

Wavefront prediction using artificial neural networks for open-loop adaptive optics

Xuwen Liu¹,¹★ Tim Morris,¹ Chris Saunter,¹ Francisco Javier de Cos Juez,² Carlos González-Gutiérrez² and Lisa Bardou¹

¹Centre for Advanced Instrumentation, Department of Physics, Durham University, South Road, Durham DH1 3LE, UK

²University Institute of Space Sciences and Technologies of Asturias, University of Oviedo, E-33004 Oviedo, Spain

Accepted 2020 May 29. Received 2020 May 29; in original form 2020 January 14

ABSTRACT

Latency in the control loop of adaptive optics (AO) systems can severely limit performance. Under the frozen flow hypothesis linear predictive control techniques can overcome this; however, identification and tracking of relevant turbulent parameters (such as wind speeds) is required for such parametric techniques. This can complicate practical implementations and introduce stability issues when encountering variable conditions. Here, we present a non-linear wavefront predictor using a long short-term memory (LSTM) artificial neural network (ANN) that assumes no prior knowledge of the atmosphere and thus requires no user input. The ANN is designed to predict the open-loop wavefront slope measurements of a Shack–Hartmann wavefront sensor (SH-WFS) one frame in advance to compensate for a single-frame delay in a simulated 7×7 single-conjugate adaptive optics system operating at 150 Hz. We describe how the training regime of the LSTM ANN affects prediction performance and show how the performance of the predictor varies under various guide star magnitudes. We show that the prediction remains stable when both wind speed and direction are varying. We then extend our approach to a more realistic two-frame latency system. AO system performance when using the LSTM predictor is enhanced for all simulated conditions with prediction errors within 19.9–40.0 nm RMS of a latency-free system operating under the same conditions compared to a bandwidth error of 78.3 ± 4.4 nm RMS.

Key words: atmospheric effects – instrumentation: adaptive optics – methods: numerical.

1 INTRODUCTION

In adaptive optics (AO) systems, time lag between wavefront detection and correction induces the bandwidth error. For extreme adaptive optics (XAO) systems for high contrast imaging (HCI) of exoplanets, the bandwidth error results in broadening of the point spread function (PSF) along dominant wind directions, which severely degrades contrast, especially at small star separations (Kasper 2012; Males & Guyon 2018). For wide-field AO systems dominated by tomographic errors, to keep bandwidth error tolerable, the integration time of wavefront sensing and thus guide star magnitude (either natural or laser) is limited, which then limits the sky coverage (Correia et al. 2014; Jackson et al. 2015). One way to overcome this problem is to predict the future wavefront from recent past wavefront measurements. Under the frozen flow hypothesis (Wang, Schöck & Chanan 2008; Poyneer, van Dam & Véran 2009), the turbulence volume is modelled as a linear composition of static, independent layers, each translating across the telescope aperture with certain velocity as a result of dominant wind at that layer.

Because of this spatial and temporal correlation, it is possible that the future wavefronts can be partially predicted using past measurements. This hypothesis is a reasonable simplification of the turbulence for wavefront prediction purposes.

Predictive control in AO is an active research area that incorporates wavefront prediction based on the frozen flow hypothesis into controller design. One of the most popular schemes is the Kalman filter based linear quadratic Gaussian (LQG) control (Paschall & Anderson 1993; Le Roux et al. 2004). Under this framework, the whole system (both turbulence and AO system) is represented by a small set of state variables. Linear models are used to describe temporal evolution of those variables as well as their links with system measurements. Priors from system telemetry and noise statistics are then combined to obtain the control law. Because of its flexibility in structure, LQG predictors allow for additional consideration of other system error sources such as static error and vibration. Numerical and laboratory implementations focusing on a single or a few Zernike modes show great improvement in terms of overall residual phase error or Strehl ratio (Le Roux et al. 2004; Kulcsár et al. 2012) and especially vibration filtering (Petit et al. 2006, 2008). Poyneer, Macintosh & Véran (2007) developed a computationally efficient Fourier based LQG predictive controller,

* E-mail: xuwen.liu@durham.ac.uk

which can be extended to non-integer loop delays (Poyneer & Véran 2008), facilitating graceful formulation of wind-blown turbulence evolution under Fourier basis. Laboratory tests demonstrate a reduction of around 67 per cent in bandwidth error using a full Fourier LQG controller (Rudy et al. 2015). Correia et al. (2014) incorporate open-loop wavefront prediction into a minimum mean square error (MMSE) tomographic reconstructor design for multi-object AO systems. This tomographic predictor allows for use of one-magnitude fainter guide stars (GSs) (corresponding to an increase in the density of available stars by a factor of 1.8) in end-to-end simulations of RAVEN (Andersen et al. 2012), which is expected to be further improved if deployed within a LQG framework. LQG based predictive control has been deployed for AO systems on HCI instrument SPHERE (Petit et al. 2014) for both turbulence correction and vibration filtering in tip-tilt modes. Stability and robustness of LQG controller in full-mode single-conjugate adaptive optics (SCAO) control has also been verified on sky (Sivo et al. 2014), showing overall performance improvement over a standard integrator controller in conditions where bandwidth error is not dominant.

The recently proposed empirical orthogonal functions framework (Guyon & Males 2017) for predictive control aims at fully exploiting linear spatio-temporal correlations within input telemetry and improving controller robustness by assuming no physical model of turbulence evolution. Numerical HCI simulations demonstrate significantly improved contrast and robustness against sensor noise. Although this feature can significantly simplify practical implementation, frequent re-learning and update is unavoidable, for such data-driven predictor and above-mentioned LQG approach, to adapt to varying turbulence conditions.

In this paper, we exploit the potential of artificial neural networks (ANNs) as a non-linear framework for wavefront prediction. Early numerical simulations adopting a feed-forward multilayer perceptron (MLP) network demonstrate promise for using this non-linear tool for slope prediction based on a time series of past noisy measurements by a Shack–Hartmann wavefront sensor (SH-WFS; Jorgenson & Aitken 1992, 1994), with further improvement over a linear predictor when signal-to-noise ratio (SNR) of wavefront sensing gets lower (Lloyd-Hart & McGuire 1996). The last few decades have seen significant advances in both the theory and applications of ANNs (LeCun, Bengio & Hinton 2015), among which the long short-term memory (LSTM) network is well-suited to time series modelling and prediction by design (Hochreiter & Schmidhuber 1997; Gers, Schmidhuber & Cummins 1999).

2 ARTIFICIAL NEURAL NETWORKS

ANNs are computational models inspired by biological neural networks. They are composed of a series of computing elements called neurons that are interconnected in a layered structure. Each neuron receives inputs, either as an input to the entire network or as outputs of connected neurons in the former layer, then transmits mathematically processed input information to connected neurons in the next layer. This forward transmission continues until the final output neurons are reached. Information flow in each type of ANNs is thus specified. A thorough tutorial on ANNs can be found in Goodfellow, Bengio & Courville (2016). A detailed description of a MLP network and its successful application for tomographic wavefront reconstruction can be found in Osborn et al. (2012).

LSTM is an advanced architecture of recurrent neural networks (RNNs) that are specially designed for processing sequential data (Graves 2012). Compared with MLPs with forward trans-

missions only, RNNs have dynamic feedback connections and shared parameters across all time-steps. An internal state vector is transferred through time to maintain memories. LSTMs can especially cope well with long-term dependencies (Goodfellow et al. 2016), which otherwise renders training in normal RNNs much more difficult. It has been successfully applied in fields such as speech recognition (Graves, Mohamed & Hinton 2013), machine translation (Sutskever, Vinyals & Le 2014), and image captioning (Karpathy & Li 2014). LSTMs have two desirable features for wavefront prediction:

(i) **No user input.** No prior knowledge of the atmosphere is assumed for the training process. No user input is required either during application.

(ii) **No user tuning.** The fluid nature of the memory elements within allows the network to learn temporal behaviours of turbulence of varied time constants and to adapt to changes in these without user tuning. The non-linearity of LSTMs enables the agility and robustness when dealing with non-frozen flow turbulence evolution (such as fluctuations in wind velocities), WFS noise or change of turbulence strength.

3 METHODOLOGY

We exploit the potential of ANNs for wavefront prediction in numerical simulations based on a SCAO system. More specifically, the ANN predictor is trained in simulation to predict uncorrected wavefront slopes at the next time-step based on a time series of past noisy slope measurements by a SH-WFS operating in open loop. The simulated SCAO system serves two purposes. The wavefront sensing subsystem is used to generate a series of time sequences of wavefront slopes as training data, with the last frame of slopes in each sequence being the training target. After training, the predictor is incorporated into the AO correction loop for evaluation.

To quantify the efficacy of the ANN predictor, we compare AO corrections in terms of root-mean-square wavefront errors (RMS WFE) under three operating conditions, depending on which WFS measurement is applied to DM at time-step t :

(i) Zero-delay or delay-compensated loop, where the current measurement s_t is used immediately.

(ii) One-frame delay loop, where the prior measurement s_{t-1} is used.

(iii) ANN predictive loop, where the predicted current measurement \tilde{s}_t from $(s_1, s_2, \dots, s_{t-1})$ is used.

3.1 SCAO simulation

The AO simulation tool used is Soapy (Simulation ‘Optique Adaptive’ with Python) (Reeves 2016). Soapy is highly modular, enabling both end-to-end simulations and fast experimentation using a subset of its modules. New modules can also be easily integrated. The architecture of the simulated SCAO system is shown in Fig. 1. Throughout the simulation we use a point source at infinity to act as a natural GS. To generate the training data, one single turbulence layer is assumed. Here, the use of a single layer is for the ease of training. We will show that an ANN predictor trained using one turbulence layer is capable of predicting in multilayer conditions. A large random phase screen with Von Karman statistics is generated within the atmosphere module `Atmos` at the start of each loop run. Pure frozen flow is assumed, under which the large phase screen is translated over the telescope aperture with a given velocity due to the wind. At each time-step, a smaller portion of

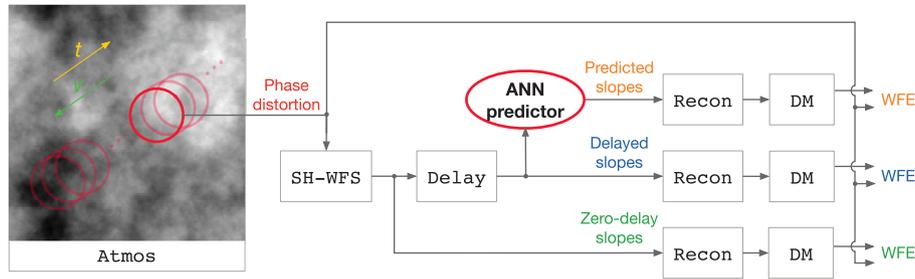


Figure 1. Composition of the simulated SCAO system and its data flow. RMS wavefront error of the predictive correction (upper) is expected to be between the delayed (centre) and delay-compensated (lower) corrections.

the large phase screen, the part of which is seen by the telescope aperture, is output to SH-WFS. SH-WFS then outputs measured noisy wavefront slopes from the image plane using thresholding centre of gravity (TCoG). The thresholding value is a flux cutoff described by a factor of the maximum intensity within a subaperture to suppress photon noise and readout noise. A single frame delay can be used in Soapy simulations (the centre loop in Fig. 1) to account for the inevitable WFS integration time. This time lag between wavefront measurement and correction can be compensated either by applying slope measurements immediately (the lower loop) or by sending the prior slopes to a ANN predictor to extrapolate the current measurements (the upper loop). A reconstructor module (Recon) combines noisy slopes (either delayed, predictive, or delay-compensated) and control matrix generated during calibration to output DM commands, which are used by DM to generate the corrected phase. RMS error between the phase distortion and DM shape is then output as RMS WFE.

Principal simulation parameters are listed in Table 1. The configuration is adopted from CANARY low-order SCAO mode (Morris et al. 2010). We train the predictor under similar atmospheric and system conditions where it will be applied. The impact of the WFS SNR on ANN training and the predictor’s robustness against changes in input statistics will be explored in Section 4.

3.2 ANN training

3.2.1 Training data generation

The wavefront sensing subsystem consisting of Atmos and SH-WFS modules is used to generate the first 100 000 training samples. Each sample is a time sequence of thirty 72-element vectors (s_1, s_2, \dots, s_{30}), with each vector, s_i , being the X and Y slope for each of the 36 subapertures. s_1, s_2, \dots, s_{29} will be ANN inputs sequentially during training, and s_{30} will be the targeted output. Wind velocity corresponding to each sample is a random vector, with its magnitude uniformly sampled from the range $10\text{--}15\text{ m s}^{-1}$ and its direction uniformly sampled from the range $0\text{--}360^\circ$. Wind velocity is constant within each sequence.

We then reverse each sequence to form the other half of the training set, with the last frame being the first and first being last. This corresponds to reversing the wind direction. We use this data augmentation approach to introduce variability in training data to improve model robustness. Resulting training input set and target set are tensors of shape $(2 \times 10^5, 29, 72)$ and $(2 \times 10^5, 72)$ respectively. The amount of training data is decided by trial and error to match both ANN architecture complexity and problem complexity to balance between training data fitting and model generalization. No further training data pre-processing is implemented.

Table 1. Principal parameters used with the Soapy SCAO simulation for ANN training and optimization.

Module	Parameter	Value
System	Frequency	150 Hz
	Pure delay	0.0067 s
	Throughput	1
	Gain	1
Atmosphere	# of phase screens	1
	Wind speed	$10\text{--}15\text{ m s}^{-1}$
	Wind direction	$0\text{--}360\text{ deg}$
	r_0 @ 500 nm	0.16 m
Telescope	L_0	25 m
	Diameter	4.2 m
	Central obscuration	1.2 m
SH-WFS	GS magnitude	10
	# of subapertures	7×7 (36 active)
	Readout noise	1 e^- RMS
	Photon noise	True
	Wavelength	600 nm
Piezo DM	Thresholding value	0.1
	# of actuators	8×8

3.2.2 ANN training and optimization

We use Keras (Chollet et al. 2015) library written in PYTHON for ANN training. The ANN architecture consists of stacked LSTM cells and a final fully connected (FC) output layer. The depth of neural networks is associated with the depth of representations that can be learnt (Goodfellow et al. 2016), thus the stacking of LSTM cells in our case.

The ANN topology comprising two LSTM cells and a FC layer is shown in Fig. 2. The display is unrolled in time, which means all components in the same colour (or row) are duplicates in time and essentially identical to inputs at any time-step. At each time-step t ($t \geq 2$), the network can output a slope prediction \bar{s}_t based on the current input s_{t-1} and two state vectors, the cell state (also called the internal state) and the cell output (also called the hidden state). Both states are either initialized as all-zero vectors ($t = 2$) or updated at each time-step ($t > 2$) using information in the input sequence so far.

Parameters (also called trainable weights) of the network determine how inputs are processed mathematically layer by layer. ANN training is the process to optimize these parameters iteratively to minimize a training error. These parameters are initialized using a Gaussian distribution. 10 per cent dropout is deployed for each LSTM cell (Gal & Ghahramani 2016). 10 per cent training samples form a validation set. The remaining 90 per cent samples are randomly split into batches of size 128 before each epoch. The training

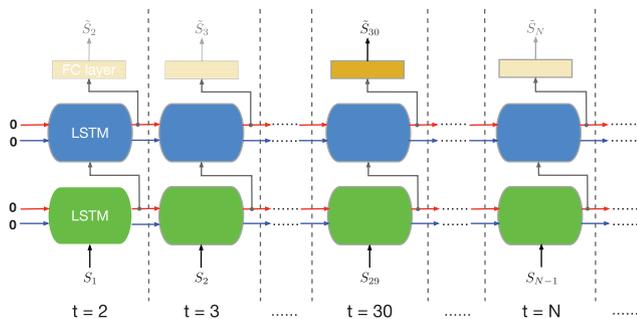


Figure 2. The ANN predictor structure unrolled in time. The predictor can start predicting from the second time-step, although initial predictions can be unstable and inaccurate due to limited temporal information. The two LSTM cells have the same inner structure, but different sets of parameters after training.

Table 2. Breakdown of computational load within the optimized ANN architecture.

Module	Input vector size	Output vector size	FLOPs
First LSTM	72	247	630 344
Second LSTM	247	226	855 184
FC	226	72	32 544
Total			1518 072

error is mean squared error (MSE) between the targeted output s_{30} and the actual output \hat{s}_{30} evaluated and averaged on the current batch. The Adam optimization algorithm is used to optimize the network parameters in a direction that minimizes the training error (Kingma & Ba 2014). It is a first-order gradient descent algorithm and features adaptive learning rate. During one epoch, every batch is evaluated once and the network parameters are updated accordingly multiple times. At the end of each epoch, the updated network is evaluated on the validation set. The initial learning rate is $1e-3$. If MSE of the validation set shows no improvement for consecutive 10 epochs, the learning rate is reduced to its 1/5 unless reaching $1e-5$. The reduced learning rate allows only small updates of the network parameters to prevent this optimization process from early stagnation. Training is terminated after 40 epochs, at which point both training and validation errors have stagnated.

The ANN optimization process, also called hyperparameter tuning, is coupled with ANN training. Hyperparameters determine either the structure of the network or the training process. These are fixed before training starts. We tune two hyperparameters that determine the physical capacity of the network: number of stacked LSTM cells (1 or 2) and length of output vectors of each LSTM cell (a random integer between 100 and 250, different for each cell). Every time a set of these two hyperparameters are chosen, the model is recompiled, re-initialized, and re-trained as is described above. The model that achieves the lowest validation MSE at the end of the 40th epoch is composed of two LSTM cells and a final FC layer (as is shown in Fig. 2). The output vector of the first LSTM cell has 247 elements and the second cell has 226 element. The resulting model has 761 000 trainable parameters in total. Breakdown of the number of floating point operations (FLOP) of the optimized ANN structure is shown in Table 2. The resulting computational load is 2.3×10^8 FLOPs (FLOP per second) for the CANARY-scale 7×7 subaperture system operating at 150 Hz.

4 RESULTS

After training, the optimized predictor is inserted between SH-WFS and ReCon to form part of a predictive correction loop. From this stage, the parameters within the network are fixed and inputs are now processed in a deterministic way. We test the predictor’s generalization and extrapolation capabilities in five different scenarios:

- (i) The predictor is tested on unseen data generated within the parameter boundaries used for the training regime.
- (ii) GS magnitude is increased from 10 (on which the predictor is trained) to 6, which increases the SNR of input slopes. In this scenario, we also investigate the SNR of training data on the predictor’s performance.
- (iii) A time-variant turbulence is considered by changing either the wind speed or the direction every 10 frames (15 Hz) after the predictor stabilizes.
- (iv) A multilayer turbulence is considered to test the predictor’s ability to track multiple wind vectors.
- (v) We extend our approach to account for a more realistic two-frame latency, where we trained a separate ANN predictor to predict two frames in advance directly, and compare that with applying the single-latency predictor twice.

In most scenarios, statistics of the input slopes to the predictor are different to what was used during training. In each scenario, we use 1000 test slope sequences each of 100 frames (0.67 s). We have found that our predictor will remain stable during a 2-min period if both system and turbulence parameters remain unchanged, which is the case in most of the scenarios. Thus, in this paper we only include results obtained from 100-frame sequences to strengthen different aspects of the performance. The predictor’s memory (both internal and hidden states) is zeroed before a new slope sequence. The predictor is expected to build up its memory and output stable predictions in 30 frames as the training is designed. Other simulation parameters are mostly the same as listed in Table 1, unless stated otherwise.

4.1 Performance within the ANN training regime

Fig. 3 shows RMS WFEs averaged over 1000 test atmospheric turbulence sequences. Shaded areas indicate the standard error of the mean RMS WFE (Hughes & Hase 2010). The atmospheric statistics of this test set lie within the bounds of the training regime, though the test set did not form part of the training data set and had not been observed by the network before. Wind speed is 15 m s^{-1} in a single direction. All other simulation parameters are the same as during training, thus in this case the predictor is expected to reach its optimal performance. The predictor output stabilizes after approximately 12 frames. The prediction is stable after this time span as the input statistics remain unchanged afterwards. This implies using shorter sequences for training and thus an alternative network that converges faster is possible. Mean RMS WFEs of the delayed, predictive, and delay-compensated correction loop (averaged after the 12th frame and across all sequences) are 253.9, 244.3, and 243.4 nm, respectively, showing an overall performance improvement brought by the predictor.

4.2 Performance with varying WFS SNR

In Fig. 4, we show the results from three ANNs when observing a bright GS of magnitude 6. In addition to the ANN used in Section 4.1

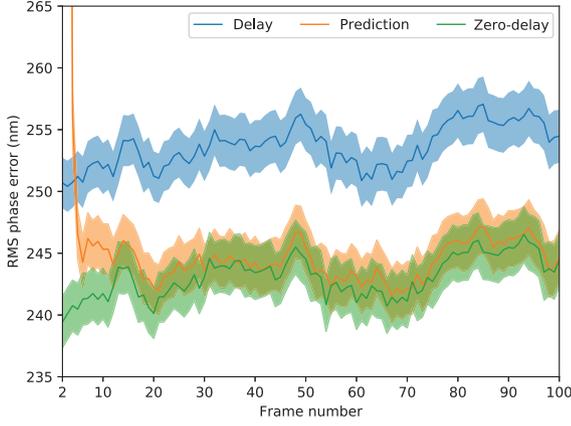


Figure 3. Mean RMS WFEs in an AO loop averaged across 1000 test sequences. The GS used to generate test slopes has a magnitude of 10, which is the same as that for training. The predictor is tested within the training regime, though this test set had not been observed by the predictor before. Wind speed is 15 m s^{-1} in a single direction.

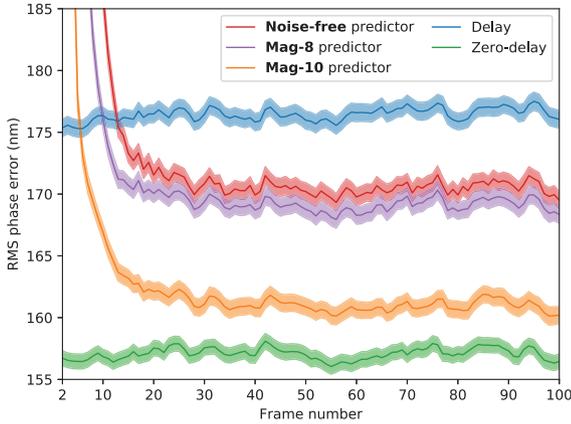


Figure 4. Mean RMS WFEs in an AO loop averaged across 1000 test sequences. The GS used to generate test slopes has a magnitude of 6, which increases the SNR of inputs to the predictor that is trained with a GS of magnitude 10 (**Mag-10**) compared with during its training. Wind speed is 15 m s^{-1} in a single direction. We also compare **Mag-10** predictor using the same set of input slopes with another two predictors that are trained with GS magnitude 8 (**Mag-8**) and trained without WFS noise (**Noise free**).

that was trained on a GS of magnitude 10, we include results from two networks have been trained at different signal to-noise levels. These three predictors are denoted as **Mag-10**, **Mag-8** (trained with a GS of magnitude 8), and **Noise-free** (trained without WFS noise), respectively. The training procedure and other simulation parameters were the same as detailed in Section 3.2, except the thresholding value that was reduced to 0.02 for the **Mag-8** predictor, and 0 for the **Noise-free** predictor. The resulting ANN architectures and computational loads are listed in Table 3. For each network, we see that prediction error decreases until the 20th frame, after which the performance of each ANN stabilizes. However, we note that the ANN trained with the lowest SNR performs far better than the ANNs trained in higher SNR regime and this behaviour was observed irrespective of GS magnitude.

In Fig. 5, we show the RMS slope error (mas) per subaperture compared with zero-delay measurements as the WFS SNR changes. This quantity is the root of the ANN training metric. For each GS magnitude, we generated 1000 slope sequences each of 100

Table 3. Training conditions and structures of the three ANN predictors.

ANN predictor	Mag-10	Mag-8	Noise free
GS magnitude during training	10	8	–
WFS SNR during training	17.6	52.5	∞
# of neurons of the first LSTM	247	247	162
# of neurons of the second LSTM	226	203	114
FLOPS @ 150 Hz frame rate ($\times 10^8$)	2.3	2.1	0.9

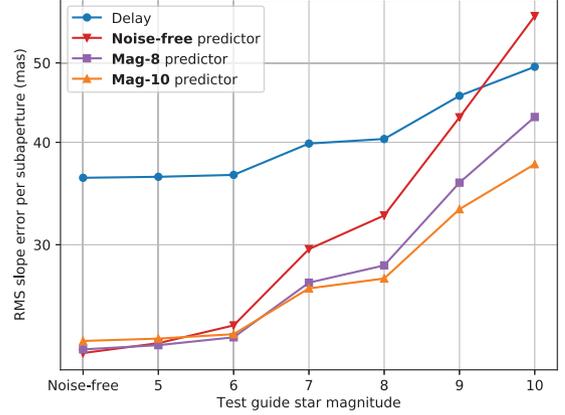


Figure 5. RMS slope error (mas) per subaperture compared with zero-delay measurements by SH-WFS as the WFS SNR varies. This quantity is the root of the ANN training metric. All predictors have lower errors around the corresponding training regimes.

frames. Values shown in Fig. 5 give the mean slope error across all subapertures after the 30th frame (by when all predictors have stabilized under all SNR conditions) in all sequences. All predictors have lower errors around the corresponding training regimes compared with slopes with one-frame delay, which shows the prediction power of ANN predictors of such type. In lower SNR regime (GS magnitude > 6), **Mag-10** predictor achieves lowest slope errors. However, at smaller GS magnitudes (GS magnitude ≤ 6) or in the noise-free condition, the performance of the **Mag-8** predictor is closer to that of the **Mag-10** predictor, rather than the **Noise-free** predictor. This is inconsistent with Fig. 4.

To assist understanding of this discrepancy in the brighter regime, we compare slope errors with reference to the slope measurements using an idealized WFS where centroiding and aliasing errors have been minimized. This WFS is defined as follows. The phase screen seen by the telescope aperture at each time step is first decomposed into its low- and high-order components

$$\phi = \phi_l + \phi_h = \sum_{i=2}^{36} a_i z_i + \phi_h, \quad (1)$$

where $a_i = z_i^T \phi$, z_i is the i th Zernike term in Noll's notation (Noll 1976). The number of Zernikes is chosen to match the structure of the subapertures. Measurement of ϕ by the ideal WFS is computed as

$$s^i \equiv \mathbf{D} \mathbf{a}, \quad (2)$$

where \mathbf{D} is a perfectly calibrated interaction matrix and \mathbf{a} is the Zernike vector (a_2, a_3, \dots, a_{36}) given in equation (1). Within \mathbf{D} , the slope measurements of each Zernike term are calculated directly from the corresponding high-resolution phase grid instead of from the WFS image plane.

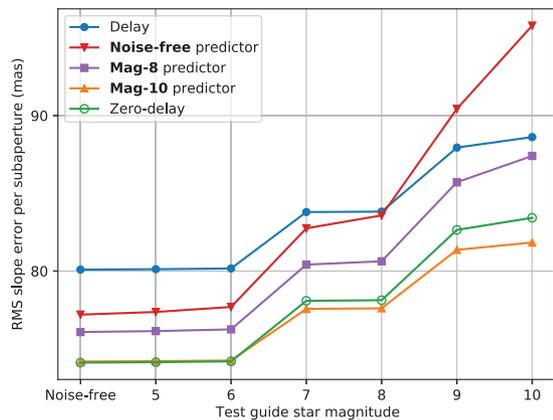


Figure 6. RMS slope error (mas) per subaperture with reference to measurements by the idealized WFS, which removes noise, aliasing, and centroiding errors in the measurement of the first 36 Zernike orders compared with zero-delay measurements by SH-WFS. This along with Figs 4 and 5 demonstrates the filtering of aliasing and centroiding errors apart from the reduction in bandwidth error brought by **Mag-10** predictor.

Fig. 6 shows the RMS slope error with reference to the ideal WFS measurements in each correction loop. Errors in zero-delay measurements are non-zero due to aliasing, centroiding, and noise errors compared to the ideal WFS. Among the three predictors, **Mag-10** predictor achieves significantly lower slope errors under all SNR conditions, which is now consistent with Fig. 4. In addition we see that in low-SNR regimes, **Mag-10** predictor has even lower errors than zero-delay measurements. This implies that the reduction in WFE brought by **Mag-10** predictor also accounts for some aliasing and/or centroiding errors in addition to the reduction in bandwidth error. We think that this is due to being exposed to much lower SNR training data where the temporal correlations within the data are less obvious and noise terms must be learnt to be ignored.

Prediction error σ_{pred} is defined as the RMSE between WFEs in the predictive loop and in the zero-delay loop

$$\sigma_{\text{pred}} = \sqrt{\overline{\text{WFE}_{\text{pred}}^2} - \overline{\text{WFE}_{\text{zero-delay}}^2}}, \quad (3)$$

where $\overline{\text{WFE}_*}$ is the average after the 30th frame and across all sequences. Bandwidth error is defined in a similar fashion

$$\sigma_{\text{BW}} = \sqrt{\overline{\text{WFE}_{\text{delay}}^2} - \overline{\text{WFE}_{\text{zero-delay}}^2}}. \quad (4)$$

σ_{pred} of **Mag-10** predictor ranges from 40.0 to 19.9 nm, decreasing as the WFS SNR is lowered due to the increasing filtering of aliasing and/or centroiding errors. The mean value of bandwidth error across all SNR conditions is 78.3 nm, with a standard deviation of 4.4 nm. This quantity also decreases slightly as GS gets fainter, due to the increasing correlation between bandwidth error and noise error.

In the following three scenarios, we show the results obtained with our optimal **Mag-10** predictor only. We also use a brighter GS of magnitude 6 to reduce the performance variations brought by wavefront sensor noise.

4.3 Performance with time-variant wind velocity

In the above scenarios, we have assumed stationary turbulence. In this section, we demonstrate the agility and robustness of our ANN predictor against fluctuations in wind velocity.

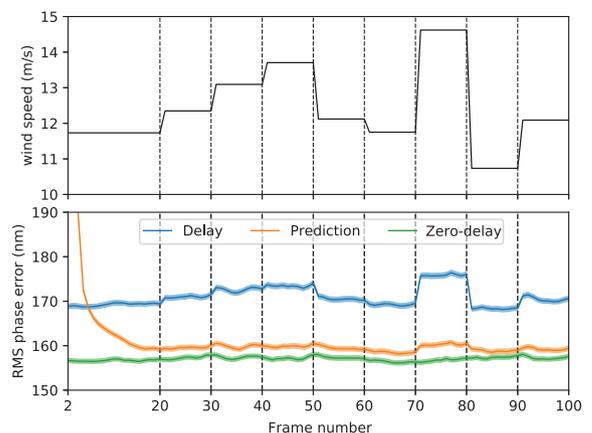


Figure 7. Robustness of the predictor against wind speed fluctuations between 10 and 15 m s^{-1} every 10 frames. Wind direction is 0 deg. GS magnitude is 6.

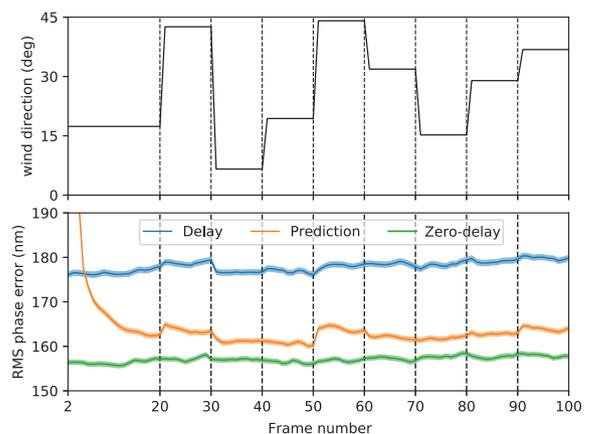


Figure 8. Robustness of the predictor against wind direction fluctuations between 0 and 45 deg every 10 frames. Wind speed is 15 m s^{-1} . GS magnitude is 6.

Here, we use a synthetic wind speed sequence (upper panel in Fig. 7) in a relatively short time-scale of 100 consecutive WFS frames (0.67 s). Wind speed changes every 10 frames (15 Hz) within 10 and 15 m s^{-1} after the first 20 frames during which time the predictor stabilizes. This fluctuation is reflected in the dynamics of the delayed correction, as a faster translation of the phase screen induces increased phase variations between adjacent frames under frozen flow.

Fig. 8 demonstrates robustness of the predictor against wind direction fluctuations between 0 and 45 deg every 10 frames (upper panel). This corresponds to a maximum instantaneous change of 8.4 m s^{-1} in wind speed along a single direction.

Recently van Kooten, Doelman & Kenworthy (2019) have used typical wind profiles from the Thirty Metre Telescope (TMT) site to demonstrate effects of wind velocity variations in a data-driven linear minimum mean square error (LMMSE) predictor over a period of 5 s in numerical simulations. Wind data are linearly interpolated to system frequency to allow for per-frame fluctuation. Two adaptive variations, resetting-batch LMMSE and forgetting LMMSE, along with LMMSE were tested. Compared with these linear predictors, variances in WFEs of the ANN predictive loop before and after the disturbance are on the same order as that of the delay-compensated loop. This robustness can be explained as

Table 4. Four-layer turbulence profile used within test data set. r_0 is 0.157 m. L_0 is 25 m. Two sets of wind directions corresponding to Figs 9 and 10, respectively, are examined.

	Layer 1	Layer 2	Layer 3	Layer 4
Height (m)	0	4000	10 000	15 500
Relative strength	0.65	0.15	0.10	0.10
Wind speed (m s^{-1})	7.6	9.5	11.4	15.2
Wind direction (deg)	0	330	135	240
	0	0	0	0

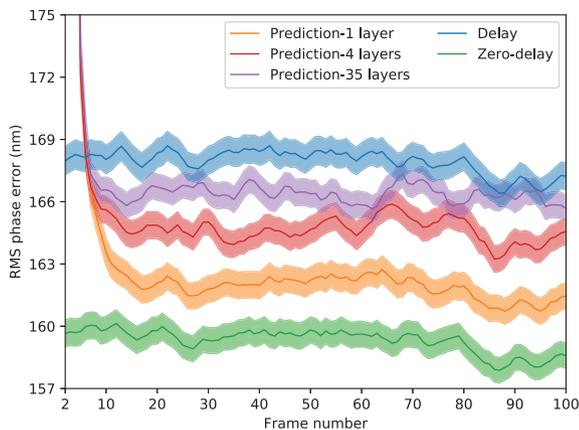


Figure 9. ANN performance with multiple turbulence layers moving along different directions. Wind speeds of either the 1- or 4-layer profile are scaled to maintain the same dynamics as that of the 35-layer profile. r_0 is 0.157 m.

the ANN predictor is allowed to use more spatial and temporal information when making inferences. Furthermore, the updating and forgetting mechanisms of our predictor are not fixed, but can constantly self-adjust according to the inputs, which by design allows for more flexible control on data flow.

4.4 Performance with multilayer turbulence

Though we train the predictor with a single turbulence layer, there usually exists several layers at high altitudes in addition to a strong ground layer (Farley et al. 2018, 2019). It is thus meaningful to test the predictor’s sensitivity to multiple layers moving with different velocities.

Here, we show the results obtained with ESO (European Southern Observatory) median 35-layer profile (Sarazin et al. 2013). r_0 is 0.157 m, slightly worse than during ANN training. We generated 1000 slope sequences each of 100 frames with this profile. For comparison, we also generated the same amount of test data of a single ground layer and of a four-layer profile (detailed in Table 4), both moving at 9.21 m s^{-1} (slightly slower than the training range), which is equivalent to the dynamics of the 35-layer profile.

Fig. 9 shows residual WFEs when wind vectors of multi-layer profiles (either the 4-layer or the 35-layer) move in different directions. For the 35-layer profile, the moving direction of each layer is a random integer between 0 and 360 deg. For the 4-layer profile, wind directions are listed in Table 4. The delayed and the delay-compensated correction loops behave similarly regardless of the number of layers, thus only values obtained from the single-layer profile are shown here. Mean RMS WFEs of the delayed, 35-layer predictive, 4-layer predictive, 1-layer predictive, and delay-

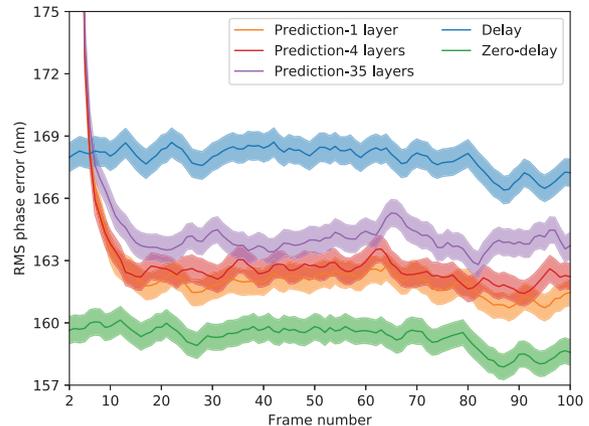


Figure 10. ANN performance with multiple turbulence layers moving along the same direction. Compared with Fig. 9, the ANN performance suffers from the increased number of wind vectors, but mainly from the variety among those vectors.

compensated correction loop after the 20th frame are 167.9, 166.4, 164.6, 161.9, and 159.2 nm, respectively.

Fig. 10 shows improved ANN performance when all layers in either multilayer profile move in the same direction (wind speeds are the same as used in Fig. 9). Mean RMS WFE of the 35-layer predictive loop decreases to 164.0 nm, slightly better than the 4-layer predictive loop when wind vectors are largely distinct from each other. Mean RMS WFE of the 4-layer predictive loop decreases to 162.4 nm, approaching that of the 1-layer predictive loop.

We think that the wind directions adopted represent two extreme conditions, and that performance with real turbulence profiles would fall within these two cases. These results show that the predictor trained on a single layer frozen-flow conditions is capable of providing performance improvement even when complex profiles with random wind directions are encountered.

4.5 Performance with two-frame latency

So far we have considered only single-frame delay in an AO loop, where we have accounted for WFS integration time only but ignored the time taken for real-time processing and the update of the surface shape of the DM.

In Fig. 11, we show the ANN performance when a more realistic loop delay of two frames is considered. We trained a separate ANN that was designed to predict two frames in advance in a single step. The training data set described in Section 3.2.1 was re-utilized in the way that $(s_1, s_2, \dots, s_{28})$ in each sequence is the ANN input and s_{30} is the training target. The training and hyperparameter searching setup follows that described in Section 3.2.2. The resulting network comprises two stacked LSTM cells and a final FC layer. The output vector sizes of the two LSTMs are 122 and 171, respectively, with a computational load of 0.9×10^8 FLOPS. The resulting mean rms WFE of this single-step predictive loop after the 30th frame is significantly reduced to 166.9 nm, compared with 225.6 and 157.1 nm of the two-frame delayed and zero-delayed loop, respectively.

As a comparison, the single-frame predictor was also used twice to provide a two-frame prediction: first, the measured (s_1, s_2, \dots, s_t) ($t \geq 2$) is fed into the predictor to generate the predicted \tilde{s}_{t+1} as it was designed; secondly, \tilde{s}_{t+1} is treated as its truth value s_{t+1} and forms part of the ANN input vector $(s_1, s_2, \dots, s_t, \tilde{s}_{t+1})$, which is then used to generate \tilde{s}_{t+2} . This resulted in a WFE of 174.4 nm,

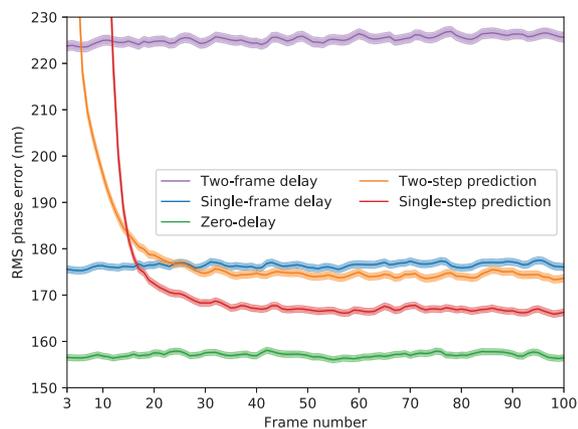


Figure 11. In a simulated system with a two-frame latency, the methodology adopted for the single-latency prediction is extended to training a separate ANN predictor (single-step prediction). In this case, the single-latency predictor can also be used twice (two-step prediction), albeit with worse performance. Both predictors improve the system performance significantly. The blue line representing the single-frame delay performance is the same as that shown in Fig. 4, and is depicted here for comparison with the two-frame delay performance. GS magnitude for test is 6. Wind speed is 15 m s^{-1} along a single direction.

worse than the two frame prediction, however, still significantly better than the two-frame delay.

5 ON-SKY IMPLEMENTATION

The results presented within this paper are based on simulations, which do not consider many of the practical issues relating to the implementation of the LSTM architecture within a real AO system. In this section, we discuss issues relating to calibration and control within a real system.

The training method presented here uses simulated Shack–Hartmann WFS slope data but could be applied to real data from any WFS data. However, the sensitivity of the ANN performance to the training regime and the requirement for a large number of wind velocities required for a robust ANN training means that the collection of a real WFS data-sets may take a significant amount of time. It may therefore be best to initially train in simulation and convert real WFS slopes to ensure that the subaperture geometry and pixel scale match that encoded within the ANN. Unlike other ANN approaches proposed for multiple GS AO (Osborn et al. 2012), the single WFS LSTM ANN does not require retraining for different targets, greatly reducing the calibration overhead of implementation within a real system.

The ANN predictor proposed here may not be applicable to all closed-loop AO systems where imperfect POLC (pseudo-open loop control) can introduce additional noise terms within the system, affecting performance and loop stability (Gilles 2005). To adapt the training regime here to closed-loop operation the training data set would have to be expanded to include the range of potential closed-loop gain values. This will increase both training time and the size of the resulting ANN, with no guarantee that the resulting ANN would be more resistant to the errors that can affect POLC stability such as misalignments and open-loop DM errors.

An on-sky implementation of the ANN presented here requires an additional processing step for each WFS before reconstruction within the system. The system simulated here uses a 7×7 subaperture Shack–Hartmann system that we selected such that

can be rapidly trained and tested. Furthermore, wind profiles can be recovered from recorded off-axis WFS data of the CANARY demonstrator (Laidlaw et al. 2019). By matching the configuration of CANARY, future comparison of multilayer predictions using real data and in simulation is feasible. Extending beyond this low-order system is possible but implies additional training time and an increase in real-time computational load. Due to the hyperparameter tuning approach adopted here, the precise computational load of a higher order system cannot be easily predicted, but implementation of this approach within any existing astronomical non-XAO system is feasible using existing hardware. There do however exist possibilities to reduce the computational load, including operating in actuator space where computational load is lower (Basden et al. 2019), or taking advantage of the sparsity of the ANN. The technique proposed here inherently scales to multiple GS systems through parallelism.

6 CONCLUSIONS

We have shown in extensive numerical simulations, the potential of ANNs as a non-linear framework for wavefront prediction. The memory elements within the LSTM network give it the ability to learn information such as wind velocity vectors from the data and to use that information in its prediction. The fluid nature of the memory allows the network to adapt to changes in such information without user tuning.

The residual wavefront error of the simulated 7×7 subaperture SCAO system with one frame delay improves significantly after the predictor is incorporated irrespective of GS magnitude and wind velocity. In addition to accurately predicting the wavefront we have also provided evidence that the ANN predictor also compensates for some centroiding and/or aliasing errors that can be temporally filtered from the wavefront. This behaviour however is dependent on the ANN training regime and was only observed when the system was trained on a low SNR tenth magnitude GS. The selection of the training regime has the greatest impact on the performance of the ANN prediction.

We have shown that the ANN predictor is robust to changes in wind velocity on sub-second time-scales, and that the ANN approach taken within this paper is transferable to systems with a two-frame delay. The ANN predictor trained on a single atmospheric turbulence layer is also capable of predicting under more complex conditions with multiple layers with independent wind vectors, albeit with reduced performance. Whilst we believe it is likely that a more realistic multilayer training environment and/or the use of multiple wavefront sensors to allow identification of layer altitudes will improve ANN performance on multilayer turbulence, this is subject to further study. Our next steps will be to investigate ANN performance on recorded CANARY data to investigate ANN stability and training in a real-world system.

ACKNOWLEDGEMENTS

The authors sincerely appreciate the comments from the reviewer, which helped us improve the paper. We gratefully acknowledge James Osborn from CfAI, Durham University for his valuable suggestions regarding the manuscript. We thank sincerely Ollie Farley from CfAI for the thought-provoking discussions with him. XL is funded by Durham University through their doctoral scholarship program and China Scholarship Council. CGG and JDCJ acknowledge financial support from the I+D 2017 project AYA2017-89121-P and JDCJ acknowledges support from the European Union’s

Horizon 2020 research and innovation programme under the H2020-INFRAIA-2018-2020 grant agreement no 210489629.

REFERENCES

- Andersen D. R., Jackson K. J., Blain C., Bradley C., Correia C., Ito M., Lardière O., Véran J.-P., 2012, *PASP*, 124, 469
- Basden A. G., Jenkins D., Morris T. J., Osborn J., Townson M. J., 2019, *MNRAS*, 486, 1774
- Chollet F. et al., 2015, Keras, Available at: <https://keras.io>
- Correia C., Jackson K., Véran J.-P., Andersen D., Lardière O., Bradley C., 2014, *J. Opt. Soc. Am. A*, 31, 101
- Farley O. J. D., Osborn J., Morris T., Fusco T., Neichel B., Correia C., Wilson R. W., 2019, *MNRAS*, 488, 213
- Farley O. J. D., Osborn J., Morris T., Sarazin M., Butterley T., Townson M. J., Jia P., Wilson R. W., 2018, *MNRAS*, 481, 4030
- Gal Y., Ghahramani Z., 2016, Proc. 30th International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, p. 1027
- Gers F. A., Schmidhuber J., Cummins F., 1999, Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470). IET, London, UK, p. 850
- Gilles L., 2005, *Appl. Opt.*, 44, 993
- Goodfellow I., Bengio Y., Courville A., 2016, Deep Learning. MIT Press, Cambridge, MA, USA
- Graves A., 2012, PhD thesis. Technische Universität at München
- Graves A., Mohamed A., Hinton G. E., 2013, IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, Piscataway, New Jersey, USA
- Guyon O., Males J., 2017, preprint ([arxiv:1707.00570](https://arxiv.org/abs/1707.00570))
- Hochreiter S., Schmidhuber J., 1997, Neural Comput., 9, 1735
- Hughes I. G., Hase T. P. A., 2010, Measurements and Their Uncertainties. Oxford Univ. Press, Oxford
- Jackson K., Correia C., Lardière O., Andersen D., Bradley C., 2015, *Opt. Lett.*, 40, 143
- Jorgenson M. B., Aitken G. J. M., 1992, *Opt. Lett.*, 17, 466
- Jorgenson M. B., Aitken G. J. M., 1994, ESO Conference and Workshop Proceedings, Vol. 48. European Southern Observatory, Garching near Munich, Germany, p. 143
- Karpathy A., Li F., 2014, IEEE Trans. Pattern Anal. Mach. Intell., 39, 664
- Kasper M., 2012, in Brent B. L., Marchetti E., Véran J.-P., eds, Proc. SPIE Conf. Ser. Vol. 8447, Adaptive Optics Systems III. SPIE, Bellingham, p. 84470B
- Kingma D., Ba J., 2014, preprint ([arxiv:1412.6980](https://arxiv.org/abs/1412.6980))
- Kulcsár C., Raynaud H.-F., Petit C., Conan J.-M., 2012, Automatica, 48, 1939
- Laidlaw D. J., Osborn J., Morris T. J., Basden A. G., Gendron E., Rousset G., Townson M. J., Wilson R. W., 2019, *MNRAS*, 491, 1287
- LeCun Y., Bengio T., Hinton G., 2015, *Nature*, 521, 436
- Le Roux B., Conan J.-M., Kulcsár C., Raynaud H.-F., Mugnier L. M., Fusco T., 2004, *J. Opt. Soc. Am. A*, 21, 1261
- Lloyd-Hart M., McGuire P., 1996, ESO Conference and Workshop Proceedings Vol. 54. European Southern Observatory, Garching near Munich, Germany, p. 95
- Males J. R., Guyon O., 2018, *J. Astron. Telesc. Instrum. Syst*, 4, 019001
- Morris T. et al., 2010, 1st International Conference on Adaptive Optics for Extremely Large Telescopes (AO4ELT), EDP Sciences, Les Ulis, France
- Noll R. J., 1976, *J. Opt. Soc. Am.*, 66, 207
- Osborn J., Juez F. J. D. C., Guzman D., Butterley T., Myers R., Guesalaga A., Laine J., 2012, *Opt. Express*, 20, 2420
- Paschall R. N., Anderson D. J., 1993, *Appl. Opt.*, 32, 6347
- Petit C., Conan J.-M., Kulcsár C., Raynaud H.-F., Fusco T., 2008, *Opt. Express*, 16, 87
- Petit C., Conan J.-M., Kulcsár C., Raynaud H.-F., Fusco T., Montri J., Rabaud D., 2006, in Ellerbrock B. L., Calia D. B., eds, Proc. SPIE Conf. Ser. Vol. 6272, Advances in Adaptive Optics II. SPIE, Bellingham, p. 62721T
- Petit C. et al., 2014, in Ramsay S. K., McLean I. S., Takami H., eds, Proc. SPIE Conf. Ser. Vol. 9148, Ground-based and Airborne Instrumentation for Astronomy V. SPIE, Bellingham, p. 91480O
- Poyneer L. A., Macintosh B. A., Véran J.-P., 2007, *J. Opt. Soc. Am. A*, 24, 2645
- Poyneer L. A., van Dam M., Véran J.-P., 2009, *J. Opt. Soc. Am. A*, 26, 833
- Poyneer L. A., Véran J.-P., 2008, *J. Opt. Soc. Am. A*, 25, 1486
- Reeves A., 2016, in Malbet F., Creech-Eakman M. J., Tuthill P. G., eds, Proc. SPIE Conf. Ser. Vol. 9909, Optical and Infrared Interferometry and Imaging V. SPIE, Bellingham, p. 99097F
- Rudy A., Poyneer L. A., Srinath S., Ammons S. M., Gavel D., 2015, preprint ([arxiv:1504.03686](https://arxiv.org/abs/1504.03686))
- Sarazin M., Le Louarn M., Ascenso J., Lombardi G., Navarrete J., 2013, Proc. Third AO4ELT Conference, INAF. Osservatorio Astrofisico di Arcetri, Firenze, Italy
- Sivo G. et al., 2014, *Opt. Express*, 22, 23565
- Sutskever I., Vinyals O., Le Q. V., 2014, in Ghahramani Z., Wlling M., Cortes C., Lawrence N. D., Weinberger K. Q., eds, Proc. SPIE Conf. Ser. Vol. 3104, Advances in Neural Information Processing Systems 27. SPIE, Bellingham, p. 3104
- van Kooten M., Doelman N., Kenworthy M., 2019, *J. Opt. Soc. Am. A*, 36, 731
- Wang L., Schöck M., Chanan G., 2008, *Appl. Opt.*, 47, 1880

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.