PURPOSE-LED
PUBLISHING™

PAPER • OPEN ACCESS

Auto-encoding quadrature components of modulated dispersion interferometers

View the article online for updates and enhancements.

Auto-encoding quadrature components of modulated dispersion interferometers

K J Brunner[*] , **G Fuchert** , **F B L de Amorim Resende**[1] , **J Knauer** ,
M Hirsch , **R C Wolf**[2] **and the W7-X Team**[3]

Max-Planck-Institute for Plasma Physics, Wendelsteinstr. 1, 17489 Greifswald, Germany

E-mail: k.j.brunner@ipp.mpg.de

CrossMark

Abstract

Phase-modulated dispersion interferometry is an increasingly popular method of measuring the line-integrated electron density in magnetic confinement fusion devices due to its inherent vibration compensation in combination with the high resilience to rapid density changes. At present, two quadrature methods of extracting the phase from a temporal interferogram of a phase modulated DI are known, both of which have been shown to exhibit strong phase nonlinearities, which could be prohibitively large in fusion reactor size devices (Brunner *et al* 2022 *Rev. Sci. Instrum.* **93** 023506). These errors are not fixed and hence, i.e. they change over the course of the day, which makes static compensation schemes difficult. In this work we present a neural network based approach to extract the phase information from a temporal dispersion interferogram using a mathematically regulated auto-encoder. No assumptions are being made on the underlying physics/optics of the data avoiding training the network with false assumptions. The training approach also avoids the common concern of extrapolation entirely, i.e. how does the network behave when it sees unseen data. We also show that the network can be trained without using plasma data. The network is shown to extract quadrature components from the interferogram with significantly reduced systematic quadrature distortions as well as lower noise and higher bandwidth than the currently established methods. It is also shown to effectively reduce the nonlinearity drifts occurring, when trained on larger datasets. The method presented here is not limited to dispersion interferometers, but should be applicable to any quadrature component based phase measurement, e.g. radiometry, reflectometry, radar or digital radio.

Keywords: neural-network processing, quadrature detection, plasma diagnostics, dispersion interferometer

1. Introduction

Interferometry is one of the key diagnostics for density feedback control on many magnetically confined fusion experiments today and is planned for future fusion reactor level devices as well [1–3]. Dispersion interferometry (DI) is a special type of self-referencing interferometer gaining increasing popularity in fusion experiments around the world due to its inherent insensitivity to geometric path length differences. For this reason it is adopted in an increasing number of large scale experiments including Wendelstein 7-X (W7-X) and ITER [4, 5]. In these devices a heterodyne detection scheme is employed, given that most plasma experiments well exceed a phase shift of π radian. However, in particular the dispersion interferometer (DI) at W7-X has revealed that they can suffer from significant distortions of the quadrature components, best seen in the constellation diagram [6].

Figure 1 provides an extreme example of such distortions as seen with the integral-electron density dispersion interferometer (IEDDI) at W7-X. The normalized constellation shown at the top was acquired using a wedge scan regularly acquired at W7-X before the plasma phase of an experiment program. During such a scan a motorized wedge moves back and forth in the measurement path thus shifting the measured phase by about 5π [4]. The constellation plots the inphase and quadrature components against each other and is a quick indicator for the quality of the measurement. It should ideally be the unit-circle as indicated in black. The bottom plot depicts the difference between the expected phase and the measured phase calculated from the quadrature components above.

Two constellations are show. The real-time integration based method as used on W7-X for density feedback control is plotted in dark green [4]. The light green one is the phase extraction method based on lock-in amplification (LIA) as originally implemented by Akiyama *et al*, albeit using an implementation using phase locked-loop down-conversion [6, 7]. It is obvious, that there are significant distortions for both methods, although one has to acknowledge that the LIA method cannot be operated in an optimal calibration point, i.e. the ideal phase retardation [6]. However, it also shows distortions beyond a simple retardation mismatch, i.e. a slight shift as well as a rotation in the ellipse. From such a wedge scan the error across the phase space can be calculated assuming the wedge to have a perfectly linear slope and moving at a constant speed [4]. To date, we have not found any indication towards the invalidity of these assumptions. The error found particularly in the integration method are at least 250 mrad, but can be as high as 1 rad (see figure 1 bottom). For comparison, a phase shift of 2π radian corresponds to roughly $7.5 \times 10^{19}\,\mathrm{m}^{-2}$ in the W7-X IEDDI system [4]. Given that the operating point for many plasma experiments at W7-X is of the order of $6 \times 10^{19}\,\mathrm{m}^{-2}$, this is an unacceptably high value and can lead to serious implications for the density feedback at W7-X (see section 4).

In this work we propose to deal with this issue using an artificial neural network (ANN), more specifically an auto-encoder (AE). ANNs have been shown to be universal approximators [8]. They usually learn to produce an output for a given input based on a well known data-set assumed as
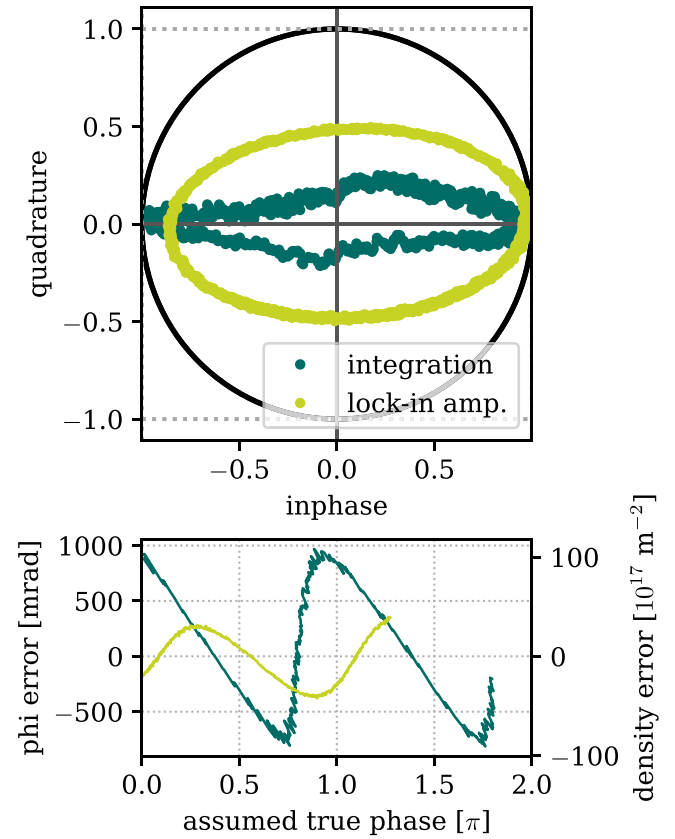
Figure 1. An extreme example for constellation distortions as seen at W7-X during experiment program 20 241 112.62. The dark green is the integration based real-time phase extraction method and the light green the method based on LIA. The constellation at the top is a quick indicator for the error, as both methods should look like the unit circle marked in black. The error on the bottom is calculated for the constellation based on a wedge scan.

ground truth mimicking the underlying mapping. They can be used to do quadrature detection, which has been used by various works in particular in the field of radio communication. The initial application for neural networks in the field of optical interferometry was for the correction of quadrature components after they had been acquired using conventional means [9]. Following this work, various groups used neural networks to correct nonlinearities in interferometers, however the field of heterodyne interferometry has thus far only predicted nonlinearities using ANN to correct a malformed constellation [10].

The problem with the aforementioned methods is, that one has to train the network based on assumptions about the underlying physics, e.g. by training the network on data generated or evaluated from a 'known model'. This approach ignores the fact that our understanding of a real system is often limited, and the analytical description often neglects certain effects. In case of the IEDDI system the distortions of the constellation are a known effect, which however does not have a physics model to describe the effect [6]. Here we propose to deal with this dilemma by using an AE to directly extract quadrature components from a temporal interferogram constraining

the latent space with mathematical properties only. We thus do not make assumptions on the underlying data but let the network extract a *coordinate system* with desired properties. To our knowledge this is the first time this approach has been taken. We apply the neural network to the raw data acquired with the W7-X IEDDI system, which is a phase modulated dispersion interferometer [4].

In section 2 we give a brief introduction to the neural network, as it is generally not well known to the plasma diagnostics audience, and give justification to the application in DI. In section 3 we describe the training approach, which is the key component in achieving robust convergence to a well-performing AE. We will then describe the performance of the ANN using plasma data from W7-X in section 4 and discuss the results in section 5.

2. Background

2.1. Justification

The temporal interferogram of a dispersion interferometer has a well known mathematical description. Given that an ANN will have to operate in the time domain, it will likely have to learn a temporal representation of the inphase (*I*) and quadrature (*Q*) components from a temporal interferogram $s(t)$[4]. In the time domain a temporal interferogram encodes the quadrature components as follows:

$$I: \quad \int_0^{T_m/2} \tilde{s}(t)\mathrm{d}t + \int_{T_m/2}^{T_m} \tilde{s}(t)\mathrm{d}t \propto C_i(\rho)\cos\phi_p \quad (1)$$

$$Q: \quad \int_0^{T_m/2} \tilde{s}(t)\mathrm{d}t - \int_{T_m/2}^{T_m} \tilde{s}(t)\mathrm{d}t \propto C_q(\rho)\sin\phi_p. \quad (2)$$

$\tilde{s}(t)$ in this case is the temporal interferogram, i.e. just the modulated part of the diode signal, without the static background [4]. An example can be seen at the top of figure 2. In the equations, T_m is the phase modulation period, $C_{i,q}$ are Bessel-type functions, which are constant given a constant ρ, which is what we refer to as the modulation depth, i.e. the amplitude of the phase retardation induced by the DI's phase modulator. ϕ_p is the quantity of interest, e.g. the phase shift induced by the plasma under investigation. Equations (1) and (2) show that the quadrature components for a modulated DI can be found by simple integration and summation. Lloyd *et al* have shown that an ANN can be used as universal integrator for a continuous real function $f: \mathbb{R}^n \to \mathbb{R}$, which applies to this problem [11]. It hence stands to reason, that an ANN can learn to extract quadrature components from the diode signal of a modulated dispersion interferometer.

2.2. AEs

AEs are a special type of ANN originally developed by Rumelhart, Hinton and Williams, who were studying pattern recognition in the human brain, with the aim of reducing the noise of an input signal via compression to essential features
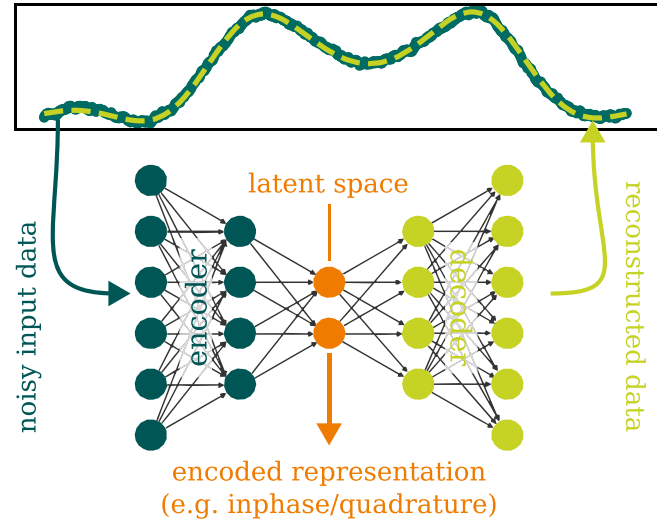
Figure 2. A schematic representation of an auto-encoder ANN. The input to the AE is a modulation period of the temporal interferogram signal (dark green), which the encoder network attempts to reproduce by compressing it to an encoded representation in the latent space (orange). This is then used by a decoder network, which reproduces the input at the output with significantly less noise (light green).

[12]. Hinton and Salakhutdinov have later shown that deep AEs, i.e. AEs with several hidden layers, can be used for dimensionality reduction of highly complex nonlinear data [13]. These are the two properties necessary for the problem posed here.

At its core an AE is a reproductive ANN, i.e. it attempts to reproduce its own input. Figure 2 gives a schematic representation of an AE, which is actually the connection of two mirrored neural networks. The input to the AE is a modulation period of the temporal interferogram signal (dark green at the top). The encoder network uses this information to compress the signal to an encoded representation in the latent space (orange), which is then used by a decoder network to reproduce the input at its output (light green). The resulting signal is usually significantly less noisy and reproduces only essential features.

The necessary dimensionality of the latent space depends on the problem. Finding the right number of latent space dimensions is a somewhat iterative approach using educated guesses. Choosing too many will result in overfitting, while using too few prevents accurate reproduction of the input. In general, the latent space can be seen as a point in a multidimensional coordinate system, which however does not have to have well-defined properties. As long as the output reproduces the input it may in fact be completely arbitrary, for which reason it is ignored in most applications. However, there are applications for the latent space dimensions, e.g. when the exact properties of the underlying coordinate system are irrelevant as long as it results in a surjective representation of the input [14]. For the most part, making use of the latent space requires it to be trained to have certain properties, which will be discussed in section 3.

Looking at the architecture in figure 2 the neural-network architecture used for integration by Lloyd *et al* is applicable to the encoder network of an AE and therefore should be able to integrate the input data. The inverse, i.e. deriving $\tilde{s}(t)$ from an integral value in the latent space, is not necessarily true (at least has not been proven), but given that we are interested in the latent space and use the decoder network as a training tool for the encoder only, a formal proof will not be attempted here.

3. Network architecture & training methodology

The quantity of interest for an interferometer is the phase shift. As such it would be natural to try to extract it directly. However, the phase itself cannot be mathematically constrained sufficiently to yield a well-behaved network. Fortunately it can be calculated from quadrature components -as is done by any heterodyne interferometer- which in the analytical form have very well-behaved properties. These can be used for regularization (see section 3.1).

Quadrature components are naturally two-dimensional, hence the latent space should have at least this many. More dimensions would be necessary to describe deviations from a circular constellation, similar to higher Fourier harmonics. From equations (1) and (2) we also see that the modulation depth ρ also defines our quadrature components. For the initial proof-of-principle presented here we keep the modulation depth ρ static within the training and evaluation domain, thus keeping C_i and C_q static. Seeing that our constellations in figure 1 are non-circular, there are likely further *hidden dimensions* describing our data, which may be responsible for these distortions. As such we let the latent space be 3-dimensional for now giving the AE one *correction dimension* in which to compress all the deviations from the 2-dimensional quadrature model. Note that the exact behavior of this extra dimension is irrelevant as we are only interested in the behavior of the first two. We will therefore not discuss its behavior going forth.

The input to the network are 1000 samples defined by the ratio of phase modulation frequency to sampling frequency in the W7-X IEDDI system. The samples are prescaled and shifted to reside within [0, 1], which improves the numerical stability and convergence of the training process. Without loss of generality we take the samples from the rising edge of the phase-modulator's reference frequency. This gives implicit information about the reference frequency to the network similar to other approaches, thus the necessary information should still be contained within the data. rectifier linear units (ReLUs) were used as activation functions, the primary reason being their compatibility with the field programmable gate array (FPGA) used in the W7-X IEDDI system [4, 15]. The AE is chosen to be an asymmetric encoder, i.e. the encoder network differs in the number of nodes in the hidden layer(s) with respect to the decoder. This yielded better performance to the traditional symmetric encoder approach, which may be the result of information loss in the 'integration' requiring a more complex ANN to reproduce the input. The number of hidden layers and neurons were otherwise optimized during training using keras_tuner [16]. In general, the performance did not differ much between a single hidden layer with 100 to 200 neurons and 2 hidden layers with 50 to 100 neurons. The fact that multi-layer networks require smaller hidden layers for the same quality is commonly observed in ANN applications (see for example Hinton *et al* 2006 [13]). Since the computational cost rises quadratically with the number of layers and linearly with the number of neurons in a layer, a single layer approach would in our case be computationally less expensive and thus preferable. It should be noted however, that other layouts may be more suitable in other environments.

3.1. Regularization

In order to ensure proper behavior of the AE within the latent space we require regularization. This is in essence a loss term in addition to the direct mean squared error (MSE) calculated between the input and output. In order to not make any assumptions about the underlying physics, we use only mathematical properties of the quadrature components:

$$\text{MSE}_\perp = \sum_N \left(x_{i,1}^2 + x_{i,2}^2 - 1 \right)^2 \tag{3}$$

$$\text{MSE}_{\text{origin}} = N \cdot \left(|\text{mean}\,(x_{i,1})| + |\text{mean}\,(x_{i,2})| \right)$$

$$= \left| \sum_N x_{i,1} \right| + \left| \sum_N x_{i,2} \right|. \tag{4}$$

In the two equations, $x_{i,1}$ and $x_{i,2}$ denote the *i*th sample of the first and second latent space component for a training batch of size N. Equation (3) enforces that the quadrature of the first two latent space components reside on the unit circle (as quadrature components should). This MSE method was often used and can effectively reduce nonlinearities. However, it relies on the assumption that the origin of the coordinate resides at $(0, 0)$ to fulfill the Pythagorean theorem, which it does not enforce itself. We hence supplement it with equation (4) enforcing that the components average is 0. This is true for quadrature components given sufficiently large N and random uniform sampling over the entire set of latent space values. Note, that we scale equation (4) by N. This is important for robust convergence, where our interpretation is that the gradients in the loss function hyperplane are too shallow along the relevant dimensions for the optimizer.

We arbitrarily enforced the regularization conditions here on the first two latent space dimensions. This makes it easier to identify them after training, since the regularization condition does not actually enforce, which of the latent space dimensions turns out to be the quadrature component and which the inphase component. This can however be automatically identified after training, knowing that they reside in the first two.

3.2. Training data conditioning

The advantage of training the AE to learn quadrature components is that one can train on the entire possible set of values potentially seen by the interferometer. More explicitly: since
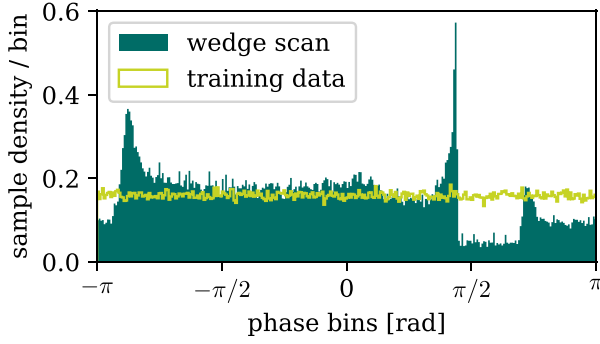
Figure 3. Histogram of the phase samples collected during the wedge scan taken at the start of W7-X experiment program 20 241 112.62 (dark green) and the distribution of phase samples used for training sampled from it (bright green).
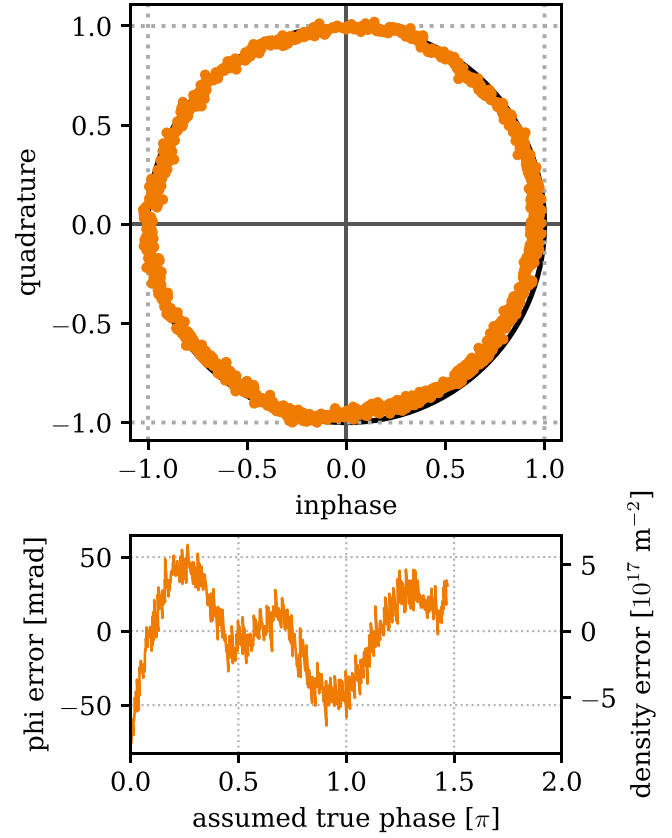

Figure 4. A quadrature constellation extracted from the acquired wedge scan of W7-X program 20 241 112.62 using a trained AE's encoder (top). The phase error estimated from the constellation is shown at the bottom.

quadrature detection measures with a cyclic boundary condition phase shifts within (-π,π) the set of possible inputs (ignoring noise) is limited and can be sampled well. For the interferometer this is simply done by a wedge scan, which the IEDDI system acquires prior to the plasma phase of every experiment program at W7-X [4]. During the scan the temporal interferogram is sampled at 50 MSamples/s by the FPGA digitizer, which streams the raw data to the W7-X archive, from where it can be downloaded for the training. It is also possible to train on plasma data, e.g. by using a density ramp with more than 2π phase shift on the interferometer.

Either approach has the issue that the data is not uniformly distributed within the measurement domain. This is shown in figure 3 in dark green. Depicted is a histogram of the phases measured by the interferometer during a wedge scan at the start of experiment program 20 241 112.62. The phases were calculated using the quadrature corrected LIA method. Note that while there may still be residual errors in the LIA-phases, its values are only used to sample the data. Hence, their quality is less important. The peaks correspond to the null measurements conducted prior to the plasma phase of an experiment program, the asymmetry is the result of W7-X running similar programs. From this distribution we sample such that the resulting distribution is uniform by taking random samples from random bins, as indicated in bright green. This will result in some samples being doubled and other bins being sparsely sampled.

Batch size was chosen to be 3 times the number of phase bins to ensure sufficient sampling for equation (4) to be valid (300 in our case). We use Hyperband as a tuning algorithm for the hidden layers as well as the ReLU slope [17]. Adam was used as an optimizer algorithm [18]. Due to the regularization, convergence happened fairly quickly such that early stopping usually occurred after 40 epochs.

4. Performance

In this section we present the neural network trained using the approach outlined in section 3. For training, 200 000 samples normalized according the method described in section 3.2 were acquired during the wedge scan preceding the plasma

phase of experiment program 20 241 112.62. Given that the plasma program for this experiment had a density ramp spanning more than $7.5 \times 10^{19} \mathrm{m}^{-2}$ we also trained AEs with plasma data from this program. The sampling and training approach are in both cases identical. No significant difference in performance was found between the data sets. This is to be expected as the AE is agnostic as to the source of the phase shift on the interferometer. As such, the wedge scan results will be shown here. The fact that plasma data and wedge data are equivalent from a training perspective is important however, as it means that one can train an individual network using a wedge for any program, even the ones with very low density that would normally not have sufficient data to train.

Figure 4 shows the constellation produced by the resulting neural network. As mentioned previously, the quadrature components composing the constellation on the top right are the first two latent space dimensions. As can be seen the constellation is very smooth with only minor distortions in the upper right and bottom left. These residuals could not be removed even using further latent space dimensions or more complex networks and indicate some intrinsic disturbance of the interferometer data. It is striking that these pointy features are similar to the features seen in the integration method (see figure 1 top, which shows similar distortion features). The source of these must be investigated in the future.
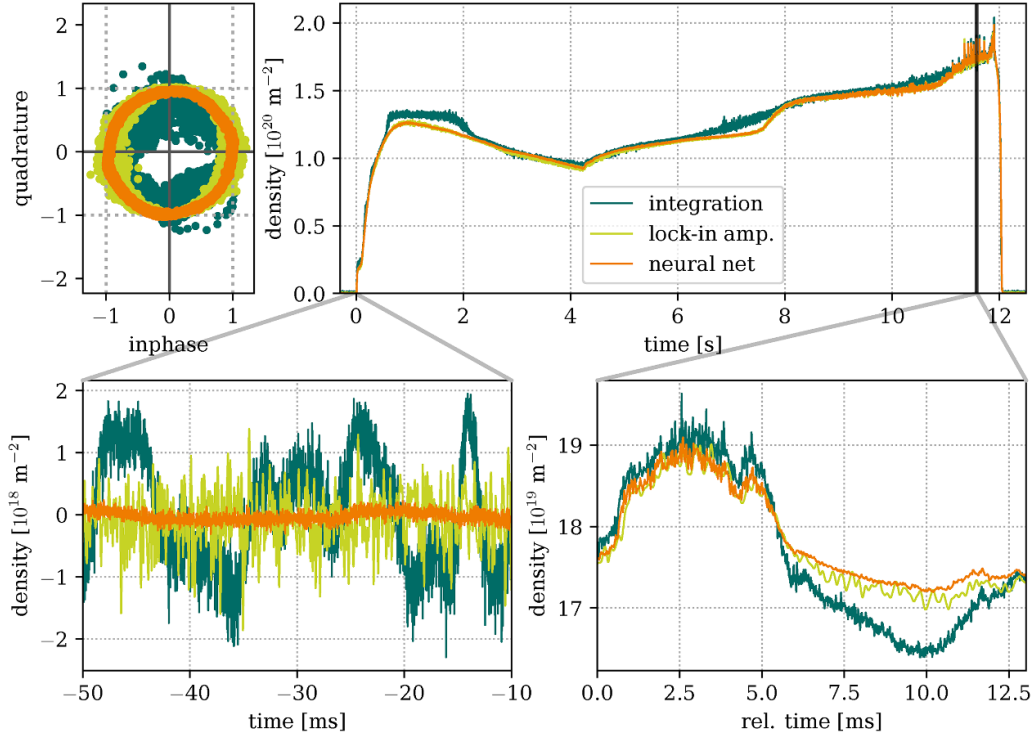
Figure 5. Comparison of different phase extraction methods for the plasma phase of W7-X program 20 241 112.62. The integration method (dark green) and LIA method (bright green) were corrected using the established quadrature correction methods [6]. The constellations are shown on the top left, the full density trace on the top right. The bottom left shows an extract from the pre-plasma phase, whereas the bottom right shows an extract from a highly dynamic plasma phase. The bottom plots use relative times.

The error calculated from the NN-constellation is shown at the bottom of figure 4. Compared to the errors from figure 1 the error is reduced by an order of magnitude and is on par with the errors from the LIA method after quadrature correction, which is currently the most accurate method. To put this into use figure 5 shows a measurement for W7-X program 20 241 112.62. All three evaluation methods are overlaid, with the integration method in dark green, the LIA method in light green and the NN method in orange. The integration and LIA method have been quadrature corrected using the wedge scan to give a fair comparison [6].

The first obvious difference is in the constellations on the top right. While the NN constellation is incredibly smooth, the integration method in particular is very noisy. For the integration method this is often due to the previously identified amplitude dependence of the method [6]. For the LIA method it is partially due to the quadrature correction, which amplifies one of the quadrature components (in this case the inphase component), which indiscriminately also amplifies its noise. We could unfortunately not operate the IEDDI system in a calibrated regime for the LIA method, as it was incompatible with experiment operation.

The bottom left plot in the figure gives a short excerpt from the pre-plasma phase, i.e. before $t = 0$ s in the top right plot. As can be seen the integration method is strongly affected by laser amplitude fluctuations, whereas the other two methods are not. The noise on the NN method appears to be much lower,

although one should keep in mind that the statistical noise of the LIA method is amplified by the quadrature correction. As such, we would for now assume that they are on par for these two methods.

More impressive are the results from the plasma phase. First one will notice that the methods disagree on the plasma density in the top right plot. While the integration method shows a flat-top region at the start and a linear ramp up to the end of the experiment program, the other two methods are significantly more 'wavy'. This is however understood, as the density feedback control at W7-X currently runs on the integration method, which for this day was in a poorly calibrated state. This resulted in strong distortions, which the density controller is unaware of. The corrected LIA method and the neural network method on the other hand agree very well with each other. The plasma ends with a highly dynamic phase, which appears to be the result of a loss of absorption of the electron cyclotron resonance heating after significant cooling of the plasma from excessive fueling. During this phase significant fast mode activity is visible on the interferometer trace, which is shown in the zoom at the bottom right. As can be seen this phase is dominated by phases of elevated density with strong mode activity followed by a form of crash with significantly reduced noise. The integration method shows this most prominently, although we previously established that the absolute rise in density cannot necessarily be trusted for this method if the amplitude also changes during

these phases [6]. However, the fact that there is a mode in the elevated density period followed by a period devoid of them is evident. The LIA method, which we trust most for the absolute density value, cannot resolve this mode at all. The apparent fluctuations on the LIA signal are a result of the phase locked-loop used by the LIA method at W7-X. But as established previously, the LIA method has a lower available bandwidth than the integration method due to the inherent necessity for low-pass filtering. In the implementation used here, the final LIA filter is a 5th order Butterworth filter with a 5 kHz critical frequency. On the other hand, the absolute value of the density is probably more accurate than the integration method's.

The neural network method in orange combines the best of these two methods. One can clearly see the modes in the elevated density period, followed by an almost completely noiseless period shortly after the 'crash' that agrees with the LIA method's values. The NN approach hence delivers exactly what we expected it to. AEs are known to excel at noise reduction, which is one of their primary applications. Given that the neural network, just like the integration method, does not perform any filtering or averaging over several modulation periods its bandwidth is theoretically only limited by the Nyquist theorem.

4.1. Constellation drift mitigation

The NN-approach presented up to this point is also capable of handling constellations drifts, which are quite prominent at W7-X. Figure 6 gives an example of constellations acquired by the IEDDI system for the first and last W7-X experiment program acquired on 21 November 2024. As can be seen, the integration method's constellation appears to change quite significantly over the course of the day, which results in significant errors for the real-time density feedback. The LIA method itself has a similar issue in that the constellation's origin shifts to the left, which modifies the necessary quadrature correction. If the correction cannot be updated on an inter-program time base (which is true on most days at W7-X given the rapid succession of programs), the systematic phase error of the interferometer measurements drift on the same time scale. Note that these two methods are normalized for comparison but have not been quadrature corrected.

The source of the drifts described here is not fully understood, but appears to be the result of a synchronized amplitude modulation, which may be modified by thermal drifts. The effects were investigated in a previous work, but no definite explanation as to their source could be given [6].

The orange constellation in figure 6 is generated by a NN trained on 400 000 random samples across all wedge constellations acquired throughout the day. The data was again sampled according to the normalization described in section 3.2. The NN generated constellation shows neither an elongation nor a shift of the origin over the course of the day. There are some minor distortions still visible that seem to appear at the final experiment program of the day, however they are likely the result of the dataset being too small.
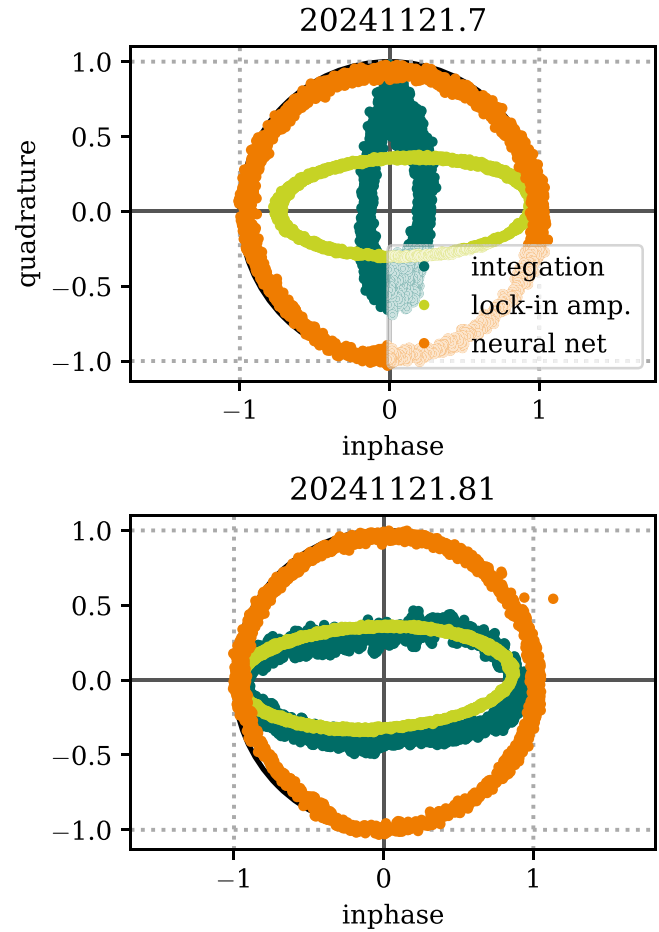
Figure 6. Wedge constellations taken at the start and end of 21 November 2024 for all three phase extraction methods, where the top shows the first plasma program of the day and the bottom the last. Obviously, the constellations drift over the course of the day for the integration method (dark green). For the LIA method (light green) the constellation's origin shifts subtly to the left, modifying the necessary quadrature correction. For the NN approach (orange) no major elongation or shifts are seen.

Preparing the training data set for a NN applicable to all experiment programs throughout a day is currently very time-consuming due to the infrastructure at W7-X and can only be done in post-processing. In consequence training an individual neural network for every program is currently more feasible.

5. Summary & discussion

We have shown that using an AE—a form of ANN—can be trained to deliver quadrature components from temporal interferograms of a modulated dispersion interferometer. Figure 5 shows the capabilities of the method when trying to extract highly dynamic but subtle effects in the plasma, which outperforms the other phase extraction techniques currently employed at Wendelstein 7-X.

The approach we took only relies on regularization using mathematical properties of quadrature components and hence can be applied to any quadrature detector as long as a 2π phase

shift can be induced in the system. Hence, this method may be applicable to diagnostics such as reflectometry, radiometry or even outside the field of fusion diagnostics, e.g. in radar or digital radio.

An important aspect of the training approach presented here is that it can be trained using artificially induced phase shifts and deliver very accurate results in the plasma phase. This is a very important aspect for future interferometer systems in large scale fusion devices, such as ITER, SPARC or DEMO, since they rely on the interferometer delivering very accurate real-time results from the first plasma onwards. At the same, while we used wedge induced phase shifts to train the neural networks presented here, we also saw that we could train our NN using plasma data without a significant difference in the performance. This means that even without the possibility for an induced phase shift the training method could be applied thus giving it even wider applicability. The primary drawback is the necessity for a sufficiently dense plasma to induce at least 2π radian, e.g. a density ramp over $7.5 \times 10^{19} \mathrm{m}^{-2}$ for W7-X.

Real-time application is an additional benefit of this method. While all phase extraction methods can and have been implemented in real-time systems, they vary dramatically in terms of latency or quality of the data they deliver. After training the encoder network, which is the only one used for phase extraction, has a single hidden layer with roughly 150 neurons. This is a fairly mundane size compared to the large scale NN that are becoming common in many places today. The size is so small that one can easily fit it onto an FPGA and calculate the phase with the same latency as the integration method, i.e. a latency of well under 1 μs after collection of the full modulation period.

As shown in section 4.1 a generally applicable NN is feasible and currently only limited by the time for preparing the training data. A solution to this problem might be continuous incremental training, which could also be a good option for systems where there is no ability to artificially induce a phase shift prior to a measurement. This will be something attempted in the future. However, one can already see with the current limited dataset that a general NN would not require constant updates to the quadrature correction parameters throughout the day unlike the other two methods.

In the study presented here, we kept the modulation depth ρ static to limit the number of degrees of freedom to consider for the initial proof-of-principle. It is reasonable to assume that the modulation depth does not change for a resonant mechanical system like a photo-elastic modulator and will certainly be true over the course of a single experiment program as discussed in section 4. However, it is conceivable that some of the long term drifts in the constellation as shown in section 4.1 could be the result of changing ρ. Moreover for electro-optically modulated DIs, like the W7-X multi-channel dispersion interferometer currently under construction, thermal drifts can result in changing modulation depths. The training approach presented in section 4.1 can be employed to scan several wedge scans of varying ρ to incorporate such drifts. This is something that will be attempted in the near future. This would likely require additional latent space dimensions and an increased layer sizes.

As such the NN approach holds much promise to improve the IEDDI system at W7-X and potentially other quadrature detection systems elsewhere.

Data availability statement

The data cannot be made publicly available upon publication because they are not available in a format that is sufficiently accessible or reusable by other researchers. The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

Sources

Much of the evaluation in this article was conducted using Python 3.12 in combination with the NumPy, Matplotlib and pandas libraries [19–22]. Neural networks were trained using Tensorflow and Keras tuner [16, 23].

Large language models were used for literature research as well as coding assistance [24]. All work was thoroughly checked and verified.

Author contributions

K J Brunner ⓘ 0000-0002-0974-0457
Conceptualization (lead), Data curation (lead), Formal analysis (lead), Investigation (lead), Methodology (lead), Project administration (equal), Software (lead), Supervision (lead), Validation (lead), Visualization (lead), Writing – original draft (lead)

G Fuchert ⓘ 0000-0002-6640-2139
Conceptualization (supporting), Methodology (supporting), Writing – original draft (supporting)

F B L de Amorim Resende ⓘ 0000-0001-8988-8939
Conceptualization (supporting), Investigation (supporting), Methodology (supporting), Software (equal), Visualization (supporting)

J Knauer ⓘ 0000-0001-7359-6472
Project administration (lead), Resources (supporting), Supervision (supporting), Visualization (supporting)

M Hirsch ⓘ 0000-0002-7120-6087
Funding acquisition (lead), Project
administration (supporting), Resources (lead),
Supervision (supporting)

R C Wolf ⓘ 0000-0002-2606-5289
Funding acquisition (lead), Project
administration (supporting), Resources (lead)

References

[1] Van Zeeland M A, Akiyama T, Becoulet M and Kim C 2023 ITER toroidal interferometer and polarimeter (TIP) beam refraction in 3D density profiles *Fusion Eng. Des.* **193** 113618

[2] Biel W *et al* 2022 Development of a concept and basis for the demo diagnostic and control system *Fusion Eng. Des.* **179** 113122

[3] Reinke M L *et al* 2024 Overview of the early campaign diagnostics for the sparc tokamak (invited) *Rev. Sci. Instrum.* **95** 103518

[4] Brunner K J, Akiyama T, Hirsch M, Knauer J, Kornejew P, Kursinski B, Laqua H, Meineke J, Trimiño Mora H and Wolf R C 2018 Real-time dispersion interferometry for density feedback in fusion devices *J. Instrum.* **13** 0900

[5] Akiyama T, Sirinelli A, Watts C, Shigin P, Vayakis G and Walsh M 2016 Design of a dispersion interferometer combined with a polarimeter to increase the electron density measurement reliability on ITER *Rev. Sci. Instrum.* **87** 11E133

[6] Brunner K J, Knauer J, Meineke J, Castillo H I C, Hirsch M, Kursinski B, Stern M and Wolf R C (The W7-X Team) 2022 Sources for constellation errors in modulated dispersion interferometers *Rev. Sci. Instrum.* **93** 023506

[7] Akiyama T, Yasuhara R, Kawahata K, Okajima S and Nakayama K 2014 Dispersion interferometer using modulation amplitudes on LHD (invited) *Rev. Sci. Instrum.* **85** 11D301

[8] Hornik K, Stinchcombe M and White H 1989 Multilayer feedforward networks are universal approximators *Neural Netw.* **2** 359–66

[9] Zhi Li, Herrmann K and Pohlenz F 2003 A neural network approach to correcting nonlinearity in optical interferometers *Meas. Sci. Technol.* **14** 376

[10] Olyaee S, Ebrahimpour R and Esfandeh S 2014 Modeling of measurement error in refractive index determination of fuel cell using neural network and genetic algorithm *Hydrogen, Fuel Cell Energy Storage* **1** 95–104

[11] Lloyd S, Irani R A and Ahmadi M 2020 Using neural networks for fast numerical integration and optimization *IEEE Access* **8** 84519–31

[12] Rumelhart D E, Hinton G E and Williams R J 1986 Learning representations by back-propagating errors *Nature* **323** 533–6

[13] Hinton G E and Salakhutdinov R R 2006 Reducing the dimensionality of data with neural networks *Science* **313** 504–7

[14] Fuchert G *et al* (W7-X Team) 2024 Calibration techniques for thomson scattering diagnostics on large fusion experiments *Rev. Sci. Instrum.* **95** 083533

[15] Glorot X, Bordes A and Bengio Y 2011 Deep sparse rectifier neural networks *Proc. 14th Int. Conf. on Artificial Intelligence and Statistics* (*Proc. Machine Learning Research) (Fort Lauderdale, FL, USA, 11–13 April 2011)*, vol 15, ed G Gordon, D Dunson and M Dudík (PMLR) pp 315–23

[16] O'Malley T *et al* 2019 Kerastuner (available at: https://github.com/keras-team/keras-tuner)

[17] Lisha Li, Jamieson K, DeSalvo G, Rostamizadeh A and Talwalkar A 2018 Hyperband: a novel bandit-based approach to hyperparameter optimization *J. Mach. Learn. Res.* **18** 185 (available at: https://jmlr.org/papers/v18/16-558.html)

[18] Kingma D P and Jimmy B 2014 Adam: a method for stochastic optimization *CoRR* (arXiv:1412.6980)

[19] Python Software Foundation 2001 Python (available at: www.python.org)

[20] McKinney W 2010 Data structures for statistical computing in python *Proc. Python in Science Conf.* ed S van der Walt and J Millman (SciPy) pp 51–56

[21] Hunter J D 2007 Matplotlib: a 2d graphics environment *Comput. Sci. Eng.* **9** 90–95

[22] van der Walt S, Colbert S C and Varoquaux G 2011 The numpy array: a structure for efficient numerical computation *Comput. Sci. Eng.* **13** 22–30

[23] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems (available at: www.Software tensorflow.org)

[24] Doosthosseini A, Decker J, Nolte H and Kunkel J M 2024 Chat AI: a seamless slurm-native solution for hpc-based services