Physics Informed Deep Learning Model of Electric Motor

Chair of Interconnected Automation Systems

Chirag Angadi

Literature Review

Data-Driven Permanent Magnet Temperature Estimation in Synchronous Motors With Supervised Machine Learning: A Benchmark

Insights: Modeling of electric motor for magnet temperature estimation using ML algorithms.

Algorithms used: Linear regression, MLP, CNN, SVM, KNN, Random forest, Ensembles.

Inputs: *Ambient temp, coolant temp, voltage, current, motor speed*

Output: Magnet temp

Libraries: Tensorflow, scikit-learn

Results:

Model	MSE in °C^2	MAE in °C	R^2	ℓ_∞ norm in °C	model size	norm. inference duration
k-NN	26.10	4.24	0.87	12.86	221k	5595
RF	16.26	3.09	0.92	10.9	1.1M	4.9
SVR	13.42	2.75	0.93	31.99	209k	337
ET	6.51	1.77	0.97	8.29	5.5M	12.1
LPTN [45]	5.73	1.98	0.97	**6.45**	**46**	57.7
RNN [16]	3.26	1.29	0.98	9.1	1.9k	60
MLP	3.20	1.32	0.98	8.34	1.8k	14.8
OLS	3.10	1.46	0.98	7.47	109	**1.0**
CNN [16]	**1.52**	**0.85**	**0.99**	7.04	67k	115

Conclusion: Liner regression and MLP are better methods considering time and number of parameters

Literature Review

Data-driven Thermal Modeling for Electrically Excited Synchronous Motors - A Supervised Machine Learning Approach

Insights: online thermal modeling of electrically excited synchronous motors (EESMs).

Algorithms used: Ordinary least square(OLS)

Inputs: Ambient temp, coolant temp, voltage, current, motor speed, torque

Output: Rotor and stator winding temp

Libraries: scikit-learn

Results:

Rotor thermal model		Stator thermal model	
MSE	MAE	MSE	MAE
2.8983	1.2797	0.4664	0.4693

Conclusion: Developed OLS model is close to true function.

Literature Review

Surrogate Modeling of Electrical Machine Torque Using Artificial Neural Networks

Insights: Modeling of motor to predict the torque using data driven models

Algorithms used: ANN & GBDT

Inputs: Current amplitude and frequency

Output: Torque

Libraries: Tensorflow, scikit-learn

Results:

Algorithm	Architecture	Simulation speed	Loss(RMSE)
ANN	2 layers with 512 neurons	68ms	1.14%
GBDT	7500 trees max depth 40	0.77s	2.17%

Conclusion: ANN model is 2900 times faster that physics based(FEA) model with good accuracy

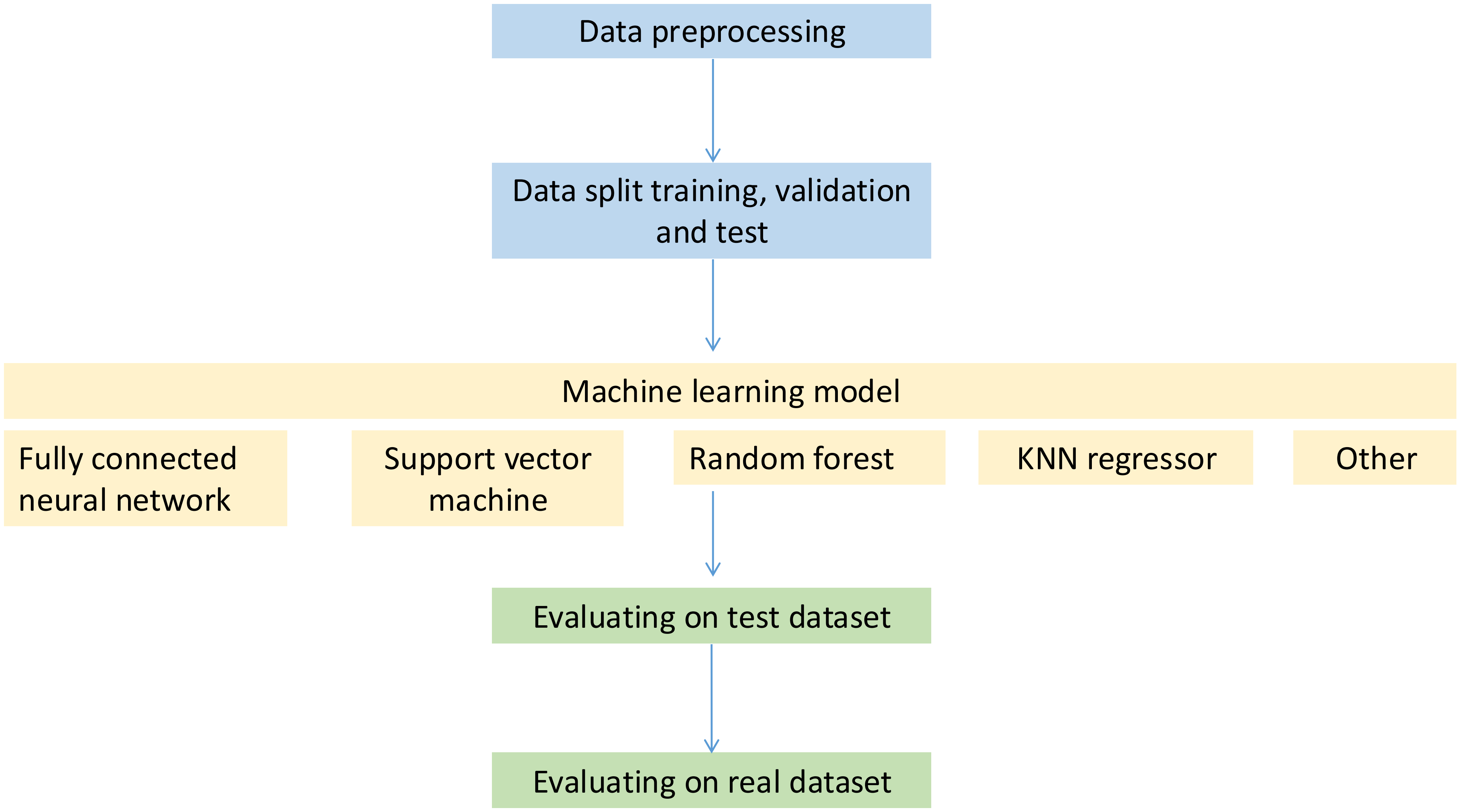
Conclusion

- Earlier research has been done for surrogate model development of different types of motors.
- Due to computation efforts of physics based model, surrogate models are more effficent.
- Mainly models were developed for temperature estimations of permanet magnet, rotar and stator windings.
- As the temperature relation has linear relation, model was developed using ordinary least square algorithm.
- One research has been done to estimate the torque using surrogate model,were more complex alogrithms line neural networks, random forest, Knearestneighbour etc. alogorithms are implemented.
- However, very less work in done on development of model to determine torque, speed and losses together. Our aim is to develop this model.

Approch

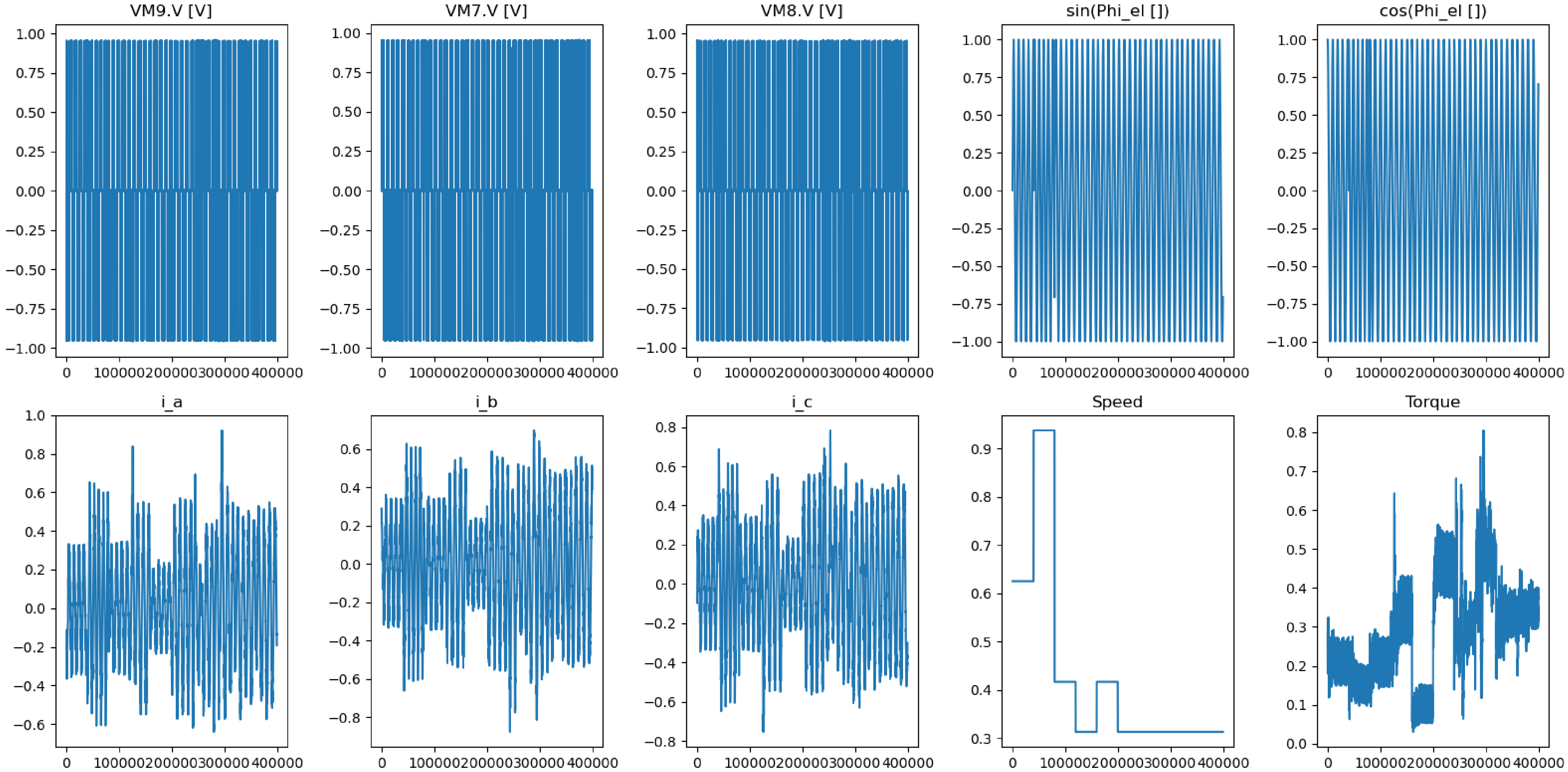
- As identified from literature review torque, speed and loss estimation are nonlinear function and requires more complex functions.
- Therefore, model will be developed with Neural networks, Random forest, Neural ordinary differential, Support vector machine.
- Further the best performing(i.e. less value of loss function) will be considered as final model.
- Training of ML algorithms will be done with data collected from FEA model with simulation.
- Training, validation and testing on simulation dataset will be done in 60-20-20 data split.
- Further, the finalized models will be tested on real data generated by motor.

Process


Data preprocessing
Data split training, validation and test
Machine learning model
Fully connected neural network
Support vector machine
Random forest
KNN regressor
Other
Evaluating on test dataset
Evaluating on real dataset

Dataset

- Normalized dataset



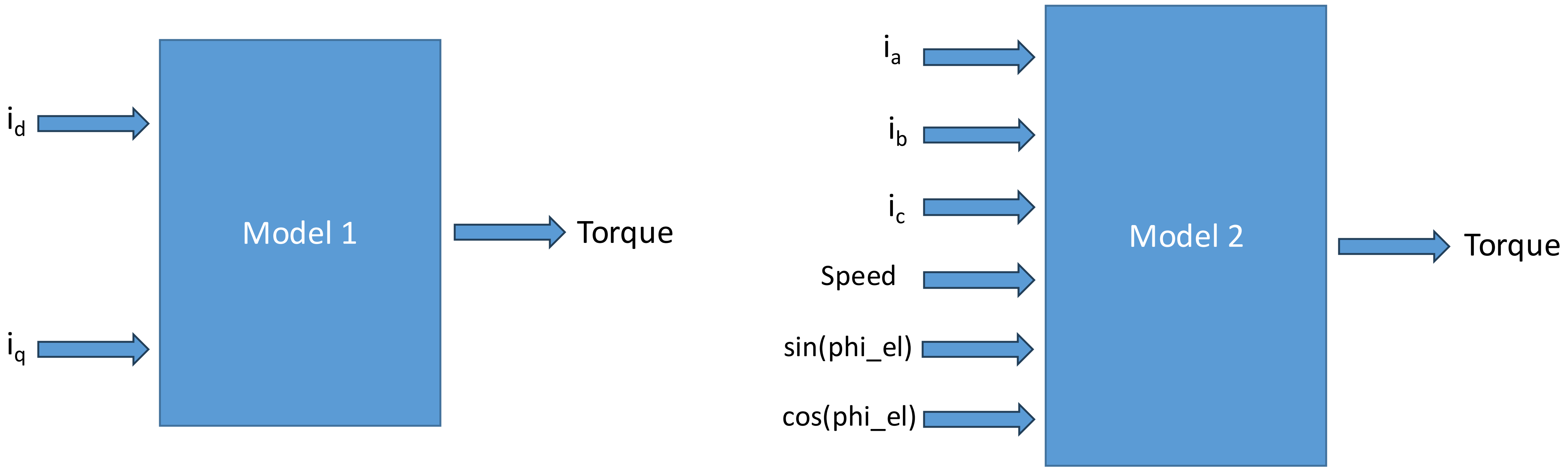
Modeling Static Behavior of Motor

Modeling static behavior of motor

- The torque produced by motor is modeled by d-q currents i_d , i_q. The model is considered as black box as it is modeled without any domain knowledge and physics-based equations.
- Alternatively, torque can be modeled using line currents and electrical angle as an inputs.
- The modeling is done with machine learning algorithms such as Artificial neural networks(ANN), Polynomial Regression, Random Forest, AdaBoost, XG-Boost.
- Random forest performs best with least mean absolute error and mean squared error.
- Other algorithms base model also performs good. All algorithms have MAE less than 2 and MSE less than 4 on non-normalised torque scale.

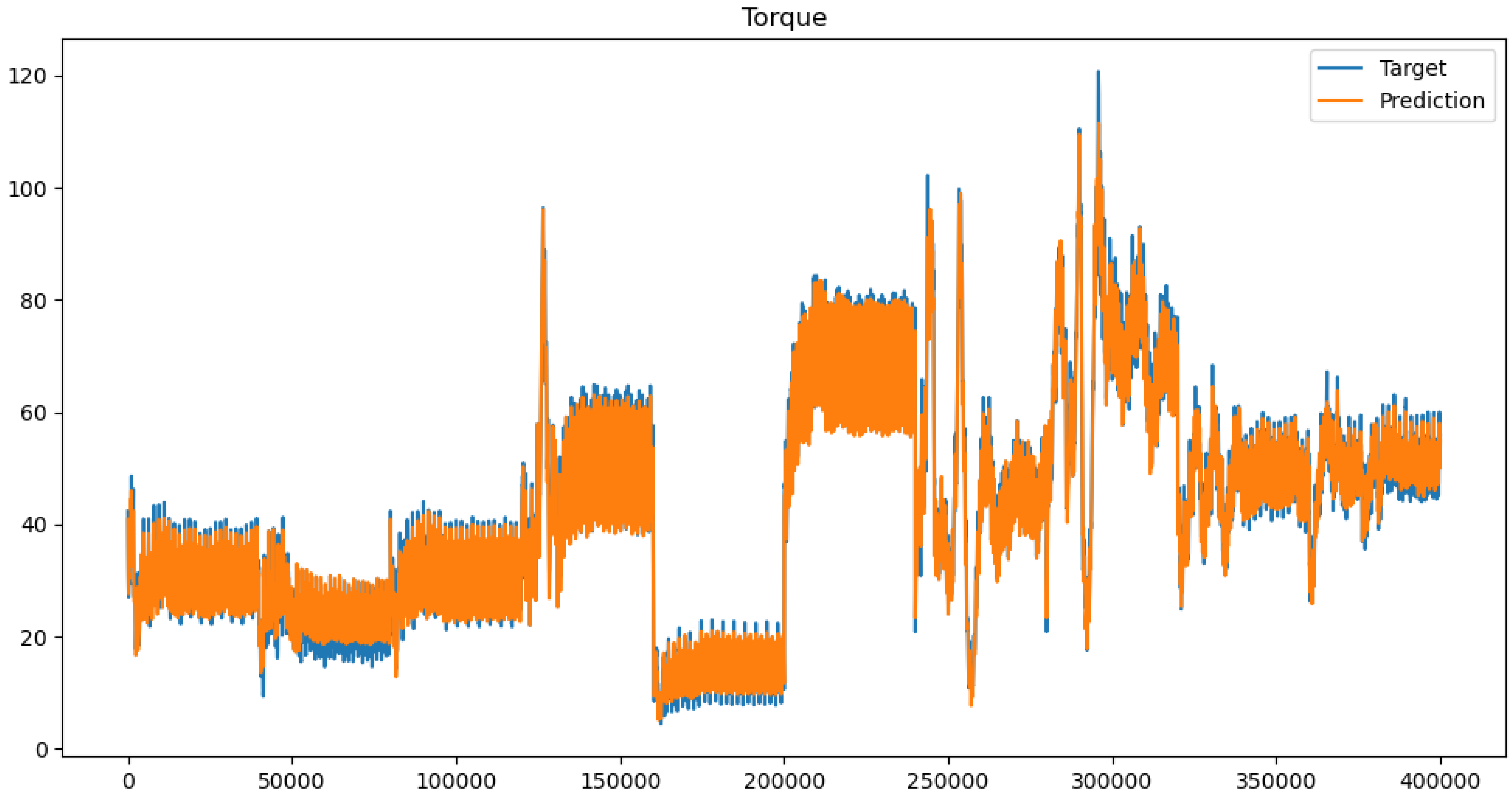
Modeling static behavior of motor

- *Static Models*



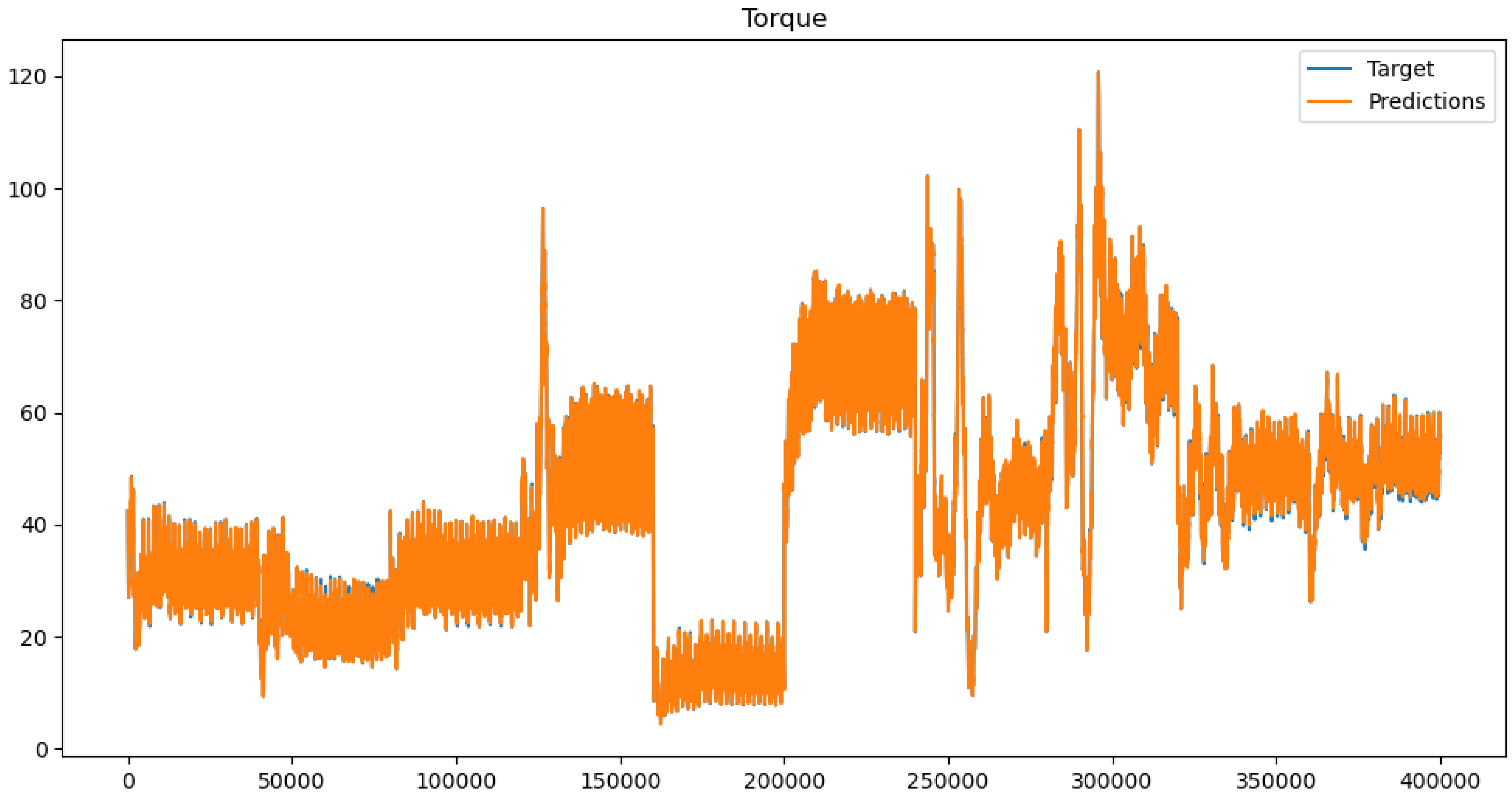
Modeling static behaviour of motor

- *Model 1 - Artificial Neural Network(ANN)*



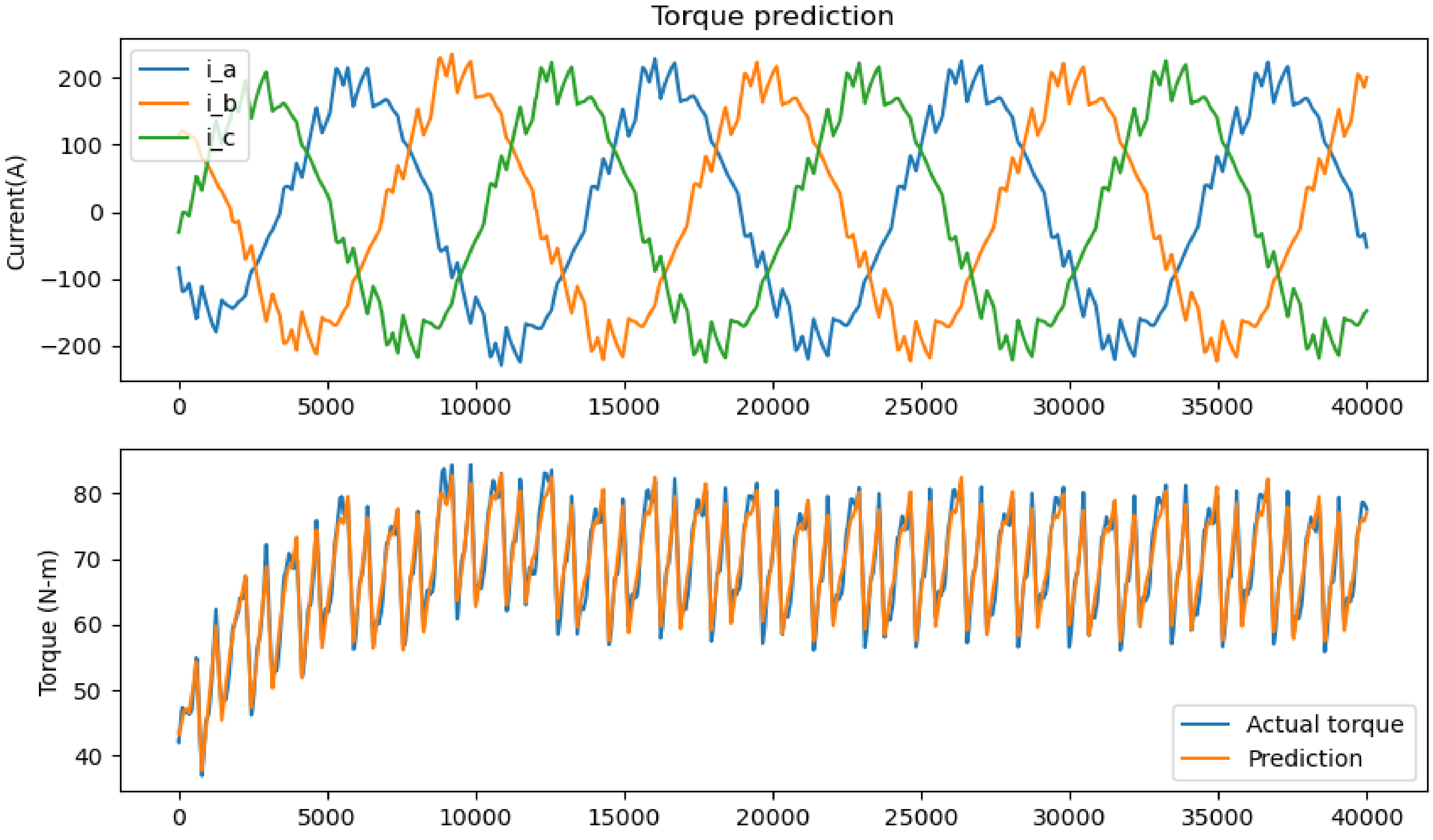
Modeling static behavior of motor

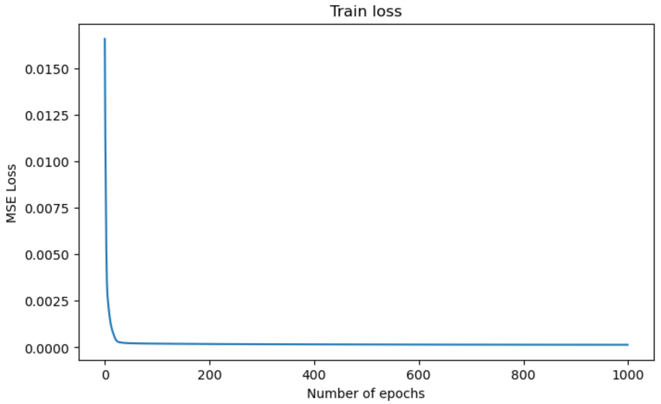
- *Model 1 - Random Forest*

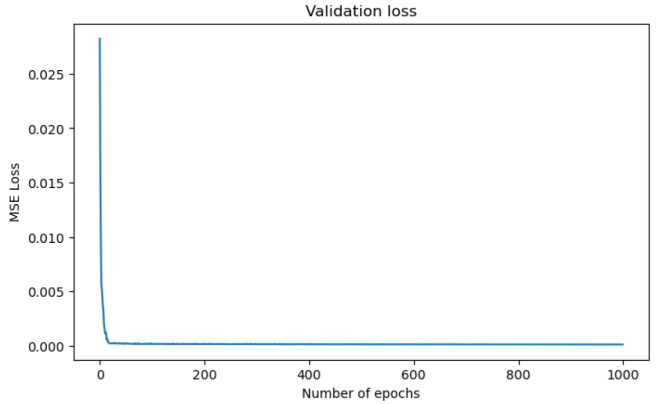


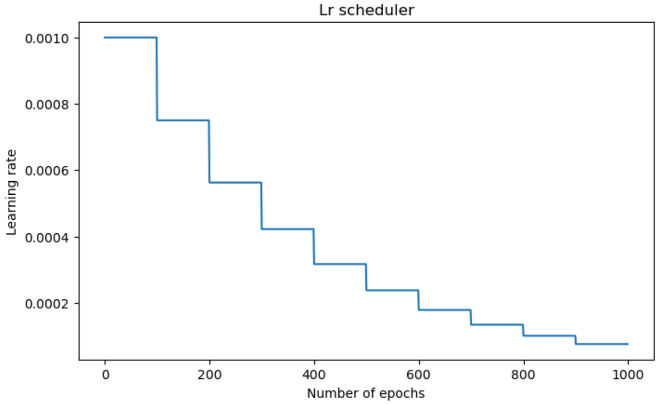
Modeling static behavior of motor

- *Model 2 - Artificial Neural Network(ANN)*









Normalised data - Test error (MAE): 0.00875, Test error (MSE): 0.000113
Denormalised data - Test error (MAE): 1.313, Test error (MSE): 2.547

Modeling static behavior of motor

- Further substitutes of random forest, advanced algorithms like XGBoost and AdaBoost algorithms are also used for modeling. However, they didn't pose much improvement in accuracy compared to random forest.
- Evaluation Matrix: Model 1

	MAE(N-m)	MSE(N-m)	Max error(N-m)
Artificial neural network(MLP)	1.679	4.649	13.08
Polynomial regression	1.498	3.66	10.35
Random Forest	0.398	0.351	5.08
XGBoost	1.322	2.973	12.7
AdaBoost	3.268	16	13.13

- Evaluation Matrix: Model 2

	MAE(N-m)	MSE(N-m)	Max error(N-m)
Artificial neural network(MLP)	1.313	2.547	4.37

Modeling static behavior of motor

- Conclusion:
- Static behavior of motor is feasible to model using machine learning algorithm. A black model will have current, motor speed and electrical angle as an input and torque as an output.
- Neural networks and Random forest algorithm poses much better algorithm compared to others. However, these are base models, optimization of algorithm needs to be done.

Modeling Dynamic Behavior of Motor

Modeling dynamic behaviour of motor

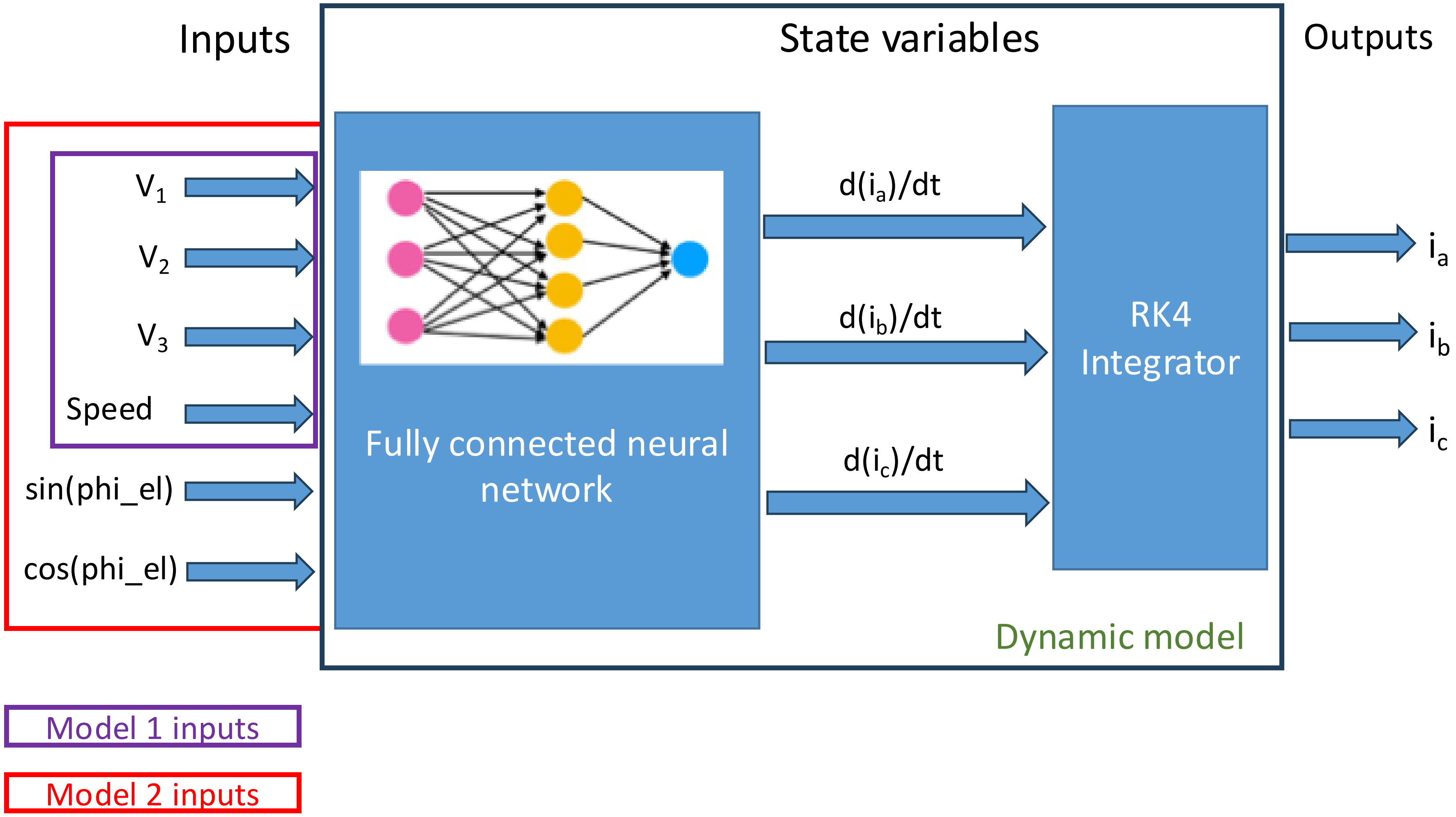
- As the motor behavior is dynamic in nature with respect to its voltage input with pulse width modulation(i.e. the current in line and torque produced at a time instance is dependent on previous inputs and previous hidden states of the machine)
- Therefore, static modeling algorithms like artificial neural network, polynomial regression etc. alone cannot model the dynamics of the motor.

- Dynamics of motor is modeled using algorithms like:

1. Neural Ordinary Differential equations (NODE)
2. Recurrent neural network
3. Long short-term memory (LSTM)

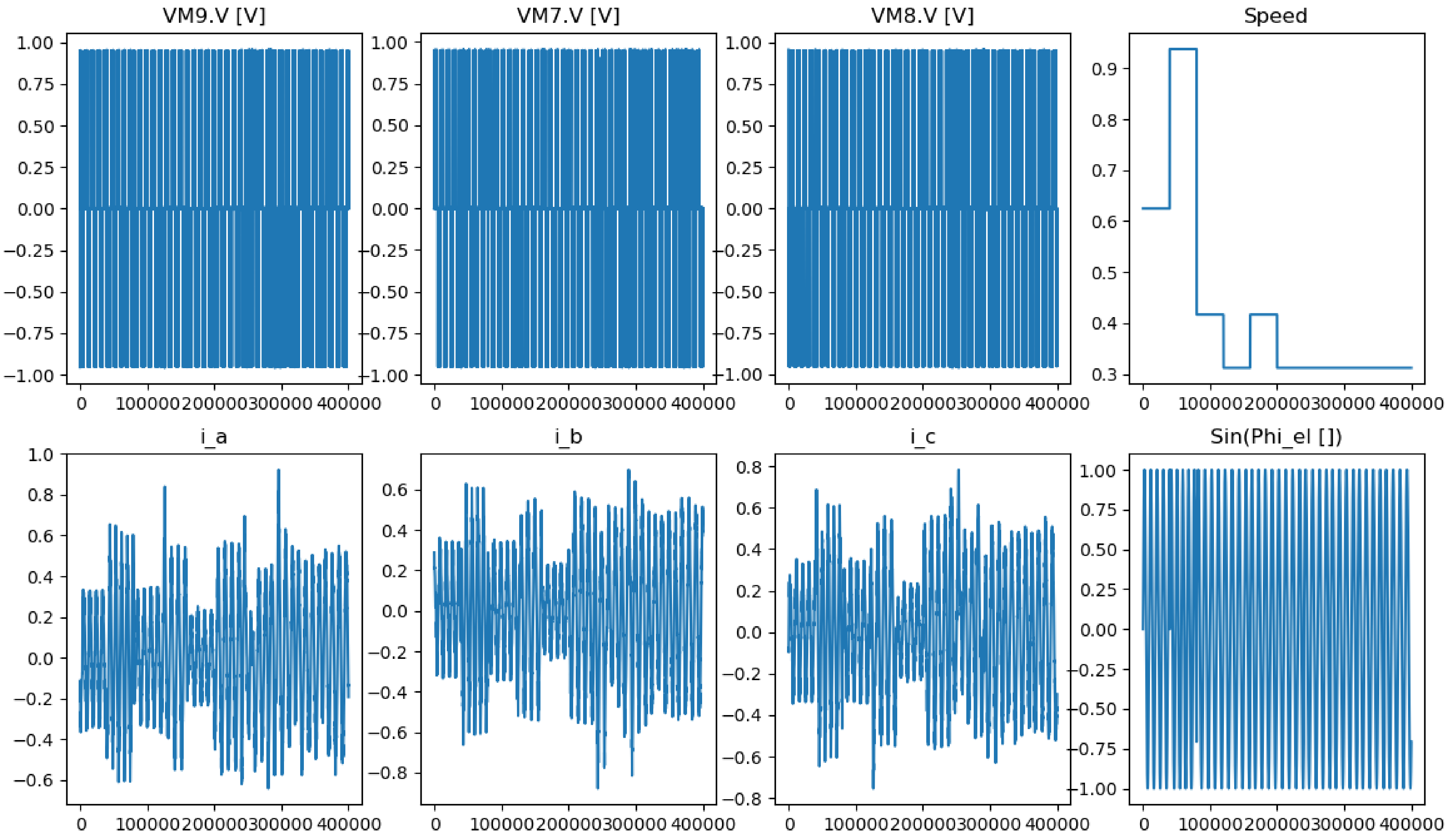
Modeling dynamic behavior of motor

- *Model Architecture - Neural ordinary differential equation(NODE)*



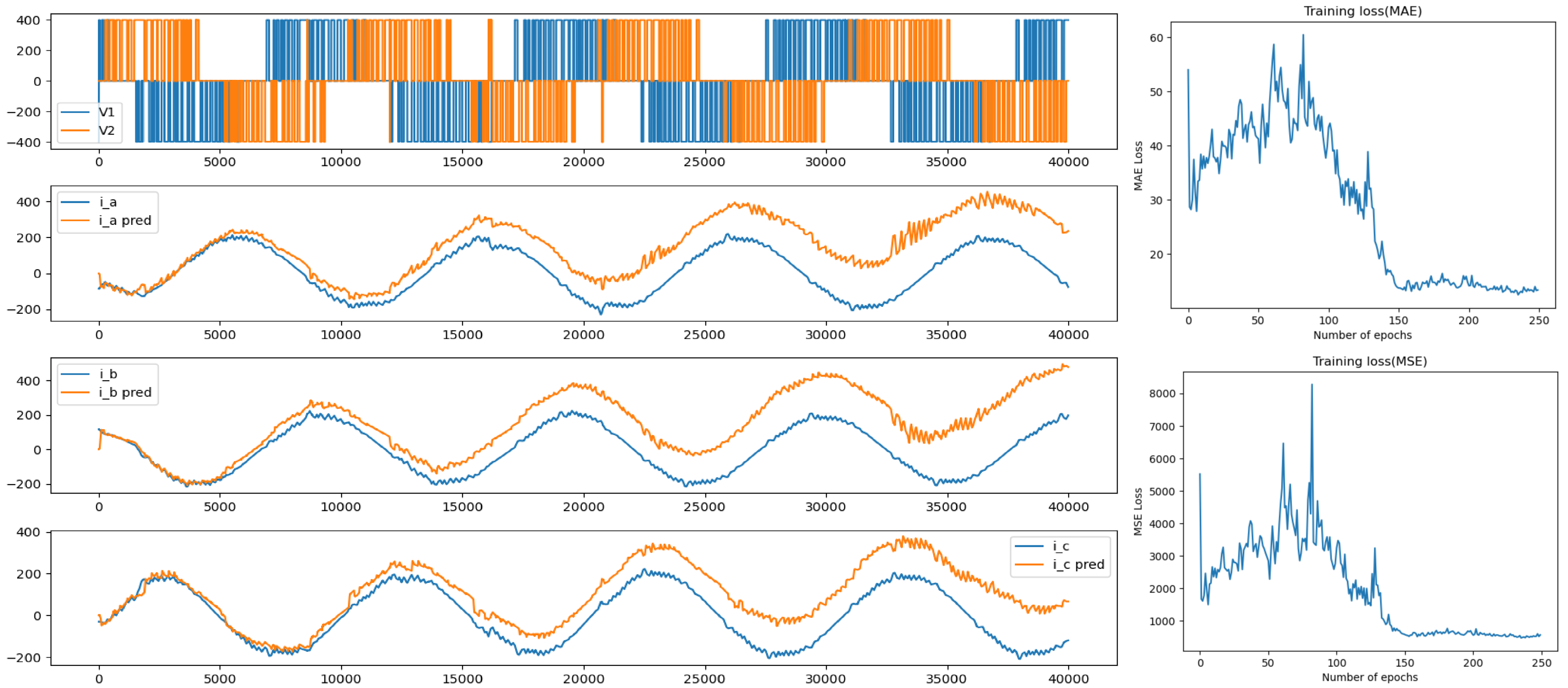
Modeling dynamic behavior of motor

- Normalised data



Modeling dynamic behavior of motor

- *NODE Model 1 - Line currents prediction (10000 sim steps)*



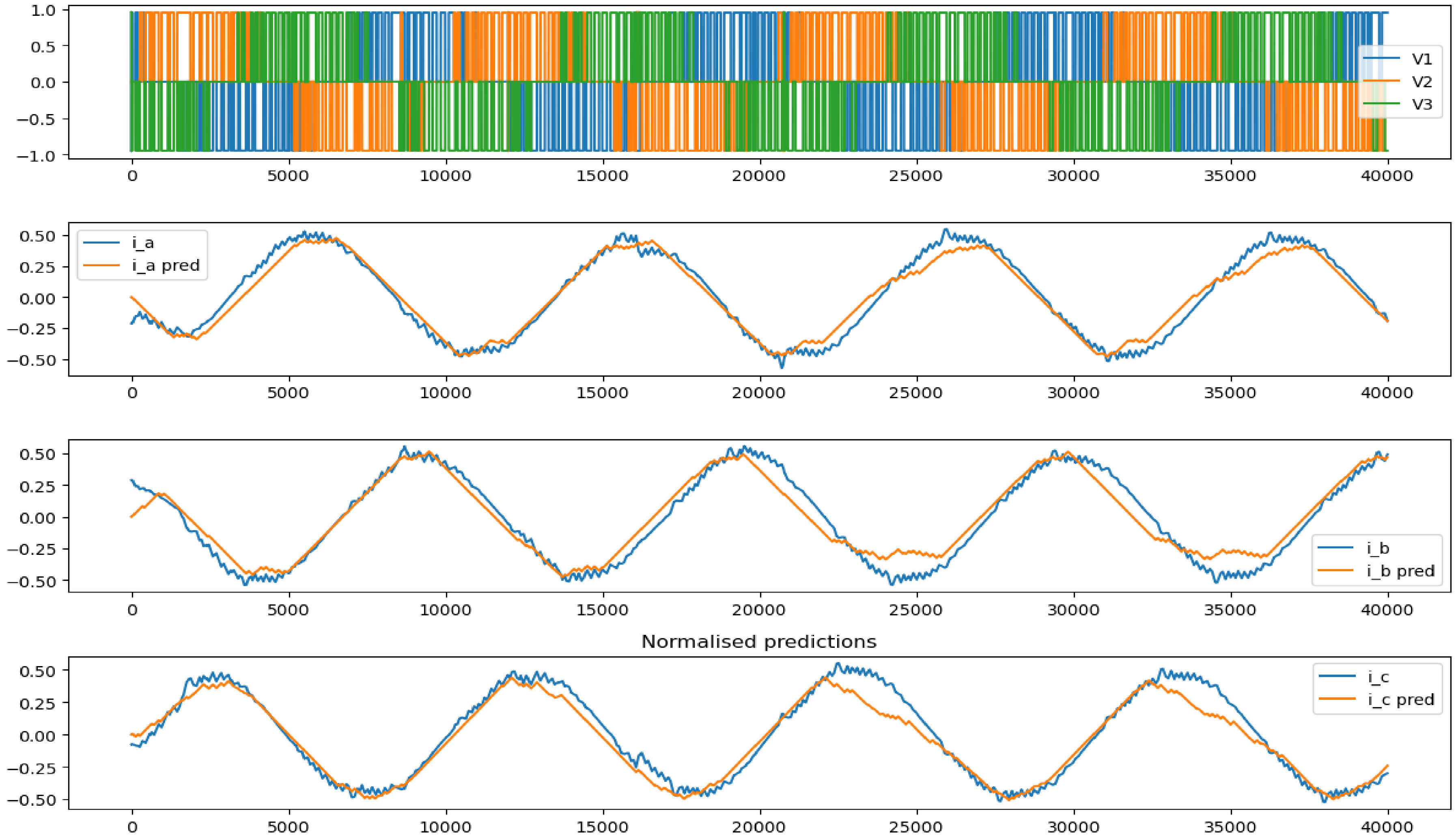
Test error (MAE): 131.689, Test error (MSE): 25260.539

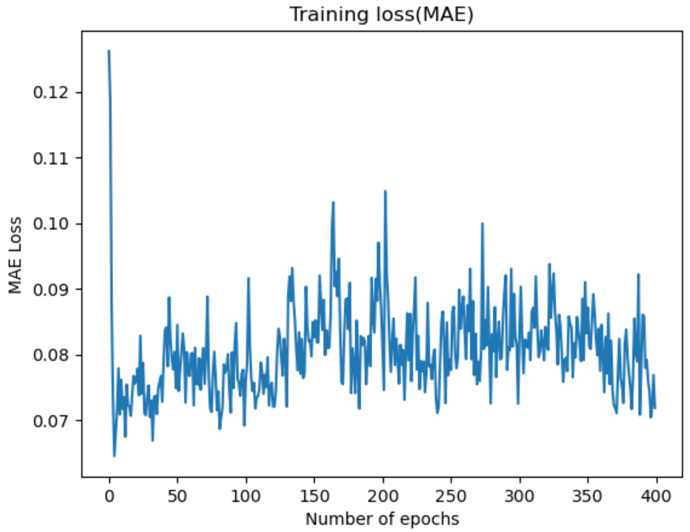
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

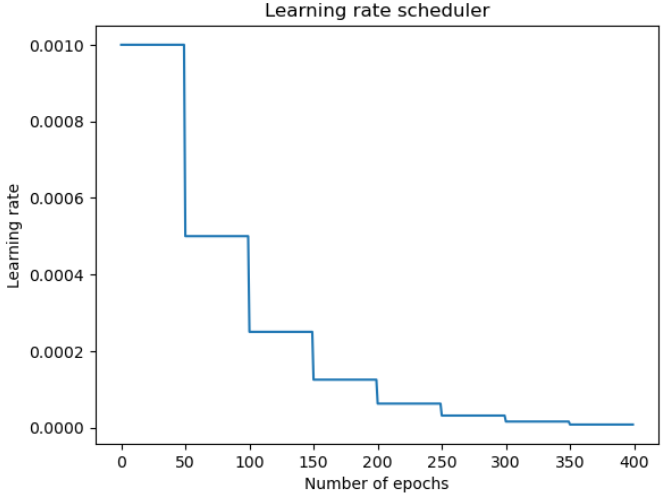
$W_1 \in R^{64\times 6}$, $W_2 \in R^{64\times 64}$, $W_3 \in R^{3\times 64}$

Modeling dynamic behavior of motor

- *NODE Model 1 (tanh activation at output) - Line currents prediction*







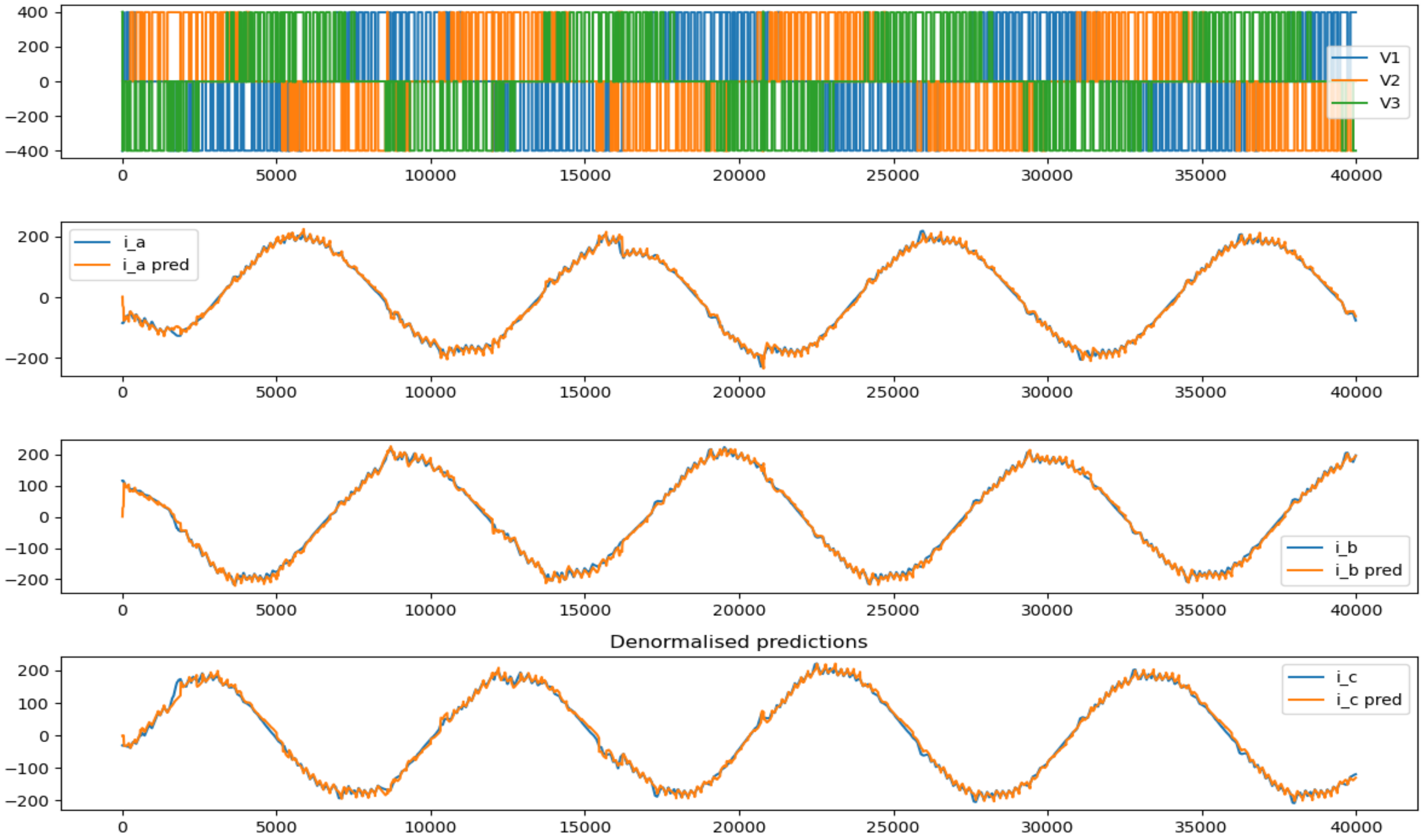
Normalised data - Test error (MAE): 0.0657, Test error (MSE): 0.0070
Denormalised data - Test error (MAE): 26.30, Test error (MSE): 1132.44
Train time: 482 min, Inference time: 5.5 sec

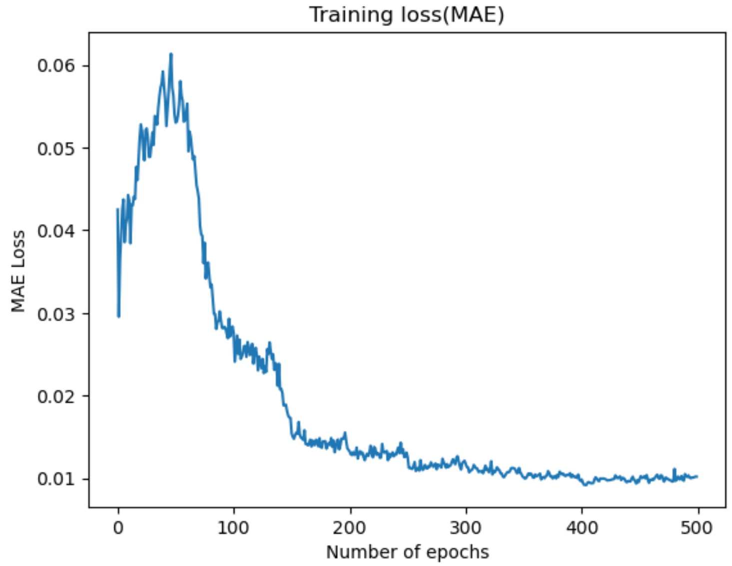
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

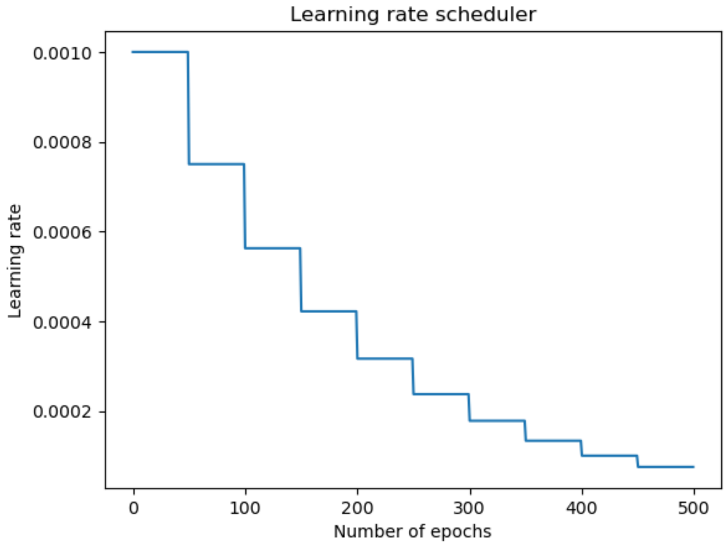
$W_1 \in R^{64\times 6}$, $W_2 \in R^{64\times 64}$, $W_3 \in R^{3\times 64}$

Modeling dynamic behavior of motor

- *NODE Model 1 - Line currents prediction (40000 sim steps)*







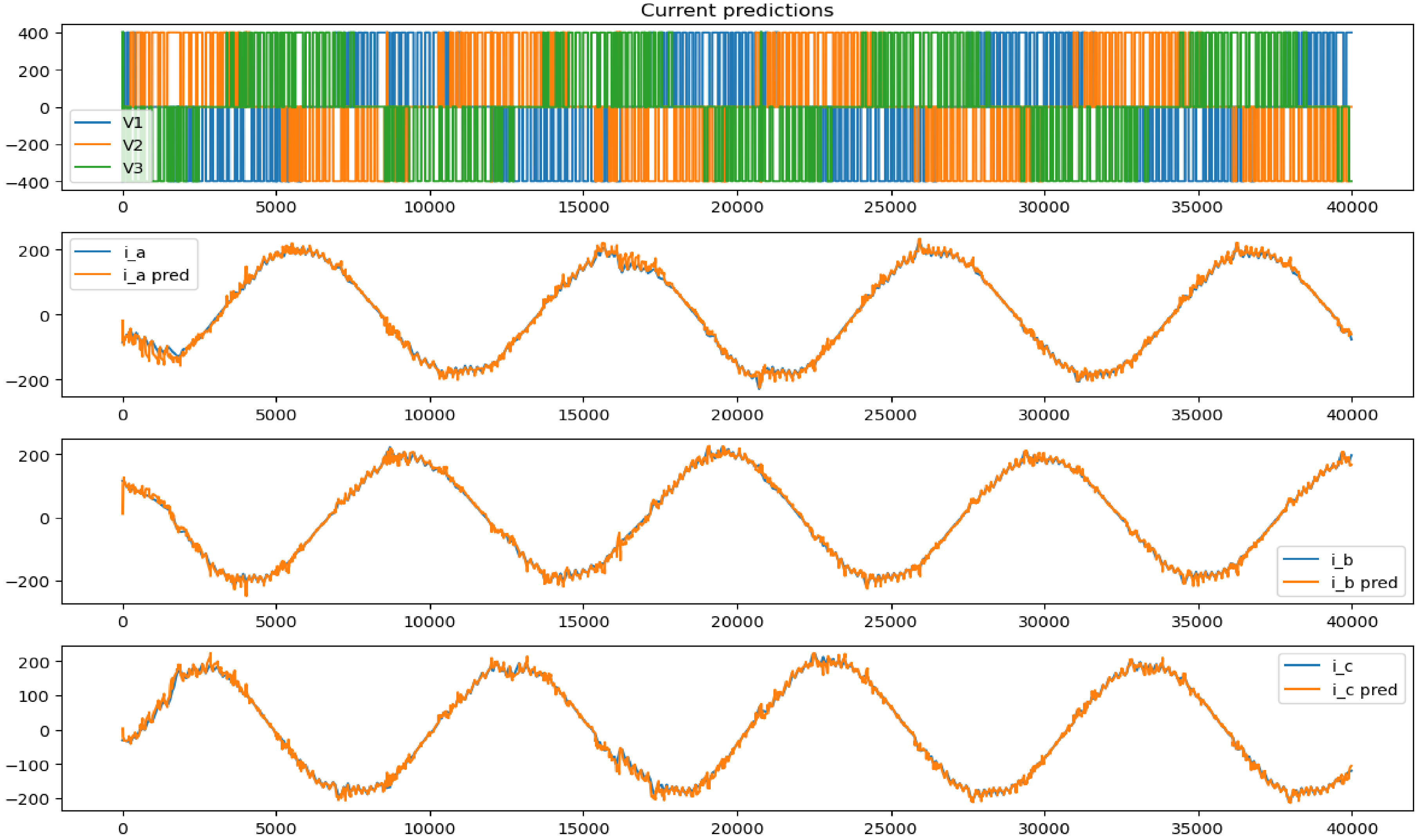
Normalised data - Test error (MAE): 0.0135, Test error (MSE): 0.00035
Denormalised data - Test error (MAE): 5.421, Test error (MSE): 57.213
Train time: 295 min, Inference time: 4 sec

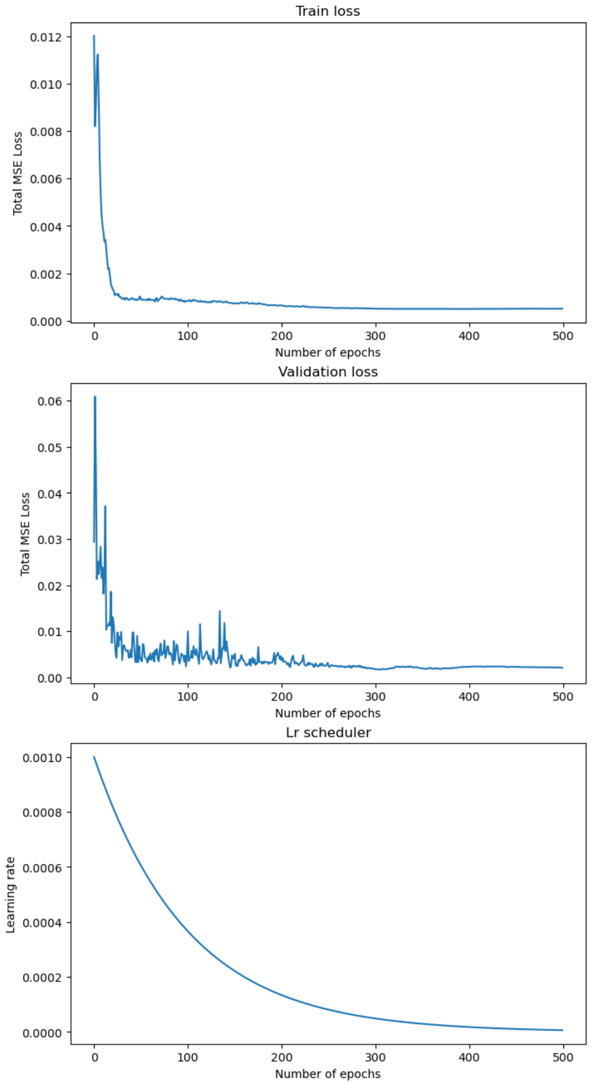
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

Modeling dynamic behavior of motor

- *NODE Model 2 – Line currents prediction*





Normalised data - Test error (MAE): 0.0134, Test error (MSE): 0.0004
Denormalised data - Test error (MAE): 5.395, Test error (MSE): 64.264

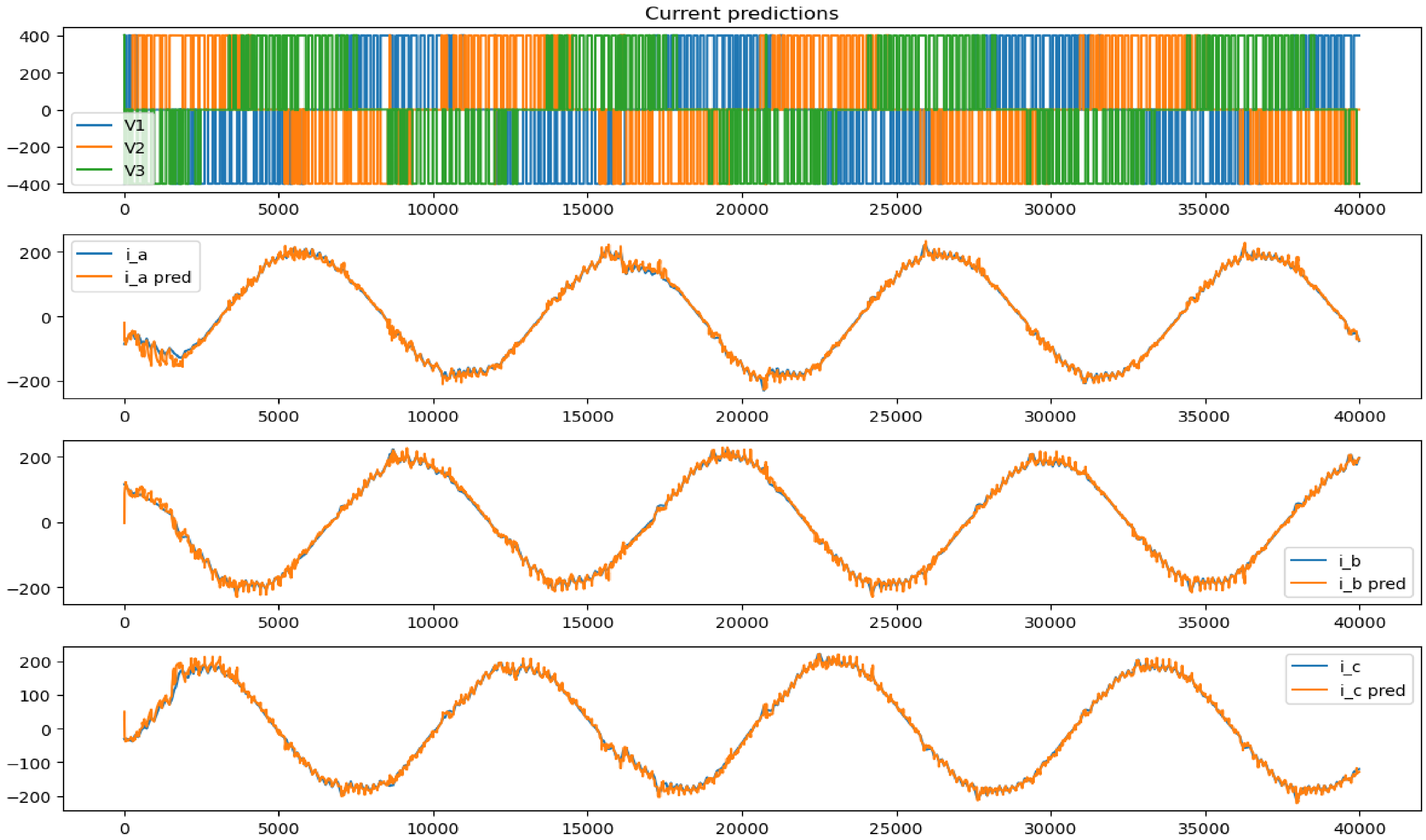
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

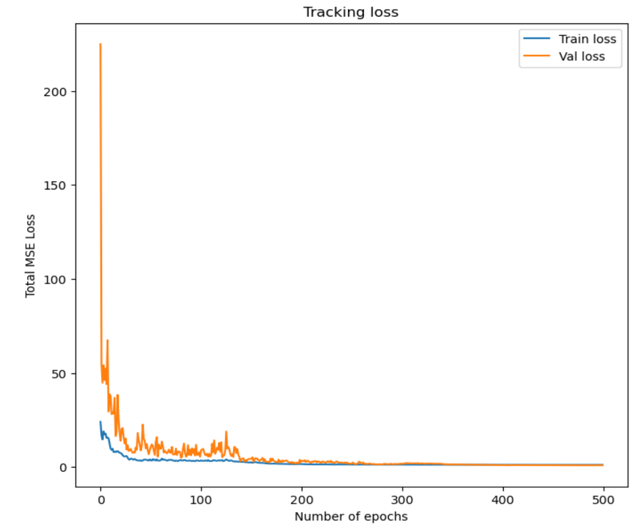
$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

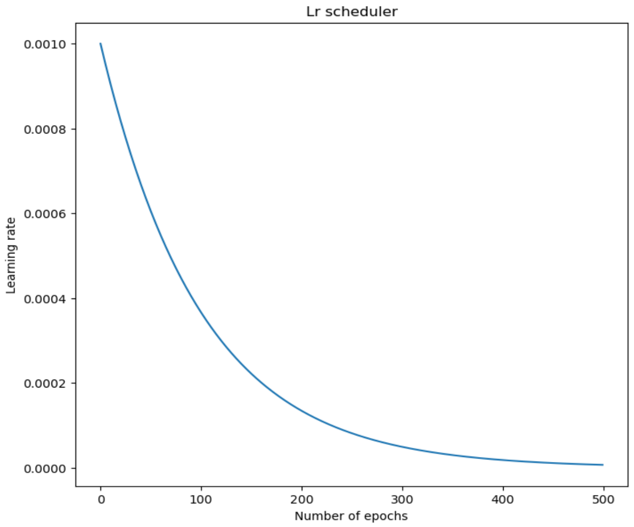
Modeling dynamic behavior of motor

- *NODE Model 2 – Line currents prediction (with tracking loss)*

Loss = instantaneous loss + 0.1*tracking loss







Normalised data - Test error (MAE): 0.0128, Test error (MSE): 0.00035
Denormalised data - Test error (MAE): 5.142, Test error (MSE): 56.597

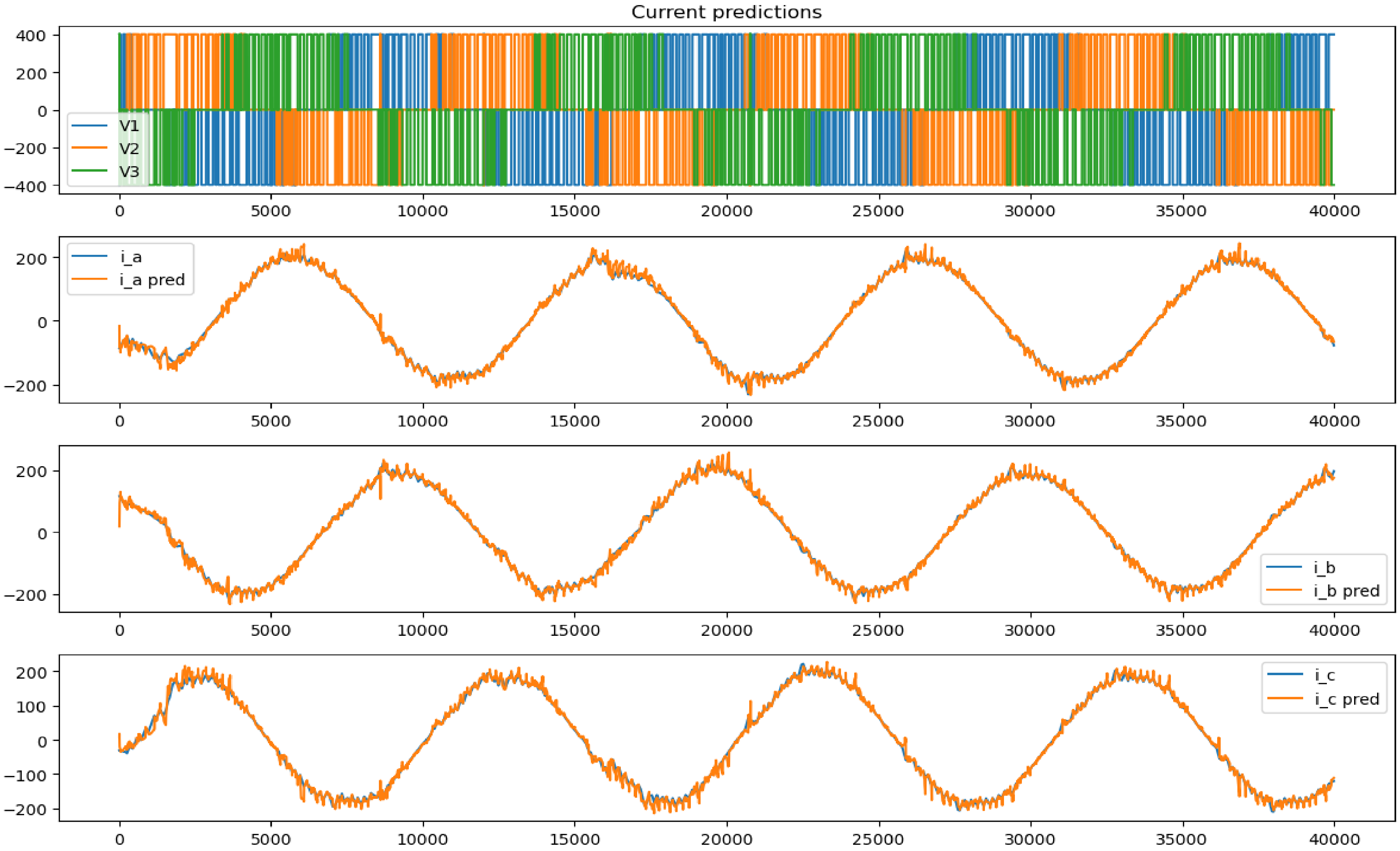
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

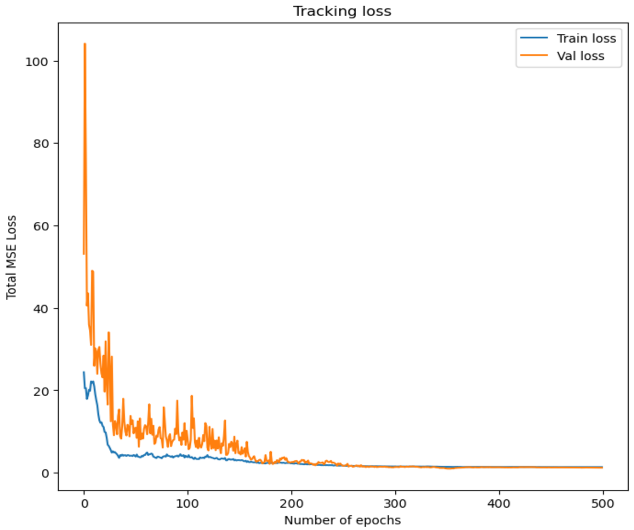
$W_1 \in R^{64\times 6}$, $W_2 \in R^{64\times 64}$, $W_3 \in R^{3\times 64}$

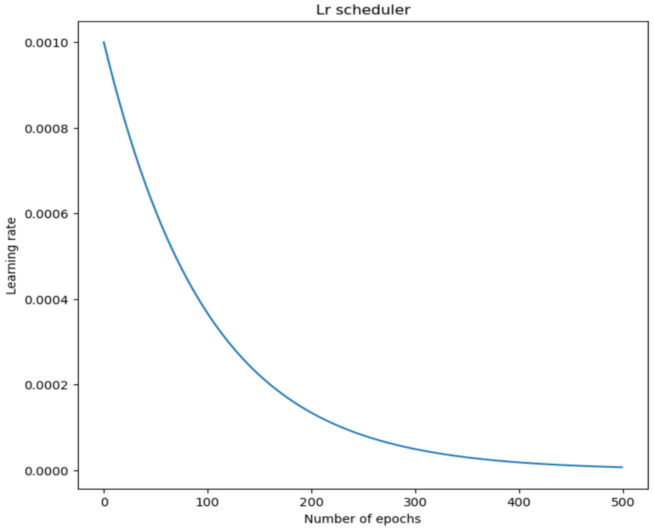
Modeling dynamic behavior of motor

- *NODE Model 3 – Line currents prediction (with tracking loss)*

Loss = instantaneous loss + 0.1*tracking loss







Normalised data - Test error (MAE): 0.0148, Test error (MSE): 0.00051
Denormalised data - Test error (MAE): 5.921, Test error (MSE): 82.096

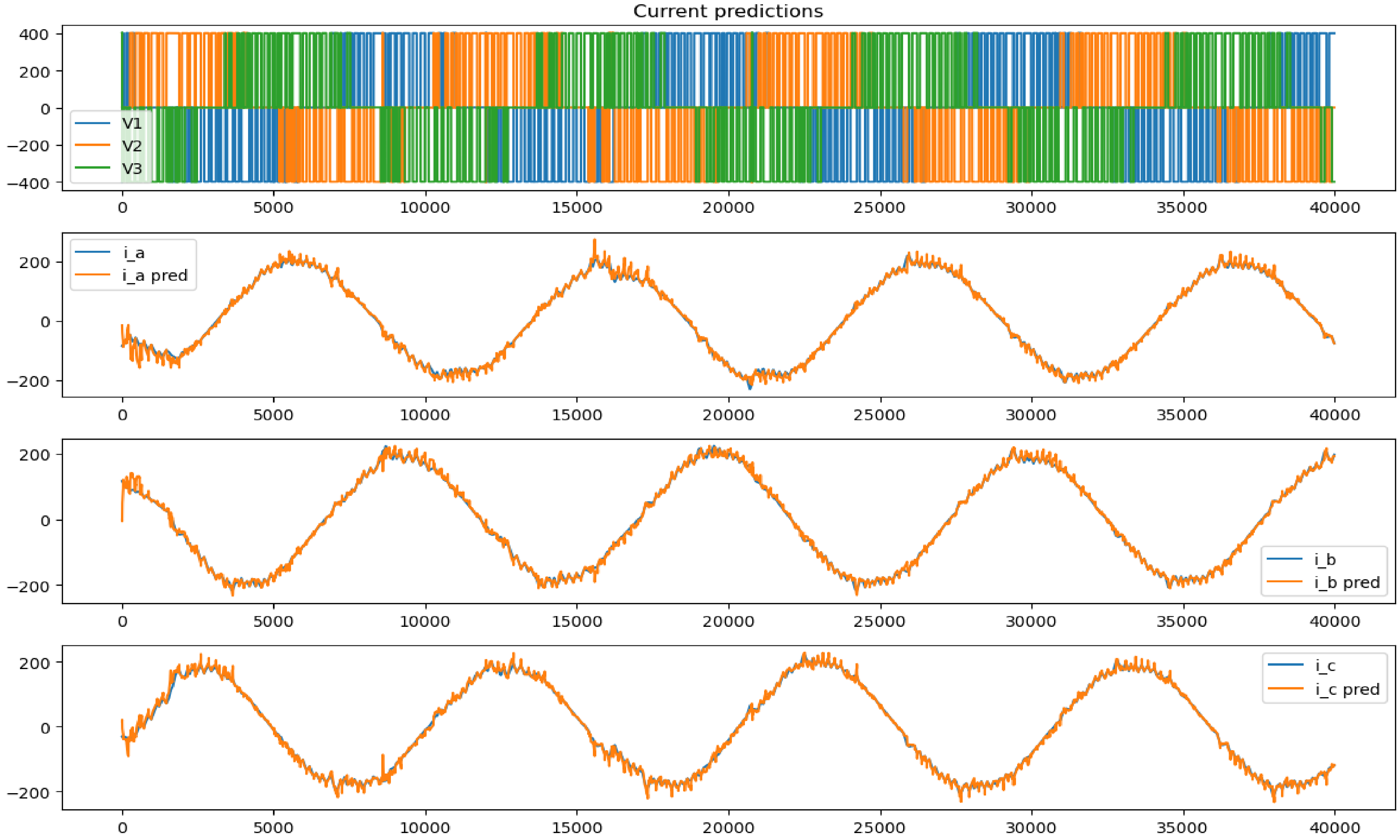
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

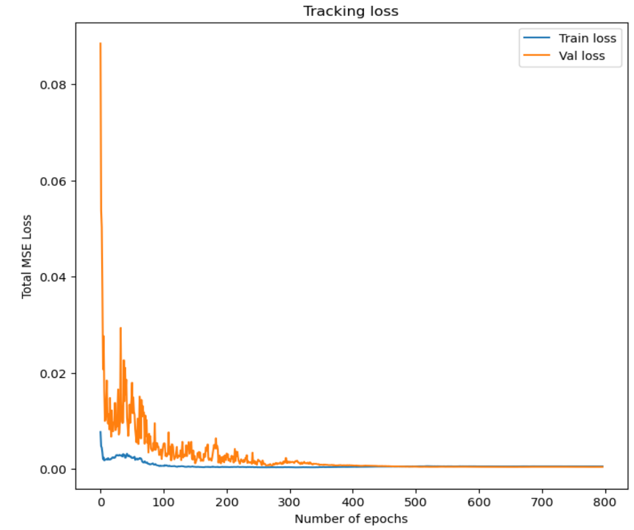
$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

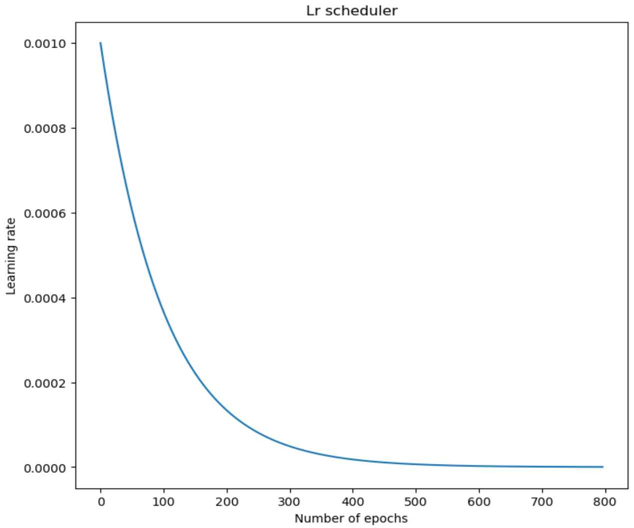
Modeling dynamic behavior of motor

- *NODE Model 2 – Line currents prediction(training with loop inside loop)*

prediction_horizon = 10







Normalised data - Test error (MAE): 0.013, Test error (MSE): 0.00042
Denormalised data - Test error (MAE): 5.226, Test error (MSE): 68.056

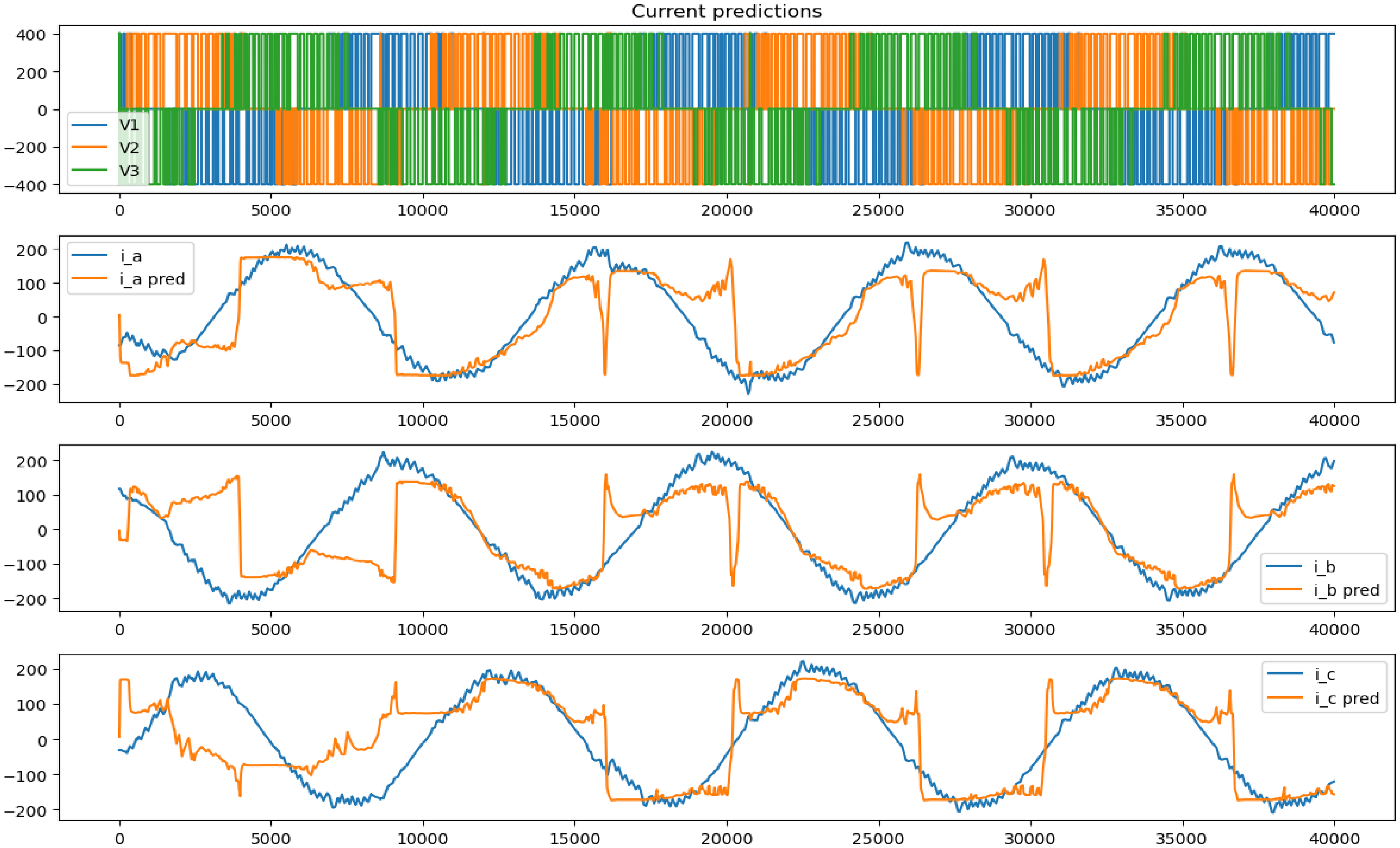
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

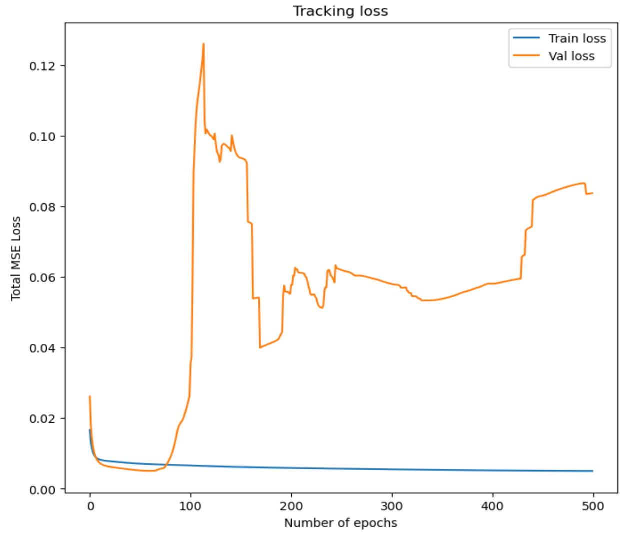
$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

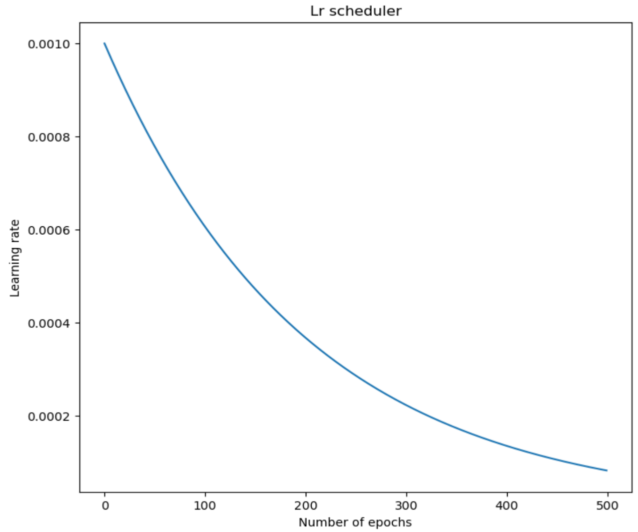
Modeling dynamic behavior of motor

- *NODE Model 2 – Line currents prediction* (*with chunks at each time step*)*- Overfitting*

prediction_horizon = 50







Normalised data - Test error (MAE): 0.1465, Test error (MSE): 0.0507
Denormalised data - Test error (MAE): 58.61, Test error (MSE): 8119.46

Epoch = 500, Train loss = 0.0045

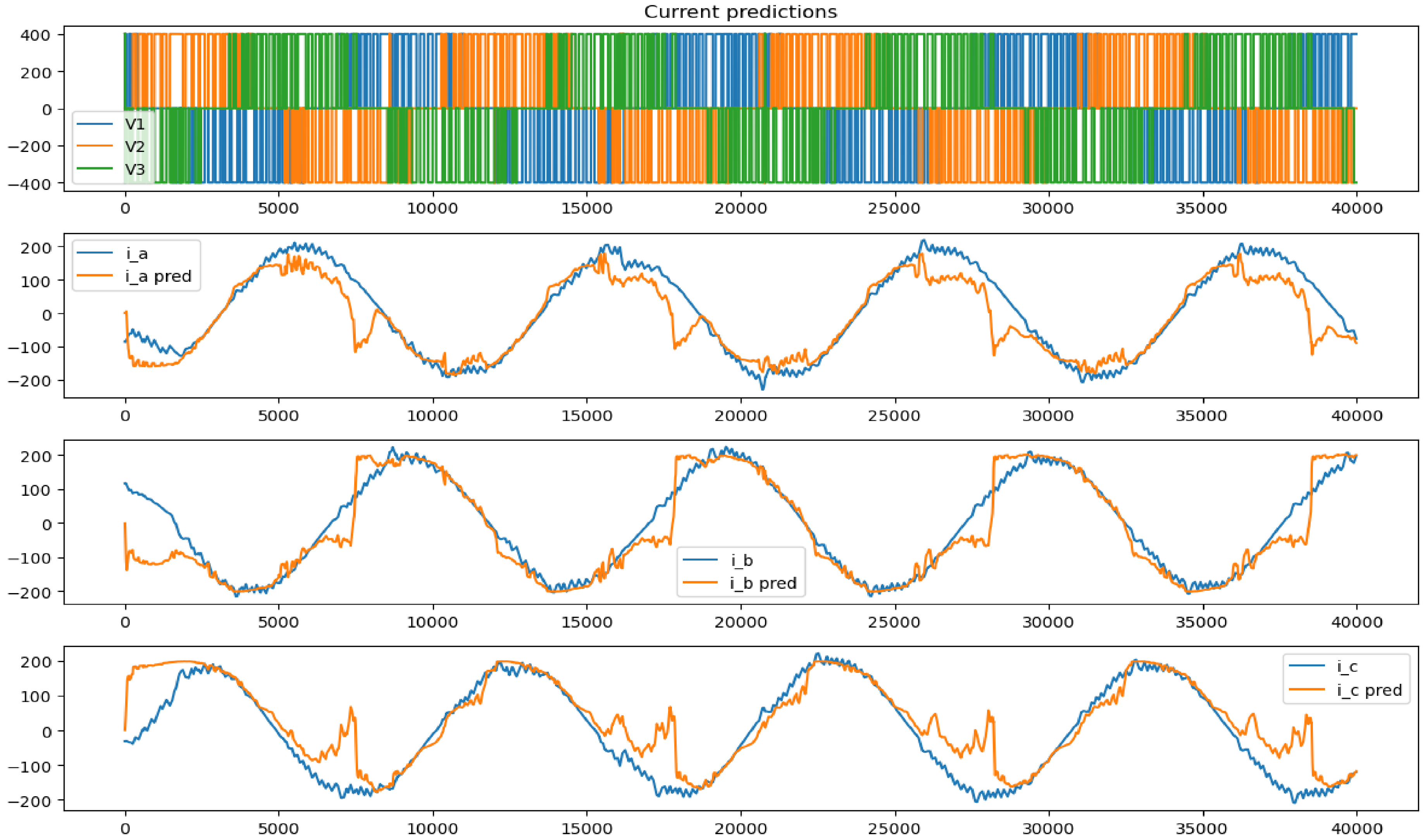
$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$

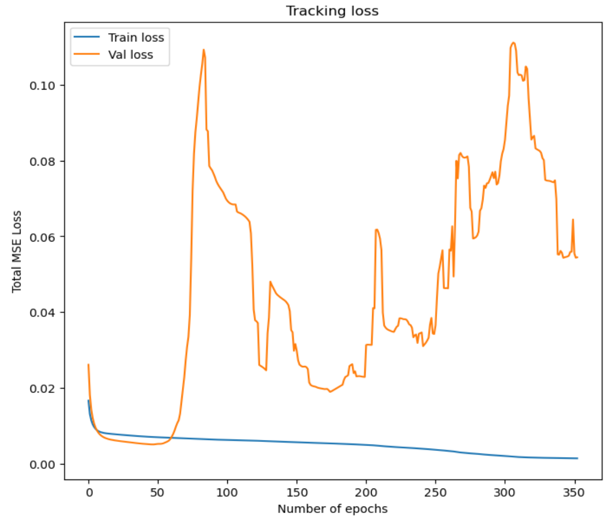
$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

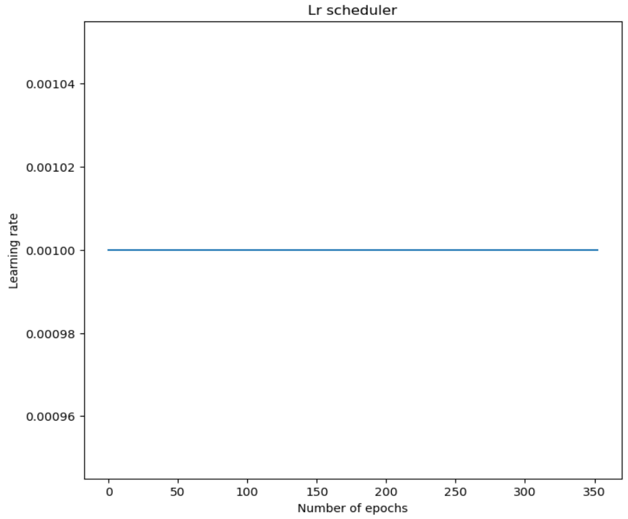
Modeling dynamic behavior of motor

- *NODE Model 2 – Line currents prediction* (*with chunks at each time step*)*- Overfitting*

prediction_horizon = 50







Normalised data - Test error (MAE): 0.0967, Test error (MSE): 0.0223
Denormalised data - Test error (MAE): 38.694, Test error (MSE): 3583.113

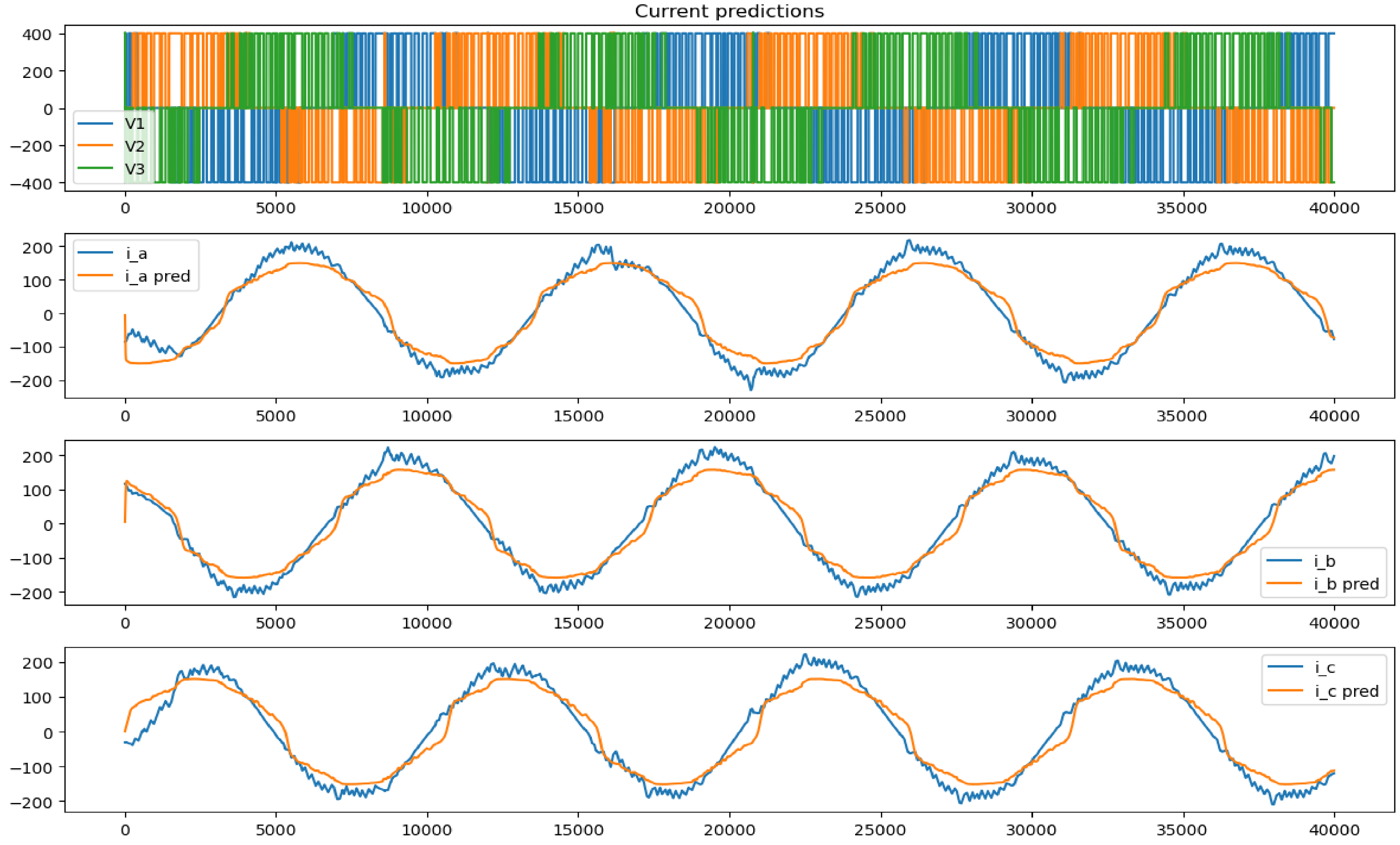
Epoch = 350, Train loss = 0.0014

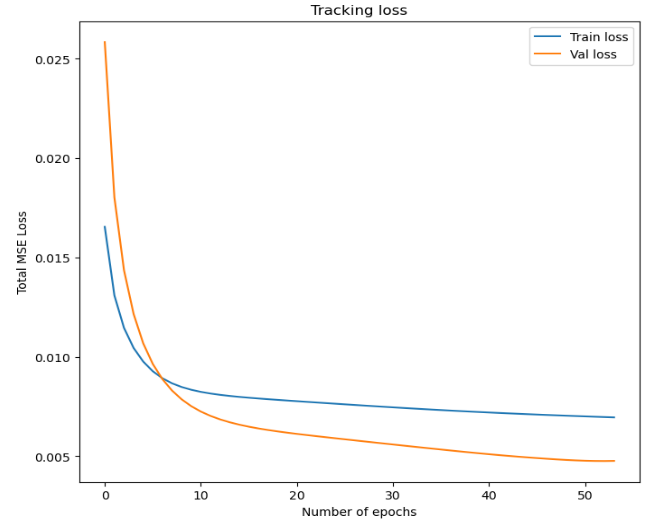
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$
$$W_1 \in R^{64\times 6},\ W_2 \in R^{64\times 64},\ W_3 \in R^{3\times 64}$$

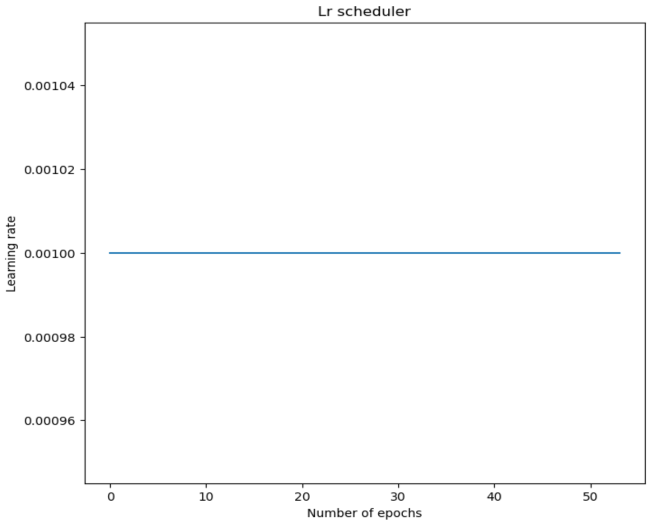
Modeling dynamic behavior of motor

- *NODE Model 2 – Line currents prediction (with chunks at each time step)- Early stopping*

prediction_horizon = 50







Normalised data - Test error (MAE): 0.0671, Test error (MSE): 0.0065
Denormalised data - Test error (MAE): 26.858, Test error (MSE): 1054.91

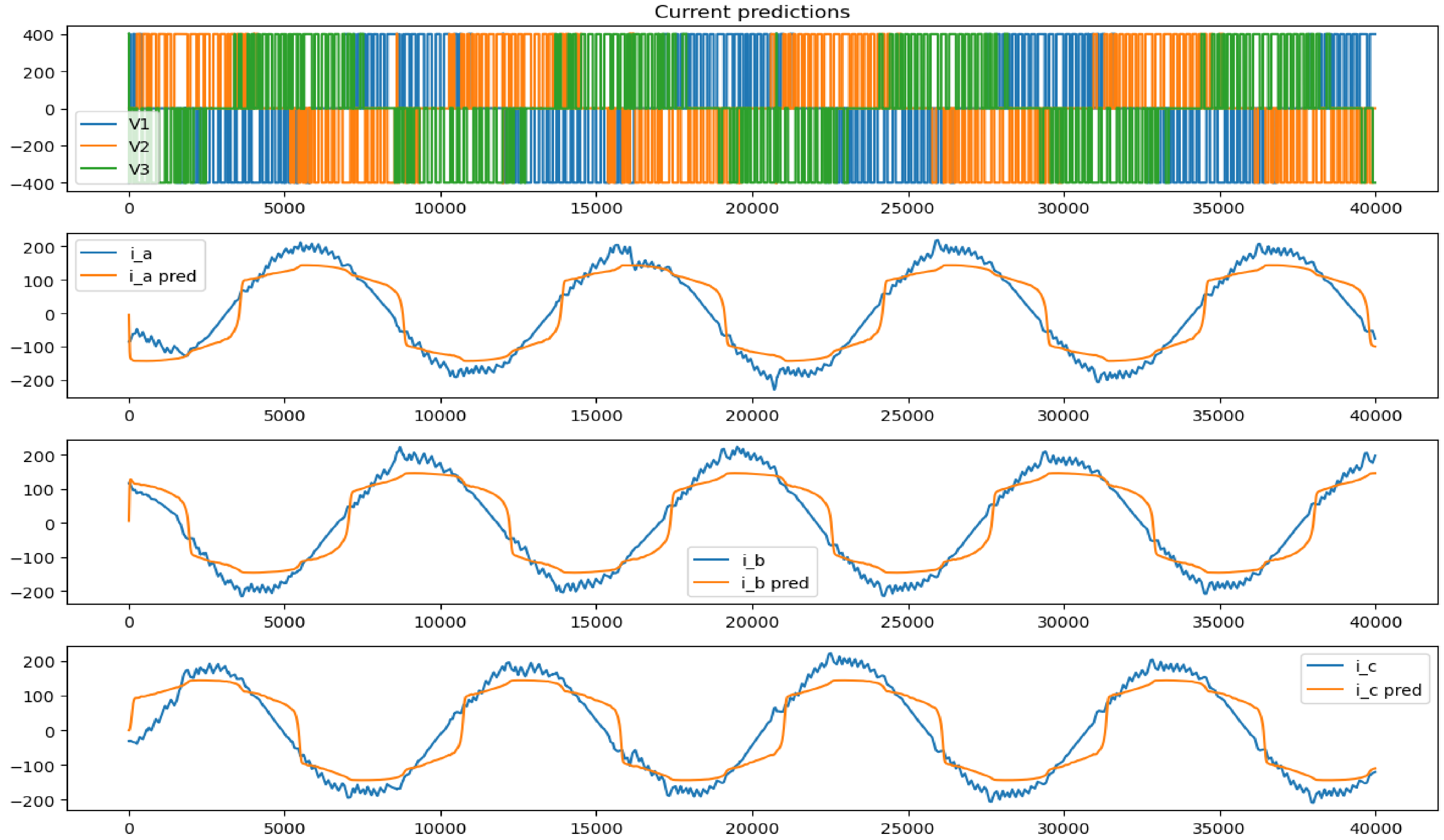
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

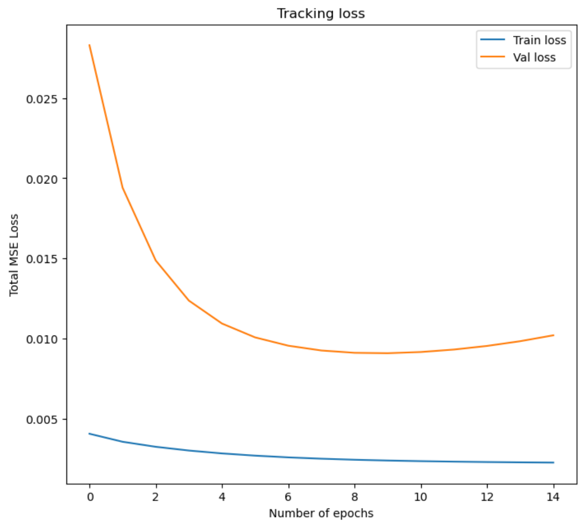
$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

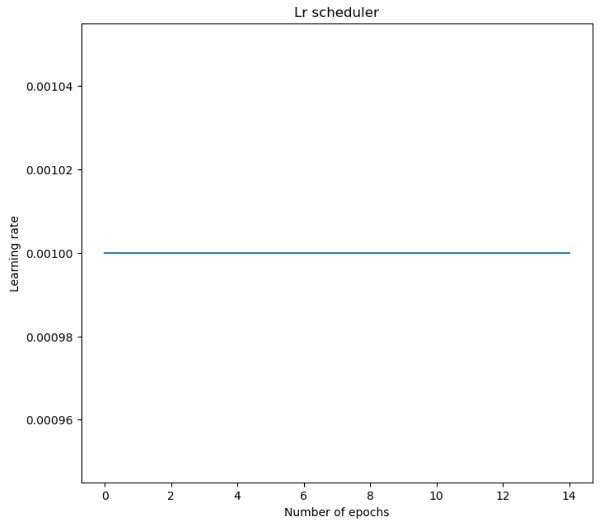
Modeling dynamic behavior of motor

- *NODE Model 2 – Line currents prediction (with chunks at each time step)- Early stopping*

prediction_horizon = 10







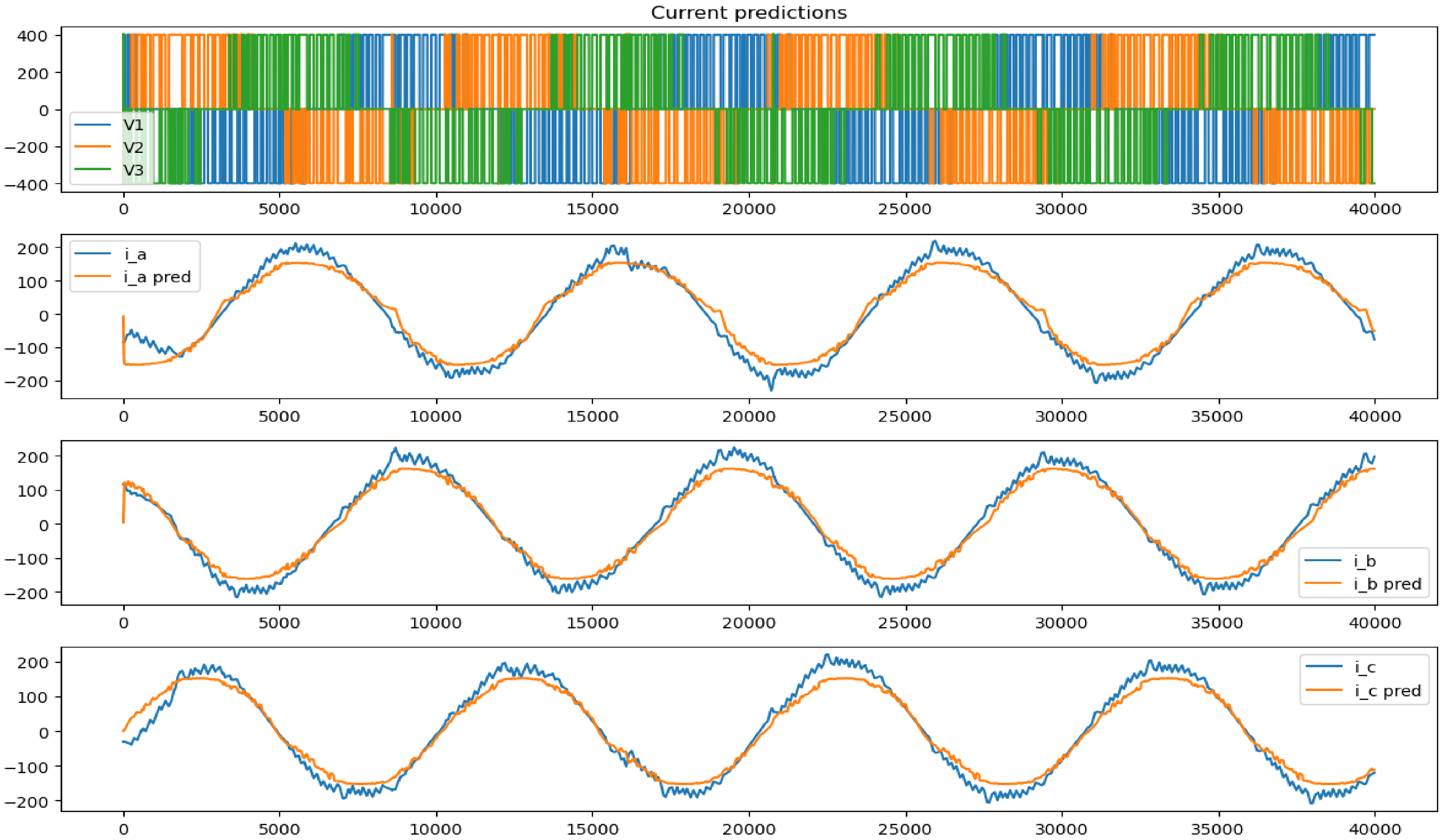
Normalised data - Test error (MAE): 0.0946, Test error (MSE): 0.0127
Denormalised data - Test error (MAE): 37.87, Test error (MSE): 2032.59

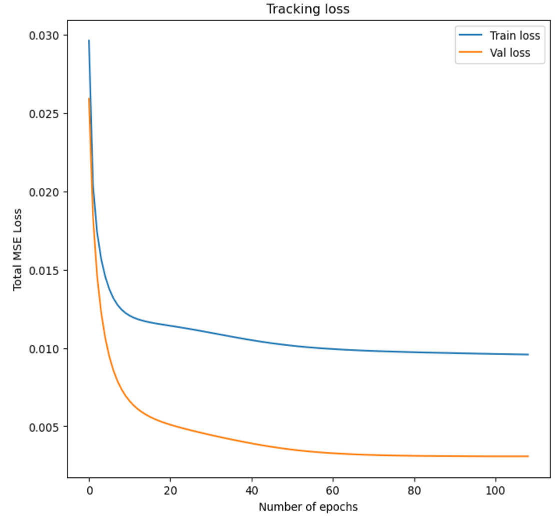
$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$
$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

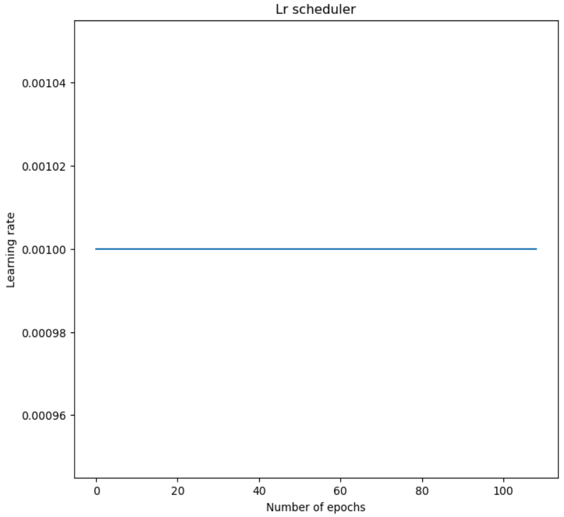
Modeling dynamic behavior of motor

- *NODE Model 2 – Line currents prediction (with chunks at each time step)- Early stopping*

prediction_horizon = 200







Normalised data - Test error (MAE): 0.0569, Test error (MSE): 0.0047
Denormalised data - Test error (MAE): 22.79, Test error (MSE): 760.454

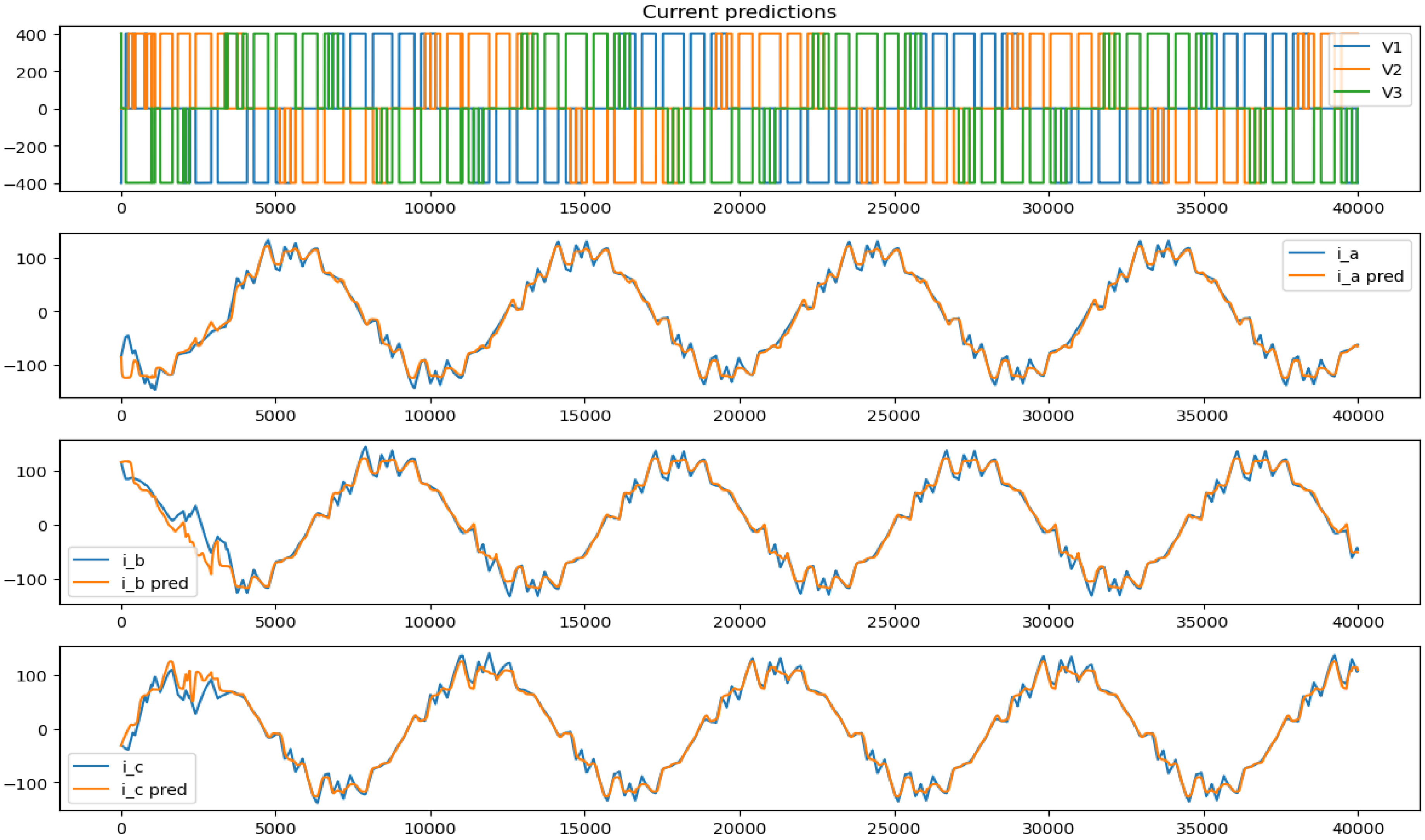
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

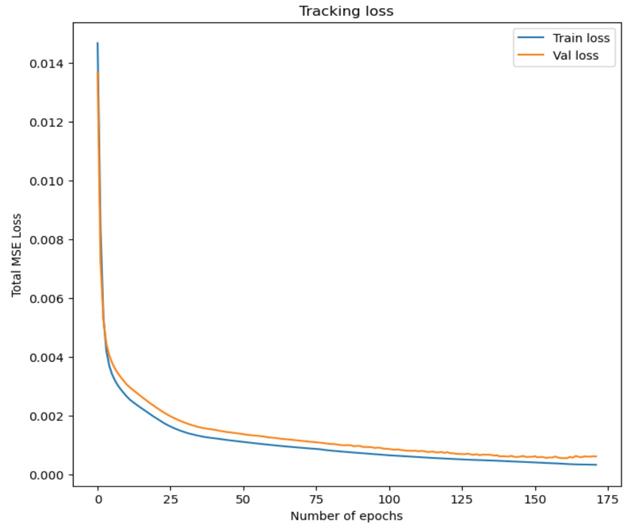
$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

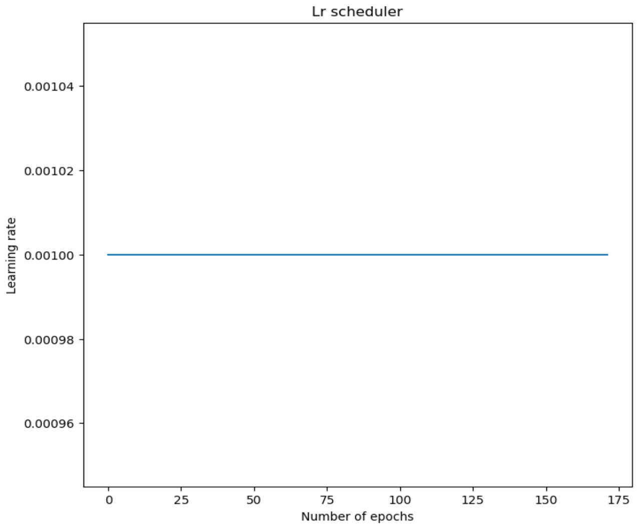
Modeling dynamic behavior of motor

- *NODE Model 2 – Training single OP(with chunks at each time step – early stopping)*

prediction_horizon = 500







Normalised data - Test error (MAE): 0.0142, Test error (MSE): 0.00059
Denormalised data - Test error (MAE): 5.610, Test error (MSE): 95

Comment: Trained with only one OPP. Made datasets with chunk size of prediction_horizon at each time steps in entire simulation.

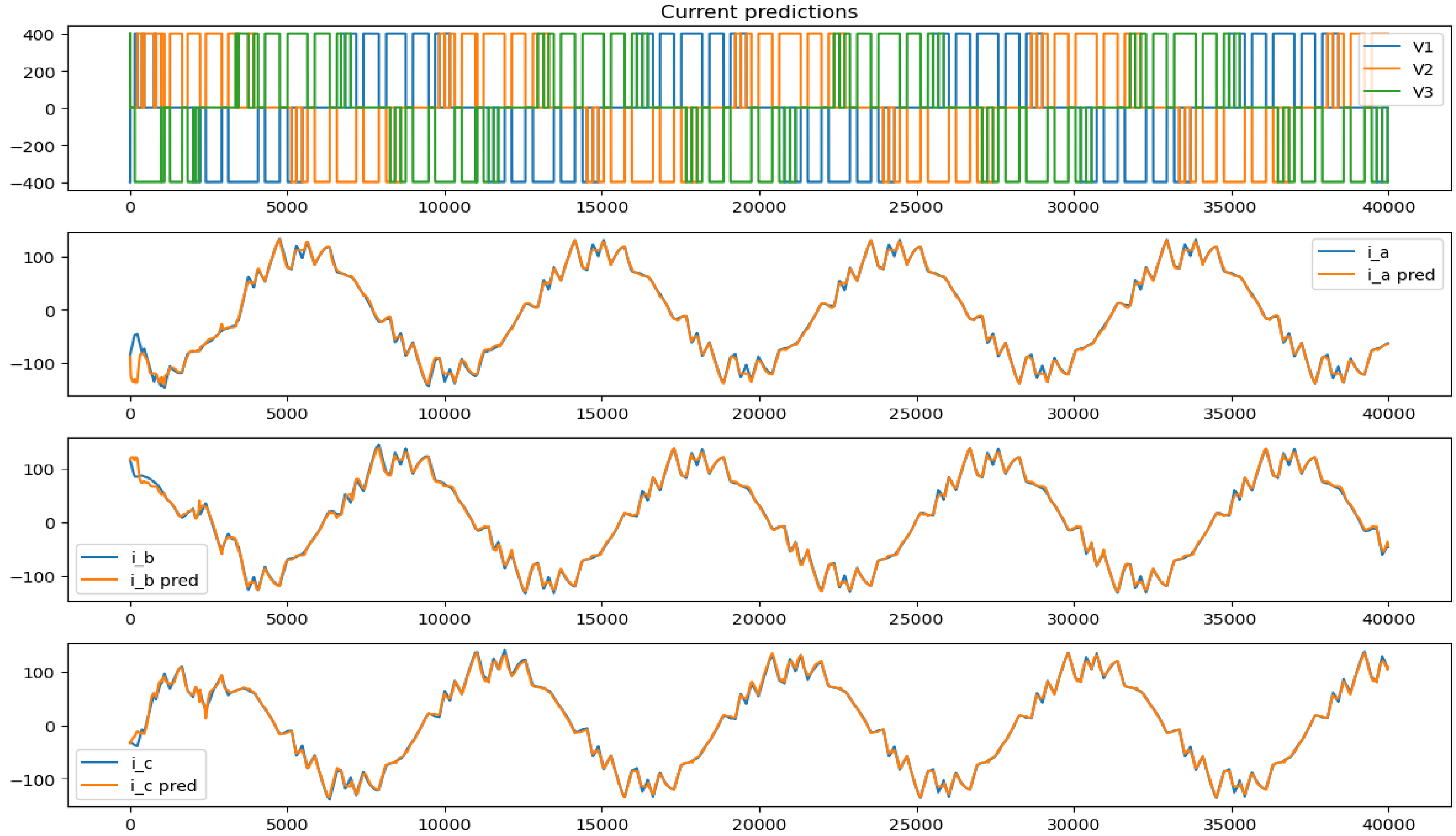
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

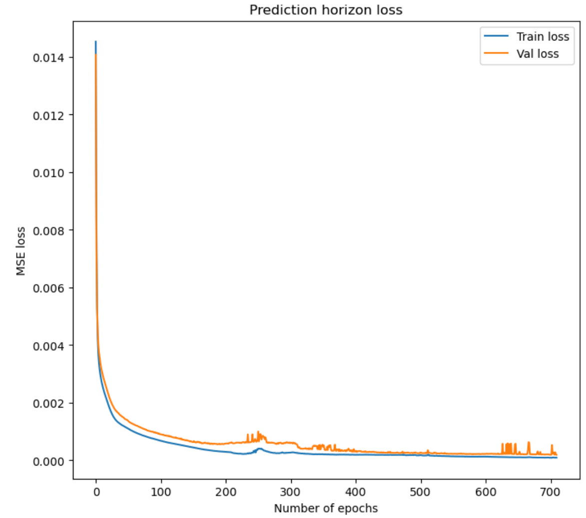
$W_1 \in R^{64\times 6}$, $W_2 \in R^{64\times 64}$, $W_3 \in R^{3\times 64}$

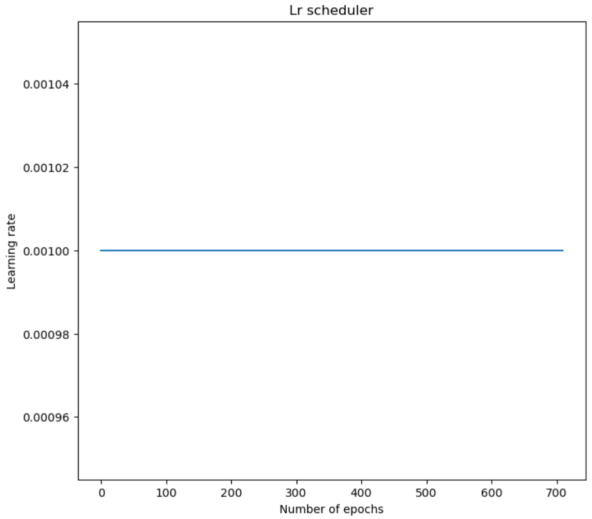
Modeling dynamic behavior of motor

- *NODE Model 2 – Training single OP(with chunks at each time step – without early stopping)*

prediction_horizon = 500







Normalised data - Test error (MAE): 0.00619, Test error (MSE): 0.00015
Denormalised data - Test error (MAE): 2.478, Test error (MSE): 24.786

Comment: Trained with only one OP. Made datasets with chunk size of prediction_horizon at each time steps in entire simulation.

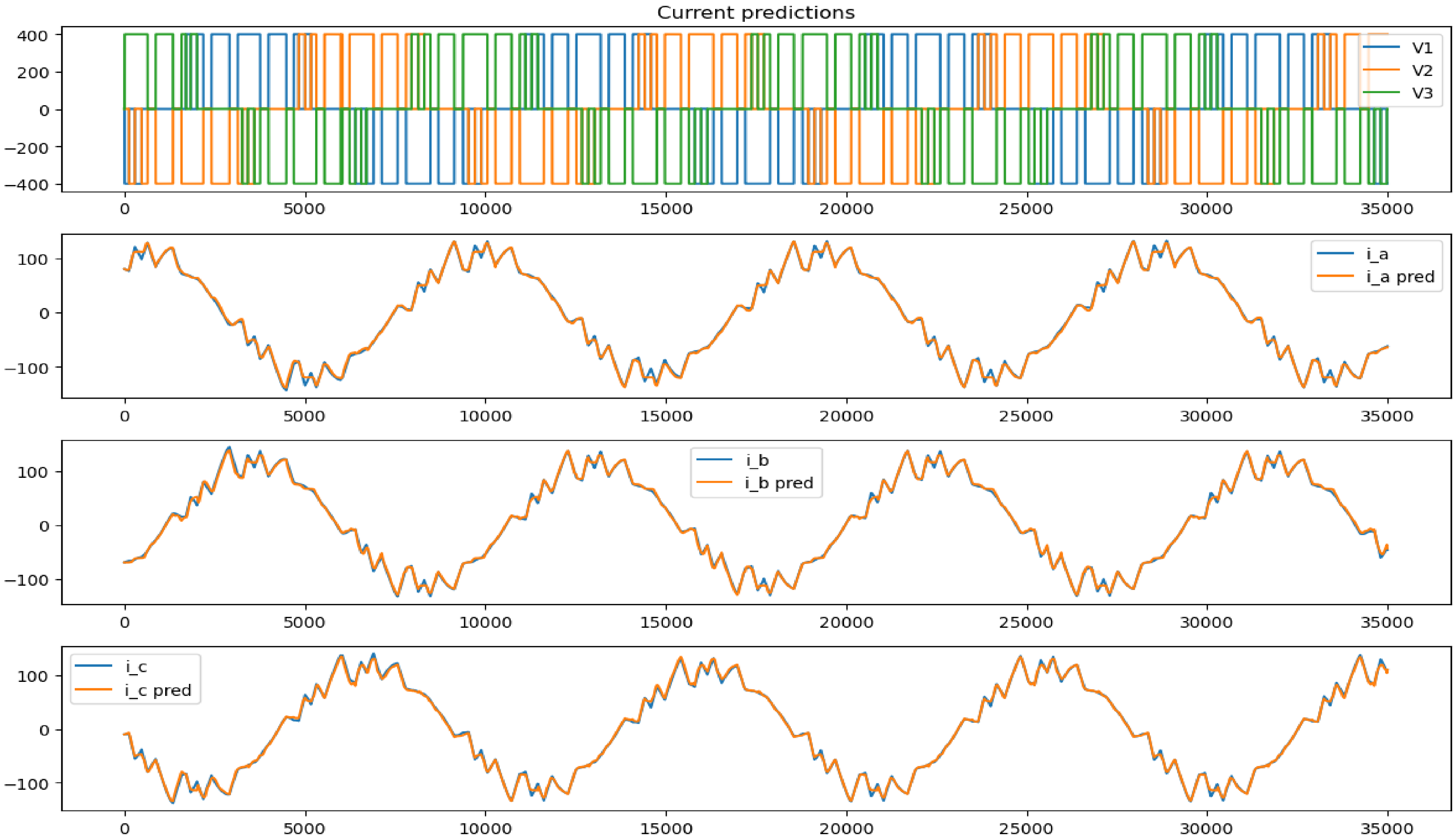
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

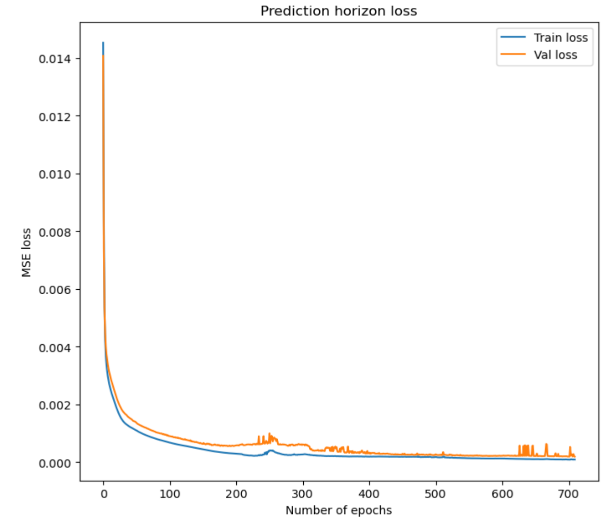
$W_1 \in R^{64\times 6}$, $W_2 \in R^{64\times 64}$, $W_3 \in R^{3\times 64}$

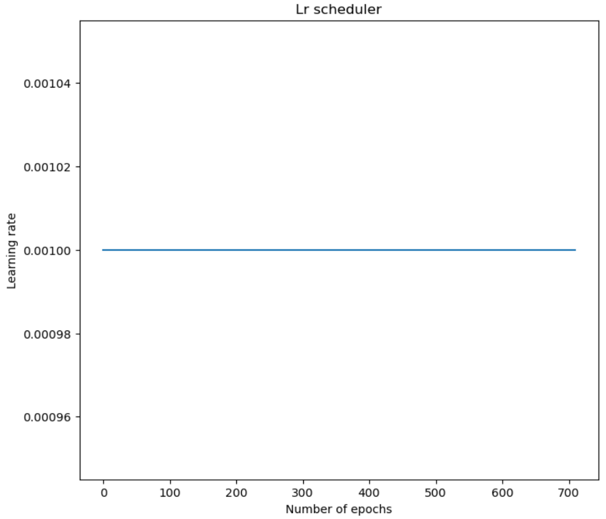
Modeling dynamic behavior of motor

- *NODE Model 2 – Training single OP(with chunks at each time step – without early stopping)*

prediction_horizon = 500







Normalised data - Test error (MAE): 0.00521, Test error (MSE): 0.000052
Denormalised data - Test error (MAE): 2.087, Test error (MSE): 8.384

Comment: Only prediction from 5000 to 40000 simulation indices of an OP.

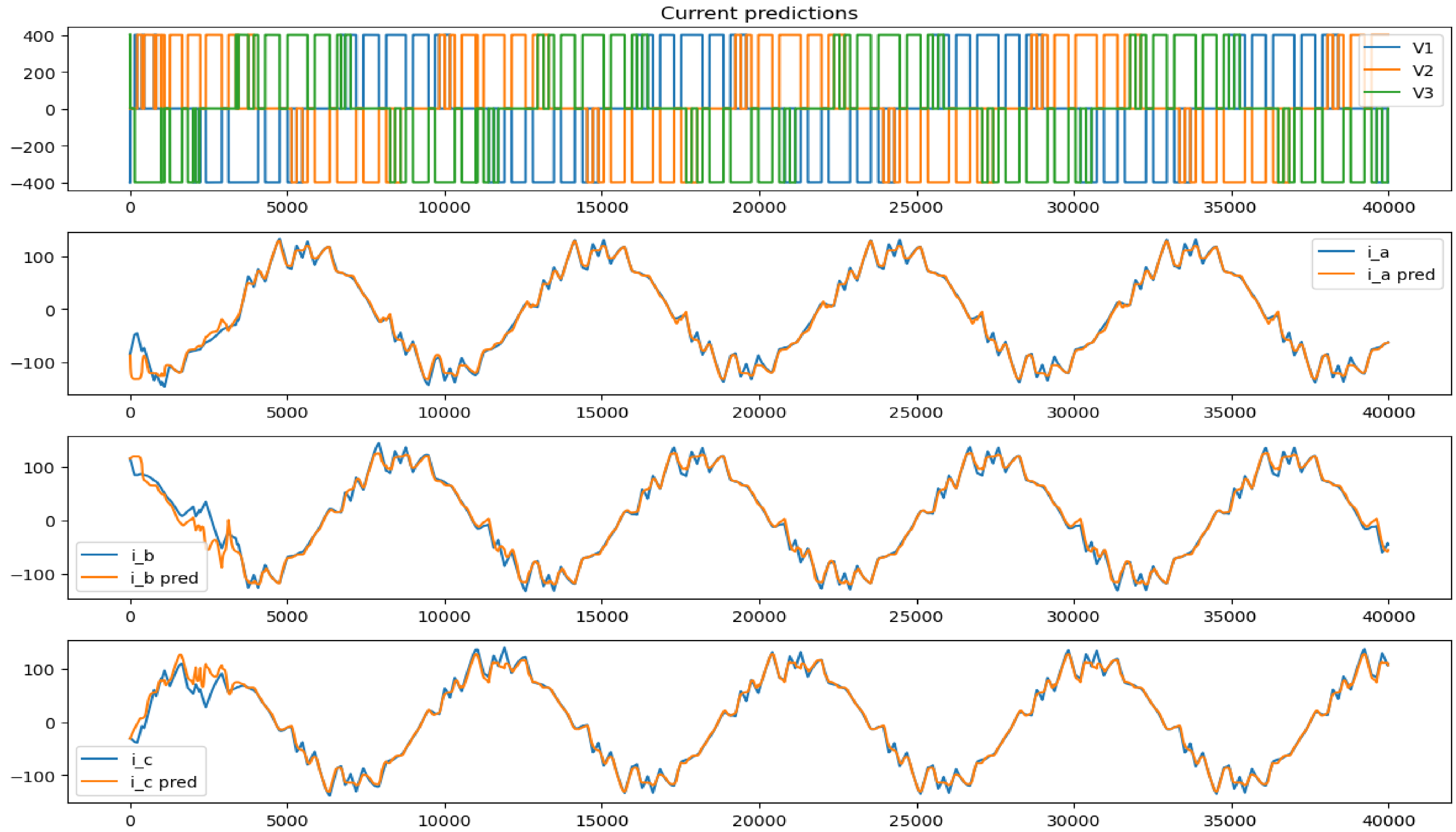
$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$

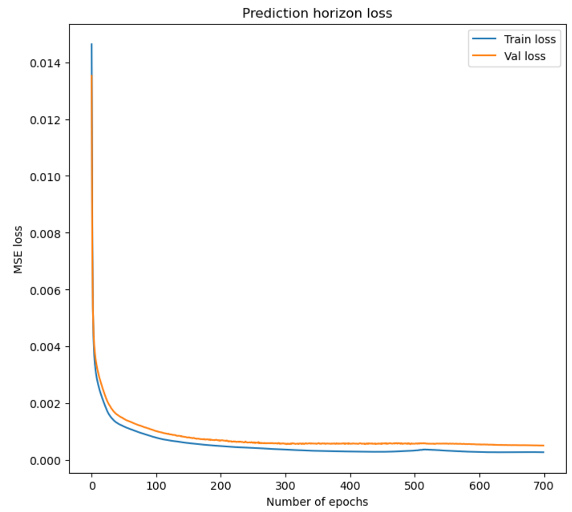
$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

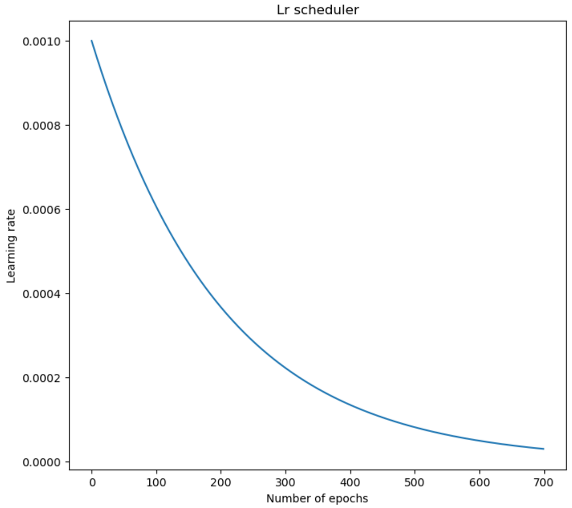
Modeling dynamic behavior of motor

- *NODE Model 2 – Training single OP(with chunks at each time step – without early stopping)*

prediction_horizon = 500







Normalised data - Test error (MAE): 0.0117, Test error (MSE): 0.00048
Denormalised data - Test error (MAE): 4.708, Test error (MSE): 77.586

Comment: Trained with decay of learning rate. Rest all parameters are same.

Optimiser without decay of learning rate is better

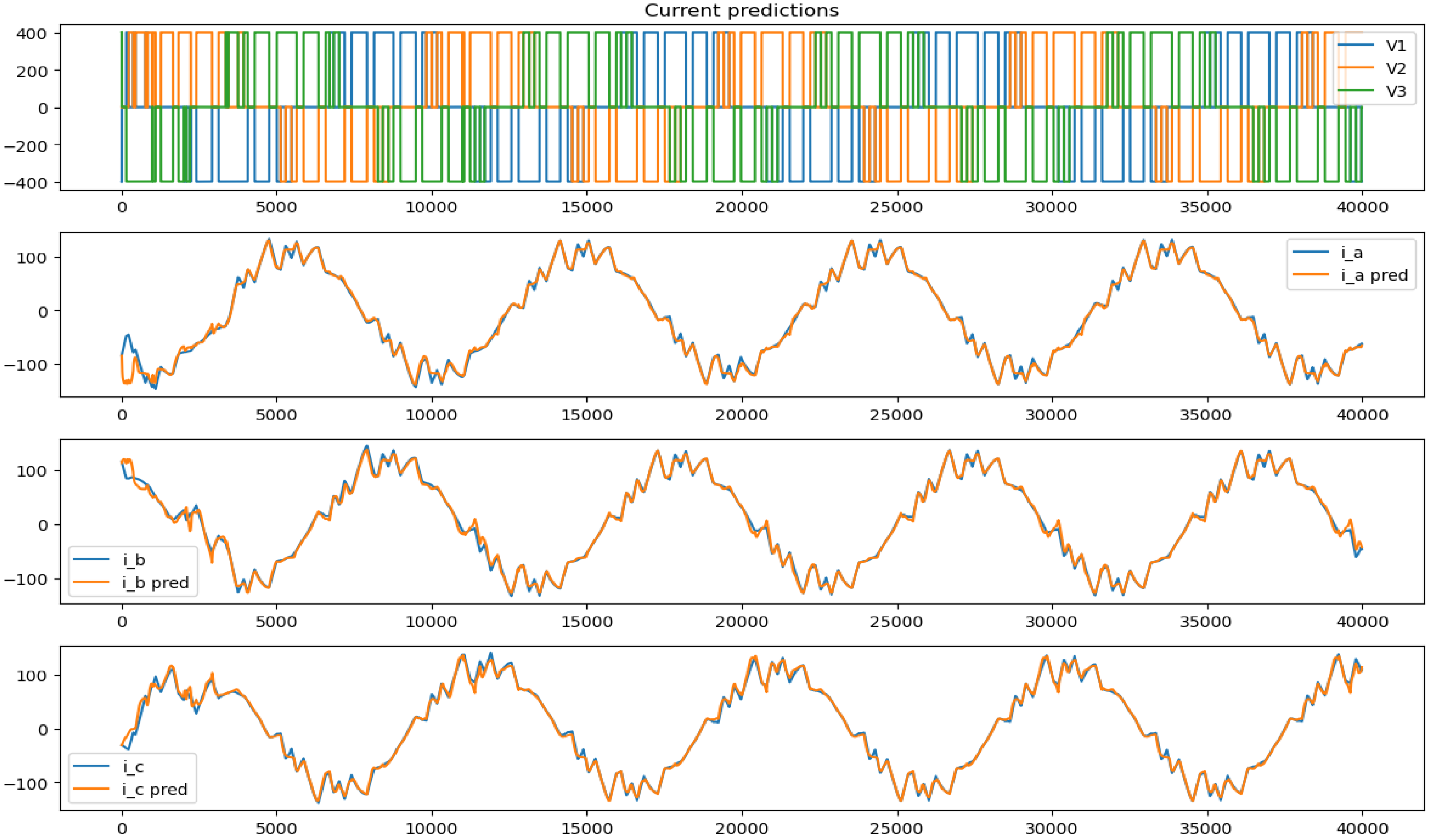
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

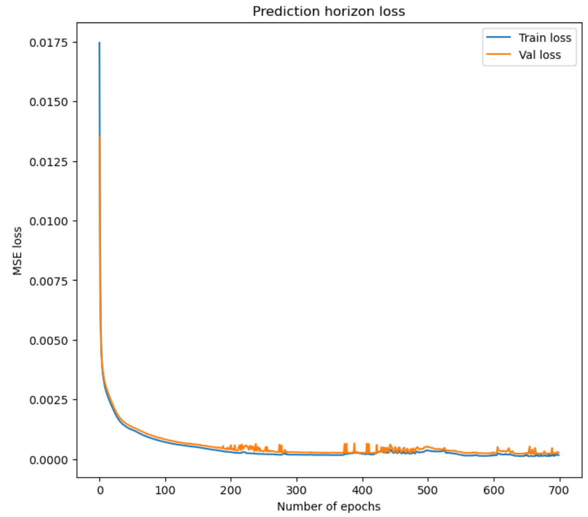
$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

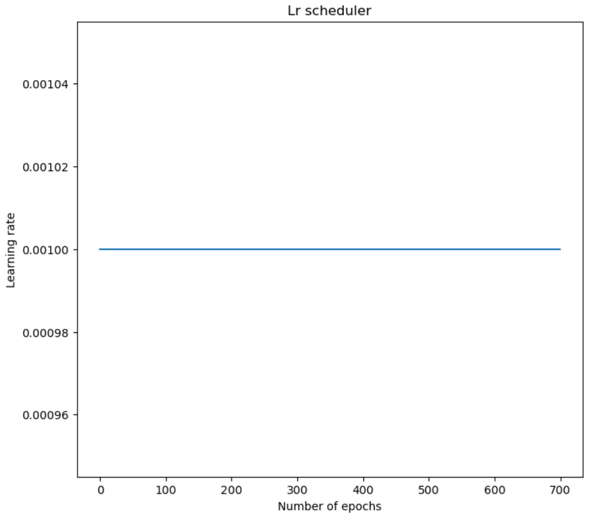
Modeling dynamic behavior of motor

- *NODE Model 2 – Training single OP(with chunks at each time step – without early stopping)*

prediction_horizon = 1000







Normalised data - Test error (MAE) :0.0085, Test error (MSE): 0.00025
Denormalised data - Test error (MAE): 3.417, Test error (MSE): 40.926

Comment: Trained with only one OP. Made datasets with chunk size of prediction_horizon at each time steps in entire simulation.

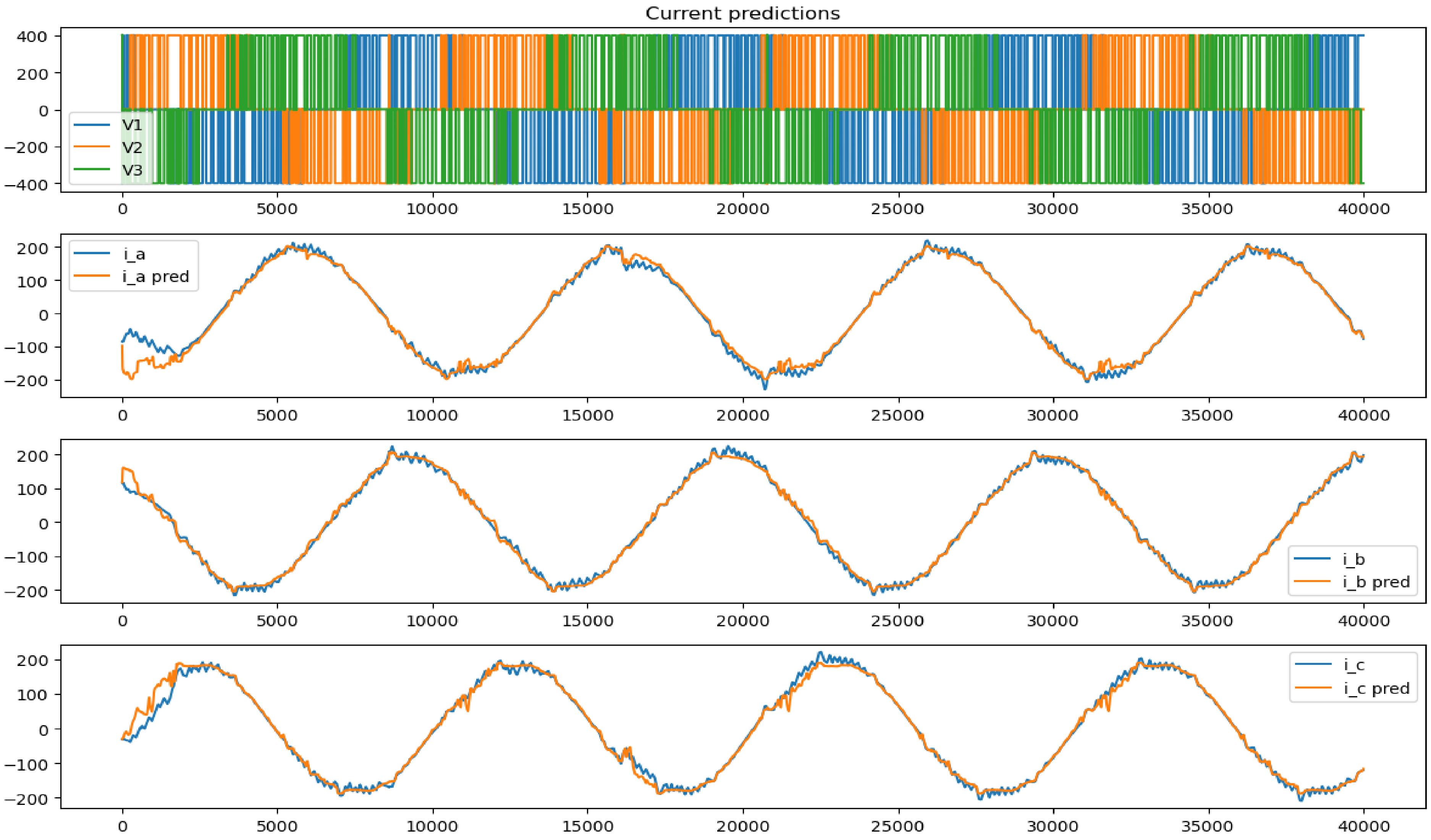
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

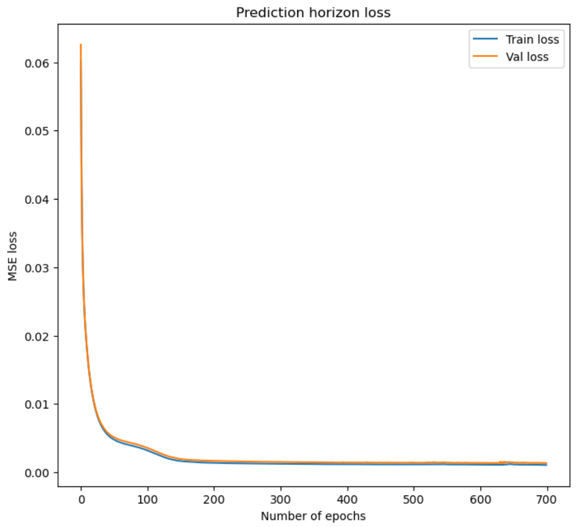
$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

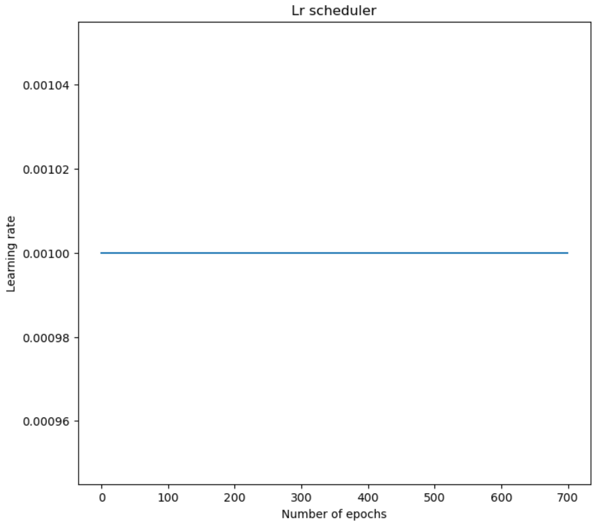
Modeling dynamic behavior of motor

- *NODE Model 2 – Training single OP-2(with chunks at each time step – without early stopping)*

prediction_horizon = 500







Normalised data - Test error (MAE) : 0.021, Test error (MSE): 0.00135
Denormalised data - Test error (MAE): 6.426, Test error (MSE): 180.149

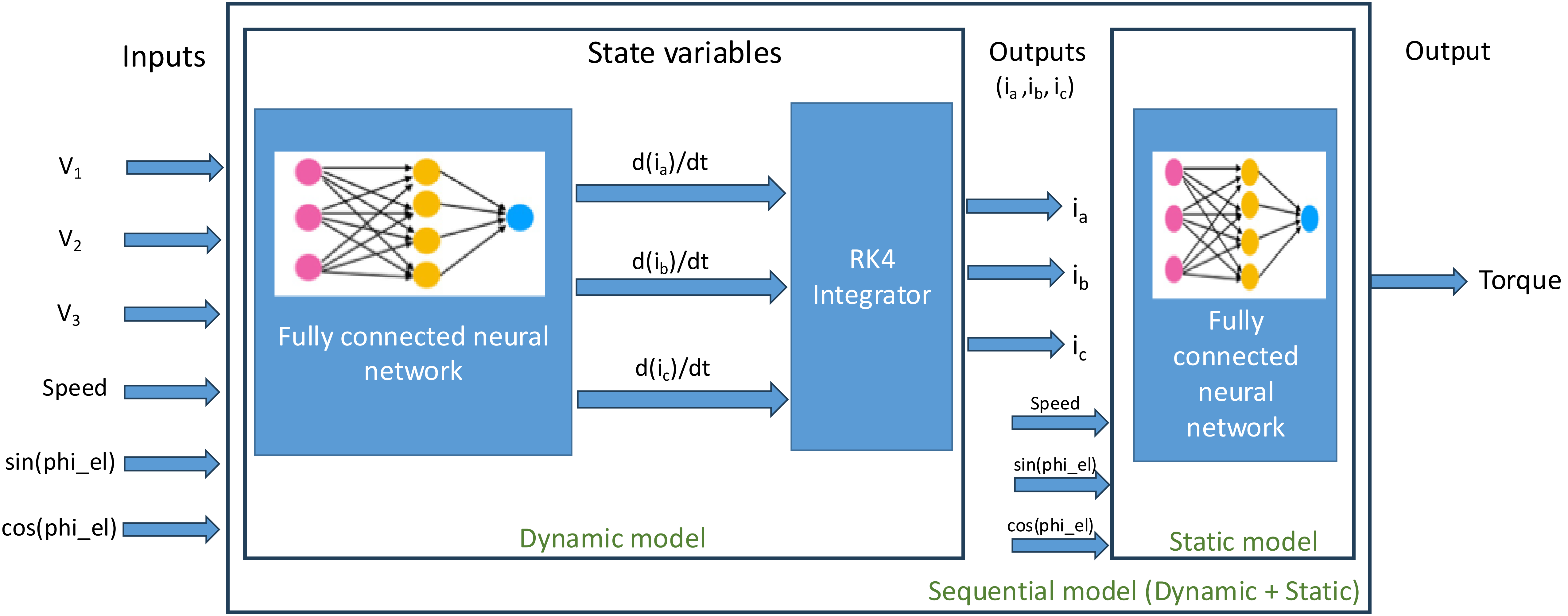
Comment: Trained with only one OP-2. Made datasets with chunk size of prediction horizon at each time steps in entire simulation.

$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$
$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

Combined Model (Dynamic + Static)

Combined Model (Dynamic + Static)

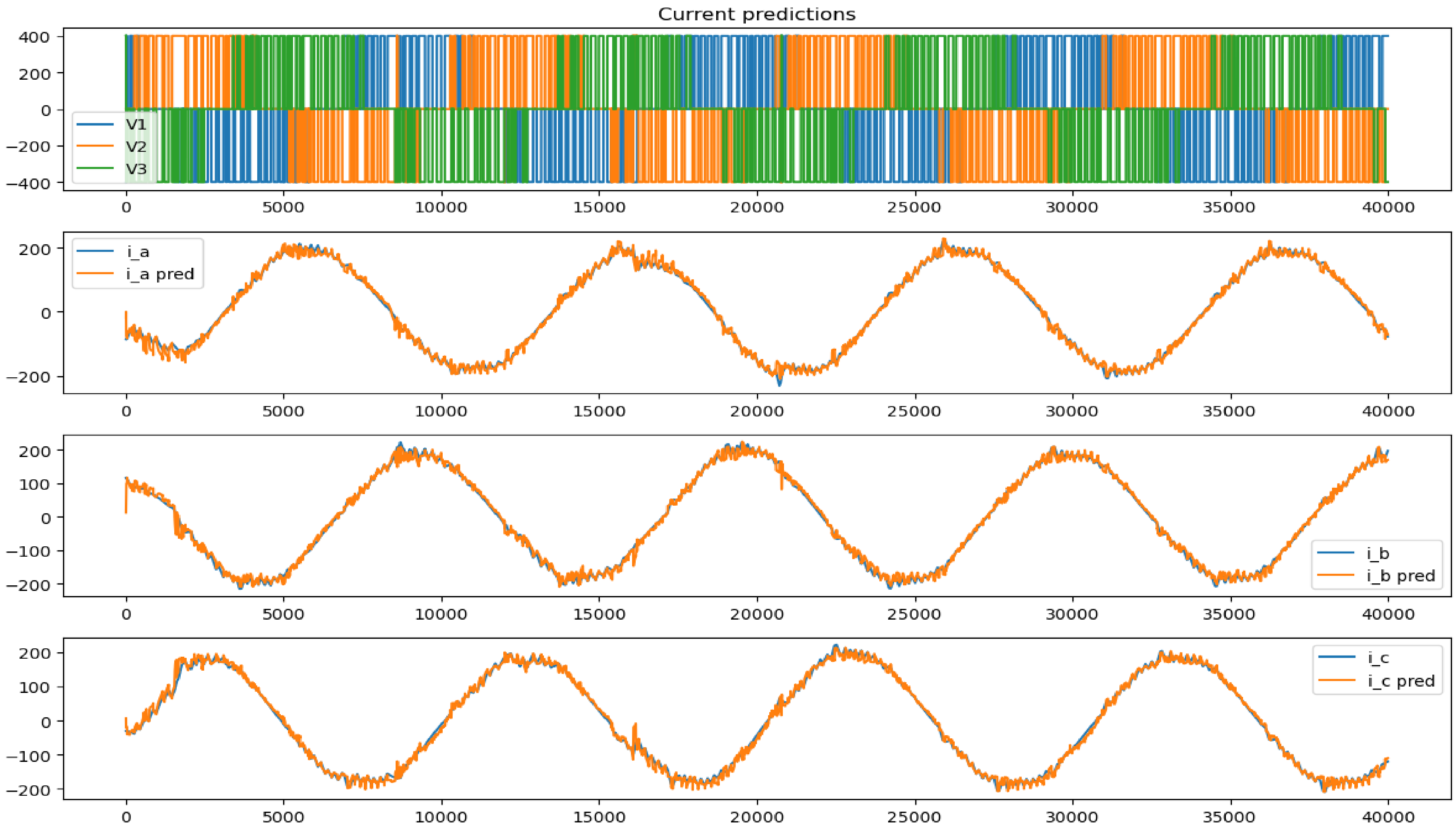
- Architecture

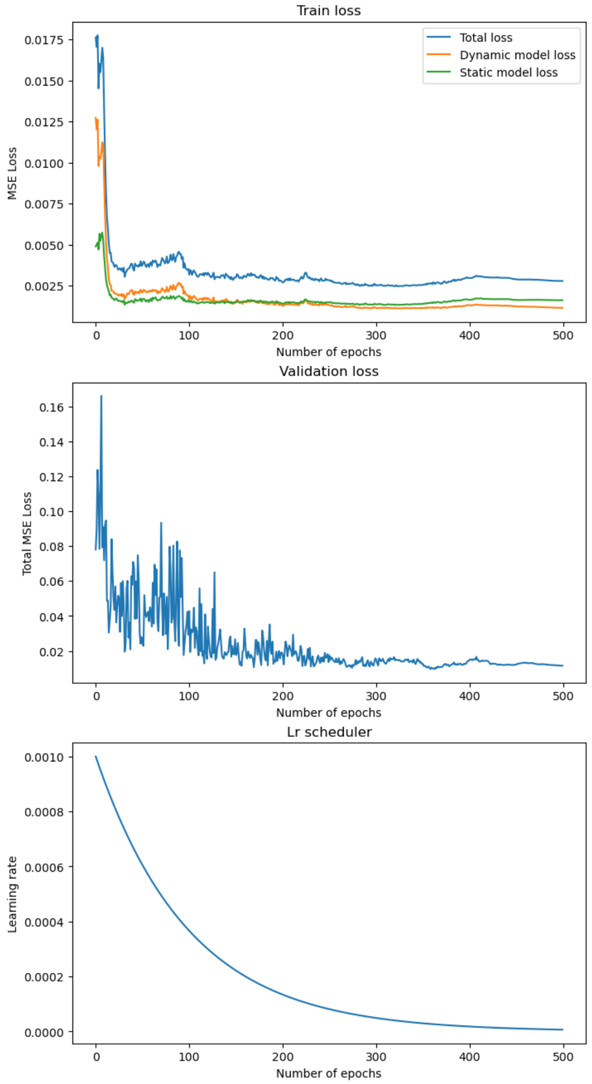


- combined_loss = dynamic_mseloss + static_mseloss (static loss will contribute after crossing fixed number or epochs)
- combined_loss = weight_1*dynamic_mseloss + weight_2 *static_mseloss (weight_1 & weight_2 changes depending on epochs)

Combined Model (Dynamic + Static)

- Parallel training - Dynamic model predictions

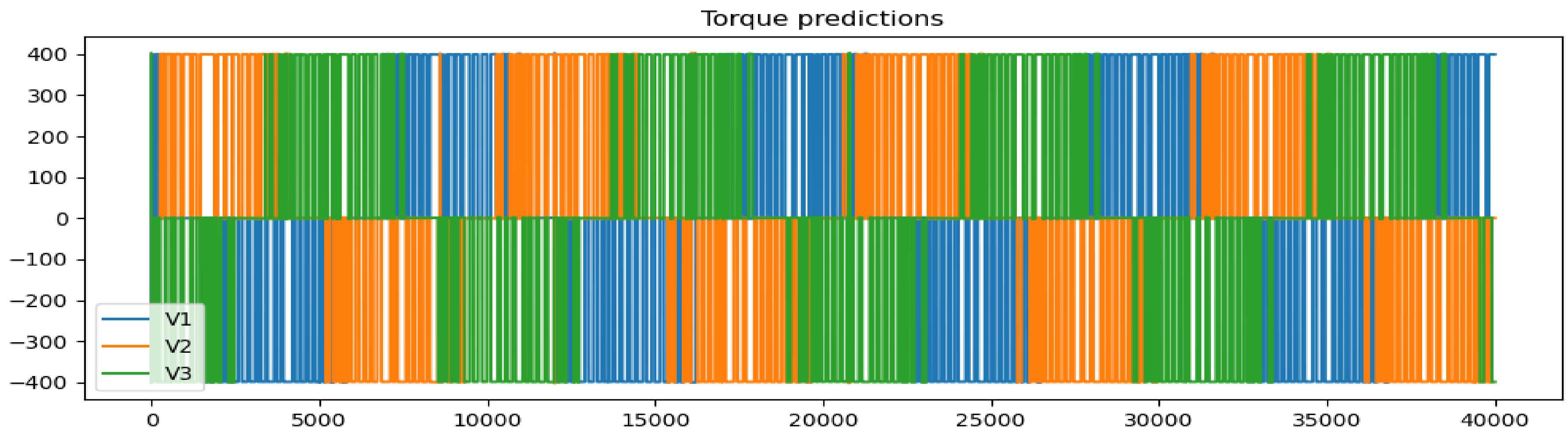


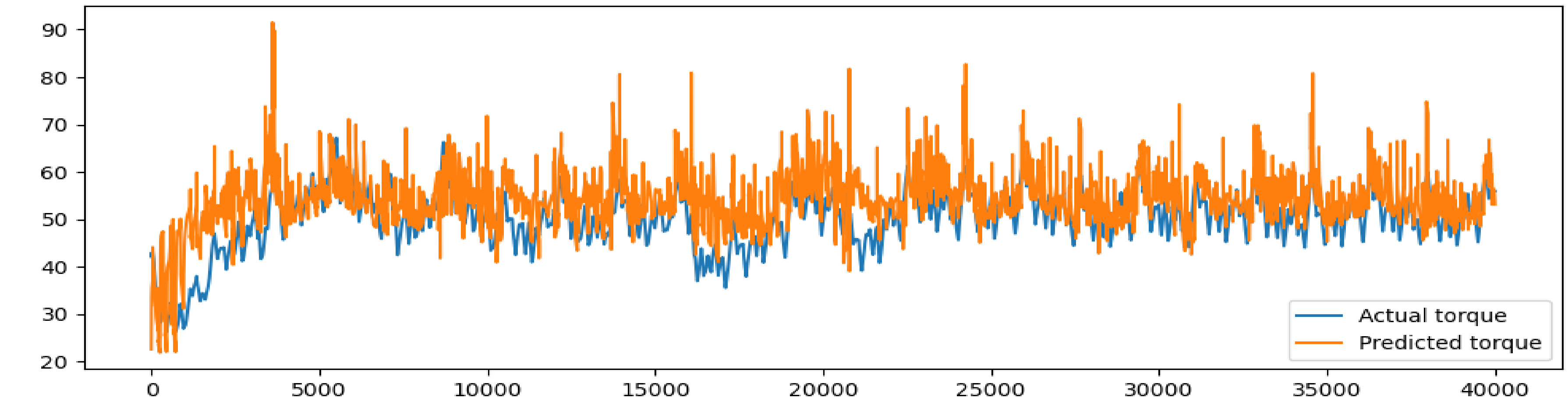


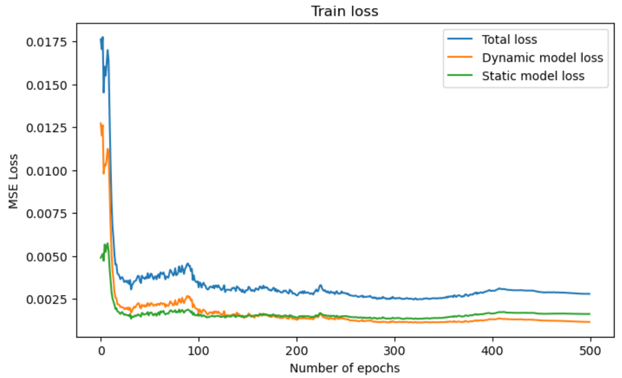
Normalised data - Test error (MAE): 0.0186, Test error (MSE): 0.0006
Denormalised data - Test error (MAE): 7.442, Test error (MSE): 108.110

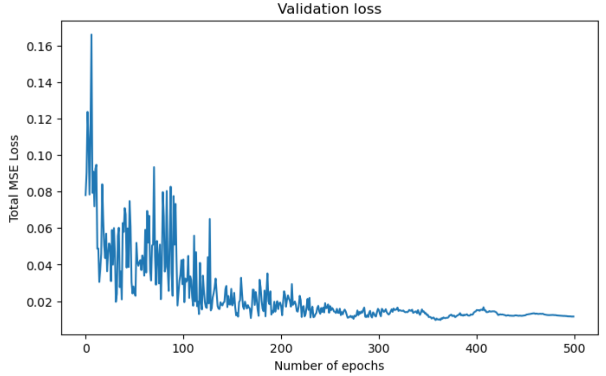
Combined Model (Dynamic + Static)

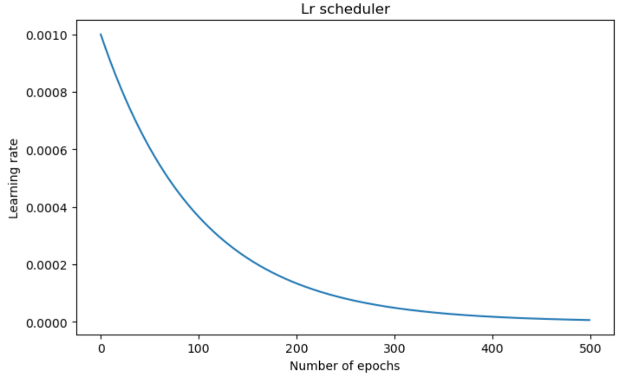
- Parallel training - Static model predictions







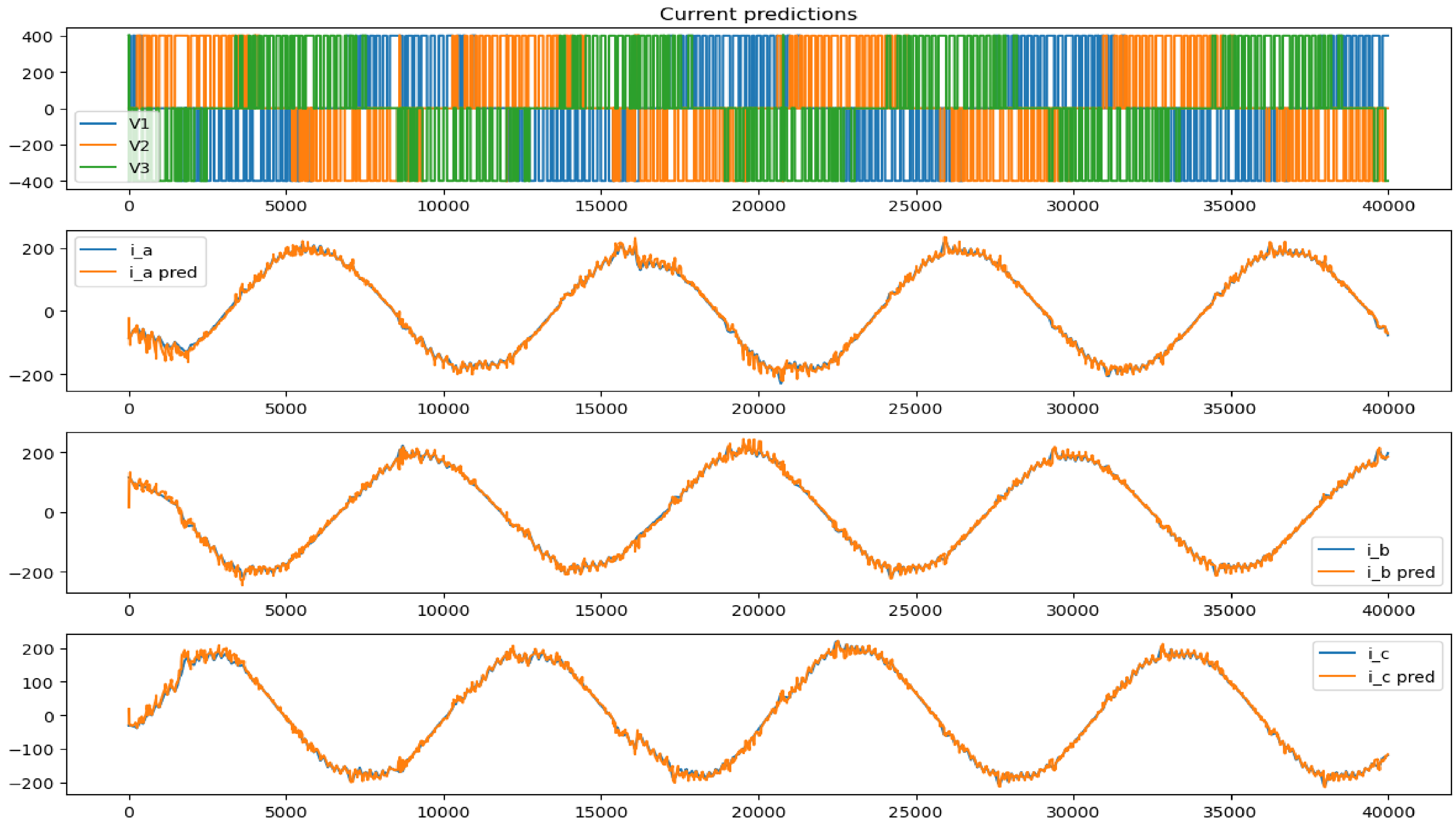


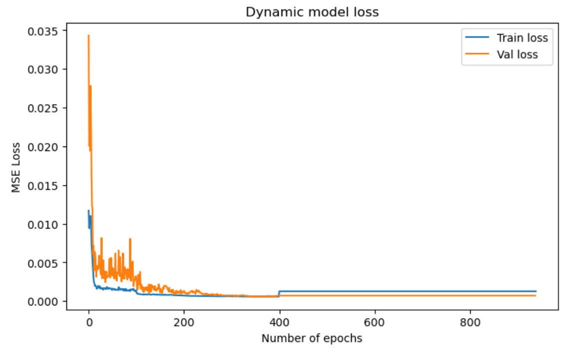


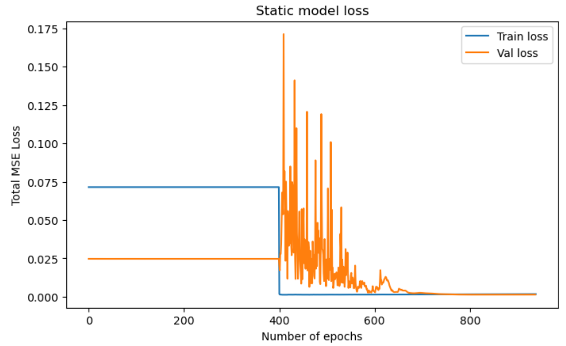
Mean absolute error: 5.037
Mean squared error: 41.648
Maximum error: 34.333

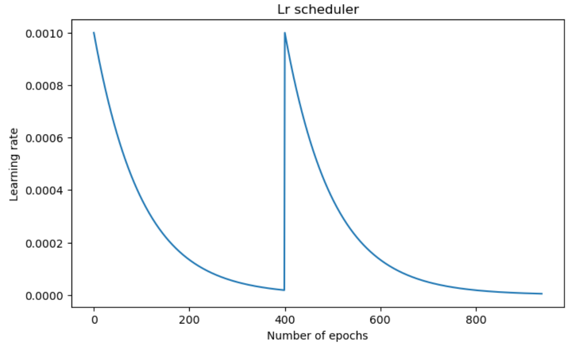
Combined Model (Dynamic + Static)

- Sequential training - Dynamic model predictions





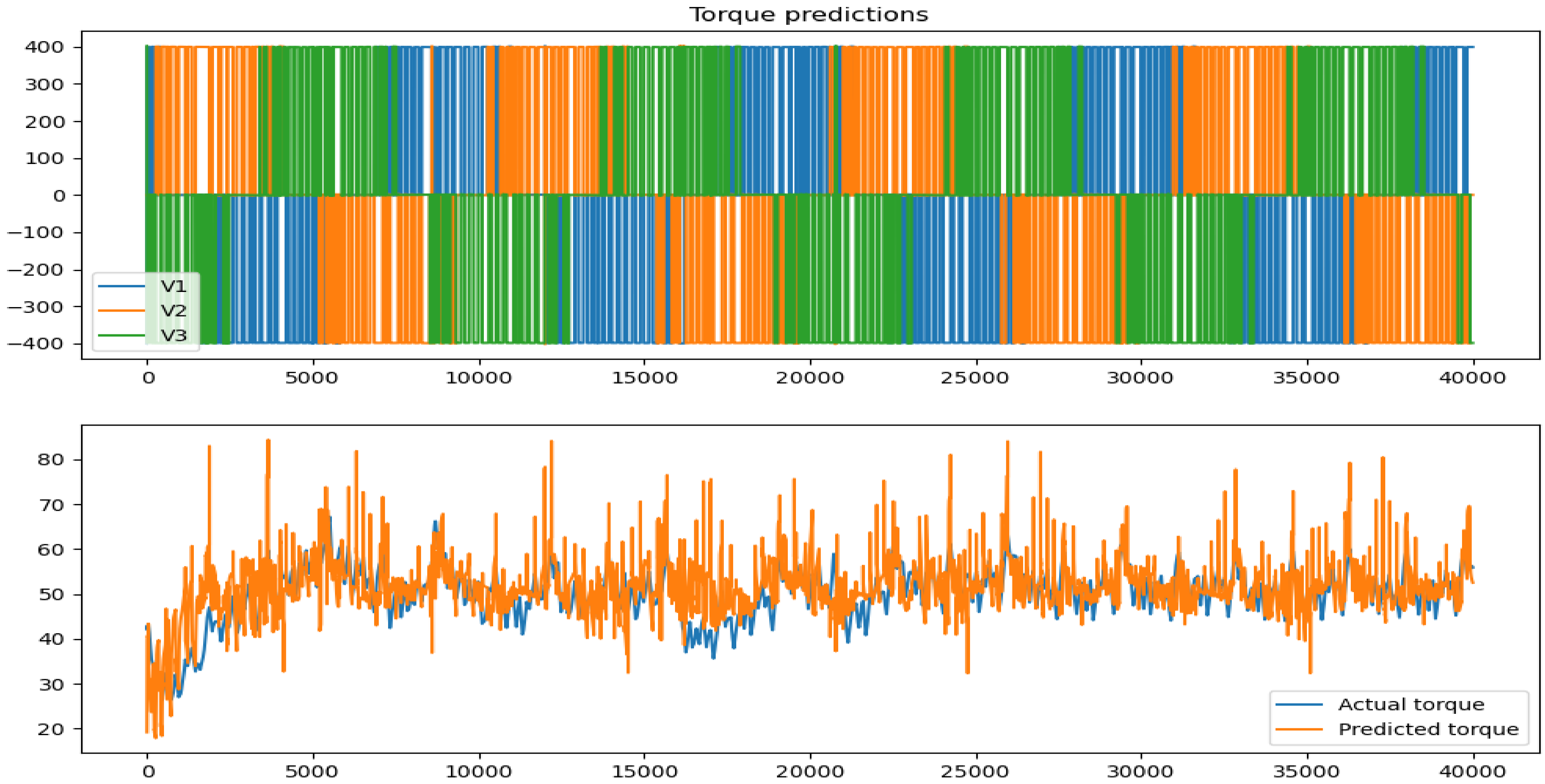




Normalised data - Test error (MAE): 0.01306, Test error (MSE): 0.00038
Denormalised data - Test error (MAE): 5.226, Test error (MSE): 60.98

Combined Model (Dynamic + Static)

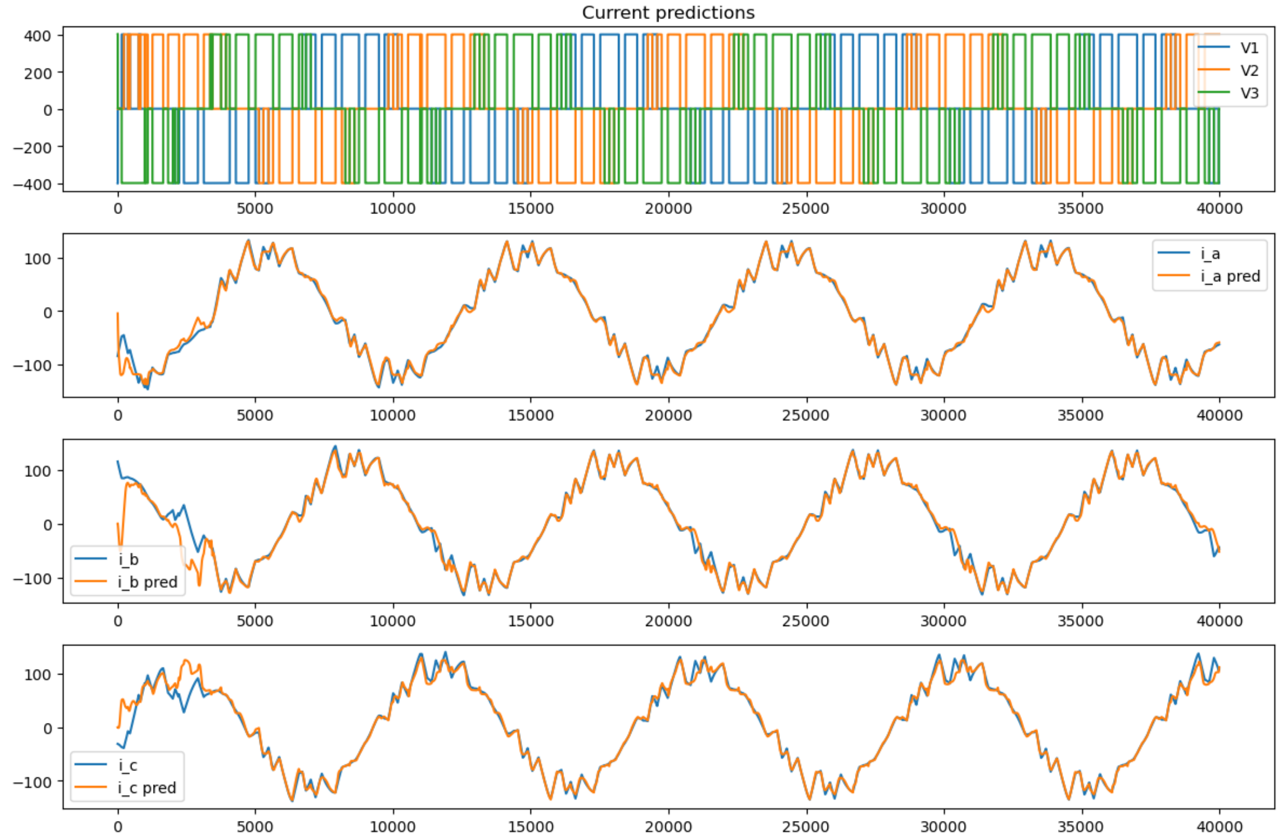
- Sequential training - Static model predictions

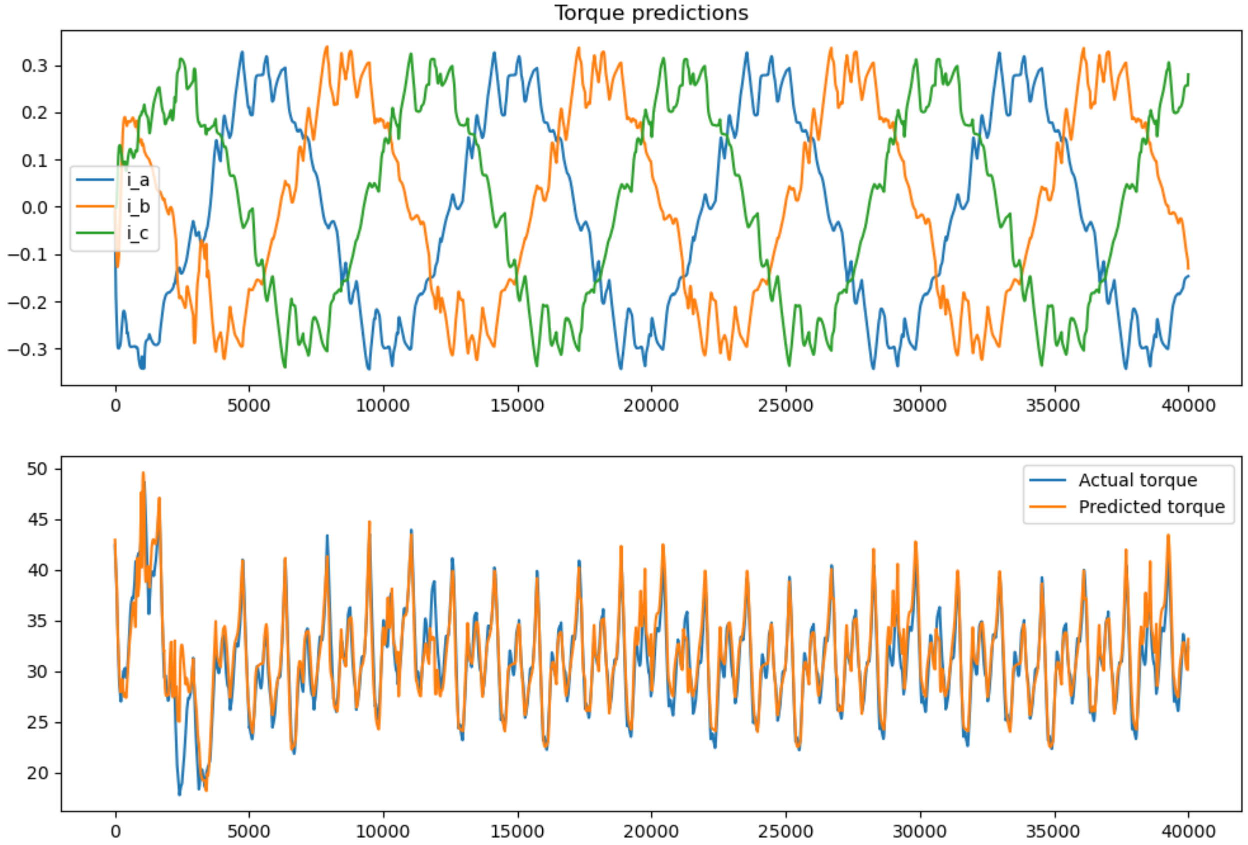


Mean absolute error: 4.164
Mean squared error: 30.52
Maximum error: 36.09

Combined Model (Dynamic + Static)

- Sequential training





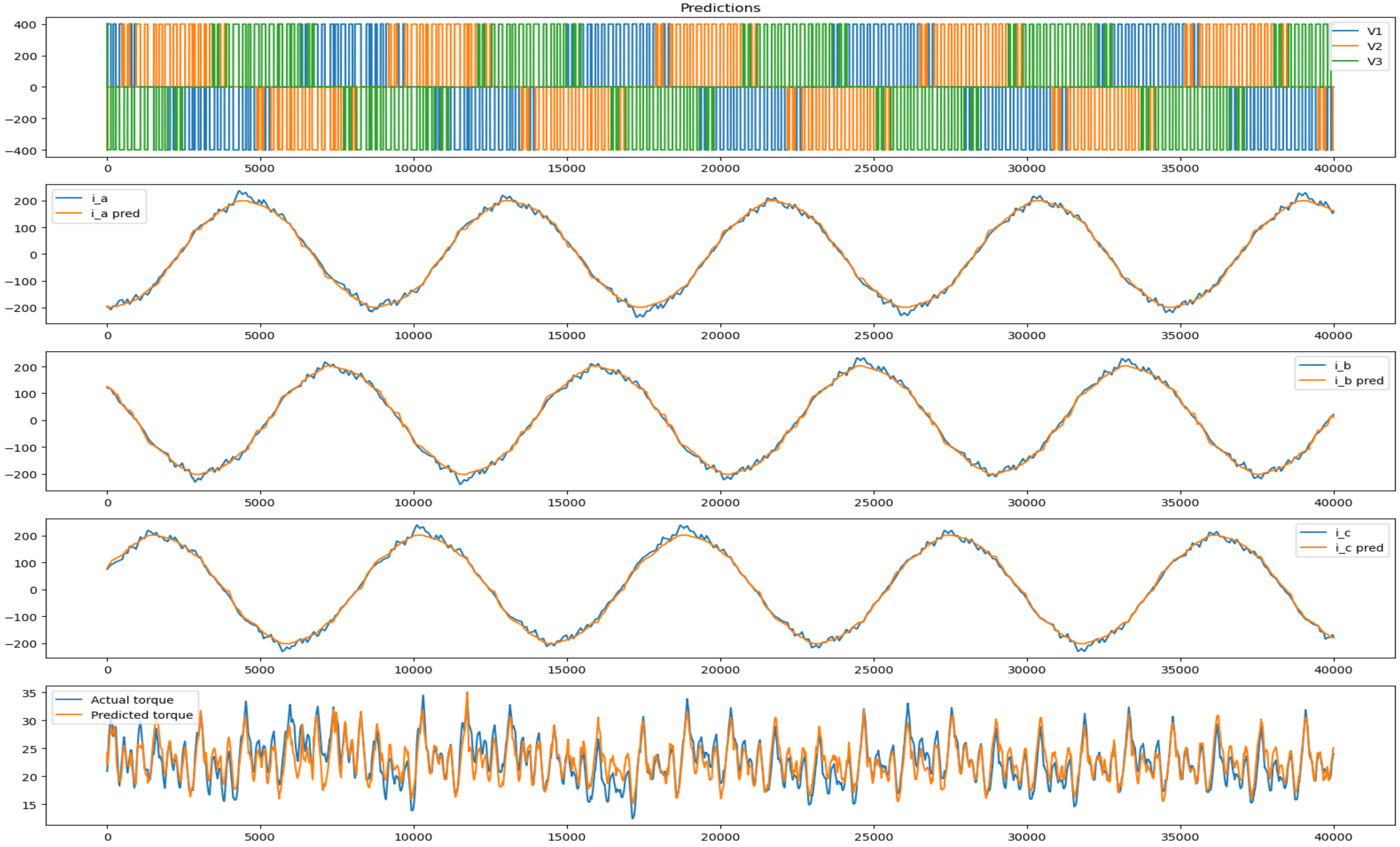
Normalised data - Test error (MAE): 0.2114, Test error (MSE): 0.0814
Denormalised data - Test error (MAE): 4.0392, Test error (MSE): 145.675

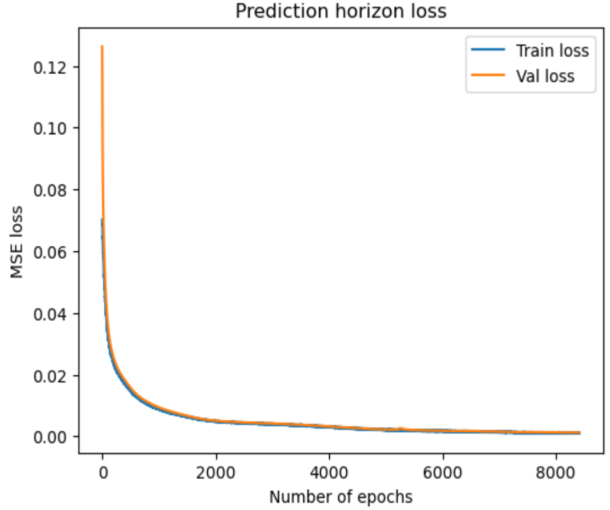
Mean absolute error: 1.201
Mean squared error: 3.145
Maximum error: 13.78

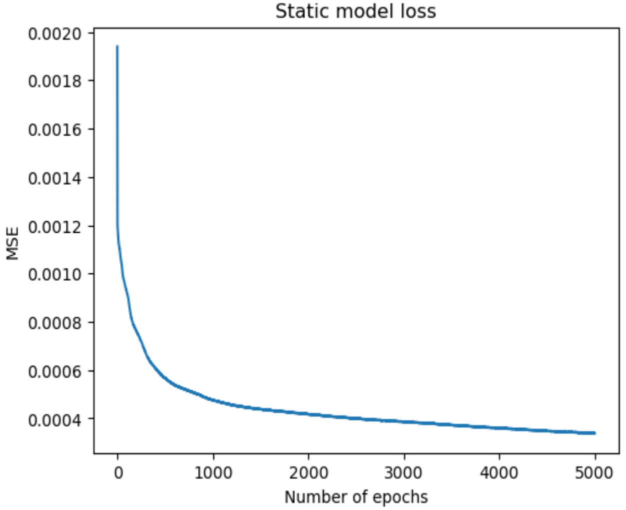
- Torque prediction MAE is meeting benchmark MAE(trained with actual currents).
- Error in current predictions are compensated in static model with sequential training

Combined Model- Sequential training – OPP7(OP1)

Speed : 9250rpm
i_d avg : -200
i_q avg : 30







Normalised data - Test error (MAE): 0.0153, Test error (MSE): 0.0004
Denormalised data - Test error (MAE): 6.12, Test error (MSE): 64.24

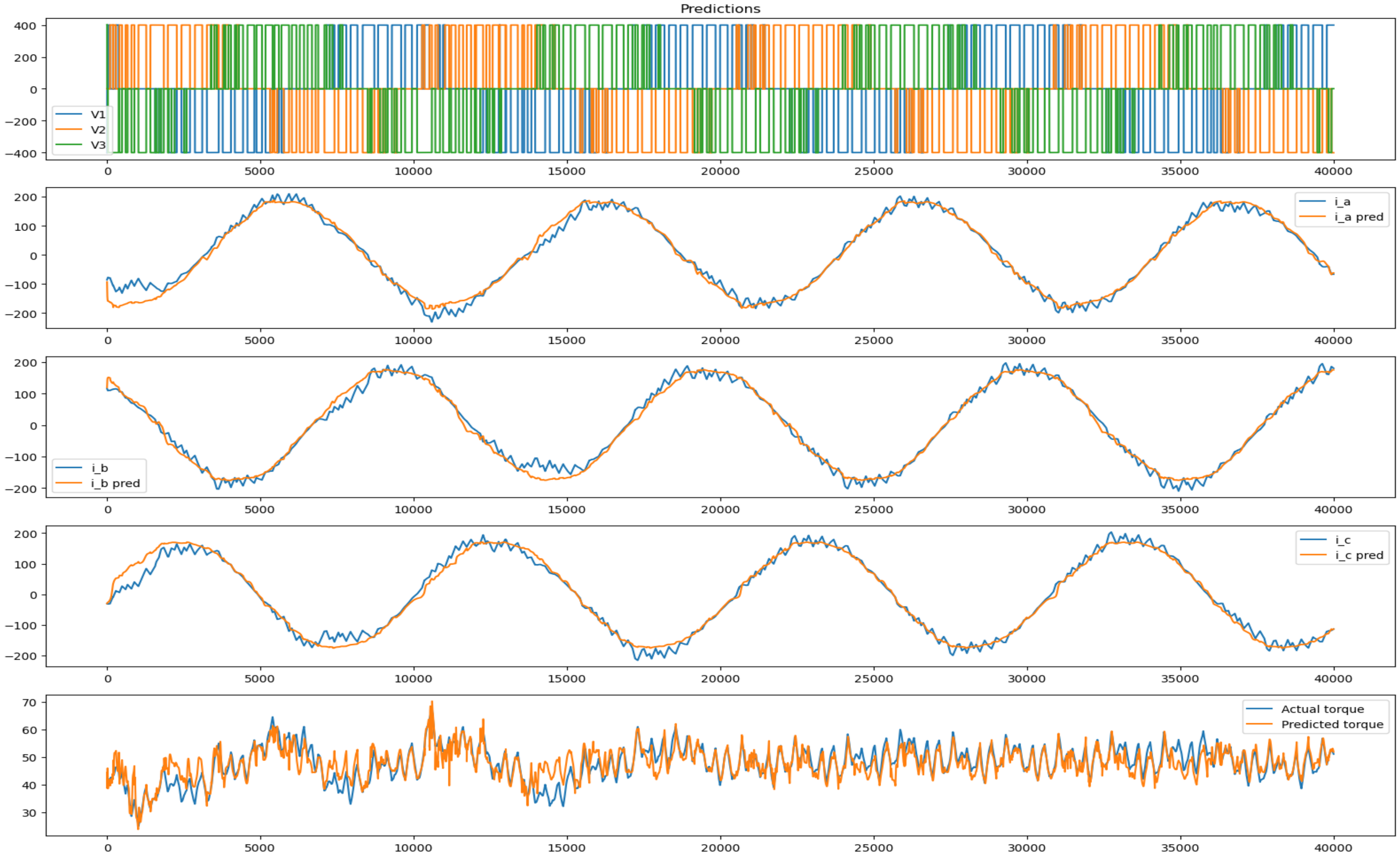
Mean absolute error: 1.569
Mean squared error: 3.644
Maximum error: 6.387

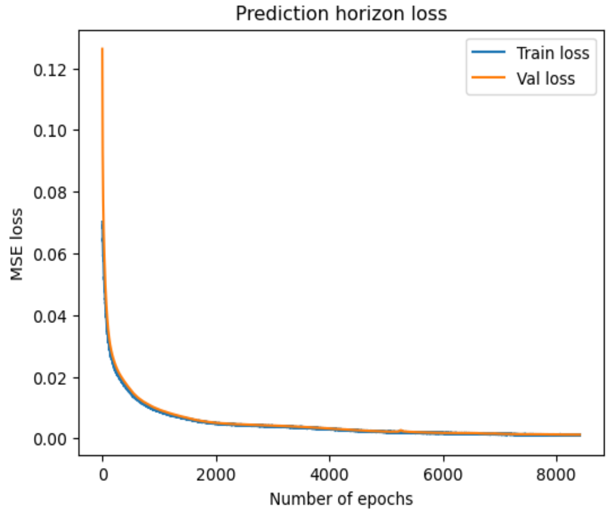
$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

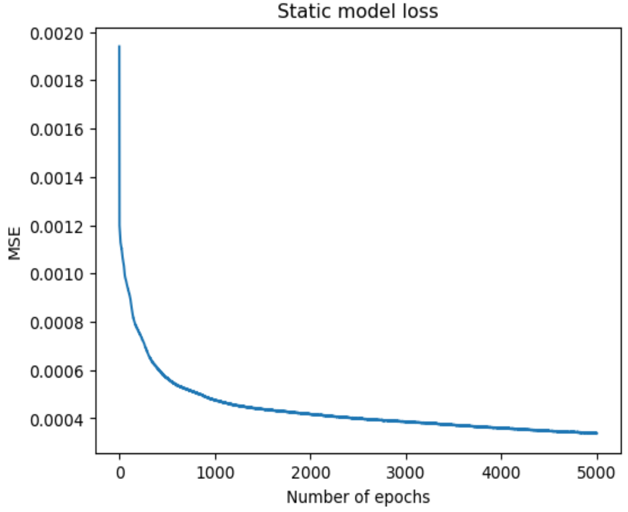
$$W_1 \in R^{64\times 6},\ W_2 \in R^{64\times 64},\ W_3 \in R^{3\times 64}$$

Combined Model- Sequential training – OPP7(OP2)

Speed : 7750rpm
i_d avg : -158
i_q avg : 70







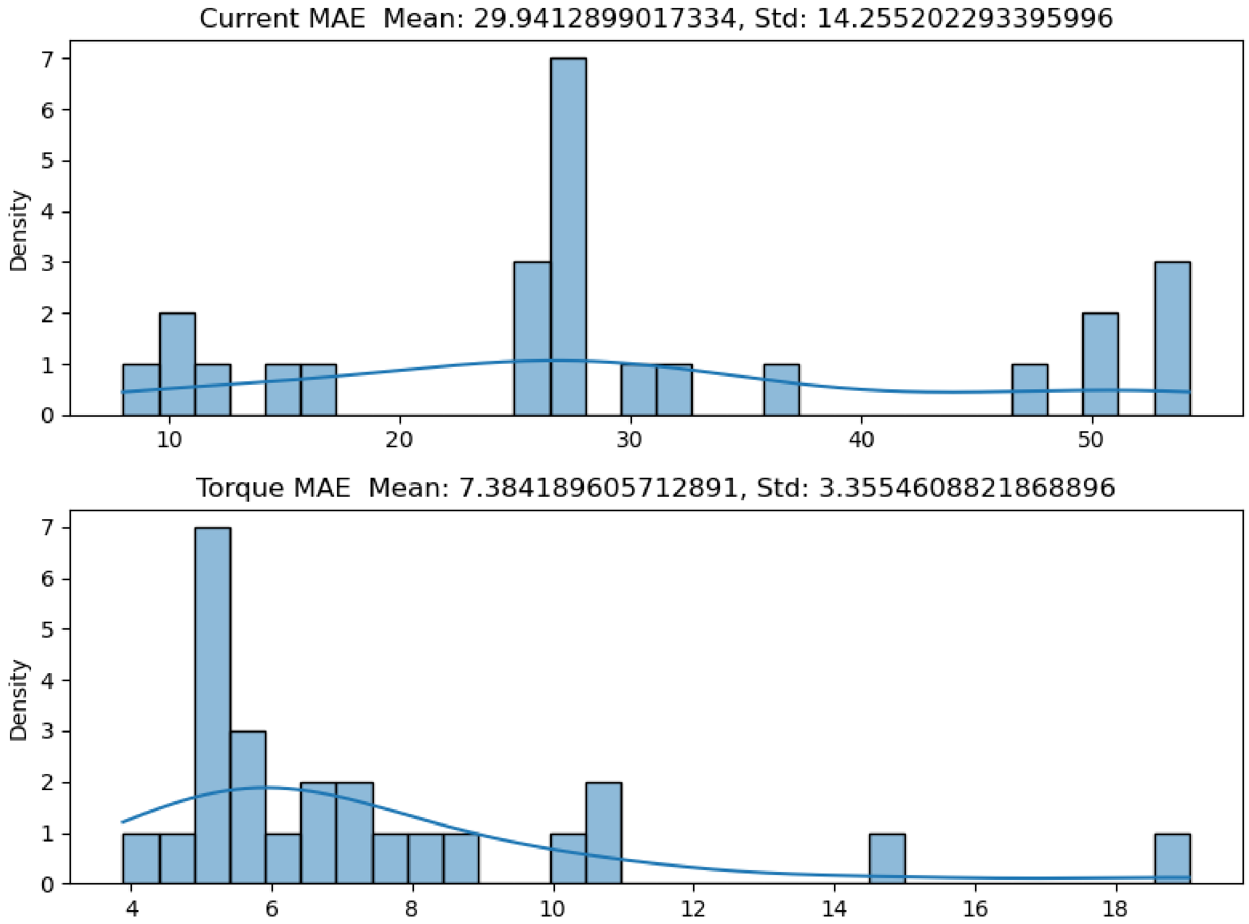
Normalised data - Test error (MAE): 0.031, Test error (MSE): 0.0019
Denormalised data - Test error (MAE): 11.41, Test error (MSE): 304.06

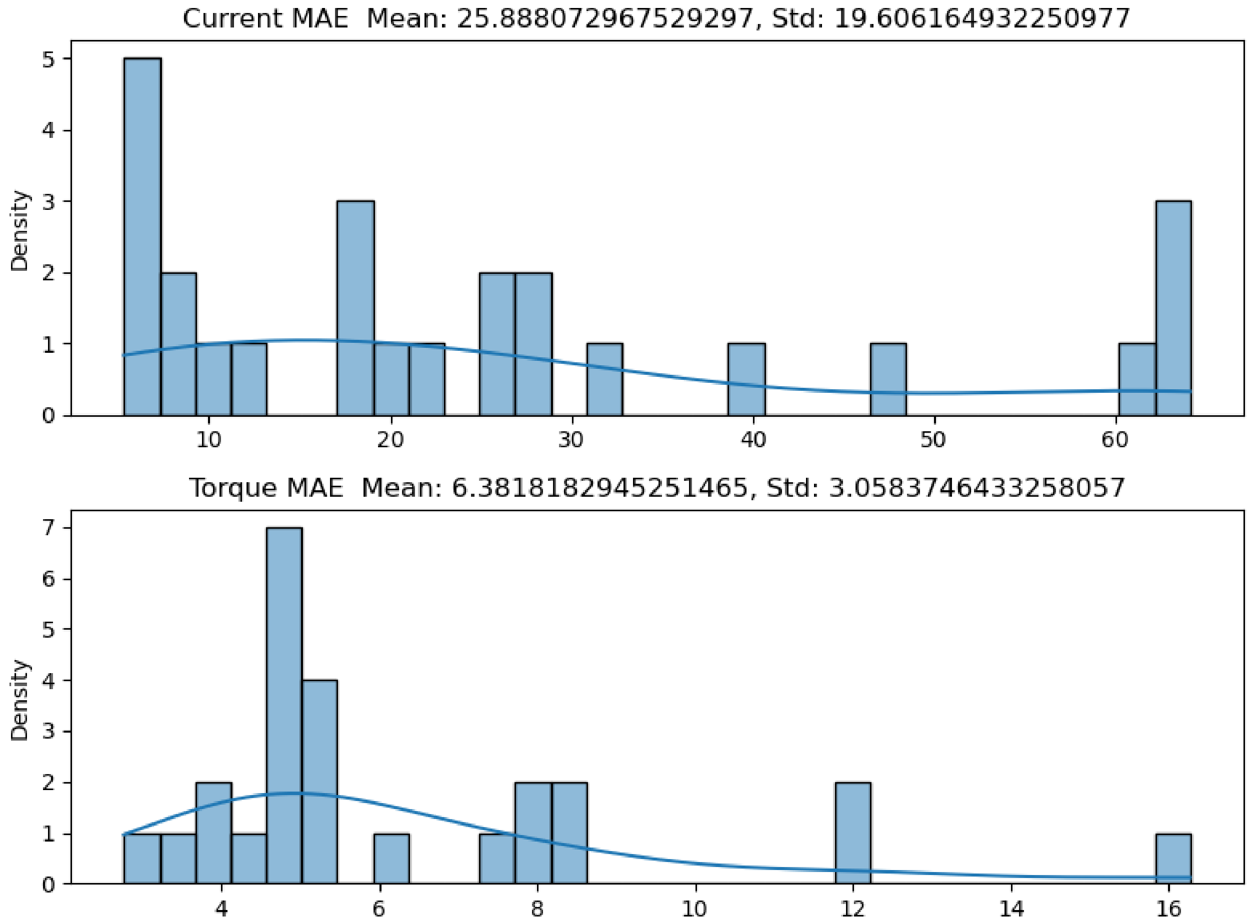
Mean absolute error: 2.428
Mean squared error: 11.234
Maximum error: 17.34

$$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$$

$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

Combined Model– DOE 1(7750rpm)





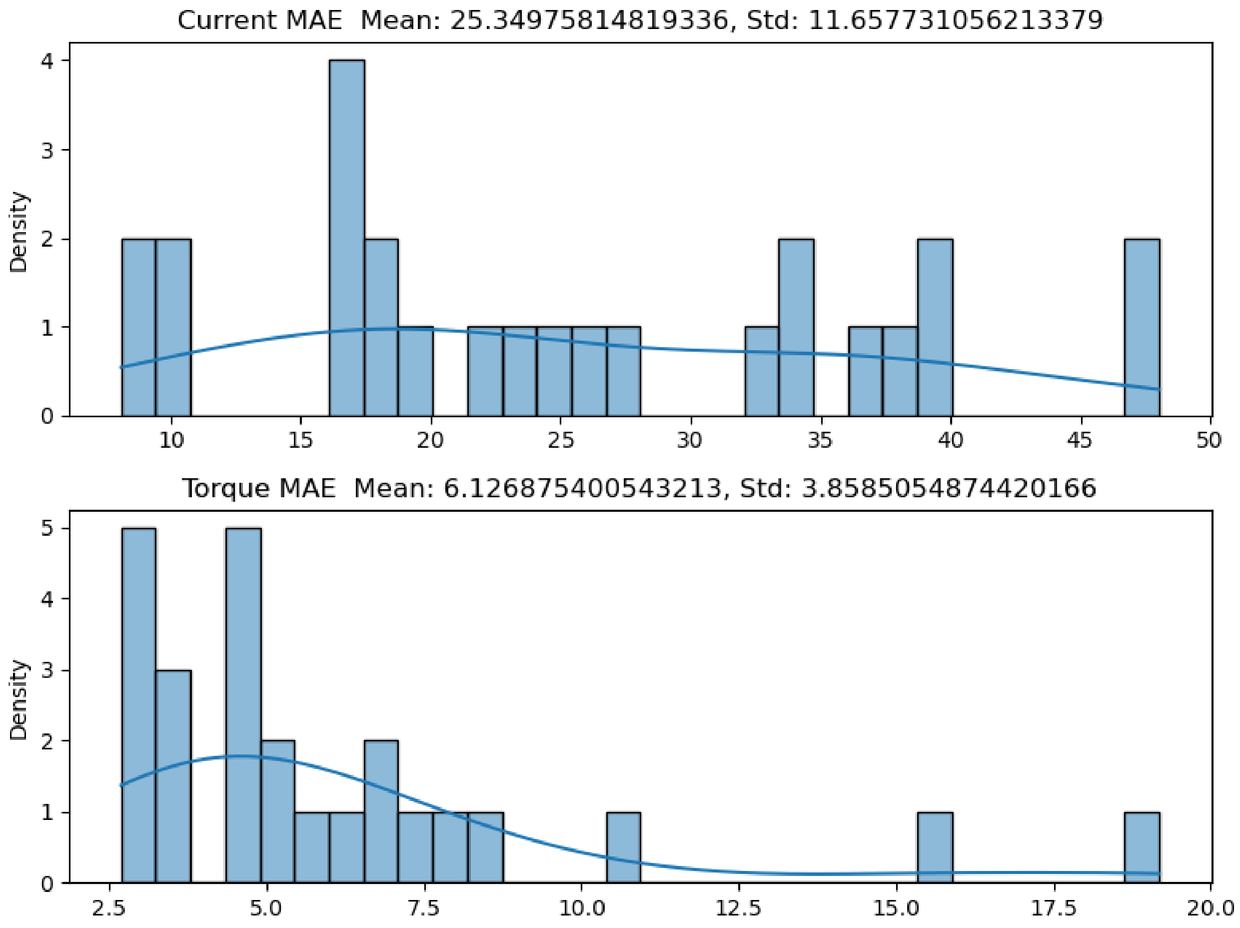
$y= \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$

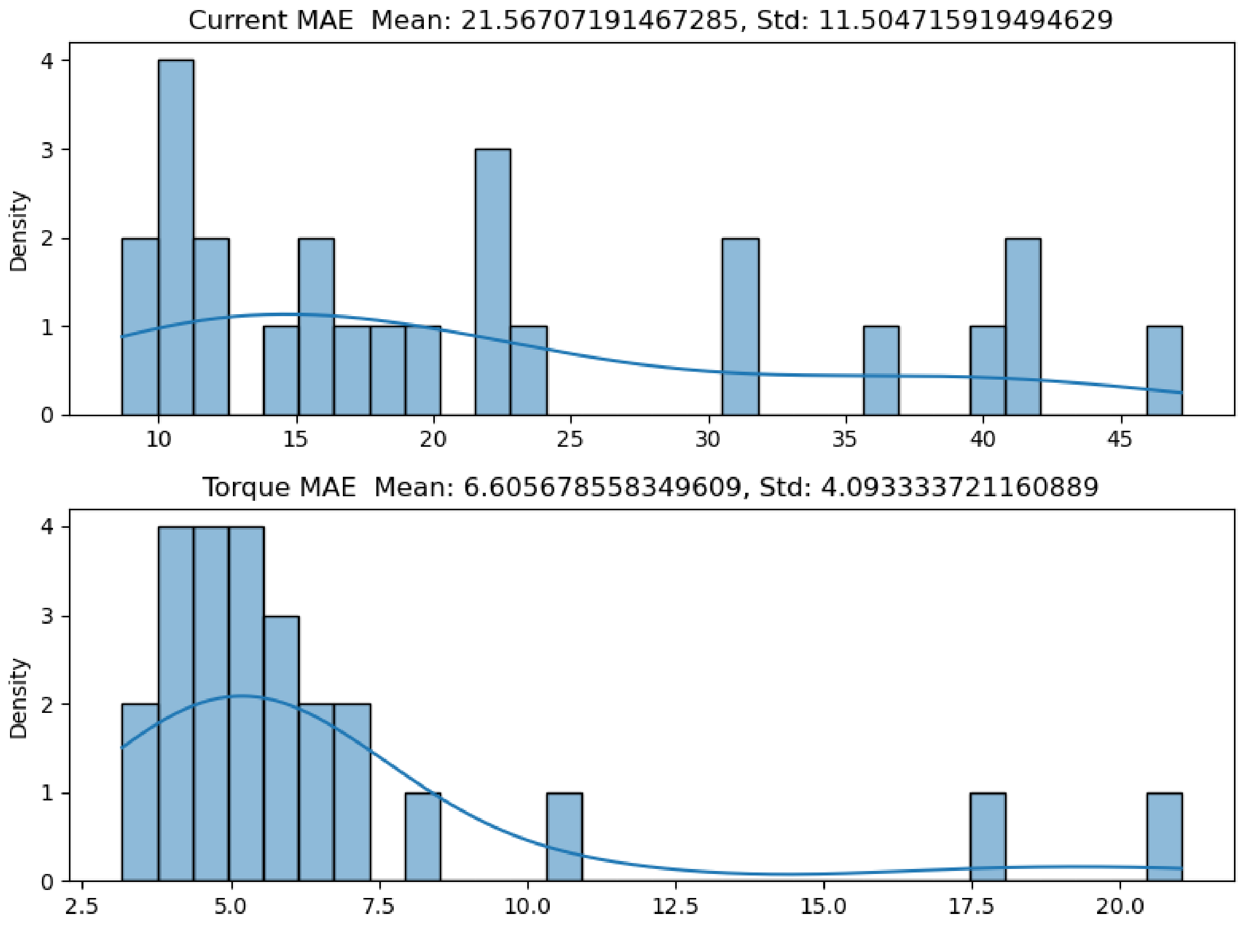
$W_1 \in R^{48\times6}$, $W_2 \in R^{48\times48}$, $W_3 \in R^{3\times48}$

$y= \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$

$W_1 \in R^{64\times6}$, $W_2 \in R^{64\times64}$, $W_3 \in R^{3\times64}$

Combined Model– DOE 1(7750rpm)





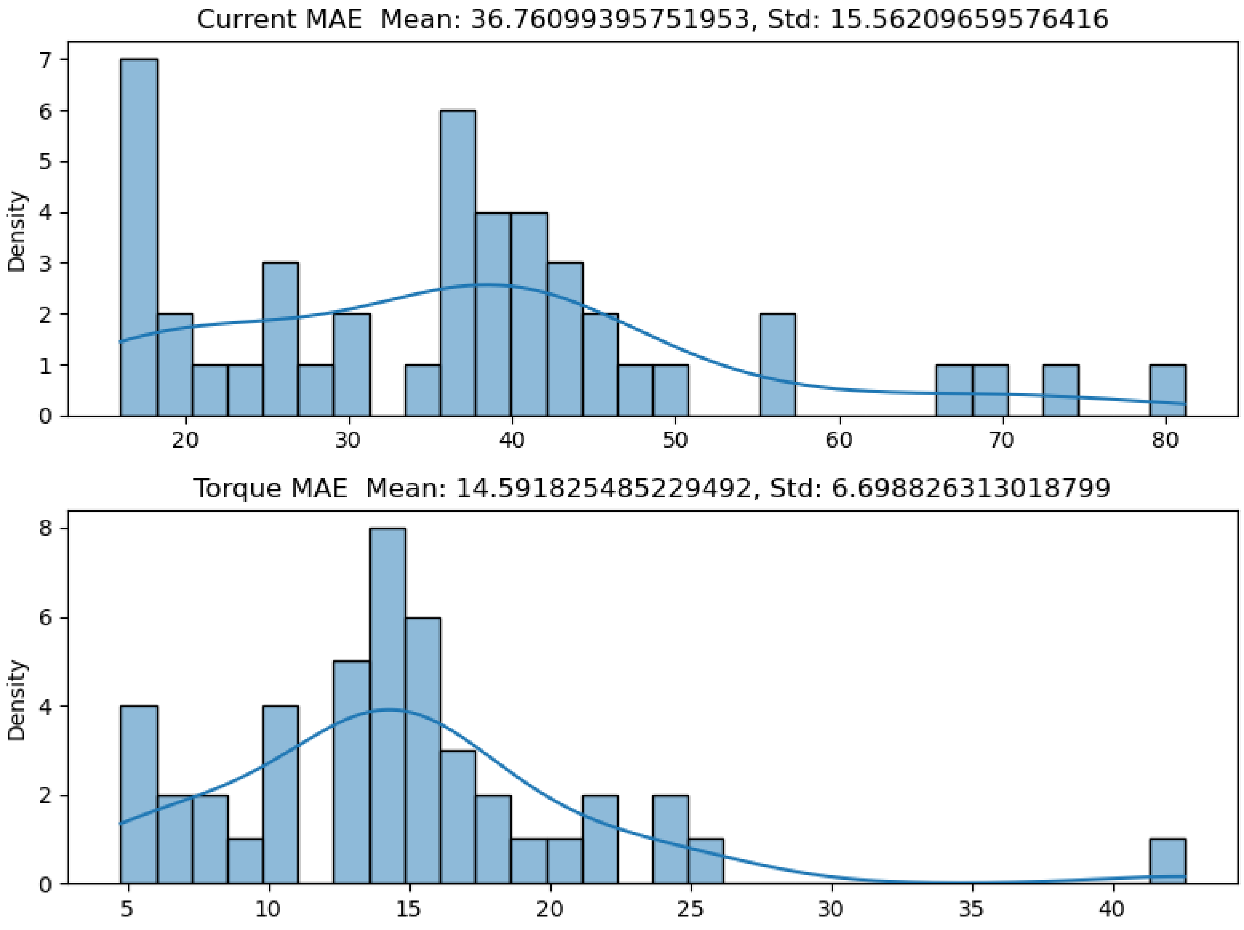
$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$

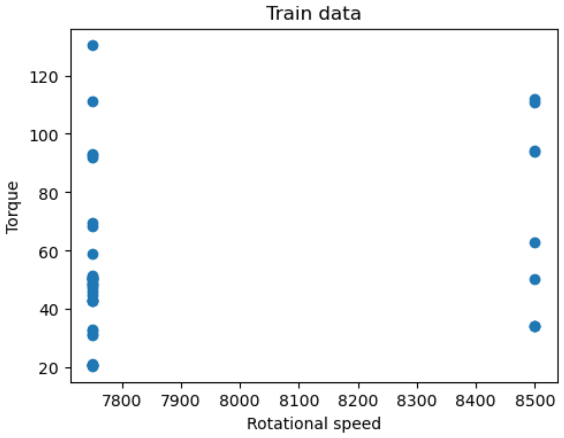
$W_1 \in R^{80\times 6}$, $W_2 \in R^{80\times 48}$, $W_3 \in R^{3\times 80}$

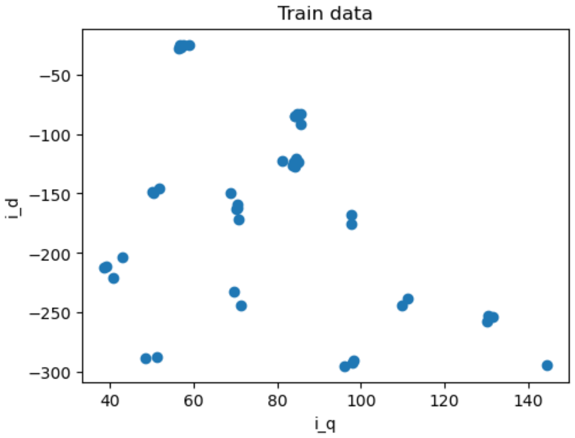
$y = \int W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 . x))$

$W_1 \in R^{64\times 14}$, $W_2 \in R^{64\times 64}$, $W_3 \in R^{3\times 64}$

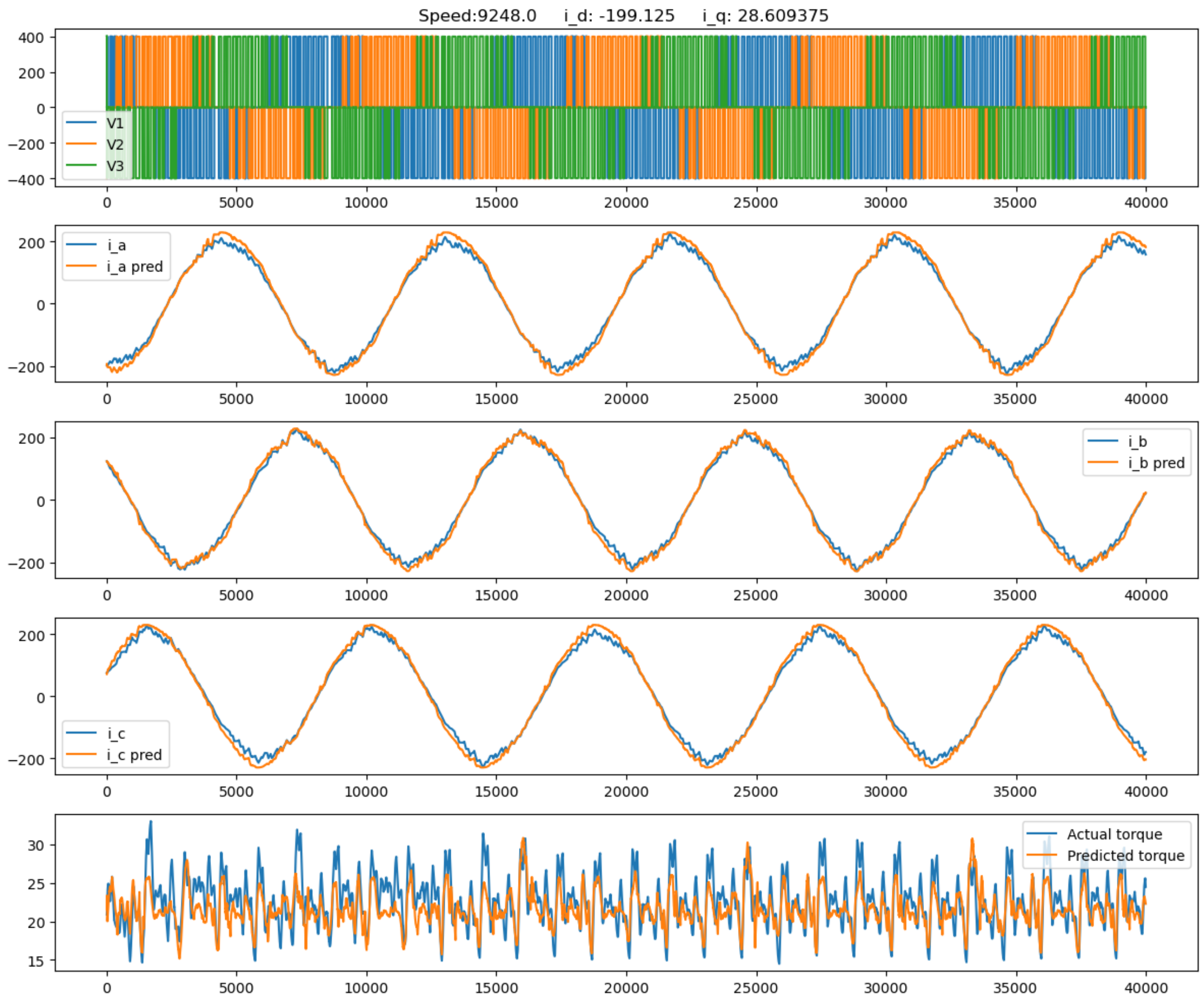
Combined Model– DOE 1(7750rpm)

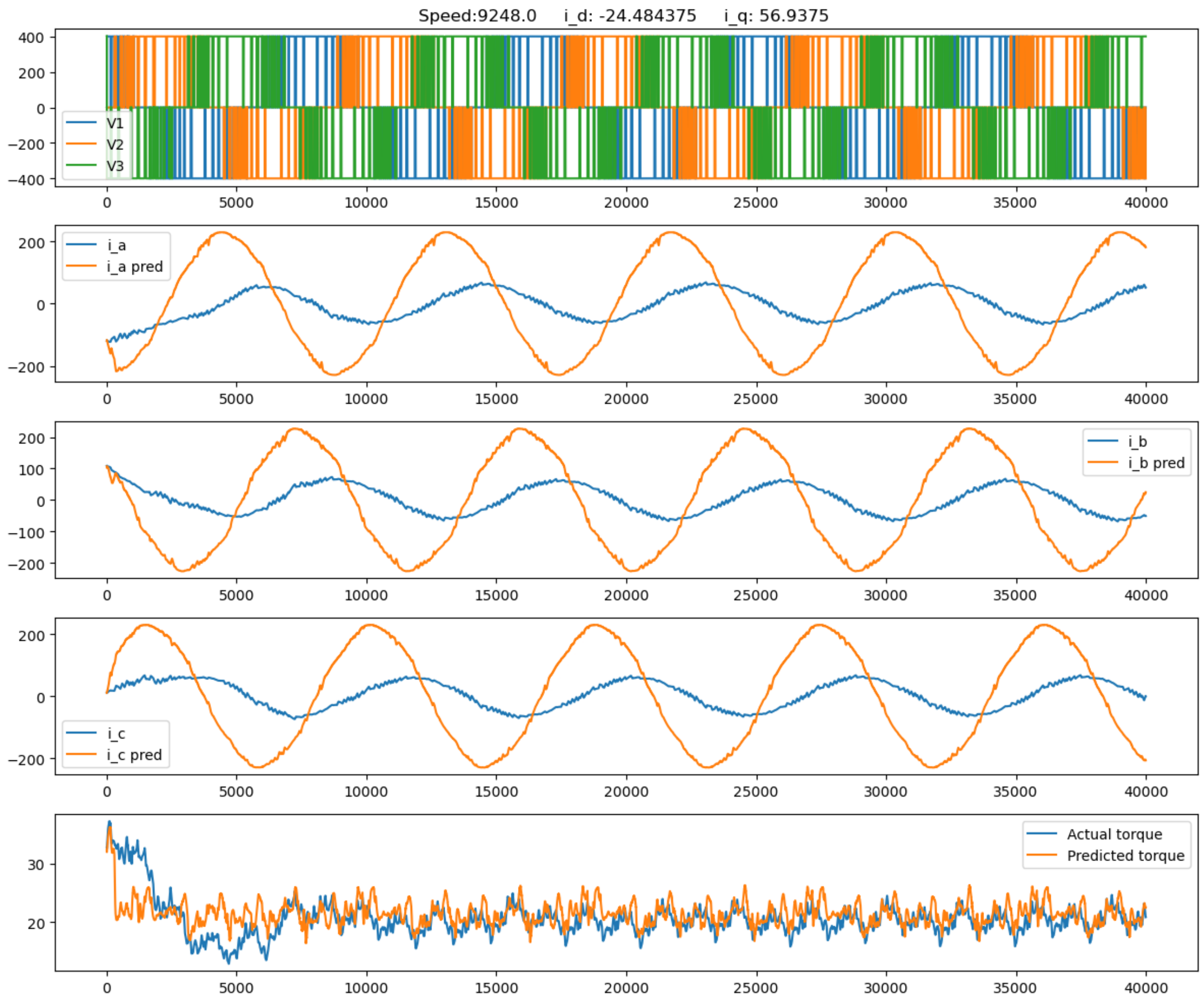




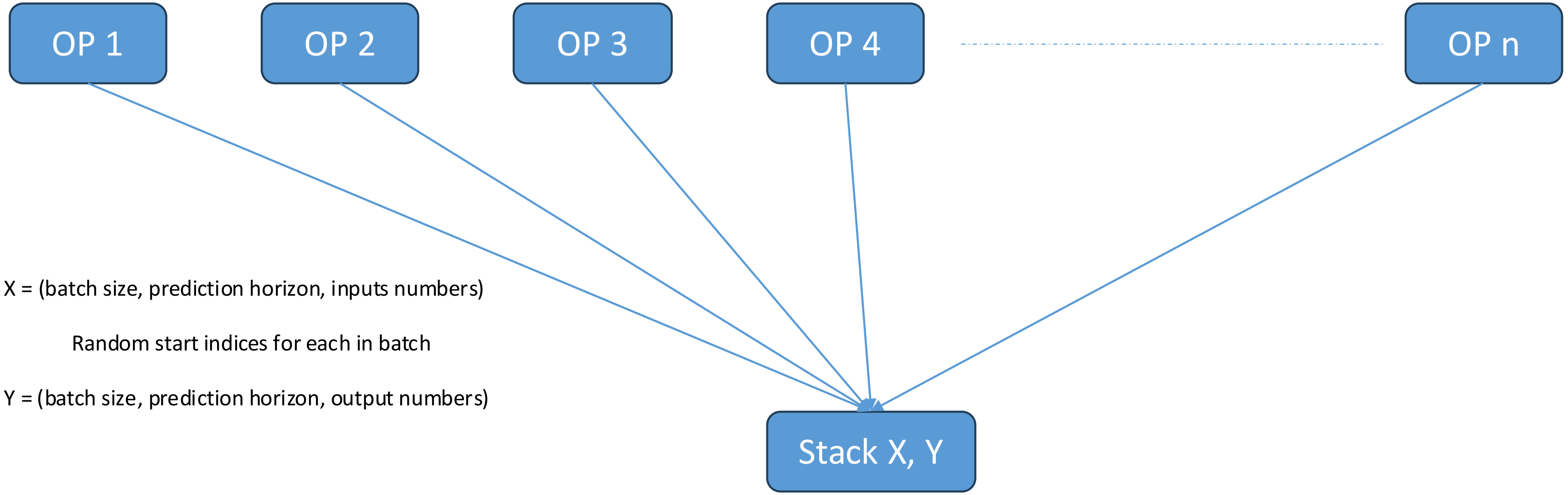


Combined Model– DOE3(9250rpm)





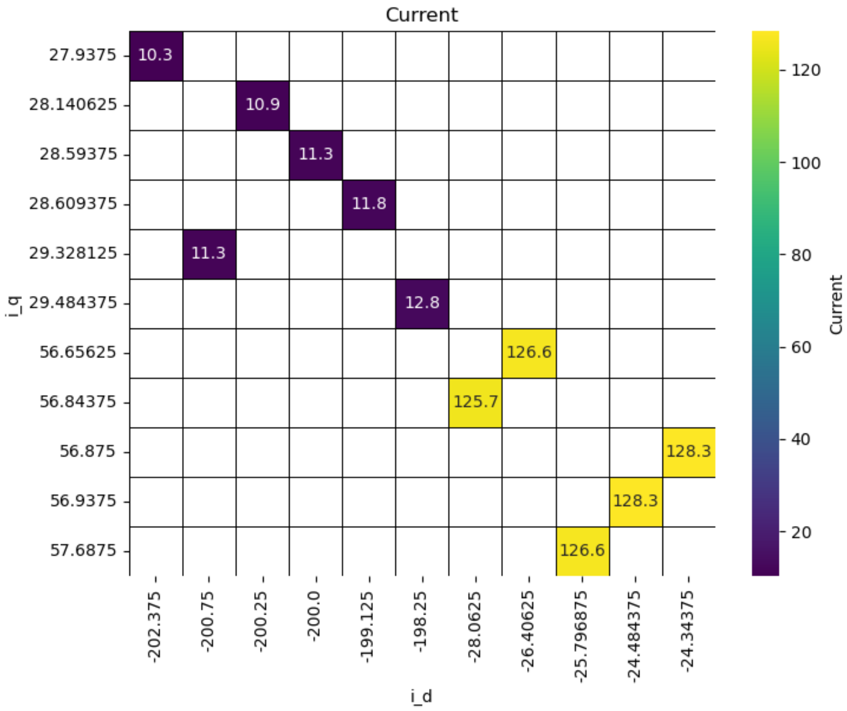
Data loader for homogeneity

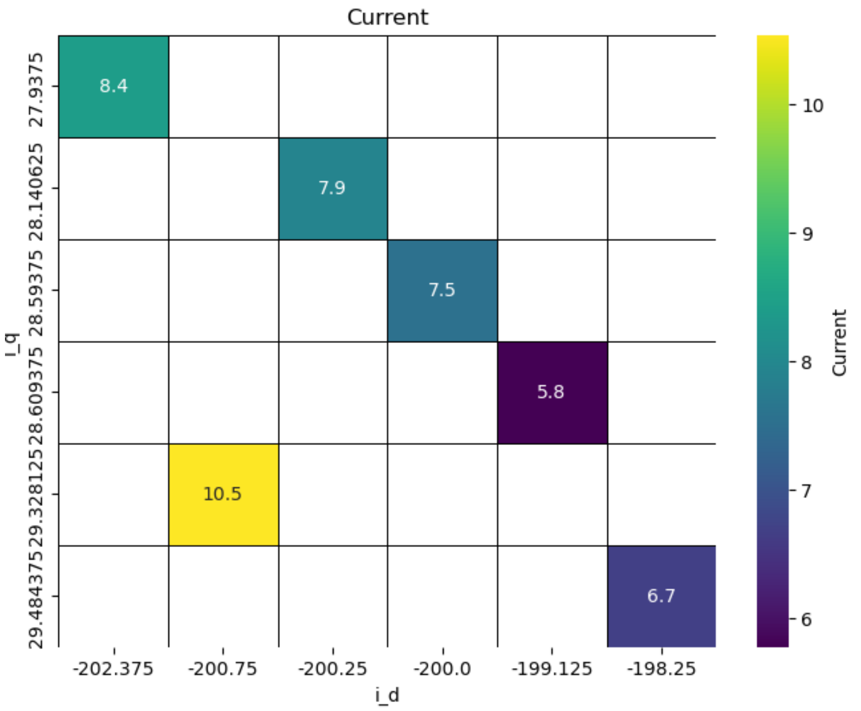


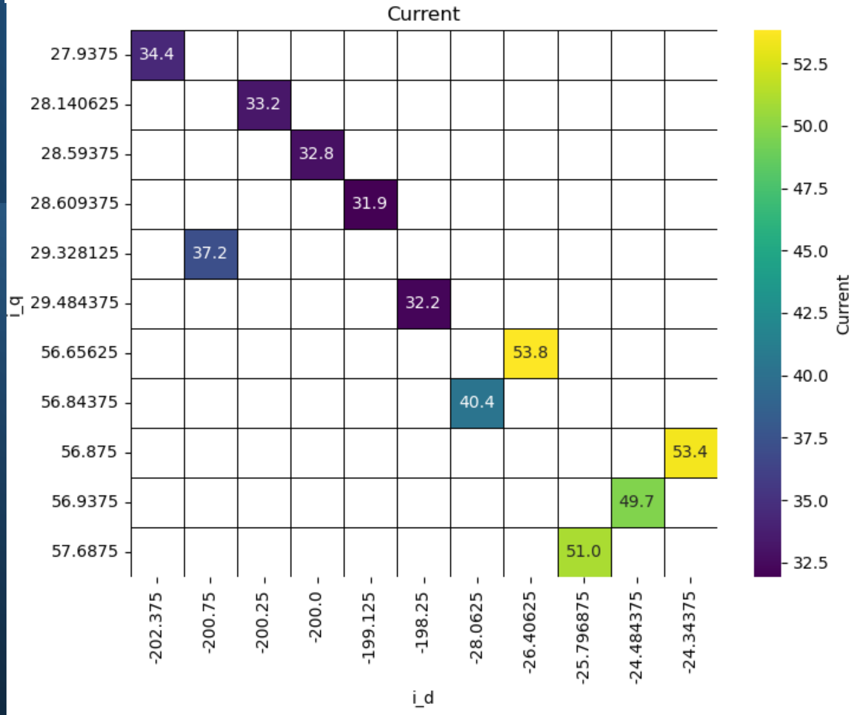
X = (n*batch size, prediction horizon, inputs numbers)

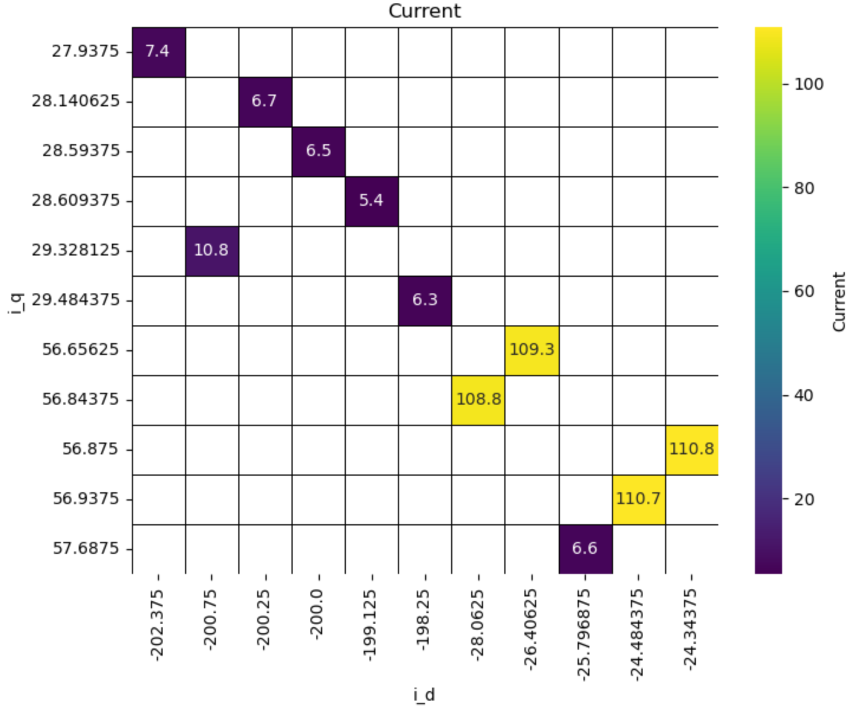
Y = (n*batch size, prediction horizon, output numbers)

Combined Model–DOE3(9250rpm)

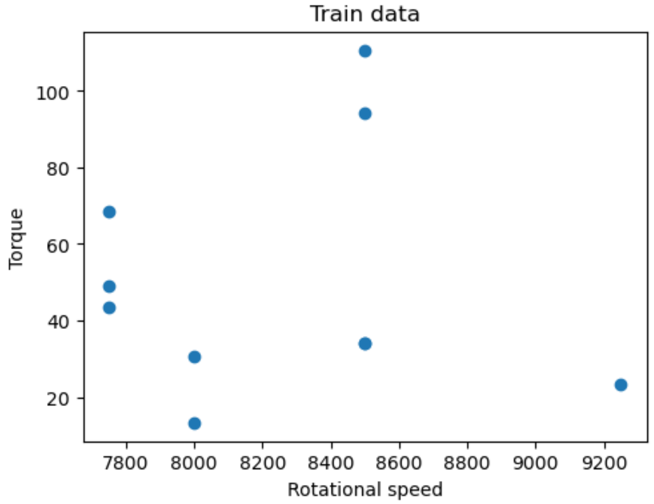


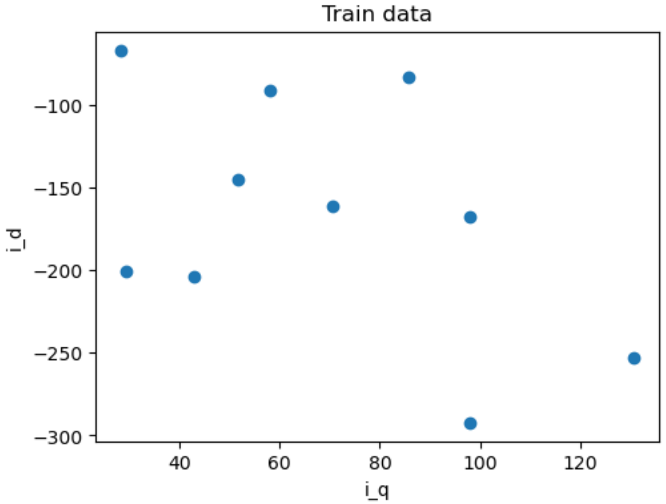




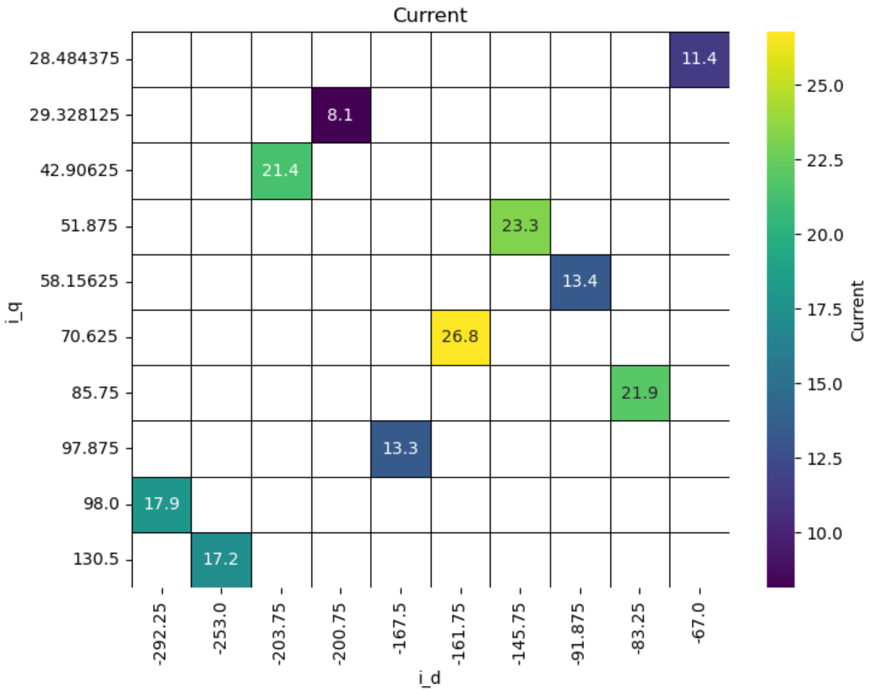


DOE 5 – Diff prediction horizon steps

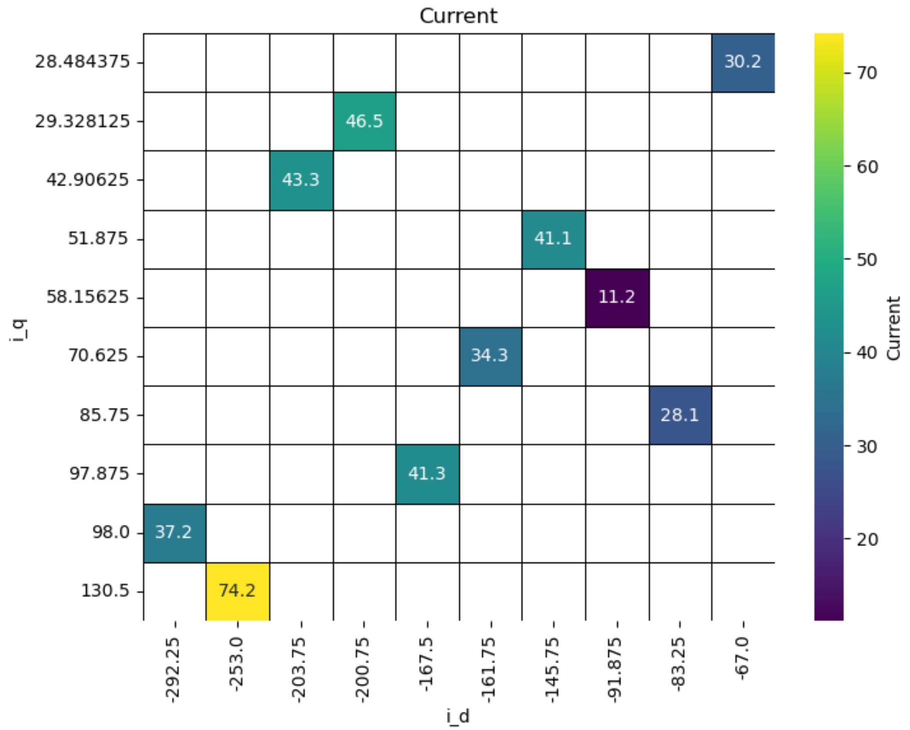




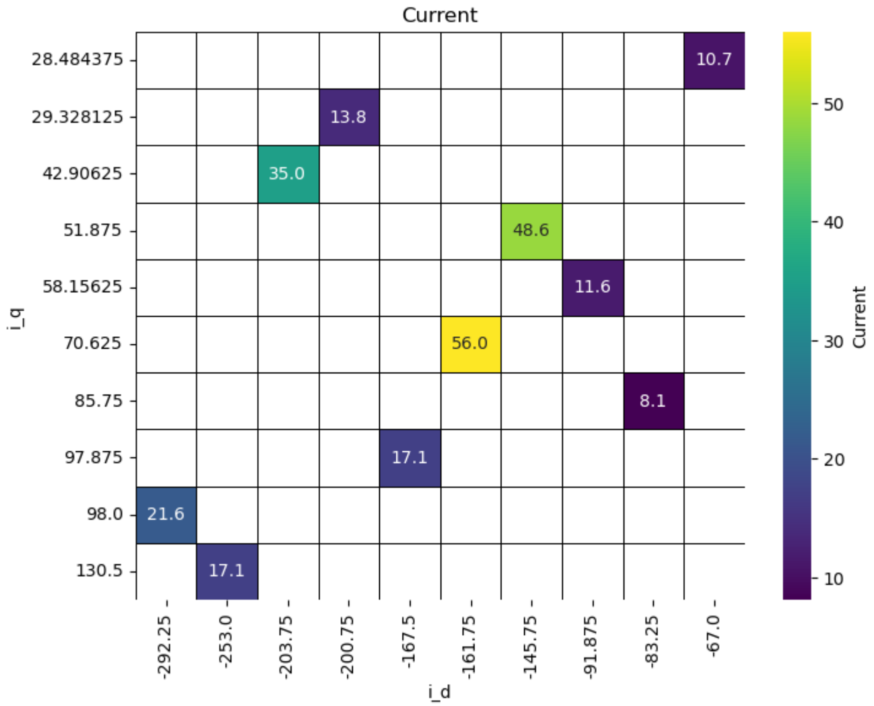
5000

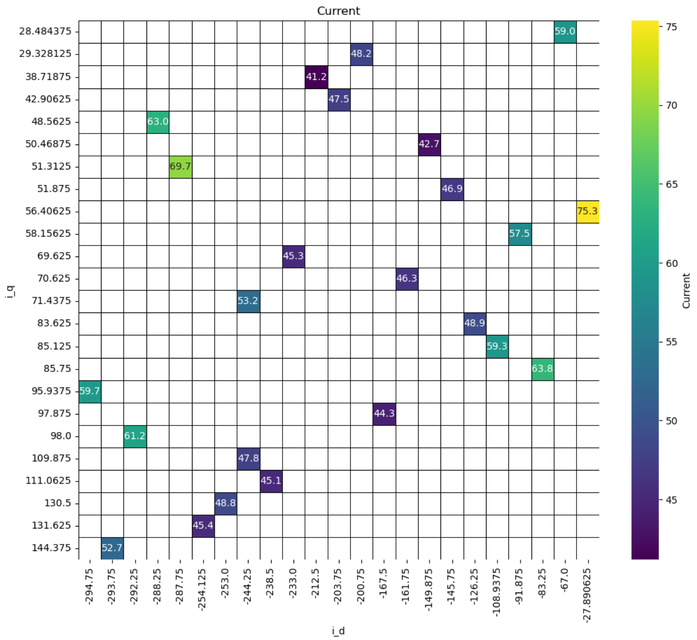


500

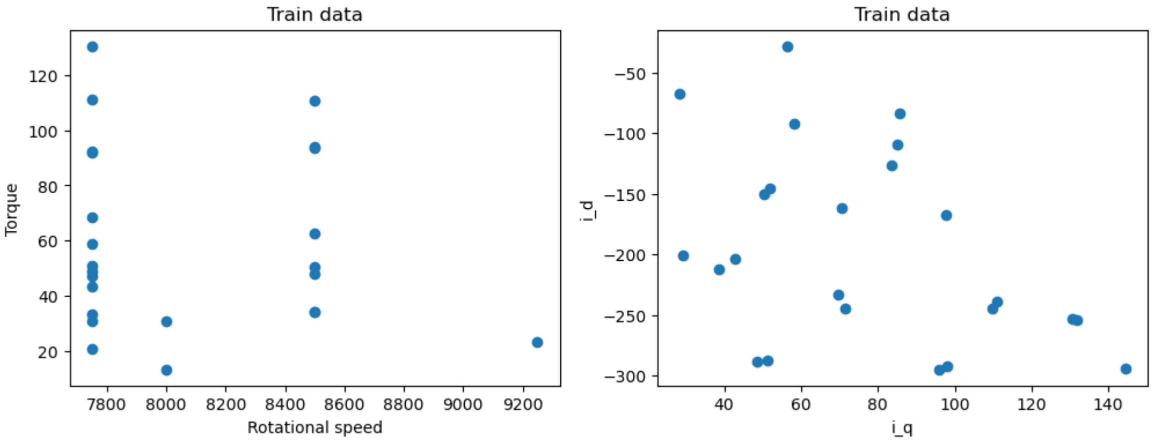


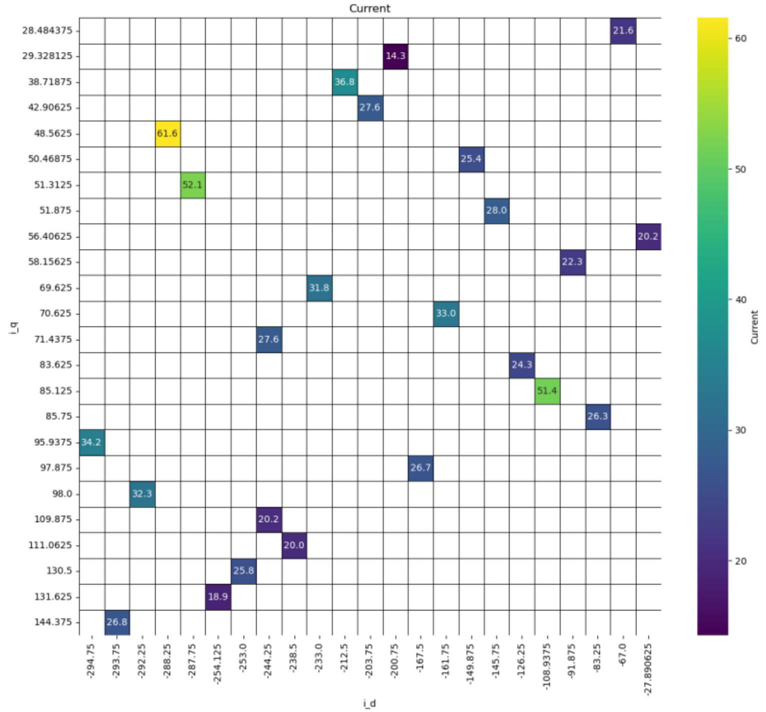
2000



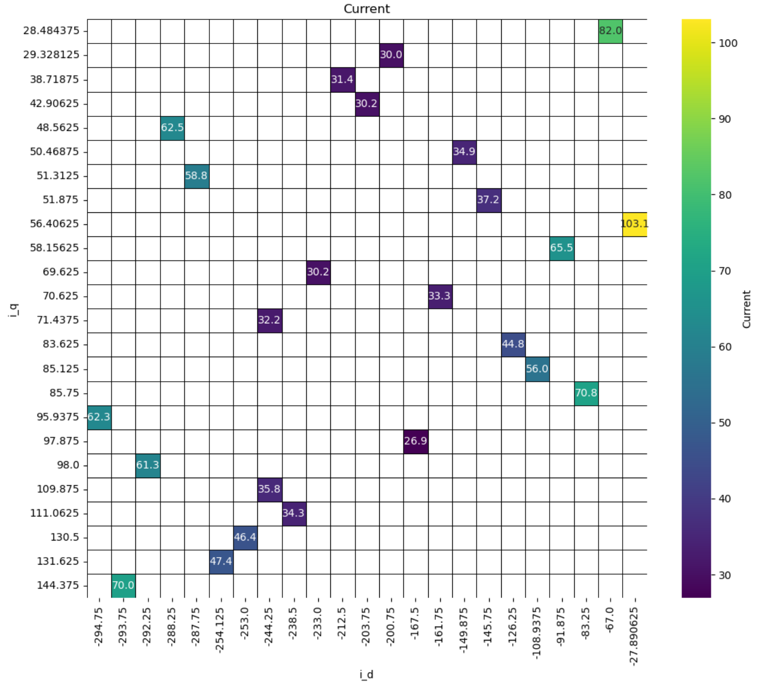


[128,128]

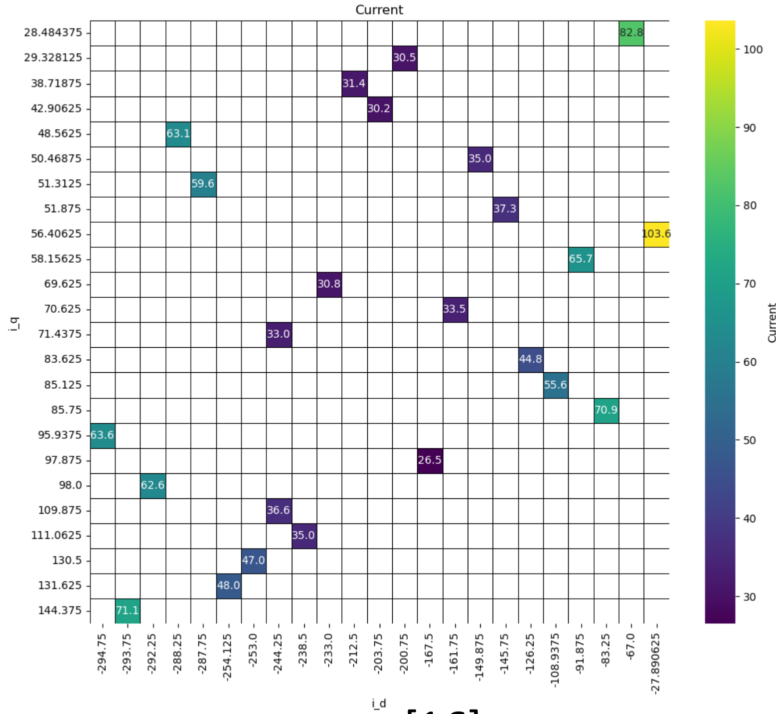




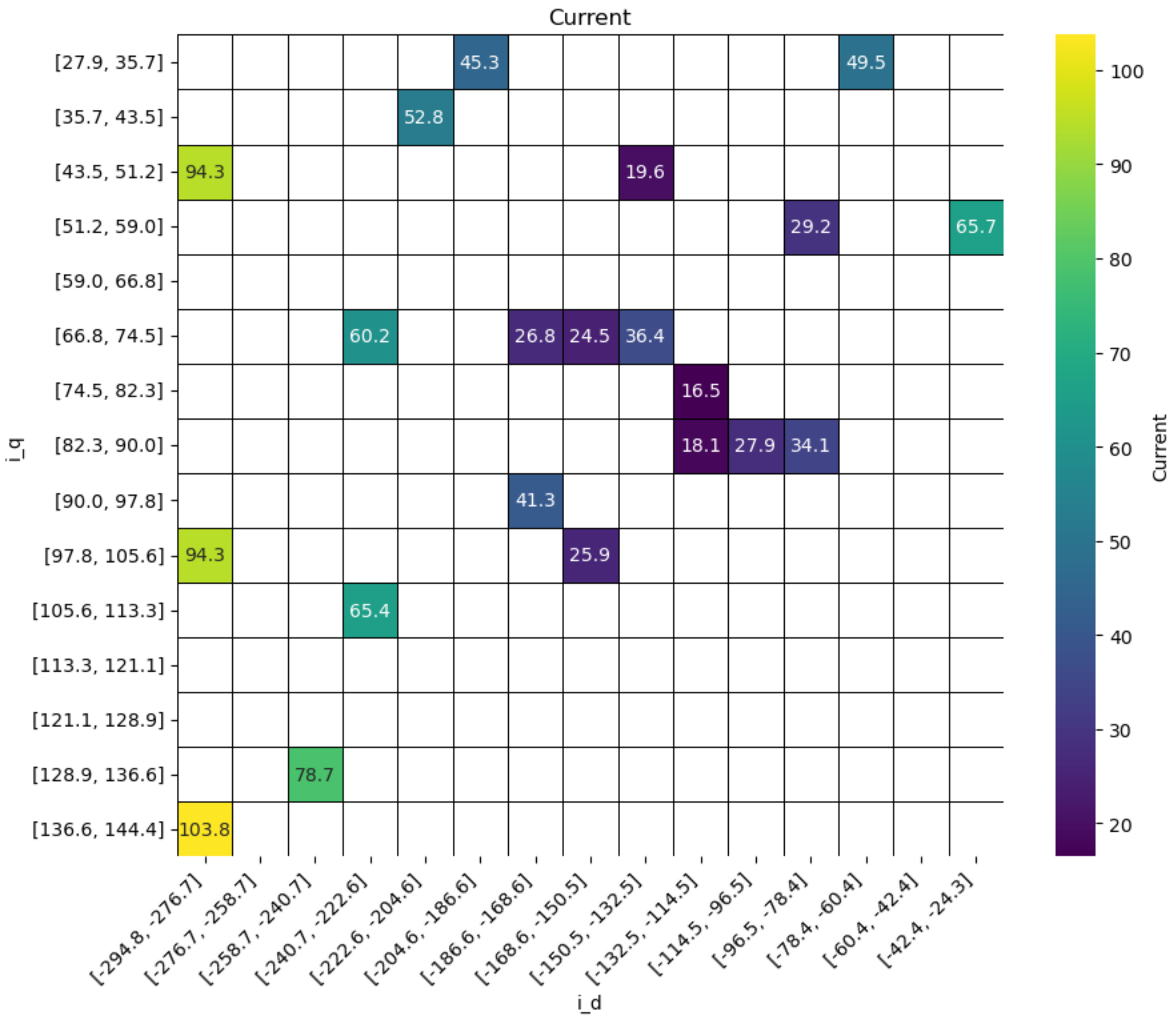
[64,64]

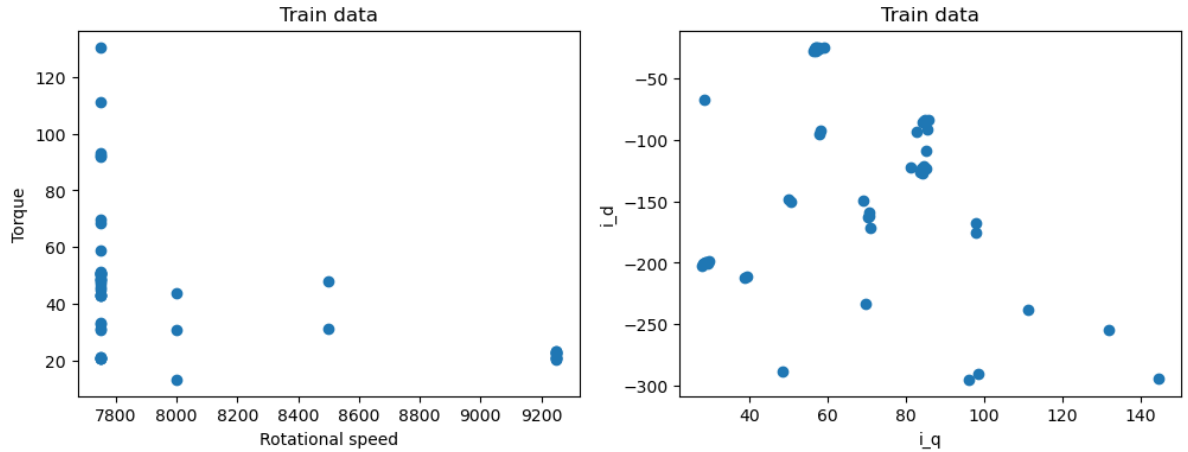


[32,32]



[16]


Current
[27.9, 35.7]
[35.7, 43.5]
[43.5, 51.2]
[51.2, 59.0]
[59.0, 66.8]
[66.8, 74.5]
[74.5, 82.3]
[82.3, 90.0]
[90.0, 97.8]
[97.8, 105.6]
[105.6, 113.3]
[113.3, 121.1]
[121.1, 128.9]
[128.9, 136.6]
[136.6, 144.4]
i_q
45.3
49.5
52.8
94.3
19.6
29.2
65.7
60.2
26.8
24.5
36.4
16.5
18.1
27.9
34.1
41.3
94.3
25.9
65.4
78.7
103.8
[-294.8, -276.7]
[-276.7, -258.7]
[-258.7, -240.7]
[-240.7, -222.6]
[-222.6, -204.6]
[-204.6, -186.6]
[-186.6, -168.6]
[-168.6, -150.5]
[-150.5, -132.5]
[-132.5, -114.5]
[-114.5, -96.5]
[-96.5, -78.4]
[-78.4, -60.4]
[-60.4, -42.4]
[-42.4, -24.3]
i_d
Current


Train data
Torque
Rotational speed
Train data
i_d
i_q

[16]