38th International Electric Vehicle Symposium and Exhibition (EVS38) Göteborg, Sweden, June 15-18, 2025

Eco-features Optimization using Digital Twins of EV Powertrain

Mohammed Mahedi Hasan[1,2,*], Mohamed El Baghdadi[1,2], Thomas Geury[1,2], Omar Hegazy[1,2]

[1] *MOBI-EPOWERS RG, ETEC Department, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Ixelles, Belgium*

[2] *Flanders Make, Gaston Geenslaan 8, 3001 Heverlee, Belgium*

** Corresponding author, email: mohammed.mahedi.hasan@vub.be*

Executive Summary

In this research, the DTs of the EV powertrain components were designed using a CFNN and then trained using a combination of Levenberg–Marquardt (LM) and scaled conjugate gradient (SCG) training algorithms. The powertrain components are represented by the actual hardware in a HiLS or using Hi-Fi multi-physics finite element models (FEM). The trained DTs are then used as a component in the improved simple optimization (iSOPT) algorithm, and supplied with the input scenario to evaluate various reference torque and temperature setpoints that will minimize the overall energy consumption of the bus. The optimization was repeated for numerous random scenarios, and a data-driven model is generated (in MATLAB) using system identification, which links the scenario conditions to the optimized output.

Keywords: electric vehicle powertrain, eco-features, optimization, digital twins, machine learning

1 Introduction

Electric buses have become a popular choice for cities looking to modernize their public transport systems towards zero emissions and carbon neutrality. Advances in battery technology, combined with growing environmental awareness and regulatory pressures, have accelerated the adoption of these vehicles. Unlike their diesel counterparts, electric buses produce zero tailpipe emissions, making them an attractive choice for reducing urban air pollution (NOx, CO, PM2.5), and greenhouse gas (GHG) emissions [1]. Stricter emissions regulations, improvements in battery and charging technology providing longer range and faster charging times, subsidies and incentives from governments and international bodies towards electrification of public transport research and development and increasing awareness of climate change and air quality issues among the public are some of the key driving factors behind the increased demand for electric buses for public transport. In addition to the reduction in GHG and improvement of air quality, electric buses are also cost efficient over their operational lifetime, compared to ICE-based buses, and contribute to reduction in vehicle noise in urban roads. However, there are challenges to adopting electric buses on a wider scale, including the need to invest in a comprehensive, but costly, charging infrastructure, the need to ensure that the electric grid can handle the increased load due to charging of large fleets of electric buses, and the need to ensure that the energy storage system (ESS) in the bus has sufficient capacity to handle the energy consumption requirements of the bus.

While battery and charging technology has advanced, issues such as limited range and long charging still pose challenges. Research and development on new battery chemistry, module and pack optimization are crucial to improving energy density, reducing costs, and increasing the durability of batteries [2, 3]. Furthermore, innovative technologies including placing solar panels on the vehicle roof further improve the driving range by reducing the energy load placed on the battery during daytime and charging the battery when the vehicle is not in motion [4, 5]. However, reducing the power load on the battery is a key technique to increase the driving range and the lifetime reliability of the battery; it also has additional benefits external to the vehicle, including reduction in charging infrastructure, reduction in the charging duration, and decreased load in the electric grid. Vehicle energy consumption can be reduced using various energy saving "eco" techniques, including eco-routing, eco-driving, and eco-comfort [6, 7]. Eco-driving offers the greatest reduction in vehicular energy requirements [6] compared to the other energy management features when the average route speed is greater. On the other hand, eco-comfort offers substantial energy savings for larger buses, such as articulated and double articulated buses, during adverse weather conditions, like very high or very low ambient temperatures [6].

1.1 Eco-driving

Eco-driving is an energy management strategy that optimizes the operation of the traction system of the electric vehicle (EV) powertrain to reduce the vehicle's traction energy expenditure. Traditionally eco-driving followed a rules-based heuristic approach that modified the original driving velocity profile to an eco-friendlier profile having ramped acceleration, lower maximum acceleration, and lower average vehicle speeds. The lower speeds helped reduce energy loss due to aerodynamic drag, while the lower acceleration improved tractive and regenerative efficiency. More recently, the application of optimization techniques offered greater energy reduction compared to a purely heuristic approach [8]; in the optimization approach the emphasis is placed on finding the optimal torque reference profile that can be commanded to the electric motor, which would overall minimize the traction energy requirements of a given scenario. Optimization is effective because it considers the three major factors that affect the traction energy consumption, including the driving velocity profile, the vehicle's load profile, and the route gradient profile.

1.2 Eco-comfort

Eco-comfort is a thermal management strategy that optimizes the climate control system of the vehicle to reduce the vehicle's auxiliary energy consumption [9]; it relies on two pillars, the modification of the cabin temperature setpoints to minimize the energy required for temperature regulation, and thermal preconditioning of the various powertrain components, including the vehicle's cabin and the energy storage system. Preconditioning refers to the strategy of utilizing those moments when the vehicle is connected to the charger to undergo the energy intensive process of temperature tracking from ambient conditions to the desired setpoints. A thermally preconditioned vehicle needs only to expend the minimal energy required for temperature regulation when the vehicle operates under battery power. The dynamic cabin temperature setpoint, estimated through optimization, aims to find a delicate balance between minimizing the energy expenditure of the heat pump for temperature regulation, as well as the discomfort level felt by the passengers. The temperature setpoints depend on the ambient temperature and the passenger quantity inside the bus.

1.3 Problem statement

The optimization process is time consuming; thus, it is not possible to deploy in real time to the vehicle control unit (VCU). A practical approach is to complete the optimization as an offline process for a given route, based on different driving scenarios, passenger load profiles, and climate conditions, along with the gradient profile of the route, which does not change. A neural network (NN) can then be trained using optimized torque and temperature references (output) to different scenario conditions (input). The trained NN once deployed offers real-time optimized torque and temperature references to dynamically changing input conditions. Another problem with the optimization process is that it generally uses low-fidelity (Lo-Fi) models to increase simulation speed and hence decrease the time required for optimization; however, Lo-Fi models have limits to the accuracy of the results they can provide. Methods that provide highly accurate results, including hardware-in-a-loop simulation (HiLS) using the actual powertrain or simulation using a high-fidelity (Hi-Fi) multi-physics-based finite element model (FEM) of the powertrain, are both prohibitively time-consuming and computationally expensive, and not suitable for real-time optimization. To solve this conundrum, an approach is required that not only gives accurate simulation results but is also faster to simulate for optimization, and the digital twin (DT) framework allows for the best of both worlds.

Section 2 provides an overview of NN and other data driven approaches that allow for the training and tuning of the DTs. Section 3 provides the methodology used to set up the DT-based rapid optimization tool. Section 4 provides the results, and section 5 concludes this article.

2 Digital Twin Framework

A DT of a physical system can be trained in one of three ways: 1) a white-box modeling approach such as using an observer model, extended Kalman filter (EKF), or model predictive control (MPC), 2) a grey-box modeling approach such as model parameter tuning via optimization, and 3) a black-box modeling approach such as designing and training a NN representing the physical system [10]. Model parameters tuning is an example of grey-box modeling, since the user has knowledge of the parameter values that need to be tuned, but not the physics of the system. On the other hand, the control approach is an example of white-box modeling since accurate knowledge of complete system (i.e., the "plant"), including the physics of the system and the parameter values, is necessary to properly tune the virtual model. Finally, designing and training an NN is an example of black-box modeling since the trained NN matrix does not give the user any insight into the physical equations that represent the system, nor the values of the parametric variables in the equation. While it is advantageous to utilize white-box modeling, as the user has full control over the model allowing the user to develop control strategies using the model, add new functionalities into the model, and adapt and customize the model to various applications; however, the high level of fidelity required for white-box modeling raises the computational cost to simulate the model, both in terms of hardware resources and time needed. There is a reason Lo-Fi models have been utilized in this research, very fast simulation speeds are essential for an iterative process such as optimization, and with Hi-Fi models, it becomes prohibitively time consuming. Comparatively, the training of NN was a faster process, and as a bonus, GPU resources were leveraged to speed up the NN training significantly. As real-time training was required for the DT, the NN approach was determined to be the best one.

Machine learning (ML), which is a branch of artificial intelligence (AI), is the process of training NNs using the measurement inputs to and the outputs from a real system, so behave as the real physical system over time; at its most basic ML uses algorithms to find patterns and then apply the patterns moving forward. To recognize such patterns, the ML algorithm first needs to be trained using a known pattern and then applies the trained algorithm to recognize patterns of similar nature but for which it has not been trained. The main goals of ML are to classify data based on models that have been developed and then make predictions regarding some future outcome based on such models. According to [11], there are a sequence of steps that the ML algorithm needs to follow: collection of raw data, data preparation to a format that the ML algorithm can understand, choosing a suitable NN architecture, model optimization (i.e., the actual training process), model evaluation using new dataset, and model deployment where the trained model can now make accurate predictions. Furthermore, ML ensures that the NN can learn and adapt to new patterns over time.

2.1 Proposed neural network architecture

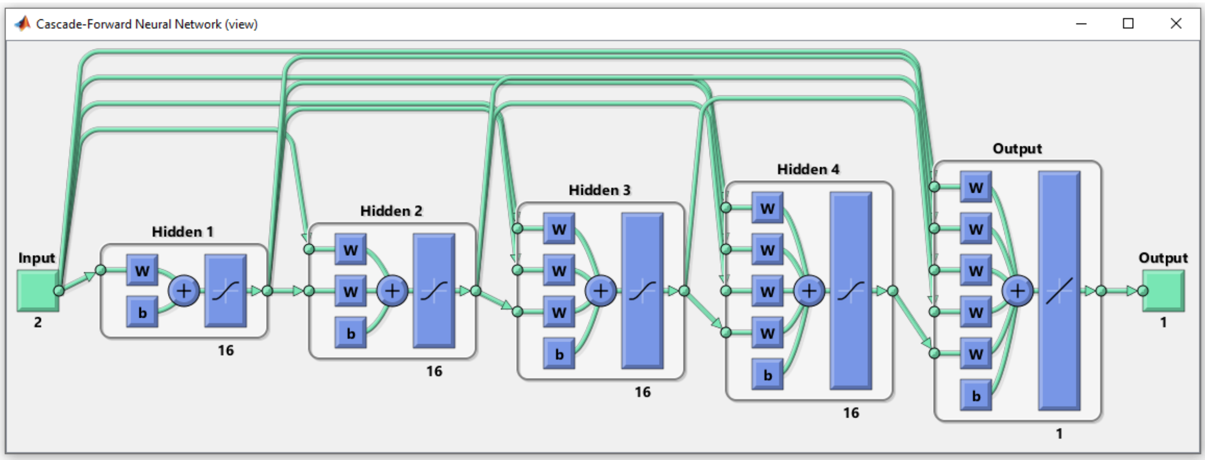

Figure 1: Design of a Cascaded Feedforward Neural Network, with 4 hidden layers, and 16 neurons per layer

NNs are computational models that mimic the neurons in the human nervous system in how they process information [12]. NNs consists of layers of nodes that transform input data into meaningful outputs through a series of mathematical operations. Each node (or neuron) can affect multiple other nodes later in the chain, while at the same time each node itself can be affected by multiple nodes earlier in the chain. The layers itself act as gateways that directs signals (information) through specific 'learned' pathways to give the required output. There are various categories of NNs, designed according to the type of data that the NN is required to handle, e.g., densely connected NNs such as feedforward neural networks (FNN) are used to process tabular data, convolutional neural networks (CNNs) are used to process spatial data such as images, and recurrent neural networks (RNNs) and their different variants, including LSTM and GRU, are used for their ability to preserve time series information (i.e., memory) so they are used to process sequential data. The NN topology that has been utilized in this research is the cascaded feedforward neural network (CFNN), where the output of each layer is cascaded to all subsequent layers coming after it; thus, the complexity of a layer increases linearly down the chain from input to output as shown in Figure 1. Equation (1) represents the output of each hidden layer using a symmetric tan sigmoid function, and (2) represents the output of the output layer using a linear transfer function.

$$y_i = A_i * \left(\frac{2}{1+e^{-2*\left(B_i * X_i + b_{1\_i}\right)}} - 1\right) + b_{2\_i} \text{, where i = 1…4} \tag{1}$$

$$y_o = A_o * \left(B_o * X_o + b_{1\_o}\right) + b_{2\_o} \tag{2}$$

Where A is the weight for each neuron in a hidden layer (1 x 16 matrix), B is the weight for each input coming to the hidden layer (16 x (2 + layer number - 1) matrix), X is the number of inputs input to a layer ((2 + layer number -1) x 1 matrix), b1 is the bias of each neuron in the layer (16 x 1 matrix), and b2 is the output bias of the layer itself (scalar value), the subscript 'i' denotes a hidden layer, while the subscript 'o' denotes the output layer. CFNNs are examples of densely connected NNs, composed of an input layer, multiple hidden layers, and an output layer, and the nodes in each layer are maximally connected to the nodes of the neighboring layers. The cascading nature of CFNNs allows the NN to detect changes in the sequential data, as new data which are cascaded forward from the input layer can be compared with processed older data in the hidden layers, thus keeping track of some form of sequential memory. The length of the memory that can be tracked by CFNN depends on the number of hidden layers used in the network. This ability is important since CFNNs are lightweight networks that allow time series data to be processed rapidly, but without the computational costs associated with RNNs.

2.2 Proposed training strategy

Two training strategies are used to train the NN. The first is using Levenberg-Marquardt (LM) algorithm, which is uniquely suited to function fitting approximation problems (i.e., problems that can be solved using regression), because it is very fast to converge to a solution (i.e., using the minimum number of iterations or 'epochs') and it results in the most accurate approximation compared to other training algorithms for function approximation problems. LM can be used to solve least squares curve fitting for non-linear least squares problems, using gradient descent method or the Gauss-Newton method; the LM algorithm adaptively switches between these two methods starting with gradient descent to achieve iterative steps with lesser computational cost initially and then switching over to GNA for better accuracy towards the end, at the cost of a much higher computation in each iterative step [13]. The disadvantages of the LM algorithm are that the speed advantage disappears for large NNs, and the algorithm requires a great deal of memory and processing power; thus, making the LM algorithm unsuitable as a real-time implementation. A second major issue is the inability to use GPU cores due to the Jacobian training required in the LM algorithm; thus, the speed boost achievable from hyper-parallel processing possible inside a GPU is negated. Finally, LM suffers from the local minima problem, when there is a non-convex least square solution space with multiple local minima; thus, a NN needs to be trained multiple times with the LM algorithm with different initial configurations of the NN weights.

To overcome the limitations of the LM algorithm, the scaled conjugate gradient (SCG) was utilized as the second training strategy. SCG is one of many conjugate gradient algorithms that are used to minimize non-linear functions. It was developed to avoid the high computation cost of Newton's method (as it does not need to calculate the Hessian matrix and its inverse) and accelerate the convergence rate of steepest descent (as the theoretical number of iterations needed to reach the closest approximation to a solution is determined by the number of eigenvalues of a matrix, which is 'n' for a n-by-n matrix) [14]. The calculation of the conjugate direction, used to point to the next location of the solution, involves mainly calculation of scalar values through mainly matrix multiplication operations. There are no matrix divisions or matrix inverse operations needed, which makes the computation load increase linearly in proportion to the size of the matrix. Finally, unlike the LM algorithm, the SCG algorithm can make full use of the GPU resources of a PC.

3 Proposed Optimization Methodology

DTs are digital models of the intended real-world physical product, system, or process that serve as a digital counterpart of it for purposes such as simulation, integration, testing, monitoring, and maintenance [15]. The underlying framework of the DTs used in this research are NNs trained using ML strategies to offer a compact simulation model that are not only lightweight and capable of rapid simulation but also produces output that closely corresponds to reality. NNs representing the DTs of the EV powertrain can be trained either from the measurement output from the HiLS of a powertrain component, or from the simulation output of a Hi-Fi multi-physics-based FEM of powertrain component. Optimization simulation on the trained NN can then be run at a faster speed. This combined method decreases the time needed from prototype simulations to deployment and improves the optimization accuracy in estimating the EV energy requirements resembling that of the actual vehicle. Figure 2 shows the proposed strategy to develop the optimization framework using DTs of physical systems to provide realistic and real-time estimates of vehicular energy requirements to determine the optimal control setpoints for torque and cabin reference temperature that minimizes traction and auxiliary energy use.

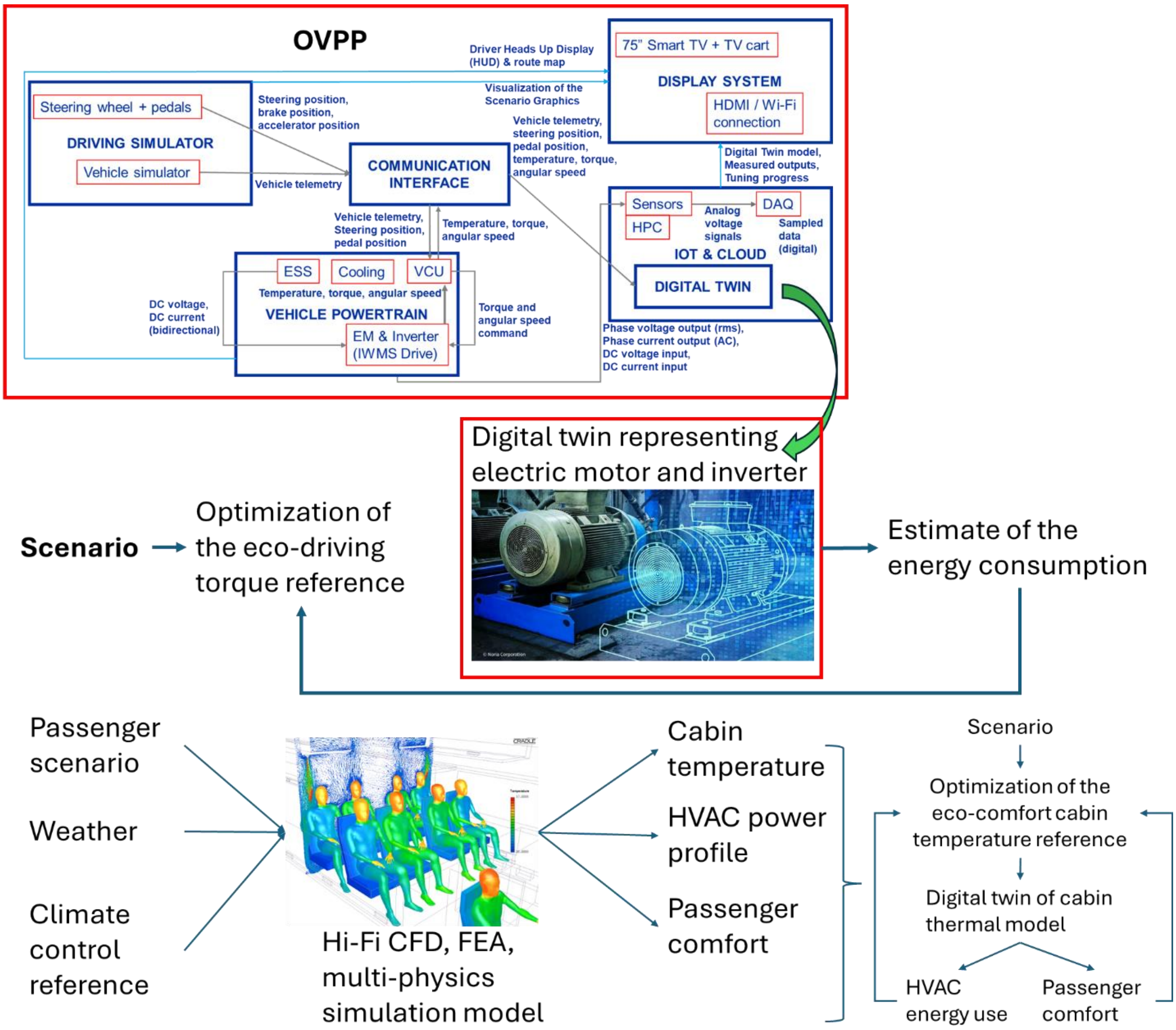

Figure 2: The proposed optimization framework to provide fast but realistic usage for vehicle energy usage

3.1 HiLS setup of the traction system

The vehicle powertrain provides the testbed framework where any component of the EV powertrain can be tested utilizing HiLS. The device under test (DuT) can either be individual powertrain components, including converters, inverters, EMs, and batteries, or they can be subsystems such as the electric drive system consisting of the inverter, EM, and the associated cooling system. For this research, the part of the EV powertrain that was the DuT was the in-wheel motor system (IWMS) from Elaphe. The system consists of the high-power inverter and an in-wheel EM with a rated continuous power of 50kW, and peak power of 75kW. The motor has a rated speed of 1200 rpm (125 rad/s), a maximum speed of 1500 rpm (157 rad/s), a maximum continuous torque of 400 Nm, and a peak torque of 700 Nm. The inverter can operate under a range of DC-link voltages from 250V to 375V and can handle currents up to 200A in both directions. In addition, the IWMS features a liquid cooling system that is used to keep the inverter MOSFETs, and the EM coils cool. The EM and inverter are rated to operate up to 150^{o}C; however, for safety purposes, the control logic in the vehicle control unit (VCU) is programmed to not allow the motor coil temperature to exceed 70^{o}C and the inverter temperature to exceed 85^{o}C. The inverter can directly be commanded by the VCU with the desired motor speed and torque at a rate of 500Hz via CAN communication; the inverter supplies the VCU with motor speed and inverter output current (I_Q) measurements at a rate of 500Hz, the DC-link current and voltage measurement at a rate of 100Hz, and the motor and inverter temperature measurements at a rate of 10Hz. The vehicle powertrain platform acts as a HiLS platform for the DuT because the dSPACE^{TM} real-time control prototyping (RCP) module that is programmed as the VCU has all the other components of the EV powertrain model in its simulation, except for the EM and inverter. Instead, the EM and inverter are represented as hardware in the form of the IWMS and is commanded with the torque and angular speed reference by the VCU, while its electrical input, mechanical output, and the temperature of the EM and inverter are monitored by the VCU. For the training of the DT of the IWMS, there are two inputs, including the torque and angular speed commands sent to the inverter from the VCU, and three outputs, including the actual torque response of the motor, the actual angular speed response motor, and the DC-link current consumed by the inverter. The DT of the traction system is used for eco-driving optimization.

3.2 Hi-Fi FEM of the cabin thermal system

In [16] the cabin's thermal model was simulated using a full 3D computational fluid dynamics (CFD) simulation. CFD is a numerical technique that solves fluid flow based on the Navier-Stokes equations, which are partial differential equations (PDE) that do not have analytic solutions in most real-life flow cases. Therefore, the equations are solved using numerical methods after discretization. The computational mesh for the simulations was produced using ANSYS Fluent and a FEM-based open-source C++ library, OpenFOAM, was used for all CFD simulations. The main benefit of CFD simulations is that they provide insights into the details of the fluid dynamic flow structures in the cabin. The CFD simulations were performed with a simplified geometry of the bus; with the computational domain consisting of the bus and a large air space to allow for the simulation of atmospheric flows, since the simulation of atmospheric flows requires special attention due to the hydrostatic pressure of air. The bus cabin was simulated with a full 3D conjugate heat transfer (CHT) approach, which accounts for the heat transfer in the solid regions as well as the fluid regions simultaneously. The simulations were performed with a Detached Eddy Simulation (DES) turbulence modelling method [17]. The benefit of such a turbulence model is that it allows lightweight simulation to be performed for unwanted regions and extensive simulation for the regions of interest, saving computational resources while giving accurate results in the areas of interest. For the time integration, a second order accurate backward scheme was used, while a total variation diminishing (TVD) scheme was used for the discretization of the convection terms.

The interior bus was modeled as separate regions, each of which represents a solid object with specified thermal properties of the solid material. The thermal properties of the windows and doors are represented using the properties of plexiglass, while the bus walls, accordion and the seats are represented using the properties of polyurethane. The passengers of the bus are modelled as volumetric heat sources to account for the heat flow from the human body. The buoyancy generated from the thermal load from the human body is considered significant in indoor air flows. Simulations were performed for different scenarios, including when the bus was driving at different speeds, when the bus was at a stop with doors open, and with different climate conditions such as different wind speeds and ambient temperatures to build a complete picture of the cabin temperature variation and their evolution over time. In this research, the inputs to and output from the extensive simulations carried out in [18] using the functional mock-up (FMU) of the CFD model developed in [16], i.e., the Hi-Fi dataset from that project, are used to train the DT of the cabin's thermal model for use with eco-comfort optimization. The DT inputs are the passenger profile, the ambient temperature profile, the waste heat profile of the various powertrain components, the bus doors opening and closing profile, the cabin setpoint temperature profile, and the constant inputs of solar irradiation and ambient humidity. The outputs from the DT are the actual cabin temperature and the heat pump and the PTC heater power profile.

4 Results and Analysis

One of the approaches of DT design is that the twin is trained only once, using whatever necessary algorithm, and then deployed. Once deployed, the twin is used to make predictions based on incoming measurement data from the field. Strong emphasis was placed on the design architecture of the underlying NN and the training method so that the DT can make accurate predictions using new data as much as it could using the data it trained on. Therefore, the training method needed to ensure that there was no ‘overfitting’ of the data during training. Another approach of DT design is that the underlying NN is required to continually evolve with incoming data, in real time, even as it is using the incoming data to make predictions. The added value of this research is the following:

1. Using a NN, find the best approximation to system dynamics of a powertrain component using limited training data. This does not have to be in real time and needs to be done only once.
2. Continually evolve and tune the NN, using incoming data. This training must be in real-time.
3. Once the confidence level of the NN rises above 95%, it can be used to make predictions; however, the training will be continued indefinitely.

Experimentally, four sets of measurement data were taken, two from an urban driving scenario with lots of traffic lights and vehicles on the road, and two from highways. The measurement data was sampled at a rate of 10k samples per second. The initial training data that is presented in Table 1 is from an urban context. For the LM algorithm, the data was down sampled to 500Hz, as the algorithm takes a long time to execute, and it was not desirable to spend an inordinate amount of time to train using a very large data set. The urban scenario consisted of a little more than 10 minutes of driving data.

Table 1: The execution characteristic of different training algorithms for 10 minutes of driving data

	Number of iterations	**Root means squared error**	**R^2 value of regression**	**Time required**
LM algorithm, down sampled to 500 Hz	Torque: 901 I_{dc}: 1000 Speed: 793	Torque: 7.308 I_{dc}: 0.874 Speed: 17.664	Torque: 0.976 I_{dc}: 0.982 Speed: 0.994	7.25h with 1-core (3h with 12-core)
SCG algorithm, down sampled to 500 Hz	Torque: 290 I_{dc}: 371 Speed: 187	Torque: 9.583 I_{dc}: 1.540 Speed: 38.910	Torque: 0.960 I_{dc}: 0.943 Speed: 0.976	2.5 minutes with 1-core
SCG algorithm, original 10kHz sample	Torque: 1000 I_{dc}: 1000 Speed: 1000	Torque: 9.165 I_{dc}: 1.449 Speed: 35.355	Torque: 0.964 I_{dc}: 0.949 Speed: 0.980	3h with 1-core (0.5h with 12-core & GPU)
SCG algorithm to re-train a LM trained NN, original 10kHz sample	Torque: 134 I_{dc}: 288 Speed: 163	Torque: 7.457 I_{dc}: 0.892 Speed: 18.028	Torque: 0.976 I_{dc}: 0.980 Speed: 0.994	4.5 minutes to 7.5 minutes with 12-core & GPU

The result shows that the approximations of the SCG algorithm are much worse compared to the approximations of the LM algorithm, when both algorithms start from the same initial starting point. However, the data also shows that if the starting point of the SCG algorithm is first determined using the LM algorithm, then the SCG algorithm can approximate as well as the LM algorithm but at a fraction of the time. The following can be determined from the data presented in Table 1: a) LM is very accurate but slow and takes a long time to converge, 2) SCG is orders of magnitude faster than LM, but the accuracy is twice as bad as the LM algorithm. 3) It is determined that with the starting points of the NN configuration optimally selected prior to the training, the SCG will achieve high accuracy during training. This is achieved by taking the pre-trained NN using LM, and then re-training using SCG; this allows the SCG algorithm to keep the accuracy of the LM, but at a real-time processing rate. As can be seen from the table, the training of a 10-minute data at the full sampling rate takes less than 7.5 minutes, so one batch of data can be processed before the next batch of data is ready for processing. Figure 3 shows the outputs of the trained NN to the original data set; two observations are noticed from the results – 1) the simulated estimates have a very good fit to measured data, 2) the simulated estimates are able to suppress somewhat the high frequency noise in the measurement data, and 3) the simulated estimates sometimes output a very short duration spike/pulse that is not present in the measurement data, easily removed using a moving median filter.

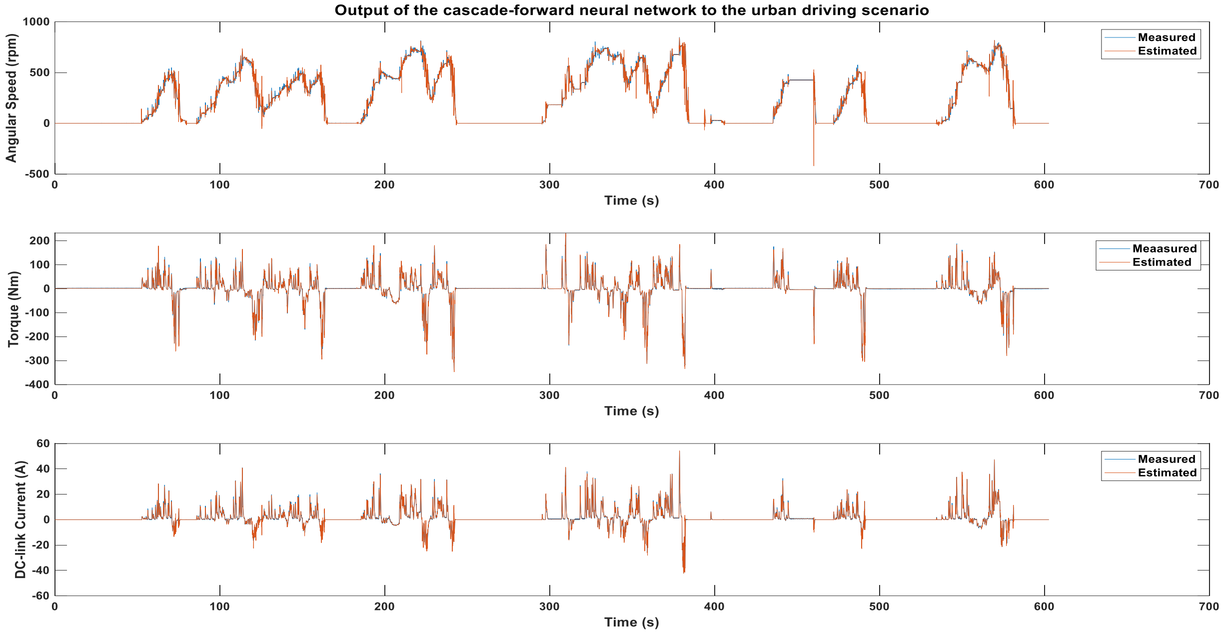

Figure 3: Output of the trained CFNN using urban driving cycle

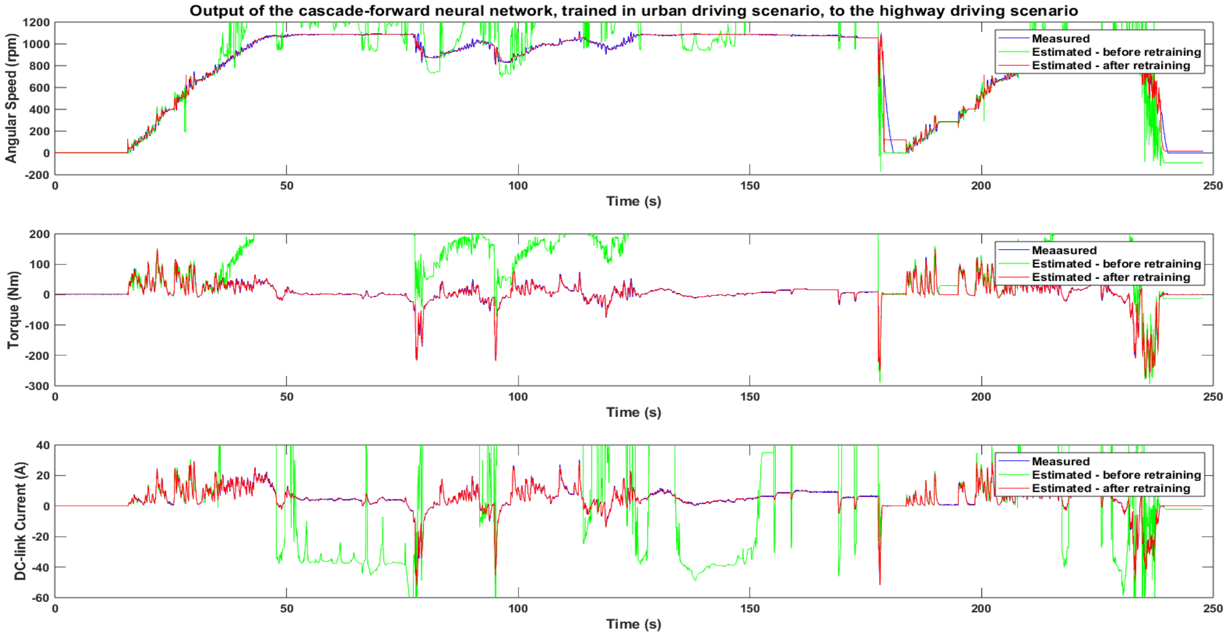

Figure 4: Output of the CFNN trained with urban driving scenario to a highway driving cycle, and then with the NN retrained to the highway scenario.

Figure 4 shows the output of the NN to a different scenario, namely the highway scenario. From the figure, it is noticed that there is a huge mismatch between the simulated estimates and the actual output; thus, there is a need to retrain the NN with the new data. The result of the retraining is also shown in the figure, and it is seen that the simulated estimates again closely match the measurement data. Furthermore, in subsequent tests it was verified from the output of the retrained NN to the original urban scenario that the retraining of the NN to adapt to the highway scenario did not remove the ability of the NN to also recognize and simulate the original urban scenario. Finally, the retrained NN was subjected to a different urban and highway scenario, and it was noticed that there was decent ability of the NN to recognize and output the proper estimate of the IWMS mechanical and electrical behavior to these new scenarios. It was also significant that the number of iterations required during the retraining process decreased by a factor of 3 because the NN weights are already configured to their optimal values from previous training and only slight tuning takes place during retraining.

5 Conclusion

The open vehicle powertrain platform (OVPP) was design and development for the purpose of vehicle powertrain testing and characterization. Towards this end, the OVPP consists of a driving simulator that allows a driver to provide realistic and real-time driving data to the vehicle powertrain platform, where the HiLS of the powertrain component (i.e., the DuT) is conducted. The measurement data from the HiLS using the IoT system is then utilized to train a DT that represents the DuT using a CFNN, that is first accurately trained using the LM algorithm, and then subsequently adapted to new data using the SCG algorithm. This provides continuous real-time training of the DT and results show that a pre-trained NN is more easily and quickly retrained to newer data. The added value of this research in comparison to past DT research was that previously the research focused on developing and then deploying a DT for various applications, but the DT only underwent training once before deployment. Thus, various techniques including regularization, cross-validation, early stopping, training with ever larger datasets, and reducing the network complexity are attempted to ensure that there was no overfitting during the training process to allow the NN to handle new (unseen) or different data scenarios [19]. All such techniques were unnecessary in this research since, unlike the first training process, the re-training process happened in real-time and applied continuously to the NN allowing the NN to quickly adapt to new scenarios while the network still retained the characterization of prior scenarios. Using this approach is the ideal training scenario for the NN as it now has access to unlimited training data. The trained DT was used to output the motor response in terms of its angular speed and torque, based on the input angular speed and torque command sent to the inverter. Furthermore, the DT was also used to generate the output of the DC-link current that was required by the motor. From these data, the dynamic power consumption of the motor was analyzed in real time. This real-time simulation ability of the DT allows it to be a component of various optimization algorithms.

Acknowledgments

This project has received funding from the European Union’s Horizon Europe Research and Innovation program under Grant Agreement No. 101095882. The authors also acknowledge Flanders Make for their support to our research group.

References

[1] Website: *The Future of Urban Mobility: How electric buses are transforming public transportation | Karsan.com*, URL: https://www.karsan.com/en/blog/e-mobility-blog/the-future-of-urban-mobility-how-electric-buses-are-transforming-public-transportation, updated: September 2024, accessed: December 2024.

[2] Capasso, C., Iannucci, L., Patalano, S., Veneri, O., Vitolo, F., *Design approach for electric vehicle battery packs based on experimentally tested multi-domain models*, Elsevier, Journal of Energy Storage, Volume 77 (2024), 109971, ISSN: 2352-152X, DOI: https://doi.org/10.1016/j.est.2023.109971.

[3] Mo, W., Lin, Y., Li, J., Lian, C., Liu, H., *An optimized hybrid battery pack with high energy density and high safety*, Elsevier, Journal of Chemical Engineering Science, Volume 297 (2024), 120290, ISSN: 0009-2509, DOI: https://doi.org/10.1016/j.ces.2024.120290.

[4] Li, Y., Li, X., Zhang, C., Zhang, Y., *Optimizing the photovoltaic-assisted electric bus network with rooftop energy supply*, Elsevier, Journal of Renewable Energy, Volume 234 (2024), 121244, ISSN: 0960-1481, DOI: https://doi.org/10.1016/j.renene.2024.121244.

[5] Romanos, I., Theano, I., Christina, I., Loukas, K., Angeliki, P., Maria-Eleni, M., Maria-Eirini, A., Nikolaos, P., Giannis, K., Panayiotis, D., Christina, P., Eleni I. V., Konstantinos, K., Nikos, M., Demetris K., *Solar-powered bus route: introducing renewable energy into a university campus transport system*, EGU, Advances in Geosciences, Volume 49 (2019), pp. 215-224, DOI: https://doi.org/10.5194/adgeo-49-215-2019

[6] Hasan, M. M., Avramis, N., Ranta, M., Saez-de-Ibarra, A., El Baghdadi, M., Hegazy, O., *Multi-objective energy management and charging strategy for electric bus fleets in cities using various eco-strategies*, In MDPI/Sustainability. 13(14), 7865 (2021), DOI: https://doi.org/10.3390/su13147865.

[7] Medina, R., Avramis, N., Rath, S., Hasan, M. M., Tran, D. D., Maleas, Z., Hegazy, O., Wilkins, S., *Multi-*

layer energy management system for cost optimization of battery electric vehicle fleets, in proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2024), Angers, France. Publication: Vinel, A., Berns, K., Ploeg, J., Gusikhin, O. (eds.), Science and Technology Publications, Lda, p. 112-124, DOI: https://doi.org/10.5220/0012627000003702.

[8] Hasan, M. M., El Baghdadi, M., Hegazy, O., *Energy management strategy in electric buses for public transport using eco-driving*, in proceedings of 15th International Conference on Ecological Vehicles and Renewable Energies (EVER 2020), Monte Carlo, Monaco, published in IEEE Xplore, ISBN (Electronic): 978-1-7281-5641-5, DOI: https://doi.org/10.1109/EVER48776.2020.9243130.

[9] Hasan, M. M., Maas, J., El Baghdadi, M., de Groot, R., Hegazy, O., *Thermal management strategy of electric buses towards eco-comfort*, in proceedings of the 8th Transport Research Arena Conference (TRA2020), April 2020, Helsinki, Finland, DOI: https://doi.org/10.5281/ZENODO.4292325.

[10] Hasan, M. M., *Co-design Optimization, eco-features, and development of digital twin with internet of things for fleets of electric buses in cities*, Ph. D Thesis, Vrije University Brussel, Jan 2024, ISBN: 9789464948103

[11] Nikolopoulou, K., Website: *What is machine learning? | A beginner's guide – (www.scribbr.com)*, URL: https://www.scribbr.com/ai-tools/machine-learning/, updated: August 2023, accessed: September 2023.

[12] Website: *Types of neural networks | GeeksforGeeks*, URL: https://www.geeksforgeeks.org/types-of-neural-networks/, updated: February 25, 2025, accessed: April 2025

[13] Pujol, J., *The solution of non-linear inverse problems and the Levenberg-Marquardt method*, in Geophysics (2007), 72(4), pp. W1-W16, ISSN: 0016-8033, DOI: https://doi.org/10.1190/1.2732552.

[14] Andrei, N., *Scaled conjugate gradient algorithms for unconstrained optimization*, in Computational Optimization and Applications (2007), Vol. 38, pp. 401–416, DOI: https://doi.org/10.1007/s10589-007-9055-7.

[15] Moi, T., Cibicik, A., Rølvåg, T., *Digital twin-based condition monitoring of a knuckle boom crane: An experimental study*, Engineering Failure Analysis, Volume 112 (2020), 104517, ISSN: 1350-6307, DOI: https://doi.org/10.1016/j.engfailanal.2020.104517.

[16] Antilla, J., *Report on the thermal design and layout of vehicle system*, e-BRT2030 project deliverable report D3.4, August 2024

[17] Strelets, M., *Detached eddy simulation of massively separated flow*, in proceedings of 39th Aerospace Sciences Meeting and Exhibit (2001), Reno, 8-11 January 2001, DOI: https://doi.org/10.2514/6.2001-879

[18] Hasan, M. M., Geury, T., *Report on the e-BRT reference concepts and e-BRT eco-system*, e-BRT2030 project deliverable report D3.3, August 2024

[19] Ying, X., "An overview of overfitting and its solutions", in Journal of Physic: Conference Series (2019), 1168(2) 022022, DOI: 10.1088/1742-6596/1168/2/022022.

Presenter Biography

Mohammed Mahedi Hasan received his M.Sc. in Mechatronics Engineering from the American University of Sharjah, Sharjah, United Arab Emirates, in 2006, and his B.Sc. in Electrical Engineering from the Georgia Institute of Technology, Georgia, United States of America, in 2002. He worked as an Electronics Engineer at the Research & Development Center, UAE Armed Forces, between 2007 and 2012; and between 2014 and 2018, he worked as a Lecturer in private universities in Bangladesh. He obtained his PhD in Engineering Science (with highest honors) from Vrije Universiteit Brussel Brussels, Belgium in January 2024. He now works as a post-doctoral researcher in EPOWERS team, MOBI research group, VUB. He has been involved with various European projects, including ASSURED, LONGRUN, URBANIZED, EBRT2030. His research interests include heavy duty vehicle powertrain modelling, co-design optimization, energy & thermal & charging management strategies, and digital twin design and development utilizing IoT framework.