

Objective

Our main objective is to control the **repetitive process** employing an iterative learning algorithm with a **global update law**. This is to produce the **innately robust neurocontroller**. The main motivation here is the lack of an innate robustness in classic ILC systems that are based on local or non-local (but not global) update rules. Several training algorithms are to be scrutinized and recommendations are to be made.

Dynamic optimization

Reduce iteratively $\mathcal{E}_{\text{ANN}}(\mathbf{w}^{(1)}(k), \mathbf{w}^{(2)}(k))$
 $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}$

subject to: system nonlinearities,
 system uncertainties,
 system nonstationarity,
 actuator delay,
 $N = \text{const}$,
 constrained weight space,
 repetitive reference,
 repetitive disturbance.

Results

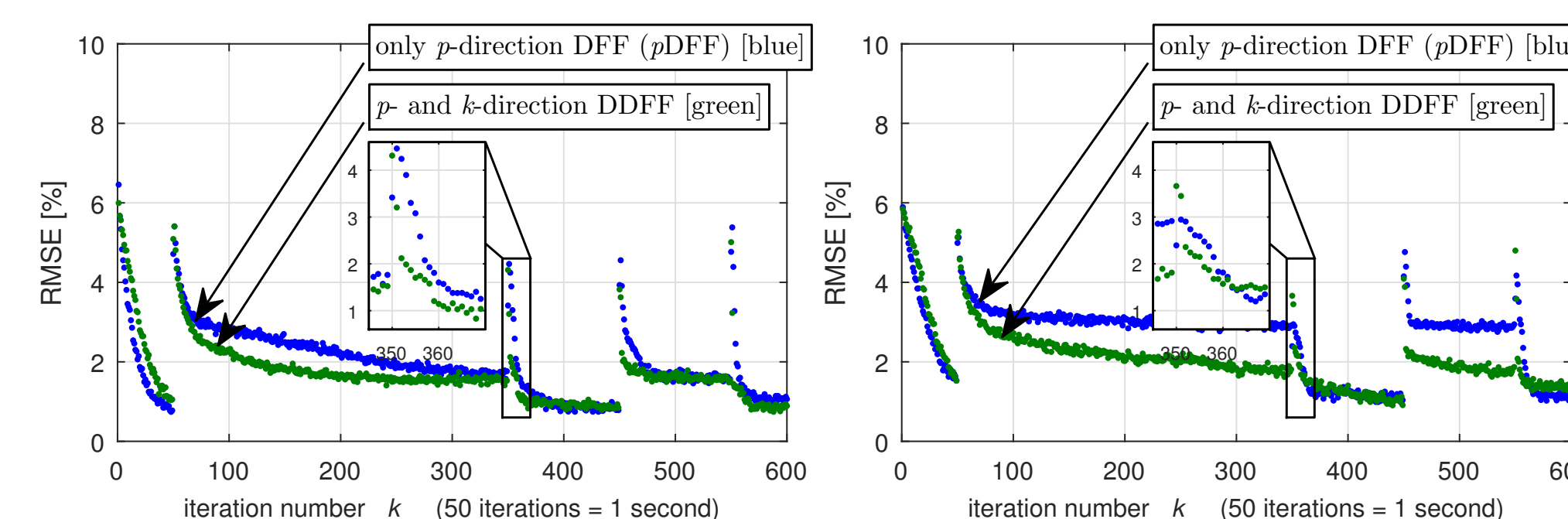


Figure 3: Comparison of the root mean square error (RMSE) decay rates for the Levenberg-Marquardt BP (trainlm) in the case of the classic disturbance feedforward and the novel disturbance dual feedforward ($N = 17$ and $N = 7$ hidden neurons).

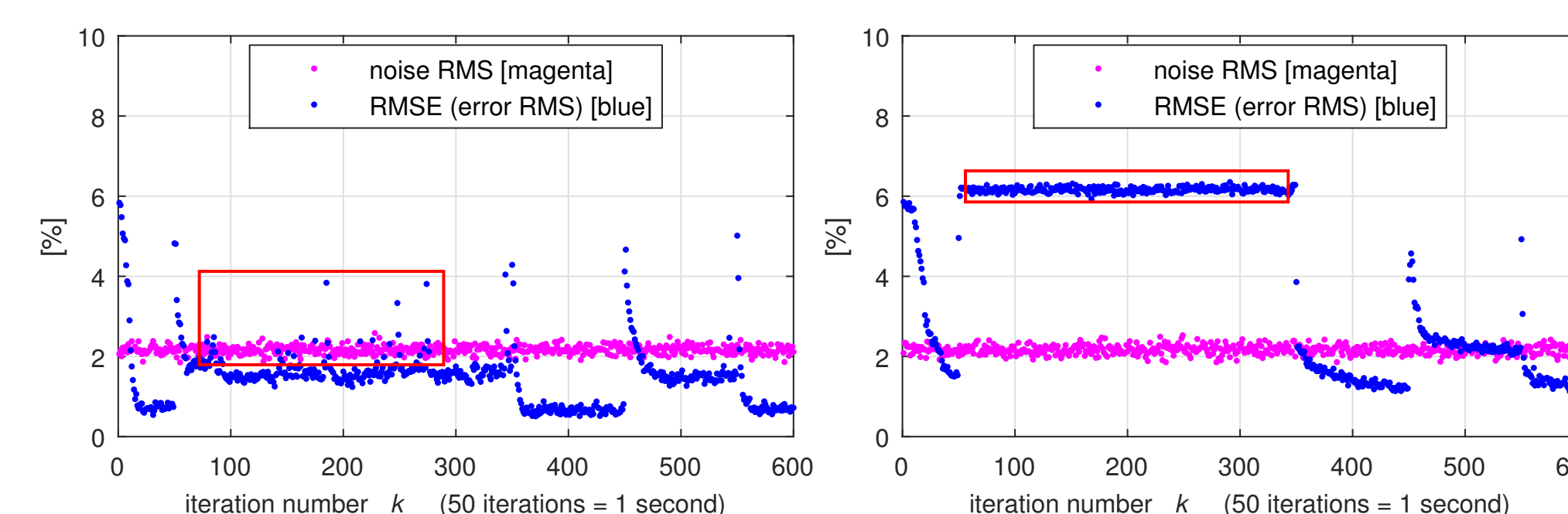


Figure 4: Undesired transient states (emphasized using red frames) caused by the Bayesian regularization mechanism in trainbr, and inconsistent behaviour of trainscg despite identical load variations at time instances 50s and 450s.

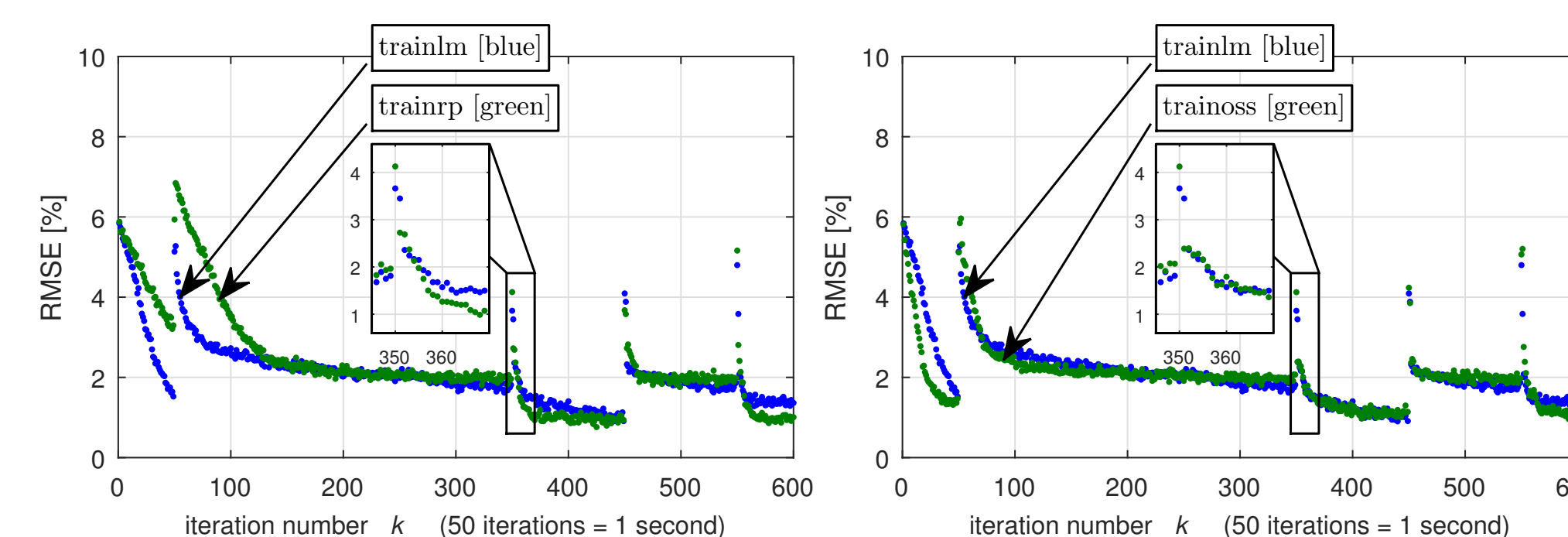


Figure 5: Root mean square error decay rate for Levenberg-Marquardt BP (trainlm), resilient BP (trainrp) and one-step secant BP (trainoss) algorithms.

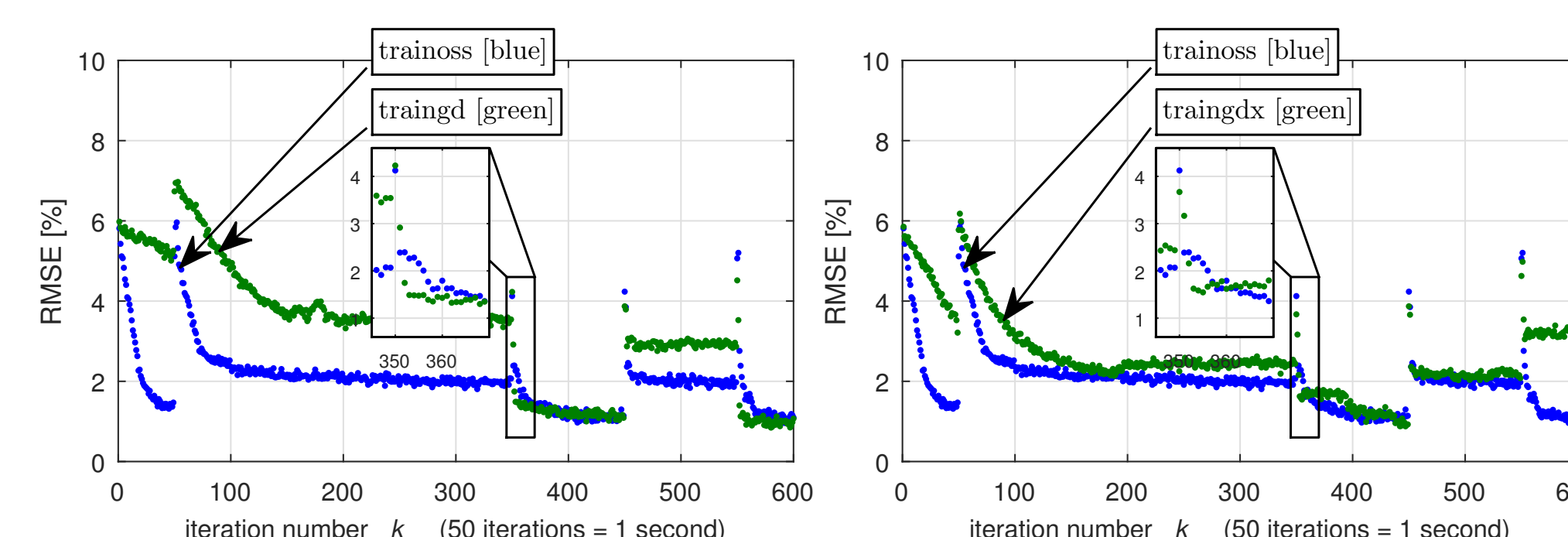


Figure 6: Root mean square error decay rate for one-step secant BP (trainoss), gradient descent (traingd) and gradient descent with momentum and an adaptive learning rate (traingdx) algorithms.

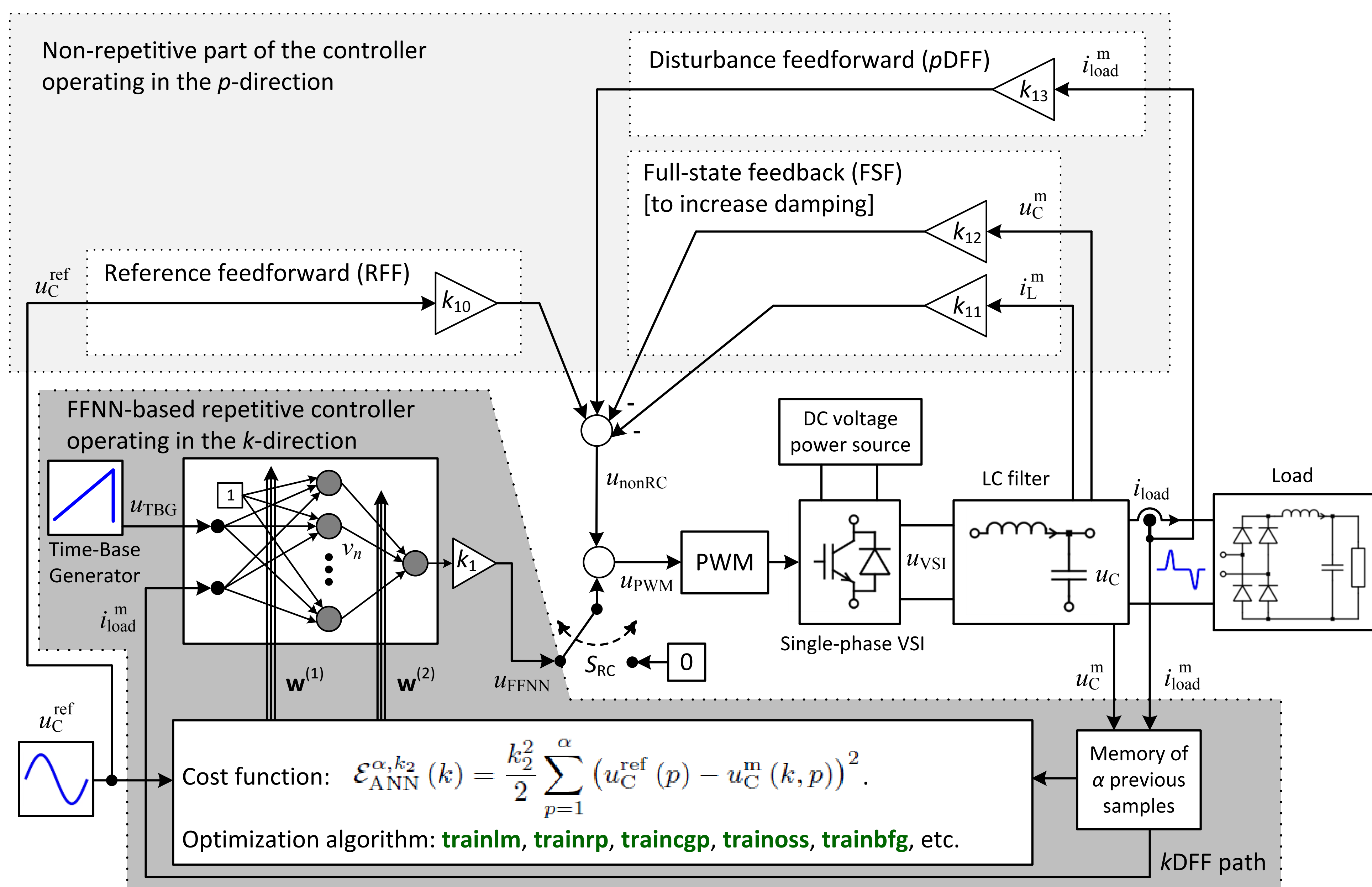


Figure 1: Schematic diagram of the proposed repetitive neurocontrol system with a disturbance dual feedforward path.

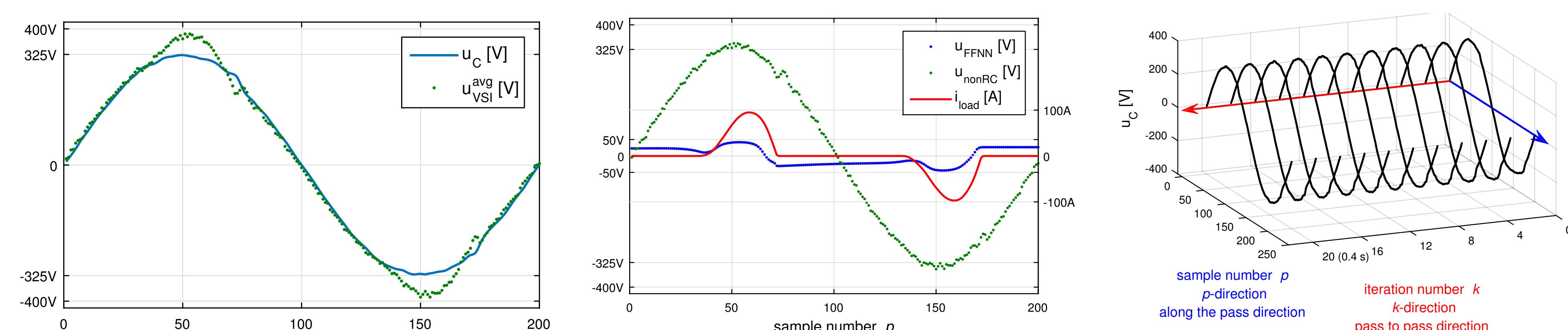


Figure 2: Steady-state waveforms (output capacitor voltage, PWM converter average voltage, control signal components and load current) under diode rectifier load and the evolution of output voltage after switching from resistive load to diode rectifier load.

Algorithm	Acronym	Exec. time*	Transients	Steady states
Levenberg-Marquardt BP	trainlm	41.2s	++	+
Resilient BP (RPROP)	trainrp	21.8s	+	++
One-step secant BP	trainoss	24.8s	++	++
BFGS quasi-Newton BP	trainbfg	24.0s	++	+
Gradient descent (GD)	traingd	22.4s	++	++
GD with momentum	traingdm	22.0s	++	++
GD with momentum & adaptive LR	traingdx	24.0s	++	++
GD with adaptive LR	traingda	22.9s	++	++
Bayesian regularization	trainbr	48.2s	-	+
Scaled Conjugate Gradient (CG)	trainscg	28.9s	-	+
Polak-Ribière CG (Conjugate Grad.)	traincgp	24.0s	+	+
Fletcher-Powell CG	traincgp	24.8s	+	++
CG with Powell/Beale restarts	traincgb	24.0s	+	++

*for 600 calls using Intel® Core™ i5-3210M CPU @ 2.50GHz

Conclusions

Repetitive processes can be effectively controlled using **dynamic optimization algorithms** such as iteratively trained artificial neural networks. It is advantageous to introduce the **disturbance feedforward** path also in the pass to pass direction. Recommendations regarding training algorithms are made. It is demonstrated that the **resilient backpropagation** algorithm – a frequent winner in the case of non-repetitive online-trained neurocontrollers – is **not a definitive winner** in the case of the repetitive neurocontroller.

Source code

The complete numerical model is available at MATLAB Central as “Repetitive Neurocontroller with Disturbance Feedforward”.

Contact information

- Web: <http://www.ee.pw.edu.pl>
- Email: bartlomiej.ufnalski@ee.pw.edu.pl
- Phone: +48 22 234 6138

Invitation

