Title

Experimental beam combining stabilization using machine learning trained while phases drift.

Journal

Optics Express, 30(8)

ISSN

1094-4087

Authors

Du, Qiang

Wang, Dan

Zhou, Tong

et al.

Publication Date

2022-04-11

DOI

10.1364/oe.450255

Peer reviewed

To be published in Optics Express:

Title:	Experimental Beam Combining Stabilization using Machine Learning, Trained While Phases Drift
Authors:	Qiang Du,DAN WANG,Tong Zhou,Antonio Gilardi,Mariam Kiran,Bashir Mohammed,Derun Li,Russell wilcox
Accepted:	15 March 22
Posted	22 March 22
DOI:	https://doi.org/10.1364/OE.450255

Experimental Beam Combining Stabilization using Machine Learning, Trained While Phases Drift

Qiang Du[1*], Dan Wang[1], Tong Zhou[1], Antonio Gilardi[1], Mariam Kiran[1], Bashir Mohammed[1], Derun Li[1], Russell Wilcox[1]

[1] *Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA*
[*] *QDu@lbl.gov*

Abstract: An 8-beam, diffractive coherent beam combiner is phase controlled by a learning algorithm trained while optical phases drift, using a differential mapping technique. Combined output power is stable to 0.4% with 95% of theoretical maximum efficiency, limited by the diffractive element.

1. Introduction

Coherent beam combining (CBC) is a powerful technique for laser power scaling, and is key to a broad range of applications [1]. Large numbers of beams can be coherently combined with high efficiency in different ways, including tiled-aperture [2], and filled-aperture combining [3–5]. In all cases, it is imperative that the coherence of the whole beam array be maintained against environmental perturbations using an active stabilization controller which can sense phase errors. Despite difficulties of the large number of degrees of freedom, incomplete diagnostics and non-linearity, a deterministic error detector is desired for optimal feedback bandwidth and stability. Such an error detector must maintain knowledge of the combining process, in order to map between observations and actions in many channels. From a control system engineering point of view, the active stabilization of the coherence state against environmental perturbation can be seen as a many-in-many-out (MIMO) feedback system. The input array to the controller is intensity data with a number of elements m, and the output array is a number of phase actuators n. The job of a pattern recognizer is to map from the observation space (intensity array, $I \in \mathbb{R}^m$) to the action space (beam phases $\Phi \in \mathbb{R}^n$).

We have previously demonstrated that deterministic stabilization control can be achieved by pattern recognition, using a basic physics model and knowledge of the diffractive combiner element's complex transfer function in a spatial combiner of 8 beams in a 3×3 array [6], and 81 beams in a 9×9 array [7]. If all relevant parameters were known and constant, this model would be sufficient for control. However, a static model may not capture all the details in an operating system, is subject to uncontrolled parameter drift, and is difficult to verify.

Machine learning can control complex systems with many inputs and outputs connected by interlinked, nonlinear processes if those connections are consistent and reproducible. For example, it was shown that a simple, fully connected neural network (NN) can be trained to combine 81 beams [8] using interference pattern recognition. Deep reinforcement learning controllers have also shown promising capabilities [9, 10], where 128 pulses were temporally stacked in simulation [11], and 100 beams were spatially combined in experiment using two spatial-light-modulators (SLM) [12].

However, there are still critical problems to be solved. Training the machine to recognize patterns and corresponding input phase states implies the patterns are stable enough to be measured and correlated with controlled phases. This may not be the case in a realistic fiber

laser system where thermal drifts cause phases to continuously vary. It is possible to stabilize the phases using a non-deterministic control algorithm such as Stochastic Parallel Gradient Descent (SPGD) while the machine learns. Alternatively, one can make the training process robust against drift. We do this by training using the difference between two observations as the input phases are changed in a controlled way, basically finding a difference rather than an absolute value. This two-state dither scheme allows the phases to drift during training, because the two measurements are acquired in a short time interval compared with the drift rate. The method is named Deterministic Differential Remapping Method (DDRM), and was proposed in previous simulation work [13, 14].

Additionally, some uncontrolled system parameters will change with time, such as beam power and alignment. Slight drifts in these factors may degrade combination efficiency, so a static model is inadequate to solve the stabilization problem. Even if the machine learns to control the combiner for one state of these uncontrolled parameters, that learning will become less valid as they drift. Thus, the system must learn initially and keep learning during operation, while not perturbing the output power. The learning process is not dependent on knowing the physical model, so it can encompass unknown effects unaccounted for by a simplified model. While the feedback system may not be able to control all these effects, it can adjust its operation and, to a large extent, compensate for them. We therefore want to establish a learning method that can learn on unstable systems, to enable both initial and continual traning of the machine. Establishing that the differential phase error mapping technique works in a real system is a key step towards this goal.

In summary, the deterministic multidimensional stabilization of coherent beam combining should address the following challenges:

1. **Incomplete measurement**: The optical phase difference under control is not directly observable (without an additional interferometer, which adds undesirable complexity) and must be derived from intensity data.

2. **Non-uniqueness**: There may be a many-to-one mapping from phase to intensity pattern depending on the combiner transmission function.

3. **Nonlinearity**: The transfer function from phase difference to intensity difference is nonlinear, with additional amplifier nonlinearity and sensor saturation.

4. **Scalability**: As the number of beams increases, the feedback rate should remain the same. This is possible if the amount of error data increases proportionally with the number of beams.

5. **Drifting phase state**: The phase state of the system is constantly drifting due to unknown perturbations from the laser and the environment. This is particularly problematic for low sampling rate systems, such as ultrafast high power lasers with kHz repetition rate or less.

6. **Drifting uncontrolled parameters**: Parameters that are not under control tend to drift, which will change the mapping. The state of a multiple-beam combiner can be perturbed by factors including beam power imbalance, polarization change, beam spatial and temporal shape mismatch, etc. Currently, we only actively correct the group delay of each beam for phase control, since other parameters drift more slowly.

In our previous studies, we have addressed and solved problems 1–4 [7, 8], and proved that a model-based approach is capable of overcoming those challenges in experiment [6]. In this paper, we show that our DDRM machine learning controller can also solve 5 in experiment, and is promising for solving 6, without knowing a mathematical model of the system.

2. Method: Pattern Recognition by Neural Network

2.1. Training on drifting systems: Deterministic Differential Remapping Method (DDRM)

We use a diffractive optical element (DOE), an n-way splitter operated in reverse, to combine beams. Each input beam will itself be split into n beams. The beams next to it, displaced by one one inter-beam angle increment, will also each produce n beams. Since these groups of step-offset beams are superimposed, they will produce an interference pattern at the output which is wider than the diffraction pattern for one beam. For example, beams in an $N \times N$ array creates a $(2N-1) \times (2N-1)$ beam interference pattern. Our general approach is to train a neural network to correlate interference pattern differences with phase differences, so that we can train it on a drifting system and retrain during operation, tracking changes. The NN becomes a device that learns which differences in interference patterns are correlated with which vectors in phase space. After the training, given an arbitrary pattern within the training range, it can find the error vector for feedback.

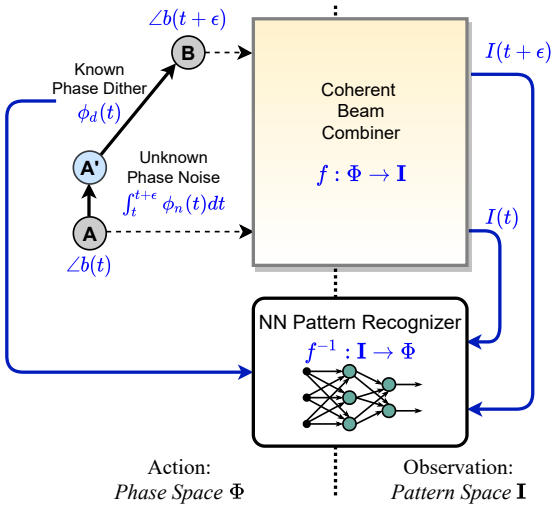


Fig. 1. Training process by pairing two pattern measurements $I(t), I(t+\epsilon)$ with a known phase dither $\phi_d(t)$, which is much faster than natural phase noise $\phi_n(t)$.

Fig. 1 shows the training process. Since the absolute beam phase state $\angle b(t)$ is unknown due to drift $\phi_n(t)$, we can inject a known phase dither $\phi_d(t)$ and measure the interference patterns before and after dithering, as $I(t)$ and $I(t+\epsilon)$, to build up a mapping between the action space and the observation space using the correlated data sample of $\{\phi_d(t), [I(t), I(t+\epsilon)]\}$.

The dither vector $\phi_d(t)$ is a random sample from an n-dimensional space, and drawn from uniformly distributed orthogonal matrices generated by the special orthogonal group $SO(n)$, which is widely used in many numerical applications (including Monte Carlo methods for best sampling efficiency in high-dimensional data spaces). The generated vectors $\phi_d(t) = \{\phi_d(t,i)\}, i \in [1, \cdots, n]$ will be statistically independent variables having zero mean and equal variances: $\langle \phi_d(t,i) \rangle = 0$, $\langle \phi_d(t,i)\phi_d(t,k) \rangle = \sigma_d^2 \delta_{ik}$, where δ_{ik} is the Kronecker delta. This is the same set of dither vectors being used in SPGD algorithms for CBC [15].

The prediction accuracy of a trained NN depends on the unknown drift amount, $\int_t^{t+\epsilon} \phi_n(t)\, dt$, the RMS dithering amount $\sigma_d(t)$, and the number of samples in the training dataset. Highly accurate prediction can be achieved if the dither speed is much faster than the natural drift rate: $\sigma_d(t) \gg \int_t^{t+\epsilon} \phi_n(t)\, dt$. by reducing the sampling period ϵ or increasing the RMS dithering

amount $\sigma_d(t)$, so that the drift contribution is negligible.

As a summary, in training process, if

$$\angle b(t+\epsilon) - \angle b(t) = \phi_d(t) + \int_t^{t+\epsilon} \phi_n(t)\, dt \simeq \phi_d(t)$$

there are mappings between phase space and pattern space:

$$f : \{\angle b(t), \angle b(t+\epsilon)\} \mapsto \{I(t), I(t+\epsilon)\}$$
$$f^{-1} : \{I(t), I(t+\epsilon)\} \mapsto \phi_d(t)$$

where $f : \Phi \to \mathbf{I}$ is the combiner function, and $f^{-1} : \mathbf{I} \to \Phi$ is the pattern recognizer function.

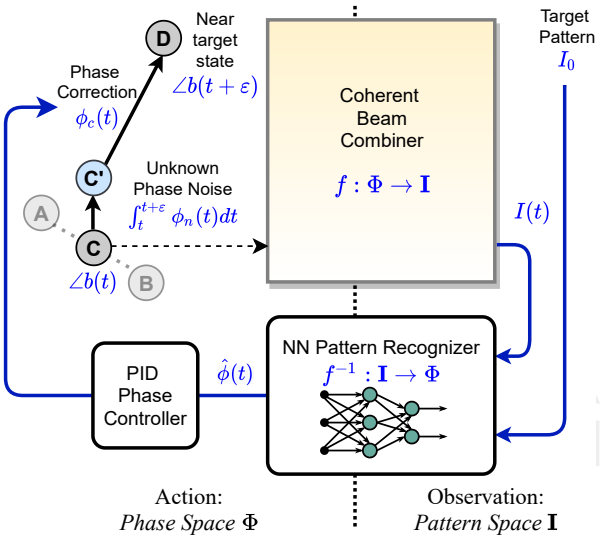


Fig. 2. Feedback process by correcting a predicted phase error vector, from a measured pattern $I(t)$ and a target pattern I_0.

Fig. 2 shows the feedback process, where the trained neural network predicts the distance vector $\hat{\phi}(t)$ in action space from any given pair of observation space samples. In case of coherent combining stabilization, it is always desired to have the best achievable combining efficiency, corresponding to a target interference pattern I_0. We find this I_0 in the training dataset, and use it as the destination (an input to the neural network) together with a measured pattern of $I(t)$, which corresponds to an arbitrary, starting phase state C. I_0 is chosen as the best pattern with highest combining efficiency, which is near the true optimal state if the random sampling grid is small enough, given enough training samples. The number of samples can be drastically reduced by learning from a SPGD controller as explained later. Although C is typically not seen in the training dataset, the inference process of the neural network acts like a multidimensional interpolation to predict the distance vector to the target state D. This is based on the NN's experience of A and B, and given that the sampling grid is smaller than the discontinuity of the multidimensional phase state landscape. Because of the group delay (latency) of feedback ε, the beam phase state may have drifted away from the measured state C by the time feedback is applied, resulting in an inaccurate correction, causing larger instability and lower combining efficiency. This is a common problem to all feedback control systems, and can be mitigated by introducing a controller such as a simple PID controller or a Kalman filter, which is more tolerant

to measurement inaccuracy. Discussions of feedback controller design for optimizing control robustness and accuracy is beyond the scope of this paper. Here our DDRM method serves as an error detector from a control system engineering point of view.

As a summary, in the feedback process, if feedback group delay ε is small, the NN pattern recognizer acts as a multidimensional phase detector, and the correction vector is predicted from a pair of interference patterns (target and measured), to stabilize the combiner phase state close to optimal, i.e. the state which has the highest combining efficiency seen from the training dataset.

$$f^{-1} : \{I(t), I_0\} \mapsto \hat{\phi}(t)$$
$$f : \{b(t) + \phi_c(t) + \int_t^{t+\varepsilon} \phi_n(t)\, dt\} \mapsto I(t+\varepsilon) \simeq I_0$$

2.2. NN architecture

For two-dimensional diffractive coherent beam combining, we show that a neural network using either multilayer perceptron (MLP) or convolutional neural network (CNN) structures are suitable for this function, and behave equally well in both simulation and experiment. In particular, a CNN recognizer reads the double-frame 2D pattern observations as if they are two color channel images, and the convolution layers look at adjacent side beams within the size of the DOE transmission function shape, extracting features out of it, before mapping to a vector in phase space. In contrast, MLP recognizer is more generic and can be applied to any MIMO control systems, where the input layer is just double the pattern space dimensionality ($2m$) and the output layer is n, flattened.

Although the degrees of freedom in phase space is actually $n-1$ (beam phases are all relative to an arbitrary reference beam), we found no problem in training and feedback with n outputs, because the training process is tolerant of the additional dimension, and apparently is able to use it when driving all n beams for combining.

MLP			CNN		
Type	Input Size	Output Size	Type	Input Shape	Output Size
Linear	2×25	60	2D convolutional	$2 \times (5 \times 5)$	64
ReLU			ReLU		
Linear	60	30	2D convolutional	64	32
ReLU			ReLU		
Linear	30	15	Flatten		
ReLU			Linear	32	15
Linear	15	8	Linear	15	8

Table 1. MLP and CNN architecture

For our 8-beam CBC in a 3×3 array with 5×5 pattern, the MLP and CNN neural network structures are listed in Table 1. The hyper parameters are tuned manually to find the best balance between size, inference time and prediction accuracy, and optimized with a minimal number of layers and neurons while still maintaining reproducible results. All experiment results shown in this paper are produced by an MLP pattern recognizer.

2.3. *Fast training in limited range: Learn while stabilized by SPGD*

Sampling a continuous multidimensional space can be costly in terms of data and time. We showed previously [8] that the many-to-one mapping and large dimensionality problems can be mitigated by training only on data from a limited range of phase space. Here, we prove that this concept works in experiment, and together with DDRM, we show that training samples can be from the recorded observation and action pair of any controller that can roughly maintain the optimal combining state, so that samples are taken near the target and the number of samples required is greatly reduced compared to a full-scale random sampling scheme.

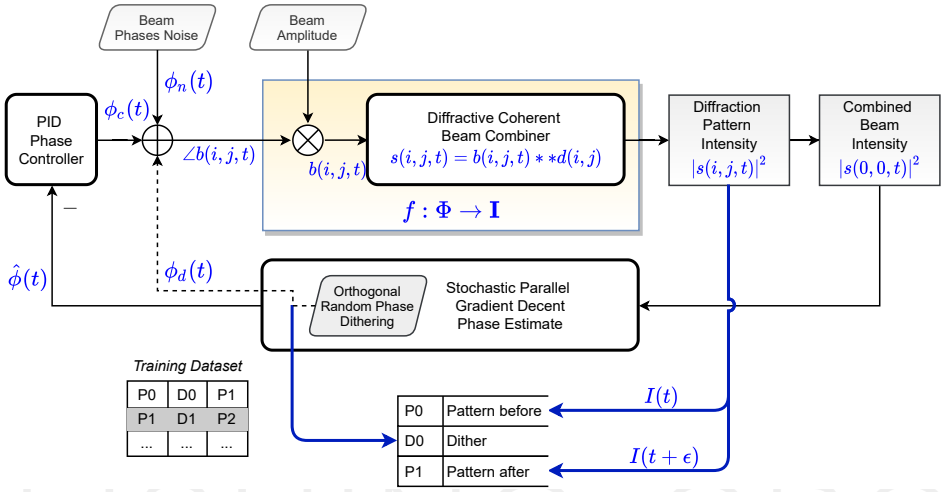


Fig. 3. Training dataset recorded from SPGD feedback. Dotted line indicates the SPGD phase dithering. Blue data path shows the training data samples.

As shown in Fig. 3, a conventional controller such as SPGD stabilizes the phase state near optimal, and the recorded observation and dithering pairs form a labelled training set for DDRM, where one sample is $\{I(t), \phi_d(t), I(t+\epsilon)\}$. In one SPGD correction step, both forward and backward dithering data are used.

For the 8-beam CBC in a 3×3 array, in simulation, we found that learning from SPGD only requires 600 samples and 20 episodes of training in order to achieve $\sim 99\%$ normalized efficiency and $< 0.5\%$ RMS stability, where similar performance requires 40,000 samples using a random dithering method without focusing near the optimal region.

3. Experiment Result

3.1. *Beam array formation and optical setup*

Having previously shown that the diffractive optic pair works with short pulses [4], we used a CW signal instead, with no change to the combining principle.

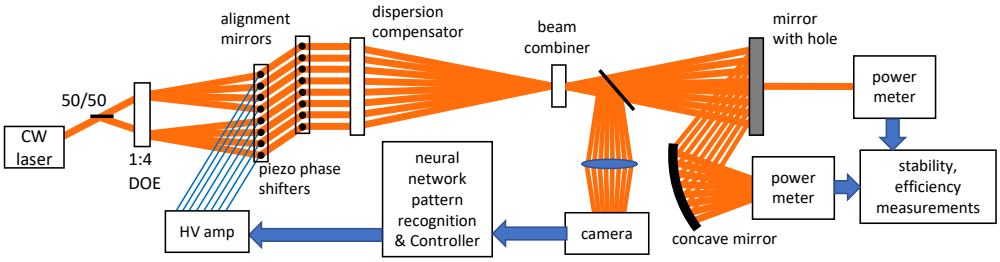


Fig. 4. Stabilized beam combining experiment optical setup

We implemented the control scheme on an 8-way beam combiner similar to that of [6], as shown in Figure 4. This device is designed to equalize delays and combine an array of high energy fiber amplifiers seeded by signals successively picked off from a central beam. For these tests we implemented piezo-actuated mirrors for phase control, using a single-frequency CW laser as a source, split eight ways using a separate diffractive optic. An array of mirrors forms the eight beams into a 3×3 matrix with no center beam, which is then incident on the angular dispersion compensating diffractive optic. After passing through the diffractive combining optic, a fraction of the output power is incident on a camera which images the far field spots, showing interference patterns associated with the input beam array phase state. This information is processed by the neural network to yield error signals applied to the piezo mirrors to shift optical phases. There are two power meters, measuring the combined beam power and the summation of side beam power respectively, for out-of-loop efficiency and stability measurements.

Latency (delay) of feedback signals, together with the bandwidth of the errors to be minimized, defines the stability of control. Latency should be much smaller than the error period, requiring fast hardware and software implementations. For our 8-beam CBC in a 3×3 array with 5×5 pattern, we found the inference time of our 4 layer MLP model is about 0.21 ms, and a CNN model with two convolutional layers is about 0.33 ms, on a typical CPU without any GPU acceleration. When quantized on an FPGA, the inference time of a 16-bit MLP model could be reduced to 85 ns (or 17 clock cycles at 200MHz).

The camera (Teledyne Dalsa Genie Nano M640 NIR) has a sensor without gamma correction, and the response is linear with respect to the number of photons received by the sensor. The camera is configured with Mono 10-bit pixel format readout through Ethernet at 100 frames per second, at a fixed exposure time of 1 ms without analog gain (39.7 dB SNR), so the sensor sensitivity is constant. The camera region-of-interest is 320×320 pixels, covering the 5×5 pattern with beam size of about 40×40 pixels. The 25 beam power values are calculated by integrating the camera pixel levels around each beam centroid, and are flattened as a floating point 1D array to feed into the MLP NN input layer, or a 2D matrix to the CNN input layer.

The piezo driver is an 8-channel, 16-bit DAC (digital-to-analog-converter) followed by an 8-channel piezo amplifier, and the measured step response rise time is about 0.4 ms. All data acquisition, neural network inference, and loop control tasks are performed on a PC (Ubuntu 18.08, Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz) without hardware acceleration, where the end-to-end group delay is about 4 ms. The 4 ms includes camera exposure, Ethernet communication, image processing, neural network inference, PID controller processing, communication to DAC FPGA, and piezo mirror actuation time. A PID controller is used to balance and optimize loop gain and stability. We operate the feedback loop at 100 Hz, which is sufficient given the noise bandwidth of the input beam as shown in Fig 5. The two power meters (Thorlabs PM16-121) have analog amplifier bandwidth of 10 Hz, and communicate through a USB interface. They are only used for out-of-loop measurement of absolute optical power and their bandwidth does not affect feedback control.

Besides the piezo mirror bandwidth limit, the feedback loop bandwidth is largely limited by CPU load, Gigabit Ethernet throughput, camera exposure and image processing time, lack of synchronous data acquisition, and the uncertainty of the non-realtime operating system. We are now implementing everything on a single FPGA chip, so the loop group delay can be reduced to sub-microsecond when using fast ADCs instead of camera.

3.2. *Input beam characterization*

We measured the input beam power and phase noise and calculated their power spectral density (PSD) as shown in Fig 5.

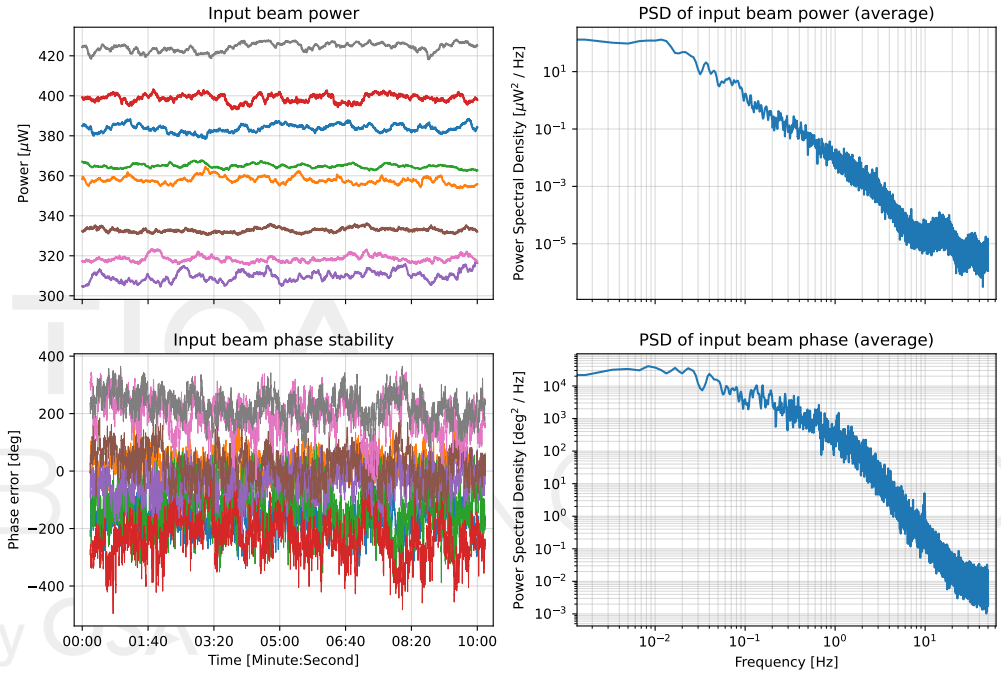


Fig. 5. Power and phase instabilities of incident beams. Data sampling rate is 100 Hz for both cases.

For power noise, we record the summation of two power meters, when only unblocking one input beam in free-running condition. The average RMS beam power stability is about 0.48% over 10 minutes, and the measured average noise spectral density rolls off at ~ 10Hz, due to the analog bandwidth of the power meters used for measurement. The power meters are out-of-loop, and we do not control beam powers. We made no attempt to balance the input beam powers either. The 0.48% RMS power stability contributes very little to the combining efficiency loss, according to the power fraction perturbation analysis $1-\eta = (\sigma_P/P)^2/4$ as in [16]. And the power induced interference pattern change is small for recognition detection, at a sampling rate of 100 Hz.

It is trickier to measure phase noise. Instead of trying to measure a free-running condition, we record the piezo mirror movement driven by each DAC output when the combiner is stabilized by a NN controller, which offers the best stability. This gives ~ 0.4% RMS in combined beam power, and $\eta =\sim 98.5\%$ normalized efficiency, which corresponds to about $\sigma_\phi = \sqrt{1-\eta} = 7.1^\circ$ RMS piston phase error [16]. Details about this estimation can be found at the end of Section 3.3. The DAC count is converted to optical phase by a calibration procedure interfering two beams

while quickly sweeping the phase of one of them. The estimated average phase stability is about $59 \pm 7.1°$ RMS over 10 minutes, and the noise bandwidth also falls within 20 Hz. The estimated phase noise includes the actual phase noise but also the NN based feedback error, detection noise and the PZT noise. Our sampling rate of 100 Hz is high enough to cover the majority phase noise bandwidth of 20 Hz.

When translating this stabilization scheme to high average power lasers, the control bandwidth will have to be increased. Sources of phase noise exhibited from high power fiber amplifiers include thermally-induced noise, amplified spontaneous emission (ASE), pump power fluctuation and thermo-acoustic phase noise, with an overall bandwidth of about 1 kHz (e.g. Optical Engines PE1000 amplifier, as shown in Fig.1(c) of [17]). Active stabilization in high power laser systems will require about ~10 kHz, and a low-latency controller with end-to-end group delay of microseconds. The DDRM control method described in this paper could achieve this goal when implemented in a realtime device such as an FPGA.

3.3. DDRM Training and Stabilization Performance

An SPGD controller is implemented with optimized gain, with dithering size of RMS 25°.

Training from SPGD records 8,000 samples (takes 80 seconds), uses < 40 episodes of training, resulting an RMS prediction error of about 11 degrees (validation loss of trained NN). We start recording data at a random phase state without waiting for the SPGD to converge, and the majority of the 8,000 samples are around the optimal state.

The trained NN is always able to achieve optimal combining efficiency with stability of < 1% in long term, regardless of any system conditions (e.g. optical alignment, beam power, camera settings, etc.). Fig. 6 shows the phase prediction error of the trained NN versus the number of episodes of training. After < 40 episodes, the phase prediction error reduces to < 13° RMS.

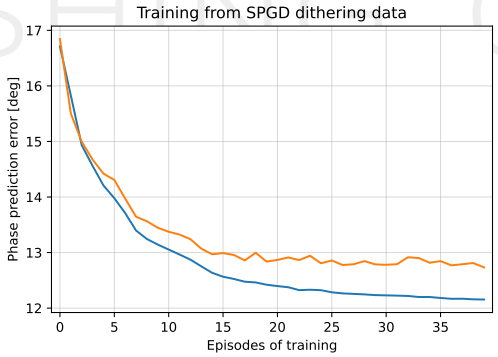
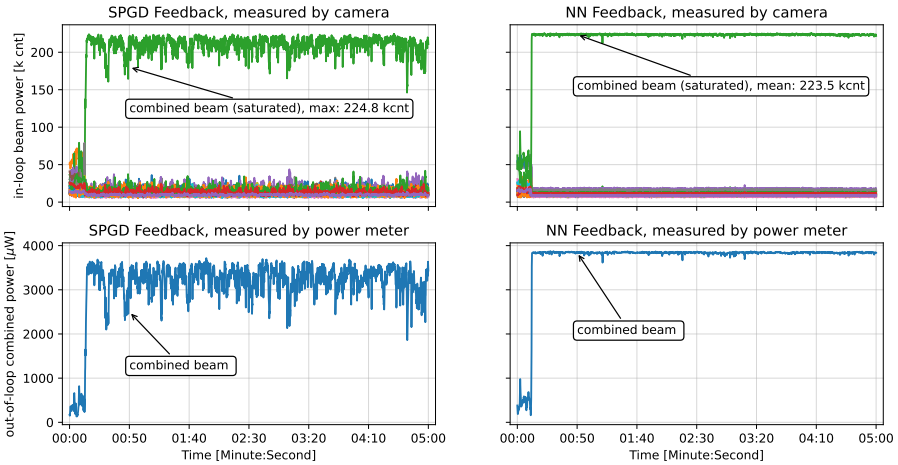


Fig. 6. Phase prediction error (calibrated from NN loss function) from SPGD data at dithering amount $\phi_d = 25°$ RMS. The blue curve is from the training process ("training loss"), and the orange curve is validation error from presenting the trained NN with new data samples (not in the training set, or "validation loss").

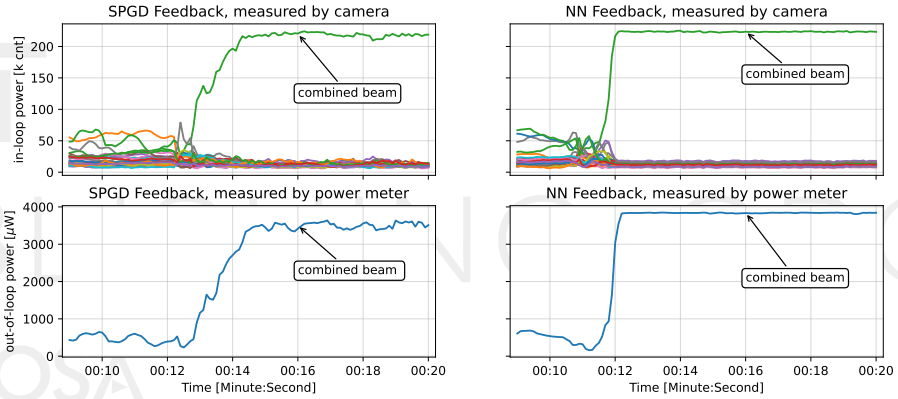
Once this training process is complete, the trained neural network is presented with a measured interference pattern $I(t)$ and the best pattern it has seen in the training dataset (the target pattern I_0). Fig 7 shows the in-loop (from camera) and out-of-loop (from power meter) measurements of SPGD feedback and NN feedback respectively, where NN is able to maintain maximum achievable efficiency without introducing dither noise.

The center beam is saturated in the camera measurement after the loop is closed. We use all 25 observable beam powers as input to NN, because they all contain detailed information for decoding the phase state, even in highly nonlinear (saturated) response regions. Presenting all

information to NN also helps during the locking process when the center beam is not saturated.



(a) Comparison between SPGD and NN feedback



(b) Zoom in at the moment when feedback is turned on

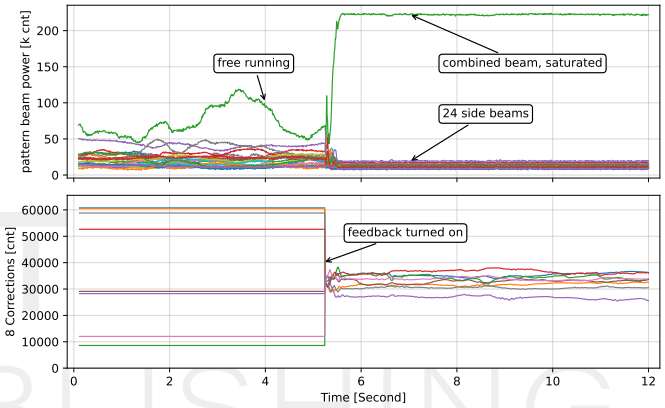
Fig. 7. Comparison between SPGD and NN feedback. Upper left: camera measurement of combined and side beams, from SPGD. Lower left: power meter measurements with SPGD. Upper right: camera-measured beam powers with NN. Lower right: power meter measurements with NN. The camera saturates at high intensity, while the power meter is linear.

Fig. 7 shows the difference between the SPGD algorithm and the NN over 5 minutes. The difference is due to several factors. At the camera sampling rate of $f_s = 100$ Hz, the cutoff frequency of SPGD algorithms is typically $f_s/10/N$ where $N = 8$ is the number of beams. This value is improved by a factor of 3 when using orthogonal dither [18], so the feedback bandwidth of SPGD is about 4 Hz in the present case. Our ultimate application will be limited in repetition rate to about 10kHz with 100 beams, giving 10Hz response, which is not adequate. Also, the SPGD is using only one sensor, that of the combined power, while the NN uses 25-spot interference patterns observed on the camera, gaining more data. This advantage scales with the number of beams. In addition, the SPGD must dither in order to determine how to correct the input phases. For training, we can use SPGD as a guide to the area around the optimal point, and not reduce performance of the NN.

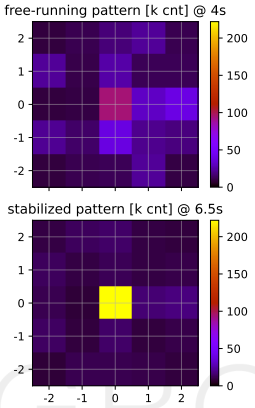
In addition to adding noise from random dither, the SPGD recovers slowly (in a fraction of

a second) from perturbations, given its cutoff bandwidth of ~ 4 Hz. Our pattern recognition scheme recovers in one cycle, and this is true for any number of beams, whereas the latency of SPGD scales with the number of beams. When feedback is turned on, both methods need multiple correction steps to reach the optimal phase state from a free-running, random phase state. The speed of convergence can be seen from Fig. 7b, where SPGD takes about 3 seconds and NN takes about 0.3 second.

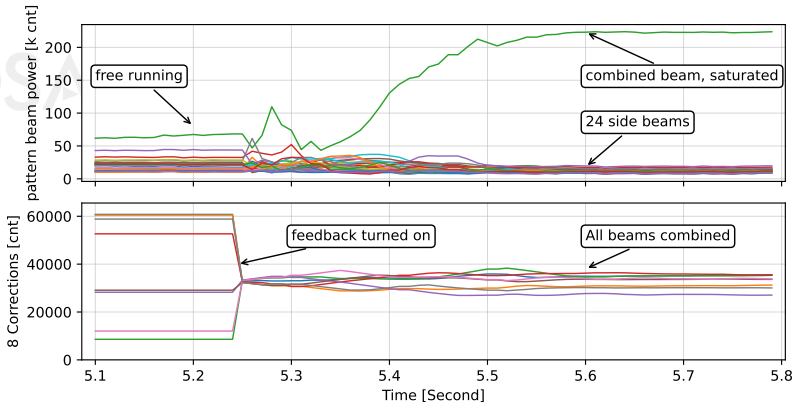
Under these conditions, the NN outperforms SPGD. In other applications with higher sample rates or lower latency, it is possible to achieve better performance with SPGD. What is demonstrated here is that for the set of conditions we have in our current application, the NN with DDRM achieves useful results and shows promise for future applications where repetition rate is limited. These comparison results clearly show the motivation behind our development of the DDRM method.



(a) History of output beam powers when activating feedback



(b) Patterns seem by camera



(c) zoom in at the moment when NN feedback is turned on

Fig. 8. Experimental 8-beam combining process, showing time near closing of the loop. (a) Upper: combined and side beam powers. Lower: DAC values. (b): interference pattern before and after locking, seen by the camera. (c): Beams combined from a random state takes less than 35 correction steps.

Fig 8a shows the typical locking process from free running state to combined state. Figure 8b shows interference patterns (center beam saturated) before and after the loop is closed. Fig 8c zooms in around the moment when the NN controller is turned on. It takes less than 35 steps for

NN controller to bring the combiner from an arbitrary phase state into the optimal combined state and maintain stability afterwards.

Fig. 9 shows the long term stability of a NN controlled loop using out-of-loop power meter measurements of the combined and total side beam powers, where stability of the combined beam is RMS 0.4% after locking, which is similar to the input beam power stability.

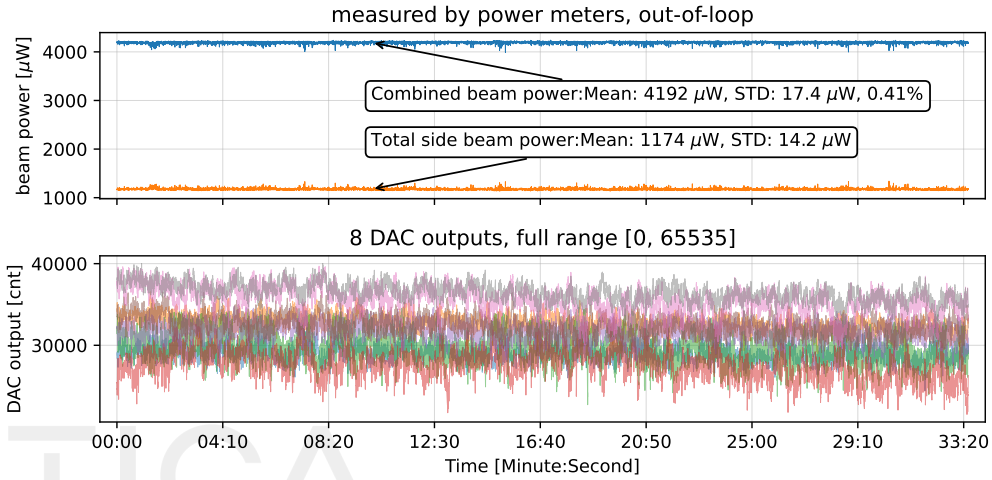


Fig. 9. Experiment data of 3×3 combiner using neural network recognizer stabilization, over 33 minutes. Upper graph: combined beam and side beam powers. Lower graph: DAC output values used to control phase during stabilization. Combining efficiency: $4192/(4192+1174) = 78.1\%$.

The practical combining efficiency is limited by the diffractive combiner, which can be determined by measuring its efficiency when used as a splitter. We determined the splitting efficiency of the combining DOE by sending one beam onto the DOE, and measuring the power of each beam versus the total power after the DOE (removing transmission loss). The resulting splitting efficiency is 82.1%, while the measured combining efficiency is 78.1%. The measured combining efficiency is therefore 95% of the theoretical maximum, indicating that our phase controller contributes little to the overall loss in efficiency. Diffractive splitters with efficiency in the high 90s% have been demonstrated.

In the current experiment, it is not possible to directly measure residual phase error when the controller is operational. This would require a separate, highly reliable diagnostic. However, an indirect, simplified estimation of the RMS error can be derived from the efficiency loss [16], $\sigma_\phi = \sqrt{1-\eta}$, where η is the normalized combining efficiency. This estimate assumes equal beam amplitudes and other ideal conditions, and a full assessment of estimation accuracy is beyond the scope of this paper, but normalized combining efficiency loss can still be used to find an upper bound of the RMS residual phase error. We calculate the average combined beam power using NN data, and find the maximum achievable combined beam power using SPGD data, so that the ratio of the two represents η. We used data shown in Figure 7, where the SPGD max power is 224.8 k counts and NN average power is 223.5 k counts. After correcting camera saturation, the measured η is 98.5%, corresponding to $\sigma_\phi = 7.1^\circ$ RMS residual phase error of the NN feedback loop. This is close to the 5.7° RMS piston phase error required for $< 1\%$ efficiency loss, according to [16]. From another point of view, based on phase estimation error during training (Figure 6) our NN has 13° RMS phase error detection accuracy. Since this is uncorrelated among 8 beams, the detection error averages down to $13/\sqrt{8} = 4.6^\circ$, which could

result in 7° RMS phase error, when using a tuned PID controller. Our numerical simulation also confirmed this estimation based on both instability and efficiency loss. Results from Monte-Carlo simulations using our 2D convolution model show that RMS phase error of 7.1 degrees leads to an instability of 0.46% and combining efficiency of 99.2%. These results are close to the simplified estimates above, and are described in more detail in an upcoming paper [14].

We ran the same experiments for 6 months, and the trained NN is always able to achieve similar combining efficiency to the SPGD controller, and maintain $< 1\%$ combined efficiency stability, measured by out-of-loop power meters.

We also studied the robustness of NN based feedback, by purposely introducing a small movement (20 jog steps, or $\sim 0.57°$) on one axis of a motorized alignment mirror, controlled by a piezo inertia motor controller (Thorlabs KIM101). This results in about 125° of phase shift due to changes in optical path length, over a few milliseconds. Observing the DAC output response, the NN control loop is able to adjust the DAC value of the corresponding channel and recover to the steady combined state in a few steps, without any noticeable degradation of stability or efficiency. The loop step response behavior is consistent with the locking process from a random state.

We also tested the robustness of the training process by blocking one beam. The retrained NN (with the same architecture) is still capable of combining 7 beams with similar performance, except the unused DAC channel is driving at the slew rate limited value, as expected.

4. Ongoing investigation of recursive learning

4.1. Method

A recursive relearning process, learning from data during stabilization, could capture and track system variations such as beam power drift, as shown in Fig. 10.

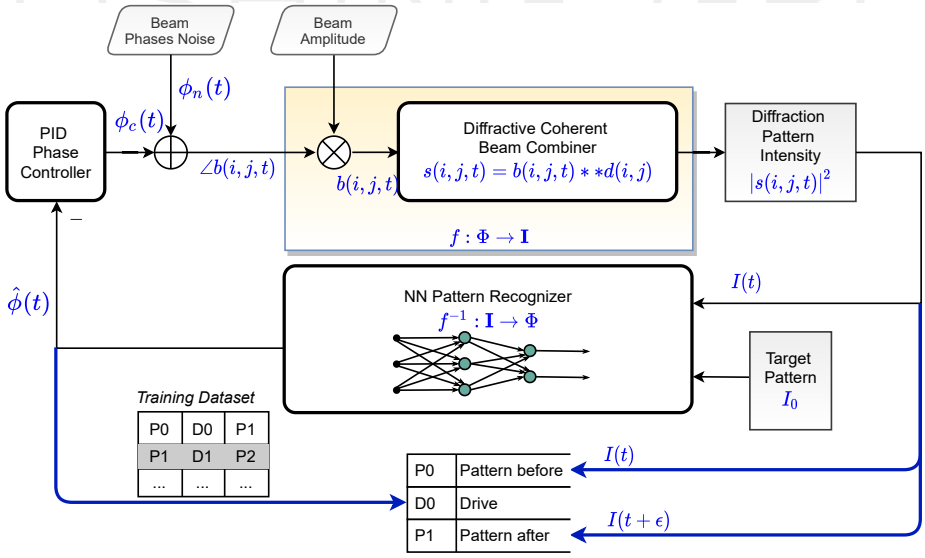


Fig. 10. NN feedback and recursive relearning processes using locking data. Blue data path shows the training data samples.

Since there is no dithering during this deterministic feedback, we use the predicted phase $\hat{\phi}(t)$ and the pattern pair as one sample in the training data set, and the NN is retrained from scratch using only the locking data without disturbing operation. In other words, there is no memory or

history from the previous iteration, and the NN weights are newly trained only from the locking data. The target pattern I_0 is also updated by selecting the highest efficiency pattern in the locking data. The training process is identical as described in Section 2.1. In this way, all external system transfer functions are captured and tracked, including the PID controller gain, optical coherent combining process, and measurement electronics response (such as camera settings).

4.2. Result and discussion

We implemented recursive relearning when recording the data in Fig. 9, where there are 27 relearning iterations. Each iteration records 4,000 samples, i.e. 40 seconds at 100 Hz sampling rate, where the RMS prediction error is $< 1^\circ$ because of the quiet locking data. The retraining process takes less than 20 seconds, followed by < 1 ms to swap NN weights when the CPU is waiting for the next camera frame. There is no loop instability or transient observed due to the NN update. The retrained NN is also capable of locking from a free running state, with performance similar to the SPGD-trained NN.

This demonstrates that the NN locking data contains enough information for retraining NN weights and maintaining consistent performance. The target pattern I_0 is also updated at each iteration, from the best of the 4,000 samples. These findings point to a potential solution to challenge 6, as stated in the introduction.

However, we found that the retraining process is not able to compensate for changes of uncontrolled parameters (e.g., large changes in the power of one beam), compared to SPGD training. This may be due to the very limited range of phase states in the training set, because there is no wide-ranging dither or exploration of the new phase landscape. Also, the effective phase dithering derived from locking data are not from an orthogonal set, and are thus less efficient in sampling. These are topics for future research toward implementation of recursive learning during operation.

In practice, with a 10kHz sampling rate, it only takes one second to get enough data samples for training with exploration (e.g. from SPGD), which is allowable for many applications that do not strictly require uninterruptible operation.

5. Conclusion

We have experimentally demonstrated that a neural network-based phase controller can learn while phases freely drift or are being controlled within some range, by employing a differential mapping technique, DDRM. This approach results in accurate data that can provide high resolution phase error estimates and tight control of beam combining efficiency. It also supports the possibility of relearning during operation, which could obviate the need to stop operation for retraining. The demonstrated neural network controller provides fast recovery from a random state, and single-step correction of small perturbations, when trained using an unconventional protocol. The DDRM method could have many other applications where it is necessary to train a machine using unstable data.

6. Acknowledgments

This work is supported by the Office of Science, Office of High Energy Physics, of the U.S. Department of Energy under contract no. DE-AC02-05CH11231.

7. Disclosures

The authors declare no conflicts of interest.

8. Data availability statement

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

9. References

References

1. R. Falcone, F. Albert, F. Beg, S. Glenzer, T. Ditmire, T. Spinka, and J. Zuegel, "Workshop report: Brightest light initiative," Tech. rep., OSA Headquarters, Washington, D.C, United States (2019).
2. I. Fsaifes, L. Daniault, S. Bellanger, M. Veinhard, J. Bourderionnet, C. Larat, E. Lallier, E. Durand, A. Brignon, and J.-C. Chanteloup, "Coherent beam combining of 61 femtosecond fiber amplifiers," Opt. Express **28**, 20152–20161 (2020).
3. S. M. Redmond, D. J. Ripin, X. Y. Charles, S. J. Augst, T. Y. Fan, P. A. Thielen, J. E. Rothenberg, and G. D. Goodno, "Diffractive coherent combining of a 2.5 kw fiber laser array into a 1.9 kw gaussian beam," Opt. letters **37**, 2832–2834 (2012).
4. T. Zhou, Q. Du, T. Sano, R. Wilcox, and W. Leemans, "Two-dimensional combination of eight ultrashort pulsed beams using a diffractive optic pair," Opt. Lett. **43**, 3269–3272 (2018).
5. M. Müller, C. Aleshire, A. Klenke, E. Haddad, F. Légaré, A. Tünnermann, and J. Limpert, "10.4 kw coherently combined ultrafast fiber laser," Opt. Lett. **45**, 3083–3086 (2020).
6. Q. Du, T. Zhou, L. R. Doolittle, G. Huang, D. Li, and R. Wilcox, "Deterministic stabilization of eight-way 2d diffractive beam combining using pattern recognition," Opt. Lett. **44**, 4554–4557 (2019).
7. Q. Du, D. Wang, T. Zhou, D. Li, and R. Wilcox, "81-beam coherent combination using a programmable array generator," Opt. Express **29**, 5407–5418 (2021).
8. D. Wang, Q. Du, T. Zhou, D. Li, and R. Wilcox, "Stabilization of the 81-channel coherent beam combination using machine learning," Opt. Express **29**, 5694–5709 (2021).
9. H. Tünnermann and A. Shirakawa, "Deep reinforcement learning for tiled aperture beam combining in a simulated environment," J. Physics: Photonics **3**, 015004 (2021).
10. H. Tünnermann and A. Shirakawa, "Deep reinforcement learning for coherent beam combining applications," Opt. Express **27**, 24223–24230 (2019).
11. A. Abuduweili, J. Wang, B. Yang, A. Wang, and Z. Zhang, "Reinforcement learning based robust control algorithms for coherent pulse stacking," Opt. Express **29**, 26068–26081 (2021).
12. M. Shpakovych, G. Maulion, V. Kermene, A. Boju, P. Armand, A. Desfarges-Berthelemot, and A. Barthélemy, "Experimental phase control of a 100 laser beam array with quasi-reinforcement learning of a neural network in an error reduction loop," Opt. Express **29**, 12307–12318 (2021).
13. Q. Du, D. Wang, T. Zhou, C. Bakalis, V. Vylta, D. Li, and R. Wilcox, "Stabilizing coherently combined beam power using a robust learning algorithm," in *2021 OSA Laser Congress,* (2021), p. JM3A.23.
14. D. Wang, Q. Du, T. Zhou, A. Gilardi, D. Li, and R. Wilcox, "Coherent combination using learned interference pattern recognition in simulations with system drift," Appl. Opt. ((manuscript in preparation) 2022).
15. P. Zhou, Z. Liu, X. Wang, Y. Ma, H. Ma, X. Xu, and S. Guo, "Coherent beam combining of fiber amplifiers using stochastic parallel gradient descent algorithm and its application," IEEE journal selected topics quantum electronics **15**, 248–256 (2009).
16. G. D. Goodno, C.-C. Shih, and J. E. Rothenberg, "Perturbative analysis of coherent combining efficiency with mismatched lasers," Opt. Express **18**, 25403–25414 (2010).
17. P. Ayyaswamy, C.L.Linslal, A.Dixit, D.Venkitesh, and B.Srinivasan, "Data-driven modeling of phase noise sources in coherent beam combining," in *2021 OSA Laser Congress,* (2021), p. JTu1A.42.
18. S. M. Redmond, J. E. Kansky, K. J. Creedon, L. J. Missaggia, M. K. Connors, G. W. Turner, T. Y. Fan, and A. Sanchez-Rubio, "Active coherent combination of > 200 semiconductor amplifiers using a spgd algorithm," in *CLEO: 2011 - Laser Science to Photonic Applications,* (2011), pp. 1–2.