustrates the uniaxial racetrack filter concept. The original history of Fig. 1(a) is condensed into the history in Fig. 1(d), eliminating amplitudes smaller than a user-specified value *r*. The idea of the filter is to draw a "racetrack" of width *2r*, bounded by upper and lower "fences" that have the same profile as the original history, see Fig. 1(b). If a "driver" racing in this racetrack needs to change its direction from upward to downward (or vice versa), then a reversal point is identified, as seen in Fig. 1(c), where the racer needed to change twice its direction, near points B and E. In this example, points C and D are filtered out, because the driver didn't have to change direction to avoid the fences associated with them. Clearly, the track width *2r* determines the number of counted reversals: wider tracks will filter out most of the original loading history, while narrow tracks will almost keep all the original reversals. In other words, the racetrack filter condenses the original variable amplitude history into a smoother history, discarding the small amplitude changes that cause negligible fatigue damage [2]. Moreover, the condensed history does not change the order of the significant load events, which is an essential feature to account for plasticity memory effects.

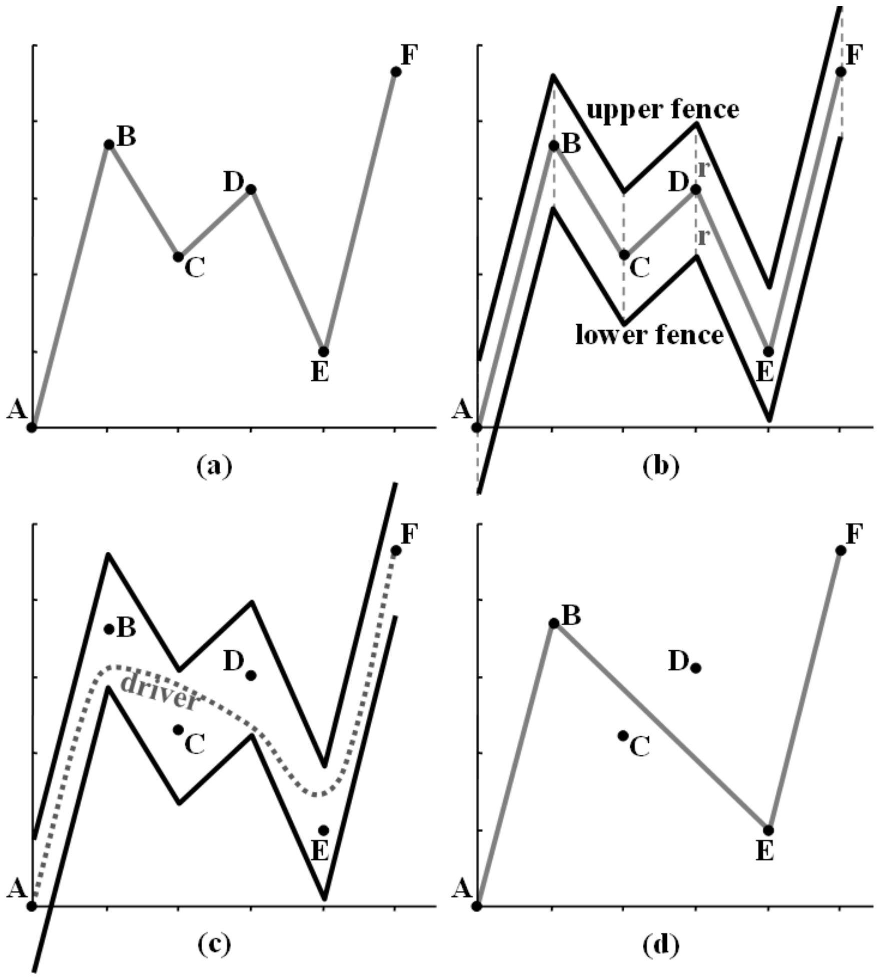

Figure 1: Uniaxial racetrack filter example, showing: (a) the original history; (b) the racetrack defined between the upper and lower fences; (c) the driver path from A to F; and (d) the filtered history.

The algorithm of the uniaxial racetrack filter is easier to implement through another physical analogy, a small round peg P that can oscillate inside a slotted plate whose center is the point O, see Fig. 2. The range that the peg can oscillate inside the slot is *2r*, the racetrack filter range. Both peg and slot center are initially aligned with point A, as seen in Fig. 2(a). During the path AB, the peg moves up until reaching the upper limit of the slotted plate, which then starts to move up, see Fig. 2(b), where the dark dashed line represents the path of point O. As shown in Fig. 2(c), the following path BCD moves the peg but does not involve any translation of the slotted plate, an indication that the points C and D can be filtered out: peg oscillations alone can be discarded. Then, both paths DE and EF involve slotted plate translations, see Fig. 2(d) and (e), so they must be maintained in the filtered history.

After the initial point A, only the peg locations P associated with the *end* of a translation of the slotted plate are stored in the filtered history. Point B is the peg location at the end of the first slotted plate upward translation, so it is stored. Similarly, points E and F are stored since they are the peg locations at the end of the next downward and upward slotted plate translations. The filtered history is then the path ABEF, as shown in Fig. 2(f), always identical to the path obtained from the original racetrack filter. This *amplitude filter* is very efficient in practice, since besides small damageless load events it can remove the unavoidable noise e.g. from strain measurements, which can induce much more small reversions than the load itself in the measured signal.

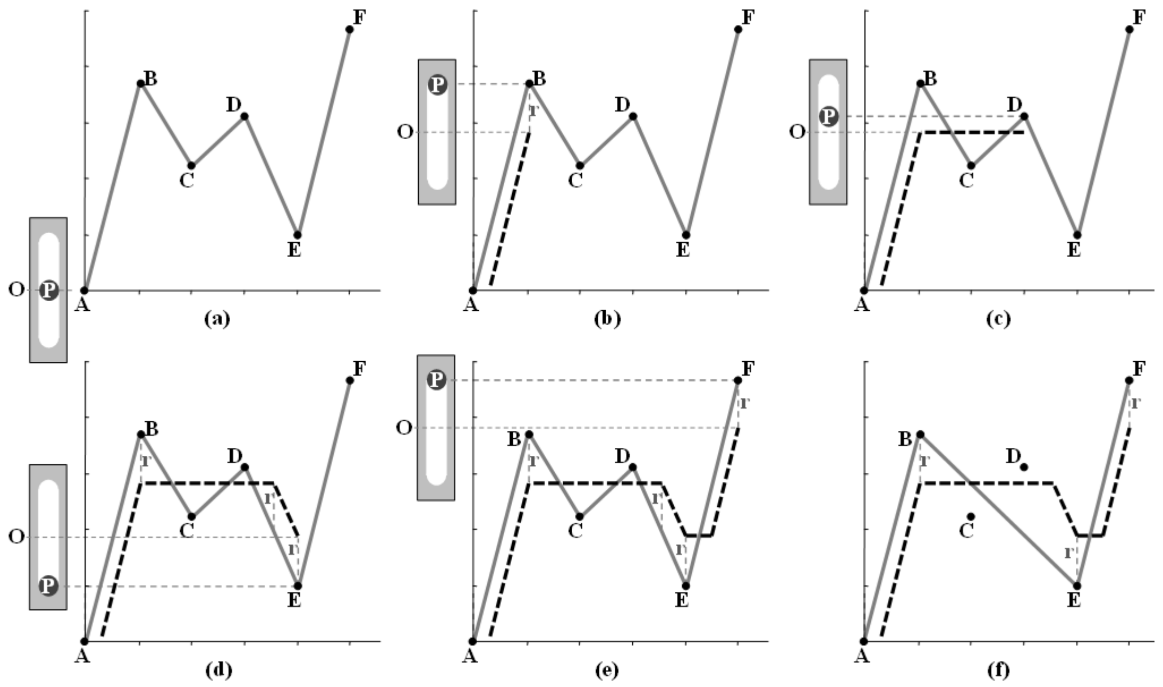

Figure 2: Physical analogy between the uniaxial racetrack filter and a peg oscillating inside a slotted plate with center O and slot range *2r*. The peg P oscillates according to the original history, resulting in translations of O represented by the dark dashed line.

## Issues with Existing Multiaxial Load Filters

A most important practical issue in multiaxial fatigue calculations, even more than in the uniaxial case, is how to filter out a load history to decrease their intrinsically high computational cost. So, like in the 1D case it may seem a good idea to first remove from the multiaxial loading all data points that are not peaks or valleys of any of their stress or strain components, since to properly measure a load signal it is necessary to oversample the digitalized data at a rate high enough to not distort the signal [3-4].

But this simple filtering practice is *not* recommended in NP multiaxial histories, for two reasons: first, the path between two load reversals is needed to evaluate the path-equivalent stress or strain associated with each rainflow count, e.g. using a convex enclosure method [5] or even better the Moment Of Inertia method [6]. Filtering out too many points in such path almost certainly results in lowering the equivalent stresses or strains, so some points along the multiaxial path should not be filtered out, even if they do not constitute a load component reversal. Moreover, reversal points obtained from a multiaxial rainflow algorithm may not occur at the reversal of one of the stress or strain components. For instance, the relative von Mises strain, used in the Wang-Brown (WB) rainflow method and its variations [7], may reach a peak value at a point that is neither a maximum nor a minimum of any strain component. But this important point would have been filtered out by any non-reversal filtering algorithm, resulting in inadmissible non-conservative predictions.
The second reason is because the reversal points obtained from a multiaxial rainflow algorithm might not occur at the reversal of one of the stress or strain components. For example, the relative Mises strain, used in the WB rainflow count, may reach a peak value at a point that is neither a maximum nor a minimum of any strain component. But such most important points would be filtered out by a non-reversal filtering algorithm, resulting in non-conservative fatigue damage and life and predictions.

The "Peaks Procedure" proposed in [8] e.g. filters out all events whose components are not peaks or valleys, potentially eliminating important load points that could have the highest Mises stresses or strains of the load history, even though each individual component was not maximized. Moreover, such procedure would store each and every event that constitutes a peak or valley from any single component, which for (unavoidably) noisy measurements could result in no events at all being filtered out, even if the noise had very low amplitudes.
To avoid such issues, how a measured multiaxial loading path deviates from its course must be evaluated by some metric, such as the Mises stress or strain. This is needed to avoid filtering out important counting points from multiaxial rainflow

algorithms or significant paths that can affect the calculation of an equivalent stress or strain, since all stress or strain components contribute altogether for the reversals that can be eliminated. Note that an *amplitude* filter is a most desirable feature in practical applications, to not only eliminate redundant measurement noise and oversampled data, but also small amplitudes that do not cause fatigue damage. In fact, it may be practically impossible to analyze unfiltered multiaxial fatigue data. A multiaxial version of the racetrack filter must perform such tasks even for NP histories.

## THE MULTIAXIAL RACETRACK FILTER

In the multiaxial racetrack algorithm proposed here, the stress or strain history is represented in a 6D space, and a suitable filtering amplitude *r* must be chosen like in the 1D case. A small peg $\vec{P}$ is then allowed to move in this 6D space, but instead of being restricted within a 1D slot, it is kept inside a 6D hyper-sphere of center $\vec{O}$ and radius *r*. When the peg reaches the hyper-sphere surface and tries to move out of it, both the peg and the hyper-sphere translate altogether, similarly to the 1D slotted plate example. Fig. 3 shows a 2D tension-torsion example of a hyper-sphere translation caused by the peg movement from its current position $\vec{P}_i$ to the next $\vec{P}_{i+1}$, where $\vec{n}_i$ is the current normal vector that defines the surface translation direction (still to be determined), and $b_i$, $a_i$, and $d_i$ are distances (measured in stress or strain units) used in the filtering algorithm.

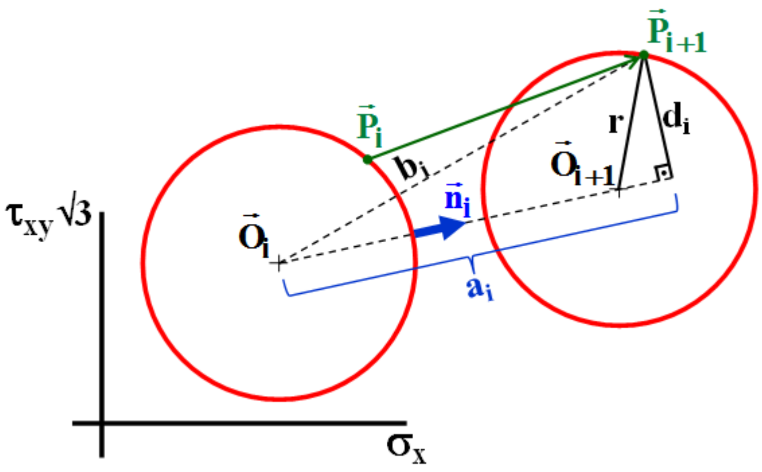

Figure 3: Hyper-sphere translation along $\vec{n}_i$, caused by a peg movement from $\vec{P}_i$ to $\vec{P}_{i+1}$. For this 2D tension-torsion example, the hyper-sphere simply becomes a circle.

The next peg location $\vec{P}_{i+1}$ is given by the input load history. When combined with the current location $\vec{O}_i$ of the hyper-sphere center, and a known translation direction $\vec{n}_i$, the values of $b_i$, $a_i$, and $d_i$ can be obtained from

$$b_i^2 = (\vec{P}_{i+1} - \vec{O}_i)^T \cdot (\vec{P}_{i+1} - \vec{O}_i), \quad a_i = (\vec{P}_{i+1} - \vec{O}_i)^T \cdot \vec{n}_i, \text{ and } d_i^2 = b_i^2 - a_i^2 \tag{1}$$

where $b_i$ must be greater than $r$ to guarantee that $\vec{P}_{i+1}$ is outside the current hyper-sphere, otherwise there is no translation. While $a_i \geq 0$ and $d_i \leq r$, the next peg location $\vec{P}_{i+1}$ can still be located on the border of the hyper-sphere translated in the $\vec{n}_i$ direction, where the center translation shown in Fig. 3 can be calculated by

$$\vec{O}_{i+1} = \vec{O}_i + (a_i - \sqrt{r^2 - d_i^2}) \cdot \vec{n}_i \tag{2}$$

The process is then repeated for the next peg location. Fig. 4 shows two consecutive translations where the conditions $a_i \geq 0$ and $d_i \leq r$ are satisfied, allowing the hyper-sphere translation direction to remain constant, i.e. $\vec{n}_i \equiv \vec{n}_{i-1}$. In this example, point $\vec{P}_i$ can be filtered out, since it does not alter the translation direction during the (multiaxial) load history path $\vec{P}_{i-1}\vec{P}_i\,\vec{P}_{i+1}$. This filtering process that happens while the hyper-sphere is translated is called here *dynamic filtering*.

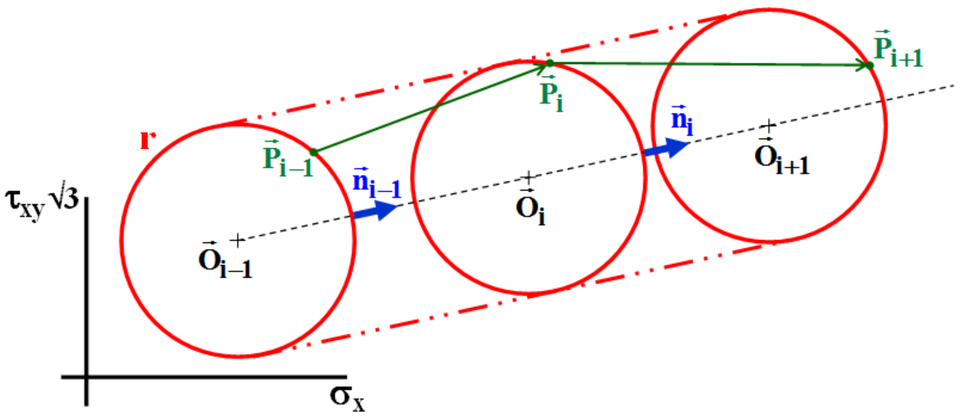

Figure 4: Hyper-sphere translations along $\vec{n}_i \equiv \vec{n}_{i-1}$, caused by the path $\vec{P}_{i-1}\vec{P}_i\,\vec{P}_{i+1}$. Dynamic filtering eliminates $\vec{P}_i$ because it does not alter the translation direction.

The initial locations $\vec{P}_1$ of the peg and $\vec{O}_1$ of the hyper-sphere center must coincide with the first loading point from the load history in the 6D stress or strain space. As seen in Fig. 5(a), no hyper-sphere translation happens while the peg moves inside it, therefore points $\vec{P}_2$ through $\vec{P}_{i-1}$ are filtered out, in a process called *static filtering* (because it does not involve hyper-sphere translations). When the peg reaches for the first time the hyper-sphere border and tries to move outside it, the translation direction $\vec{n}_i$ must be defined. A simple translation direction rule assumes here that $\vec{n}_i$ is determined from the segment that joins the current hyper-sphere center $\vec{O}_i$ and the next peg location $\vec{P}_{i+1}$, i.e. $\vec{n}_i = (\vec{P}_{i+1} - \vec{O}_i)/b_i$, see Fig. 5(a). Other improved rules for the translation direction of the hyper-sphere center could be defined.

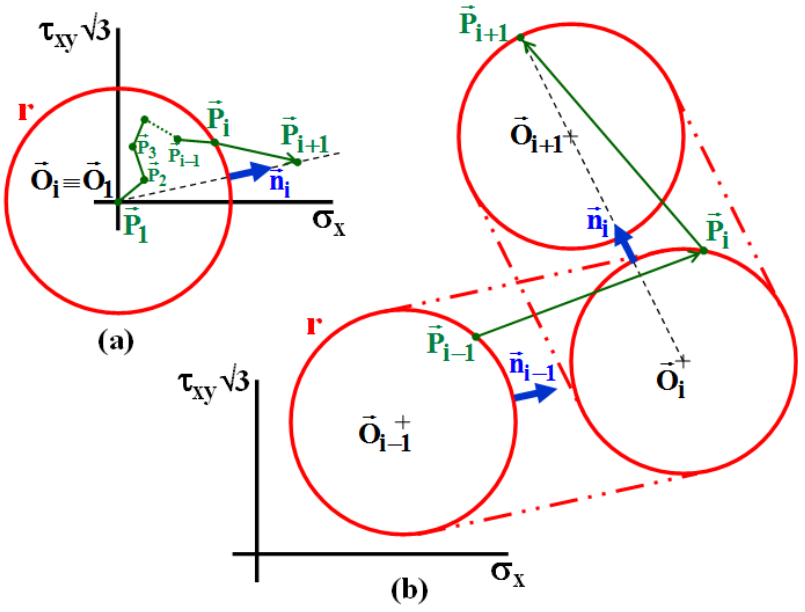

Figure 5: Hyper-surface location (a) during its first translation and (b) during a load kinking.

In summary, static filtering happens while the next peg location $\vec{P}_{i+1}$ is not outside the current hyper-sphere, i.e. $b_i \le r$ in Fig. 3, while dynamic filtering happens during hyper-sphere translations where $b_i > r$, $a_i \ge 0$, and $d_i \le r$.

Besides the initial $\vec{P}_1$, the only points that are not filtered out are the ones where some significant path kinking happens due to $d_i > r$, and the ones where a load "reversal" forces a change of more than $90^o$ in the hyper-surface translation direction, i.e. $\vec{n}_{i-1} \cdot \vec{n}_i < 0$ and therefore $a_i < 0$. In these kinking or reversal cases, the new hyper-surface translation direction could be determined by the simple rule $\vec{n}_i = (\vec{P}_{i+1} - \vec{O}_i)/b_i$ .

Note that the kinking and reversal criteria could happen at the same time, as exemplified in Fig. 5(b), where the path kinks at $\vec{P}_i$ because $\vec{P}_{i+1}$ has $d_i > r$ and thus the translation direction needs to change, while $\vec{P}_i$ can also be interpreted as a load reversal point because $\vec{n}_{i-1} \cdot \vec{n}_i < 0$ and thus $a_i < 0$.

Since the distance between $\vec{O}_i$ and $\vec{P}_{i+1}$ is equal to $b_i$, see Fig. 3, the hyper-surface center translation during kinking and/or reversal, shown in Fig. 5(b), can be calculated by

$$\vec{O}_{i+1} = \vec{O}_i + (b_i - r) \cdot \vec{n}_i \tag{3}$$

The filtered history is then composed of the initial point and all kinking and reversal points. The reversal criterion $a_i < 0$ keeps track of abrupt changes in loading direction that might characterize a "peak" condition, while the kinking criterion given by $d_i > r$ guarantees that a significant curvature of the load history path between two points is accounted for without assuming it is a straight line. Such path curvature is important in the calculation of path-equivalent stress and strain ranges, because the use of a straight path could lead to a non-conservative range calculation when compared to the actual NP curved stress or strain path. Fig. 6 summarizes the procedures involved in the multiaxial racetrack algorithm. Note that it exactly reproduces the classical uniaxial racetrack algorithm if $\vec{P}$ and $\vec{O}$ are represented by scalars.

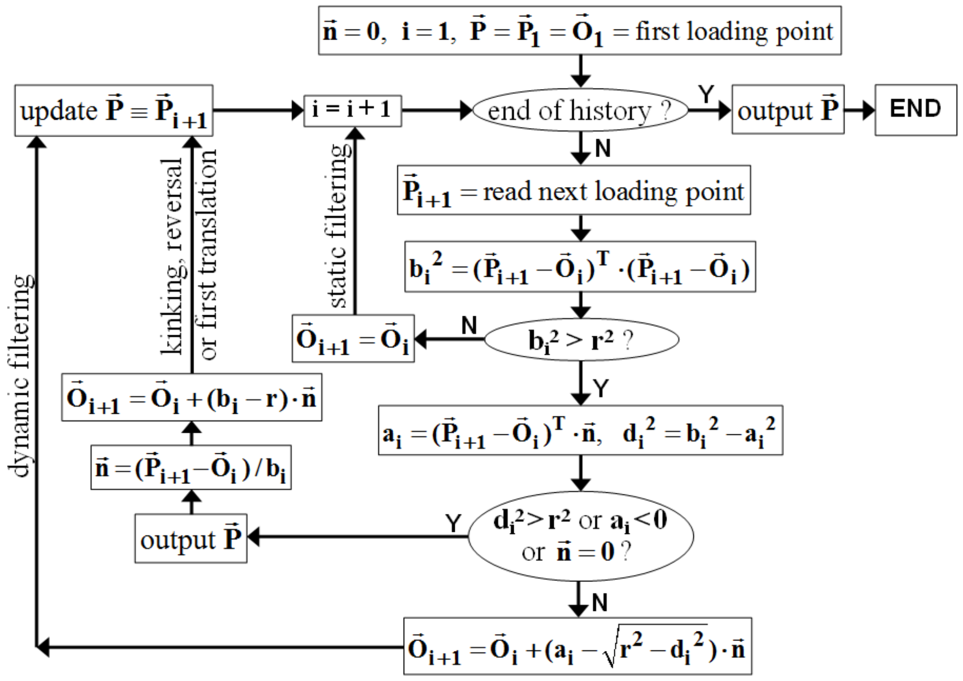
$\vec{n} = 0$, $i = 1$, $\vec{P} = \vec{P}_1 = \vec{O}_1$ = first loading point
end of history ?
Y
output $\vec{P}$
END
N
update $\vec{P} \equiv \vec{P}_{i+1}$
$i = i + 1$
$\vec{P}_{i+1}$ = read next loading point
$b_i^2 = (\vec{P}_{i+1} - \vec{O}_i)^T \cdot (\vec{P}_{i+1} - \vec{O}_i)$
$\vec{O}_{i+1} = \vec{O}_i$
N
$b_i^2 > r^2$ ?
Y
static filtering
kinking, reversal or first translation
dynamic filtering
$\vec{O}_{i+1} = \vec{O}_i + (b_i - r) \cdot \vec{n}$
$\vec{n} = (\vec{P}_{i+1} - \vec{O}_i) / b_i$
$a_i = (\vec{P}_{i+1} - \vec{O}_i)^T \cdot \vec{n}$, $d_i^2 = b_i^2 - a_i^2$
output $\vec{P}$
Y
$d_i^2 > r^2$ or $a_i < 0$ or $\vec{n} = 0$ ?
N
$\vec{O}_{i+1} = \vec{O}_i + (a_i - \sqrt{r^2 - d_i^2}) \cdot \vec{n}$

Figure 6: Multiaxial racetrack algorithm, with static and dynamically-filtered history resulting from the $\vec{P}$ output values, where $\vec{n}$ is the hyper-sphere translation direction.

## TENSION-TORSION EXAMPLE

Fig. 7(a) shows an example of a normal × effective shear stress path obtained under tension-torsion, used to exemplify the multiaxial racetrack algorithm. Fig. 7(b) shows the filtered history for a given relatively large filter amplitude *r*, where only four out of the sixteen original data points were not filtered, significantly decreasing the computational cost of multiaxial fatigue life calculations for this load history. In this figure, the points that suffered static filtering are marked with an ×, while the dynamically-filtered ones are represented with triangular markers.

Note once again the immense practical importance of these *amplitude* filter procedures. Since the calculation of multiaxial fatigue damage accumulation is an intrinsically intensive computational procedure, it is most important to eliminate from the calculation effort all points that are not essential for its result, i.e. all points that do not cause significant fatigue damage.

Fig. 7(c) compares the (dashed) original history with the filtered one. The filtered data tends to the original one as the filter amplitude *r* decreases, at the cost of increasing the number of unfiltered points, see Fig. 7(d) and (e). In practice, a relatively large *r* value can be initially chosen, and then decreased until the calculated damage converges.

Note that the adopted 6D space can be defined for example by the normal and by the effective shear components, $\left[\sigma_x \quad \sigma_y \quad \sigma_z \quad \tau_{xy}\sqrt{3} \quad \tau_{xz}\sqrt{3} \quad \tau_{yz}\sqrt{3}\right]^T$ or $\left[\varepsilon_x \quad \varepsilon_y \quad \varepsilon_z \quad \gamma_{xy}/\sqrt{3} \quad \gamma_{xz}/\sqrt{3} \quad \gamma_{yz}/\sqrt{3}\right]^T$, respectively for stress or strain histories.

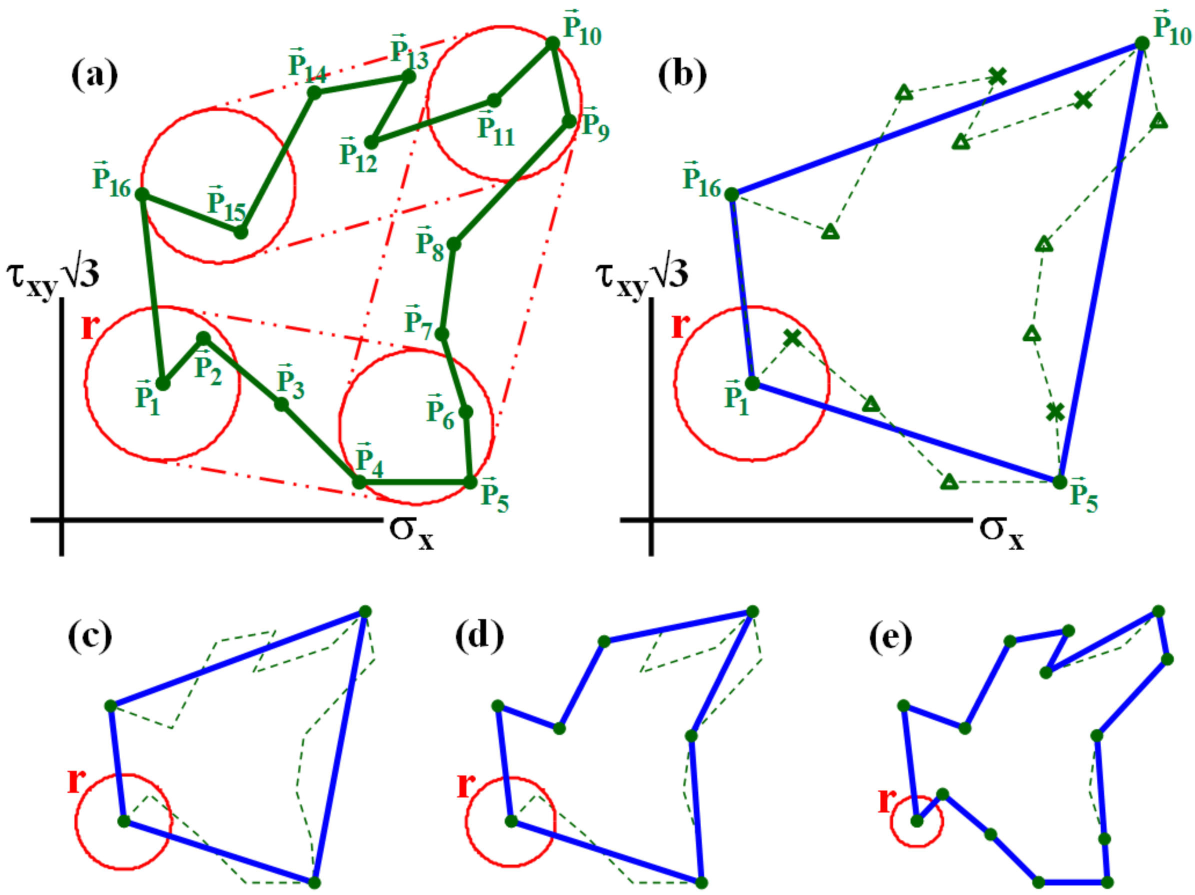

Figure 7: Multiaxial racetrack filter applied to a tension-torsion history path with 16 points, showing (a) the translating hyper-spheres, (b) the static and dynamically-filtered points (respectively × and triangles), and the effect of decreasing the filter amplitude *r*, resulting in histories with (c) four, (d) seven, and (e) fourteen points.

## CONCLUSIONS

Amplitude filters are most important in practical fatigue analyses to manage their computational cost when (as usual) the input history is oversampled, too long, and/or contains too many non-damaging low-amplitude cycles. The proposed multiaxial racetrack algorithm is very versatile, allowing the synchronous filtering of stress and strain histories acting at a given material point, based on a single scalar amplitude value.

## REFERENCES

[1] Fuchs, H.O., Nelson, D.V., Burke, M.A., Toomay, T.L., Shortcuts in Cumulative Damage Analysis, SAE Automobile Engineering Meeting Paper 730565, (1973).

[2] Nelson, D.V., Fuchs, H.O., Predictions of cumulative fatigue damage using condensed load histories, in Fatigue Under Complex Loading, SAE, (1977).

[3] Bannantine, J.A., Socie, D.F., A variable amplitude multiaxial fatigue life prediction method, Fatigue Under Biaxial and Multiaxial Loading, ESIS publ. 10 (1991) 35-51.
[4] Langlais, T.E., Vogel, J.H., Chase, T.R., Multiaxial cycle counting for critical plane methods, Int. J. Fatigue, 25 (2003) 641-647.
[5] Meggiolaro, M.A., Castro, J.T.P., An Improved Multiaxial Rainflow Algorithm for Non-Proportional Stress or Strain Histories - Part I: Enclosing Surface Methods, Int. J. Fatigue, 42 (2012) 217-226.
[6] Meggiolaro, M.A., Castro, J.T.P., Prediction of non-proportionality factors of multiaxial histories using the Moment Of Inertia method, Int. J. Fatigue, 61 (2014) 151-159.
[7] Meggiolaro, M.A., Castro, J.T.P., An Improved Multiaxial Rainflow Algorithm for Non-Proportional Stress or Strain Histories - Part II: The Modified Wang-Brown Method, Int. J. Fatigue, 42 (2012) 194-206.
[8] Bannantine, J.A., A Variable Amplitude Multiaxial Fatigue Life Prediction Method, FCP Report n.151, University of Illinois at Urbana-Champaign, (1989).