Analysis of Heavy Vehicles Rollover with Artificial Intelligence Techniques

Analysis of heavy vehicles rollover with artificial intelligence techniques

Filippo Velardocchia[1][0000-0001-6329-2256], Guido Perboli[1][0000-0001-6900-9917], and Alessandro Vigliani[2][0000-0001-6423-9852]

[1] DIGEP - Politecnico di Torino, Turin, Italy
[2] DIMEAS - Politecnico di Torino, Turin, Italy
{filippo.velardocchia,guido.perboli,alessandro.vigliani}@polito.it

Abstract. The issue of heavy vehicles rollover appears to be central in various sectors. This is due to the consequences entailed in terms of driver and passenger safety, other than considering aspects as environmental damaging and pollution. Therefore, several studies proposed estimating and predicting techniques to avoid this critical condition, with especially good results obtained by the using of Artificial Intelligence (AI) systems based on neural networks. Unfortunately, to conduct these kind of analyses a great quantity of data is required, with the same often difficult to be retrieved in sufficient numbers without incurring in unsustainable costs. To answer the problem, in this paper is presented a methodology based on synthetic data, generated in a specifically designed Matlab environment. This has been done by defining the characteristics of an heavy vehicle, a three axles truck, and making it complete maneuvers on surfaces and circuits purposely created to highlight rollover issues. After this phase, represented by the generation and processing of the data, follows the analysis of the same. This represents the second major phase of the methodology, and contains the definition of a neural networks based algorithm. Referring to the nets, these are designed to obtain both the estimate and the prediction of four common rollover indexes, the roll angle and the Load Transfer Ratios (LTR, one for each axle). Very promising results were achieved in particular for the estimative part, offering new possibilities for the analysis of rollover issues both for the generation and the analysis of the data.

Keywords: heavy vehicles · rollover risk indicators · neural networks.

1 Introduction

The work presented will focus on the analysis of heavy vehicles rollover, in the specific case of an articulated truck (three axles), and the possibility of estimating and preventing the same through Artificial Intelligence (AI) techniques. The study of this issue is of particular interest since this type of vehicle, in Europe alone and in Canada and the United States, is responsible for the transport of more than 80% of goods [18]. In particular, referring to the U.S, rollover accounts for 13.9% of large truck fatal crashes [16]. As a matter of fact, statistics on road

safety show that accidents involving at least one heavy vehicle are often more dangerous than those involving other types of vehicles [5,14,15], as can be noted specifically in [14], since they represent only 4.7% of accidents (in France in 2014) but cause more than 14% of fatalities. It has also to be highlighted that rollover accidents tend to have several implications, such as the damaging of roads or even environmental pollution [6]. This is a direct consequence of the fact that they are often correlated to heavy vehicles, characterized by a high center of gravity and articulated steering mechanism [1]. To prevent rollover accidents, it is therefore necessary to design a successful safety warning system to notice potential rollover danger [4,19]. This led to the development of different models and techniques, both from the point of active [3, 7, 11] and passive [10, 17, 20] rollover protection system, over than, specifically, studies focused on heavy vehicles characterized by articulated steering. In this last case, in particular, it is important to distinguish between simplified [8] and complex [12] models, with the latter characterized by a non suitable real-time application and the first ones with a limited range of operations. To solve these problems, recently, in the vehicle research, different models based on machine learning and empirical data were built [4, 14], allowing the creation of data-driven models characterized by rapid and precise responses. The main issue in these cases is the fact that is not always possible to have a sufficient number of empirical data to generate an efficient algorithm, both for time and costs reasons [2]. In this paper, therefore, we developed a methodology based on Recurrent Neural Networks (RNN) receiving and analyzing synthetic data, provided to them by a realistic heavy vehicle model. This was done connecting these two aspects in a Matlab environment, using the Simscape and Deep Learning software to define, respectively, the three axles truck generating the data and the networks estimating and predicting rollover indicators from the same.
The main contributions are:

- The generation of data-driven models based on two different typologies of recurrent neural networks, one with estimation and the other with prediction purposes of four rollover risk indicators. These are represented by the roll angle and the three load transfer ratio (one for each axle), common parameters in this kind of analysis [9]. Recurring to this form of artificial intelligence allows having very flexible, adaptable and updatable models, other than the possibility of obtaining immediate response with good accuracy and reducing significantly time and economic costs.
- The creation of a realistic environment to generate a sufficient quantity of data for the neural networks. This kind of algorithm needs a high amount of data to work effectively, and it is often difficult to obtain the same from empirical operations, both for on-field problems and costs reasons. As a solution, in this paper is proposed a three axles articulated truck model, obtained using Simscape and, more precisely, a version originally developed by Steve Miller, Simscape Vehicle Templates. This allowed us to design maneuvers, surfaces, Pre and Post Processor codes, specifically developed to generate realistic data regarding the movement and the conduct of the considered ve-

hicle, as well as the implementation of the main modifications in a practical User Interface (UI) through the Matlab livescript.

The article is organized in three main sections (besides the introduction), inevitably linked to the two great phases of the work, the generation and elaboration of the data and the analysis of the same. In particular, after a brief overall overview in Section 2, Section 3 provides a more specific presentation of the methodology developed, in particular analyzing first the generation of the data (subsection 3.1) and, then, focusing on what procedure has been followed at an AI level (subsection 3.2). Finally, Section 4 concludes the paper, adding some considerations on possible future developments.

2 System overview

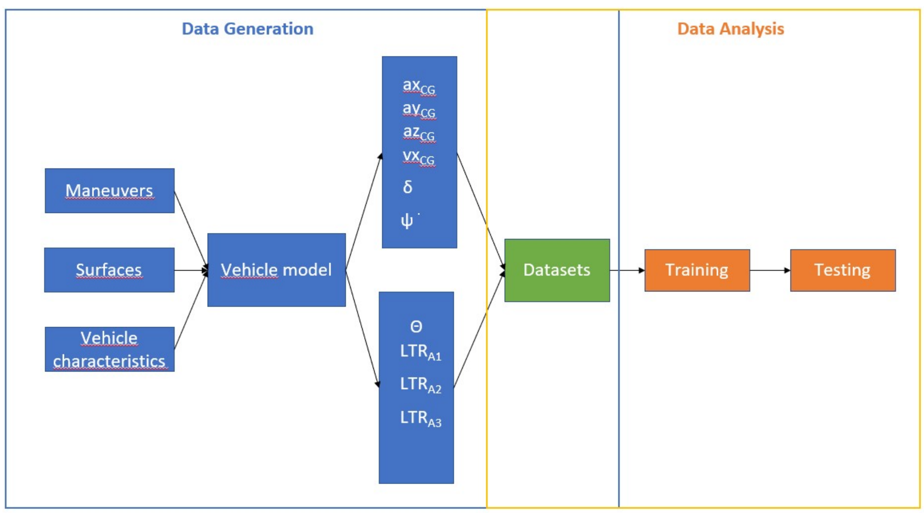

Fig. 1. System overview

As is possible to see, in Figure 1 a brief overview of the overall system is provided. This is organized into two main parts, the data generation and the analysis of the same. Referring to the first phase, this is characterized by the definition of the simulation environment (Simscape), particularly in terms of maneuvers, surfaces and vehicle characteristics. Once designed and introduced them in a specific User Interface (UI), it is possible to utilize the latter to run the vehicle model previously composed, an articulated truck with three axles, obtaining different parameters of interest. These, retrieved by several Post Processor codes, detail the inputs and the outputs characterizing the AI algorithm. In particular, the first ones are represented by the steering angle (δ), the acceleration of the

Center of Gravity (CG) in its three components, the longitudinal velocity and the yaw rate($\dot{\psi}$), while the second ones by the roll angle (θ) and the Load Transfer Ratio (LTR). Knowing the elements defining our data-driven models, after the generation of datasets containing the same, one for each specific maneuver requested to the truck (23 in total), it is possible to elaborate the RNN based on them. This process is characterized by different steps, for the sake of brevity reassumed in the training and testing blocks reported in Figure 1. In the first one, in particular, is provided an architecture definition of the two typologies of neural networks (as anticipated, one for estimation, the other for predictive purposes), followed by the standardizing of the parameters and the effective training of the algorithms, based on 19 of the 23 datasets generated before. After this part, it is possible to continue with the testing phase of the algorithm, done with 4 datasets excluded by the training of the networks and, therefore, completely unknown to the same.

This system led to the obtaining of promising results especially for what regards the estimation algorithm, as it will be possible to see in the next sections more in detail.

3 Methodology

3.1 Simscape: data generation and elaboration

In order to generate and obtain realistic data for the definition of the AI algorithms and their effectiveness, it is, as anticipated, proposed a methodology based on the Simscape software in the Matlab/Simulink environment. This instrument allows building models of physical components (such as electric motors, suspensions, etc.) realistically interacting with each other, thus resulting in a suitable program for this part of the discussion. It has to be noted that, to be able to affirm it, a sensitivity analysis was conducted on different types of vehicles and standard maneuvers (e.g., step steer, ramp steer etc.), obtaining results extremely similar to the ones that can be found in literature, thus confirming the reliability of the software. Regarding the same, a particular template was used, "Simscape Vehicle Templates", originally developed by Steve Miller. This contains several predefined components, presenting the possibility to design the model of interest without having the need of starting completely anew. In our specific case, this allowed us to rapidly design our vehicle (2), a three axles truck. This was built keeping in mind the goal of the research, highlighting rollover issues for heavy vehicles and proposing both a simple way to generate reliable data and an AI system to analyze the same.

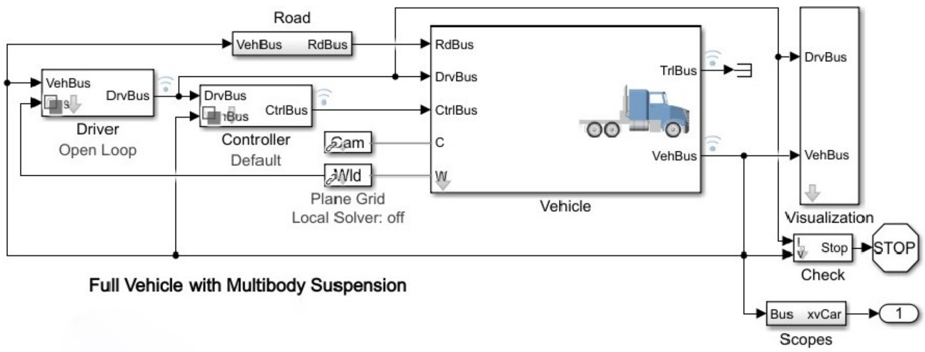

Fig. 2. Simulink model used for the data generation

As can be seen in Figure 2, the model is characterized by the presence of numerous blocks interconnected with each other (e.g.: Road, Vehicle, Driver) and each having several subsystems composed of different elements. To interact effectively with them, other than recurring to the use of specific scripts, it is advisable to resort to a User Interface (UI). This is a simple yet powerful instrument that allows to rapidly define the key elements of the vehicle and its simulation environment (maneuvers, surfaces etc.), and it is here introduced in order to help the reader in the eventual replica of the methodology proposed. In particular, considering the original UI that allowed to define and recall the starting elements of our model, in Figure 3 are reported its main characteristics for this research.

In order to understand the actual functioning of the UI, and how much the elements present within it are strongly correlated with the programming environment, it is highlighted that, by selecting a generic maneuver (e.g. WOT Braking, left tab of Figure 3), there is an automatic interaction with a code (which can be accessed through the Code button) that takes care of setting components such as path, trajectory and surface on which the vehicle will have to perform its motion. To update any characteristic of the maneuver, like the steering angle, it is necessary to interact with the databases of the same (Figure 3, central part) while, to observe the results, with specific codes in the Results section (right tab of Figure 3).

Defined the model used and its correlated UI, it is now possible to present what effectively has been introduced anew to generate the data. The first issue was the creation of maneuvers and surfaces specifically designed for a three axles truck. As a matter of fact, Simscape Vehicle Templates provides already several maneuvers and surfaces in its predefined environment, but none of them are designed for the vehicle of interest, even more for the study of rollover eventualities. Therefore, we defined and introduced in the Simscape environment 23 new maneuvers purposely thought for our vehicle and the casuistry of this research, in order to be able to generate a sufficient quantity of quality data for the

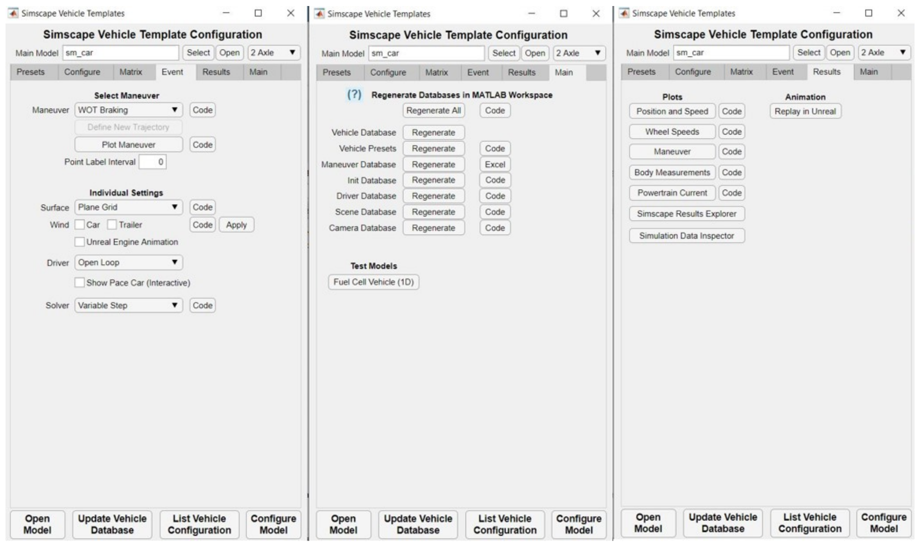

Fig. 3. User Interface without addons

neural networks based algorithm. Each new maneuver required the definition of appropriate trajectory parameters, such as the steering angle or the longitudinal distance, depending on the characteristics of the aforementioned vehicle and of the circuits considered. The number of new maneuvers, 23, has been chosen in order to be sure to have enough data to be able to make effective estimates and predictions with the neural networks. As a matter of fact, each maneuver represents a database for the AI algorithm. The dimensions of these databases are very similar (approximately four thousands data each), to avoid changes in the weights of neural networks simply due to differences in the dimensions of the datasets. It has to be highlighted, moreover, that the maneuvers are very different one from another, leading to a very general composition of the data, that are composed by typical parameters of vehicle dynamics (e.g., acceleration, velocity, etc.). These, clearly, are time dependent ones, leading to the use, in subsection 3.2, of LSTM layers for defining the neural networks.

Once that the new maneuvers, and therefore our future datasets, were designed, there was the necessity to save the newly generated data. For this purpose, after the running of the simulation, we implemented new post processing codes that aimed both to save the data, and also to generate the same through analytical correlations starting from the ones provided by the final bus of the model (Figure 2, in case the obtaining could not be done directly). Moreover, through the implementation of post processor codes, it was also possible to verify again, graphically, after the general sensitivity analysis, the correctness of the results obtained by the simulation in comparison with what proposed in literature. This confirmed that the vehicle dynamics parameters (36 in total for each dataset) attained were reliable and ready to be used. Several other additions were then made in order to facilitate the processes described above in aspects as the uploading of the updated maneuver database and the communication between the various scripts defining the simulation environment. This led to the introduction of a new UI (using the Matlab livescript), Figure 4, that has here the purpose of reassuming the main contributions of this first part of the methodology, and in a practical scenario the goal to help the system created to be as user friendly as possible.

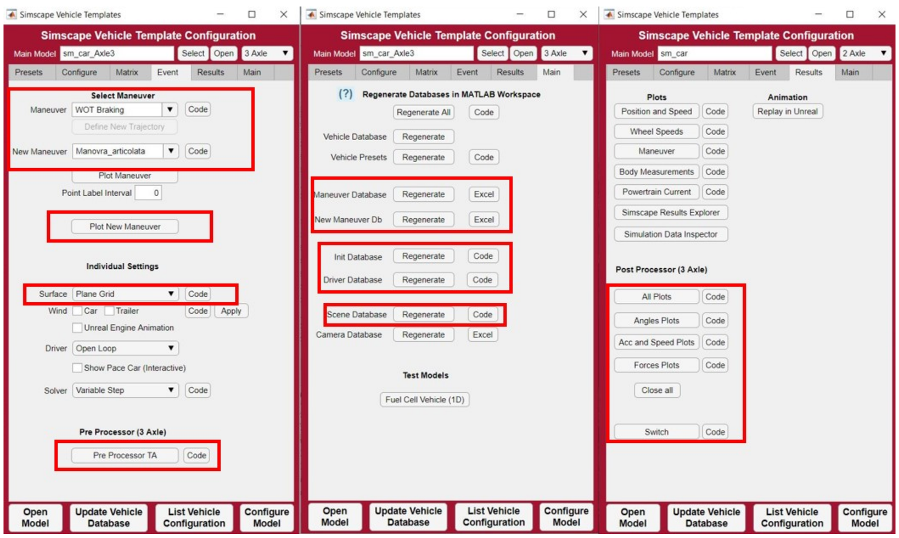

Fig. 4. Modified UI, Simscape

After completing the above steps, it was then possible to continue, knowing the datasets of interest and the tipping indicators to be obtained (Load Transfer Ratio of the three axles and roll angle), with the analysis of the same through AI using the generated synthetic data.

3.2 AI: data analysis and Results

After the generation and selection of the data of interest in the first part of the discussion, it is now necessary to understand how to analyze them to obtain the rollover index indicators. With this in mind, after having seen the state of the art in literature [4], the second part of the methodology is presented. This consists, as previously anticipated, in the definition of an AI algorithm based on neural networks. This has been done in Matlab/Simulink environment using the Deep Learning Toolbox, allowing a very efficient communication between the nets and the Simscape environment by adequately coding the UI. Purpose of the algorithm developed is to achieve good estimates and predictions of the rollover indicators object of our research: the roll angle and the three LTR of each axles. This has to be done by analyzing the datasets whose generation has been described in the subsection 3.1. In particular, the analysis requires a training and a testing phase, considering that we are using neural networks.

Entering more in details, the process has been structured as follows. First, clearly, the training phase of the algorithm. This includes, at the beginning, the retrieving of the datasets containing the data of interest by using a dedicated script. After doing that, it is necessary to operate a selection on the 36 variables associated to each different maneuvers. Trying to simulate a realistic situation, we chose to train our networks on 6 main parameters, typical of trajectory studies and easily attainable by the ECU of the considered typology of heavy vehicle. This parameters, as mentioned in Section 2, are: the steering angle, the acceleration of the Center of Gravity (CG) in its three components, the longitudinal velocity and the yaw rate. Selected the parameters, before the definition and the training of the neural networks, it was necessary to standardize the same. For doing that, we used the mean and the standard deviation of each variable, calculable since, in the training phase, the entire dataset is known. Defined the parameters and their standardization, it is now possible to design the core of our algorithm: the architecture of the neural networks. First of all, we defined two different typologies of nets, one for estimative purposes, net_1, and the other for predictive ones, net_1_1. These networks share the same architecture, but are characterized by being one (net_1_1) subsequent to the other (net_1). As a matter of fact, hypothesising a real time application, purpose of net_1 is to estimate the four rollover indicators by using the six parameters aforementioned, therefore giving results at the same time of what are, effectively, its inputs. We then have a net that has 6 inputs (features) and 4 outputs (responses). This phase of estimate is strongly necessary because, otherwise, it would be impossible to attain rollover indicators with immediate response times (this is one of the reasons to use neural networks and not a classic dynamic model). Obtained the four rollover indexes, the same are then put out of phase forward in time compared to the input ones, in a predictive perspective. Synthesising, net_1 is characterized by having 6 different inputs and 4 different outputs, while net_1_ 4 inputs coinciding with the four outputs obtained by net_1, and 4 outputs that are simply the inputs shifted of time unities (in general corresponding to 3-4 seconds, but it is at discretion of the user). These two networks, as anticipated, share the same structure, and are characterized by being RNN, typical for time series analysis [13].

In particular, the architecture is composed of 4 different layers, with the first defined as a sequence input one (where the number of features is exploited), the second as an LSTM, the third as a fully connected (number of responses) and the fourth as a regression layer. It has to be noted that the LSTM layer is the one that learns long-term dependencies between time steps in time series and sequence data, and is therefore the one directly related to the number of neurons (hidden units) selected. In our case, we present neural networks with 100 neurons, number defined after having tried empirical rules to identify the same, and then after several tries and having seen that, with more than 100 of them, the results did not improve in accuracy and the process was becoming too time consuming (while with a minor number of neurons, the results start to present a decline in accuracy). The number and the typology of layers have been chosen following similar criteria. After having designed the main parts of the architecture, several training options were then defined. More specifically, we implemented an adaptive moment estimation with gradient threshold equal to 1, an initial learn rate of 0.01 with a drop factor of 0.2 after 120 epochs (with a maximum of 200), and an option that does not allow the data to be shuffled (since we want to maintain the original order). Other minor settings were than defined, but always after comparing the same with their alternative (for example, the stochastic gradient descent with momentum gave us results not suitable for any application, at least with our data) and evaluating the following results.

Designed the architecture, the training could take place. We used 19 of the 23 databases for this purpose, and the entirety of each dataset has been used (for what regards the 6 features) to train the networks and update their weights progressively. It is important to highlight again that, in a real case simulation, the first has the task of obtaining the overturning indicators (otherwise difficult to calculate quickly and precisely) and, by providing them in output, give way to the second type of network to predict an advanced trend over time. The training of the networks is done by trying to simulate this kind of situation, subjecting the two types of networks to a continuous update with the progressive addition of the data of each maneuver, in order to obtain a system as adaptable as possible. It has to be noted that we defined 2 different typologies of neural networks but, in reality, we obtained 19 of them. As a matter of fact, for each update of the networks, the script is programmed to save the new version in a specific folder, allowing us to evaluate, by seeing the graphs, if introducing a certain type of maneuver improves or damages the performance (in our case, there is a general improvement, obtained also by the elimination of seven other maneuvers that did not contribute to the improvement of the net). Clearly, during the training phase, it is exploited the fact of knowing in advance the target values of roll and LTR on which the networks must be trained. In our case, these are known from virtual simulation on Simscape, in case of not synthetic data a specific script should be prepared to calculate the same values from those available.

Completed the training phase, it is possible to describe the final part of the methodology developed, the testing of the AI system. This has been done by using 4 of the 23 maneuvers defined in the subsection 3.1, unknown to the networks and characterized by being very different one from each other, other than being of particular interest from the point of rollover analysis (strong stresses for the vehicle). The datasets tested are characterized by passing their 6 characteristics parameters to net_1, that obtains the 4 rollover indexes that will be the input for net_1_1. The testing phase is simulated in the perspective of highlighting the general behavior of the neural networks and the efficiency of the training. Therefore, as an example of the results obtained, in Figure 5 and 6, it is possible to see the ones achieved with the network corresponding to the end of the training session (the nineteenth). It is important to highlight that the maneuver in exam was created simulating a movement of the truck on the F1 circuit of Suzuka (in a simplified version). It should be noted that, contrary to the training phase (where the times vary from minutes to hours, depending on the machine used and the possibility of using parallel computing), in this case, the time of response are immediate (order of tenths of a second), justifying the using of neural networks to obtain the indexes researched.

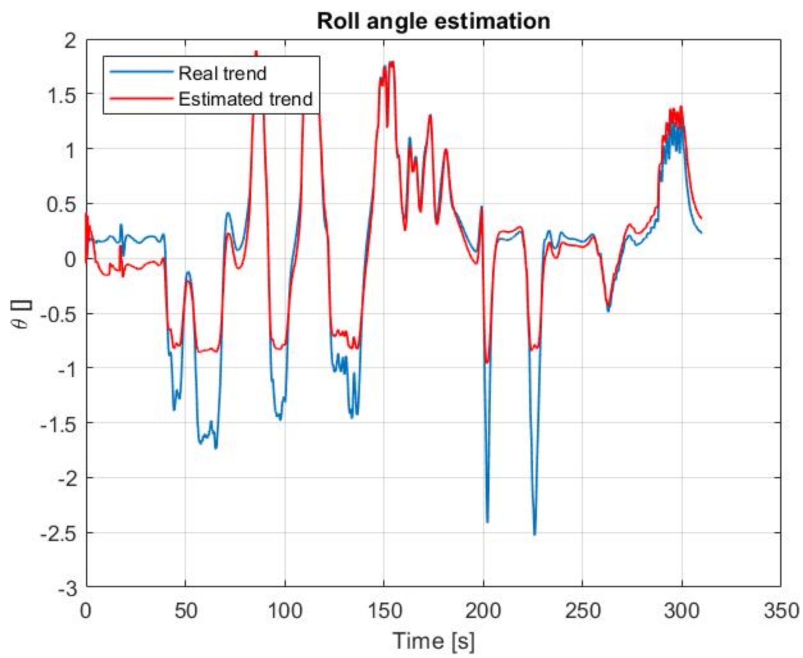

Fig. 5. Roll trend, net_1, maneuver on Suzuka circuit

Referring to Figures 5 and 6, it is easy to observe really promising results from net_1. As a matter of fact, this neural network seems able, despite being trained in a general way (considering the variety in maneuvers of the datasets composing the training), to estimate with a very short time (tenth of seconds) and with good accuracy the requested indexes. This is confirmed even in a heavily conditioned noise situation, as can be seen in Figure 7.

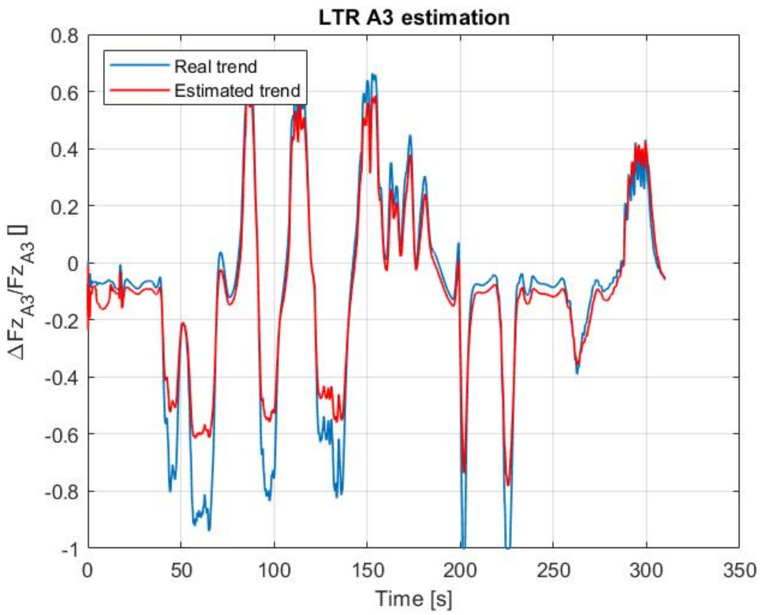

Fig. 6. LTR A3 trend, net_1, maneuver on Suzuka circuit

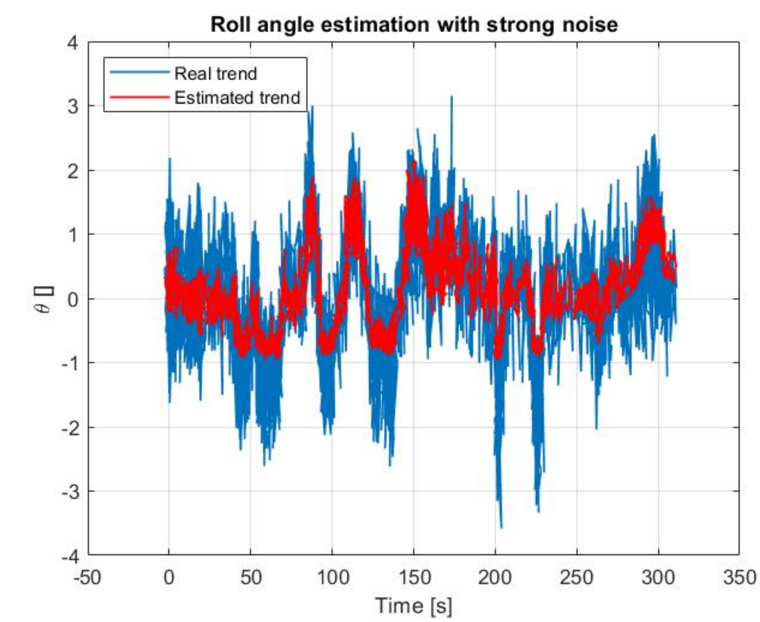

Fig. 7. Roll trend, net_1, Suzuka circuit, noise presence

It has to be noted that, since the datasets are composed almost in their entirety by maneuvers carried out on flat surfaces, net_1 tends to be able to correctly estimate the parameters of unknown ones but, if the same take place in terrain characterized by bumps or hills (respectively, Rough Road and Plateau, Figure 9), as it is easily understandable, the estimation of the indicators loses in precision and accuracy.

Regarding what it has been obtained with net_1_1, as it is possible to observe in Figure 8, the results still show the needing for an improvement of the same, probably both from code writing and, in particular, datasets training aspects.

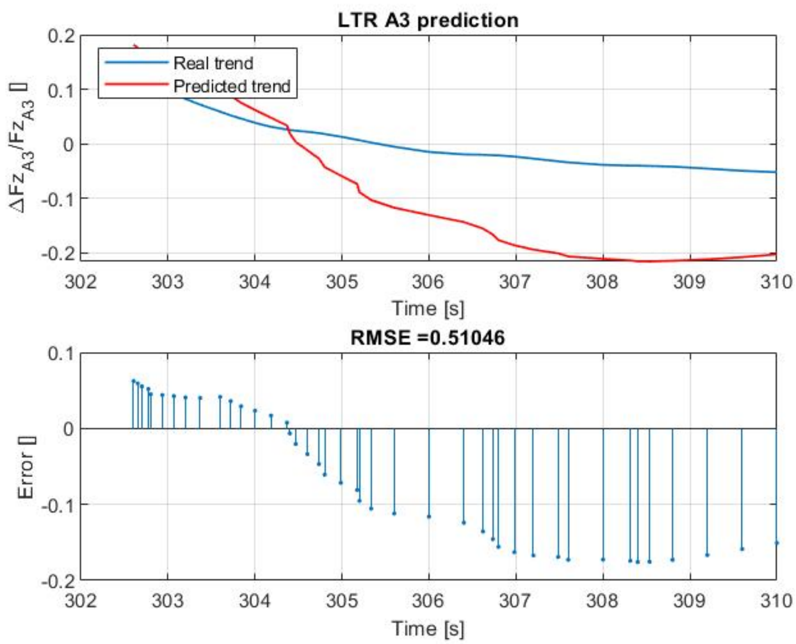

Fig. 8. LTRA3 trend, net_1_1, maneuver on Suzuka circuit

As anticipated, even in this case the main changes were introduced in the UI. This includes the possibility to interact immediately with the training and testing codes, modifying or adjusting them varying on necessities.

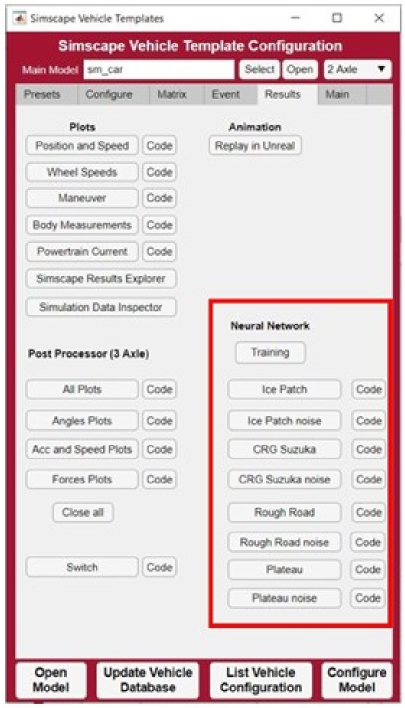

Fig. 9. UI with neural networks implementation

4 Conclusions and possible future developments

The methodology presented revealed to be particularly effective in generating realistic synthetic data, representing a low cost alternative to data otherwise difficult to attain in acceptable time, quantity and costs. These data were then used to design an AI system based on two different typologies of neural networks, one for estimative, the other for predictive purposes. Even if the latter did not perform efficiently enough in terms of accuracy, the first one has already shown encouraging results as an estimator of rollover risk indicators, thanks to immediate responses and good accuracy. This suggests that the algorithm developed could be a valid opportunity for rollover risk estimation, representing a system that is easily adaptable, updatable and versatile. In order to improve the same we are expanding the databases and implementing new training techniques to obtain useful results also for the predictive part.

References

1. Azad, N., Khajepour, A., McPhee, J.: A survey of stability enhancement strategies for articulated steer vehicles. International Journal of Heavy Vehicle Systems **16**(1-2), 26 – 48 (2009). https://doi.org/10.1504/IJHVS.2009.023853

2. Baldi, M.M., Perboli, G., Tadei, R.: Driver maneuvers inference through machine learning. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **10122 LNCS**, 182 – 192 (2016). https://doi.org/10.1007/978-3-319-51469-7_15
3. Braghin, F., Cheli, F., Corradi, R., Tomasini, G., Sabbioni, E.: Active anti-rollover system for heavy-duty road vehicles. Vehicle System Dynamics **46**(SUPPL.1), 653 – 668 (2008). https://doi.org/10.1080/00423110802033064
4. Chen, X., Chen, W., Hou, L., Hu, H., Bu, X., Zhu, Q.: A novel data-driven rollover risk assessment for articulated steering vehicles using rnn. Journal of Mechanical Science and Technology **34**(5), 2161 – 2170 (2020). https://doi.org/10.1007/s12206-020-0437-4
5. Evans, J.L., Batzer, S.A., Andrews, S.B.: Evaluation of heavy truck rollover accidents (2005)
6. Evans, J.L., Batzer, S.A., Andrews, S.B., Hooker, R.M.: Evaluation of heavy truck rollover crashworthiness. vol. 2006, p. 91 – 96 (2006)
7. Ghazali, M., Durali, M., Salarieh, H.: Path-following in model predictive rollover prevention using front steering and braking. Vehicle System Dynamics **55**(1), 121 – 148 (2017). https://doi.org/10.1080/00423114.2016.1246741
8. Iida, M., Nakashima, H., Tomiyama, H., Oh, T., Nakamura, T.: Small-radius turning performance of an articulated vehicle by direct yaw moment control. Computers and Electronics in Agriculture **76**(2), 277 – 283 (2011). https://doi.org/10.1016/j.compag.2011.02.006
9. Imine, H., Benallegue, A., Madani, T., Srairi, S.: Rollover risk prediction of heavy vehicle using high-order sliding-mode observer: Experimental results. IEEE Transactions on Vehicular Technology **63**(6), 2533 – 2543 (2014). https://doi.org/10.1109/TVT.2013.2292998
10. Li, H., Zhao, Y., Wang, H., Lin, F.: Design of an improved predictive ltr for rollover warning systems. Journal of the Brazilian Society of Mechanical Sciences and Engineering **39**(10), 3779 – 3791 (2017). https://doi.org/10.1007/s40430-017-0796-7
11. Li, L., Lu, Y., Wang, R., Chen, J.: A three-dimensional dynamics control framework of vehicle lateral stability and rollover prevention via active braking with mpc. IEEE Transactions on Industrial Electronics **64**(4), 3389 – 3401 (2017). https://doi.org/10.1109/TIE.2016.2583400
12. Li, X., Wang, G., Yao, Z., Qu, J.: Dynamic model and validation of an articulated steering wheel loader on slopes and over obstacles. Vehicle System Dynamics **51**(9), 1305 – 1323 (2013). https://doi.org/10.1080/00423114.2013.800893
13. Rahman, A., Srikumar, V., Smith, A.D.: Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. Applied Energy **212**, 372 – 385 (2018). https://doi.org/10.1016/j.apenergy.2017.12.051
14. Sellami, Y., Imine, H., Boubezoul, A., Cadiou, J.C.: Rollover risk prediction of heavy vehicles by reliability index and empirical modelling. Vehicle System Dynamics **56**(3), 385 – 405 (2018). https://doi.org/10.1080/00423114.2017.1381980
15. Service de l'Observation et des Statistiques: Bilan social annuel du transport routier de marchandises. Tech. rep., SOeS (2015)
16. Us Dept. of Transportation: Traffic safety facts 2016: A compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimates system. Tech. rep., NHTSA (2017)
17. Zhang, X., Yang, Y., Guo, K., Lv, J., Peng, T.: Contour line of load transfer ratio for vehicle rollover prediction. Vehicle System Dynamics **55**(11), 1748 – 1763 (2017). https://doi.org/10.1080/00423114.2017.1321773

18. Zhu, T., Yin, X., Li, B., Ma, W.: A reliability approach to development of rollover prediction for heavy vehicles based on svm empirical model with multiple observed variables. IEEE Access **8**, 89367 – 89380 (2020). https://doi.org/10.1109/ACCESS.2020.2994026
19. Zhu, T., Yin, X., Na, X., Li, B.: Research on a novel vehicle rollover risk warning algorithm based on support vector machine model. IEEE Access **8**, 108324 – 108334 (2020). https://doi.org/10.1109/ACCESS.2020.3001306
20. Zhu, T., Zong, C., Wu, B., Sun, Z.: Rollover warning system of heavy duty vehicle based on improved ttr algorithm. Jixie Gongcheng Xuebao/Journal of Mechanical Engineering **47**(10), 88 – 94 (2011). https://doi.org/10.3901/JME.2011.10.088