Accelerated Methods for Solving Optimal Power Flow Problems

Priyank Srivastava
Massachusetts Institute of Technology

Collaborative work with: A. Parashar, R. Haider, A. M. Annaswamy (MIT)
G. Ferro (Univ. of Genoa), A. Chakraborty, B. Dey (Siemens)

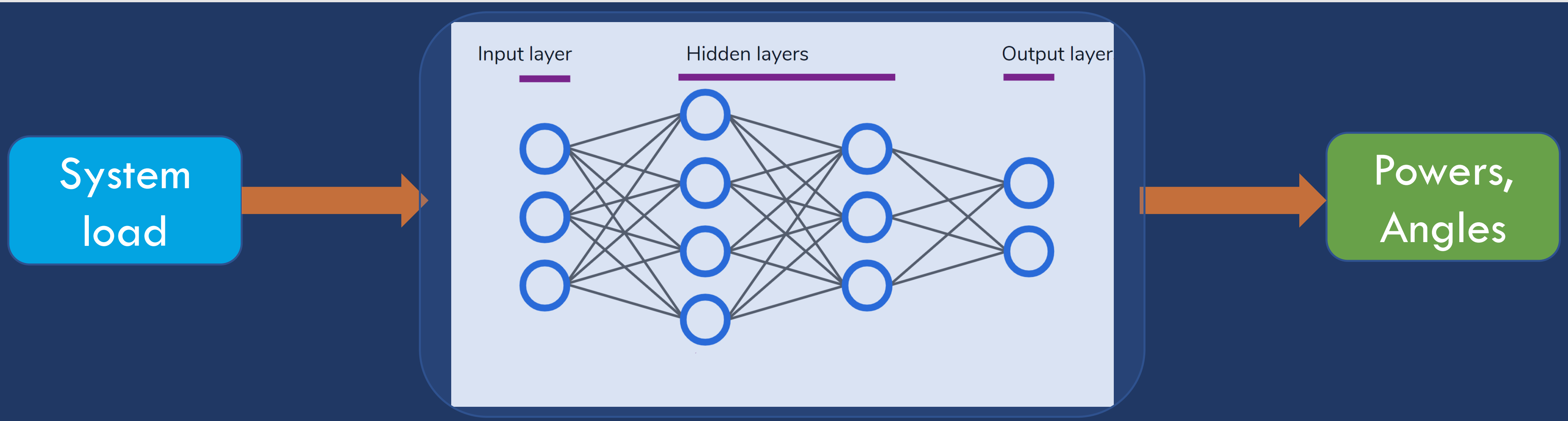
NREL Workshop on Autonomous Energy Systems
Golden, CO

July 15, 2022

Ubiquity of Distributed Energy Resources

- Increased volatility and uncertainty
- Optimal Power Flow (OPF) problem needs to be solved more often
- Curse of dimensionality due to increasing DER penetration
- Need for fast solvers

Overall Idea



Key Challenges

- Training the neural network to handle constraints
- What if there is a need to re-train?

Outline

- Learning-based constrained optimization method
- Capable of adjusting weights quickly with changes in the underlying structure
- Anytime satisfaction of constraints
- Simulation results on realistic power system models

Learning-Based Constrained Optimization Problem Solver

Partition	Partition the decision variables into dependent and independent
Train	Train NN to predict only independent variables with soft penalty
	Reduces the dimension of the NN
Complete	Compute the dependent variables from equality constraints
	Automatically satisfies the equality constraints
Verify	Ensure feasibility with respect to the inequality constraints
	Post-processing if needed

[1] X .Pan, T. Zhao, M. Chen and S. Zhang, "DeepOPF: A Deep Neural Network Approach for Security-Constrained DC Optimal Power Flow," in *IEEE Transactions on Power Systems*

DC-OPF using Neural Networks

$$\min_{P_g,\theta} \quad \sum_{i=1}^{n} f_i(P_{gi})$$ Generation Cost

$$\text{s.t.} \quad \underline{P_g} \leq P_g \leq \overline{P_g}$$ Capacity Constraints

$$B\theta = P_g - P_l$$ Power Balance

$$b_{ij}(\theta_i - \theta_j) \leq \overline{P}_{ij} \quad \forall (i,j)$$ Flow limits

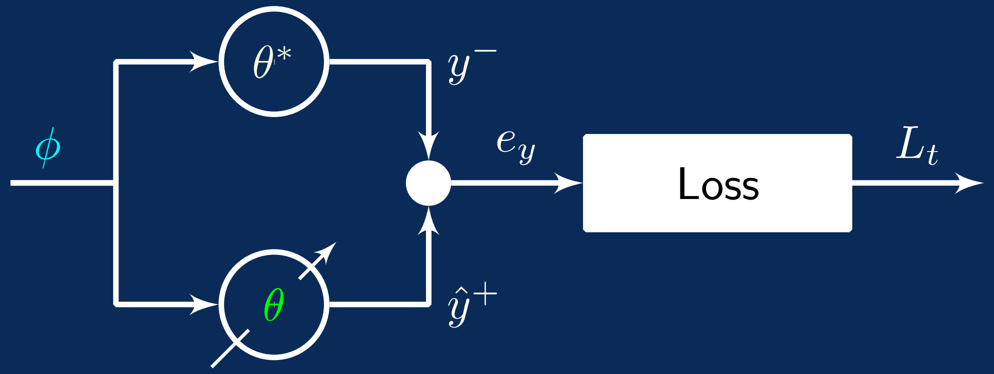
- NN is trained to output the generator powers
- Output layer consists of the sigmoidal function to handle capacity constraints
- Loss function consists of error and soft loss pertaining to the line flows
- Line angles are computed using the power flow equation
- If needed, post-processing is performed

[1] X .Pan, T. Zhao, M. Chen and S. Zhang, "DeepOPF: A Deep Neural Network Approach for Security-Constrained DC Optimal Power Flow," in *IEEE Transactions on Power Systems*

Accelerated Training

- The underlying structure of the network might change often
- Model needs to be re-trained
- To ensure faster training, there is a need to go beyond vanilla gradient-descent
- The size of NN prohibits the use of Newton-like methods as computing the inverse of the Hessian is expensive
- Momentum-based methods which rely on only the gradient information need to be explored

Linear Regression Models



Plant: $y = \phi^T \theta^*$

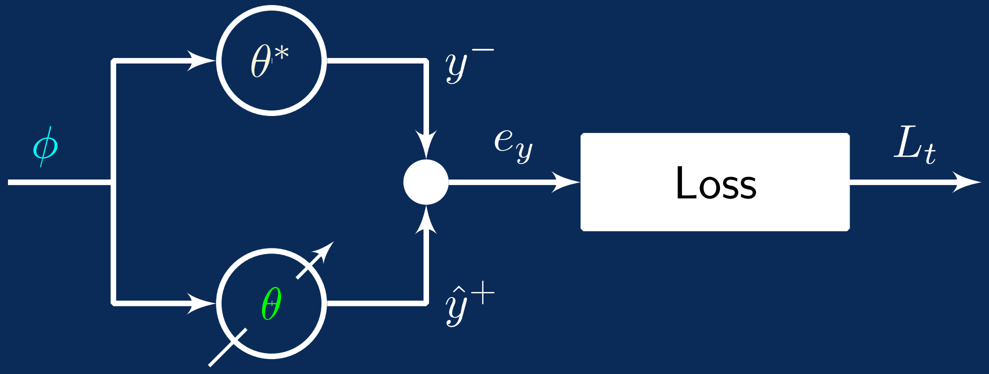
Estimator: $\hat{y} = \phi^T \theta$

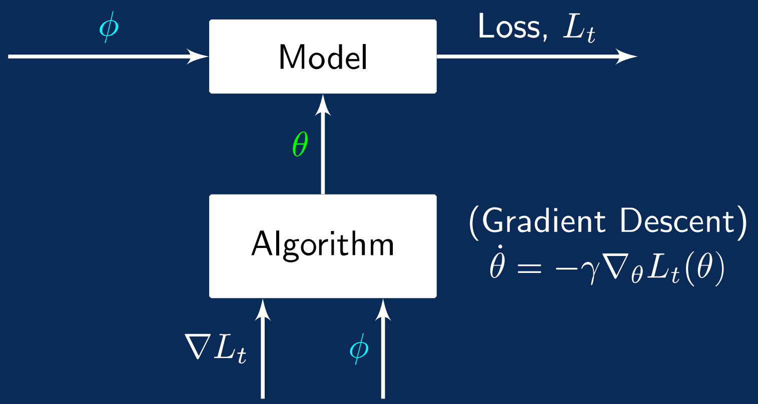
Loss:

$$L_t(\theta) = \frac{1}{2} \| \phi^T \theta - y \|_2^2$$

(any convex function of θ)

Linear Regression Models





Plant: $y = \phi^T \theta^*$

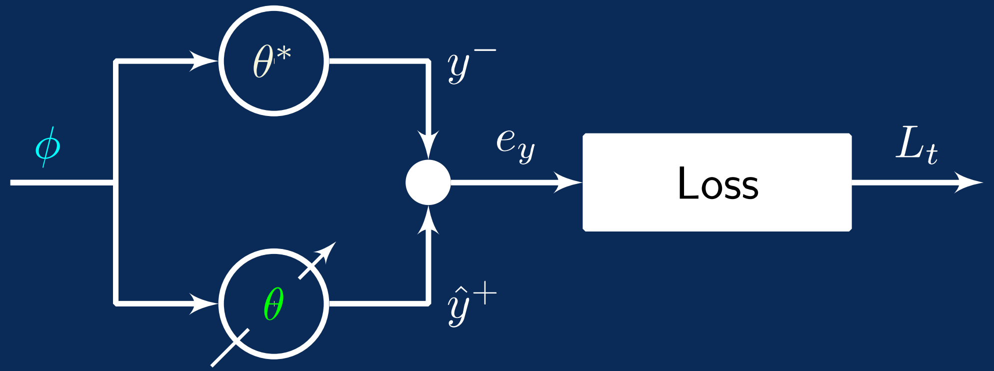
Estimator: $\hat{y} = \phi^T \theta$

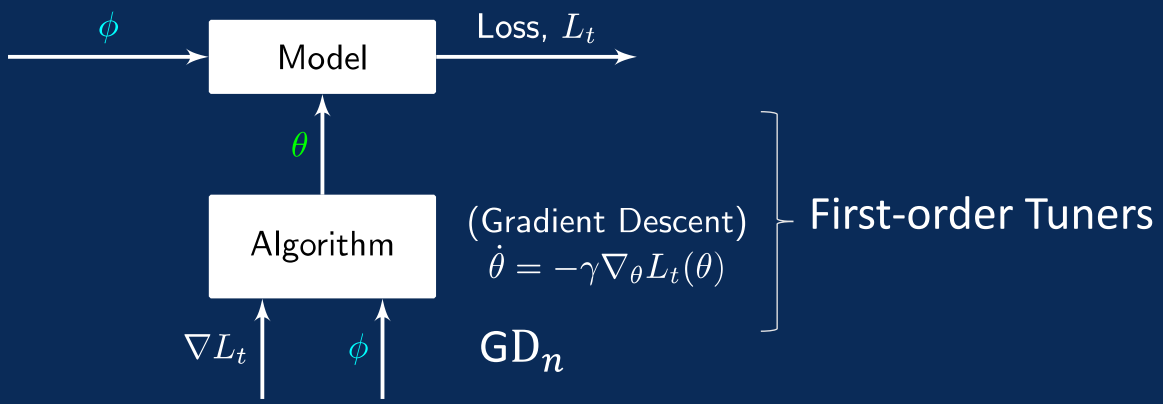
Loss:

$$L_t(\theta) = \frac{1}{2} \|\phi^T \theta - y\|_2^2$$

(any convex function of θ)

Linear Regression Models





Plant: $y = \phi^T \theta^*$

Estimator: $\hat{y} = \phi^T \theta$

Loss:

$$L_t(\theta) = \frac{1}{2}\|\phi^T\theta - y\|_2^2$$

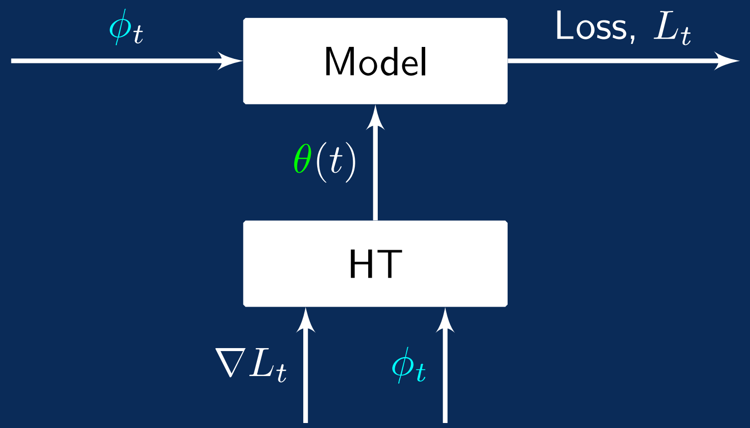
(any convex function of θ)

Gradient Descent, Normalized ($\mathrm{GD_n}$):

$$\dot{\theta}(t) = -\frac{\Gamma}{\mathcal{N}_t}\nabla_\theta L_t(\theta)$$

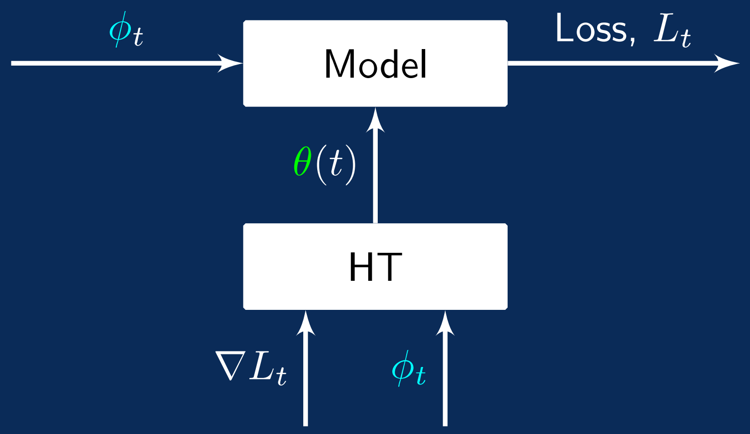
Γ : learning rate > 0; $\quad \mathcal{N}_t = 1 + \|\phi\|_2^2$: Normalization

Accelerated Performance with a High-order Tuner



[1] J. E. Gaudio, A. M. Annaswamy, M. A. Bolender, E. Lavretsky, and T. E. Gibson (2020). "A Class of High Order Tuners for Adaptive Systems". *IEEE Control Systems Letters*.

Accelerated Performance with a High-order Tuner



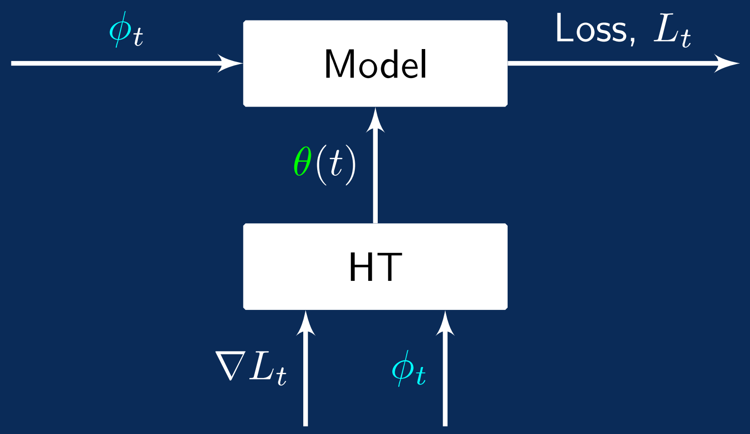
High-Order Tuner (HT)[1]:

$$\dot{\vartheta}(t) = -\frac{\gamma}{\mathcal{N}_t}\nabla L_t(\theta(t)), \qquad \mathcal{N}_t = 1 + \|\phi_t\|^2$$

$$\dot{\theta}(t) = -\beta(\theta(t) - \vartheta(t)).$$

[1] J. E. Gaudio, A. M. Annaswamy, M. A. Bolender, E. Lavretsky, and T. E. Gibson (2020). “A Class of High Order Tuners for Adaptive Systems”. *IEEE Control Systems Letters*.

Accelerated Performance with a High-order Tuner



High-Order Tuner (HT)[1]:

$$\dot{\vartheta}(t) = -\frac{\gamma}{\mathcal{N}_t}\nabla L_t(\theta(t)), \qquad \mathcal{N}_t = 1 + \|\phi_t\|^2$$

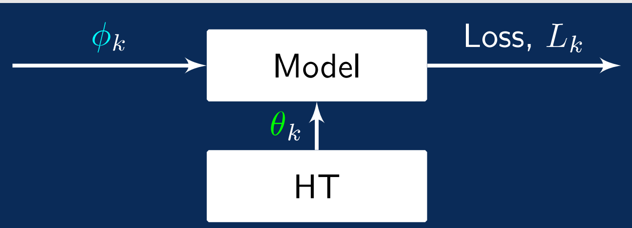
$$\dot{\theta}(t) = -\beta(\theta(t) - \vartheta(t)).$$

Theorem: All solutions are globally bounded, with a Lyapunov function

$$V = \frac{1}{\gamma}\|\vartheta - \theta^*\|^2 + \frac{1}{\gamma}\|\theta - \vartheta\|^2$$

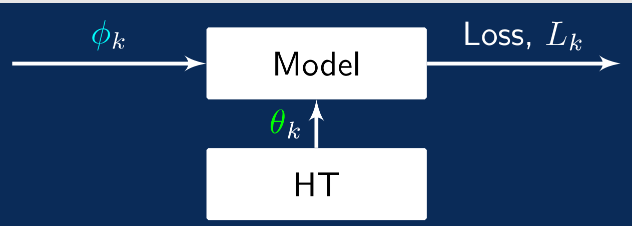
[1] J. E. Gaudio, A. M. Annaswamy, M. A. Bolender, E. Lavretsky, and T. E. Gibson (2020). "A Class of High Order Tuners for Adaptive Systems". *IEEE Control Systems Letters*.

Accelerated Performance (discrete-time)



[1] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

Accelerated Performance (discrete-time)



Discrete and continuous High-Order Tuner[1]:

Discrete

$$\bar{\theta}_k = \theta_k - \gamma\beta\frac{\nabla L_k(\theta_k)}{\mathcal{N}_k}, \qquad \mathcal{N}_k = 1 + \|\phi_k\|^2,$$

$$\theta_{k+1} = \bar{\theta}_k - \beta(\bar{\theta}_k - \vartheta_k),$$

$$\vartheta_{k+1} = \vartheta_k - \gamma\frac{\nabla L_k(\theta_{k+1})}{\mathcal{N}_k}$$

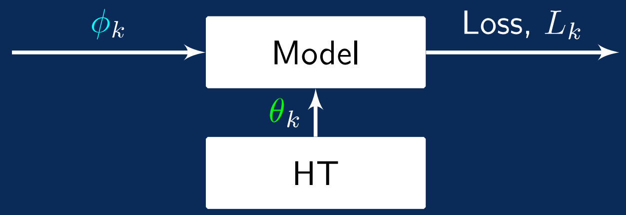
Continuous

$$\dot{\vartheta} = -\frac{\gamma}{\mathcal{N}_t}\nabla_\theta L_t(\theta),$$

$$\dot{\theta} = -\beta(\theta - \vartheta)$$

[1] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

Accelerated Performance (discrete-time)



Discrete and continuous High-Order Tuner[1]:

Discrete

$$\bar{\theta}_k = \theta_k - \gamma\beta\frac{\nabla L_k(\theta_k)}{\mathcal{N}_k}, \qquad \mathcal{N}_k = 1 + \|\phi_k\|^2,$$

$$\theta_{k+1} = \bar{\theta}_k - \beta(\bar{\theta}_k - \vartheta_k),$$

$$\vartheta_{k+1} = \vartheta_k - \gamma\frac{\nabla L_k(\theta_{k+1})}{\mathcal{N}_k}$$

Continuous

$$\dot{\vartheta} = -\frac{\gamma}{\mathcal{N}_t}\nabla_\theta L_t(\theta),$$

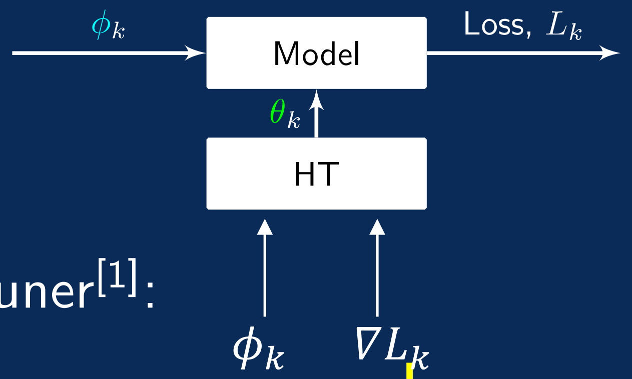
$$\dot{\theta} = -\beta(\theta - \vartheta)$$

Theorem: All solutions are globally bounded, with a Lyapunov function

$$V_k = \frac{1}{\gamma}\|\vartheta_k - \theta^*\|^2 + \frac{1}{\gamma}\|\theta_k - \vartheta_k\|^2$$

[1] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

Accelerated Performance (discrete-time)



Current Nnet algorithms:

$$\theta_{k+1} = \theta_k - \gamma_k \nabla_\theta L(\theta_k)$$

Discrete and continuous High-Order Tuner[1]:

Proposed Discrete HT

$$\bar{\theta}_k = \theta_k - \gamma\beta \frac{\nabla L_k(\theta_k)}{\mathcal{N}_k}, \qquad \mathcal{N}_k = 1 + \|\phi_k\|^2,$$

$$\theta_{k+1} = \bar{\theta}_k - \beta(\bar{\theta}_k - \vartheta_k),$$

$$\vartheta_{k+1} = \vartheta_k - \gamma \frac{\nabla L_k(\theta_{k+1})}{\mathcal{N}_k}$$

Proposed Continuous HT

$$\dot{\vartheta} = -\frac{\gamma}{\mathcal{N}_t} \nabla_\theta L_t(\theta),$$

$$\dot{\theta} = -\beta(\theta - \vartheta)$$

Theorem: All solutions are globally bounded, with a Lyapunov function

$$V_k = \frac{1}{\gamma}\|\vartheta_k - \theta^*\|^2 + \frac{1}{\gamma}\|\theta_k - \vartheta_k\|^2$$

[1] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

Non-Asymptotic Tools

Adaptive Control tools: Convergence of errors to zero.

- ▷ Asymptotic Tools: $f(\theta_k) - f(\theta^*) \to 0$ as $k \to \infty$

[1] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

[2] Y. Nesterov (2018). *Lectures on Convex Optimization*. Springer.

Non-Asymptotic Tools

Adaptive Control tools: Convergence of errors to zero.

- Asymptotic Tools: $f(\theta_k) - f(\theta^*) \to 0$ as $k \to \infty$
- Non-asymptotic tools:
  - GD: $f(x_k) - f(x^*) \leq \epsilon$ if $k \geq \mathcal{O}(1/\epsilon)$
  - Nesterov[2]: $f(x_k) - f(x^*) \leq \epsilon$ if $k \geq \mathcal{O}(1/\sqrt{\epsilon})$

[1] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, “Accelerated Learning with Robustness to Adversarial Regressors,” 3rd L4DC Conference, 2021.

[2] Y. Nesterov (2018). *Lectures on Convex Optimization*. Springer.

Non-Asymptotic Tools

Adaptive Control tools: Convergence of errors to zero.

- ▷ Asymptotic Tools: $f(\theta_k) - f(\theta^*) \to 0$ as $k \to \infty$
- ▷ Non-asymptotic tools:
  - ▷ GD: $f(x_k) - f(x^*) \leq \epsilon$ if $k \geq \mathcal{O}(1/\epsilon)$
  - ▷ Nesterov[2]: $f(x_k) - f(x^*) \leq \epsilon$ if $k \geq \mathcal{O}(1/\sqrt{\epsilon})$

Theorem : HT guarantees that[1]

$$L_k(\theta_k) - L_k(\theta^*) \leq \epsilon \text{ for } k \geq \mathcal{O}(1/\sqrt{\epsilon} \cdot \log(1/\epsilon))$$

$$f_k = \bar{L}\left(\frac{L_k}{N_k} + g_k\right) \quad (g_k \text{ small; ensures strong convexity})$$

[1] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.
[2] Y. Nesterov (2018). *Lectures on Convex Optimization*. Springer.

Non-Asymptotic Tools

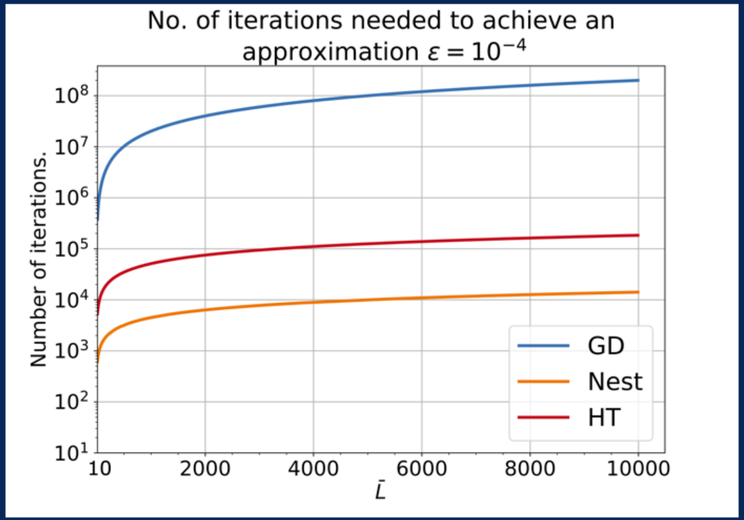
Adaptive Control tools: Convergence of errors to zero.

- Asymptotic Tools: $f(\theta_k) - f(\theta^*) \to 0$ as $k \to \infty$
- Non-asymptotic tools:
  - GD: $f(x_k) - f(x^*) \leq \epsilon$ if $k \geq \mathcal{O}(1/\epsilon)$
  - Nesterov[2]: $f(x_k) - f(x^*) \leq \epsilon$ if $k \geq \mathcal{O}(1/\sqrt{\epsilon})$

Theorem : HT guarantees that[1]

$$L_k(\theta_k) - L_k(\theta^*) \leq \epsilon \text{ for } k \geq \mathcal{O}(1/\sqrt{\epsilon} \cdot \log(1/\epsilon))$$

$$f_k = \bar{L}\left(\frac{L_k}{N_k} + g_k\right) \quad (g_k \text{ small; ensures strong convexity})$$



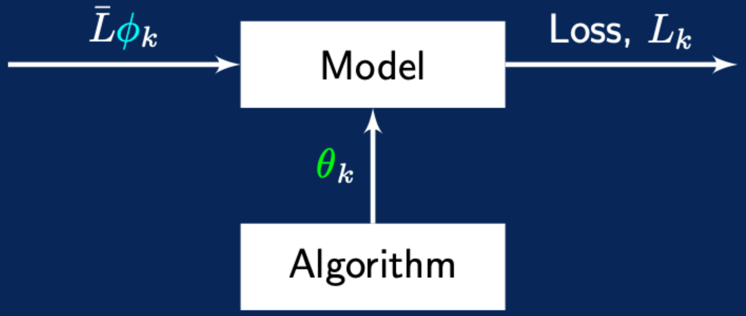
$\bar{L}$: Smoothness parameter.

[1] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

[2] Y. Nesterov (2018). *Lectures on Convex Optimization*. Springer.

Non-asymptotic Properties: Example 1

Modified Smooth-Hard Problem, with time-varying regressors



$$L_k(\theta_k) = \|\phi_k^T \theta_k\|^2 + B_k^T \theta_k$$

(quadratic, non-homogeneous, convex)

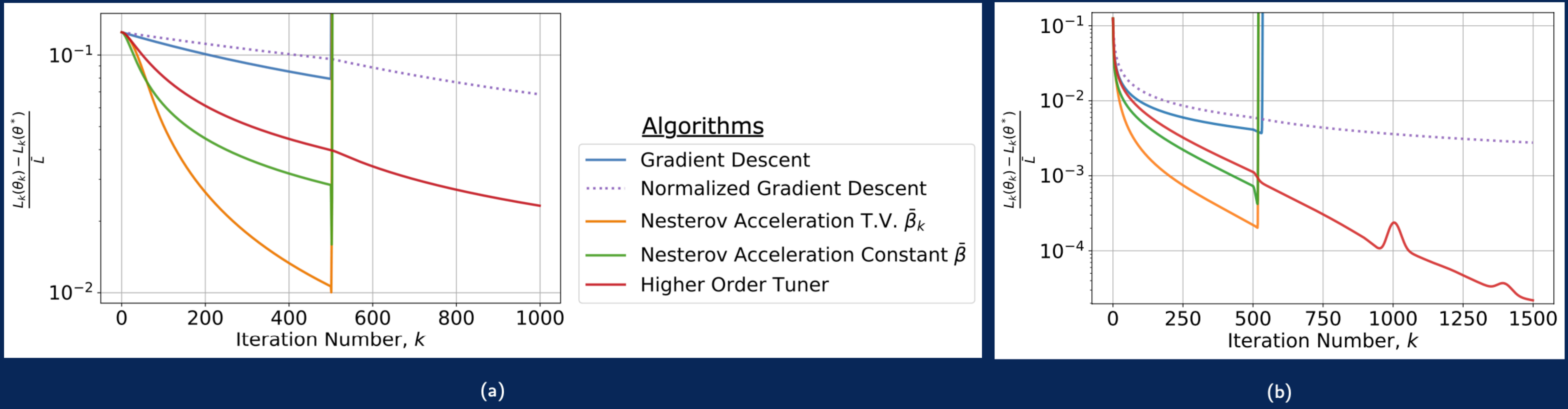


Figure: [1] (a) At iteration $k = 500$, step change in $\bar{L}$ from 2 to 8000. (b) At iteration $k = 500$, step change in $\bar{L}$, from 2 to 8.

[1] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, “Accelerated Learning with Robustness to Adversarial Regressors,” 3rd L4DC Conference, 2021.

Image Deblurring: Example 2

Blurring can be caused by many factors:

- Movement during the image capture process, by the camera or, when long exposure times are used, by the subject
- Out-of-focus optics, use of a wide-angle lens, atmospheric turbulence, or a short exposure time, which reduces the number of photons captured
- Scattered light distortion in confocal microscopy

- Model for blur[1]:

$$y = \phi^T \theta^* + n$$

[1] https://www.mathworks.com/help/images/image-deblurring.html

Non-asymptotic Properties: Example 2

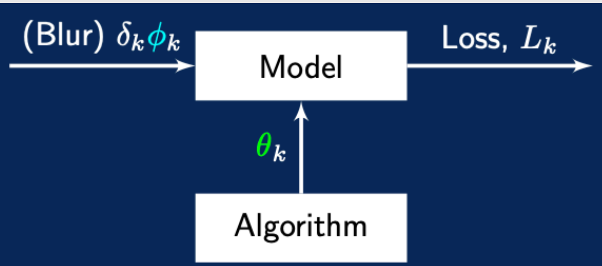


Image Deblurring Problem: True image: θ^*

Goal: Minimize Loss $L_k(\theta_k) = \frac{1}{2}\|\phi_k^T \theta_k - y_k\|^2$

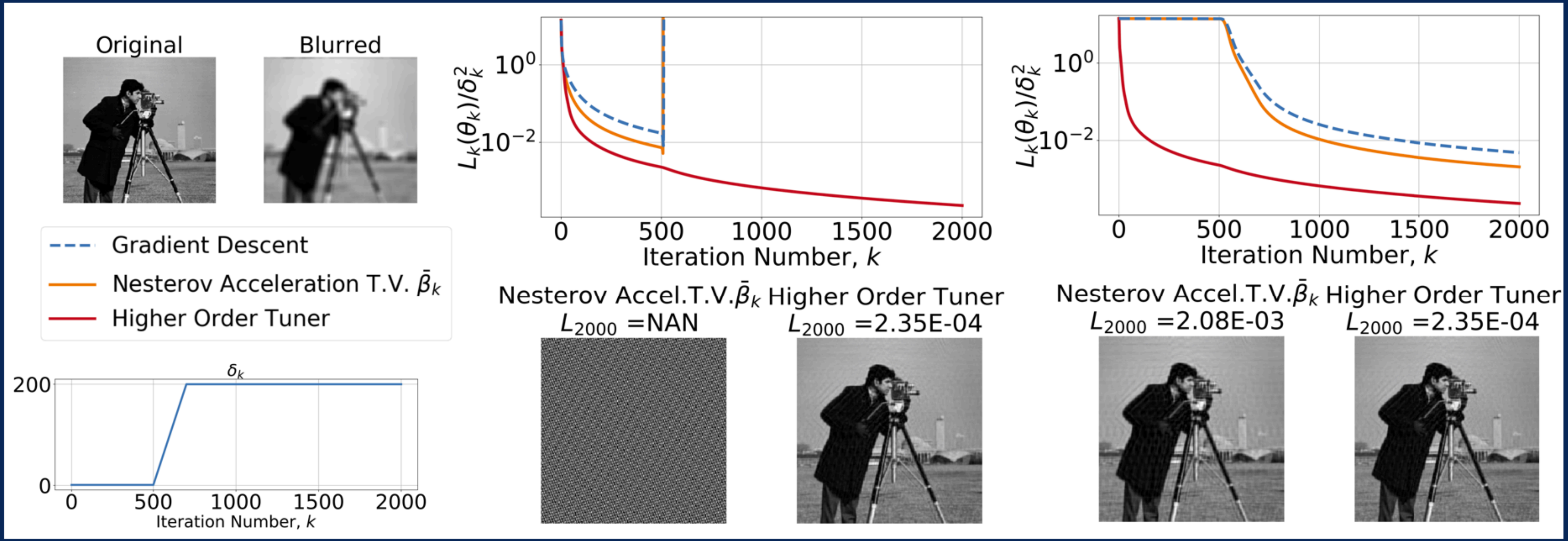


Figure: [1](Left) Original and blurred images; increase of δ_k from 1 to 200 in 200 iterations, starting at $k = 500$. (Center) Loss values and reconstructed images when only ϕ_0 is known *a priori*. (Right) Loss values and reconstructed images when all ϕ_k are known *a priori*.

[1] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

De-Blurring an Image with a Time-Varying Blur

[1] Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, *2*(1), 183-202.
[2] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

High-Order Tuner with General Convex Function

Lyapunov function: $V = \frac{1}{\gamma}\|\vartheta - \theta^*\|^2 + \frac{1}{\gamma}\|\theta - \vartheta\|^2$

↓

Stability: $\Delta V_k \leq 0$

If in addition f is also μ-strongly convex

Theorem Parameter convergence of HT with Hessian[1]

For a μ-strongly convex loss function $L_k(\cdot)$, Algorithm 2 with $0 < \beta < 1$ and $0 < \gamma \leq \frac{\beta(2-\beta)}{16+\beta+\mu}$ ensures that V is a Lyapunov function. Furthermore, for constant regressors, $V_k \leq \exp(-\mu C k)\, V_0$, where $C = \frac{\gamma\beta}{4\mathcal{N}}$.

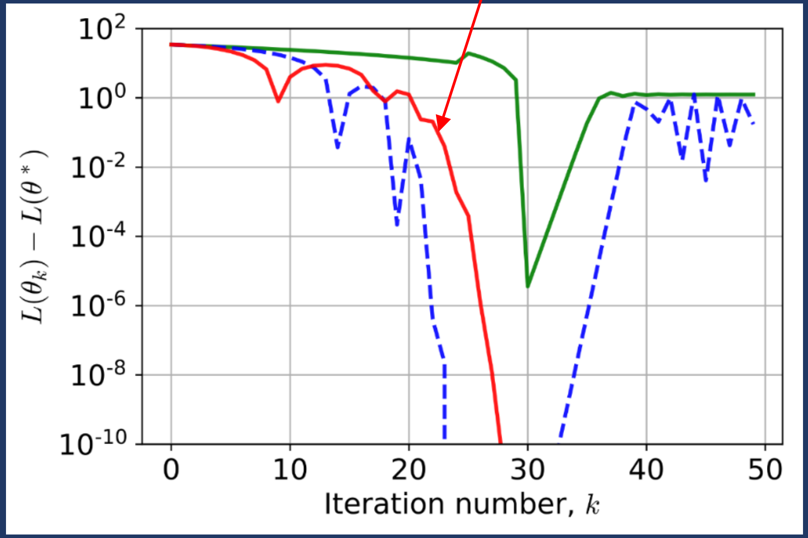
Algorithm HT for general convex function[1]

1: **Input:** initial conditions θ_0, ϑ_0, gains γ, β, μ
2: **for** $k = 0,1,2,\ldots$ **do**
3: **Receive** regressor ϕ_k
4: Compute $\nabla L(\theta_k)$ and let $\mathcal{N}_k = 1 + H_k$,
$\nabla f_k(\theta_k) = \frac{\nabla L_k(\theta_k)}{\mathcal{N}_k}$,
$\bar{\theta}_k = \theta_k - \gamma\beta\nabla f_k(\theta_k)$
5: $\theta_{k+1} \leftarrow \bar{\theta}_k - \beta(\bar{\theta}_k - \vartheta_k)$
6: Compute $\nabla L(\theta_{k+1})$ and let
$\nabla f_k(\theta_{k+1}) = \frac{\nabla L_k(\theta_{k+1})}{\mathcal{N}_k}$
7: $\vartheta_{k+1} \leftarrow \vartheta_k - \gamma \frac{\nabla f_k(\theta_{k+1})}{\mathcal{N}_k}$
8: **end for**

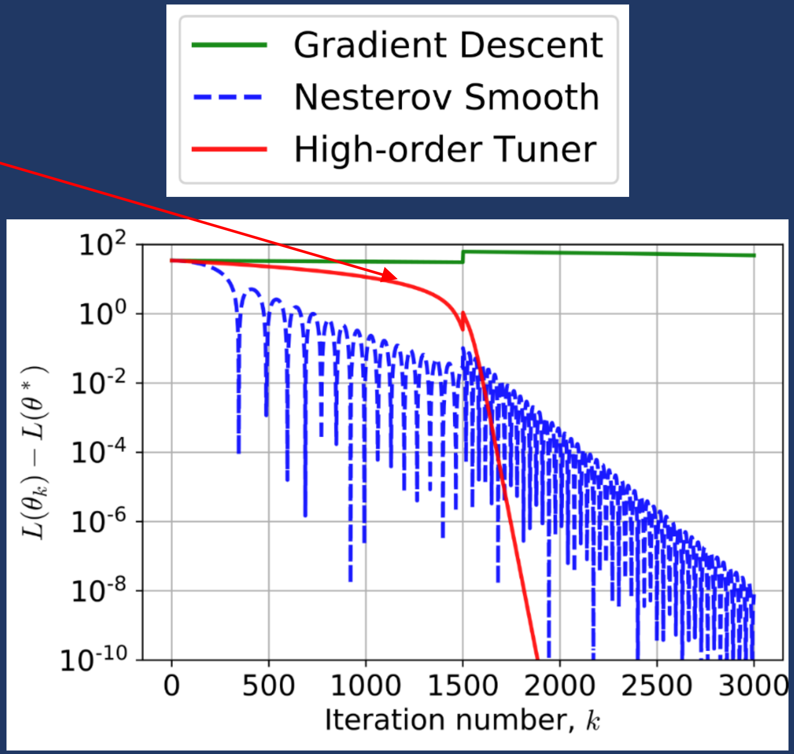
[1] Moreu, José M., and Anuradha M. Annaswamy. "A Stable High-order Tuner for General Convex Functions." *IEEE L-CSS, 2021.*

High-order Tuner for Convex and Dynamic Loss Functions

Stable performance with dynamics

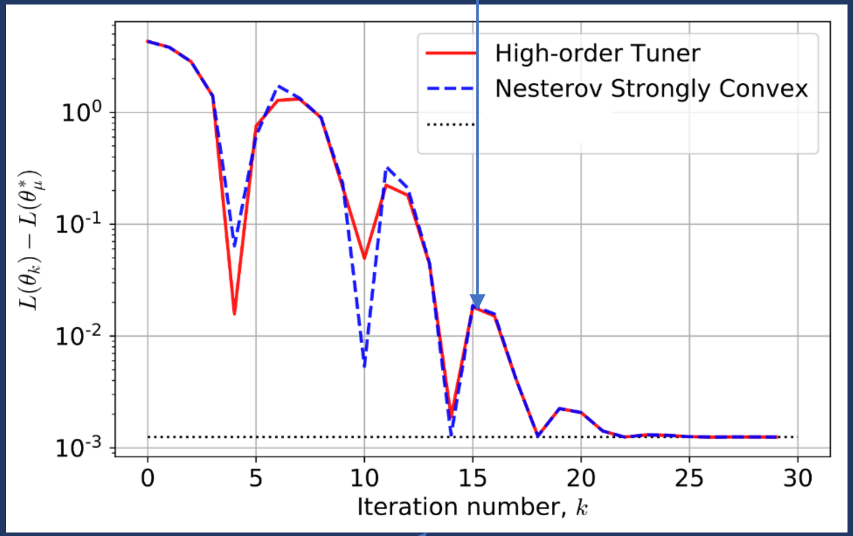


Step change in b_k from 7 to 14 at $k = 25$



Step change in b_k from 7 to 14 at $k = 1500$

Accelerated performance



No change in b_k

Loss: $L_k(\theta) = \log(a_k e^{b_k\theta} + a_k e^{-b_k\theta})$

Loss: $L_k(\theta) = \log(a_k e^{b_k\theta} + a_k e^{-b_k\theta}) + \frac{\mu}{2}\|\theta - \theta_0\|^2$

[1] Moreu, José M., and Anuradha M. Annaswamy. "A Stable High-order Tuner for General Convex Functions." *IEEE L-CSS, 2021.*
[2] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.
[3] Gaudio, Joseph E., et al. "A Class of High Order Tuners for Adaptive Systems." *IEEE L-CSS, 2020.*

Summary So Far

- Solving OPF problems via learning the mapping between loads and optimal generator set points
- A new algorithm utilizing HT with nice learning properties

Algorithm	Constant Regressor # Iterations	Time-Varying Regressor
Gradient Descent Normalized	$\mathcal{O}(1/\epsilon)$	Stable
Gradient Descent Fixed	$\mathcal{O}(1/\epsilon)$	Unstable
Nesterov Acceleration Varying	$\mathcal{O}(1/\sqrt{\epsilon})$	Unstable
Nesterov Acceleration Fixed	$\mathcal{O}(1/\sqrt{\epsilon} \cdot \log(1/\epsilon))$	Unstable
HT	$\mathcal{O}(1/\sqrt{\epsilon} \cdot \log(1/\epsilon))$	**Stable**

- Theoretical guarantees only for convex loss functions!! Could we say anything for general NN?

[1] J.E. Gaudio, A.M. Annaswamy, M.A. Bolender, E. Lavetsky, and T.E. Gibson, "Accelerated Learning with Robustness to Adversarial Regressors," 3rd L4DC Conference, 2021.

Case Studies on IEEE 30 and 300-Bus Systems

$$\min_{P_g,\theta} \sum_{i=1}^{n} f_i(P_{gi})$$

$$\text{s.t.} \quad \underline{P_g} \le P_g \le \overline{P_g}$$

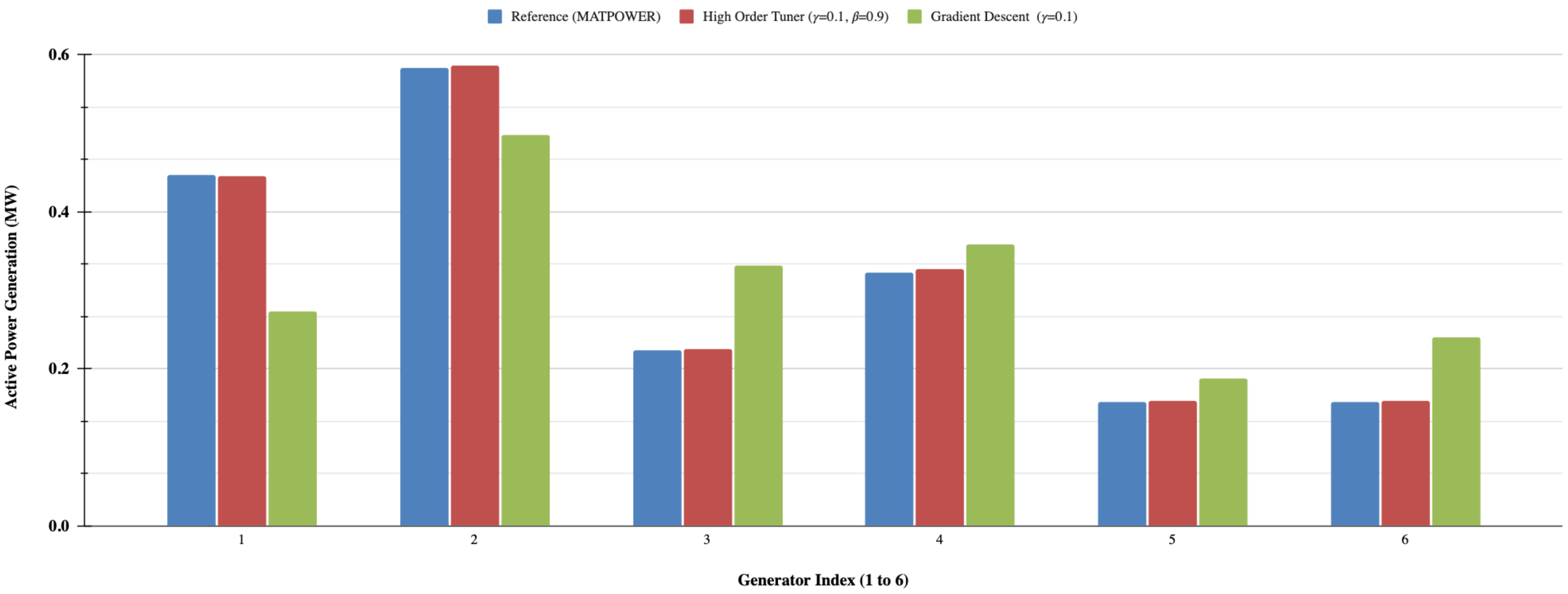
$$B\theta = P_g - P_l$$

$$b_{ij}(\theta_i - \theta_j) \le \overline{P}_{ij} \quad \forall (i,j)$$

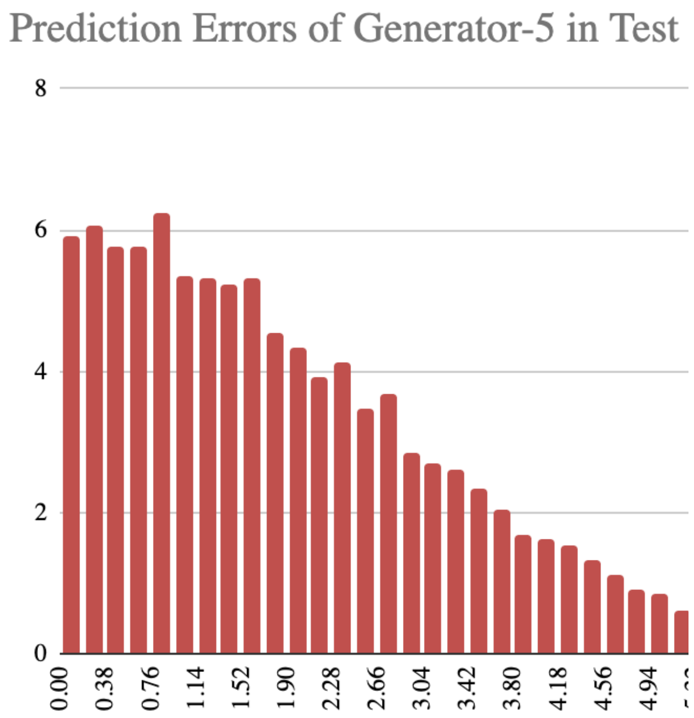
IEEE Case	N-load	N-hidden	N-neuron	N-gen	N-variables
30-Bus	20	2	16	6	36
300-Bus	199	6	128	69	369

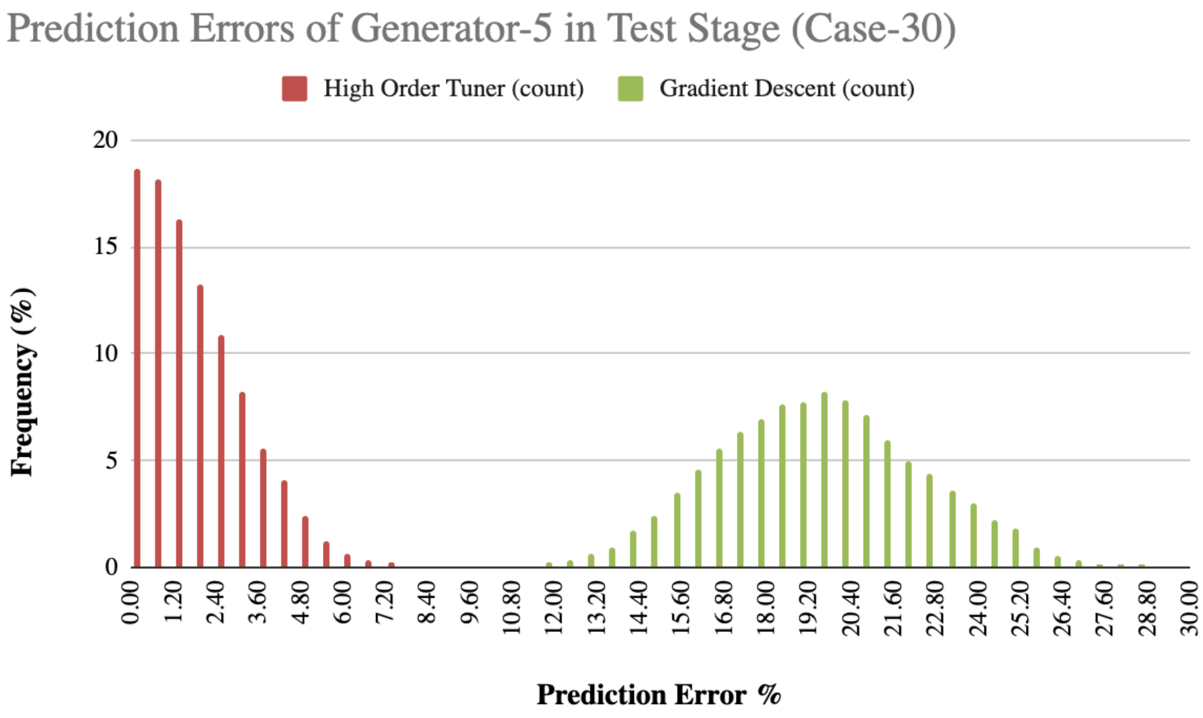
IEEE Case	Algorithm	% Feasible Solution	No. of epochs	Average Cost $/hr
30-Bus	Reference	-	-	565.3692
	HT	100	20	565.5349
	GD	100	20	582.3305
300-Bus	Reference	-	-	706322.2341
	HT	100	50	706612.225
	GD	100	50	706625.6001

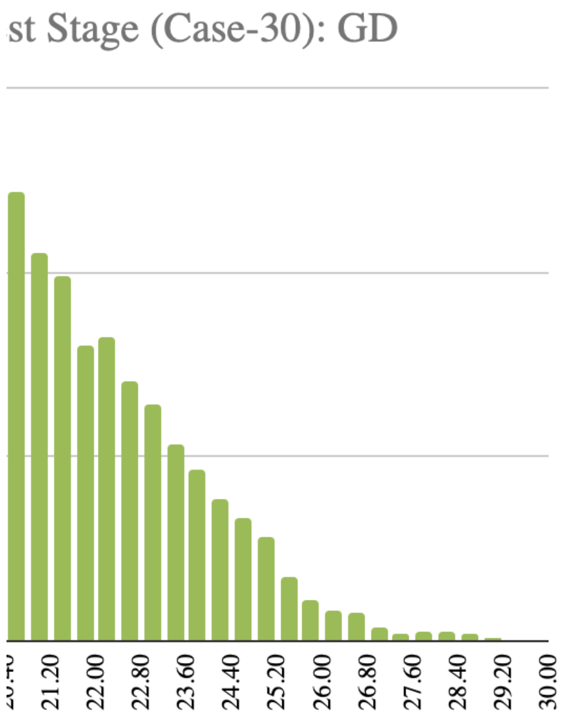
Generator Powers: IEEE 30-Bus



Prediction Error Comparison







Future Work

- Testing on larger Test Cases (e.g., New England ISO)
- Extend the proposed framework to AC-OPF and eventually to Mixed-Integer Problems
- Explore the training of NN to output LMPs
- Bridge the gap between theoretical results and simulation observations
- Investigate HT acceleration for certain classes of non-convex functions
- Accelerated HT techniques for constrained-optimization problems[1]

[1] A. Parashar, P. Srivastava, A.M. Annaswamy, B. Dey, and A. Chakraborty, "Accelerated Algorithms for a Class of Optimization Problems with Constraints," CDC 2022 (Submitted)

Thank You!