Nonlinear Interference Mitigation via Deep Neural Networks

Christian Häger[1,2] and Henry D. Pfister[2]

[1] Department of Electrical Engineering, Chalmers University of Technology, Gothenburg

[2] Department of Electrical and Computer Engineering, Duke University, Durham

Optical Fiber Communication Conference (OFC)
San Diego, USA, March 14, 2018

FORCE FIBER-OPTIC COMMUNICATIONS RESEARCH CENTER

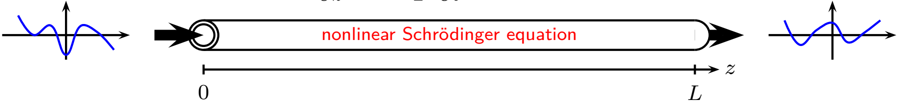
Duke UNIVERSITY

CHALMERS

Outline

Digital Backpropagation

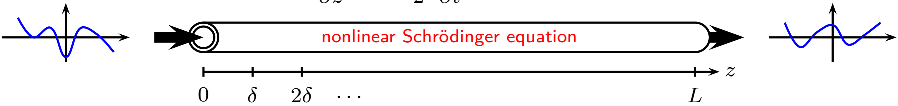
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



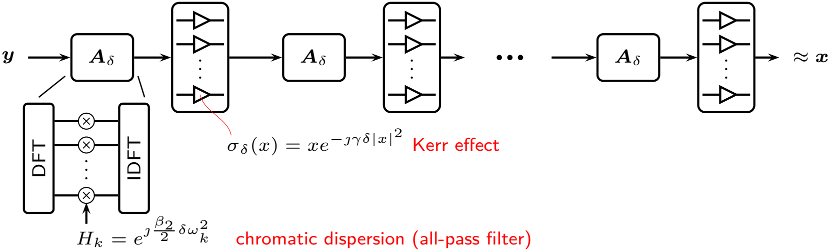
- Invert partial differential equation in real time ([Paré et al., 1996], [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008])

Digital Backpropagation

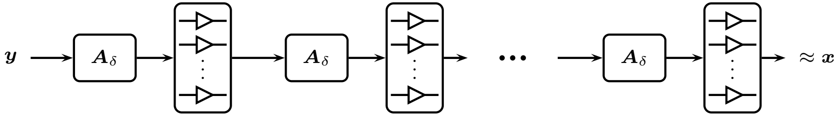
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



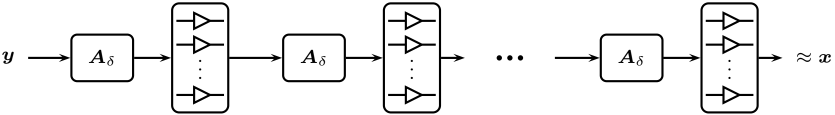
- Invert partial differential equation in real time ([Paré et al., 1996], [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008])
- Split-step Fourier method with M steps ($\delta = L/M$):



Complexity-Reduced Digital Backpropagation

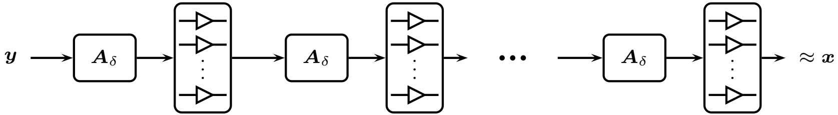


Complexity-Reduced Digital Backpropagation



Extensive literature: [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .

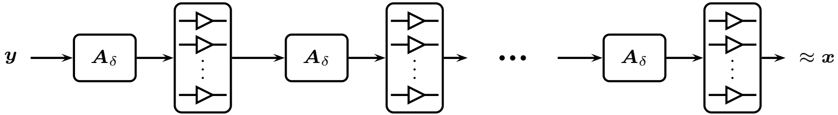
Complexity-Reduced Digital Backpropagation



Extensive literature: [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .

- Complexity increases with the number of steps M

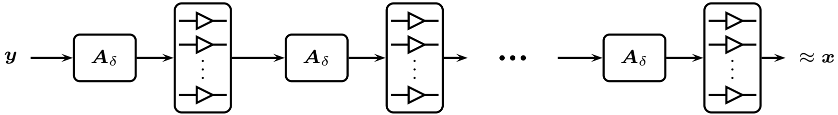
Complexity-Reduced Digital Backpropagation



Extensive literature: [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .

- Complexity increases with the number of steps M
- Therefore, reduce M as much as possible (step-reducing approaches)

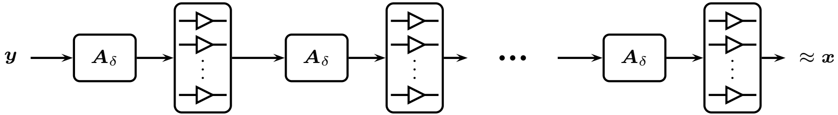
Complexity-Reduced Digital Backpropagation



Extensive literature: [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .

- Complexity increases with the number of steps M
- Therefore, reduce M as much as possible (step-reducing approaches)
- Intuitive, but . . .

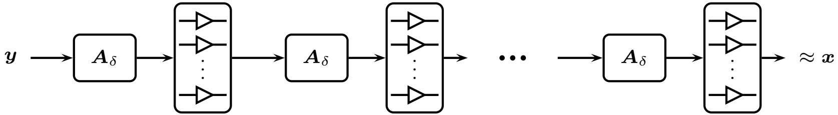
Complexity-Reduced Digital Backpropagation



Extensive literature: [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .

- Complexity increases with the number of steps M
- Therefore, reduce M as much as possible (step-reducing approaches)
- Intuitive, but . . .
- . . . this corresponds to flattening a deep (multi-layer) computation graph

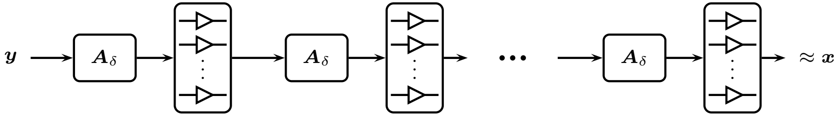
Complexity-Reduced Digital Backpropagation



Extensive literature: [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .

- Complexity increases with the number of steps M
- Therefore, reduce M as much as possible (step-reducing approaches)
- Intuitive, but . . .
- . . . this corresponds to flattening a deep (multi-layer) computation graph
- Machine learning: deep computation graphs work much better and are more parameter efficient than shallow ones

Complexity-Reduced Digital Backpropagation

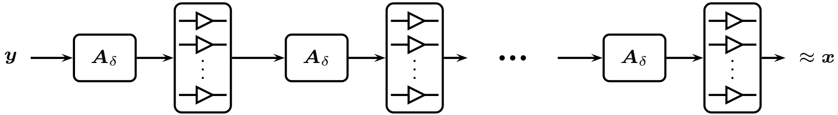


Extensive literature: [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .

- Complexity increases with the number of steps M
- Therefore, reduce M as much as possible (step-reducing approaches)
- Intuitive, but . . .
- . . . this corresponds to flattening a deep (multi-layer) computation graph
- Machine learning: deep computation graphs work much better and are more parameter efficient than shallow ones

Are many steps really that bad in terms of computational efficiency?

Complexity-Reduced Digital Backpropagation



Extensive literature: [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .

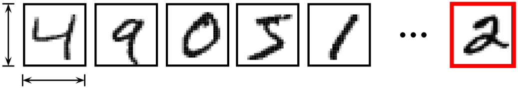
- Complexity increases with the number of steps M
- Therefore, reduce M as much as possible (step-reducing approaches)
- Intuitive, but . . .
- . . . this corresponds to flattening a deep (multi-layer) computation graph
- Machine learning: deep computation graphs work much better and are more parameter efficient than shallow ones

Are many steps really that bad in terms of computational efficiency?

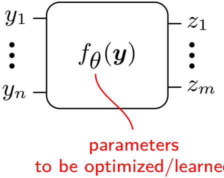
This work: use machine learning to jointly optimize all dispersion steps

Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)



28×28 pixels $\implies n = 784$

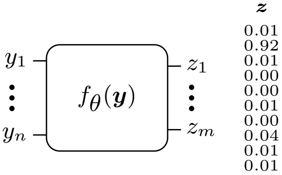


$\boldsymbol{z}$

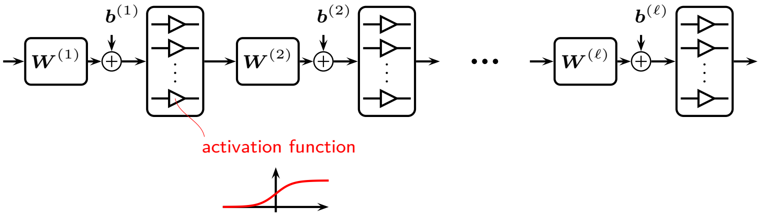
0.01
0.92
0.01
0.00
0.00
0.01
0.00
0.04
0.01
0.01

Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)

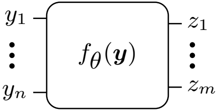


How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks



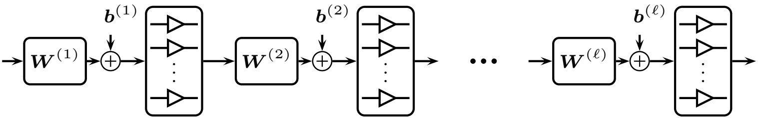
Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)



$\boldsymbol{z}$	$\boldsymbol{x}$
0.01	0
0.92	1
0.01	0
0.00	0
0.00	0
0.01	0
0.00	0
0.04	0
0.01	0
0.01	0

How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks

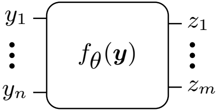


How to optimize $\theta = \{\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(\ell)}, \boldsymbol{b}^{(1)}, \dots, \boldsymbol{b}^{(\ell)}\}$? Deep learning

$$\min_\theta \sum_{i=1}^{N} \text{Loss}(f_\theta(\boldsymbol{y}^{(i)}), \boldsymbol{x}^{(i)}) \triangleq g(\theta) \qquad \text{using} \quad \theta_{k+1} = \theta_k - \lambda \nabla_\theta g(\theta_k) \qquad (1)$$

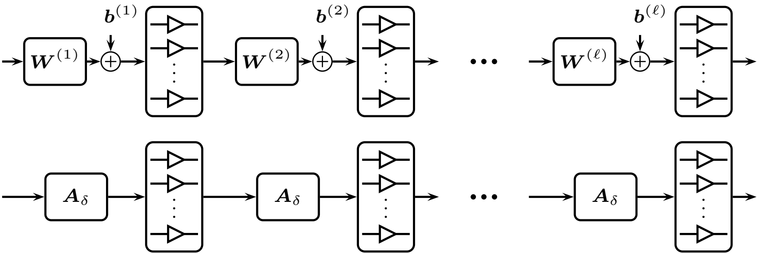
Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)



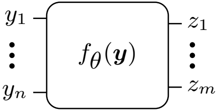
$\boldsymbol{z}$	$\boldsymbol{x}$
0.01	0
0.92	1
0.01	0
0.00	0
0.00	0
0.01	0
0.00	0
0.04	0
0.01	0
0.01	0

How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks



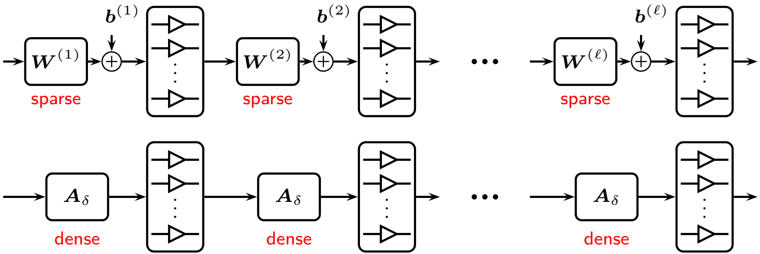
Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)

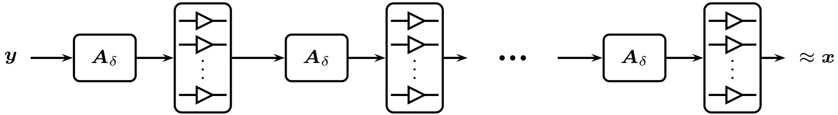


$\boldsymbol{z}$	$\boldsymbol{x}$
0.01	0
0.92	1
0.01	0
0.00	0
0.00	0
0.01	0
0.00	0
0.04	0
0.01	0
0.01	0

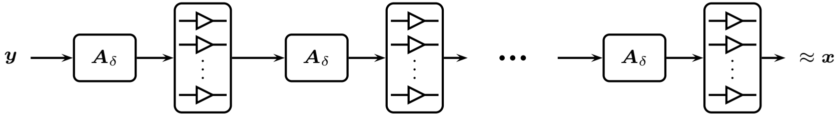
How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks

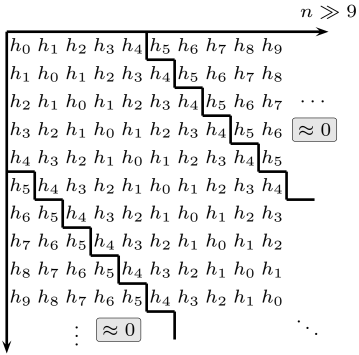


Time-Domain Implementation and Truncation

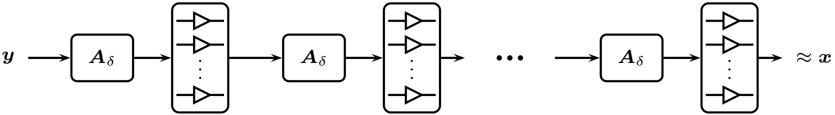


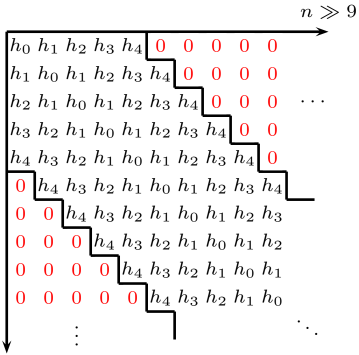
Time-Domain Implementation and Truncation

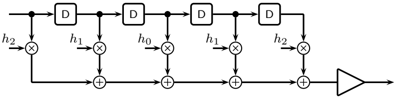




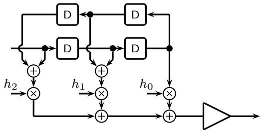
Time-Domain Implementation and Truncation





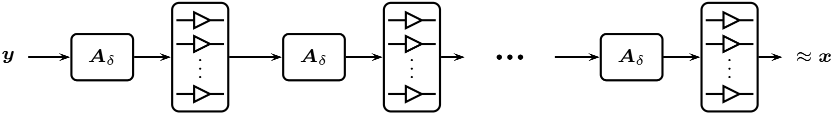


finite impulse response (FIR) filter

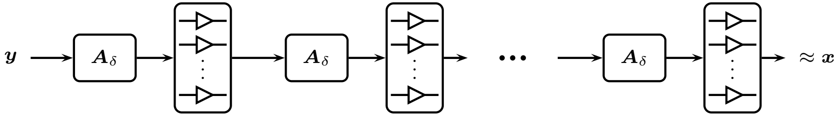


symmetric filter coefficients
$\Longrightarrow$ folded implementation

Time-Domain Implementation and Truncation

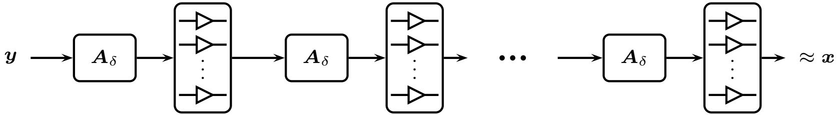


Time-Domain Implementation and Truncation



- Time-domain (FIR) DBP: [Li et al., 2008], [Ip and Kahn, 2008], [Zhu et al., 2009], [Goldfarb and Li, 2009], [Fougstedt et al., 2017a], [Fougstedt et al., 2017b]

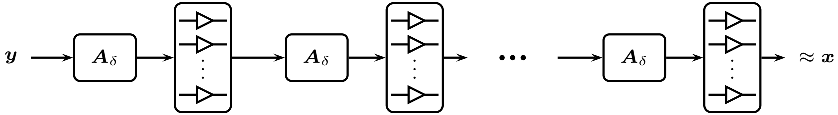
Time-Domain Implementation and Truncation



- Time-domain (FIR) DBP: [Li et al., 2008], [Ip and Kahn, 2008], [Zhu et al., 2009], [Goldfarb and Li, 2009], [Fougstedt et al., 2017a], [Fougstedt et al., 2017b]

Nontrivial to get good performance–complexity tradeoff

Time-Domain Implementation and Truncation

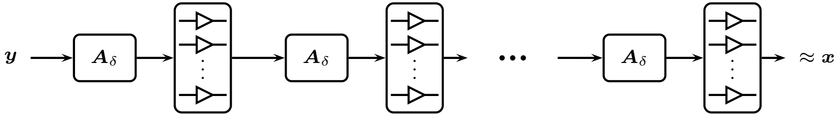


- Time-domain (FIR) DBP: [Li et al., 2008], [Ip and Kahn, 2008], [Zhu et al., 2009], [Goldfarb and Li, 2009], [Fougstedt et al., 2017a], [Fougstedt et al., 2017b]

Nontrivial to get good performance–complexity tradeoff

Example for $R_{\text{symb}} = 10.7$ Gbaud, $L = 2000\,\text{km}$ [Ip and Kahn, 2008]

Time-Domain Implementation and Truncation



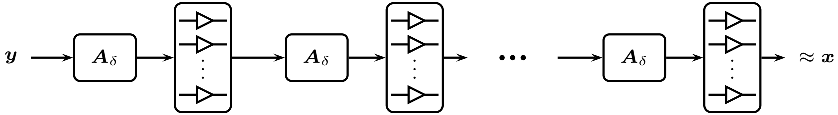
- Time-domain (FIR) DBP: [Li et al., 2008], [Ip and Kahn, 2008], [Zhu et al., 2009], [Goldfarb and Li, 2009], [Fougstedt et al., 2017a], [Fougstedt et al., 2017b]

Nontrivial to get good performance–complexity tradeoff

Example for $R_{\text{symb}} = 10.7$ Gbaud, $L = 2000$ km [Ip and Kahn, 2008]

- Delay spread formula: $T \approx 2\pi\beta_2 L R_{\text{symb}}^2 \rho$ (required number of taps)

Time-Domain Implementation and Truncation



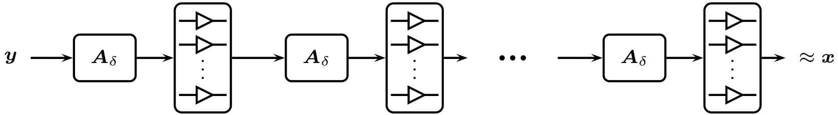
- Time-domain (FIR) DBP: [Li et al., 2008], [Ip and Kahn, 2008], [Zhu et al., 2009], [Goldfarb and Li, 2009], [Fougstedt et al., 2017a], [Fougstedt et al., 2017b]

Nontrivial to get good performance–complexity tradeoff

Example for $R_{\text{symb}} = 10.7$ Gbaud, $L = 2000\,\text{km}$ [Ip and Kahn, 2008]

- Delay spread formula: $T \approx 2\pi\beta_2 L R_{\text{symb}}^2 \rho$ (required number of taps)
- Linear equalization, $\rho = 1.5$ samples/symbol: $T \approx 47$ ✓

Time-Domain Implementation and Truncation



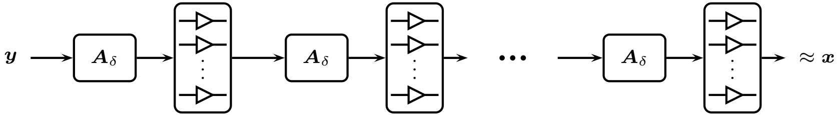
- Time-domain (FIR) DBP: [Li et al., 2008], [Ip and Kahn, 2008], [Zhu et al., 2009], [Goldfarb and Li, 2009], [Fougstedt et al., 2017a], [Fougstedt et al., 2017b]

Nontrivial to get good performance–complexity tradeoff

Example for $R_{\text{symb}} = 10.7$ Gbaud, $L = 2000$ km [Ip and Kahn, 2008]

- Delay spread formula: $T \approx 2\pi\beta_2 L R_{\text{symb}}^2 \rho$ (required number of taps)
- Linear equalization, $\rho = 1.5$ samples/symbol: $T \approx 47$ ✓
- DBP 1 step per span, $\rho = 3$ samples/symbol: $T \approx 125$

Time-Domain Implementation and Truncation



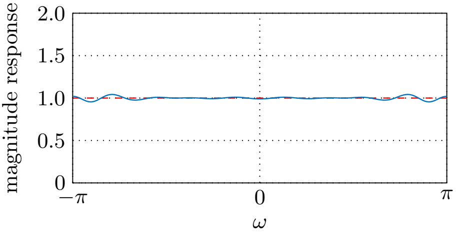
- Time-domain (FIR) DBP: [Li et al., 2008], [Ip and Kahn, 2008], [Zhu et al., 2009], [Goldfarb and Li, 2009], [Fougstedt et al., 2017a], [Fougstedt et al., 2017b]

Nontrivial to get good performance–complexity tradeoff

Example for $R_{\text{symb}} = 10.7$ Gbaud, $L = 2000\,\text{km}$ [Ip and Kahn, 2008]

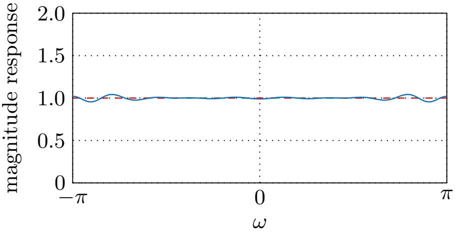
- Delay spread formula: $T \approx 2\pi\beta_2 L R_{\text{symb}}^2 \rho$ (required number of taps)
- Linear equalization, $\rho = 1.5$ samples/symbol: $T \approx 47$ ✓
- DBP 1 step per span, $\rho = 3$ samples/symbol: $T \approx 125$ ✗
- However, $\gg 1000$ taps required for good performance (70 taps per step)

Problem: Truncation Errors

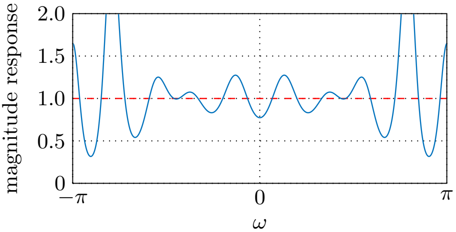


$$\boldsymbol{h}^{(1)} = \boldsymbol{h}^{(2)} = \cdots = \boldsymbol{h}^{(M)}$$

Problem: Truncation Errors

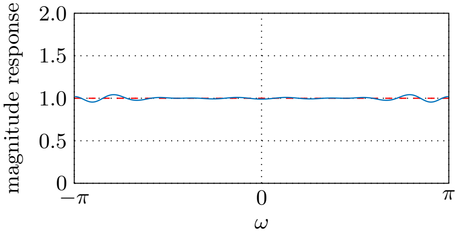


$$\boldsymbol{h}^{(1)} = \boldsymbol{h}^{(2)} = \cdots = \boldsymbol{h}^{(M)}$$



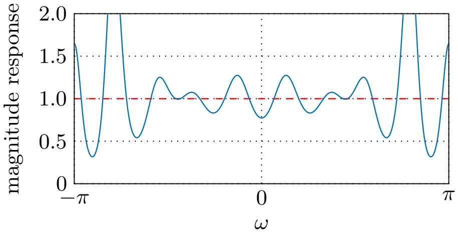
$$\boldsymbol{h}^{(1)} * \boldsymbol{h}^{(2)} * \cdots * \boldsymbol{h}^{(M)}$$

Problem: Truncation Errors



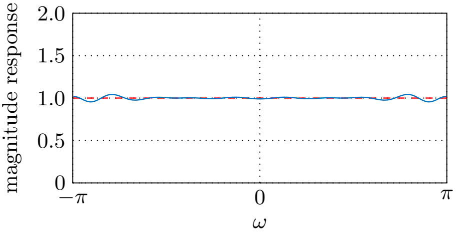
$$\boldsymbol{h}^{(1)} = \boldsymbol{h}^{(2)} = \cdots = \boldsymbol{h}^{(M)}$$

FIR filter design methods:
- time sampling and truncation [Savory, 2008]
- frequency-domain sampling [Ip and Kahn, 2008]
- wavelets [Goldfarb and Li, 2009]
- least-squares [Eghbali et al., 2014], [Sheikh et al., 2016]



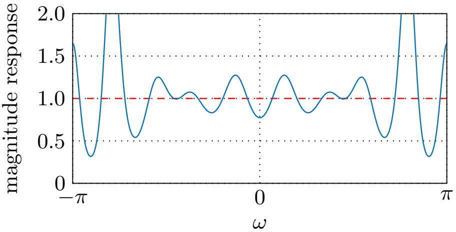
$$\boldsymbol{h}^{(1)} * \boldsymbol{h}^{(2)} * \cdots * \boldsymbol{h}^{(M)}$$

Problem: Truncation Errors



$$\boldsymbol{h}^{(1)} = \boldsymbol{h}^{(2)} = \cdots = \boldsymbol{h}^{(M)}$$

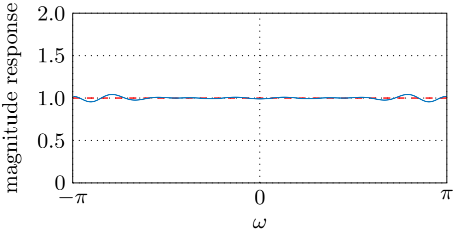
FIR filter design methods:
- time sampling and truncation [Savory, 2008]
- frequency-domain sampling [Ip and Kahn, 2008]
- wavelets [Goldfarb and Li, 2009]
- least-squares [Eghbali et al., 2014],
[Sheikh et al., 2016]



$$\boldsymbol{h}^{(1)} * \boldsymbol{h}^{(2)} * \cdots * \boldsymbol{h}^{(M)}$$

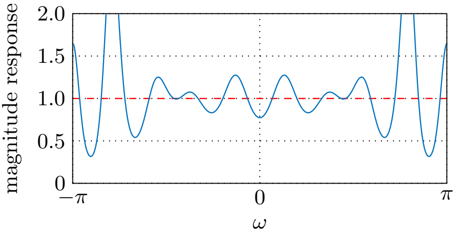
?

Problem: Truncation Errors



$$\boldsymbol{h}^{(1)} = \boldsymbol{h}^{(2)} = \cdots = \boldsymbol{h}^{(M)}$$

FIR filter design methods:
- time sampling and truncation [Savory, 2008]
- frequency-domain sampling [Ip and Kahn, 2008]
- wavelets [Goldfarb and Li, 2009]
- least-squares [Eghbali et al., 2014], [Sheikh et al., 2016]

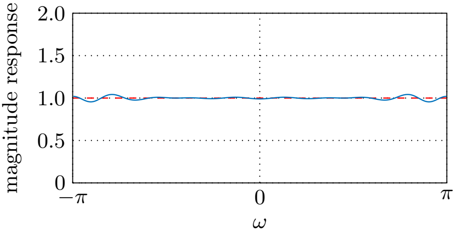


$$\boldsymbol{h}^{(1)} * \boldsymbol{h}^{(2)} * \cdots * \boldsymbol{h}^{(M)}$$

- filter pair design [Zhu et al., 2009]

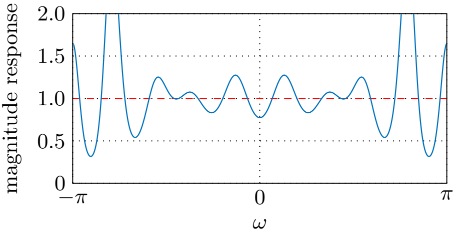
Filter coefficient quantization:
- dithered quantization [Fougstedt et al., 2017a]
- ISI minimization [Fougstedt et al., 2017b]

Problem: Truncation Errors



$$h^{(1)} = h^{(2)} = \cdots = h^{(M)}$$

FIR filter design methods:
- time sampling and truncation [Savory, 2008]
- frequency-domain sampling [Ip and Kahn, 2008]
- wavelets [Goldfarb and Li, 2009]
- least-squares [Eghbali et al., 2014], [Sheikh et al., 2016]



$$h^{(1)} * h^{(2)} * \cdots * h^{(M)}$$

- filter pair design [Zhu et al., 2009]

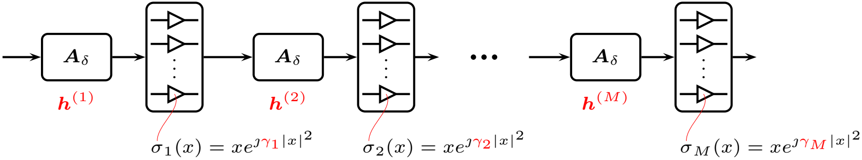
Filter coefficient quantization:
- dithered quantization [Fougstedt et al., 2017a]
- ISI minimization [Fougstedt et al., 2017b]

This work:
Optimize all M filters jointly

Learned Digital Backpropagation

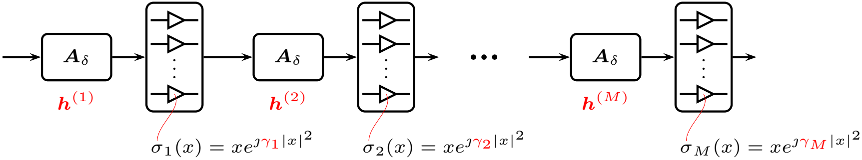
Learned Digital Backpropagation

TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:



Learned Digital Backpropagation

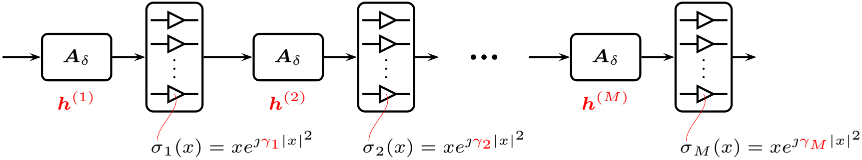
TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:



Not a standard black-box neural network as in, e.g., [Shen and Lau, 2011], [Jarajreh et al., 2015], [Giacoumidis et al., 2015], [Estarán et al., 2016], ...

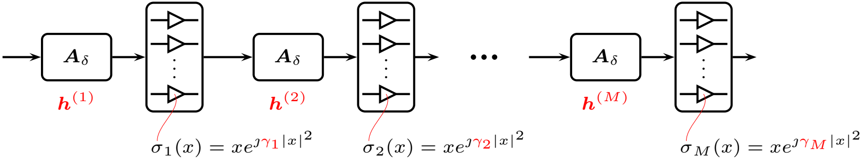
Learned Digital Backpropagation

TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:



Learned Digital Backpropagation

TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:



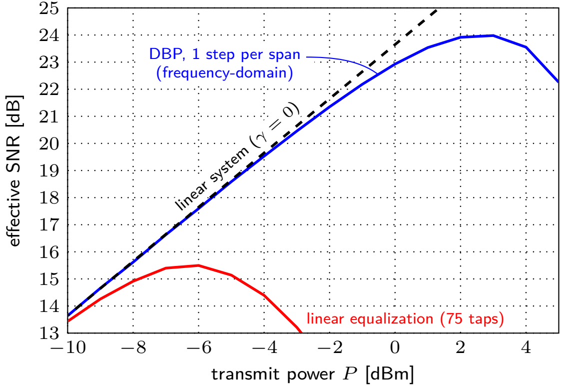
Deep learning of parameters $\theta = \{\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(M)}, \gamma_1, \dots, \gamma_M\}$

$$\min_\theta \sum_{i=1}^{N} \text{Loss}(f_\theta(\boldsymbol{y}^{(i)}), \boldsymbol{x}^{(i)}) \triangleq g(\theta) \qquad \text{using} \quad \theta_{k+1} = \theta_k - \lambda \nabla_\theta g(\theta_k)$$

mean squared error

Adam optimizer, fixed learning rate

Simulation Results

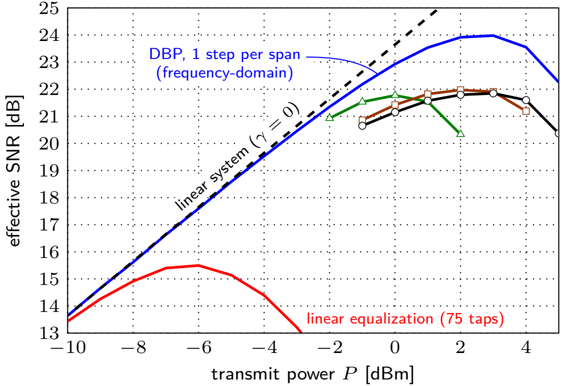


Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

$T \approx 75$ taps

Simulation Results

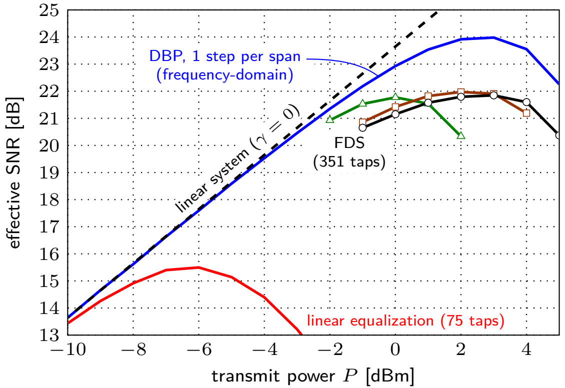


Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

$T \approx 75$ taps

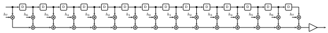
Simulation Results



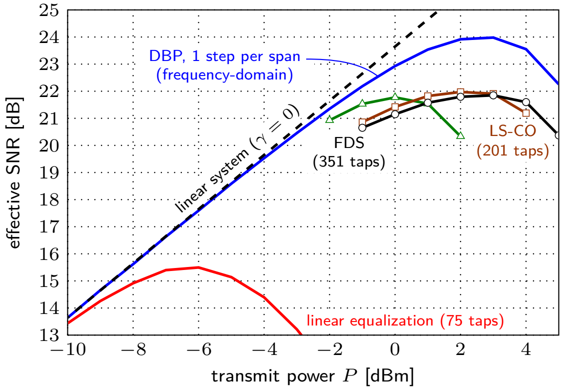
Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

$T \approx 75$ taps

FDS  · · · frequency-domain sampling (15 taps per step)

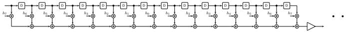
Simulation Results

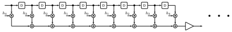


Parameters similar to [Ip and Kahn, 2008]:

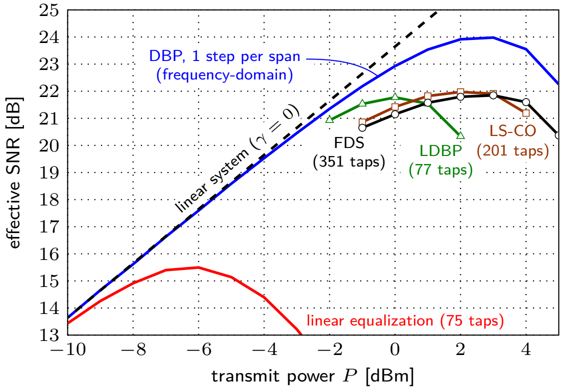
- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

$T \approx 75$ taps

FDS  frequency-domain sampling (15 taps per step)

LS-CO  constrained least-squares [Sheikh et al., 2016] (9 taps)

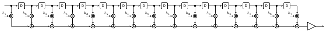
Simulation Results

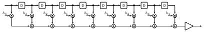


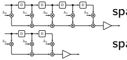
Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

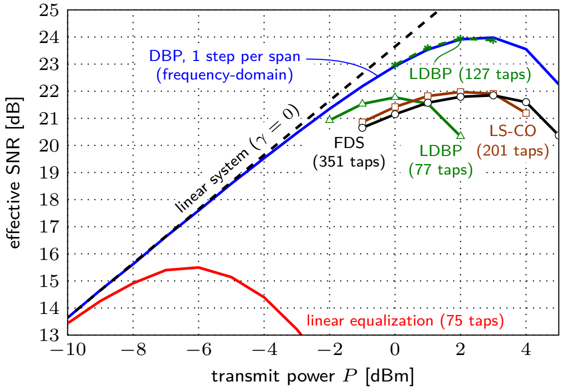
$T \approx 75$ taps

FDS  . . . frequency-domain sampling (15 taps per step)

LS-CO  . . . constrained least-squares [Sheikh et al., 2016] (9 taps)

LDBP  span 1, 3, . . . / span 2, 4, . . . learned digital backpropagation (alternate 5 and 3 taps and use different coefficients in all steps)

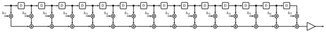
Simulation Results

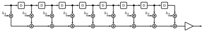


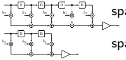
Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

$T \approx 75$ taps

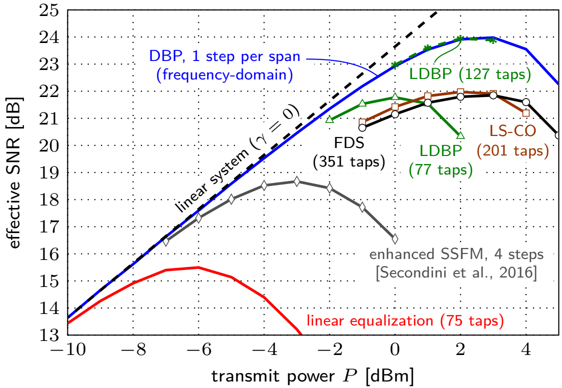
FDS  . . . frequency-domain sampling (15 taps per step)

LS-CO  . . . constrained least-squares [Sheikh et al., 2016] (9 taps)

LDBP 
span 1, 3, . . .
span 2, 4, . . .

learned digital backpropagation (alternate 5 and 3 taps and use different coefficients in all steps)

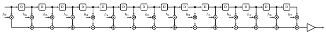
Simulation Results

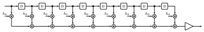


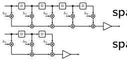
Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

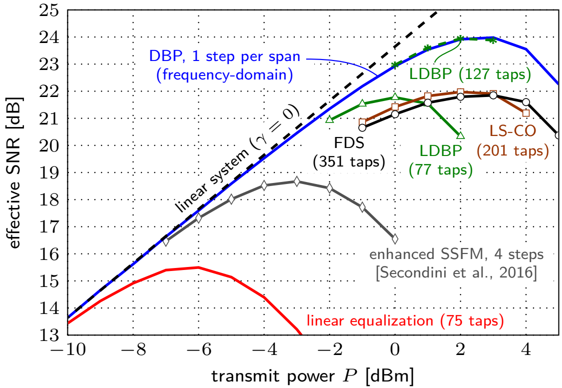
$T \approx 75$ taps

FDS  . . . frequency-domain sampling (15 taps per step)

LS-CO  . . . constrained least-squares [Sheikh et al., 2016] (9 taps)

LDBP  span 1, 3, . . . / span 2, 4, . . . learned digital backpropagation (alternate 5 and 3 taps and use different coefficients in all steps)

Simulation Results

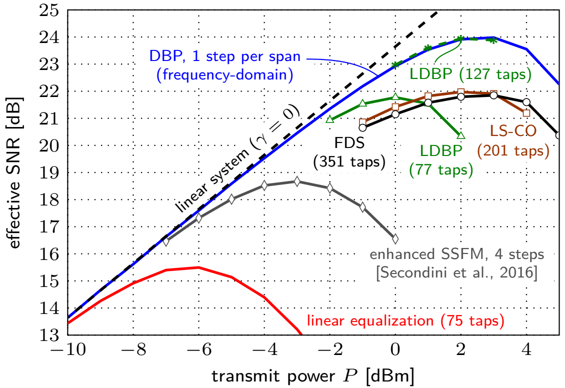


Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

$T \approx 75$ taps

Simulation Results



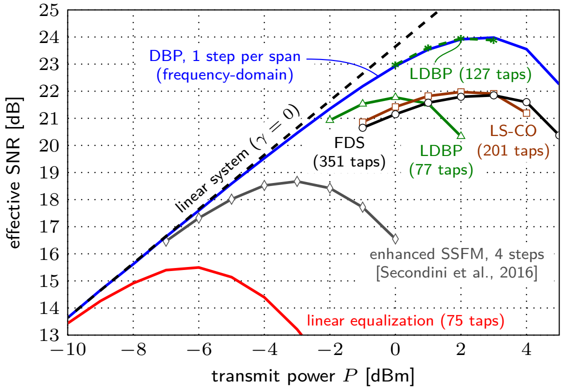
Parameters similar to [Ip and Kahn, 2008]:

- $25 \times 80\,\text{km}$ SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

$T \approx 75$ taps

- WDM: possibility to get 1–2 dB with even less complexity, e.g., keep large number of steps, but simplify the nonlinearity, use coarse quantization, etc.

Simulation Results



Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

$T \approx 75$ taps

- WDM: possibility to get 1–2 dB with even less complexity, e.g., keep large number of steps, but simplify the nonlinearity, use coarse quantization, etc.
- Higher baudrates (per 100 km SSFM): 20 Gbaud: ≈ 12 taps (✓ verified including finite precision effects), 30 Gbaud: ≈ 30 taps (preliminary results)

Conclusions

Conclusions

- Split-step Fourier method leads to a computation graph reminiscient of a deep feed-forward neural network
- Joint filter optimization can be solved by applying deep learning to significantly reduce the number of required filter taps

Conclusions

- Split-step Fourier method leads to a computation graph reminiscient of a deep feed-forward neural network
- Joint filter optimization can be solved by applying deep learning to significantly reduce the number of required filter taps

Thank you!

FORCE FIBER-OPTIC COMMUNICATIONS RESEARCH CENTER

Duke UNIVERSITY

References I

Du, L. B. and Lowery, A. J. (2010).
Improved single channel backpropagation for intra-channel fiber nonlinearity compensation in long-haul optical communication systems.
Opt. Express, 18(16):17075–17088.

Eghbali, A., Johansson, H., Gustafsson, O., and Savory, S. J. (2014).
Optimal least-squares FIR digital filters for compensation of chromatic dispersion in digital coherent optical receivers.
J. Lightw. Technol., 32(8):1449–1456.

Essiambre, R.-J. and Winzer, P. J. (2005).
Fibre nonlinearities in electronically pre-distorted transmission.
In *Proc. European Conf. Optical Communication (ECOC)*, Glasgow, UK.

Estarán, J., Rios-Müller, R., Mestre, M. A., Jorge, F., Mardoyan, H., Konczykowska, A., Dupuy, J. Y., and Bigo, S. (2016).
Artificial neural networks for linear and non-linear impairment mitigation in high-baudrate IM/DD systems.
In *Proc. European Conf. Optical Communication (ECOC)*, Düsseldorf, Germany.

Fougstedt, C., Mazur, M., Svensson, L., Eliasson, H., Karlsson, M., and Larsson-Edefors, P. (2017a).
Time-domain digital back propagation: Algorithm and finite-precision implementation aspects.
In *Proc. Optical Fiber Communication Conf. (OFC)*, Los Angeles, CA.

Fougstedt, C., Svensson, L., Mazur, M., Karlsson, M., and Larsson-Edefors, P. (2017b).
Finite-precision optimization of time-domain digital back propagation by inter-symbol interference minimization.
In *Proc. European Conf. Optical Communication (ECOC)*, Gothenburg, Sweden.

References II

Giacoumidis, E., Le, S. T., Ghanbarisabagh, M., McCarthy, M., Aldaya, I., Mhatli, S., Jarajreh, M. A., Haigh, P. A., Doran, N. J., Ellis, A. D., and Eggleton, B. J. (2015).
Fiber nonlinearity-induced penalty reduction in CO-OFDM by ANN-based nonlinear equalization.
Opt. Lett., 40(21):5113–5116.

Goldfarb, G. and Li, G. (2009).
Efficient backward-propagation using wavelet- based filtering for fiber backward-propagation.
Opt. Express, 17(11):814–816.

Ip, E. and Kahn, J. M. (2008).
Compensation of dispersion and nonlinear impairments using digital backpropagation.
J. Lightw. Technol., 26:3416–3425.

Jarajreh, A. M., Giacoumidis, E., Aldaya, I., Le, S. T., Tsokanos, A., Ghassemlooy, Z., and Doran, N. J. (2015).
Artificial neural network nonlinear equalizer for coherent optical OFDM.
IEEE Photon. Technol. Lett., 27(4):387–390.

Li, L., Tao, Z., Dou, L., Yan, W., Oda, S., Tanimura, T., Hoshida, T., and Rasmussen, J. C. (2011).
Implementation efficient nonlinear equalizer based on correlated digital backpropagation.
In *Proc. Optical Fiber Communication Conf. (OFC)*, page OWW3, Los Angeles, CA.

Li, X., Chen, X., Goldfarb, G., Mateo, E., Kim, I., Yaman, F., and Li, G. (2008).
Electronic post-compensation of WDM transmission impairments using coherent detection and digital signal processing.
Opt. Express, 16(2):880–888.

References III

Napoli, A., Maalej, Z., Sleiffer, V. A. J. M., Kuschnerov, M., Rafique, D., Timmers, E., Spinnler, B., Rahman, T., Coelho, L. D., and Hanik, N. (2014).
Reduced complexity digital back-propagation methods for optical communication systems.
J. Lightw. Technol., 32(7):1351–1362.

Paré, C., Villeneuve, A., Bélanger, P.-A. A., and Doran, N. J. (1996).
Compensating for dispersion and the nonlinear Kerr effect without phase conjugation.
Optics Letters, 21(7):459–461.

Rafique, D., Zhao, J., and Ellis, A. D. (2011).
Digital back-propagation for spectrally efficient wdm 112 gbit/s pm m-ary qam transmission.
Opt. Express, 19(6):5219–5224.

Roberts, K., Li, C., Strawczynski, L., O'Sullivan, M., and Hardcastle, I. (2006).
Electronic precompensation of optical nonlinearity.
IEEE Photon. Technol. Lett., 18(2):403–405.

Savory, S. J. (2008).
Digital filters for coherent optical receivers.
Opt. Express, 16(2):804–817.

Secondini, M., Rommel, S., Meloni, G., Fresi, F., Forestieri, E., and Poti, L. (2016).
Single-step digital backpropagation for nonlinearity mitigation.
Photon. Netw. Commun., 31(3):493–502.

Sheikh, A., Fougstedt, C., Graell i Amat, A., Johannisson, P., Larsson-Edefors, P., and Karlsson, M. (2016).
Dispersion compensation FIR filter with improved robustness to coefficient quantization errors.
J. Lightw. Technol., 34(22):5110–5117.

References IV

Shen, T. and Lau, A. (2011).
Fiber nonlinearity compensation using extreme learning machine for dsp-based coherent communication systems.
Optoe Electronics and Communications ..., (3):816–817.

Yan, W., Tao, Z., Dou, L., Li, L., Oda, S., Tanimura, T., Hoshida, T., and Rasmussen, J. C. (2011).
Low complexity digital perturbation back-propagation.
In *Proc. European Conf. Optical Communication (ECOC)*, page Tu.3.A.2, Geneva, Switzerland.

Zhu, L., Li, X., Mateo, E., and Li, G. (2009).
Complementary FIR filter pair for distributed impairment compensation of WDM fiber transmission.
IEEE Photon. Technol. Lett., 21(5):292–294.