Neural-network-based nonlinear iterative learning control: Magnetic brake study

2021 International Joint Conference on Neural Networks – IJCNN 2021, Shenzhen, China, July 18-22

Krzysztof Patan, Maciej Patan

Institute of Control and Computation Engineering
University of Zielona Góra, Poland

Introduction

- Intelligent control – a very popular and important branch of control methods
- Iterative learning control (ILC) – modern intelligent control strategy
- Neural networks – useful when dealing with nonlinear problems
  - nonlinear plant modeling
  - time-varying nonlinear learning controller realization
- Robustness of ILC – to consider state and output disturbances/uncertainty as well as initial state and initial feedback controller errors
- Experimental verification – magnetic brake system
- Magnetic brakes are often used in different areas such as big trucks, high speed railways, commercial vehicles or industrial elevators

Problem formulation

Consider a class of discrete-time nonlinear systems

$$\begin{aligned} \boldsymbol{x}_p(k+1) &= g(\boldsymbol{x}_p(k), u_p(k)) + \boldsymbol{w}_p(k), \;\; k = 0, \ldots, N-1, \\ y_p(k) &= \boldsymbol{C}\boldsymbol{x}_p(k) + v_p(k), \end{aligned} \tag{1}$$

where $p \geq 0$ – a trial number, N – a trial length
$\boldsymbol{x}_p(k)$, $u_p(k)$, $y_p(k)$ – system state, input and response
$\boldsymbol{w}_p(k)$, $v_p(k)$ – state and output disturbances/uncertainty
g – some nonlinear function
$\boldsymbol{C}$ – output (observation) matrix

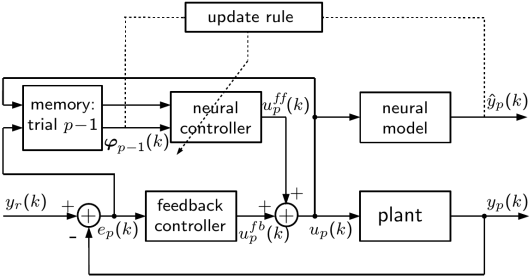
Problem: to design a control scheme such that the tracking error $e_p(k) = y_r(k) - y_p(k)$ ($y_r(k)$ – the reference trajectory) remains bounded in the presence of model uncertainty and disturbances

Iterative learning control

- Adaptation of so-called first-order ILC scheme (*current iteration ILC*)
  - use of existing feedback controller for stabilization,
  - adding supporting feedforward neural controller for tracking improvement,

$$u_p(k) = u_p^{fb}(k) + u_p^{ff}(k) \tag{2}$$

where p – trial number, $u_p^{fb}(k)$ – feedback control, $u_p^{ff}(k)$ – ILC update



- Feedback controller

$$\begin{aligned} \boldsymbol{z}_p(k+1) &= \boldsymbol{A}_c \boldsymbol{z}_p(k) + \boldsymbol{B}_c e_p(k) \\ u_p^{fb}(k) &= \boldsymbol{C}_c \boldsymbol{z}_p(k) + \boldsymbol{D}_c e_p(k) \end{aligned}, \tag{3}$$

where $\boldsymbol{z}_p(k)$ – the controller state

$\boldsymbol{A}_c$, $\boldsymbol{B}_c$, $\boldsymbol{C}_c$, $\boldsymbol{D}_c$ – controller matrices

- Learning controller

$$u_p^{ff}(k) = f(u_{p-1}(k), e_{p-1}(k)) \tag{4}$$

where f – a nonlinear function

Basic concept – use the neural network to provide time-varying realization of the function $u_p^{ff}(k)$ (being implicitly an inverted model of the plant)

Neural controller

- Neural network with one hidden layer

$$u_p^{ff}(k) = f(\boldsymbol{\varphi}_{p-1}(k)) = \boldsymbol{W}_{2,p}\sigma(\boldsymbol{W}_{1,p}\boldsymbol{\varphi}_{p-1}(k) + \boldsymbol{b}_{1,p}) + \boldsymbol{b}_{2,p},$$

where $\boldsymbol{\varphi}_{p-1}(k) = [u_{p-1}(k) \quad e_{p-1}(k)]$
$\boldsymbol{W}_{1,p}$, $\boldsymbol{W}_{2,p}$ – weight matrices
$\boldsymbol{b}_{1,p}$, $\boldsymbol{b}_{2,p}$ – bias vectors
σ – hidden neurons activation function

- Neural network parameters are updated after each process trial
- Stochastic gradient based training algorithm

  K. Patan and M. Patan, "Neural-network-based iterative learning control of nonlinear systems," *ISA Transactions*, vol. 98, pp. 445–453, 2020.

Neural model

- System modeling – state space neural network model

$$\hat{\boldsymbol{x}}_p(k+1) = \hat{g}(\hat{\boldsymbol{x}}_p(k), u_p(k), \varepsilon_p(k))$$
$$\hat{y}_p(k) = \boldsymbol{C}\hat{\boldsymbol{x}}_p(k)$$

where $\hat{\boldsymbol{x}}_{\boldsymbol{p}} \in \mathbb{R}^n$, $u_p \in \mathbb{R}^1$, $\hat{y}_p \in \mathbb{R}^1$ – model state, input and output
$\varepsilon_p(k) = y_p(k) - \hat{y}_p(k)$ – prediction error

- Implementation of nonlinear function $\hat{g}(\cdot,\cdot,\cdot)$:

$$\hat{g}(\cdot,\cdot,\cdot) = \boldsymbol{A}\hat{\boldsymbol{x}}_p(k) + \boldsymbol{V}_2\boldsymbol{\sigma}(\boldsymbol{V}_1^x\hat{\boldsymbol{x}}_p(k) + \boldsymbol{V}_1^u u_p(k) + \boldsymbol{V}_1^\varepsilon \varepsilon_p(k) + \boldsymbol{V}_1^b) + \boldsymbol{V}_2^b)$$

where $\boldsymbol{V}_1^u \in \mathbb{R}^{v_m \times 1}$, $\boldsymbol{V}_1^\varepsilon \in \mathbb{R}^{v_m \times 1}$, $\boldsymbol{V}_1^x \in \mathbb{R}^{v_m \times n}$, $\boldsymbol{V}_2 \in \mathbb{R}^{n \times v_m}$ – weight matrices
$\boldsymbol{V}_1^b \in \mathbb{R}^{v_m}$, $\boldsymbol{V}_2^b \in \mathbb{R}^n$ – bias vectors
$\boldsymbol{\sigma} : \mathbb{R}^{v_m} \to \mathbb{R}^{v_m}$ – the vector-valued activation function
v_m – the number of hidden neurons

$$\boldsymbol{A} = \begin{bmatrix} 0 & \dots & 0 & 0 \\ 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & 0 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix},$$

$$\boldsymbol{V}_1^x = \left[\begin{array}{c} \text{tunable weights} \\ \hdashline \mathbf{0}_{(v_m-n)\times n} \end{array}\right],$$

$$\boldsymbol{V}_2 = \left[\begin{array}{c:c} \mathbf{0}_{(n-1)\times n} & \text{tunable weights} \\ \hdashline \text{tunable weights} & \mathbf{0}_{1\times(v_m-n)} \end{array}\right].$$

- Training in batch mode (off-line) based on historical measurement data

Convergence analysis

Assumption A1. Let $\boldsymbol{w}_p(k) = 0$, $v_p(k) = 0$ and $y_r(k)$ be a reference defined over a discrete-time $k \in N$, which is assumed to be realizable, that is there exists a unique $u_r(k)$ and an initial state $\boldsymbol{x}_r(0)$, i.e.

$$\begin{aligned} \boldsymbol{x}_r(k+1) &= g(\boldsymbol{x}_r(k), u_r(k)) \\ y_r(k) &= \boldsymbol{C}\boldsymbol{x}_r(k) \end{aligned}.$$

Assumption A2. Let $\forall k \in N$, $\forall p$ state and output disturbances/uncertainties satisfy

$$\|\boldsymbol{w}_p(k)\| \leq \epsilon_w, \quad \|v_p(k)\| \leq \epsilon_v,$$

where $\epsilon_w \geq 0$, $\epsilon_v \geq 0$ are finite bounds. Moreover, $\forall p$ the initial system state error and initial feedback controller state satisfy

$$\|\Delta\boldsymbol{x}_p(0)\| \leq \epsilon_x, \quad \|z_p(0)\| \leq \epsilon_z, \tag{5}$$

where $\Delta\boldsymbol{x}_p(k) = \boldsymbol{x}_k(k) - \boldsymbol{x}_p(k)$, $\epsilon_x \geq 0$ and $\epsilon_z \geq 0$ are some positive constants.
Assumption A3. The nonlinear function g satisfies the global Lipschitz condition

$$\|g(\boldsymbol{x}_1, u_1) - g(\boldsymbol{x}_2, u_2)\| \leq L\left(\|\boldsymbol{x}_\mathbf{1} - \boldsymbol{x}_2\| + |u_1 - u_2|\right)$$

where $L > 0$ stands for the Lipschitz constant.

Theorem 1

Let us consider the nonlinear system (1) *which satisfies the assumptions (A1)–(A3) and the reference trajectory* $y_r(k)$ *satisfying the assumption (A1). Then using the control of the form* (2)-(4) *satisfying the condition*

$$\left| \sup_k \|f_u(k)\| + \sup_k \alpha_k L \frac{1 - \beta^{-(\lambda-1)N}}{\beta^\lambda - \beta} \right| < 1 \tag{6}$$

where $f_u(k) = \frac{\partial f}{\partial u_p^{ff}(k)}$, $f_e(k) = \frac{\partial f}{\partial e_p(k)}$,

$$\alpha_k = \max\{\|f_u(k)\|\|\boldsymbol{B}_c\|, \|f_u(k)\|\|\boldsymbol{D}_c\|\|\boldsymbol{C}\| + \|f_e(k)\|\|\boldsymbol{C}\|\},$$

$$\beta = \max\{L\|\boldsymbol{C}_c\| + \|\boldsymbol{A}_c\|, L + L\|\boldsymbol{D}_c\|\|\boldsymbol{C}\| + \|\boldsymbol{B}_c\|\|\boldsymbol{C}\|\},$$

guarantees that the tracking error is bounded, i.e.

$$\lim_{p\to\infty} \|y_r(k) - y_p(k)\|_\lambda \le \sigma \tag{7}$$

where constant $\sigma > 0$ *is dependent on* ϵ_w, ϵ_v, ϵ_x, ϵ_z.

Sketch of proof

- proof has been obtained as an extension of works:

1. K. Patan and M. Patan, "Neural-network-based iterative learning control of nonlinear systems," *ISA Transactions*, vol. 98, pp. 445–453, 2020

2. C.-J. Chien, "A discrete iterative learning control for a class of nonlinear time-varying systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 5, pp. 748–752, 1998

- proof is based on deriving uniform convergence property

$$\lim_{p\to\infty} u_p(k) = u_r(k),$$

through analysis of the induced norm imposed on the control law

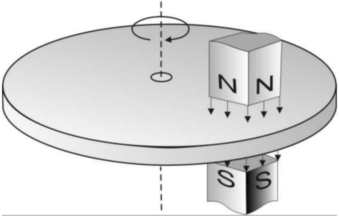
$$\|s(k)\|_\lambda = \sup_{k\in[0,N-1]} \beta^{-\lambda k}\|s(k)\|$$

- to deal with a nonlinear representation, the learning controller is expanded into Taylor series
- recursive nature of the state-space representation is also used

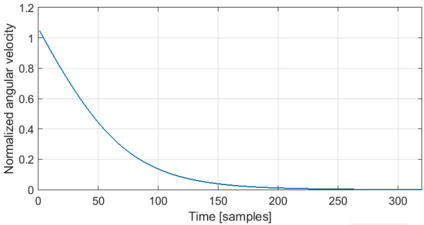
Illustrative example – magnetic brake

- System consists of a magnet inducing currents in a rotating disc of conductive material
- Aluminum disk with a radius of 10cm and a thickness of 1cm
- System input – the magnetic flux, system output – the angular velocity
- Initial value of velocity: 200 RPS

Scheme of magnetic brake system



Reference profile



Model developing

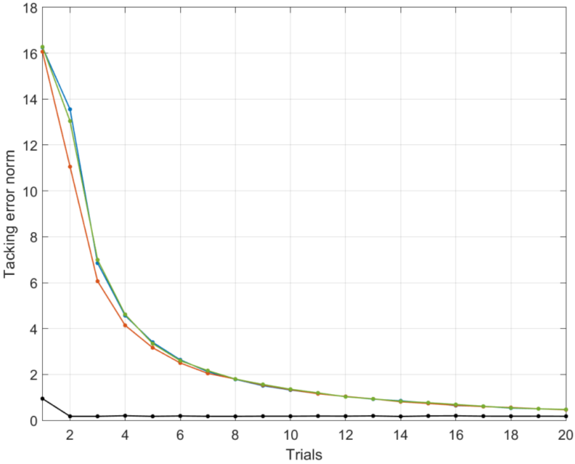
- Magnetic brake is a nonlinear system governed by spatio-temporal dynamics
- Problems with modeling:
  - physical effects such as nonlinear saturation, skin effects and eddy currents induced by motion must be considered simultaneously
  - a fine spatial mesh is required due to very small skin depths
  - a transient solution with time-stepping is necessary
- State-space neural networks provide an important alternative
- Investigated is stable – data recorded in the open-loop control feeding the system with different input signals
- Structure of the state-space neural network model: the model order: 3, the number of hyperbolic tangent neurons of the first layer: 15, the number of linear neurons of the second layer: 3
- Model training: Levenberg-Marquardt method
- Lipschitz constant for the trained model: $L = 1.66$

ILC controller synthesis

- open-loop ILC is used (without feedback controller)
- random initial neural controller parameters
- training dataset: $\{e(k), u(k)\}_{k=1}^{N}$ – recorded during the previous working cycle of the system
- structure of the neural controller: $v_c = 12$, $\sigma_h \equiv \tanh$
- training carried out after each trial

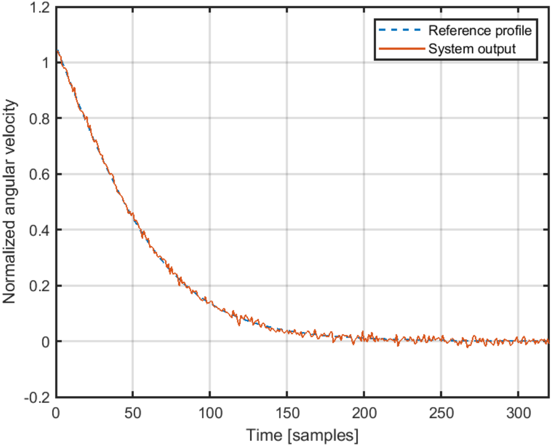
Results – control

Convergence: random initial parameters (blue, red, green)
preliminarily trained controller (black)



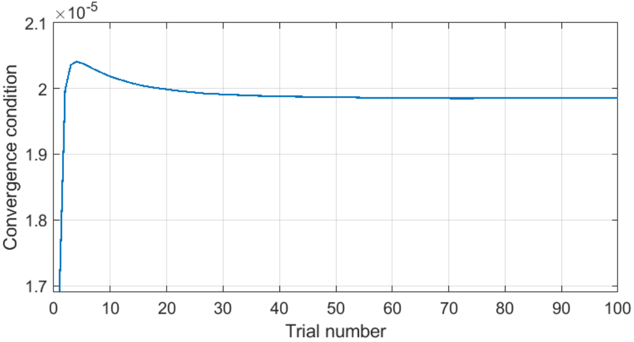
Results – output tracking

Output vs reference after 20 trails



Results – convergence condition satisfaction

Convergence condition for $\lambda = 10$



Concluding remarks

- A novel approach for robust ILC synthesis based on neural networks is proposed
- Learning controller has time-varying structure
- Modeling uncertainty and disturbances are taken into account
- Sufficient conditions guaranteeing convergence of the proposed neural-network-based ILC are provided
- Future research directions:
  - experiments using current-iteration setting
  - comparative studies with alternative control schemes dedicated to magnetic brake system