Magnetohydrodynamic eigenfunction classification with a Neural Network

M. D. Kuczyński, M. Borchardt, R. Kleiber, A. Könies, C. Nührenberg

Max Planck Institute for Plasma Physics
Greifswald, Germany

Abstract

We present a Fourier-decomposition-based approach aided by a Neural Network for the classification of the eigenfunctions of an operator appearing in ideal magnetohydrodynamics. The Neural Network is trained on individual Fourier modes, which enhances the robustness of the classification. In our tests, the algorithm correctly classified 93.5% of the data and returned the remaining 6.5% for manual classification. The probability of misidentifying the eigenfunctions is estimated as 0.03%. The discussion is kept quite general allowing for potential applications in other fields.

Keywords: Magnetohydrodynamics, Neural Networks, Alfvén Waves

1. Introduction

Magnetohydrodynamics (MHD) serves as the working horse in explaining equilibrium states of plasmas in magnetic fields and their stability in a wide range of plasma parameters.

For the development and design of fusion devices, the stability of the configurations of interest plays an important role. In the seminal work by Bernstein et al. [1] it was shown that information on stability can be obtained by minimizing the energy of these systems. For small perturbations, this leads to the so-called energy principle of ideal MHD.

Normal modes for small-field perturbations can be associated with a generalized eigenvalue problem [2]. There is a large number of numerical approaches to solve this problem using a variety of methods for the numerical

Email address: mkuc@ipp.mpg.de (M. D. Kuczyński)

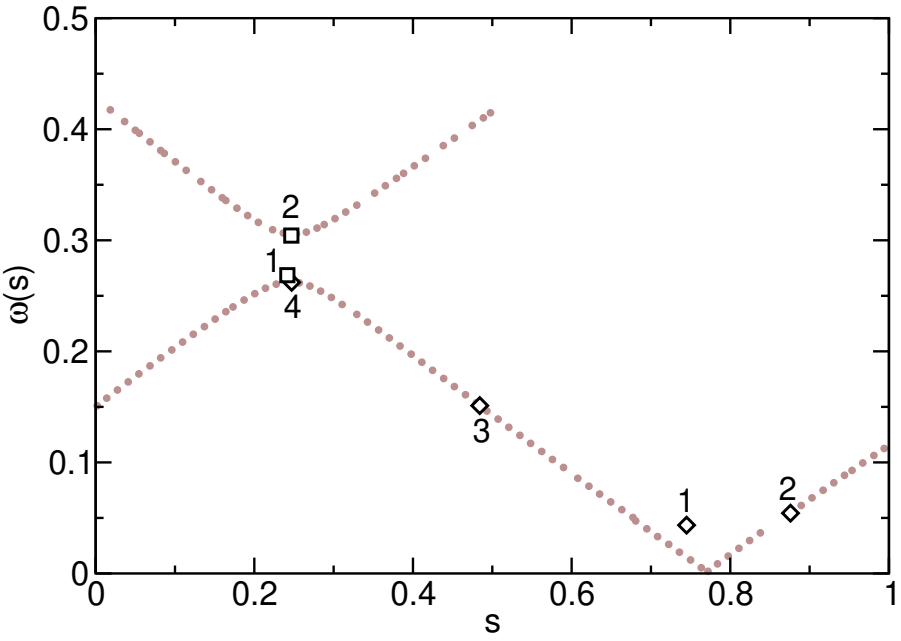

Figure 1: Example MHD spectrum. Each point corresponds to one eigenvalue. The frequency shown is that of the eigenvalue and the radial position is the maximum of the associated eigenfunction. Example eigenfunctions (enumerated boxes and diamonds) are shown in Fig. 2.

discretisation of the perturbations in space [3, 4, 5]. In case of ideal MHD, which treats the plasma as a perfectly conducting fluid and does not include resistivity nor viscosity, the eigenvalue problem is self-adjoint with real eigenvalues ω^2. Negative eigenvalues correspond to instabilities, while positive ones describe stable plasma oscillations. The solution of this problem allows one to calculate the spectrum of sound and Alfvén waves [2, 6].

The spectrum (Fig. 1) consists of discrete (continuously differentiable) and continuum (singular) modes. The latter ones are localized near flux surfaces (labeled by s) and form branches $\omega(s)$, which are named according to their physical properties. Examples of continuum and discrete modes are shown in Fig. 2.

In magnetic fields which are homogeneous, the branches are usually continuous and can intersect. In contrast, in shaped plasmas, due to the interaction mediated by the inhomogeneities of the magnetic field, the crossings resolve into gaps similarly to the electron band structure in solid states physics. In these gaps global (square-integrable) modes may exist, as is visible in Fig.

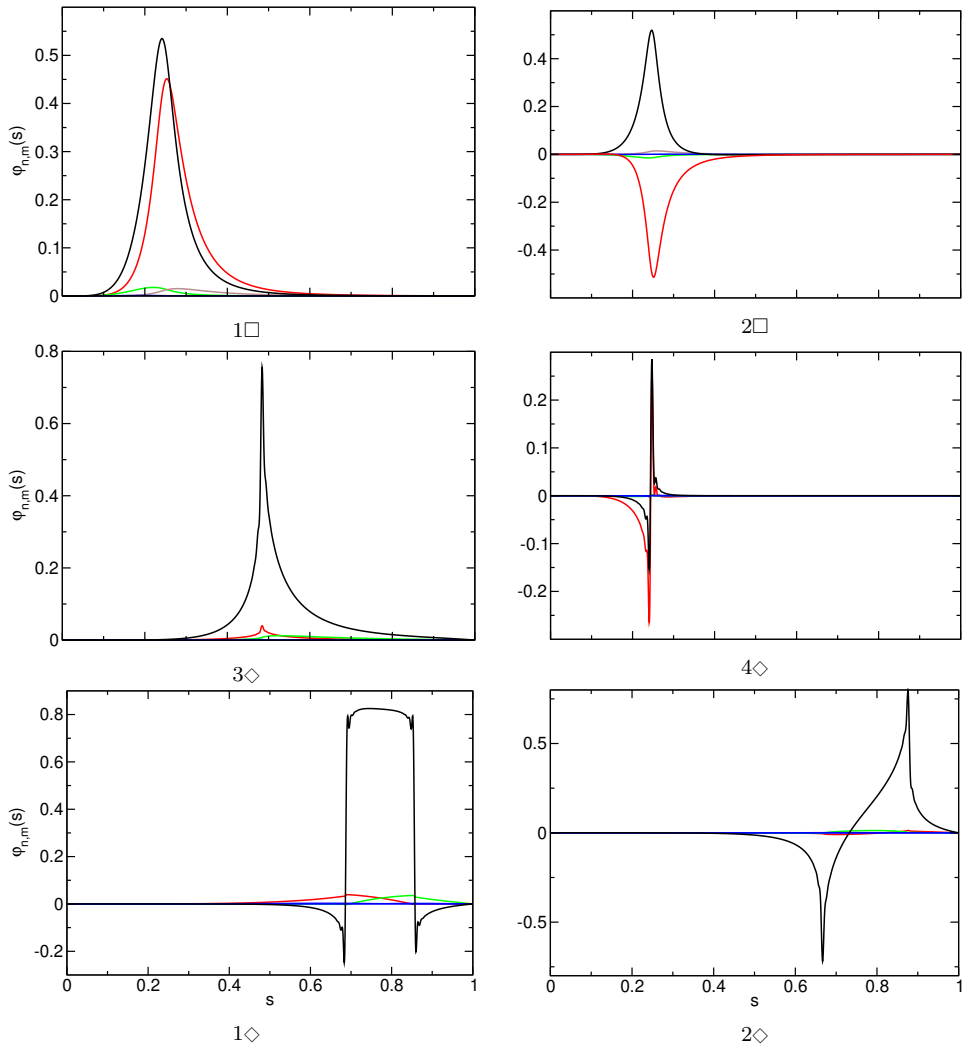

Figure 2: Example eigenfunctions selected from the spectrum shown in Fig. 1. Different colours correspond to different Fourier modes. The horizontal axis denotes the flux surface label. The only two gap modes in the spectrum are visible in panels 1□ and 2□. Some continuum modes are shown in panels 1◇-4◇. There exist examples of other modes which do not strictly belong to the mentioned two classes. However, the algorithm can incorporate an arbitrary number of classes.

1.

In practice, especially the unstable modes are of interest as they may harm plasma performance, but also the stable global modes are often analysed in order to explain plasma oscillations observed in experiments and to study their excitation by resonant interaction with fast particles. The latter process again may impact machine performance.

While for axisymmetric tokamaks the arising eigenvalue problems are of moderate size even if high resolution is required, for stellarators, their lack of a continuous symmetry and the increased configuration complexity lead to much larger matrices. Consequently, the number of eigenvalues forming the MHD spectrum is also much larger. When scanning through numerically calculated eigenvalues for a certain case, converged and unconverged eigenvalues have to be distinguished. Furthermore, a distinction between modes belonging to the continuum and global modes is desirable. Usually the number of eigenmodes delivered by an eigenvalue code is very large, and manually classifying the modes is a rather tedious task. Modes belonging to the MHD continuum are usually heavily damped [7, 8] and are not observed in experiments while global modes may resonantly interact with e.g. a fast particle population and thus become unstable [9, 10]. A distinction by the damping rate is not practical because its calculation is a task requiring considerable effort [7, 11, 12].

In this work we apply a supervised Neural Network (NN) based on the architecture of [13] to classify the MHD eigenmodes. We demonstrate that a simple implementation of a NN leads to satisfying results which can, without a doubt, be further improved. We note that the creation of the perfect machine learning model with the highest possible performance for this problem is not the focus of this article.

2. Theoretical Formulation

The desired result of an eigenfunction Classification Algorithm (ECA) is to associate a label $l \in \mathbb{L}$ to an input eigenfunction $f(s, \phi, \theta)$ signifying whether the mode is singular, $l = 1$, or not[1], $l = 0$. Since the potentially singular behaviour of an eigenfunction is best seen by looking at its Fourier components, it is first decomposed into Fourier modes through a two dimensional Fourier transform (FT) over the toroidal ϕ and poloidal θ directions.

[1] More labels can be introduced if it is desired.

Consequently the Fourier modes $\varphi_{n,m}(s)$ are functions of only the radial variable s:

$$f(s,\phi,\theta) = \sum_{n,m} \varphi_{n,m}(s) e^{i(n\phi+m\theta)}. \tag{1}$$

We assume that each Fourier mode $\varphi_{n,m}(s)$ can itself be classified with a label $l_{n,m} \in \mathbb{L}$. The algorithm then proceeds in two steps: First, a total of $N_{n,m}$ Fourier modes with largest absolute value are selected, giving a set of indices $\mathcal{S}$. Next, the NN assigns a label $l_{m,n}$ to each Fourier mode with $(m,n) \in \mathcal{S}$. Formally this can be written:

$$f(s,\phi,\theta) \overset{\text{FT}}{\mapsto} \varphi_{n,m}(s), \tag{2}$$

$$\forall (n,m) \in \mathcal{S}; \quad \varphi_{n,m}(s) \overset{\text{NN}}{\mapsto} l_{n,m}. \tag{3}$$

In general the Fourier modes are complex valued, so we assume that it is sufficient to only consider their real part[2]. As a consequence, we reformulate step two into finding the mapping:

$$\Re\{\varphi_{n,m}(s)\} \overset{\text{NN}}{\mapsto} l_{n,m} \tag{4}$$

(from now on we shall treat the Fourier modes as real valued functions, i.e. we will write $\varphi_{n,m}(s) \equiv \Re\{\varphi_{n,m}(s)\}$). The architecture and key hyperparameters of this NN are discussed in the following sections.

Not all the Fourier modes $\varphi_{n,m}$ of an eigenfunction need to have the same label $l_{n,m}$. Even if the eigenfunction is singular, only some of its Fourier components may be singular while others may be regular. Additionally there is the possibility that the NN may simply classify a Fourier component wrongly. Therefore, the last step of the algorithm is to apply a filter[3] F which infers the true label l of the eigenfunction from the set $\{l_{m,n}\}$:

$$\{l_{n,m}\}_{(n,m)\in\mathcal{S}} \overset{\text{F}}{\mapsto} l. \tag{5}$$

The proposed eigenfunction classification algorithm is summarised in Fig. 3. The main advantage of this approach is its robustness. The overall performance of the ECA is expected to be greater than the performance of the NN.

[2]The NN can easily be extended to cope with complex values. Here, however, we consider only the real part for simplicity.

[3]As an example, this can be just the most frequent member of $\{l_{m,n}\}$.

This is because even if the NN mislabels some Fourier modes, an appropriate choice of F will overcome this problem and the final label of ECA will be the correct one.

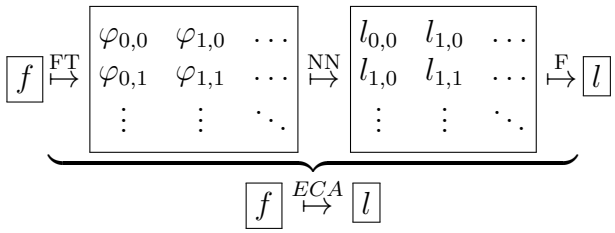

Figure 3: Schematic breakdown of the ECA. FT represents a 2D Fourier decomposition of the eigenfunction f. NN is the Neural Network that assigns a label $l_{m,n}$ to each of the Fourier modes $\varphi_{m,n}$. F stands for the filter that infers the eigenfunction label l from various $l_{m,n}$.

3. NN Architecture

This section focuses on the technical details of the NN. The code was implemented in Python with the use of the Keras library [14]. The architecture is heavily inspired by [13] and is a variant of what the authors refer to as Fully Convolutional Networks (FCN).

3.1. Forward Propagation

The FCN contains a total of 3 hidden layers. Each of them consists of a convolutional layer followed by a Batch Normalization layer (BN) and a Rectified Linear Unit (ReLU) activation layer. Formally, the output $\mathbf{h}$ of such a hidden layer given input $\mathbf{x}$ is written as:

$$\begin{aligned} \mathbf{y} &= W \otimes \mathbf{x} + \mathbf{b} \\ \mathbf{s} &= \mathrm{BN}(\mathbf{y}) \\ \mathbf{h} &= \mathrm{ReLU}(\mathbf{s}). \end{aligned} \tag{6}$$

Here, $\otimes$ is the 1D convolution operator with kernel sizes $\{k_1, k_2, k_3\}$ correspondingly to each hidden layer. The convolution is performed without striding. W is the matrix of weights and $\mathbf{b}$ is the bias term. The number of filters in each hidden layer is $\{f_1, f_2, f_3\}$ accordingly. After the convolutional

layers, the features are fed into a global average pooling layer and finally the labels $l_{n,m}$ are extracted via a softmax layer. Figure 4 summarises the FCN architecture described above (the reader can refer to [15] for details on NN terminology).

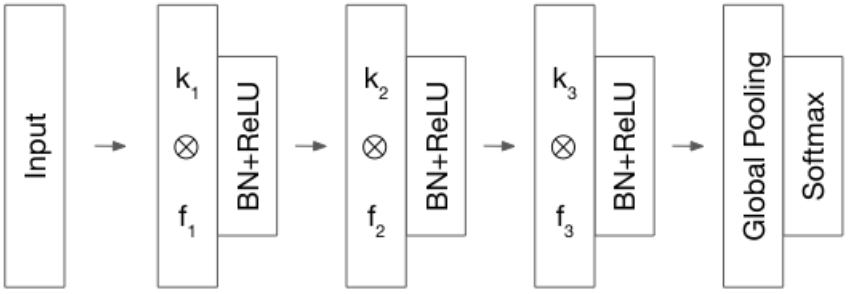

Figure 4: The architecture of the FCN.

3.2. Backward Propagation

The model is trained with an Adam optimizer with a learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon =$ 1e-8. The loss function is the categorical cross entropy. The training is stopped when the validation accuracy decreases for two subsequent epochs.

4. Data Preparation

In this section we discuss how the data were prepared for training. We also show an approach for generating artificial data resembling the gap type Fourier modes. Finally, we discuss a criterion that was used to measure the performance and to tune the hyperparameters of the FCN.

4.1. Desired Output and Input

In our application, we focus on the binary classification $\mathbb{L} = \{0, 1\}$, however, the algorithm presented supports an arbitrary number of classes. Here, the label $l = 0$ corresponds to a gap mode and the label $l = 1$ to other modes. The latter class contains mainly continuum modes, modes with mixed gap and continuum characteristics, and also modes dominated by numerical noise. Examples of eigenfunctions decomposed into Fourier modes from each class

are shown in Fig. 5. To avoid confusion we refer to labels l of the eigenfunction as gap- or non-gap modes, but the labels $l_{n,m}$ of $\varphi_{n,m}$ are said to be of gap- or non-gap type.

The input of the FCN needs to have a fixed size. Each of the Fourier modes was interpolated using 3rd order splines to contain $n_r = 512$ radial points (note that r now labels the radial point, $r \in [1, n_r]$). Changing the number of radial points in a sample can lead to problems, for example, smoothing out discontinuities or missing high frequency components. However, it was found that a 3rd order spline with $n_r = 512$ or more radial points is optimal. Lowering the interpolation order gave slightly less accurate results.

Since the calculated Fourier modes are prone to numerical errors, we chose to only use a few $N_{n,m}$ Fourier modes for training. The discussion below considers the case $N_{n,m} = 3$, which we compare later with the choice of $N_{n,m} = 4$. For smaller harmonics the numerical noise often blurs the crucial characteristics of the Fourier modes to the point that it is not reasonable to include them in the classification.

Lastly, for the benefit of the convergence of the training, the Fourier modes were rescaled by:

$$\tilde{\varphi}_{n,m}(r) \equiv \frac{\varphi_{n,m}(r)}{\max\limits_{r\in[1,n_r]} \{|\varphi_{n,m}(r)|\}}. \tag{7}$$

The above input preprocessing steps are summarised in Fig. 5, where we show the spectral decomposition of an example eigenfunction and the corresponding normalised input.

4.2. Generation of Training Data

For training, a total of 12455 pre-classified Fourier modes therein $G_0 = 67$ gap type and $N_0 = 12388$ non-gap type were manually classified. The training data were taken from two different sources: results from the CAS3D [16] and the CKA [17] code. Each of the codes was run in cylinder (minor radius 1m, length 20πm), tokamak (circular cross-section, minor radius 1m, aspect ratio 4) and stellarator (W7-AS [18] and W7-X [19]) geometry in order to provide plenty of mode structure. Clearly, the class of gap modes was underrepresented which is due to the fact that they occur much more rarely in real data. To overcome this difficulty we relied on artificial data generation. Based on the observation that, in contrast to all the Fourier

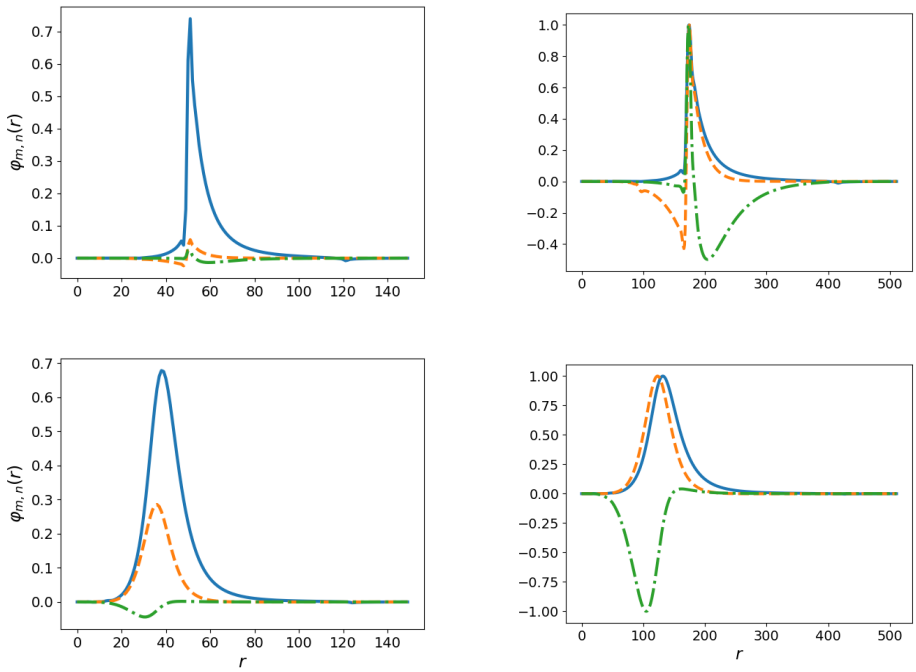

Figure 5: Example eigenfunctions decomposed into Fourier modes. Top: a continuum mode before (left) and after (right) preprocessing. Bottom: the same for a gap mode. During preprocessing the $N_{n,m} = 3$ harmonics largest in absolute value were selected, interpolated to contain 512 radial points and rescaled.

modes in class $l_{n,m} = 1$, the gap type Fourier modes are smooth functions, we created new gap type functions by applying the following techniques:

1. *Inclusion of elementary smooth functions.*

   We considered:

$$\Phi_k(r) = \sin\left(\pi k \frac{r}{n_r}\right), \tag{8}$$

$$\Psi_k(r) = e^{-2\frac{r}{n_r}} \sin\left(\pi k \frac{r}{n_r}\right), \tag{9}$$

   for an integer $k \leq 5$. After normalisation, $\{\tilde{\Phi}_k(r)\}$ and $\{\tilde{\Psi}_k(r)\}$ were added to the database. Adding S smooth functions results in an increase in the number of gap modes: $G \to G + S$.

2. *Sum of two Fourier modes.*

   Since the sum of two gap modes resembles a gap mode, the normalised sums $\tilde{\Phi}(r)$, where $\Phi(r) = \tilde{\varphi}_{n,m}(r) + \tilde{\varphi}_{n',m'}(r)$, $m > m'$ and $n > n'$ were added to the database. The corresponding increase in the number of gap type Fourier modes is $G \to (G^2+G)/2$. This step can be reiterated to further increase G, but this poses a risk of creating complex, non-smooth shapes with each iteration. We decided to do this step twice.

3. *Reflection symmetry.*

   Since reflections preserve the smooth character of functions, we included $\tilde{\Phi}(r) = -\tilde{\varphi}_{n,m}(r)$ and $\tilde{\Phi}(r) = \tilde{\varphi}_{n,m}(n_r - r + 1)$. This results in in $G \to 2G$ for each case.

As a result, the training data contained $G = 52012$ and $N = 27388$ Fourier modes. N was augmented using only the method 2 mentioned above. The final database is dominated by gap type Fourier modes what will affect the training in a favourable way. The FCN will favour the precision of class $l_{n,m} = 0$ detection since it appears more often in the training data. This is a desirable property, which can also be achieved, for example, by using non-constant weights in the calculation of the loss function.

5. FCN Hyperparameters

In this simple binary classification the FCN's performance can be evaluated with two parameters: $\hat{g}$ and $\hat{n}$. We define $\hat{g}$ to be the gap type Fourier modes detection coefficient, i.e. the ratio of the class zero Fourier modes detected by the FCN to the total number of class zero Fourier modes in the data. $\hat{n}$ is defined accordingly for non-gap type Fourier modes. Since the presented algorithm aims at filtering out the gap modes, the FCN architecture and the hyperparameters[4] were chosen in an attempt to maximise $\hat{g}$ while ensuring that $\hat{n}$ was reasonably high.

To test the generalizability of the FCN and tweak its hyperparameters, the following procedure was proposed:

1. A test subset, i, was excluded from the training set. The FCN was then trained on the remaining subsets further splitting them into training sample and test sample.
2. The predicted labels were compared with those from the pre-classification and counted for all Fourier modes in the test subset. That is, $c_i^{(0,0)}$ ($c_i^{(1,1)}$) was the total number of gap (non-gap) type Fourier modes correctly classified and $c_i^{(0,1)}$ ($c_i^{(1,0)}$) was the total number of gap (non-gap) type Fourier modes falsely classified. This information results in a confusion matrix C_i. An example is shown in Fig. 6.
3. A new test subset was chosen and the procedure was repeated.

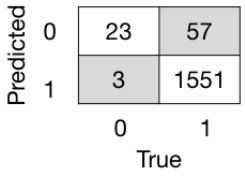

Figure 6: An example result of the generalization test. White: the count of correctly classified Fourier modes. Grey: the misclassified Fourier modes.

Once the loop ran through all the subsets in the database, the following two ratios were defined to measure performance:

[4]For example, the filter and kernel sizes, activation function type, etc.

Tests for three dominant Fourier modes

type	$\hat{g}$	$\hat{n}$	f_1	f_2	f_3	k_1	k_2	k_3
1*	0.65	0.95	128	256	128	8	5	3
2	0.86	0.93	64	128	64	8	5	3
3	0.61	0.96	128	64	128	8	5	3
4	0.82	0.95	128	256	128	16	8	4
5	0.80	0.96	64	128	64	16	8	4
6	0.74	0.95	128	64	128	16	8	4
7	0.67	0.94	64	32	16	32	16	8
8	0.86	0.96	128	64	32	32	16	8

Tests for four dominant Fourier modes†

type	$\hat{g}$	$\hat{n}$	f_1	f_2	f_3	k_1	k_2	k_3
1*	0.75	0.95	128	256	128	8	5	3
2	0.85	0.89	64	128	64	8	5	3
3	0.82	0.94	128	64	128	8	5	3
4	0.81	0.93	128	256	128	16	8	4
5	0.90	0.94	64	128	64	16	8	4
6	0.75	0.94	128	64	128	16	8	4
7	0.84	0.92	64	32	16	32	16	8
8	0.86	0.94	128	64	32	32	16	8

Table 1: The ratios $\hat{g}$ and $\hat{n}$ for different choices of the hyperparameters f_{1-3} and k_{1-3}.
* Kernel and filter sizes suggested in [13].
† Note that adding the 4th strongest Fourier mode to the data set increases G and N compared to the values given in section 4.2

$$\hat{g} = \frac{1}{G_0} \sum_i c_i^{(0,0)}, \quad \hat{n} = \frac{1}{N_0} \sum_i c_i^{(1,1)}. \tag{10}$$

We summarise this information for eight different sets of hyperparameters in Tab. 1 and also present the corresponding results for $N_{n,m} = 4$ dominant harmonics. One final property that we imposed on the FCN is that $\hat{g}$ and $\hat{n}$ should not vary drastically when $N_{n,m}$ is changed. It follows that the FCN type most consistent across the tested $N_{n,m}$ values is the FCN of type 8. These are the hyperparameters that we choose in the following (together with $N_{n,m} = 4$).

6. Practical Application

We split the output of the FCN into five groups, each characterised by the number of gap type Fourier modes detected for each eigenvalue. For example, group 0 will have zero class $l_{n,m} = 0$ Fourier modes and four class $l_{n,m} = 1$ Fourier modes, etc[5]. We prepared two test sets A and B, comprising 85 and 117 eigenfunctions, respectively. The result of running the FCN on the test sets is summarised in Tab. 2. As we noted in the introduction, in an ideal situation the FCN should place the eigenfunctions in either the group 0 corresponding to only non-gap type Fourier modes or the group 4 corresponding to only gap type Fourier modes. This is the case for the test set A. In the test set B, however, we have some eigenfunctions in groups 1, 2 and 3. Here, we suggest that they are returned as 'unclassified' so that they have to be labelled manually. Alternatively, one could define a filter that infers the labels l from $l_{n,m}$ even in the mixed classes. However, we see that even the most restrictive approach significantly reduces the manual labour needed for the classification of the eigenfunctions.

7. Conclusions

We have presented a Fourier-decomposition-based approach aided with a Neural Network for the classification of ideal MHD operator eigenfunctions. In the past, this classification was usually done by visual inspection, which can be tedious. Typically, one is not interested in the plethora of singular modes but only in the rare, global gap modes, which can be destabilised by fast particles. Training the NN to recognise the individual Fourier modes allows for a secure algorithm for gap modes detection. This approach has been applied to Alfvén modes calculated with two different codes: CAS3D and CKA. In the presented example we managed to filter out correctly 93.6% of the data, leaving the remaining 6.4% for classification by a user defined filtering procedure. The probability of misidentifying each gap type Fourier mode of an eigenfunction is on average $1 - \hat{g} = 14\%$. Therefore, the probability of misidentifying all its four components is roughly $0.14^4 \approx 0.03\%$,

[5] We could treat groups like $(1, 0, 0, 0)$ and $(0, 0, 0, 1)$ differently. An eigenfunction being a member of the latter group might hint towards the fact that the numerical noise overwhelms the important characteristics in the fourth most dominant mode. With our proposed group classification we do not consider this subtlety in order to simplify the presentation of the results.

Test set A

NN

group	*count*
0	83
1	0
2	0
3	0
4	3

actual

group	*count*
non-gap	83
gap	3

Test set B

NN

group	*count*
0	104
1	9
2	3
3	1
4	0

actual

group	*count*
non-gap	117
gap	0

Table 2: The result of applying the ECA on the test sets.

which is sufficient for practical applications. We conclude that NNs are well suited for the laborious task of MHD eigenfunction classification.

Finally, we would like to outline that there are numerous ways of further improving the algorithm. These include incorporating the imaginary part of the eigenfunctions, including additional information for the training (for example the mode number), or employing unsupervised learning.

8. Acknowledgement

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

[1] I. Bernstein, E. A. Frieman, M. D. Kruskal, R. M. Kulsrud, An energy principle for hydromagnetic stability problems, Proceedings of the

Royal Society of London. Series A. Mathematical and Physical Sciences 244 (1236) (1958) 17–40. doi:10.1098/rspa.1958.0023.

[2] C. Z. Cheng, M. S. Chance, Low-n shear Alfvén spectra in axisymmetric toroidal plasmas, Physics of Fluids 29 (11) (1986) 3695–3701. doi:10.1063/1.865801.

[3] W. Kerner, J. Goedbloed, G. Huysmans, S. Poedts, E. Schwarz, CASTOR: Normal-mode analysis of resistive MHD plasmas, Journal of Computational Physics 142 (2) (1998) 271–303. doi:10.1006/jcph.1998.5910.

[4] C. Nührenberg, Global ideal magnetohydrodynamic stability analysis for the configurational space of Wendelstein 7–X, Physics of Plasmas 3 (6) (1996) 2401–2410. doi:10.1063/1.871924.

[5] D. A. Spong, E. D'Azevedo, Y. Todo, Clustered frequency analysis of shear Alfvén modes in stellarators, Physics of Plasmas 17 (2) (2010) 022106. doi:10.1063/1.3313818.

[6] C. Nührenberg, Computational ideal MHD: Alfvén, sound and fast global modes in W7-AS, Plasma Physics and Controlled Fusion 41 (9) (1999) 1055–1070. doi:10.1088/0741-3335/41/9/301.

[7] H. L. Berk, J. Van Dam, Z. Guo, D. M. Lindberg, Continuum damping of low-n toroidicity-induced shear eigenmodes, Phys. Fluids B 4 (7) (1992) 1806–1835.

[8] G. W. Bowden, M. J. Hole, A. Könies, Calculation of continuum damping of Alfvén eigenmodes in tokamak and stellarator equilibria, Physics of Plasmas 22 (9) (2015) 1. doi:10.1063/1.4930209.

[9] G. Y. Fu, J. W. Van Dam, Excitation of the toroidicity induced shear Alfvén eigenmode by fusion alpha particles in an ignited tokamak, Phys. Fluids B 1 (10) (1989) 1949–1952.

[10] Y. I. Kolesnichenko, V. V. Lutsenko, H. Wobig, V. Yakovenko, Alfvén instabilities driven by circulating ions in optimized stellarators and their possible consequences in a Helias reactor, Physics of Plasmas 9 (2) (2002) 517–528. doi:10.1063/1.1432993.

[11] A. Könies, R. Kleiber, A computational approach to continuum damping of Alfvén waves in two and three-dimensional geometry, Physics of Plasmas 19 (12) (2012) 122111. doi:10.1063/1.4769115.

[12] G. W. Bowden, M. J. Hole, A singular finite element technique for calculating continuum damping of Alfvén eigenmodes, Physics of Plasmas 22 (2) (2015) 022116. doi:10.1063/1.4907792.

[13] Z. Wang, W. Yan, T. Oates, Time series classification from scratch with deep neural networks: A strong baseline, CoRR abs/1611.06455 (2016). arXiv:1611.06455.
URL `http://arxiv.org/abs/1611.06455`

[14] F. Chollet, Keras, `https://github.com/fchollet/keras` (2015).

[15] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, `http://www.deeplearningbook.org`.

[16] C. Schwab, Ideal magnetohydrodynamics: Global mode analysis of three-dimensional plasma configurations, Physics of Fluids B: Plasma Physics 5 (9) (1993) 3195–3206.

[17] A. Könies, 10th IAEA TM on Energetic Particles in Magnetic Confinement Systems (Kloster Seeon,Germany) (2007).

[18] F. Wagner, et al., W7-as: One step of the wendelstein stellarator line, Physics of Plasmas 12 (7) (2005) 072509. doi:10.1063/1.1927100.

[19] R. C. Wolf, et al., Performance of wendelstein 7-x stellarator plasmas during the first divertor operation phase, Physics of Plasmas 26 (8) (2019) 082504. doi:10.1063/1.5098761.