NEURAL ORDINARY DIFFERENTIAL EQUATIONS

AI tea talk, April 28, 2023

Jakub Ševčík

FACULTY OF ELECTRICAL ENGINEERING
UNIVERSITY OF WEST BOHEMIA

RICE

ORDINARY DIFFERENTIAL EQUATIONS

- Vector valued $\boldsymbol{y}$ changes in time respecting

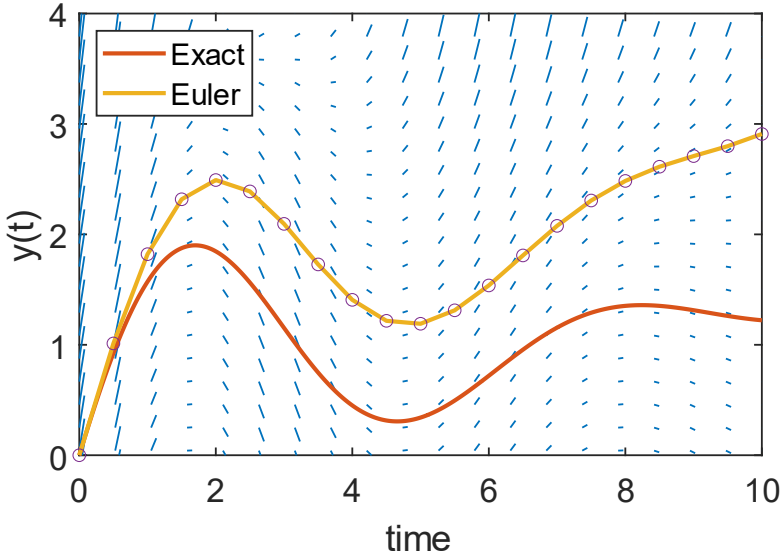
$$\frac{d\boldsymbol{y}}{dt}(t) = f_\theta(t, \boldsymbol{y}(t))$$

- Initial-value problem: given $\boldsymbol{y}(t_0)$, find:

$$\boldsymbol{y}(t_1) = \boldsymbol{y}(t_0) + \int_{t_0}^{t_1} f_\theta\big(t, \boldsymbol{y}(t)\big)dt$$

- Euler scheme

$$\boldsymbol{y}(t_0 + \Delta t) = \boldsymbol{y}(t_0) + \Delta t f(\boldsymbol{y}(t_0), t_0)$$



EULER AS AN ANALOGY TO RESNET

- Transformation of a hidden state in ResNets

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + f_{\theta_k}(k, \boldsymbol{y}_k)$$

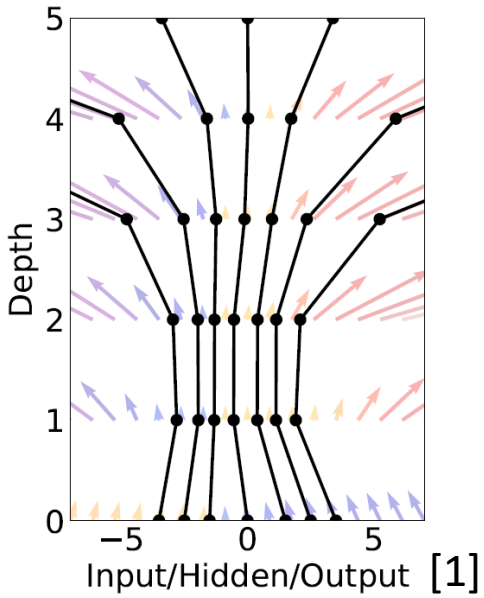
- The difference $\boldsymbol{y}_{k+1} - \boldsymbol{y}_k$ can be interpreted as a discretization of the derivative and letting $\Delta t \to 0$

$$\lim_{\Delta t \to 0} \frac{\boldsymbol{y}_{k+} - \boldsymbol{y}_k}{\Delta t} = \frac{d\boldsymbol{y}(t)}{dt} = f_\theta(t, \boldsymbol{y}(t))$$

- The Neural ODE can be seen as a continuous variant of ResNet

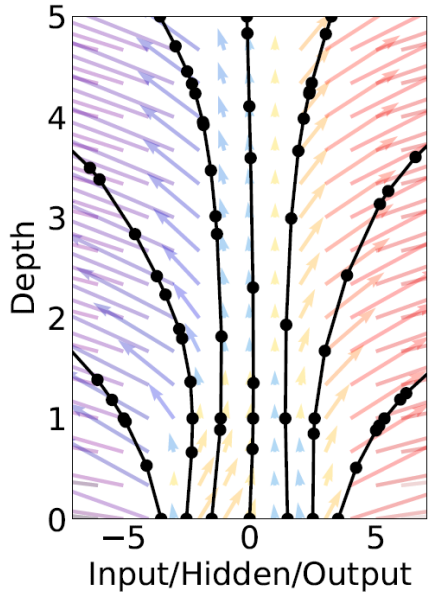
RESNET vs NEURAL ODE

```
def f(y, t, θ):
        return nnet(y, θ[t])

def resnet(z):
        for t in [1:T]:
                y = y + f(y, t, θ)
        return y
```



[1]

```
def f(y, t, θ):
        return nnet([y, t], θ)

def neuralODE(y):
        return ODESolve(f, y0, t0, t1, θ)
```



[1]

TRAINING OF NEURAL ODE

- Consider a scalar-valued loss function L, minimize

$$L\big(\boldsymbol{y}(t_1)\big) = L\left(\boldsymbol{y}(t_0) + \int_{t_0}^{t_1} f_\theta(\boldsymbol{y}(t), t)dt\right) = \mathrm{L}\big(\mathrm{ODESolve}(f_\theta, y(t_0), t_0, t_1, \theta)\big)$$

- How to get $\dfrac{\partial L\big(\boldsymbol{y}(t_1)\big)}{\partial \theta}$?
- Discretise-then-optimise
  - backpropagate through the internal operations of the differential equation solver
- Optimise-then-discretise (continuous adjoint method)
  - produce a backwards-in-time differential equation and solve it

- Reversible ODE solver

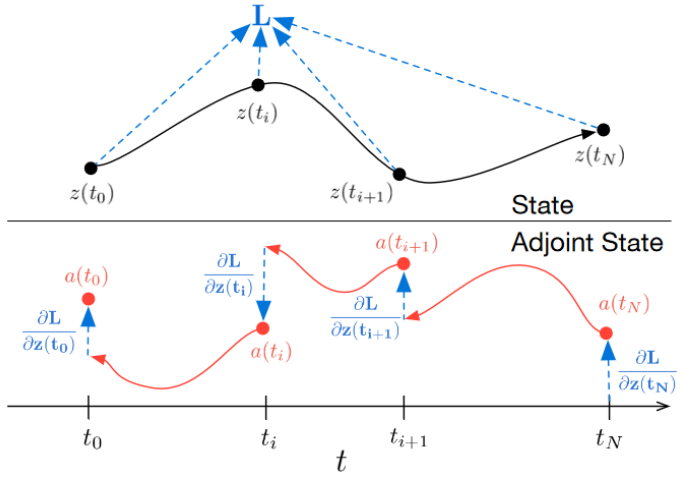
[1] Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, *31*.

TRAINING OF NEURAL ODE

- Optimise-then-discretise (continuous adjoint method)
  - produce a backwards-in-time differential equation and solve it
  - Define function $\boldsymbol{a}(t) = \partial L / \partial \boldsymbol{y}(t)$, then

$$\frac{d\boldsymbol{a}(t)}{dt} = -\boldsymbol{a}(t)^T \frac{\partial f_\theta(t, \boldsymbol{y}(t))}{\partial \boldsymbol{y}}$$

$$\frac{\partial L}{\partial \theta} = -\int_{t_1}^{t_0} \boldsymbol{a}(t)^T \frac{\partial f_\theta(t, \boldsymbol{y}(t))}{\partial \theta} \, \mathrm{dt}$$



- Solve ODE set with initial point $\left[\boldsymbol{y}(t_1), \frac{\partial L}{\partial \boldsymbol{y}(t_1)}, \boldsymbol{0}_{|\theta|}\right]^T$ and dynamics $\begin{bmatrix} f_\theta(t, \boldsymbol{y}(t)) \\ -\boldsymbol{a}(t)^T \frac{\partial f_\theta(t, \boldsymbol{y}(t))}{\partial \boldsymbol{y}} \\ -\boldsymbol{a}(t)^T \frac{\partial f_\theta(t, \boldsymbol{y}(t))}{\partial \theta} \end{bmatrix}$

  to get $\left[\boldsymbol{y}(t_0), \frac{\partial L}{\partial \boldsymbol{y}(t_0)}, \frac{\partial L}{\partial \theta}\right]^T$

[1] Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, *31*.

TRAINING OF NEURAL ODE

- Discretise-then-optimise
  - memory inefficient
  - accurate and fast
  - ODE solver in an AD framework
- Optimise-then-discretise
  - memory efficient
  - approximate and a little slow
  - ODE solver can be a black-box

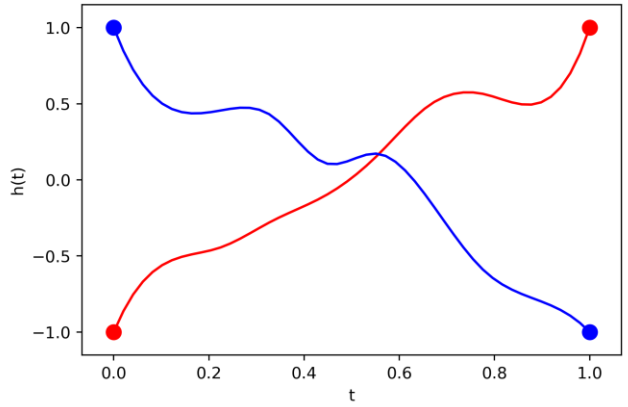
[4] Kidger, P. (2022). On neural differential equations. *Doctoral thesis, University of Oxford. arXiv:2202.02435.*

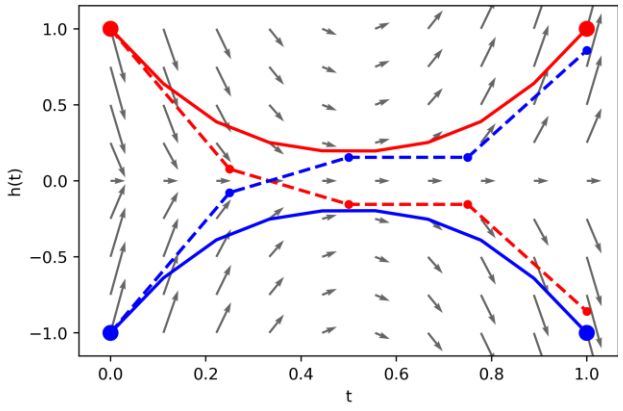
LIMITATIONS OF NODE LEAD TO ANODE

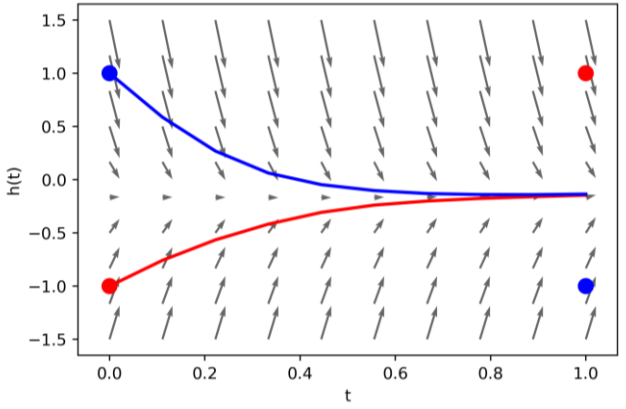
- NeuralODE are not universal approximators
- Consider function $g: \mathbb{R} \to \mathbb{R}$ such that
  - $g(-1) = 1$
  - $g(1) = -1$
- Augmented NODE: augments the ODE space from $\mathbb{R}^d$ to $\mathbb{R}^{d+d_0}$

$$\frac{d}{dt}\begin{bmatrix}\boldsymbol{y}(t)\\ \boldsymbol{h}(t)\end{bmatrix} = f_\theta\left(t, \begin{bmatrix}\boldsymbol{y}(t)\\ \boldsymbol{h}(t)\end{bmatrix}\right), \quad \begin{bmatrix}\boldsymbol{y}(0)\\ \boldsymbol{h}(0)\end{bmatrix} = \begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{0}\end{bmatrix}$$

- Augmented NODEs are universal approximators

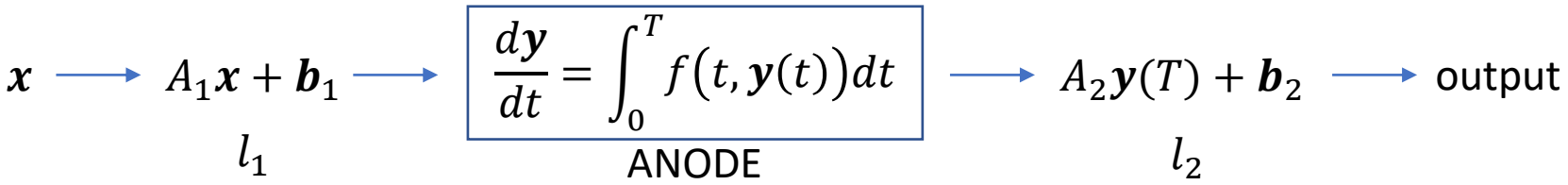






[2] Dupont, E., Doucet, A., & Teh, Y. W. (2019). Augmented neural odes. *Advances in neural information processing systems*, *32*.

ANODES ARE UNIVERSAL APPROXIMATORS



THEOREM

Fix $d, d_l, d_0 \in \mathbb{N}$ with $d_l \geq d + d_0$. For $f \in Lip(\mathbb{R} \times \mathbb{R}^{d_l}; \mathbb{R}^{d_l})$, $l_1 \in L_b(\mathbb{R}^d; \mathbb{R}^{d_l})$, $l_2 \in L_b(\mathbb{R}^{d_l}; \mathbb{R}^{d_0})$, let $\varphi_{f,l_1,l_2}: \mathbb{R}^d \to \mathbb{R}^{d_0}$ denotes the map $x \to z$ with

$$y(0) = l_1(x), \quad \frac{dy}{dt}(t) = f(t, y(t)), \quad \text{for } t \in [0, T], \quad z = l_2(y(T)).$$

Then

$$\left\{\varphi_{f,l_1,l_2} \,|\, f \in Lip(\mathbb{R} \times \mathbb{R}^{d_l}; \mathbb{R}^{d_l}), l_1 \in L_b\left(\mathbb{R}^d; \mathbb{R}^{d_l}\right), l_2 \in L_b(\mathbb{R}^{d_l}; \mathbb{R}^{d_0})\right\}$$

is a universal approximator for $C(\mathbb{R}^d; \mathbb{R}^{d_0})$.

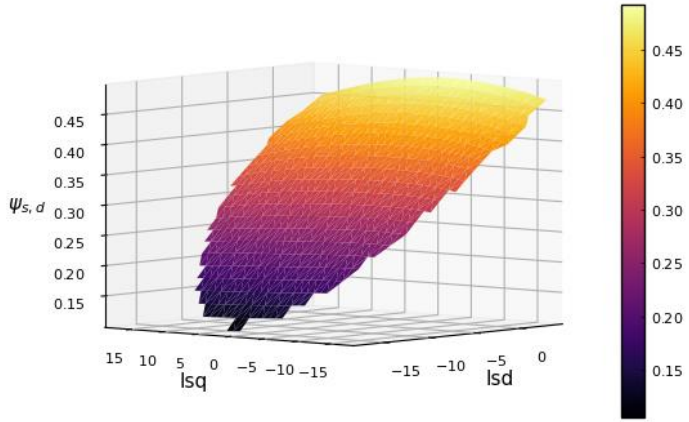
[3] Zhang, H., Gao, X., Unterman, J., & Arodz, T. (2020). Approximation capabilities of neural ODEs and invertible residual networks. In *International Conference on Machine Learning* (pp. 11086-11095).

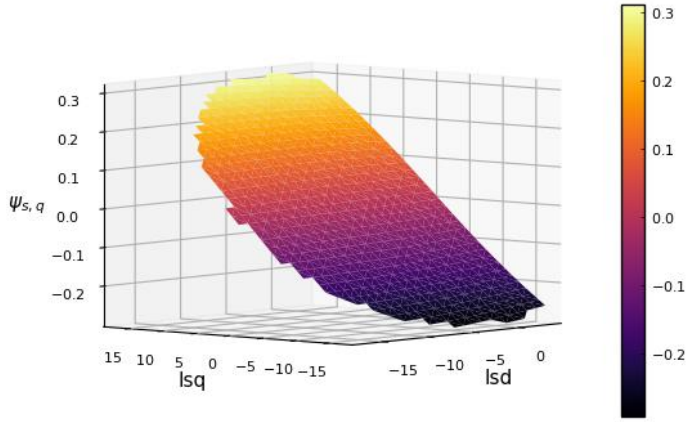
CASE STUDY: Neural ODE for Synchronous Drives

- Estimation of magnetic field of an IPMSM (Interior Permanent-Magnet Synchronous Motor)
- Synchronous machine electrical model in the rotating (d, q)- reference frame

$$\boldsymbol{u}_s = R\boldsymbol{i}_s + \omega_{el} J\boldsymbol{\psi}(\boldsymbol{i}_s, \boldsymbol{x}) + \frac{d}{dt}\boldsymbol{\psi}(\boldsymbol{i}_s, \boldsymbol{x})$$

- The aim is to get a good approximation of $\widehat{\boldsymbol{\psi}} \approx \boldsymbol{\psi}(\boldsymbol{i}_s, \boldsymbol{x})$



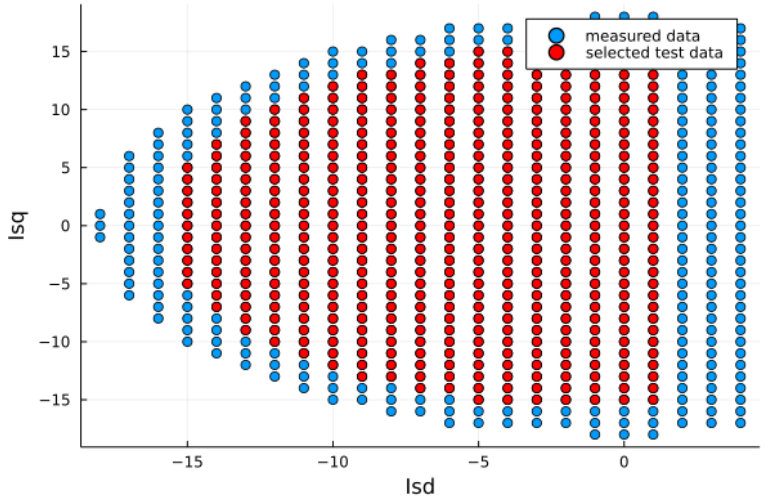


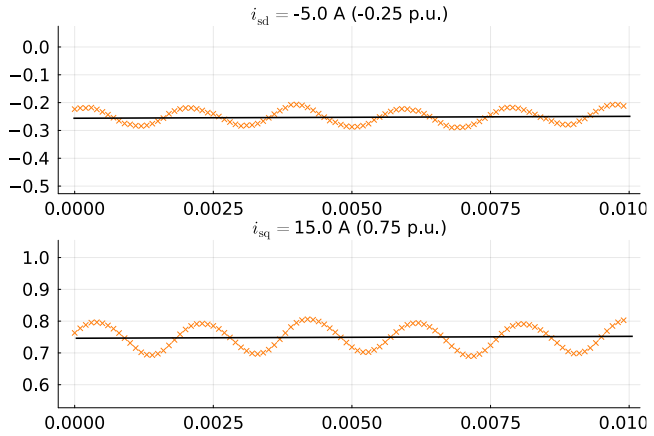
CASE STUDY: Neural ODE for Synchronous Drives

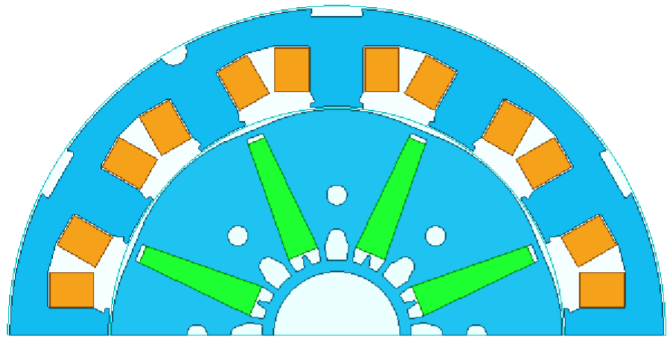
- Common approaches use look-up tables (LUT)
- It assumes steady regimes

$$\boldsymbol{u}_s = R\boldsymbol{i}_s + \omega_{el} J\boldsymbol{\psi}(\boldsymbol{i}_s, \boldsymbol{x}) + \frac{d}{dt}\boldsymbol{\psi}(\boldsymbol{i}_s, \boldsymbol{x}), \qquad \text{since} \quad \frac{d}{dt}\boldsymbol{\psi}(\boldsymbol{i}_s, \boldsymbol{x}) = 0$$

- The data for a constant set point $[i_{sdw}, i_{sqw}]$ are clustered and a common flux-linkage map (LUT) is obtained using the OLS (simple averaging)
- $2N^2$ parameters with inputs $\boldsymbol{i}_s$, $2N^{2+|x|}$ parameters when additional inputs are added





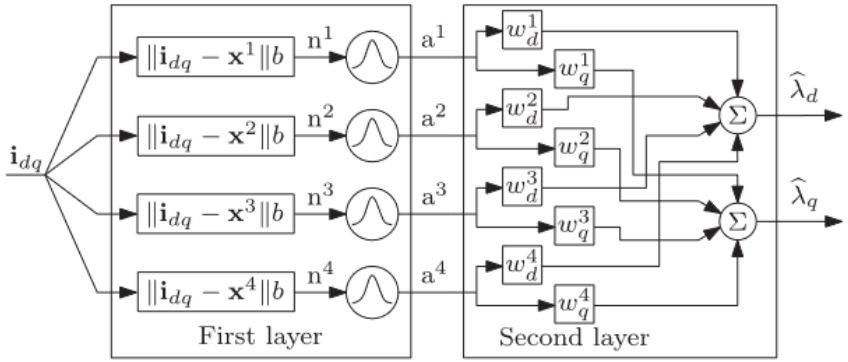


CASE STUDY: Neural ODE for Synchronous Drives

- Instead of a common LUT, we can use Radial Basis Functions (RBF)

$$\hat{\psi}_{RBF,sd}(\boldsymbol{i}_s) = \sum_{k=1}^{K} w_{d,k} \exp(-(b\|\boldsymbol{i}_s - \boldsymbol{x}_k\|)^2), \qquad \hat{\psi}_{RBF,sq}(\boldsymbol{i}_s) = \sum_{k=1}^{K} w_{q,k} \exp(-(b\|\boldsymbol{i}_s - \boldsymbol{x}_k\|)^2),$$

where $\boldsymbol{x}_k$ is a fixed center of the k-th RBF, and $w_{d,k}$ and $w_{q,k}$ are unknown weights of the model.



- Advantages of RBF model:
  - Weights $w_{d,k}$ and $w_{q,k}$ can be estimated using OLS from steady model
  - Weights $w_{d,k}$ and $w_{q,k}$ can be estimated using OLS even from dynamic model (considering $\frac{d}{dt}\hat{\boldsymbol{\psi}}_{RBF}(\boldsymbol{i}_s)$)

[5] Ortombina, L., Pasqualotto, D., Tinazzi, F., & Zigliotto, M. (2020). Magnetic model identification of synchronous motors considering speed and load transients. IEEE Transactions on Industry Applications, 56(5), 4945-4954.

CASE STUDY: Neural ODE for Synchronous Drives

- From voltage equation

$$\boldsymbol{u}_s = R\boldsymbol{i}_s + \omega_{el} J\boldsymbol{\psi}(\boldsymbol{i}_s, \boldsymbol{x}) + \frac{d}{dt}\boldsymbol{\psi}(\boldsymbol{i}_s, \boldsymbol{x}), \quad \text{using } \frac{d\boldsymbol{\psi}(\boldsymbol{i}_s,\boldsymbol{x})}{dt} = \frac{\partial \boldsymbol{\psi}(\boldsymbol{i}_s,\boldsymbol{x})}{\partial \boldsymbol{i}_s}\frac{d\boldsymbol{i}_s}{dt} + \frac{\partial \boldsymbol{\psi}(\boldsymbol{i}_s,\boldsymbol{x})}{\partial \boldsymbol{x}}\frac{d\boldsymbol{x}}{dt}$$

we can express the dynamic equation for current (if $\boldsymbol{x} = \emptyset$)

$$\frac{d}{dt}\hat{\boldsymbol{i}}_s = \underbrace{\text{inv}\left(\frac{\partial \hat{\boldsymbol{\psi}}(\boldsymbol{i}_s)}{\partial \boldsymbol{i}_s}\bigg|_{\hat{\boldsymbol{i}}_s}\right)}_{L(\boldsymbol{i}_s)} \cdot \left(\boldsymbol{u}_s - R\hat{\boldsymbol{i}}_s - \omega_{el} J\hat{\boldsymbol{\psi}}(\hat{\boldsymbol{i}}_s)\right)$$

or more generally

$$\frac{d}{dt}\hat{\boldsymbol{i}}_s = \text{inv}\left(\frac{\partial \hat{\boldsymbol{\psi}}(\boldsymbol{i}_s,\boldsymbol{x})}{\partial \boldsymbol{i}_s}\bigg|_{\hat{\boldsymbol{i}}_s,\hat{\boldsymbol{x}}}\right) \cdot \left(\boldsymbol{u}_s - R\hat{\boldsymbol{i}}_s - \omega_{el} J\hat{\boldsymbol{\psi}}(\hat{\boldsymbol{i}}_s,\hat{\boldsymbol{x}}) - \frac{\partial \hat{\boldsymbol{\psi}}(\boldsymbol{i}_s,\boldsymbol{x})}{\partial \boldsymbol{x}}\bigg|_{\hat{\boldsymbol{i}}_s,\hat{\boldsymbol{x}}}\frac{d}{dt}\hat{\boldsymbol{x}}\right)$$

CASE STUDY: Neural ODE for Synchronous Drives

- Dynamic equation for stator current with an unknown term $\widehat{\boldsymbol{\psi}}(\boldsymbol{i}_s, \boldsymbol{x})$ that is modeled as NN
  - Sometimes called universal differential equation [6]
- Training:
  - Statically, i.e. minimizing $\left\| \widehat{\boldsymbol{\psi}}(\boldsymbol{i}_s, \boldsymbol{x}) - \frac{J^{-1}(\boldsymbol{u}_s - R\boldsymbol{i}_s)}{\omega_{el}} \right\|_2$
  - Dynamically, i.e. ODE solution for current vs. measurement ($\Delta t = 10^{-4}$)
- $\widehat{\boldsymbol{\psi}}(\boldsymbol{i}_s, \omega_{el})$ vs. $\widehat{\boldsymbol{\psi}}(\boldsymbol{i}_s, \omega_{el}, \sin(K_s\vartheta), \cos(K_s\vartheta))$

[6] Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., ... & Edelman, A. (2020). Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*.

CASE STUDY: Neural ODE for Synchronous Drives

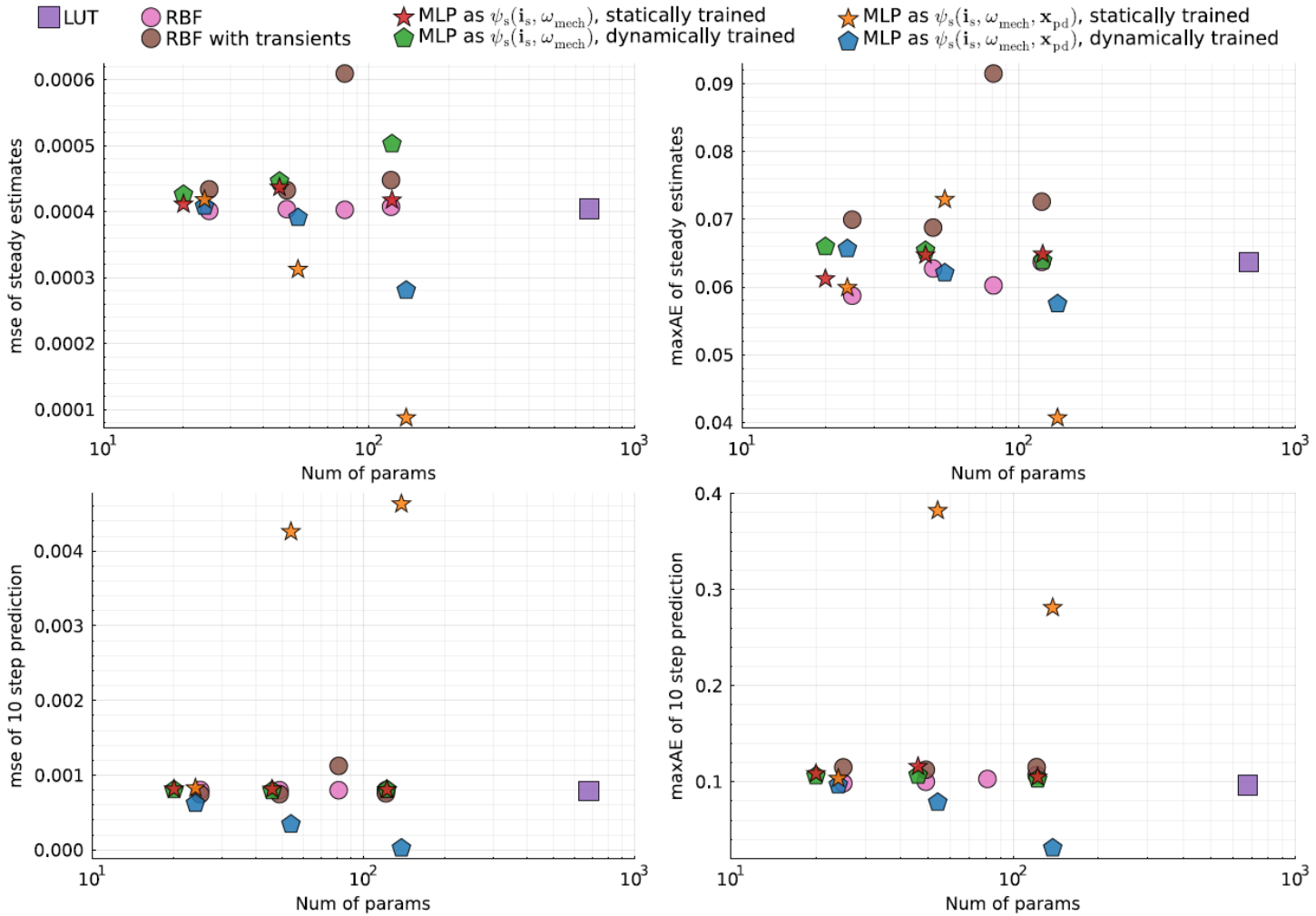


Fig. 4. Estimation errors of IPMSM's flux linkages models using validation data set with the constant speed $\omega^*_{mech} = -1300$ rpm. MSE in the left column, maxAE in the right column, errors of steady evaluation in the top row, errors of 10-step prediction in the bottom row. (10-step predictions of RBF-transient model with 9x9 grid has a great maxAE error (6.48) and therefore it is not displayed in the graph.)

CASE STUDY: Neural ODE for Synchronous Drives

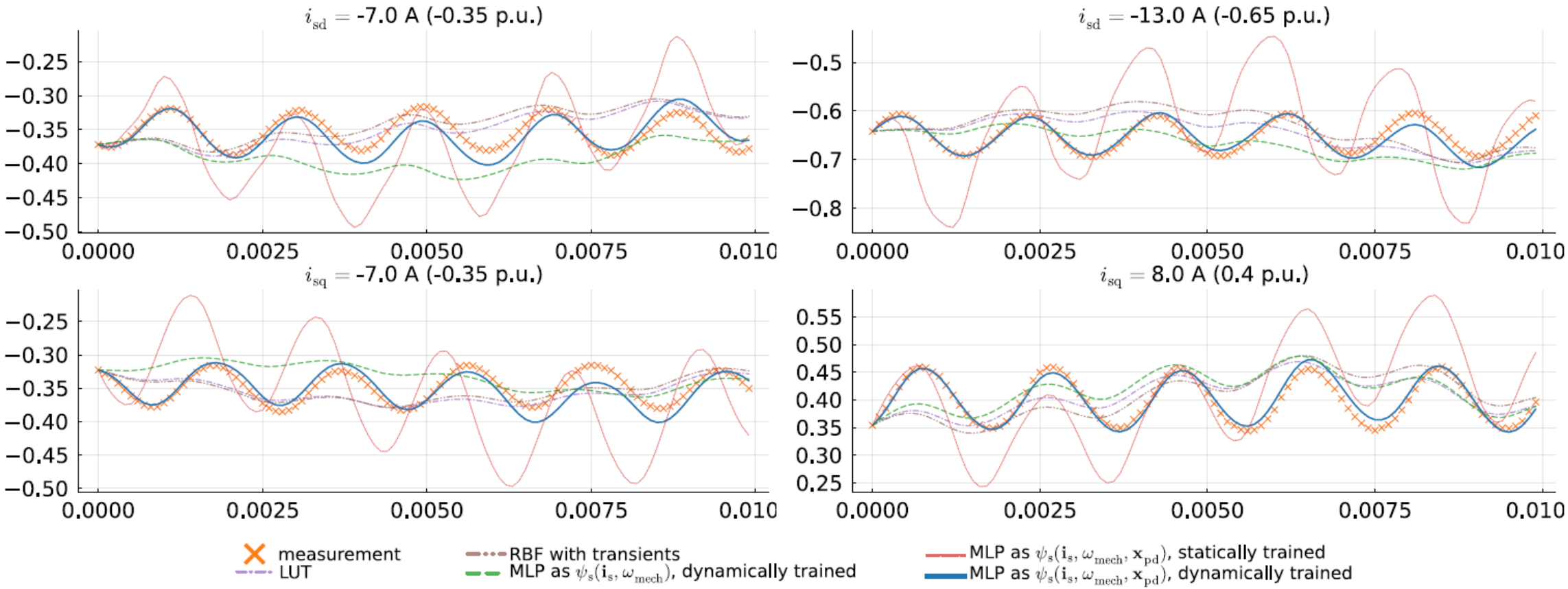


Fig. 5. Example of 100-step ahead predictions of stator currents i_{sd} (top row) and i_{sq} (bottom row) using trained flux linkage model of the IPMSM. The measured data has been acquired at constant speed -1300 RPM.

LITERATURE

- Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, *31*.
- Dupont, E., Doucet, A., & Teh, Y. W. (2019). Augmented neural odes. *Advances in neural information processing systems*, *32*.
- Zhang, H., Gao, X., Unterman, J., & Arodz, T. (2020). Approximation capabilities of neural ODEs and invertible residual networks. In *International Conference on Machine Learning* (pp. 11086-11095).
- Kidger, P. (2022). On neural differential equations. *Doctoral thesis, University of Oxford. arXiv:2202.02435.*
- Ortombina, L., Pasqualotto, D., Tinazzi, F., & Zigliotto, M. (2020). Magnetic model identification of synchronous motors considering speed and load transients. IEEE Transactions on Industry Applications, 56(5), 4945-4954.
- Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., ... & Edelman, A. (2020). Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*.

NEURAL ORDINARY DIFFERENTIAL EQUATIONS

Thank you for Your attention

Jakub Ševčík
377 634 180
jsevcik@fel.zcu.cz
fel.zcu.cz

FACULTY OF ELECTRICAL ENGINEERING
UNIVERSITY OF WEST BOHEMIA

RICE