

Nested-list Approach for the Stageless Evaluation of a Vector Preisach Model based on Rotational Operators

Michael Nierla¹, Stefan J. Rupitsch¹, Manfred Kaltenbacher²

¹ Chair of Sensor Technology, University Erlangen-Nuremberg, 91052 Erlangen, Germany,

² Institute of Mechanics and Mechatronics, Technical University Vienna, 1060 Vienna, Austria
michael.nierla@fau.de

Abstract:

The Everett function is a widely used tool to evaluate the output of scalar Preisach hysteresis operators. Compared to a matrix-based implementation, the Everett function does not only allow a stageless evaluation of the switching state but, furthermore, reduces the required storage to a single list of dominant input minima and maxima. To deal with magnetic fields that change direction over time, a vector extension of the scalar Preisach model based on rotational operators can be utilized. This straight-forward extension enables an efficient description of vector hysteresis phenomena. Due to the dependency between the newly added rotational operator and the original switching operator, neither the usage of a single list nor the direct application of the Everett function is possible. In this contribution, we propose a nested-list data structure for the vector Preisach model based on rotational operators. Exploiting this data structure allows the usage of an adapted Everett function and, thus, an efficient, stageless representation of both the rotational operator as well as the switching operator.

Key words: Everett Function, Hysteresis Modeling, Vector Preisach

Introduction

In order to describe not only the effects of saturation and hysteretic branching but also the change of direction under rotating magnetic fields, vector hysteresis models have been developed. Well established approaches are based on vectorial extensions of either the Preisach hysteresis model [1,4,8,9] or the Jiles-Atherton model [2,7]. A direct comparison between scalar Preisach and Jiles-Atherton models shown in [3], reported a slight advantage of the Preisach model in terms of accuracy, especially with respect to minor loops at the cost of higher computational effort. Typical vector extensions of the scalar Preisach model lead to a further increase in computational cost, e.g., by requiring the integration of the scalar model over the whole half-space like in case of the well-established Mayergoyz vector hysteresis model [8,9]. To reduce the computational effort while maintaining a high accuracy, Sutor et. al. developed two similar light-weight vector extensions to the scalar Preisach model that exploit rotational operators to describe vector hysteresis properties [12,13]. The major drawback of these models lies in their implementation. Whereas scalar Preisach models can efficiently evaluate the hysteretic behavior using the Everett function [5,6], a less accurate and computationally more expensive

matrix-implementation has to be used for these vector models. In this contribution, we present an efficient, stageless evaluation scheme based on a nested-list data structure that aims at relieving the two main disadvantages of the matrix-based approach. Thereby, we focus on the revised vector model presented in [11]. A similar nested-list approach was already presented in [10] for the original model [12] but due to significant changes in the revised model, a separate modeling is reasonable. In the following section, the vector Preisach model based on rotational operators will shortly be presented. Afterwards, the stageless evaluation scheme and its nested-list data structure are discussed. The forth section provides a set of updating and deletion rules that ensures a correct and memory saving handling of the nested-list data structure. Section five shows a comparison between the newly developed stageless evaluation and the classical matrix-based implementation. This contribution ends with a conclusion in section six.

Vector Preisach Model based on Rotational Operators

The vector Preisach models as proposed in [11,12] extend the classical scalar Preisach model by two slightly different rotational operators. Let $\vec{e}(t) = \tilde{E}(t)/E_{\text{saturation}}$ denote the normalized input of the hysteresis operator,

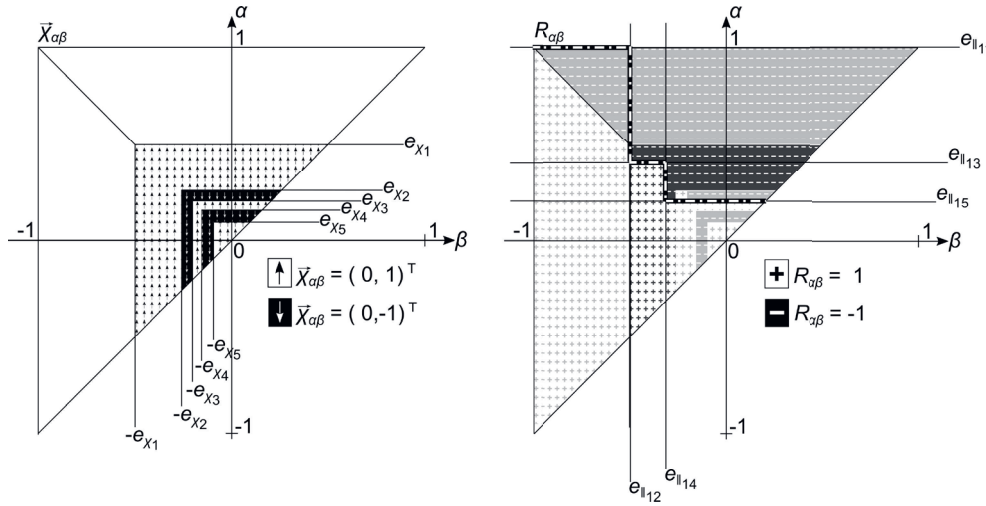


Fig. 1. **Left:** Exemplary rotation state $\tilde{X}_{\alpha\beta}$ of the vector Preisach model for $k = 0.5$, $\Delta\phi = 0$ after an input history of $(0.0, 1.0)^T$, $(0.0, -0.5)^T$, $(0.0, 0.4)^T$, $(0.0, -0.3)^T$, $(0.0, 0.2)^T$; **Right:** Corresponding switching state $R_{\alpha\beta}$; the dash-dotted line displays the stair-case separation of $R_{\alpha\beta}$ by the values $e_{||11}$ to $e_{||15}$; this separation affects only the rotation state enclosed by $\pm e_{x1}$ and $\pm e_{x2}$ (darker area).

then its normalized output $\tilde{p}(t) = \tilde{p}(t)/p_{\text{saturation}}$ is obtained by

$$\tilde{p}(t) = \iint_{\alpha \geq \beta} \mu(\alpha, \beta) R_{\alpha\beta}[\tilde{e}](t) \tilde{X}_{\alpha,\beta}[\tilde{e}](t) d\alpha d\beta$$

Thereby, $\mu(\alpha, \beta)$ stands for the Preisach weight function, $R_{\alpha\beta}[\tilde{e}](t)$ for the switching operator known from the scalar Preisach model and $\tilde{X}_{\alpha\beta}[\tilde{e}](t)$ for the newly added rotational operator. α and β are the coordinates in the triangular shaped Preisach plane

$$S = \{(\alpha, \beta) \in \mathbb{R}^2, |\alpha| \leq 1, |\beta| \leq 1, \beta \leq \alpha\}.$$

Following the revised vector model, presented in [11], the rotational operator $\tilde{X}_{\alpha\beta}[\tilde{e}](t)$ is defined as

$$\tilde{X}_{\alpha\beta}[\tilde{e}](t) = \begin{cases} \tilde{d}_{\text{new}}(t) & \text{if } e_x > \max(|\alpha|, |\beta|) \\ \tilde{d}_{\text{old}}(t) & \text{else} \end{cases}. \quad (1)$$

\tilde{d}_{old} denotes for the previous operator state. e_x is called rotational threshold and computes to

$$e_x = k \|\tilde{e}(t)\|, \quad (2)$$

with k being a material specific parameter. In case of $e_x > \max(|\alpha|, |\beta|)$, the rotational operator changes its state to

$$\tilde{d}_{\text{new}} = \begin{cases} \tilde{d}_{\Delta\phi} & \text{if } \angle(\tilde{d}_{\text{in}}, \tilde{d}_{\text{old}}) > \Delta\phi \\ \tilde{d}_{\text{old}} & \text{else} \end{cases}, \quad (3)$$

where $\tilde{d}_{\text{in}} = \tilde{e}(t)/\|\tilde{e}(t)\|$ is the direction of the current input and $\tilde{d}_{\Delta\phi}$ the unit vector that is obtained by rotating \tilde{d}_{old} towards \tilde{d}_{in} leaving an angular distance

$$\Delta\phi = \Delta\phi_0(1 - \|\tilde{e}(t)\|).$$

The value $\Delta\phi_0$ represents a second material specific parameter. As shown in the left half of Fig. 1, the presented setting rules for $\tilde{X}_{\alpha\beta}[\tilde{e}](t)$

separate the Preisach plane S into Γ -shaped areas that are bounded by the values of e_{xj} .

Besides the addition of the rotational operator $\tilde{X}_{\alpha\beta}[\tilde{e}](t)$, another crucial point of the vector Preisach model is the update rule for the switching operator $R_{\alpha\beta}[\tilde{e}](t)$. In the original scalar Preisach model, $R_{\text{scalar},\alpha\beta}[e](t)$ is set directly by the scalar input value e . A sequence (e_1, e_2, \dots, e_M) of dominant input minima and maxima will lead to a characteristic stair-case splitting of S into a positive ($R_{\text{scalar},\alpha\beta}[e](t) = 1$) and a negative ($R_{\text{scalar},\alpha\beta}[e](t) = -1$) part [5,6]. The vector Preisach model based on rotational operators utilizes the same update rule as the scalar model to set $R_{\alpha\beta}[\tilde{e}](t)$, i.e.,

$$R_{\alpha\beta}[\tilde{e}](t) = \begin{cases} +1 & \text{if } e_{||} > \alpha \\ -1 & \text{if } e_{||} < \beta \\ R_{\alpha\beta}[\tilde{e}](t^-) & \text{else} \end{cases},$$

where $R_{\alpha\beta}[\tilde{e}](t^-)$ stands for the previous value. The only difference to the scalar model lies in the usage of

$$e_{||} = \tilde{e}(t) \cdot \tilde{X}_{\alpha\beta}[\tilde{e}](t)$$

instead of e . $e_{||}$ stands for the projection of the input vector \tilde{e} onto the current rotation state $\tilde{X}_{\alpha\beta}[\tilde{e}](t)$. This adaption is physically motivated by the assumption that only the projection of the outer field along the already taken rotation direction determines whether domains switch.

Stageless Nested-List Approach: Evaluation

The scalar Preisach model can efficiently be implemented exploiting the Everett function [4]. The idea behind this approach is to not store a discretized version of the actual switching state $R_{\text{scalar},\alpha\beta}[e](t)$ but rather a list of dominant input minima and maxima. Let (e_1, e_2, \dots, e_M)

| j | \vec{e}_j | e_{χ_j} | $\vec{\chi}_{\alpha\beta,j}$ | $e_{\parallel_{ji}} = \vec{e}_i \cdot \vec{\chi}_{\alpha\beta,j}$ |
|-----|---------------|--------------|------------------------------|---|
| ① 1 | $(0, 1.0)^T$ | 0.50 | $(0, 1)^T$ | $1.0 / \max$ |
| 2 | $(0, -0.5)^T$ | 0.25 | $(0, -1)^T$ | $1.0 / \max$ |
| 3 | $(0, 0.4)^T$ | 0.20 | $(0, 1)^T$ | $1.0 / \max$ |
| 4 | $(0, -0.3)^T$ | 0.15 | $(0, -1)^T$ | $1.0 / \max$ |
| 5 | $(0, 0.2)^T$ | 0.10 | $(0, 1)^T$ | $1.0 / \max$ |



 entry gets inherited
 entry gets replaced by a smaller minimum / larger maximum

Fig.2. Nested-list data structure after inserting five input vectors \vec{e}_j with decreasing rotational threshold e_{χ_j} (shown example for $k = 0.5$); ①: a new entry is added to a previously empty outer list (rows of the table); a nested inner list of values $e_{\parallel_{ji}}$ (last column) is initialized with the projection $e_{\parallel_{11}} = (0.0, 1.0)^T \cdot (0.0, 1.0)^T$ according to (4); ②: whenever an entry is inserted into the outer list, it inherits the inner list of the previous entry; ③: after adding a new entry to the outer list, all inner lists are updated with corresponding projections $e_{\parallel_{ji}}$; ④: the application of the wiping-out rules known from the scalar Everett function removes unneeded entries from the inner list, i.e., consecutive minima / maxima of smaller / larger value overwrite previous minima / maxima of larger / smaller value.

be a list of such dominant input values, then the output of the scalar Preisach operator can be computed via

$$p(t) = \text{sign}(e_1) \mathcal{E}(-e_1, e_1) + 2 \sum_{i=1}^{M-1} \text{sign}(e_{i+1} - e_i) \mathcal{E}(e_i, e_{i+1}), \quad (4)$$

where

$$\mathcal{E}(e_i, e_{i+1}) = \int_{\Delta_{e_i, e_{i+1}}} \mu(\alpha, \beta) d\alpha d\beta \quad (5)$$

defines the Everett function.

$$\Delta_{e_i, e_{i+1}} = \{(\alpha, \beta) \in \mathbb{R}^2, \min(e_i, e_{i+1}) \leq \alpha \leq \max(e_i, e_{i+1}), \min(e_i, e_{i+1}) \leq \beta \leq \alpha\} \quad (6)$$

is the triangular area enclosed by e_i and e_{i+1} . A direct application of this approach to the vector Preisach model is not possible. This originates from the usage of $e_{\parallel} = \vec{e}(t) \cdot \vec{\chi}_{\alpha\beta}[\vec{e}](t)$ during the evaluation of $R_{\alpha\beta}[\vec{e}](t)$, which causes a dependency between switching and rotational operator. Due to that dependency, each Γ -shaped subarea corresponding to a rotation state $\vec{\chi}_{\alpha\beta,j}[\vec{e}](t)$ will be separated into a +1 and -1 part by an own set $(e_{\parallel_{j1}}, e_{\parallel_{j2}}, \dots, e_{\parallel_{jM}})$ of projection minima and maxima as depicted in Fig. 1. Therewith, it is not possible to store all values of e_{\parallel} that would be needed to reconstruct the full state of $R_{\alpha\beta}[\vec{e}](t)$ in a single list. To resolve this issue, we propose a nested-list data structure as sketched in Fig. 2. The idea behind this data structure is as follows. For each input vector \vec{e}_j , an entry is added to an

outer list. This outer list is sorted by the values of e_{χ_j} in descending order. Each entry j of the outer list stores e_{χ_j} , the corresponding rotation state $\vec{\chi}_{\alpha\beta,j}$ as well as an inner list of projections $e_{\parallel_{ji}} = \vec{\chi}_{\alpha\beta,j} \cdot \vec{e}_i$. Using this kind of data structure, the evaluation of $\vec{p}(t)$ can be performed via an adapted form of the Everett function in the following way. For each entry j of the outer list, calculate its contribution

$$\vec{p}_j(t) = \vec{\chi}_{\alpha\beta,j}(\text{sign}(e_{\parallel_{j1}}) \tilde{\mathcal{E}}(-e_{\parallel_{j1}}, e_{\parallel_{j1}}) + 2 \sum_{i=1}^{M-1} \text{sign}(e_{\parallel_{ji+1}} - e_{\parallel_{ji}}) \tilde{\mathcal{E}}(e_{\parallel_{ji}}, e_{\parallel_{ji+1}}))$$

to the overall output

$$\vec{p}(t) = \sum_{j=1}^K \vec{p}_j(t).$$

Thereby, the adapted Everett function

$$\tilde{\mathcal{E}}(e_{\parallel_{ji}}, e_{\parallel_{ji+1}}) = \int_{\Delta_{e_{\parallel_{ji}}, e_{\parallel_{ji+1}}}^{e_{\chi_j}}} \mu(\alpha, \beta) d\alpha d\beta - \int_{\Delta_{e_{\parallel_{ji}}, e_{\parallel_{ji+1}}}^{e_{\chi_j}-1}} \mu(\alpha, \beta) d\alpha d\beta \quad (7)$$

gets utilized. This adapted version restricts the triangular integration area to the boundaries of the corresponding rotation area. The restricted integration areas can be expressed as

$$\Delta_{e_{\parallel_{ji}}, e_{\parallel_{ji+1}}}^{e_{\chi_j}} = \{(\alpha, \beta) \in \mathbb{R}^2, \min(e_{\parallel_{ji}}, e_{\parallel_{ji+1}}) \leq \alpha \leq \min(\max(e_{\parallel_{ji}}, e_{\parallel_{ji+1}}), e_{\chi_j}), \max(\min(e_{\parallel_{ji}}, e_{\parallel_{ji+1}}), -e_{\chi_j}) \leq \beta \leq \alpha\}. \quad (8)$$

For $j = 0$, the second integral in (7) becomes 0, which is equivalent to setting $e_{\chi_0} = 0$,

| j | \vec{e}_j | e_{χ_j} | $\vec{\chi}_{\alpha\beta,j}$ | $e_{\parallel j} = \vec{e}_j \cdot \vec{\chi}_{\alpha\beta,j}$ | |
|-----|-----------------|--------------|------------------------------|--|------------|
| 1 | $(0, 1.0)^T$ | 0.50 | $(0, 1)^T$ | 1.0 / max | -0.5 / min |
| ⑤ 2 | $(0.39, 0.8)^T$ | 0.445 | $(39, 80)^T$ 89 | 1.0 / max | -0.5 / min |
| 3 | $(0, -0.5)^T$ | 0.25 | $(39, 80)^T$ 89 | 1.0 / max | -0.4 / min |
| 4 | $(0, 0.4)^T$ | 0.20 | $(39, 80)^T$ 89 | 1.0 / max | -0.3 / min |
| 5 | $(0, -0.3)^T$ | 0.15 | $(39, 80)^T$ 89 | 1.0 / max | -0.2 / min |
| 6 | $(0, 0.2)^T$ | 0.10 | $(39, 80)^T$ 89 | 1.0 / max | 0.2 / min |

↓ entry gets inherited ↘ entry gets replaced by a smaller minimum / larger maximum
 ↘ entry gets deleted as previous entry was replaced

Fig.3. Nested-list data structure of Fig. 2 after inserting a sixth input vectors with a rotational threshold of 0.445; ⑤: a new entry is inserted at the second position of the list due to its value of e_{χ_j} ; ⑥: new entry inherits the inner list from the entry with larger e_{χ_j} ; ⑦: the stored rotation state $\vec{\chi}_{\alpha\beta,k}$ of all entries with $e_{\chi_k} < e_{\chi_j}$ gets overwritten according to (5) (shown example for $\Delta\phi = 0$); ⑧: the inner lists gets updated according to (4) using the already overwritten rotation states $\vec{\chi}_{\alpha\beta,k}$; ⑨: apply the wiping-out rules used for the scalar model.

Stageless Nested-List Approach: Updating Rules

In the previous section, the nested-list data structure was introduced briefly and it was shown that it can be evaluated using an adapted Everett function. In the next step, we want to specify how the nested list has to be updated correctly. At first, consider the case of adding an input $\vec{e}(t)$, which rotational threshold e_{χ} is the smallest value so far. In that case, the updating is done using the steps as shown in Fig. 2:

- ① Evaluate the current rotation state $\vec{\chi}_{\alpha\beta}$ using (1) and append a new entry to the end of the outer list storing e_{χ} and $\vec{\chi}_{\alpha\beta}$.
- ② Inherit (i.e., copy) the inner list from the previous entry (if any).
- ③ For each entry j of the outer list, update the corresponding inner lists with $e_{\parallel j} = \vec{\chi}_{\alpha\beta,j} \cdot \vec{e}(t)$.
- ④ Apply the wiping-out rules as used for the scalar Preisach model [4] to remove unneeded entries from the inner lists.

In case that an input $\vec{e}(t)$ has to be added which has a larger rotational threshold e_{χ} than some or all currently stored entries, the updating steps are as follows (see Fig. 3):

- ⑤ Compute e_{χ} using (3) and insert a new entry at position n to the outer list, such that $e_{\chi_{n-1}} < e_{\chi} \leq e_{\chi_n}$. Evaluate $\vec{\chi}_{\alpha\beta}$ using (1) and store e_{χ} and $\vec{\chi}_{\alpha\beta}$ in the new entry.
- ⑥ Inherit the inner list from the previous entry.

⑦ For each entry j of the outer list with $j > n$, set $\vec{\chi}_{\alpha\beta,j} = \vec{\chi}_{\alpha\beta}$.

⑧ For each entry j of the outer list, update the corresponding inner lists with $e_{\parallel j} = \vec{\chi}_{\alpha\beta,j} \cdot \vec{e}(t)$ using the already overwritten rotation states.

⑨ Apply the wiping-out rules from the scalar Preisach model.

It shall be noticed that entries of the outer list are not removed in neither of both cases. The reason for this lies in the fact that the switching state $R_{\alpha\beta}[\vec{e}](t)$ may not directly change if the values of $\vec{\chi}_{\alpha\beta}[\vec{e}](t)$ changes, but only due to the projection $e_{\parallel} = \vec{e}(t) \cdot \vec{\chi}_{\alpha\beta}[\vec{e}](t)$ of the current input. Previously computed values of e_{\parallel} remain unaffected by later changes of $\vec{\chi}_{\alpha\beta}(t)$. This persistence of $R_{\alpha\beta}[\vec{e}](t)$ is taken care of by the inheritance steps ②/⑥ and the replacement of the rotation state in step ⑦ that is done instead of removing an outer list entry. Apparently, the lack of deletion rules of the outer list leads to an ever growing data structure. To remove entries from the outer list without violating the named persistence of $R_{\alpha\beta}[\vec{e}](t)$, one can check for the following case. If two consecutive entries $j, j+1$ of the outer list have the same rotation state $\vec{\chi}_{\alpha\beta,j} = \vec{\chi}_{\alpha\beta,j+1}$ as well as identical inner lists, the entry with smaller value e_{χ_j} can be removed from the list without the loss of information. This can easily be seen by considering (7) and (8). Unluckily, both criteria are seldom satisfied after applying the insertion steps ① to ④ or ⑤ to ⑨. A save deletion can, nevertheless, be achieved after performing the steps as depicted in Fig. 4:

| j | \vec{e}_j | e_{xj} | $\vec{X}_{\alpha\beta,j}$ | $e_{\parallel j} = \vec{e}_j \cdot \vec{X}_{\alpha\beta,j}$ | |
|-----|------------------------------------|----------|--|---|--------------|
| 1 | $(0, 1.0)^T$ | 0.50 | $(0, 1)^T$ | 0.50 / max -0.50 / min | ⑩ 0.8 / max |
| 2 | $(0.39, 0.8)^T$ | 0.445 | $(39, 80)^T$ 89 | 0.445 / max -0.445 / min | ⑪ 0.89 / max |
| ⑫ 3 | $(0, 0.5)^T$ | 0.25 | $(39, 80)^T$ 89 | 0.25 / max -0.25 / min | 0.89 / max |
| 4 | $(0, 0.4)^T$ | 0.20 | $(39, 80)^T$ 89 | 0.20 / max -0.20 / min | 0.89 / max |
| 5 | $(0, 0.3)^T$ | 0.15 | $(39, 80)^T$ 89 | 0.15 / max -0.15 / min | 0.89 / max |
| 6 | $(0, 0.2)^T$ | 0.10 | $(39, 80)^T$ 89 | 0.10 / max 0.10 / min | 0.89 / max |





 entry gets inherited
 entry gets replaced by a smaller minimum / larger maximum
 entry gets deleted as previous entry was replaced
 merged with previous row as $\vec{X}_{\alpha\beta,j}$ and all $e_{\parallel j}$ are the same

Fig.4. Final state of the nested-list data structure after applying optional simplification steps; ⑩: after adding a new entry to the outer list but before inserting a new value to the inner lists, restrict all previously stored entries with $e_{\parallel ji} > e_{xj}$ to e_{xj} and all entries with $e_{\parallel ji} < -e_{xj}$ to $-e_{xj}$; ⑪ insert the unrestricted new values of $e_{\parallel ji}$ to the inner lists and apply the wiping-out rules of the scalar Preisach model; ⑫: starting at the end of the list, remove all entries k from the outer list, if $\vec{X}_{\alpha\beta,k} = \vec{X}_{\alpha\beta,k-1}$ and $e_{\parallel ki} = e_{\parallel k-1i} \forall i$.

⑩ Before updating the inner list according to step ③ or ⑧, restrict all values $e_{\parallel ji}$ of the inner list to the corresponding values of e_{xj} , i.e., set all values $e_{\parallel ji} \leq -e_{xj}$ to $-e_{xj}$ and all values $e_{\parallel ji} \geq e_{xj}$ to e_{xj} .

⑪ Update the inner list with the current, unrestricted value $e_{\parallel j} = \vec{X}_{\alpha\beta,j} \cdot \vec{e}(t)$, then apply the wiping out step ④ or ⑨.

⑫ Starting with the last entry of the outer list, remove each entry j , if $\vec{X}_{\alpha\beta,j} = \vec{X}_{\alpha\beta,j-1}$ and $e_{\parallel ki} = e_{\parallel k-1i} \forall i$.

Results

After proposing the stageless, nested-list approach for the evaluation of the vector Preisach model, a comparison to the formerly required matrix-based evaluation shall be given. The matrix-based approach is based on the discretization of the Preisach plane S into N rows and columns. For each discrete cell (p, q) with coordinates (α_p, β_q) a switching state $R_{\alpha_p\beta_q}[\vec{e}](t)$ and a rotation state $\vec{X}_{\alpha_p\beta_q}[\vec{e}](t)$ can be evaluated. These values are stored in matrices of $N \times N$ elements. The evaluation of $\vec{p}(t)$, finally, is performed via matrix-matrix multiplications. Obviously, this kind of discretization only reacts to input variations in steps of $\Delta e = 2/N$. Figure 5 displays the normalized in- and output of the vector Preisach operator for the matrix-based approach with $N = 10, 100, 1000$ as well as for the presented stageless evaluation scheme. For $N = 10$ and $N = 100$ the computed output clearly suffers from discrete steppings, although the height of the discrete steps reduces strongly from $N = 10$ to $N = 100$. Taking a closer look at the output

curves reveals that for $N = 1000$, the output of the hysteresis operator still is non-smooth and shows discrete stepping. On the other hand, the nested-list approach produces a stageless, i.e., smooth output. Furthermore as seen by looking at the results under logarithmic scaling, the proposed nested-list approach is able to resolve the output signal up to machine precision, whereas the minimal possible output of the matrix based schemes scales with $1/N^2$. At this point, it has to be stated that the accuracy of the nested-list approach is limited by the evaluation of the integrals in the adapted Everett function (7). As far as these integrals can be solved exactly, i.e., due to an analytically integrable weighting function $\mu(\alpha, \beta)$, the evaluation via the Everett function will be precise. In practice, however, this property seldom can be found even if analytical functions like the $\mu_{\text{Dat}}(\alpha, \beta)$ -function [11] are used. In that case, we propose the following approach. Besides the nested-list data structure, a matrix of N rows and columns is used to store a discretized form of $\mu(\alpha, \beta)$. N should be chosen sufficiently large to represent the weighting function as accurate as needed. The evaluation of the Everett integrals is performed by summing up all discrete weights that are inside the integral area. Cells with coordinates (α_p, β_q) that are only partially overlapped by the integral bounds have to be scaled by the amount of actual overlap. The shown results for the nested-list approach

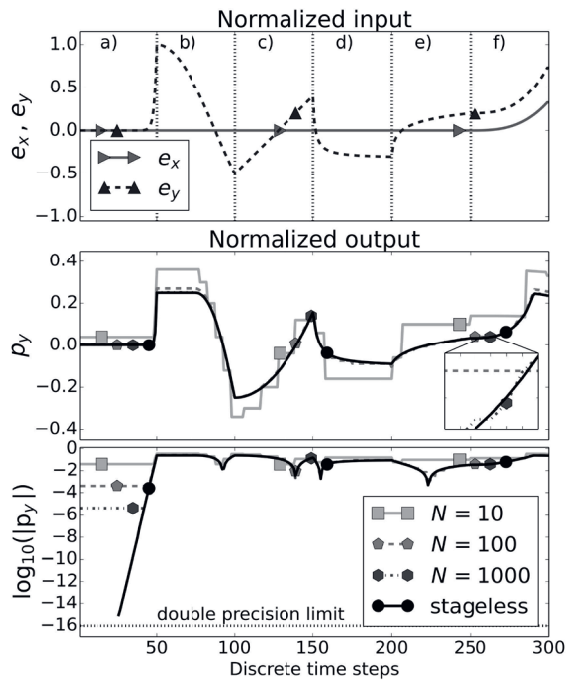


Fig. 5. **Top:** Normalized two dimensional input $\vec{e} = (e_x, e_y)^T$ of the vector Preisach operator; a) exponential increase; b) cosine decrease; c) linear increase; d) $1/t$ decrease; e) logarithmic increase; f) t^3 increase; **Middle:** Normalized y-component of the output $\vec{p} = (p_x, p_y)^T$ for matrix-based implementation ($N = 10, 100, 1000$) and nested-list approach (stageless); for $N = 10$ and $N = 100$, discretizational errors are clearly visible; **Close-up:** Even for $N = 1000$, the output is non-smooth, i.e., features discrete steps; **Bottom:** logarithmic output; nested-list approach can resolve output up to double precision limit; accuracy limit of matrix-based implementation scales with $1/N^2$.

were calculated using this kind of approach with $N = 10, 100, 1000$. Due to a used constant weight function $\mu(\alpha, \beta) = 0.5$ no difference could be observed in the results. Of course, the computational time increases with N . However, compared to a pure matrix-based approach and for the shown example, the nested-list approach achieved a speed-up of 2.5 for $N = 100$ and a speed-up of 30 for $N = 1000$. Therewith, it is not only more accurate but also computationally more efficient.

Conclusion

In this contribution, we presented a newly developed evaluation scheme for a vector Preisach model based on rotational operators. This scheme exploits a nested-list data structure and is able to evaluate the vector Preisach operator in a stageless way. Appropriate updating and deletion rules were presented that reduce the amount of stored data to a minimum. A comparison to the previously required matrix-based evaluation

approach demonstrated a clear superiority in terms of accuracy and runtime.

References

- [1] A. A. Adly, S. K. Abd-El_Hafiz, Efficient vector hysteresis modeling using rotationally coupled step functions, *Physica B* 407(9), 1350-1353 (2012); doi: 10.1016/j.physb.2011.05.053
- [2] A. J. Bergqvist, A simple vector generalization of the Jiles-Atherton model of hysteresis, *IEEE Trans. Magn.* 32(5), 4213-4215 (1996); doi: 10.1109/20.539337
- [3] A. Benabou, S. Clenet, F. Piriou, Comparison of Preisach and Jiles-Atherton models to take into account hysteresis phenomenon for finite element analysis, *J. Magn. Magn. Mater* 261(1-2), 139-160 (2003); doi: 10.1016/S0304-8853(02)01463-4
- [4] G. R. Kahler, E. D. Torre, E. Cardelli, Implementation of the preisach-stoner-wohlfarth classical vector model, *IEEE Trans. Magn.* 46(1), 21-28 (2010); doi: 10.1109/TMAG.2009.2030676
- [5] B. Kaltenbacher, M. Kaltenbacher, Modelling and iterative identification of hysteresis via Preisach operators in pdes, In: J. Kraus, U. Langer, ed. *Lectures on Advanced Computational Methods in Mechanics*, Berlin: De Gruyter, 1-46 (2007)
- [6] M. Kaltenbacher, *Numerical Simulation of Mechatronic Sensors and Actuators*, Berlin: Springer (2015)
- [7] J. V. Leite, N. Sadowski, P. Kuo-Peng, N. J. Batistela, J. P. A. Bastos, A. A. de Espindola, Inverse Jiles-Atherton Vector Hysteresis Model, *IEEE Trans. Magn.* 40(4), 1769-1775 (2004); doi: 10.1109/TMAG.2004.830998
- [8] I. D. Mayergoyz, Mathematical models of hysteresis, *IEEE Trans. Magn.* 22(5), 603-608 (1986); doi: 10.1109/TMAG.1986.1064347
- [9] I. D. Mayergoyz, *Mathematical Models of Hysteresis and Their Applications*, New York: Elsevier (2003)
- [10] M. Nierla, A. Sutor, S. J. Rupitsch, M. Kaltenbacher, Stageless Evaluation for a Vector Preisach Model based on Rotational Operators, Proceedings of 16th International IGTE Symposium, Graz Austria (2016)
- [11] A. Sutor, S. Bi, R. Lerch, Identification and verification of a preisach-based vector model for ferromagnetic materials, *Applied Physics A* 118(3), 939-944 (2015); doi: 10.1007/s00339-014-8817-1
- [12] A. Sutor, J. Kallwies, R. Lerch, An efficient vector Preisach hysteresis model based on a novel rotational operator, *Applied Physics A* 111(7), 1-3 (2012); doi: 10.1063/1.3672069
- [13] A. Sutor, S. J. Rupitsch, R. Lerch, A preisach-based hysteresis model for magnetic and ferromagnetic hysteresis, *Applied Physics A* 100(2), 425-430 (2010); doi: 10.1007/s00339-010-5884-9