Chapter

Investigations of Different Approaches for Controlling the Speed of an Electric Motor with Nonlinear Dynamics Powered by a Li-ion battery - Case study

Roxana-Elena Tudoroiu, Mohammed Zaheeruddin,
Nicolae Tudoroiu, Sorin Mihai Radu and Hana Chammas

Abstract

This research investigated different nonlinear models, state estimation techniques and control strategies applied to rechargeable Li-ion batteries and electric motors powered and adapted to these batteries. The finality of these investigations was achieved by finding the most suitable design approach for the real-time implementation of the most advanced state estimators based on intelligent neural networks and neural control strategies. For performance comparison purposes, was chosen as case study an accurate and robust EKF state of charge (SOC) estimator built on a simple second-order RC equivalent circuit model (2RC ECM) accurate enough to accomplish the main goal. An intelligent nonlinear autoregressive with exogenous input (NARX) Shallow Neural Network (SSN) estimator was developed to estimate the battery SOC, predict the terminal voltage, and map the nonlinear open circuit voltage (OCV) battery characteristic curve as a function of SOC. Focusing on nonlinear modeling and linearization techniques, such as partial state feedback linearization, for "proof concept" and simulations purposes in the case study, a third order nonlinear model for a DC motor (DCM) drive was selected. It is a valuable research support suitable to analyze the performance of state feedback linearization, system singularities, internal and zero dynamics, and solving reference tracking problems.

Keywords: Li-ion battery, SOC, Simscape generic model, PID control, state feedback linearization, NARX shallow neural network, NARMA-L2 neuro controller

1. Introduction

Clean and efficient transportation across the planet is only possible if governments and scientists focus on stimulating and sustaining the automotive industry of electric vehicles (EVs) by developing and deploying the most advanced battery technologies. Nowadays, Li-ion battery technologies have made significant progress and have undoubtedly proven to have a promising future and great potential for development. These are recommended for their excellent features, such as lightweight, high-energy density, low memory effect and relatively low self-discharge, outperforming almost all other competing batteries of different chemistries on the market [1, 2]. Thanks to several improvements in Li-ion battery technologies recently, they have become safer, eliminating explosion hazards as much as possible and their chemistry is less toxic, both to nature and to humans. Battery state of charge (SOC) is an essential internal parameter that plays a vital role in utilizing battery energy efficiency, operating safely under various realistic conditions and environments, and extending battery life [3, 4]. The SOC is a piece of valuable information on the remaining capacity available during the operation of EV car. As the central internal state of the battery, the SOC is continuously supervised by a battery management system (BMS), which is integrated into the EV energy storage system (ESS) structure to power the traction powertrain [1–6]. The SOC can be calculated directly by a simple open-loop integration operation, known as the coulomb counting method or the ampere method since it accumulates the charge transferred between the battery and the environment over time. However, this measurement method is prone to the initial value of SOC and accuracy of the current profile data set measurement. The ampere method accumulates significant errors caused by the integration operation that accumulates errors over time [7]. The battery SOC estimation is one of the main tasks of a BMS. An extensive critical review of Lithium-ion battery SOC, and a smarter BMS description for EV applications are made in [8]. Being interpreted as a remaining capacity of the battery, the SOC is also an important support for energy management (EM) and control strategies. An interesting comprehensive review on Energy Management Strategies (EMS) for EVs taking into consideration the realistic conditions of Li-ion battery degradation based on aging models is found in HAL Open Science that includes the most representatives research papers from 2021 IEEE Access, with a new release version in 2023 [9]. The accuracy of Li-ion battery SOC estimation has a significant impact on the efficient operation and EMS of the battery. Many of studies are dedicated to advancing the BMS functions, such as intelligent cell balancing and charging control strategies for lithium-ion battery packs [10], SOC and state of health (SOH) monitoring [11–13], and thermal battery control temperature [14].

Nowadays, an impressive amount of work has been done in the research field to investigate and study large-scale new developments and implementations of SOC estimation algorithms to be applied to an extensive range of applications in the EV automotive industry. The main flaw of coulomb counting method is that it is not suitable in real time online SOC estimation. Also, it is noteworthy to know that the battery model accuracy significantly impacts SOC estimation. The well-known equivalent circuit model (ECM) is suitable for online estimation due to its simplicity and mastering well the relationship between parameters [1], [3–6]. The traditional methods include the most popular Kalman filter (KF) algorithms, among them linear KF and linearized extended KF (EKF) [5, 6, 15, 16], and nonlinear unscented KF (UKF) [7], ensemble KF (EnKF) [17], particles filter (PF) [18], which are commonly used as a nonlinear filter estimation methods. Only the linear KF is an optimal state estimator compared to the EKF, a suboptimal estimation algorithm. Still, it is an appropriate state estimator for complex working conditions with severe current fluctuations [7]. Compared to EKF, the UKF method uses an unscented transform to obtain the statistics of the process noise covariance and reaches a fast convergence speed and high estimation accuracy [7]. Also, its robustness is better when estimating the SOC of different chemistry batteries. To achieve higher accuracy of state estimation, various intelligent algorithms based on Machine Learning (ML) and Deep Learning (DL) Artificial Intelligence (AI) models are applied to the SOC estimation and terminal voltage prediction, as those developed in [7, 12, 13, 17, 19–38] easily to be adapted to all types of batteries and chemistries. The neural networks (NNs) learning techniques have a wide range of applications and are suitable for all types of batteries chemistry. Well, these learning techniques such as machine learning (ML) and deep learning (DL) models require large amounts of accurate training data [27–30]. The estimation accuracy and the convergence speed of the Li-ion battery SOC depend on the chosen training method, architecture structures, number of hidden layers and hidden neurons, learning rate, gradient value and on number of samples and epochs [29, 30]. The flow of this research paper is organized into four sections, as follows. For "proof concept" and simulation purposes, in Section 2 a generic Simulink Simscape model, simple and accurate is adopted to power a particular small EV car. The model parameters are extracted from a Simulink Simscape battery block set up for a preset model of the Li-ion battery. Additionally, the battery Simscape model is used for performance comparison with the adopted second-order ECM battery model (2RC ECM) used as support in Section 3 to build an accurate and robust EKF SOC estimator. In Section 3 is developed and implemented a NARX SNN intelligent SOC estimator. Section 4 is chosen as a case study of a DCM Drive nonlinear model of the high complexity of applying the state feedback linearization as a powerful tool for nonlinear control systems in a closed loop. Also, the model singularities, internal and zero dynamics stabilizability and reference tracking problems to solve represent some issues /challenges that merit being studied. Additionally, the traditional PID control strategy is a valuable tool used in this last section for performance comparison. At the end of Section 4, a learning NARMA-L2 controller intelligent strategy is applied to learn and linearize the DCM Drive nonlinear model. Therefore, this research work opens other directions of research to explore the application of clever neuro-control strategies on a large scale in future developments in the EV automotive industry.

2. Li-ion battery: model selection, accuracy, robustness, SOC estimation and terminal voltage prediction

In this research, we try to develop new approaches for identification, modeling, state estimation, and linear and nonlinear speed control strategies for a typical choice of DC or AC electric motor powered by a Li-ion battery and suitable for possible cars' mid-size EVs applications. Nowadays, we have a great opportunity to take advantage of the significant advances in modeling, identification and control systems inspired by the latest achievements in artificial intelligence, statistics and machine learning, deep learning, signal process analysis. Therefore, our research objectives are expanded with new approaches.

2.1 Li-ion battery model selection

In the case study, for "proof-of-concept" and simulation reasons, our selection strategy is to adopt a Li-ion battery model that meets the following requirements:

simple, accurate, easy to implement and suitable for online real-time simulations for a wide range of applications in the automotive industry of EVs.

The experience gained so far in process identification, modeling, design, and implementation confirms the existence of such a Li-ion battery model; depending on the complexity of the proposed research objectives, this could be an ECM, i.e., an electrical circuit consisting of a Thevenin-type voltage source OCV connected to the internal resistance Rint of the battery in series with one, two or three RC polarization cells. The number of RC cells substantially increases the accuracy of the model, which undoubtedly fulfills the above-mentioned characteristics, thus becoming valuable support for achieving research goals [3]. In our research, a second order ECM Li-ion battery model (abbreviated 2RC ECM) with a nominal voltage of 3.7 V and a rated capacity of 7.5 Ah was adopted, and the model was validated for different operating conditions in [3]. The electrical circuit diagram is shown in **Figure 1**, where the first polarization cell R1C1 captures the fast dynamics of the battery. In contrast, the second R2C2 polarization cell captures the slow dynamics of the Li-Ion battery. From the systemic perspective, it is a single-input, single-output (SISO) system with the input u = i indicating the charge (i < 0) or discharge (i > 0) current, and y = Vbat denoting the voltage at the output terminal of the battery. The voltages across the bias cells, Vc1 and Vc2, and the internal SOC state of the battery represent the internal states of the system. In addition, the proposed Li-ion battery SOC estimator based on the adopted model is expected to perform much better in terms of accuracy and robustness of battery SOC estimates for different operating conditions [3, 5, 7].

The adopted Li-ion battery (LIB) ECM model plays an important role in our research since it is valuable to support building a model-based EKF SOC and terminal voltage estimator, whose accuracy and robustness depended on the accuracy and robustness of the battery model. In addition, the ECM battery model is used for training data set generation to develop a data-driven intelligent neural network learning technique for SOC estimation and terminal voltage prediction, as a viable alternative to the traditional EKF estimator. The new modeling and estimation approach has proven its use for all battery types and different chemistries, outperforming traditional model-based state estimators. Their convergence does not depend on the complexity of model nonlinearity, unmodelled parts or model uncertainties. Also, intelligent learning techniques are much more suitable for real time online applications in the EVs industry. Model selection is also suggested due to its simplicity and ability to accurately describe the static and dynamic behavior of Li-ion battery.

The new modeling approaches use a specialized Simulink Simscape battery block to preset a specific Li-ion battery operating for different temperature ranges that

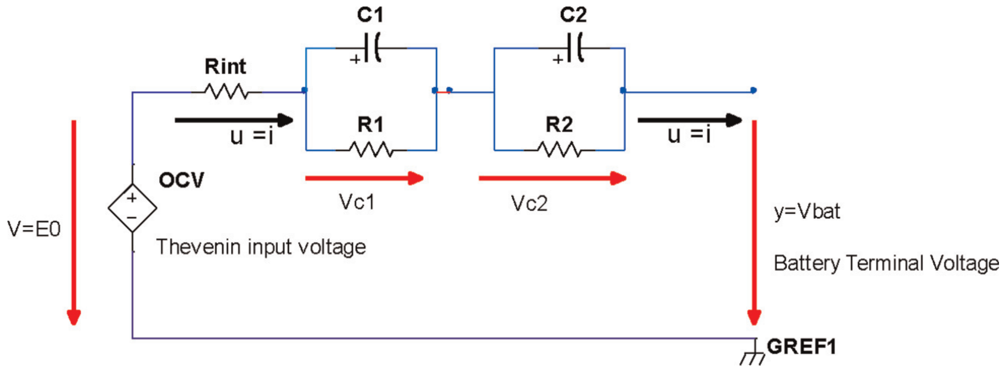


Figure 1.
2RC ECM wiring diagram.

significantly affect battery internal resistance $R_{\int}$, SOC, and RC components of the bias cells. Therefore, depending on user settings, Simscape multi-object models, built by a kind of specific object-oriented programming language, become valuable tools for generating different measurement input-output data sets for physical systems or subsystems. Then, these data sets are trained to develop and implement in MATLAB Simulink intelligent neural networks (NNs) learning techniques or to extract the model parameter values of a particular Li-ion battery, as well as to adjust and obtain the optimal values of them. For interested readers, such a Simscape block is shown in **Figure 2**.

Figure 3a and **b** show the discharge characteristics curves of rated current at 0.9, 5.4, 10.8, and 27 A versus battery capacity (Ah) as in **Figure 3a**, and time (h) in **Figure 3b**. The parameter values of a generic preset Li-ion battery model are also disclosed.

2.2 Li-ion 2RC ECM analytical model state space representation and continuous time domain

The analytical model of the Li-Ion battery is described in the continuous time domain by three first-order linear differential state equations and a highly nonlinear static output-state-input equation:

$$\frac{dx_1}{dt} = \frac{-\eta}{Q_{nom}} u(t) \tag{1}$$

$$\frac{dx_2}{dt} = \frac{-1}{R_1 C_1} x_2 + \frac{1}{C_1} = \frac{-1}{T_1} x_2 + \frac{1}{C_1}, T_1 = R_1 C_1 [s] \tag{2}$$

$$\frac{dx_3}{dt} = \frac{-1}{R_2 C_2} x_3 + \frac{1}{C_2} = \frac{-1}{T_2} x_3 + \frac{1}{C_2}, T_2 = R_2 C_2 [s] \tag{3}$$

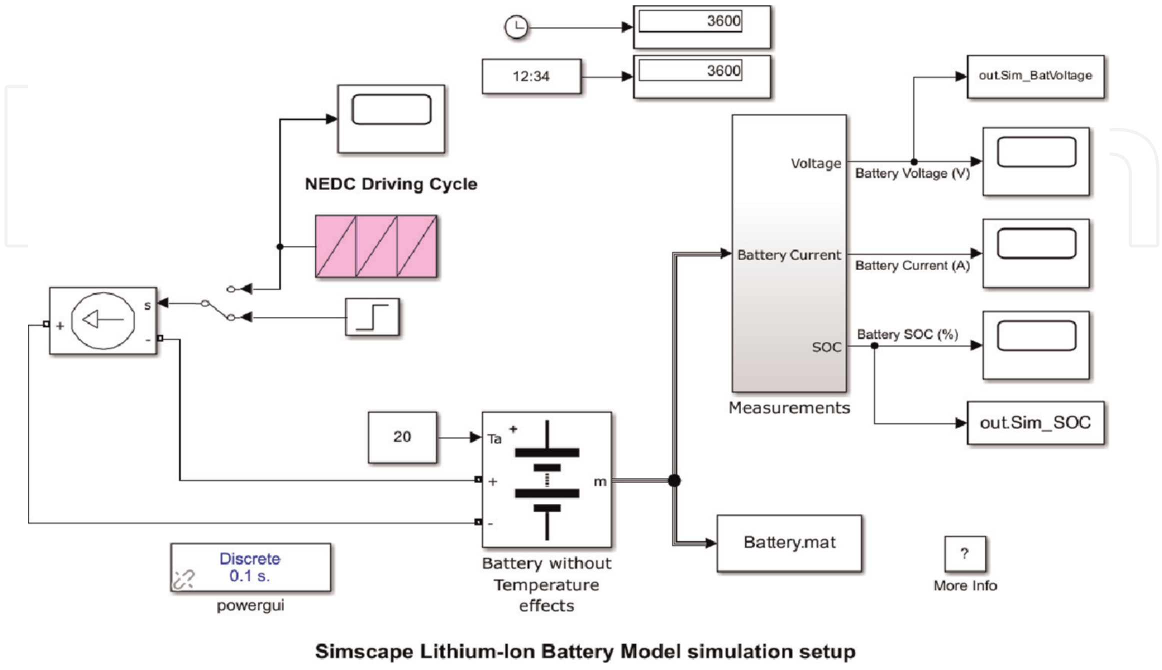


Figure 2.
Simulink Simscape block configured for Li-ion battery and NEDC driving cycle current profile test.

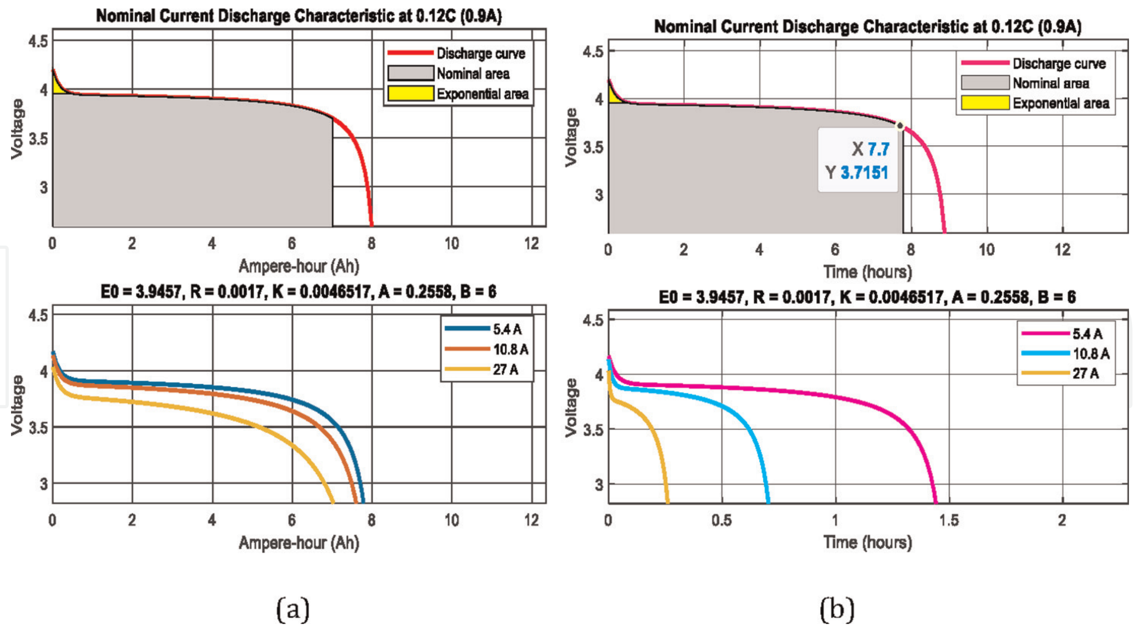


Figure 3.
Li-ion battery-rated current discharge characteristics and battery terminal voltage curves: (a) battery voltage-rated capacity curve (V-Ah); (b) battery voltage-time curve (V-h).

$$y = h(x_1) - x_2 - x_3 - R\int u(t) \tag{4}$$

where,

$$h(x_1) = OCV(x_1) = k_0 - k_1\frac{1}{x_1} - k_2x_1 + k_3\,ln\,(x_1) + k_4\,ln\,(1 - x_1) \tag{5}$$

is a nonlinear static function that represents the dependence of OCV on SOC, that is, OCV(SOC), and SOC = x_1, the first component of the state vector x, given by

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\begin{bmatrix} \ \end{bmatrix} \tag{6}$$

Both, T_1 and T_2, denote the time constant of the first and second polarization cells, respectively. Also, the constants η, and Q_{nom} designate the coulombic efficiency (η) for both, charging and discharging cycles, and the nominal capacity of the battery (Q_{nom}) respectively.

It is worth noting that, Eqs. (5) and (6) represent a combination of three separate models, namely Shepherd, Unnewehr and Nernst, reported in [3, 5] to predict terminal voltage based on SOC measurements. The combined model performs better than using either model separately. The constants $k_0, k_1, k_2, k_{3,}, k_4$ can be estimated using a least square estimation (LSE) procedure provided in the MATLAB System Identification Toolbox. To understand how LSE technique works, a simple offline (batch) processing method that calculates all these unknown parameters is well described in [3, 5]. For tuning these parameters and finding their optimal values, a Global Pattern Search (GPS) genetic algorithm from MATLAB Optimization Toolbox is a valuable tool to use [3].

2.3 Analytical model of Li-ion 2RC ECM in state space and discrete time domain representation

The Li-ion battery model described in the continuous-time state space representation can be discretized using a Taylor series expansion and then keeping only the first linear term, obtained by using the following approximation

$$\frac{d\boldsymbol{x_i}}{dt} = \boldsymbol{x_i} \tag{7}$$

where,

$\boldsymbol{T_s}$ is the sampling time, $\boldsymbol{t_k} = \boldsymbol{k} \times \boldsymbol{T_s}, \boldsymbol{k}\ \ \boldsymbol{Z}^+$ denotes the time instance, and $\boldsymbol{k}$ is a sample index

A matrix representation describes a compact Li-ion battery model of the following form:

$$x(k+1) = Ax(k) + Bu(k), y(k) = Cx(k) + Du(k) + h_1(x(k)) \tag{8}$$

where

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 - \dfrac{T_s}{T_1} & 0 \\ 0 & 0 & 1 - \dfrac{T_s}{T_2} \end{bmatrix}, B = \begin{bmatrix} \dfrac{-\eta T_s}{Q_{nom}} \\ \dfrac{T_s}{C_1} \\ \dfrac{T_s}{C_2} \end{bmatrix}, C = [-k_2 \quad -1 \quad -1], D = \tag{9}$$

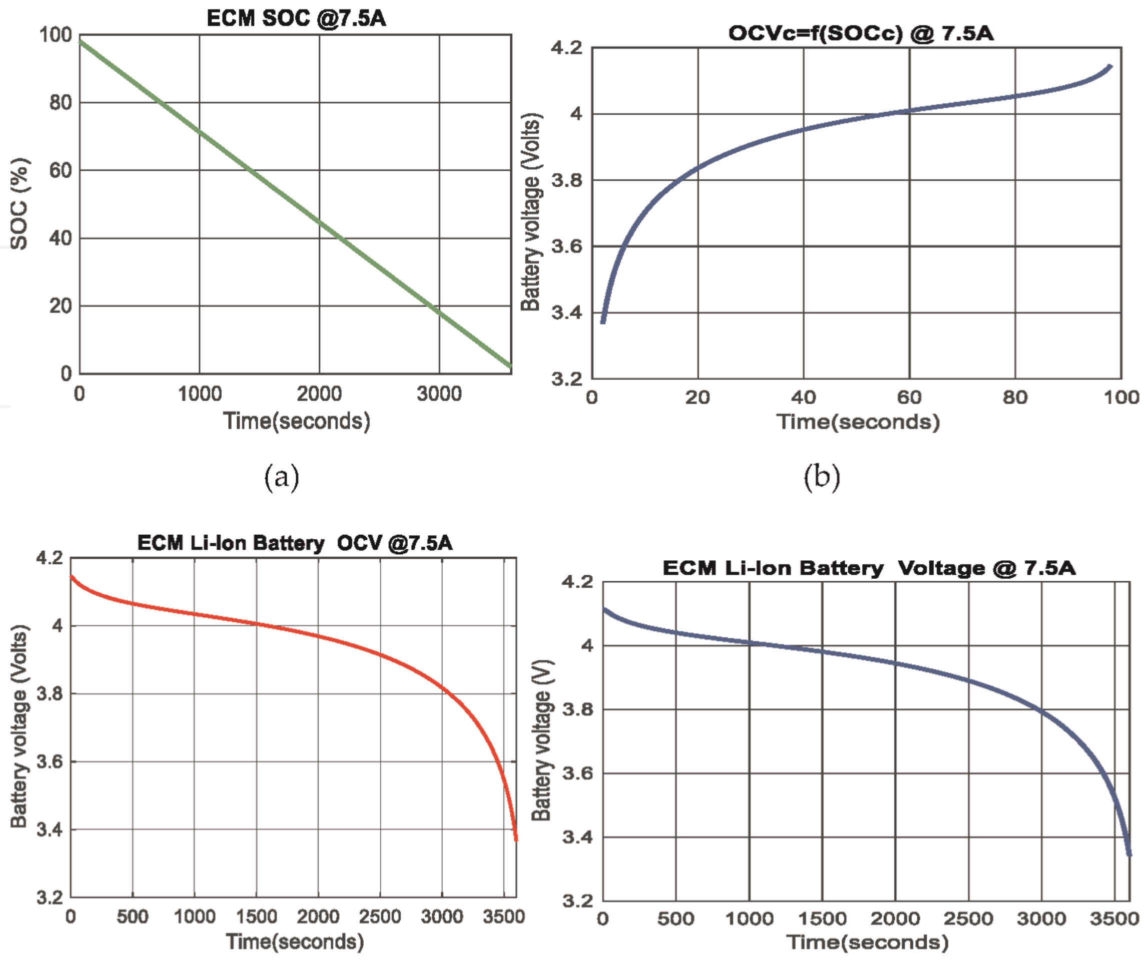
and,

$$h_1(x(k)) = k_0 - k_1 \frac{1}{x_1} + k_3\, ln\, (x_1) + k_4\, ln\, (1 - x_1) \tag{10}$$

The MATLAB simulation results obtained by iteratively solving the discrete-time Eqs. (8)–(10) describing the dynamics of the battery model over a finite time horizon defined by the number of samples are shown in **Figure 4a**–**c**. **Figure 4a** shows the SOC evolution of the battery over a full one-hour discharge cycle for a discharge current of 1C (7.5A), decreasing from 100% (fully charged battery) to 0% (fully discharged battery). **Figure 4b** reveals one of the main characteristics of the proposed Li-ion battery, which can be found in any catalog specifications of battery manufacturers. This well-known non-linear OCV-SOC curve differs from battery to battery. The time evolution of the battery terminal voltage and its OCV are illustrated in **Figure 4c** (Vbat) and 4d (OCV), respectively. The development of OCV and the terminal voltage of the Li-Ion battery is studied over the same full discharge cycle of 1h @7.5A (1C rate) discharge current.

3. SOC estimation and terminal voltage prediction of Li-ion battery

Section 1 details the vital role of the SOC battery in energy efficiency, safe operation and life extension. Also, information on the best-known state estimators and



(a)

(b)

(c)

(d)

Figure 4.
MATLAB simulation results @ constant current = 7.5 A: (a) discharged battery SOC during one-hour discharge cycle; (b) OCV characteristics of the Li-ion battery; (c) Li-ion battery terminal voltage during one-hour discharge cycle; (d) OCV battery during a one-hour discharge cycle.

parameters reported in the literature is provided to interested readers and implementers. Among these estimators, the most popular is the Kalman filter estimator in various versions, such as KF, EKF, UKF, and EnKF, as mentioned in the Introduction section. This section briefly introduces the EKF state estimator, which is easily adapted to estimate the SOC of the Li-ion battery proposed in the case study.

3.1 EKF state estimator design and MATLAB implementation

The Kalman Filter is reported in the literature field since 1960. It is "an optimal recursive model-based data processing algorithm for linear filtering purposes" [3].

A new modified form of KF, abbreviated EKF, was adopted for nonlinear process dynamics, a non-optimal filter estimator based on the linearization of process dynamics. Only KF is the optimal filter that addresses Gaussian linear processes whose measurement and process noises are zero-mean and uncorrelated of covariance matrices, e.g., Q for process noise and R for measurement noise, respectively. The state estimation of a process minimizes the minimum mean squared error, abbreviated MMSE, between state estimate values and true state model values given the measured input data set and the observed output data set [3, 5].

The EKF is a discrete-time recursive algorithm based on the LIB model whose states need to be estimated. For simplicity, the steps of the general iterative KF procedure when applied to linear systems are shown; it consists of the following two update phases [3–6, 15, 16]:

1. Prediction (forecasting) phase (time update) steps

   1.1 Predict the LIB state given in Eq. (8) one step ahead

$$\hat{x}_{k+1,k} = A\hat{x}_{k,k} + Bu(k) \tag{11}$$

   1.2 Predict one step ahead the error state covariance

$$\hat{P}_{k+1,k} = A\hat{P}_{k,k}A^T + Q_k \tag{12}$$

2. Correction phase (measurement update)

   2.1 Kalman gain computation

$$K_{k+1} = \hat{P}_{k+1,k}C^T\left(C\hat{P}_{k+1,k}C^T + R_{k+1}\right)^{-1} \tag{13}$$

   2.2 Update the state estimate with a new available measurement

$$\hat{x}_{k+1,k+1} = \hat{x}_{k+1,k} + K_{k+1}\left(y_{k+1} - C\hat{x}_{k+1,k}\right) \tag{14}$$

   2.3 Update error state covariance

$$\hat{P}_{k+1,k+1} = (I - K_{k+1}C)\hat{P}_{k+1,k} \tag{15}$$

Summarizing, in the prediction/forecast phase are predicted the LIB states $\hat{x}_{k+1,k}$and the covariance of the states $\hat{P}_{k+1,k}$. The second is a correction phase that occurs only if a new output measurementy_{k+1} (terminal voltage) is available, as it must calculate the Kalman gain $K_{k+1}$and, based on it, are updated the estimated state $\hat{x}_{k+1,k+1}$and its estimated covariance $\hat{P}_{k+1,k+1}$ [3, 5, 15, 16].

For general recursive EKF procedure, the matrices $A, B, C \wedge D$ are obtained by linearizing the following nonlinear functions around the last predicted estimate in each iteration:

$$x_{k+1} = f(x_k, u_k) + w_k \tag{16}$$

$$y_k = g(x_k, u_k) + v_k \tag{17}$$

where

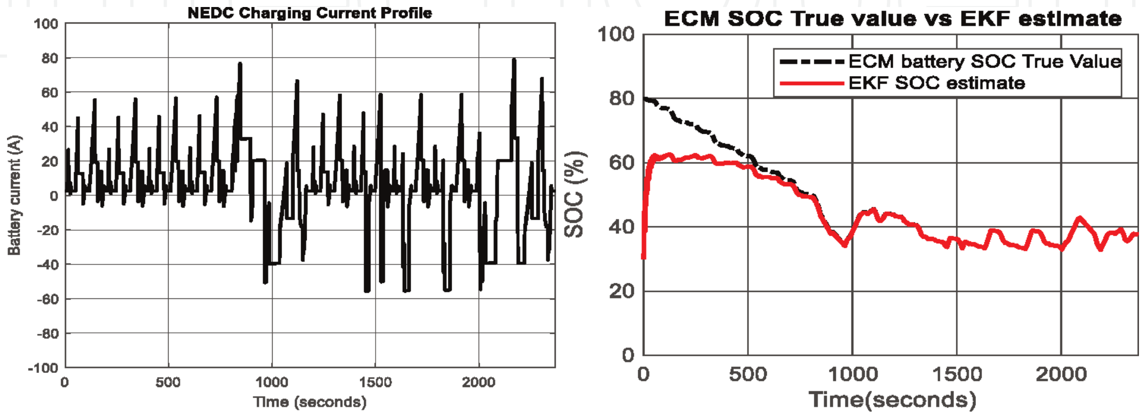
f, g are nonlinear time-varying functions, w_k and are both zero-mean and uncorrelated Gaussian process and measurement noises, respectively, with the statistics represented by the covariance matrices $Q_k \wedge R_k$ given or calculated at iteratively at the time instance zero mean and uncorrelated, of covariance matrix The matrices represent the so-called Jacobian matrices required in the Eqs. (11)–(15), defined as

$$A_k = \left(\frac{\partial f}{\partial x}\right) \hat{x}_{k,k}, C_k = \left(\frac{\partial f}{\partial x}\right) \hat{x}_{k,k} \tag{18}$$

This is just the unique difference between KF and EKF steps; the latter requires the calculation of the Jacobian matrices $A_k, \wedge C_k$ first and then the steps remain the same. It is worth noting that the matrices $A_k \wedge C_k$ are time-varying, compared to matrices (A, C) which have constant elements.

For a rigorous SOC performance analysis of the EKF algorithm in terms of accuracy and robustness, some of the MATLAB simulation results for the adopted 2RC ECM Li-Ion model, are shown below in **Figures 5**–**7** some of the MATLAB simulation results for the adopted 2RC ECM Li-Ion model. In **Figure 5**, the adopted Li-Ion model is tested with a New European Driving Cycle (NEDC) input current profile test, and the SOC of the battery is estimated following the steps of the EKF state estimation algorithm. For more details on the EKF state estimator, the reader can refer to [3–6, 29, 30]. **Figure 5a** shows the NEDC driving cycle current profile test, and **Figure 5b** show the SOC values generated by the 2RC ECM model versus the SOC EKF estimated values. **Figure 5b** reveals excellent SOC steady-state accuracy and robustness to changes in SOC initial value from 80% (in the model) to 30% in the initial SOC estimate value. Therefore, the EKF estimator performs very well with high SOC accuracy and robustness.

Instead, in **Figure 6a**–**d**, the EKF SOC estimator starts from the same initial SOC value that initializes the Li-Ion battery model; the MATLAB simulation results show now the battery model SOC versus its predicted EKF value, such as in **Figure 6a**, the model terminal voltage versus its predicted EKF value in **Figure 6b**, and, newly, both SOC and terminal voltage error residuals in **Figure 6c** and **d**, which provide valuable information on SOC accuracy and terminal voltage prediction. The lower values of both residuals (close to zero) indicate an excellent performance of the EKF SOC estimator and prove that the assumption regarding the accuracy of the proposed Li-ion battery model is correct; thus there is an accuracy transfer from the model to the EKF estimator.



(a) (b)

Figure 5.
MATLAB simulation results: (a) NEDC driving cycle current profile test; (b) Li-ion battery SOC ECM model versus EKF SOC estimator.

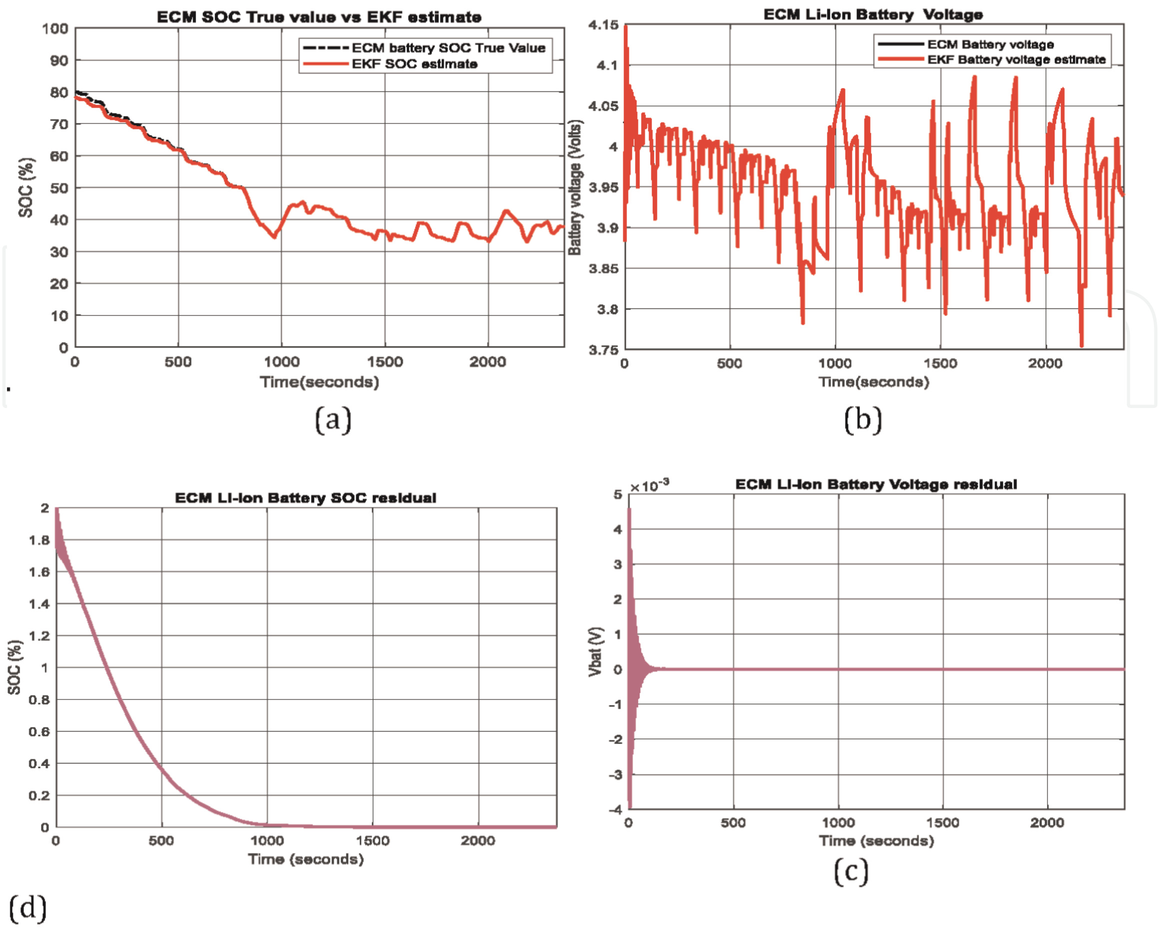
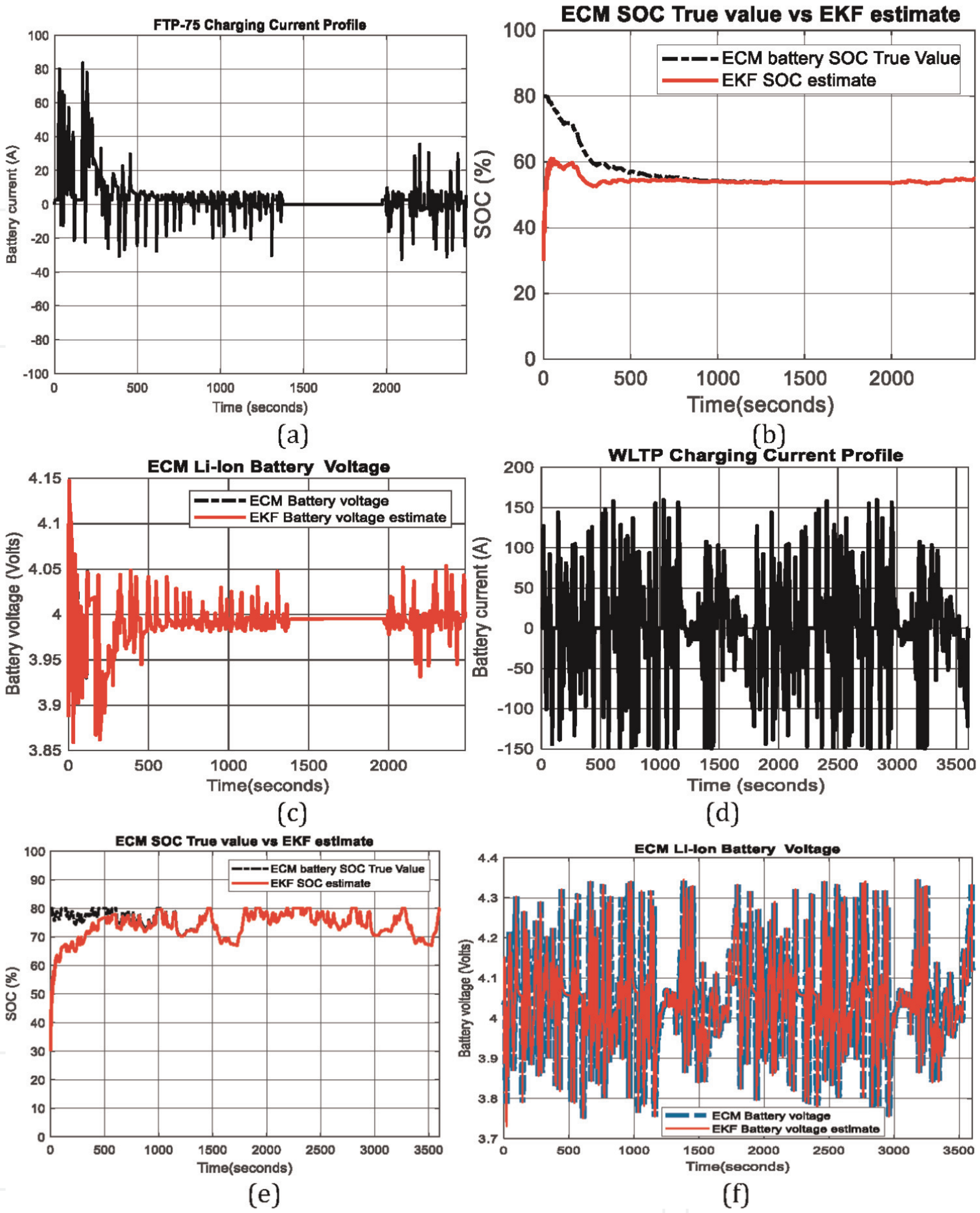


Figure 6.
MATLAB Simulation results: (a) Battery SOC ECM model vs. EKF SOC estimate (same initial conditions); (b) ECM terminal voltage vs. EKF predicted terminal voltage; (c) Battery SOC residual; (d) Battery terminal voltage residual.

Similarly, the MATLAB simulation results depicted in **Figure 7a–e** by changing the driving style, switching from a European NEDC cycle drive current profile test to an American Federative Transport Procedure FTP-75, shown in **Figure 7a**, and the response of the EV car is reflected in the high accuracy of the battery SOC shown in **Figure 7b**, an excellent prediction of the terminal voltage, as in **Figure 7c**. Again, the driving style is changed by moving from the American FTP-75 cycle drive current profile test to the latest European World Harmonized Light Transport Procedure (WLTP) with a more aggressive behavior, very close to the operating conditions of an environment realistic, shown in **Figure 7d**, and the impact reflected in **Figure 7e** by high SOC accuracy and excellent terminal voltage prediction is revealed in **Figure 7e**.

Summarizing this subsection, the EKF SOC estimator performs excellently in high SOC accuracy and terminal voltage prediction and with excellent robustness revealed by the MATLAB simulation results depicted in **Figures 5–7**. All these results are obtained for different operating conditions, mainly some changes in the initial battery SOC values and driving style by switching three driving cycles current profile tests, NEDC, FTP-75, and WLTP. In any of these scenarios the EKF SOC estimator performs excellently, with no significant impact from one driving scenario to another.



(a) (b) (c) (d) (e) (f)

Figure 7.
MATLAB simulation results – Robustness of EKF estimator to change the driving style and initial SOC value from 80 to 30%. (a) FTP-75 driving cycle current profile test; (b) ECM SOC vs. ECM EKF SOC estimate for FTP-75; (c) ECM terminal voltage vs. ECM EKF prediction terminal voltage; (d) WLTP driving cycle current profile test; (e) ECM SOC vs. ECM EKF SOC estimate for WLTP; (f) ECM terminal voltage vs. ECM EKF prediction terminal voltage.

3.2 Li-ion battery neural network learning techniques for nonlinear OCV characteristic estimation, SOC and terminal voltage estimation

This subsection explores a new approach to designing and implementing data-driven intelligent learning structures using neural networks (NNs). The intention is to use these smart structures for possible applications as state estimators. All these investigations aim to find a viable alternative to traditional model-based state estimators, such as the EKF used in the case study. This is why it is necessary to evaluate the capability and accuracy of their performance in a state estimation to see if they are suitable for mapping the non-linear characteristic OCV = f(SOC) characteristic of the battery and for estimating the battery SOC and terminal voltage. Therefore, three intelligent structures of nonlinear autoregressive neural networks with external input (NARX) are investigated. They are easily accessible from MATLAB Simulink Deep Learning Toolbox. Since they are data-driven structures, performance accuracy does not depend on battery model accuracy, modeling uncertainties, and the unmodeled part. In addition, they are suitable for real-time online applications and can be easily updated for all types of batteries and chemistries. NARX is a learning regression shallow neural network (SNN) estimator that consists of a single hidden layer with a certain number of hidden neurons, having a significant impact on the performance of neural structure. The input-output measurement data set is collected under the signals from the battery sensors using appropriate instruments provided by a specialized data acquisition (DAQ) system. Then the data set is processed for denoising, outliers removal, and data sharpness that significantly affect the performance accuracy. Essentially, these intelligent regressive neural network structures solve nonlinear time series problems using dynamic neural networks, including feedback networks [27–30]. They can be applied in open-loop, closed-loop, and open/closed-loop multistep prediction [27, 28]. Dynamic feedback networks can switch from open-loop to closed-loop to make multistep predictions, i.e. continue to predict when external feedback is missing using internal feedback [27, 30].

The design and implementation in MATLAB Simulink follow the steps inspired by [27–30] and are summarized in this subsection as follows:

1. Load the input (Predictors)-output (Target) data in the format required by the algorithm (e.g., sequences of cells, row or column vectors, matrices, tables, etc.)

2. Partition the data into training data XTrain and TTrain (in the case study the format is a sequence of cells), and data for prediction XPredict

3. Create a NARX network. Define the input delays, feedback delays, and size of the hidden layers, using the MATLAB

$$\text{net} = \text{narxnet}\ (1:5, 1:5, 65)$$

i.e., 0.5 samples delay for input and output, and 65 hidden neurons.

Prepare the time series data using a MATLAB-specific function ‘preparets’ and the MATLAB code line (the interpretation of the arguments Xs, Xi, Ai is given in step 6):

$$[\text{Xs}, \text{Xi}, \text{Ai}, \text{Ts}] = \text{preparets}\ (\text{net}, \text{XTrain}, \{\}, \text{TTrain})$$

4. Train the NARX network, using the MATLAB-specific functions ‘net’ and ‘train’, writing the following code line (in open-loop)

$$\mathrm{net} = \mathrm{train}\ (\mathrm{net}, \mathrm{Xs}, \mathrm{Ts}, \mathrm{Xi}, \mathrm{Ai})$$

5. Display the trained network diagram, using the MATLAB code line

$$\mathrm{view}\ (\mathrm{net})$$

6. Calculate the network output Y, final input states Xf and final layer states Af of the open-loop network from the network input Xs, initial input states Xi, and initial layer states Ai.

$$[\mathrm{Y}, \mathrm{Xf}, \mathrm{Af}] = \mathrm{net}\ (\mathrm{Xs}, \mathrm{Xi}, \mathrm{Ai})$$

7. Calculate the network performance using the specific MATLAB function ‘perform’ and the MATLAB code line:

$$\mathrm{perf} = \mathrm{perform}\ (\mathrm{net}, \mathrm{Ts}, \mathrm{Y})$$

8. Setup the closed-loop form of the NN architecture using the following MATLAB code line

$$[\mathrm{netc}, \mathrm{Xic}, \mathrm{Aic}] = \mathrm{closeloop}\ (\mathrm{net}, Xf, Af)$$

Remark 1. The final input states Xf and layer states Af of the open-loop network net become the initial information states Xic and layer states Aic of the closed-loop network netc.

9. Display the closed-loop network diagram, using following MATLAB code line

$$[\mathrm{netc}, \mathrm{Xic}, \mathrm{Aic}] = \mathrm{closeloop}\ (\mathrm{net}, Xf, Af)$$

10. Run the prediction for Ts-time steps ahead desired in closed-loop mode, using MATLAB code line

$$\mathrm{Yc} = \mathrm{netc}\ (\mathrm{XPredict}, \mathrm{Xic}, \mathrm{Aic})$$

$$\mathrm{Ycmatrix} = \mathrm{cell2mat}\ (\mathrm{Yc})$$

11. Plot Yc to visualize the simulation results

Following these steps, the MATLAB simulation results are presented below, followed by a performance analysis for all three NARX Shallow NN learning estimators.

3.2.1 The innovative NARX shallow neural network learning SOC estimator

The input-output data set is given by following sequence of cells, converted from a row vector format to a sequence of cells using the MATLAB line of code:

X = num2cell (u), u is the battery input sequence of current profile as Predictor.
T = num2cell (y), y is the battery terminal voltage output sequence as target.
XP = num2cell ([(u*]), u* is the Predicted input sequence proposed for test.

The MATLAB simulation results are depicted in **Figure 8a**–**d** with following interpretations:

a. Training phase progress;

b. NARX NN structure;

c. Battery SOC estimate versus the SOC ECM 2RC model (Target);

d. The best performance validation reached at epoch 242;

Performance analysis: The NN structure consists of a hidden layer with 60 hidden neurons and a 'sigmoid' activation function, an input layer with two input sequences, an output layer with one output sequence, and a linear 'pureline' activation function. The best validation performance is obtained at epoch 242. The learning rate is lr = 0.01, and a mean square error (MSE) accuracy performance of $2.79 \times 10^{-11} is$revealed in the **Figure 8a**,**c** and **d**.

In conclusion, with an MSE accuracy performance of 2.79×10^{-11} the NARX SOC estimator undoubtedly outperforms the EKF SOC estimator.

3.2.2 The innovative NARX shallow neural network learning OCV = f(SOC) estimation

X = num2cell (SOC), u is the input battery SOC as Predictor
T = num2cell (OCV), OCV is the battery OCV as Target
XP = num2cell ([(SOC*]), SOC* is the SOC Predicted input sequence for test purpose

The MATLAB simulation results are shown in **Figure 9a**–**c**, with the following interpretation for each subfigure

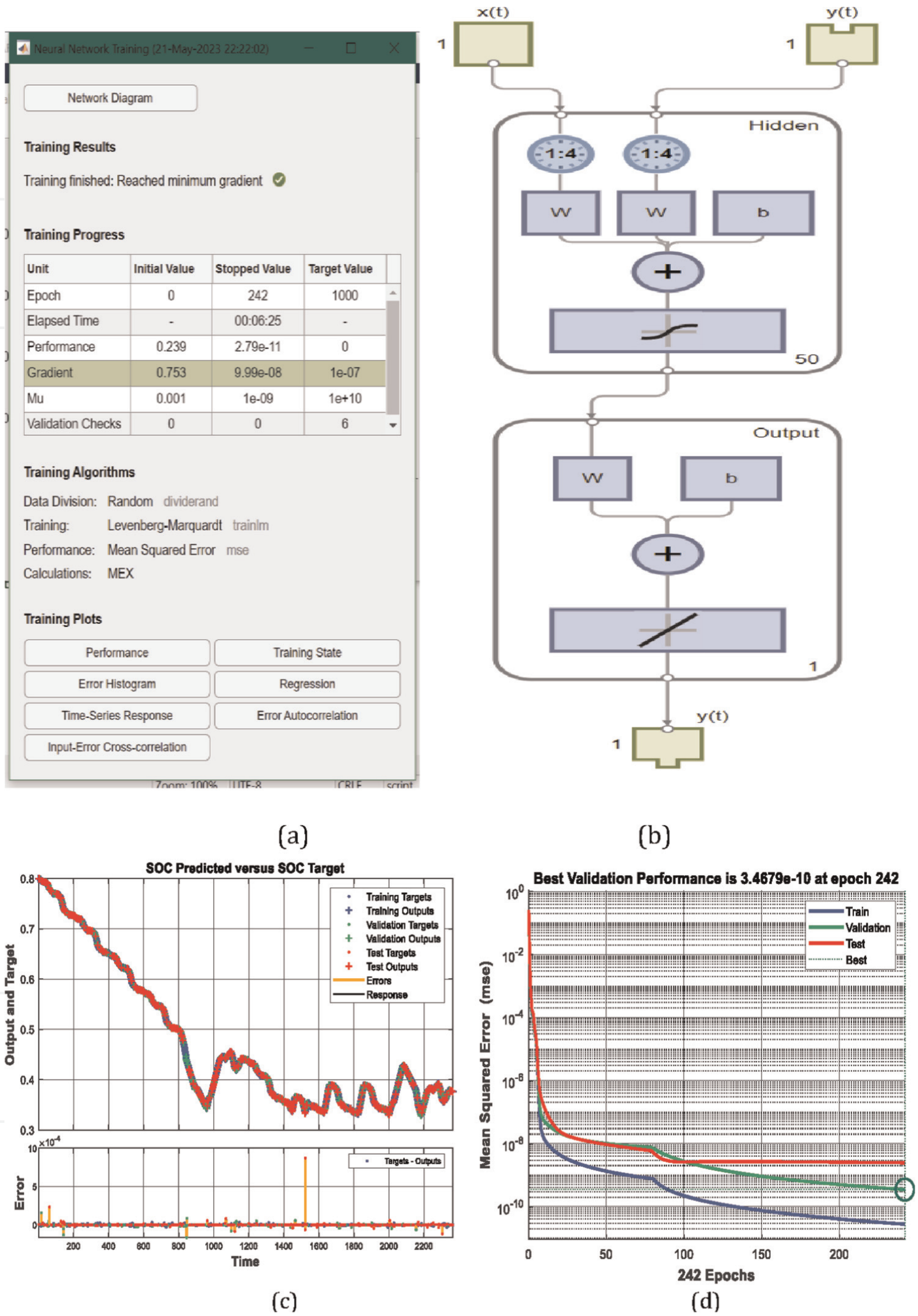
a. Li-Ion battery OCV estimate versus the OCV ECM 2RC model;

b. The best validation performance reached at epoch 45;

c. Training phase progress;

Performance analysis: The structure of the NN is identical to the first NN NARX. The best validation performance is achieved much faster than the first NARX estimator, i.e., at epoch 45. The learning rate is lr = 0.01, and a better mean square error (MSE) accuracy performance than the first NARX SOC estimator of 1.14×10^{-11} revealed in the **Figure 9a**–**c**.

In conclusion, with an MSE accuracy performance of 1.14×10^{-12},the NARX OCV estimator also outperforms the EKF OCV estimator.



Neural Network Training (21-May-2023 22:22:02)

Training Results

Training finished: Reached minimum gradient ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	242	1000
Elapsed Time	-	00:06:25	-
Performance	0.239	2.79e-11	0
Gradient	0.753	9.99e-08	1e-07
Mu	0.001	1e-09	1e+10
Validation Checks	0	0	6

Training Algorithms

Data Division: Random dividerand

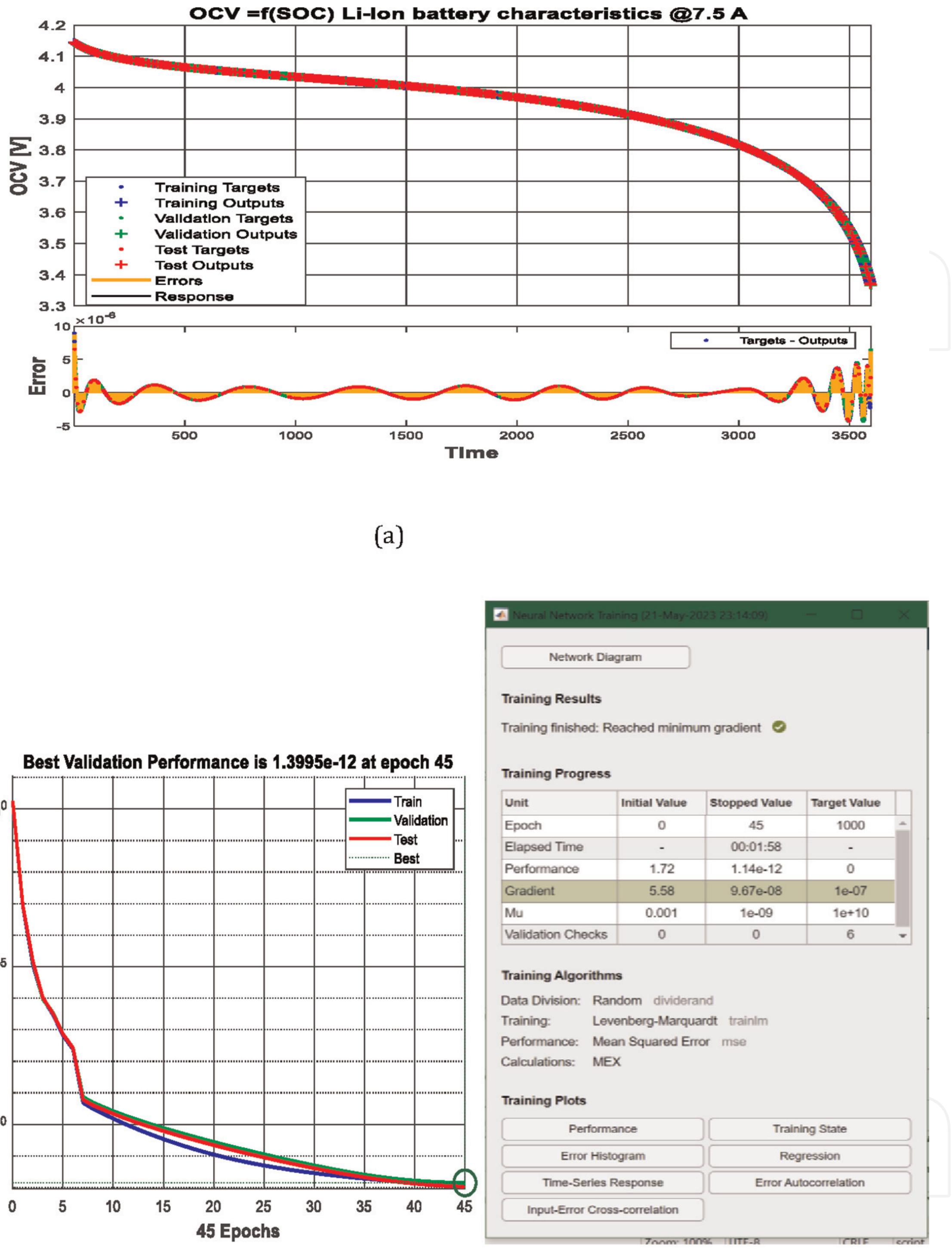
Training: Levenberg-Marquardt trainlm

Performance: Mean Squared Error mse

Calculations: MEX

Training Plots

Figure 8.
NN NARX SOC estimator of Li-ion battery: (a) training phase progress; (b) NARX NN structure; (c) battery SOC estimate versus ECM 2RC SOC model (Target); (d) best performance validation achieved in epoch 242.



Training Results

Training finished: Reached minimum gradient ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	45	1000
Elapsed Time	-	00:01:58	-
Performance	1.72	1.14e-12	0
Gradient	5.58	9.67e-08	1e-07
Mu	0.001	1e-09	1e+10
Validation Checks	0	0	6

Training Algorithms

Data Division: Random dividerand
Training: Levenberg-Marquardt trainlm
Performance: Mean Squared Error mse
Calculations: MEX

Training Plots

(a)

(b)

(c)

Figure 9.
NN NARX estimator of Li-ion battery OCV as a function of SOC: (a) Li-ion battery OCV estimate versus the ECM 2RC OCV model; (b) best validation performance achieved in epoch 45; (c) training phase progress.

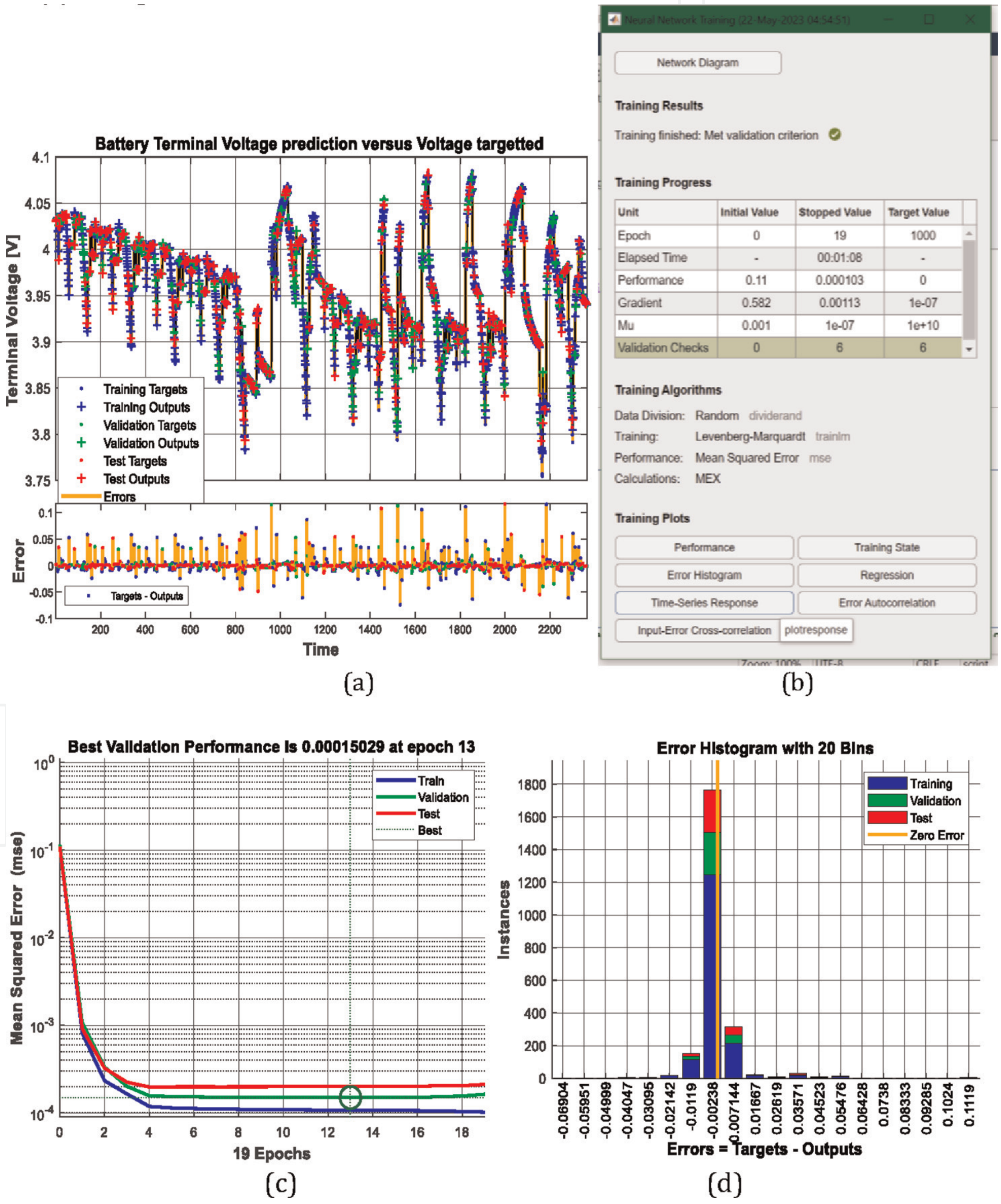
3.2.3 NN NARX Li-ion battery terminal voltage predictor

The input-output data set is given by following sequence of cells, converted from a row vector format to a sequence of cells using the MATLAB line of code:

X = num2cell ([SOC; u]), u denotes the input current profile and SOC the input battery (Predictors)
T = num2cell (y), y signifies the output terminal voltage (Target)
XP = num2cell ([SOC*; u]), is the Predicted input sequence proposed for the test

The MATLAB simulation results are revealed in **Figure 10a–d**, each capture of the figure with following interpretations:

a. Li-Ion battery terminal voltage predicted versus ECM 2RC terminal voltage model;



(a)

Neural Network Training (22-May-2023 04:54:51)

Training Results

Training finished: Met validation criterion ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	19	1000
Elapsed Time	-	00:01:08	-
Performance	0.11	0.000103	0
Gradient	0.582	0.00113	1e-07
Mu	0.001	1e-07	1e+10
Validation Checks	0	6	6

Training Algorithms

Data Division: Random dividerand
Training: Levenberg-Marquardt trainlm
Performance: Mean Squared Error mse
Calculations: MEX

Training Plots

(b)

(c)

(d)

Figure 10.
NN NARX terminal voltage predictor: (a) Li-ion battery terminal voltage predicted versus ECM 2RC terminal voltage model; (b) training phase progress; (d) histogram error; (c) best validation performance achieved after 19 epochs.

b. Training phase progress;

c. The best validation performance reached at 19 epochs;

d. Histogram error;

Performance analysis: The NN structure consists of one hidden layer with 60 hidden neurons and a 'sigmoid' activation function, an input layer with two input sequences, an output layer with one output sequence, and a 'pureline' linear activation function. The best validation performance is achieved at epoch 19. The learning rate is lr = 0.01, and a mean square error (MSE) accuracy performance of 1.03×4 shown in **Figure 8a**–**c**.

In conclusion, for an MSE accuracy performance of 1.03×10^{-4}, it can say that the NARX SOC estimator outperforms the EKF SOC estimator.

Summarizing all the investigations made in subsection 3.2, the valuable information provided by MATLAB simulation results depicted in **Figures 8**–**10** demonstrate that all three NN NARX estimators perform better than the traditional model-based EKF, and are indeed a viable alternative to conventional estimators using an EKF algorithm.

4. Electrical drives: nonlinear DC model with singularities, feedback linearization and closed-loop speed control strategies

Electric propulsion is a central system integrated into the architectural structure of an EV. It consists of three main subsystems: an AC or DC electric motor, power converters, and electronic controllers. A Li-ion battery powers the electric motor, like the one proposed in the case study. It converts the electric energy received from the battery into mechanical energy to propel the vehicle or to generate electricity during regenerative braking periods for recharging the same battery [33].

The power converter is controlled by an electronic controller to supply a regulated voltage and current to the electric motor. The main task of the controller is to generate a suitable control law (PID, fuzzy, neuro, sliding mode, state feedback linearization, etc.) such that the electric motor can produce the appropriate torque and speed converted by sensors (transducers) into electric signals through an interface circuitry [33]. These signals are then conditioned to the appropriate level and fed to be processed by a processor. For a good insight into the electric propulsion system the reader can find details in [33]. Choosing an electric drive system for a particular EV depends on the driver's expectations, the vehicle's constraints, and the power source. About driver's expectations (driving style) were mentioned in the previous section, three driving cycles, FTP-75, NEDC and WLTP, for analyzing robustness performance analysis of the EKF SOC estimator. Vehicle constraints, including "volume and weight, depend on vehicle type, weight, and payload" [33]. In the case study, the power source refers to the selection of a Li-ion battery such that its terminal voltage, rated capacity, and rated power are adapted to the electric motor of certain EVs. A traditional DC motor drive typically needs a commutator and brushes to "feed current into the armature, making it less reliable and unsuitable for maintenance-free operation and high speed" [33]. Also, the "wound-excited DC motors have a low specific power density"; however, "because of their mature technology and simple control, DC motor drives have been prominent in electric propulsion systems" [33]. Five constructive configurations of DC motors are manufactured: series excited, shunt excited, and compound excited, separately excited and permanent magnet (PM) excited motors, as mentioned in [33]. For “proof of concept” and simulation purposes, this section is limited to a specific nonlinear DC motor drive model, focusing on the main aspects, such as nonlinearities, feedback linearization, challenges/issues when faced with the presence of singularities, the design and implementation in MATLAB Simulink of suitable nonlinear control strategies to be applied in realistic environments for a large range of EVs. If the results obtained will satisfy the expectations, an extension of them to AC motor drives will be a big challenge. A nonlinear third-order separately excited DC motor drive model (i.e., three internal states: armature current, field current and rotor speed) is adopted for the case study.

4.1 Field controlled DC motor drive-nonlinear model

4.1.1 DC motor drive – physical model

Applying Kirchhoff’s voltage law (KVL) for both armature circuit and the field circuit, the DC motor drive model is described in continuous time by the following equations [35]:

For the field circuit: i_f is the field current, v_f is the field voltage:

$$L_f \frac{di_f}{dt} = v_f - R_f i_f \tag{19}$$

For the armature circuit: i_a is the armature current, v_a means the armature voltage, and $e_f = c_1 \omega i_f$ represents the back electromotive force (emf) induced in the armature circuit:

$$L_a \frac{di_a}{dt} = v_a - R_a i_a - c_1 \omega i_f \tag{20}$$

For shaft motion: $T = c_2 i_a i_f$ is the torque due to the interaction of armature current with the field circuit flux:

$$J \frac{d\omega}{dt} = c_2 i_a i_f \tag{21}$$

State-output equation:

$$y = \omega \tag{22}$$

4.1.2 DC motor drive: State space representation

After some mathematical manipulations the DC motor drive is described in a simple form of state space representation [32, 33]:

$$\frac{dx_1}{dt} = -a x_1 + u \tag{23}$$

$$\frac{dx_2}{dt} = -b x_2 - c x_1 x_3 + q \tag{24}$$

$$\frac{dx_3}{dt} = \theta x_1 x_2, \tag{25}$$

$$y = x_3 \tag{26}$$

where

$$x_1 = i_f, x_2 = i_a, x_3 = \omega \tag{27}$$

and, the numerical values are set to the same values as in [35]:

$$a = \frac{R_f}{L_f} = 103.995, b = \frac{R_a}{L_a} = 35.4034, c = \frac{c_1}{L_a} = 1.45, q = \frac{v_a}{L_a} = 52.7588 \tag{28}$$

$$\theta = \frac{c_2}{J} = 230.769, \omega_0 = 10\left[\frac{rad}{s}\right] \tag{29}$$

The DC motor drive model given by Eqs. (23)–(26) is nonlinear since the product x_1x_3 appears in the second equation, and the product x_1x_2 appears in the third equation.

4.1.3 DC motor drive: State feedback linearization

The DC motor drive model can be writing in a general vectorial field form:

$$\frac{dx}{dt} = f(x) + g(x)u \tag{30}$$

$$y = h(x) \tag{31}$$

where the field vectors f(x), g(x) and h(x) have the vectorial form:

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \end{bmatrix} = \begin{bmatrix} -ax_1 \\ -bx_2 - cx_1x_3 + q \\ \theta x_1 x_2 \end{bmatrix}, g(x) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, h(x) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{32}$$

Preliminaries: Feedback Linearization, Lie derivative definition, Diffeomorphism – Definitions.

Definition 1. A nonlinear system described in a state space representation (30) and (31), where the nonlinear functions f: $D_0 \mapsto R^n$, $g : D_0 \mapsto R^n$ are sufficiently smooth on a domain $D_0 \subset R^n$ is said to be feedback linearizable (or input-state linearizable) if there exists a diffeomorphism, $T : D_0 \mapsto R^n$ such that $Dz = T(D_0)$ contains the origine and the change of variable $z = T(x)$ transforms the system given by Eqs. (30) and (31) into a linear system of the form

$$\frac{dz}{dt} = Az + B\gamma(x)[u - \alpha(x)] \tag{33}$$

that has the pair of the matrices (A, B) controllable and $\gamma(x)$ inversible (or nonsingular) for all $x \in D_0$,as is also mentioned in [34].

Remark 2. If linearizing the state equation does mean that at a same time is also linearized the output equation [34].

Remark 3. The state feedback control or partially feedback linearizable systems is an excellent control tool that can solve both system stabilization and reference tracking control [34].

Definition 2. If the DC motor drive described by Eqs. (30) and (31) can be put in the form given by Eq. (33) then it can be linearized via the state feedback, choosing for the input u the following control law [34]

$$u = \alpha(x) + \beta(x)v \tag{34}$$

and

$$\beta(x) = \gamma(x)^{-1} \tag{35}$$

To stabilize the DC motor drive model, the new control law v introduced in Eq. (34) is designed as

$$v = -Kx \tag{36}$$

such that the matrix $(A - BK)$ is Hurwitz, equivalent to say that the pair (A, B) is controllable (stabilizable).

The original nonlinear stabilizing state feedback control in closed-loop is given by

$$u = \alpha(x) - \beta(x)Kx \tag{37}$$

Remark 4. The DC motor drive state feedback linearized (FL) model is not unique and depends on the choice of the state variables.

Definition 3. (Lie derivative operator). Assuming that the vector field functions f(x), g(x) and h(x) are sufficiently smooth in a domain $D_0 \subset R^n$, then the mapping functions f and g on $D_0 \subset R^n$ and the first derivative of the output variable $\frac{dy}{dt}$ can be put in the following form [34–36]

$$\frac{dy}{dt} = \frac{dy}{dx}\frac{dx}{dt} = \frac{\partial h}{\partial x}[f(x) + g(x)u] \triangleq L_f h(x) + L_g h(x)u \tag{38}$$

were the Lie derivative operators $L_f h$ and $L_g h$ defined as

$$L_f h(x) = \frac{\partial h}{\partial x} f(x), L_g h(x) = \frac{\partial h}{\partial x} g(x) \tag{39}$$

are called the Lie derivatives of h to respect to f (or equivalent, along f), and of h to respect to g (along g), respectively.

In fact, the notation $L_f h(x) = \frac{\partial h}{\partial x} f(x)$is an adapted concept of the derivative of h along the trajectories of the system $\frac{dx}{dt} = f(x)$. It is an appropriate formal procedure for repeating the calculation of the derivative with respect to the same vector field or a new one [34], such as

$$L_g L_f h(x) = \frac{\partial(L_f h)}{\partial x} g(x), L_f^0 h(x) = h(x), L_f^2 h(x) = L_f L_f h(x) \tag{40}$$

Or, in a general repetitive form given by

$$L_f^k h(x) = L_f L_f^{k-1} h(x) = \frac{\partial\left(L_f^{k-1} h\right)}{\partial x} f(x) \tag{41}$$

Remark 5. The repetitive notation (41) applied on DC motor drive output y, leads to the following main result of using the full state feedback linearization (FSFL):

If $L_g h(x) = 0$, it means that

$$\frac{dy}{dt} = L_f h(x), \tag{42}$$

and can be interpreted as independence of function $\frac{dy}{dt}$ with respect to the input u.

If $L_g L_f h(x)$=0, then

$$y^{(2)} = \frac{d}{dt}\left(\frac{dy}{dt}\right) = \frac{d}{dt}\left(L_f h(x)\right) = \frac{\partial\left(L_f h\right)}{\partial x}\frac{dx}{dx} = \partial\frac{(L||fh)}{\partial x}[f(x) + g(x)u] \tag{43}$$

Then, the final result is

$$y^{(2)} = L_f^2 h(x) + L_g L_f h(x) u = L_f^2 h(x) \tag{44}$$

and it means that the second derivative of the output$y^{(2)}$, is also independent of input u.

Iteratively, by repeating the procedure, and

If $L_g L_f^{k-1} h(x)$= 0, for $k = 1, \overline{\rho - 1}$, and $L_g L_f^{\rho-1} h(x) \neq 0$, therefore in the first $(\rho - 1)$ derivatives of the output y the input u does not appear, it only occurs in the last derivative $y^{(\rho)}$ [34]

$$y^{(\rho)} = L_f^\rho h(x) + L_g L_f^{\rho-1} h(x) u \tag{45}$$

The last equation (45) is showing without doubt that the model is input-output linearizable, of ρ the relative degree, and is in the required form given in Eqs. (30) and (31). The control law u, can be written in the following form

$$u = \frac{L_f^\rho h(x)}{L_g L_f^{\rho-1} h(x)}\left[-L_f^\rho h(x) + v\right] \tag{46}$$

that leads to a following form for the input-output map [34]

$$y^{(\rho)} = v \tag{47}$$

as the result of a chain of ρ integrators applied to the output y.

For the normal canonical form, the new control law v from Eq. (46) is chosen as

$$v = -Kx \tag{48}$$

where, the feedback gain K is calculated using a pole placement procedure, using the following MATLAB code line:

$$K = place\left(A, B, \left[p_1, p_2, \dots, p_\rho\right]\right) \tag{49}$$

where $p_1, p_2, \ldots, p_\rho$ represent the poles' locations in the left-half plane of the complex domain, such as the matrix $(A - BK)$ is Hurwitz (the system in closed-loop and canonical form is stable).

If the relative degree $\rho = n = dim(x \in R^x)$, the state-space dimension, then the system is full-state linearizable (FSL) and stabilizable. Also, the transformation $T : D_0 \mapsto R^n$ with $z = T(x)$ that convert the system given by Eqs. (30)–(31) into the form defined in Eq. (33) is a diffeomorphism. The new variables z are defined as follows [34]:

$$z = T(x) = \begin{bmatrix} h(x) \\ L_f h(x) \\ \cdot \\ \cdot \\ L_f^{n-1} h(x) \end{bmatrix} \tag{50}$$

that converts the system into normal minimal form defined by [34]

$$\frac{dz}{dt} = A_c z + B_c \gamma(x)[u - \alpha(x)] \tag{51}$$

$$y = C_c z \tag{52}$$

The following choice of the input u [21, 22]

$$u = \alpha(x) + \gamma(x)^{-1} v \tag{53}$$

brings the system in a canonical form [34–36]

$$\frac{dz}{dt} = A_c z + B_c v, y = C_c z \tag{54}$$

Therefore, the state transformation defined in Eq. (50) is a full-state linearization, and in the same time also, is an input-output linearization [34].

In the case when the relative degree $\rho < n$ then the state feedback linearization problem is brining some complications (issues) such singularities and instability. In this case the transformation $z = T(x)$ converts the original system into following form [34]

$$z = T(x) = \begin{bmatrix} \eta \\ \ldots .. \\ \xi \end{bmatrix} = \begin{bmatrix} \phi_1(x) \\ \cdot \\ \cdot \\ \phi_{n-\rho}(x) \\ \ldots \ldots \ldots .. \\ h(x) \\ L_f h(x) \\ \cdot \\ \cdot \\ L_f^{\rho-1} h(x) \end{bmatrix} = \begin{bmatrix} \Phi(x) \\ \ldots \ldots \ldots \\ \Psi(x) \end{bmatrix} \tag{55}$$

where the functions $\phi_1(x)$, $\phi_2(x)$, … …, $\phi_{n-\rho}(x)$are selected such that the transformation $T(x)$ still to remains a diffeomorphism. Additionally, in [34] is stated that such of functions $\phi_i(x), i = 1, \overline{n - \rho}$, do exist at least locally, satisfying the following conditions:

$$\frac{\partial \phi_i(x)}{\partial x} g(x) = 0, \forall i = 1, \overline{n - \rho} \tag{56}$$

The transformation given by Eq. (55) converts the system into the following canonical form

$$\frac{d\eta}{dt} = f_0(\eta, \xi) \tag{57}$$

$$\frac{d\xi}{dt} = A_c \xi + B_c \gamma(x)[u - \alpha(x)] \tag{58}$$

$$y = C_c \xi \tag{59}$$

$$u = \alpha(x) + \gamma(x)^{-1} v \tag{60}$$

and,

$$f_0(\eta, \xi) = \frac{d\Phi(x)}{dx} = \frac{\partial \Phi(x)}{\partial x} f(x)_{x = T(x)^{-1}}, \gamma(x) = L_g L_f h(x), \alpha(x) = \frac{-L_f^{\rho} h(x)}{L_g L_f^{\rho - 1} h(x)} \tag{61}$$

The following main valuable results can be extracted from this general modeling approach:

1. Internal dynamics of the system, described by the Eq. (57)

$$\frac{d\eta}{dt} = f_0(\eta, \xi)$$

It is noticeable that this dynamic is unobservable for the system output y, therefore it does not directly affect the other states ξ or the system output y since it does not appear in Eq. (58) or Eq. (59).

For the internal dynamics of the system is required to solve the **stabilization problem** around the origin of its zero dynamics

2. Zero dynamics of the internal dynamics of the system

It is obtained for $\xi = 0$, and first equation of the system dynamics in normal form becomes

$$f_0(\eta, 0) = 0$$

3. Input-output linearization of the system

The system defined by Eqs. (58) and (59) is a linear system for which two critical closed-loop problems can be solved:

a. Stabilization problem

b. Tracking problem to an input reference, constant or time varying, with zero steady-state error.

Interestingly, by solving the tracking problem for the linear part of the system dynamics described by Eqs. (58) and (59), the general tracking problem for the nonlinear system is also solved, but only if the internal dynamics of the same system is stable at the origin of zero dynamics of the internal dynamics of the system given by $f_0(\eta, 0) = 0$.

Repeating the Lie derivative operator in the Eq. (45) three times on DC motor drive output y, the following results are obtained

$$\frac{dy}{dt} = \frac{dx_3}{dt} = \theta x_1 x_2, \; \frac{d^2y}{dt^2} = \theta \frac{dx_1}{dt} x_2 + \theta x_1 \frac{dx_2}{dt} = M(x_1, x_2, u) + N(x_1, x_2, x_3) \tag{62}$$

where
$M(x_1, u) = \theta(-ax_1 + u)x_2$, depends on u, instead.
$N(x_1, x_2, x_3) = \theta x_1(-bx_2 + q - cx_1 x_3)$ it is not dependent on u.
therefore,

if $x_2 \neq 0$, then $\frac{d^2y}{dt^2} \neq 0$, and since u appears in the second derivative the relative degree of the DC motor drive model is $\rho = 2$. In conclusion, the DC motor drive model has $\rho = 2$ on the set $D = \{x \in R^3 | x_2 \neq 0\}$, so it is worth to notice a singularity of the DC motor drive model when the state $x_2 = 0$.

A transformation of state variables $z = T(x)$ such is defined in Eq. (55) has the form

$$z = T(x) = \begin{bmatrix} \phi_1(x) \\ h(x) \\ L_f h(x) \end{bmatrix} \tag{63}$$

By calculation,

$$L_f h(x) = \frac{\partial h}{\partial x} f(x) = [001] \begin{bmatrix} -ax_1 \\ -bx_2 - cx_1 x_3 + q \\ \theta x_1 x_2 \end{bmatrix} = \theta x_1 x_2$$

$\frac{\partial \phi_1}{\partial x} g(x) = \left[\frac{\partial \phi_1}{\partial x_1} \; \frac{\partial \phi_1}{\partial x_2} \; \frac{\partial \phi_1}{\partial x_3}\right] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \frac{\partial \phi_1}{\partial x_1} = 0,$ according to the constraint given in Eq. (56) to exist the function $\phi_1(x)$ and the transformation $z = T(x)$ to be diffeomorphism on the set $D = \{x \in R^3 | x_2 \neq 0\}$.

The equation $\frac{\partial \phi_1}{\partial x_1} = 0$ shows an independence of function $\phi_1(x)$ on x_1. A simpler solution for this equation is only $\phi_1(x) = x_2, \vee \phi_1(x) = x_3$, but the second solution is not good since the transformation $z = T(x)$ is not more a diffeomorphism.

One of the possible solutions can be chosen $\phi_1(x) = x_2$, even if exist other choices, but not simple than this last selection [34].

So, we can write that $z = T(x) = \begin{bmatrix} x_2 \\ x_3 \\ \theta x_1 x_2 \end{bmatrix} = \begin{bmatrix} \eta \\ \xi_1 \\ \xi_2 \end{bmatrix}$ and after some math manipulations the dynamics in the new variables can be written as

a. The internal dynamics is given by:

$$\frac{d\eta}{dt} = -b\eta + q - c\frac{\xi_1\xi_2}{\theta\eta} = f_0(\eta, \xi), \xi = (\xi_1, \xi_2) \tag{64}$$

b. The normal form of the DC motor drive dynamics is given by:

$$\frac{d\xi_1}{dt} = \xi_2 \tag{65}$$

$$\frac{d\xi_2}{dt} = \theta x_2\left(u - (a+b)x_1 + q\frac{x_1}{x_2} - c\frac{x_1^2 x_3}{x_2}\right) \tag{66}$$

$$y = \xi_1 \tag{67}$$

For a simple choice of the input u to cancel all the nonlinearities in Eq. (66) such as

$$u = (a+b)x_1 - q\frac{x_1}{x_2} + c\frac{x_1^2 x_3}{x_2} + \frac{1}{\theta x_2}v \tag{68}$$

For this selection of the input u, the normal form of the DC motor drive dynamics is linear and described by two linear first order differential equations and a linear state output equation

$$\frac{d\xi_1}{dt} = \xi_2, \frac{d\xi_2}{dt} = v, y = \xi_1 \tag{69}$$

Since the DC motor drive model is linearized, both stabilization and tracking problems will be easily solved, but only if the stability of internal dynamics must be ensured, required to ensure the existence of the function $\phi_1(x) = x_2$, and the transformation $z = T(x)$ to be diffeomorphism [34]. The analysis of the internal stabilization of the DC motor drive model dynamics is performed by checking if the equation

$$\frac{d\eta}{dt} = -b\eta + q - c\frac{\xi_1\xi_2}{\theta\eta} = f_0(\eta, \xi) \tag{70}$$

is asymptotic stable around the origin $\xi = (\xi_1, \xi_2) = (0, 0)$, known also as zero dynamics, given by the following equation.

$f_0(\eta, 0) = 0$, so equivalent for asking that the equation given below is stable around an equilibrium point $(\eta, \xi) = \left(\eta_{ss} = \frac{q}{b}, \xi = 0\right)$.

$$\frac{d\eta}{dt} = -b\eta + q \tag{71}$$

Lemma 1. "The origin of the closed loop system is asymptotic stable if the origin of zero dynamics is asymptotic stable", as is stated in [34].

According to this Lemma, the DC motor drive connected in closed-loop is asymptotic stable only if the origin of its zero dynamics model is asymptotic stable. In other words, the asymptotic stability of zero dynamics guarantees the asymptotic stability of closed loop linearized system.

4.2 Zero dynamics interpretation

As long as $b > 0$, the origin of zero dynamics (71) of the DC motor drive model is exponentially asymptotic stable since the solution of the homogeneous equation (71) is given by:

$$\eta_0(t) = e^{-bt}\eta(0) \tag{72}$$

since $\eta_0(t) \to 0, when t \to \infty$, then the origin of zero dynamics is exponentially asymptotic stable, and according to Lemma 1, the origin of the closed-loop system is also asymptotic stable.

Conclusion: If zero dynamics of the system is not asymptotically stable then the linearization of the nonlinear system is not possible to be realized.

In closed loop, the input u (DC motor armature voltage) is defined in the original state variables x_1, x_2, x_3 as

$$u = (a+b)x_1 - q\frac{x_1}{x_2} + c\frac{x_1^2x_3}{x_2} - \frac{1}{\theta x_2}v, \wedge v = -(k_1\xi_1 + k_2\xi_2) \tag{73}$$

and replacing ξ_1, ξ_2 by $\xi_1 = x_3, \xi_2 = \theta x_1 x_2$, u becomes:

$$u = (a+b)x_1 - q\frac{x_1}{x_2} + c\frac{x_1^2x_3}{x_2} - \frac{1}{\theta x_2}(k_1x_3 + k_2\theta x_1 x_2) \tag{74}$$

The Simulink model of the open loop and closed loop of the DC motor drive is shown in **Figure 11**.

The MATLAB Simulink simulations results for the nonlinear DC motor drive (DCMD) connected in open loop are shown in **Figure 12a** and **b** for field current $i_f = x_1$, armature current $i_a = x_2$, and DCMD speed ω=x_3 [rad/s] for a step input voltage test, u = 12 [V].

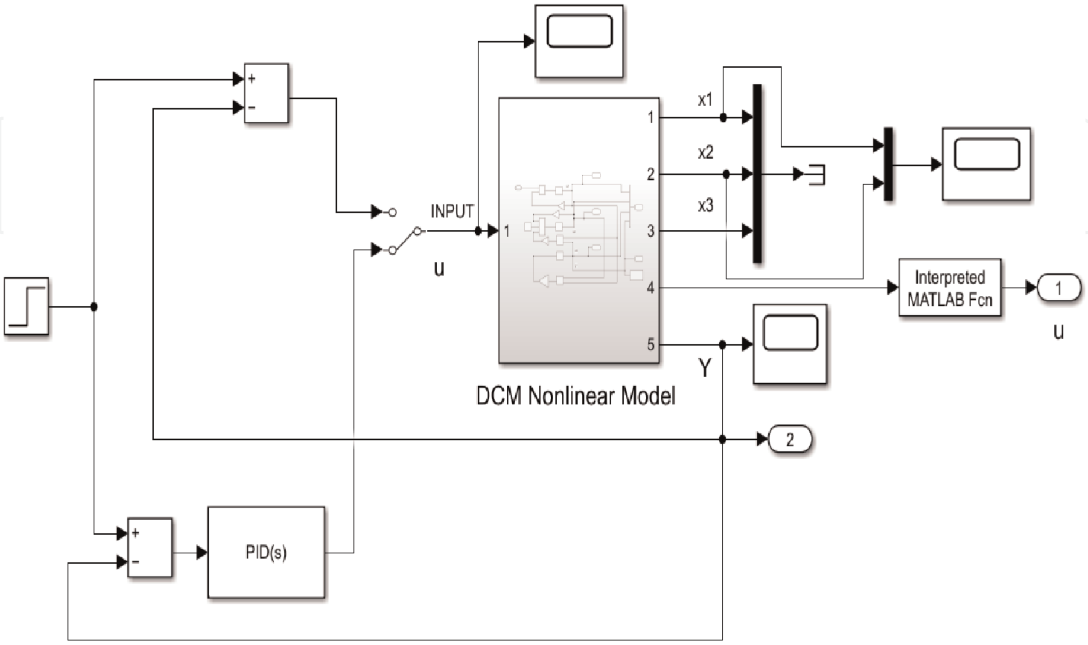


Figure 11.
DCM drive – Simulink nonlinear model, closed-loop control loop and PID control law for reference tracking.

In **Figure 12a** are presented the DCMD field and armature currents and in **Figure 12b** is depictured the DCM speed in rad/s. It is noticeable that the current field remains constant when the DCMD is running and the armature current increases from initial condition value at approximative 1.5 A and then during the steady-state decreases to zero when RPM speed of DCMD is stabilized in steady state to approximate 314 rad/s.

In closed loop for DCMD state feedback linearized whose dynamics is described in Eq. (69) and the new input v is given by

$$v = -K\xi = -k_1\xi_1 - k_2\xi_2 \tag{75}$$

and the values of the coefficients $k_1, k_2$are obtained through a pole placement technique that stabilizes the linearized system (69)

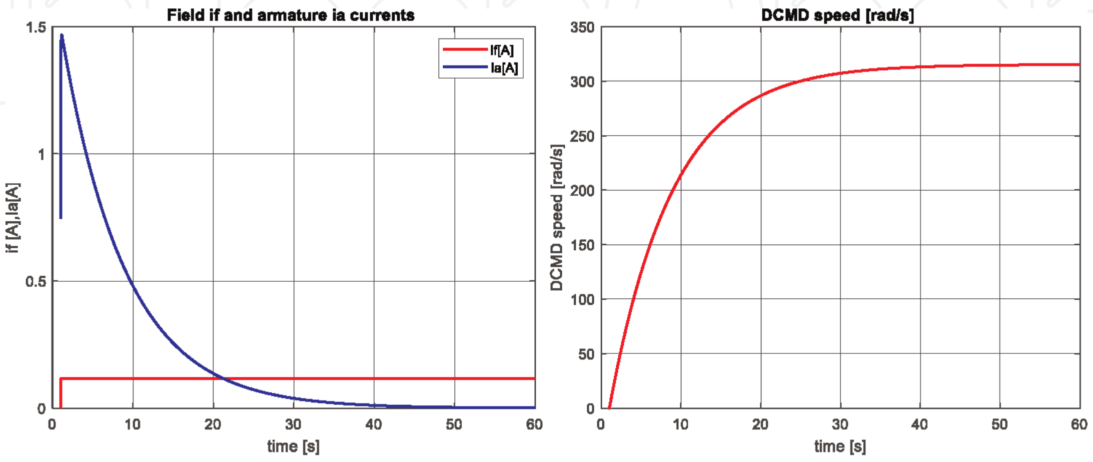
$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = place(A_c, B_c, [-2; -5]), A_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, k_1 = 10, k_2 = 7 \tag{76}$$

The MATLAB simulation results of the DCMD in closed loop canonical form (69) in the new state variables (η, ξ_1, ξ_2) are shown for the evolution of each variable and for actuator effort (v to stabilize these state variables are shown in the **Figure 13a** and **b**.

Figure 13a reveals the evolution of the states during the transient and in steady state, where zeta1 and zeta2 are stabilized to zero, and eta is stabilized to the equilibrium point (close to 1.49, that is$\eta_{ss} = \frac{q}{b}$, thus it validates the zero dynamics result). In **Figure 13b** is presented the actuator effort to stabilize all three variables around the equilibrium point of zero dynamics ($\eta_{ss} = \frac{q}{b} = 1.49, \xi_1 = 0, \xi_2 = 0$.

The MATLAB simulation results for closed loop partial state feedback linearization and state feedback input u given by Eq. (74) are shown in **Figure 14a–c**.

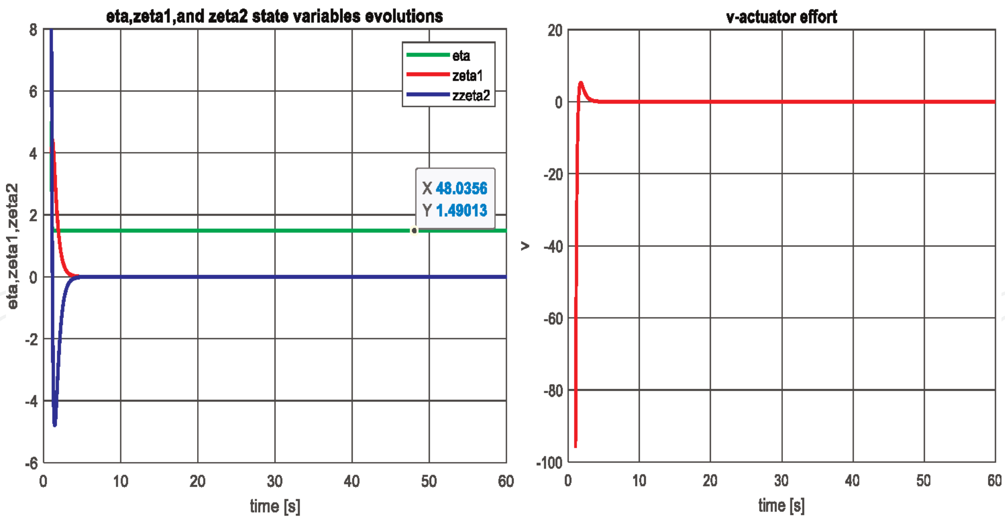
In **Figure 14a** is depicted the DCMD speed stabilized in steady state at equilibrium point of the closed system, similar as for zero dynamics. **Figure 14b** reveals the field and armature currents stabilized at same equilibrium point as for zero dynamics. In **Figure 14c** is visualized the effort of the actuator to keep stabilized the nonlinear



(a)

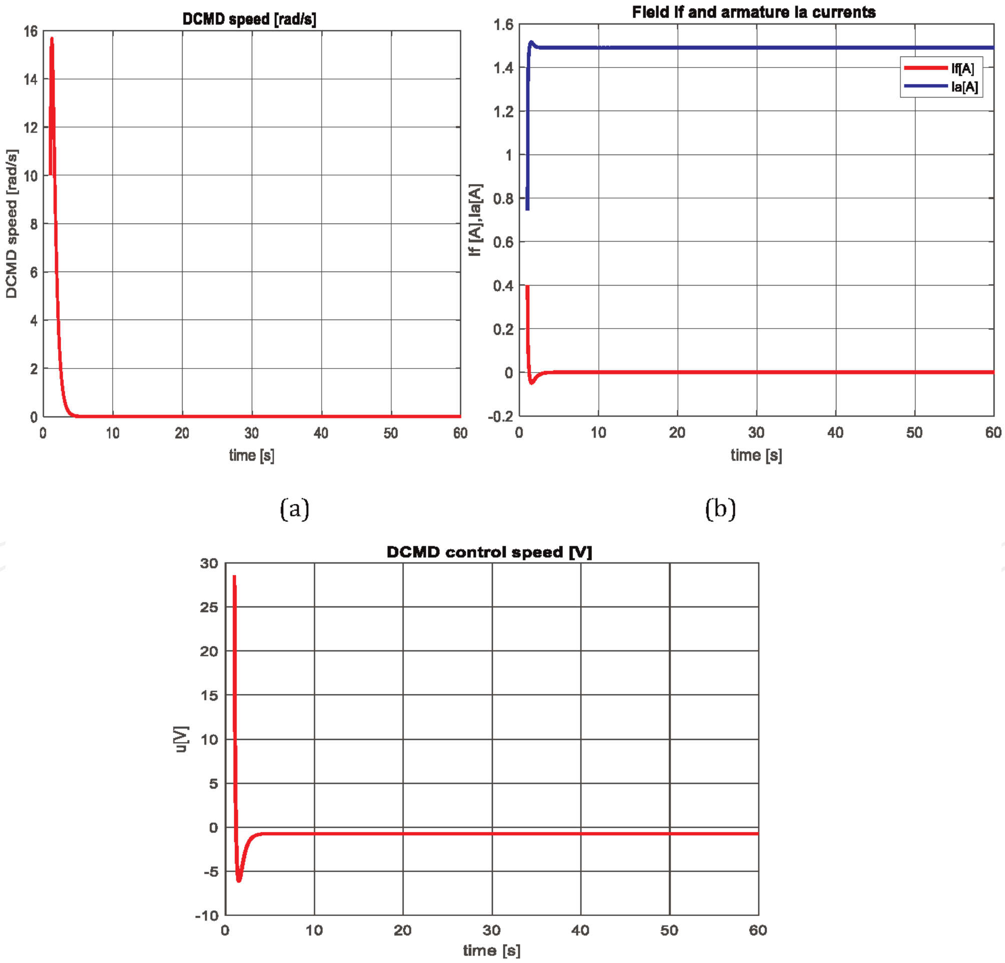
(b)

Figure 12.
MATLAB Simulink simulations results: (a) field and armature currents; (b) DCMD speed [rad/s].



(a) (b)

Figure 13.
MATLAB closed-loop simulations for DCMD model in canonical form: (a) the evolution of the new states (eta, zeta1, zeta2); (b) the actuator effort for stabilization.



(a) (b)

(c)

Figure 14.
MATLAB simulation results in closed-loop by feedback linearization; (a) the DCMD speed stabilization; (b) field and armature currents stabilization at equilibrium point; (c) the actuator effort to stabilize the DCMD in steady-state at equilibrium point.

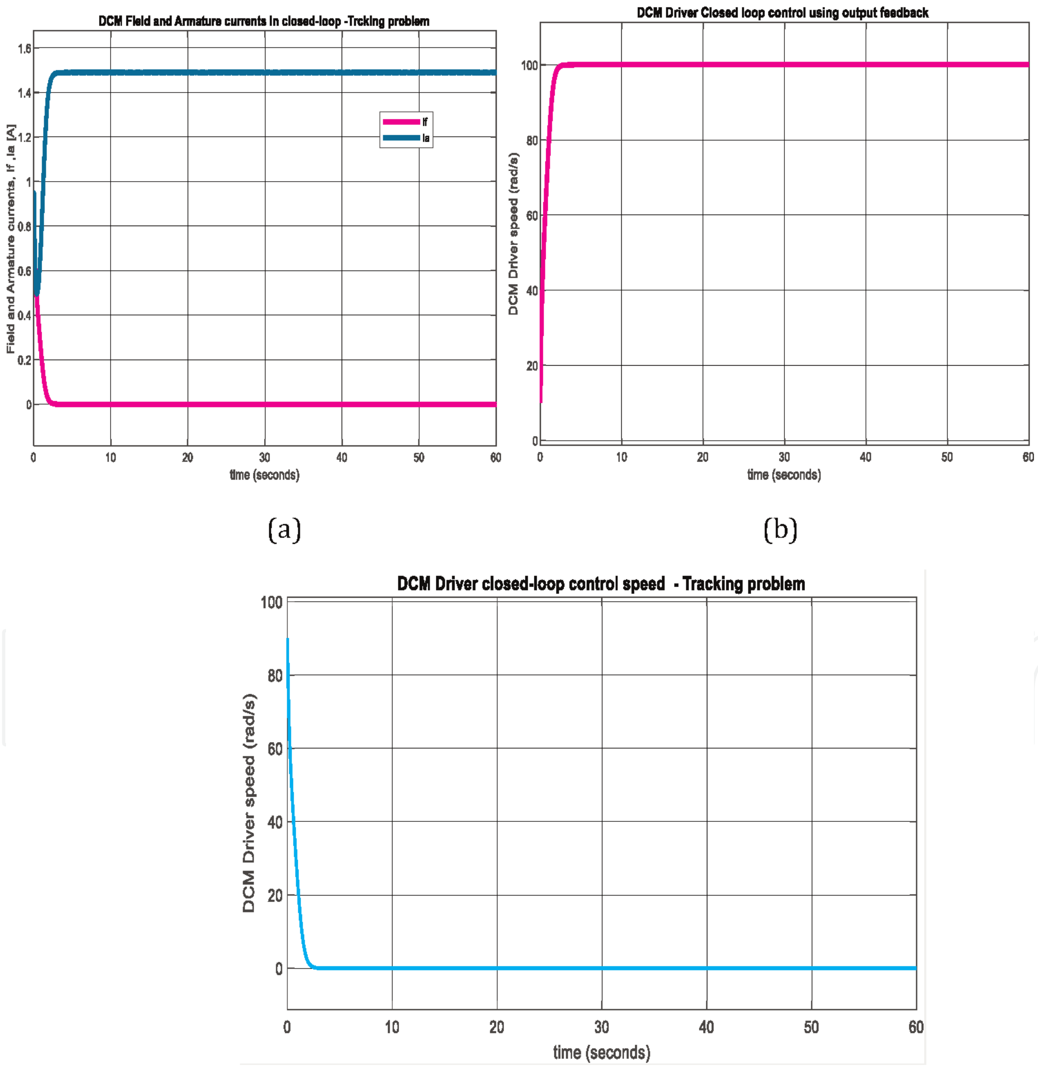
system at an equilibrium point, same as for zero dynamics, therefore the result of Lemma 1 is now validated.

Tracking problem can be solved similar as stabilization but we need to substract from the output the reference input r, such that the error:

$$e(t) = y(t) - r(t) \to 0, \quad \text{when } \; t \to \infty \tag{77}$$

In **Figure 11** it can be seen both options for reference tracking of the DCM drive speed r = 100 rad/s. First option is an integration of the DCM nonlinear model in a unit output feedback closed loop that compares the output measured value to the desired track reference value r = 100 rad/s. The MATLAB Simulink simulations results are depictured in **Figure 15a** and **c**, as follows, field and armature currents in **Figure 15a**, the DCM Driver speed in rad/s revealed in **Figure 15b**, and the actuator effort is visualized in **Figure 15c**.

The second control law is a PID traditional law for which the MATLAB Simulink simulation results are presented in **Figure 16a** and **d**, with same significance as in



(a)

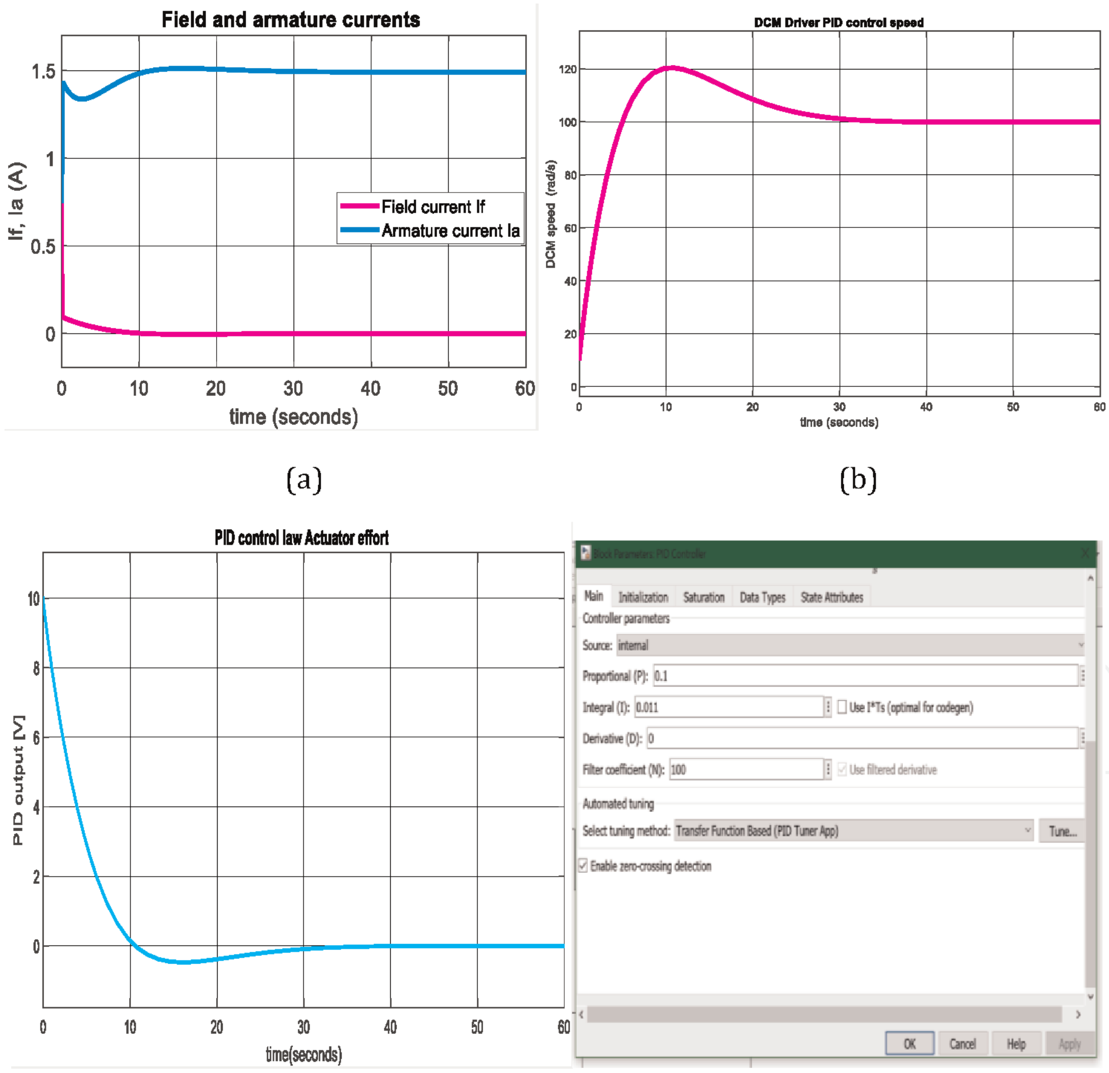
(b)

(c)

Figure 15.
DCM driver closed-loop control speed – MATLAB Simulink simulation results: (a) field and armature currents; (b) DCM driver tracking speed in rad/s; (c) actuator control effort.

Figures 15a and **c**, and the tuning settings values of PID controller parameters KP, K_I and K_D are shown in **Figure 16d**. Comparing the results obtained by using both closed-loop controls it is worth to notice that for first option the DCM speed is an aperiodic fast response (sharply) but with a big actuator effort, instead the PID controller response is slightly slower and smooth with a smaller actuator effort.

The feedback linearization can be performed also by taking advantage of deep training learning techniques to build intelligent neuro controllers. In MATLAB Simulink the Deep Learning Toolbox includes three interesting block sets to develop intelligent neuro controllers, namely NARMA-L2, Model Predictive and Reference Model. The end of this section is an introduction to nonlinear autoregressive moving average (ARMA), similar to SOC and battery terminal voltage, for which an intelligent NARX neural network estimator was developed in the Section 3. The NARMA-L2 smart controller is a brilliant tool for feedback linearization of nonlinear systems, such as the DCM drive described in this section, designed for use in possible EV applications. This intelligent neuro controller is data-driven, so does not require an accurate



Controller parameters

Source: internal

Proportional (P): 0.1

Integral (I): 0.011 □ Use I*Ts (optimal for codegen)

Derivative (D): 0

Filter coefficient (N): 100 ☑ Use filtered derivative

Automated tuning

Select tuning method: Transfer Function Based (PID Tuner App)

☑ Enable zero-crossing detection

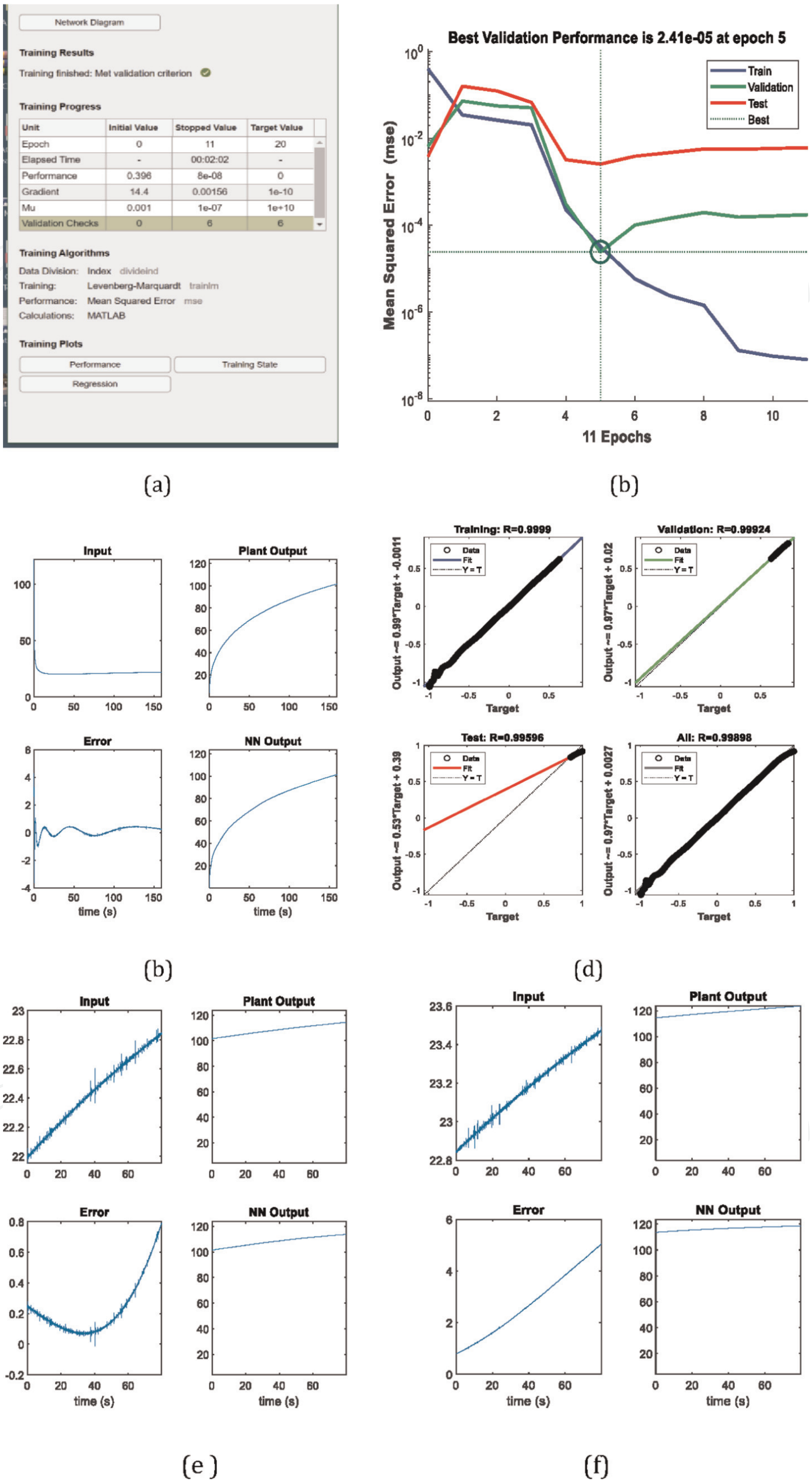
(a)

(b)

(c)

(d)

Figure 16.
MATLAB Simulink simulation results for DCM driver tracking problem using a PID control law: (a) field and armature currents; (b) DCM driver speed (rad/s); (c) PID actuator control law effort; (d) PID parameters tuning values settings.



Network Diagram

Training Results

Training finished: Met validation criterion

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	11	20
Elapsed Time	-	00:02:02	-
Performance	0.396	8e-08	0
Gradient	14.4	0.00156	1e-10
Mu	0.001	1e-07	1e+10
Validation Checks	0	6	6

Training Algorithms

Data Division:	Index	divideind
Training:	Levenberg-Marquardt	trainlm
Performance:	Mean Squared Error	mse
Calculations:	MATLAB	

Training Plots

Performance
Training State
Regression

(a)

(b)

(b)

(d)

(e)

(f)

Figure 17.
MATLAB Simulink simulation results during learning step of DCM drive nonlinear model (feedback linearization): (a) process training results; (b) the best validation performance reached at epoch 5; (c)training phase process; (d) regression performance; (e) validation phase process; (f) testing phase process.

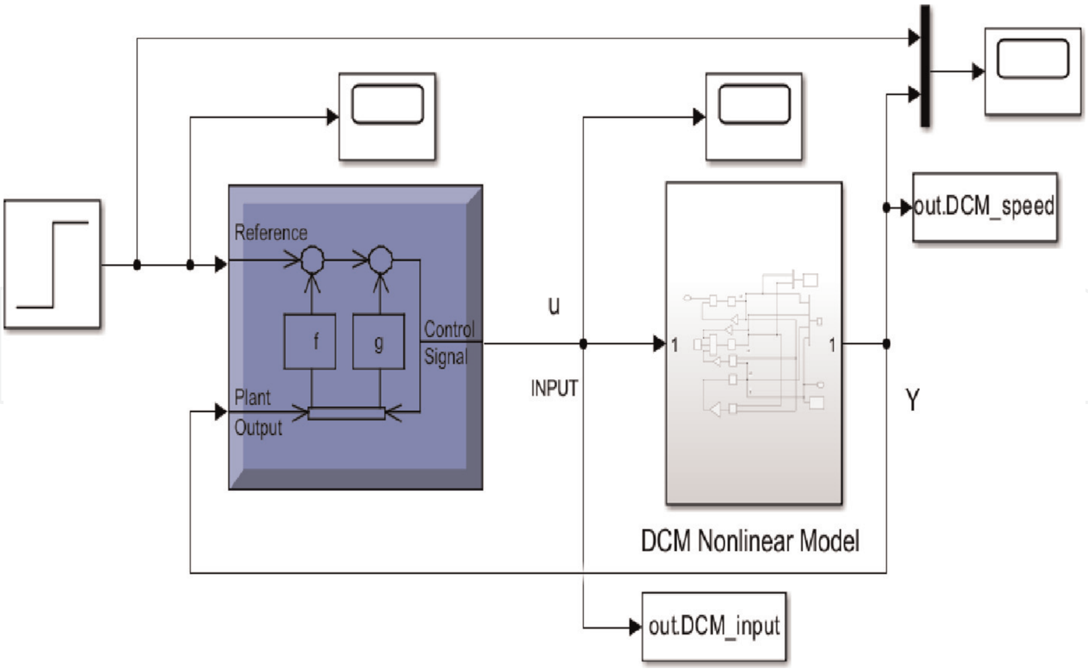


Figure 18.
MATLAB Simulink model of the DCM drive speed control in closed-loop using a NARMA-L2 controller.

model to perform well, and therefore the impact of some modeling imperfections will be significantly attenuated. To design such a controller, two steps must be followed. In the first step, the Deep learning Simulink block is trained separately to learn the nonlinear model, which is to be feedback linearized. In the second step, the trained neural network will be connected to the NARMA-L2 neuro controller to perform a reference tracking problem, similar to nonlinear feedback linearization. So, all the nonlinearities will be canceled in the first step, and in the second step, the controller solve a reference tracking problem. The main reason for finishing this research is to open some exciting research directions in developing intelligent neuro controllers that deal with all types of nonlinearities. Only the first step is solved in this research, i.e., the Simulink neuro controller block learns the nonlinear model of the DCM drive to be connected to the neuro controller for performing the reference tracking problem. The MATLAB Simulink simulation results are presented in **Figure 17a–f**, with the following meaning, training process results in **Figure 17a**, the best validation performance during the training process in **Figure 17b**, training data phase in **Figure 17c**, regression performance in all the stages (training, validation and test), is shown the in **Figure 17d**, validation data phase in **Figure 17e**, and test data phase in **Figure 17f**.

The Simulink model block of NARMA-L2 controller is illustrated in **Figure 18**.

5. Conclusion

The following results are worth highlighting among the authors' main contributions to this book chapter.

- 2RC ECM Li-ion battery model selection, design and implementation, a rigorous performance analysis.

- EKF SOC estimation and terminal voltage prediction design, implementation, and performance analysis in terms of SOC accuracy and robustness.

- An intelligent advanced NARX neural network deep learning SOC estimator, terminal voltage predictor, and curve fitting of nonlinear Li-ion battery characteristics OCV = f(SOC).

- A DCM drive nonlinear model selection with a detailed description of nonlinear linearization techniques such as full and partial state feedback, input-output linearization and advanced intelligent NARMA-L2 neuro-controller. The main issues/challenges problems were revealed, focussing on possible singularities, internal dynamics, zero dynamics, closed loop stabilization and reference tracking problems to be solved.

- Closed-loop DCM drive speed control techniques, such as full state feedback linearization, the closed-loop output unit negative feedback, PID controller, and performance comparison.

All the algorithms' implementation and extensive simulations were conducted on MATLAB Simulink R2023b software platform, a valuable and the strongest implementation tool from the software engineering market, in a beautiful and user-friendly environment. The MATLAB Simulink simulation results proved that the advanced intelligent estimators outperform the traditional ones. For future work, the research continues in the direction of design and implementation of new advanced intelligent learning techniques and neuro controllers such as NARMA-L2, Model Predictive and Model Reference neuro controllers. The energy management performance of the Li-ion battery connection to the DCM drives will be further investigated.

Conflict of interest

The authors declare no conflict of interest.

Author details

Roxana-Elena Tudoroiu[1], Mohammed Zaheeruddin[2], Nicolae Tudoroiu[3]*,
Sorin Mihai Radu[1] and Hana Chammas[3]

1 University of Petrosani, Petrosani, Romania

2 Concordia University from Montreal, Montreal, Canada

3 John Abbott College, Sainte-Anne-de-Bellevue, Canada

*Address all correspondence to: ntudoroiu@gmail.com

References

[1] Xia B, Zheng W, Zhang R, Lao Z, Sun Z. Mint: A novel observer for Lithium-ion battery state of charge estimation in electric vehicles based on a second-order equivalent circuit model. Énergies. 2017;**10**(8):1150. DOI: 10.3390/en10081150. Available from: http://www.mdpi.com/1996-1073/10/8/1150/htm

[2] Young K, Wang C, Wang LY, Strunz K. Electric vehicle battery technologies–chapter 2. In: Garcia-Valle R, Lopes JAP, editors. Electric Vehicle Integration into Modern Power Networks. 1st, 9 and 32 ed. New-York, USA: Springer Link: Springer-Verlag; 2013. pp. 15-26. DOI: 10.1007/978-1-4614-0134-6

[3] Farag M. Lithium-ion batteries. In: Modeling and State of Charge Estimation. Ontario, Canada: McMaster University of Hamilton; 2013. p. 169

[4] Tudoroiu RE, Zaheeruddin M, Radu SM, Tudoroiu N, Martinez LR, Prieto MD, editors. New trends in electrical vehicle powertrains. New Trends Electrical Vehicles in Powertrains. 2019;**4**:55-81. DOI: 10.5772/intechopen.76230.ch4

[5] Plett GL. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification. Power Sources. 2004;**134**:262-276. DOI: 10.1016/j.jpowsour.2004.02.032

[6] Zhang R, Xia B, Li B, Cao L, Lai Y, Zheng W, et al. State of the art of Li-ion battery SOC estimation for electrical vehicles. Energies. 2018;**11**:1820

[7] Zhongqiang W, Shang M, Shen D, Qi S. SOC estimation for batteries using MS-AUKF and neural network. Journal of Renewable and Sustainable Energy. 2019;**11**:024103. DOI: 10.1063/1.5064479

[8] Ali MU, Zafar A, Nengroo SH, Hussain S, Junaid Alvi M, Kim H-J. Towards a smarter battery management system for electric vehicle applications: A critical review of Lithium-ion battery state of charge estimation. Energies. 2019;**12**:446. DOI: 10.3390/en12030446

[9] Alyakhni A, Boulon L, Vinassa J-M, Briat O. A comprehensive review on energy management strategies for electric vehicles considering degradation using aging models. IEEE Access. 2021;**9**: 143922-143940. DOI: 10.1109/ACCESS.2021.3120563

[10] Ghaeminezhad N, Monfared M. Charging control strategies for lithium-ion battery packs: Review and recent developments. IET Power Electronics. 2022;**15**:349-367. DOI: 10.1049/pel2.12219

[11] Nuroldayeva G, Serik Y, Adair D, Uzakbaiuly B, Bakenov Z. State of health estimation methods for Lithium-ion batteries. International Journal of Energy Research. 2023;**2023**:21. DOI: 10.1155/2023/4297545

[12] Pang B, Chen L, Dong Z. Data-driven degradation Modeling and SOH prediction of Li-ion batteries. Energies. 2022;**15**:5580. DOI: 10.3390/en15155580

[13] Camargos M, Angelov P. State of health and lifetime prediction of lithium-ion batteries using self-learning incremental models. In: Proceedings of the 7th European Conference of the Prognostics and Health Management Society, Turin, Italy. 2022

[14] Li M, Liu F, Han B, et al. Research on temperature control performance of

battery thermal management system composited with multi-channel parallel liquid cooling and air cooling. Ionics. 2021;**27**:2685-2695. DOI: 10.1007/s11581-021-04033-w

[15] Yuan S, Hongjie W, Yin C. State of charge estimation using the extended Kalman filter for battery management systems based on the ARX battery model. MDPI, Energies Journal. 2013;**6**: 444-470. DOI: 10.3390/en6010444

[16] Rzepka B, Bischof S, Blank T. Implementing an extended Kalman filter for SoC estimation of a Li-ion battery with hysteresis: A step-by-step guide. Energies. 2021;**14**:3733. DOI: 10.3390/en14133733

[17] Cheng M, Fang F, Navon IM, Pain C. Ensemble Kalman filter for GAN-ConvLSTM based long lead-time forecasting. Journal of Computational Science. 2023;**69**:102024. DOI: 10.1016/j.jocs.2023.102024

[18] Askari I, Haile MA, Tu X, Fang H. Implicit particle filtering via a Bank of nonlinear Kalman filters. Systems and Control. 2022;**145**:110469. DOI: 10.1016/j.automatica.2022.110469

[19] Namdari A, Samani MA, Durrani TS. Lithium-ion battery prognostics through reinforcement learning based on entropy measures. Algorithms. 2022;**15**:393. DOI: 10.3390/a15110393

[20] Zraibi B, Mansouri M, Okar C. Comparing single and hybrid methods of deep learning for remaining useful life prediction of lithium-ion batteries. In: The 4th International Conference of Computer Science and Renewable Energies (ICCSRE'2021), E3S Web of Conferences, Agadir, Morocco. Vol. 297. July 2021. p. 8. DOI: 10.1051/e3sconf/202129701043

[21] Wang S, Ren P, Takyi-Aninakwa P, Jin S, Fernandez C. A critical review of improved deep convolutional neural network for multi-timescale state prediction of lithium-ion batteries. Energies. 2022;**15**:5053. DOI: 10.3390/en15145053

[22] Cui Z, Dai J, Sun J, Li D, Wang L, Wang K. Hybrid methods using neural network and Kalman filter for the state of charge estimation of lithium-ion battery. Hindawi Mathematical Problems in Engineering. 2022;**2022**:11. DOI: 10.1155/2022/9616124

[23] MathWorks MATLAB Version R2021b on-line Documentation, Neuro-Adaptive Learning and ANFIS. Available from: https://www.mathworks.com/help/fuzzy/neuro-adaptive-learning-and-anfis.html

[24] Konsoulas IS. Adaptive neuro-fuzzy inference systems (ANFIS) library for Simulink MATLAB Central File Exchange. Available from: https://www.mathworks.com/matlabcentral/fileexchange/36098-adaptive-neuro-fuzzy-inference-systems-anfis-library-for-simulink [Accessed: January 19, 2022]

[25] MathWorks MATLAB Version R2021b on-line Documentation, Neuro-Fuzzy Designer. Available from: https://www.mathworks.com/help/fuzzy/neurofuzzydesigner-app.html

[26] Bellali B, Hazzab A, Bousserhane IK, Lefebvre D. A decoupled parameters estimators for in nonlinear systems fault diagnosis by ANFIS. International Journal of Electrical and Computer Engineering (IJECE). 2012;**2**(2):166-174. DOI: 10.11591/ijece.v2i2.221

[27] MathWorks MATLAB Version R2023a on-line Documentation. Shallow neural network time-series prediction and modeling. 2023. https://www.math

works.com/help/deeplearning/gs/neural-network-time-series-prediction-and-modeling.html [Accessed May 23, 2023]

[28] MathWorks MATLAB Version R2023a on-line Documentation. Multistep Neural Network Prediction. 2023. https://www.mathworks.com/help/deeplearning/ug/multistep-neural-network-prediction.html [Accessed May 23, 2023]

[29] Tudoroiu N, Zaheeruddin M, Tudoroiu R-E, Radu MS, Chammas H. Intelligent deep learning estimators of a Lithium-ion battery state of charge design and MATLAB implementation-a case study. Vehicles. 2023;**5**:535-564. DOI: 10.3390/vehicles5020030

[30] Tudoroiu N, Zaheeruddin M, Tudoroiu R-E, Radu MS, Chammas H. Investigations on using intelligent learning techniques for anomaly detection and diagnosis in sensors signals in Li-ion battery-case study. Inventions. 2023;**8**:74. DOI: doi.org/10.3390/inventions8030074

[31] Tiezhou W, Wang M, Xiao Q, Wang X. The SOC estimation of power Li-ion battery based on ANFIS model. Smart Grid and Renewable Energy. 2012;**3**:51-55. DOI: 10.4236/sgre.2012.31007

[32] Jamlouie MHA. Accuracy Improvement of SOC Estimation IN LITHIUM-Ion Batteries by ANFIS Vs Ann MODELING of Nonlinear Cell Characteristics. Ontario, Canada: Ryerson University; 2018

[33] Ehsani M, Gao Y, Gay SE, Emadi A. Modern electric, hybrid electric, and fuel cell vehicles-fundamentals, theory, and design. In: Power Electronics and Applications Series, Rashid M.H., 2005, 1–419, University of West Florida, Boca Raton, FL. USA

[34] Khalil KH. Nonlinear Systems. 3rd ed. New Jersey, USA: Prentice-Hall; 2002

[35] Willson SS, Mullhaupt P, Bonvin D. Avoiding feedback-linearization singularity using a quotient method – The field-controlled DC motor case. In: 2012 American Control Conference; 27–29 June 2012. Montreal, Canada: IEEE; 2012. pp. 1-7. DOI: 10.1109/ACC.2012.6315095

[36] Philip DO. Feedback linearization of DC motors. IEEE Transactions on Industrial Electronics. 1991;**8**(6)

[37] MathWorks MATLAB version R2023a. Design NARMA-L2 Neural Controller in Simulink. 2023. Available from: https://www.mathworks.com/help/deeplearning/ug/design-narma-l2-neural-controller-in-simulink.html [Accessed: May 26, 2023]

[38] Konsoulas I. Recurrent fuzzy neural network (RFNN) library for Simulink MATLAB Central File Exchange. 2022. Available from: https://www.mathworks.com/matlabcentral/fileexchange/43021-recurrent-fuzzy-neural-network-rfnn-library-for-simulink [Accessed: January 19, 2022]