

Longitudinal Dynamic Modeling and Driving Cycle Tracking Control of an Electric-Driven Vehicle by Means of MATLAB/Simulink/Simscape

András Szántó^{1*}, Sándor Hajdu¹, Krisztián Deák¹

¹ Department of Mechanical Engineering, Faculty of Engineering, University of Debrecen, H-4028 Debrecen, Óttemető utca 2–4., Hungary

* Corresponding author, e-mail: andras.szanto@eng.unideb.hu

Received: 10 April 2020, Accepted: 08 July 2020, Published online: 28 January 2022

Abstract

MATLAB, Simulink, and Simscape are market-leading products in Model-Based Design (MBD). Applying acausal and causal modeling methods to model the physical plant and control algorithms of mechatronic systems results in high-fidelity virtual prototype models that can be used for Model-in-Loop (MiL) development. In this paper, an electric-driven vehicle is modeled, which can execute various driving cycle inputs. PID controllers are used in order to get the appropriate plant inputs. Idealized anti-lock braking system (ABS) and traction control system (TCS) algorithms provide robustness when the driving cycle input is near-infeasible. The control algorithm is validated in the cases of feasible and infeasible driving cycle inputs.

Keywords

electric vehicle, longitudinal driving cycle, model-based design, acausal modeling, model-in-loop development

1 Introduction

In this article, the Model-Based Design (MBD) methodology (Friedman, 2006; Mahapatra et al., 2008; Jensen et al. 2011; Wan et al. 2015) is applied to model an electric-driven vehicle, which can execute various driving cycle inputs. This work can be the basis of future work with regards to the analysis and optimization of electric-driven light vehicles with real driving cycle inputs. However, this article is limited only to the presentation of the modeling methods of such vehicles and driver controllers (Sections 2 and 3), the verification of the vehicle model, and the validation of the control algorithms are also presented (Section 4).

In MBD, it is preferred to use acausal modeling methods (Kofránek et al., 2008; Dizqah et al., 2013) for the modeling purpose of the physical plant, and causal methods to model the control algorithms. The modeling environment used in this article is Simulink (for causal modeling), which has Simscape built-in for acausal modeling. Acausal, component-based environments provide reliable sources for physical modeling in various physical domains: apart from high-standard built-in components, custom components can be created to broaden the possibilities of model creation. The lumped-element models can be solved numerically by variable-step or fixed-step ODE/DAE solvers.

Besides the continuous physical plant, discrete or continuous control algorithms can be modeled. MBD has several development phases: the main phase for conceptual design is called the Model-in-Loop phase. In this phase, the system is analyzed, optimized and validated by the means of modeling and simulation.

In the case of vehicle dynamics modeling, it is important to select the most appropriate vehicle model for the application. There are detailed/truth vehicle models (Sayers and Han, 1996; Szántó and Hajdu, 2019) which are multibody systems that can capture all details of the real dynamics of vehicles, road unevenness, and other environmental factors. On the other hand, simplifications can be performed in the longitudinal, lateral or vertical directions of the vehicle body, the irrelevant degrees of freedom for a specific application can be excluded from the model. In the case of powertrain and energy management analysis (Pérez et al., 2006; Mahapatra et al., 2008), simplified longitudinal vehicle models are considered to be a sufficient approximation of vehicle dynamics.

For the controller validation, simple driving cycles are used in this article (Section 4). World-harmonized Light-duty Test Procedure (WLTP) driving cycle is presented in

the article (Demuynck et al., 2012), which is the standard applied driving cycle in reality. References (Maurice and Pacejka, 1997; Pacejka, 2012) are cited in connection with tire modeling. In (Szíki et al., 2020) a series-wound DC motor simulation is presented.

2 Physical modeling

In this section, the physical modeling of the vehicle is presented. Physical modeling was performed in Simscape environment, using acausal, component-based methods. The documentation of all utilized components is available, the source codes of the components may or may not be visible. The equations of these components are not intended to be presented in this article, because they are not new and custom components. Constant parameters and the list of utilized components are presented in Table 1 and Table 2. Fig. 1 shows the full model, Fig. 2 presents a detailed view of the physical plant model.

2.1 Vehicle dynamics model

The vehicle model (Fig. 2, Vehicle Dynamics Model) is a simplified, longitudinal bicycle model. The Vehicle Body component has one translational freedom, the rear and the front wheels each have one rotational freedom, which results in a 3 degrees of freedom vehicle model. Power loss effects like drag and rolling resistance can be customized using the parameters of well-known models, as well as geometry and mass properties of the vehicle body and the wheels. There are two inputs of the vehicle body with regards to environmental factors: longitudinal wind speed and incline angle of the road.

The μ (slip) characteristics of tires are important in the case of tire modeling, where μ is the coefficient of friction, and the slip is the relative motion between the tire and the road surface. Several tire models exist, which describe these characteristics mathematically, these are empirical or semi-empirical formulas. In the Simscape Tire component, Pacejka's Magic Formula (Pacejka, 2012) is used, which is valid for steady-state. Tires have transient behavior called relaxation length (Maurice and Pacejka, 1997), this property is modeled with an additional freedom between the tire and the rim, using spring and damper between them (so more precisely, the vehicle has 5 DOF).

2.2 Drivetrain model

The drivetrain model of the vehicle (Fig. 2, Drivetrain) is relatively simple, because the gear ratio is constant, without gear shifts. The rear wheel is driven. In the case of

Table 1 Constant parameters related to the physical plant

DC motor parameters	
$L = 1e-4$ [H]	Inductivity of the DC motor
$R = 0.011$ [Ohm]	Resistance of the DC motor
$k_v = 0.225$ [V/(rad/s)]	Back-EMF constant
Vehicle body parameters	
$m = 750$ [kg]	Mass of the vehicle
$l_r = 1.6$ [m]	Horizontal distance from center of gravity (CG) to rear axle
$l_f = 1.4$ [m]	Horizontal distance from CG to front axle
$h = 0.5$ [m]	CG Height above ground
$A = 3$ [m ²]	Frontal area
$c_d = 0.4$ [1]	Drag coefficient
Wheel model parameters	
$r_w = 0.16$ [m]	Radius of wheels
$I_w = 0.6$ [kg*m ²]	Wheel inertia
$k_w = 2e5$ [N/m]	Longitudinal stiffness (relaxation length model)
$c_w = 1e5$ [N/(m/s)]	Longitudinal damping (relaxation length model)
$F_{z1} = 3000$ [N]	Rated vertical load
$F_{x1} = 2300$ [N]	Peak longitudinal force at rated load
$s_1 = 0.1$ [1]	Slip at peak force at rated load
Driveline parameter	
$i = 1$ [1]	Gear ratio

Table 2 Simscape Domains and Components used

Simscape Physical Domains		
Electrical	Blue	
Translational Mechanical	Dark green	
Rotational Mechanical	Light green	
Simscape Components	Documentation	Source Code
Vehicle Body	yes	no
Tire (Magic Formula)	yes	no
Simple Gear	yes	no
Rotational Electromechanical Converter	yes	yes
Inductor	yes	yes
Resistor	yes	yes
Controlled Voltage Source	yes	yes
Ideal Torque Source	yes	yes
PS Lookup Table (1D)	yes	yes

multiple gears, an idealized shifting algorithm could be modeled based on the fact that it can be determined when it is beneficial to shift gears (although in the real application, fuzzy logic is used to prevent high-frequency up and down shifting).

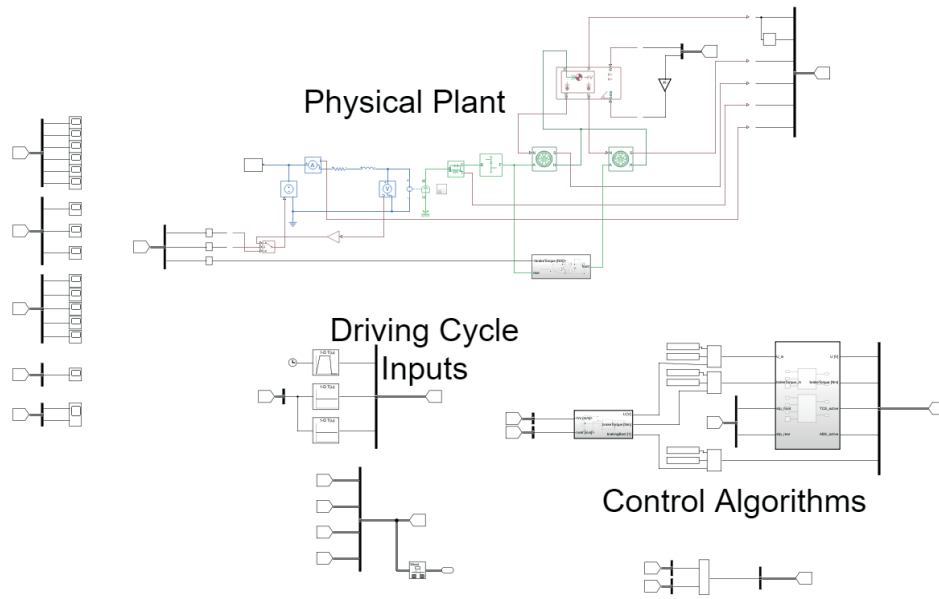


Fig. 1 Full model: Simscape Physical Plant, and Simulink Control Algorithms

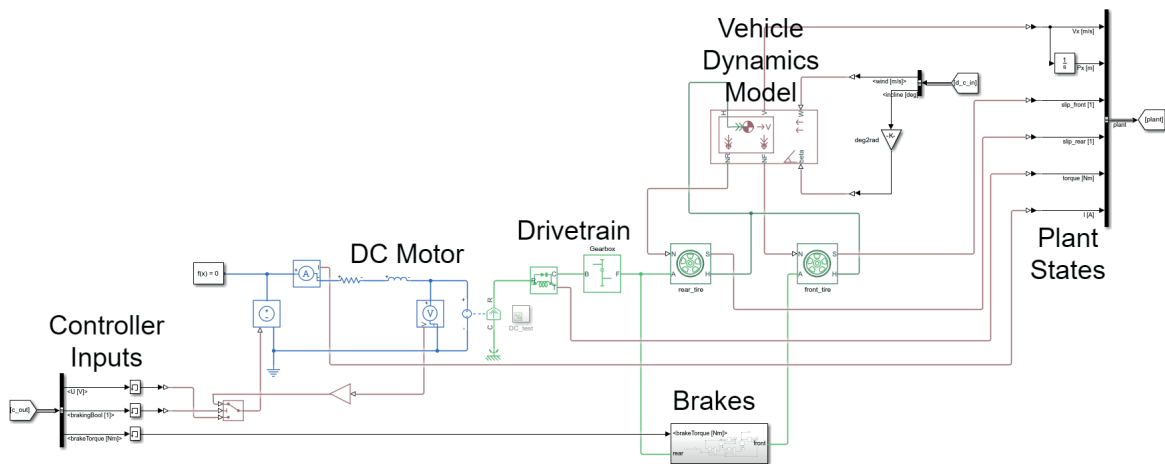


Fig. 2 Physical Plant model in Simscape

2.3 Brake model

The braking input of the vehicle (Fig. 2, Brakes) is directly the brake torque, which is equal for both wheels. A simple stick-slip law is defined in a Lookup Table (LUT) component, which is a torque(velocity) function. In the stick region (when the velocity is between a – and + threshold region around zero) the function is linear, outside of this region the linear function is saturated. The output of the LUT is scaled according to the torque input and applied to the wheel. There are two independent LUTs for the two wheels, only the scalar input is common. The torque input has a limit.

2.4 Model of the electric motor

A simple, permanent-magnet DC motor is modeled for the vehicle (Fig. 2, DC Motor), with linear characteristics. The input of the motor is the applied voltage, which has a limit

according to the battery. This is a simplified, averaged model of the electric drive, various hardware elements are necessary in the field of power electronics to apply the desired voltage in the reality, which is not in the scope of this article.

3 Models of control algorithms

In this section, the control algorithms are presented, the aim of which is to execute the desired driving cycle input for the vehicle. The discrete algorithms are modeled in Simulink. The physical plant has two control inputs: voltage of the DC motor, and brake torque. All – even not directly measurable – states of the physical plant are available to create algorithms with the best reference tracking capabilities because the utilization of the control algorithms is limited only for the simulation. In Fig. 3 the overview of

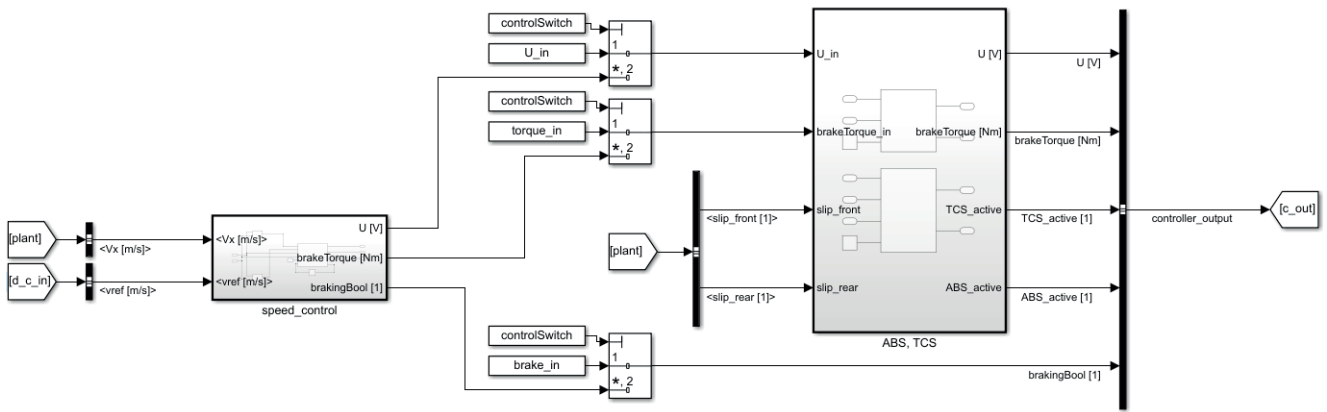


Fig. 3 Simulink Control Algorithms

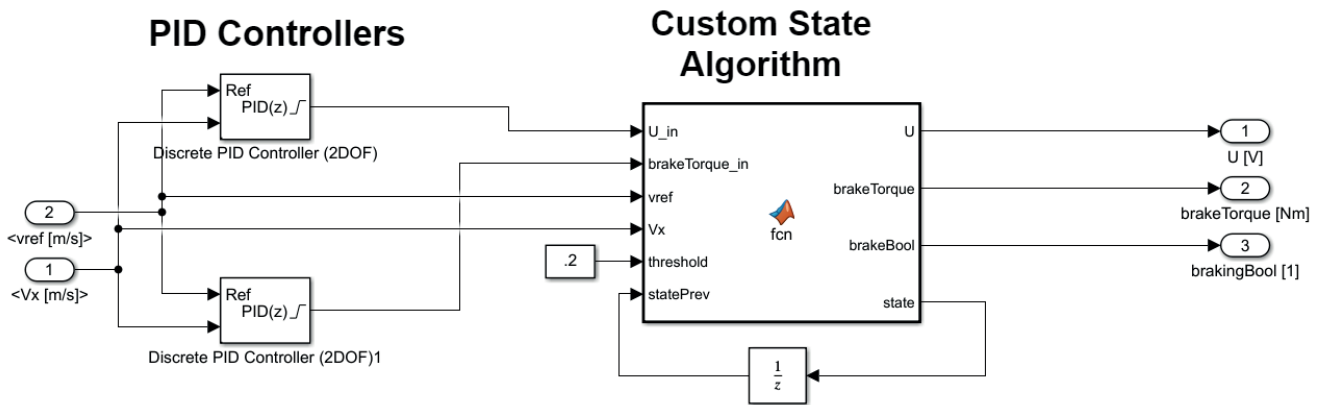


Fig. 4 Content of "speed_control" subsystem: driver model with PID controllers and custom algorithm

the control algorithms can be seen, in Fig. 4 and Fig. 5 the contents of "speed_control" and "ABS, TCS" subsystems are presented.

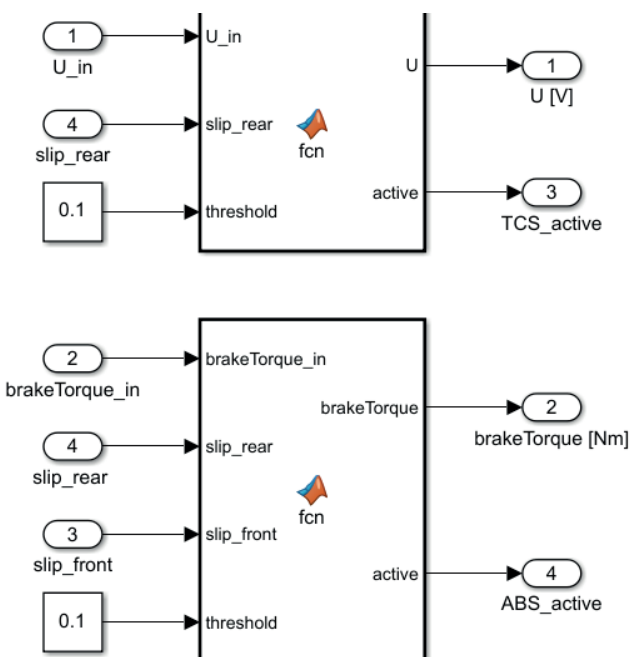


Fig. 5 Content of "ABS, TCS" subsystem: TCS and ABS algorithms

3.1 Driver model with PID controllers

There are two PID controllers (Fig. 4) modeling the driver's behavior: one for the voltage, and one for the brake torque input of the plant. The input of the controllers is the longitudinal velocity, and the reference is the driving cycle velocity in time (Fig. 6). The outputs of the controllers are saturated according to hardware properties (Table 3, U_{max} and T_{max}). The PID controllers are evaluated in every step, a custom algorithm (Fig. 4, Custom State Algorithm) decides which state the vehicle is in (acceleration or deceleration), and applies the PID outputs to the plant according to the state. There is a velocity threshold value (Table 3, v_{th}) to perform the state change. The PI gains are adjusted manually, the D gain is zero.

Table 3 Constant parameters related to the control algorithms

Control algorithm parameters	
$T_{sample} = 1e-3$ [s]	Sample time of the control algorithm
$s_{th} = 0.1$ [1]	Slip threshold for ABS/TCS
$v_{th} = 0.2$ [m/s]	Velocity difference threshold for state change
Control outputs, plant inputs	
$U_{max} = 40$ [V]	Maximum DC motor voltage
$T_{max} = 500$ [Nm]	Maximum brake torque

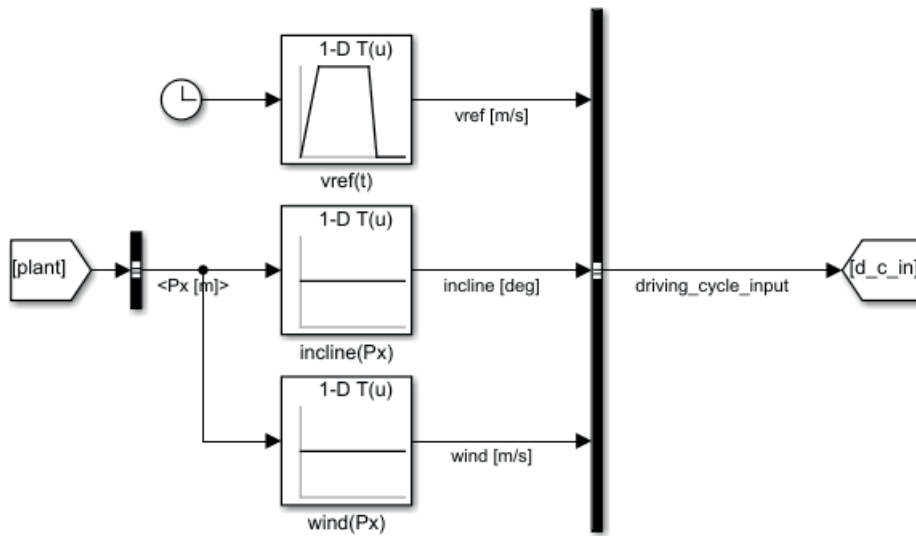


Fig. 6 Driving Cycle Inputs

In the case of braking, the applied voltage is equal to the back EMF voltage of the DC motor so that there is no current flow. In this case, there is no braking effect of the DC motor, and there is no kinetic energy recovered by it. The hardware considerations and optimization of an energy recovery system, and its combination with the braking system are not in the scope of this article.

3.2 Ideal ABS and TCS algorithms

When the velocity(time) function of the driving cycle is too steep, there could be cases when the slip values are out of the adhesion limit (according to the tire model). In this case, the applied voltage/brake torque is released, which resulted by an idealized anti-lock braking system (ABS) and traction control system (TCS) model (Fig. 5). The model is idealized because there are no hardware considerations in it. Compared to the original operation (Figs. 7, 8), this operation results in oscillations in the applied voltage, brake torque, and slip values (Figs. 9, 10). The slip thresholds (Table 3, s_{th}) are according to the tire model. Fully infeasible driving cycle input is unlikely in the real application. However, in the case of near-infeasible driving cycles, ABS and TCS provide robustness, preventing the wheels from spinning or locking, thus providing better tracking of the reference.

4 Simulation results

In this section, numerical solutions of the model are presented. First, the plant model with constant inputs was verified by observing and interpreting its states, searching for mistakes in the model building process. After the verification of the plant, the control algorithm was designed and validated. In the validation process of the algorithm,

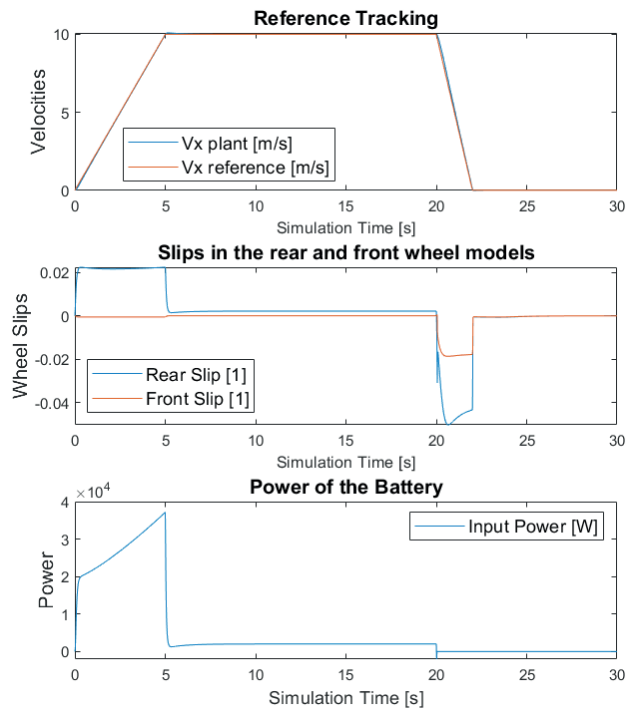


Fig. 7 Feasible driving cycle, plant states

it is necessary to check that if the requirements are met (in this case, tight reference tracking, minimal overshoot, and smooth algorithm output values). In this section, the results of the control algorithm validation process are presented, in the case of a feasible and an infeasible driving cycle input.

4.1 Feasible driving cycle input

Feasible driving cycle input means that the velocity reference can be achieved within the stick tire slip region, ABS and TCS are not activated. The results are presented in Figs. 7 and 8.

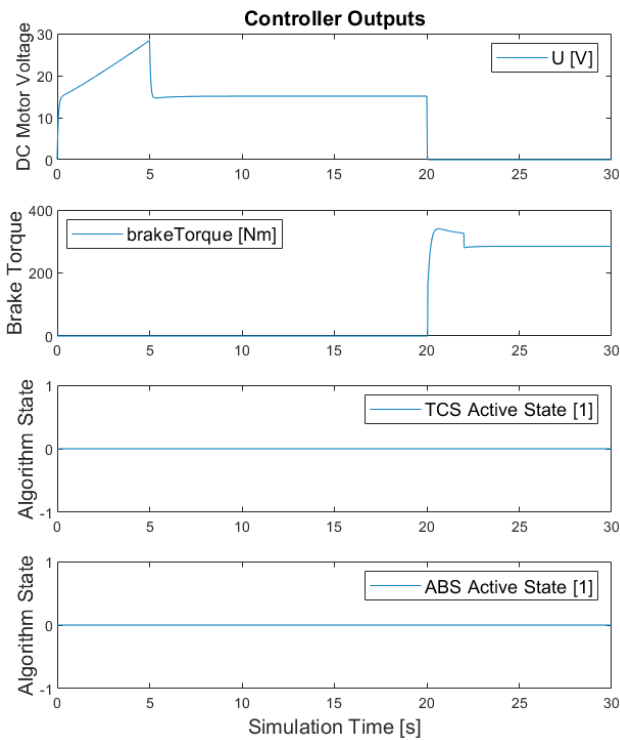


Fig. 8 Feasible driving cycle, control algorithm states

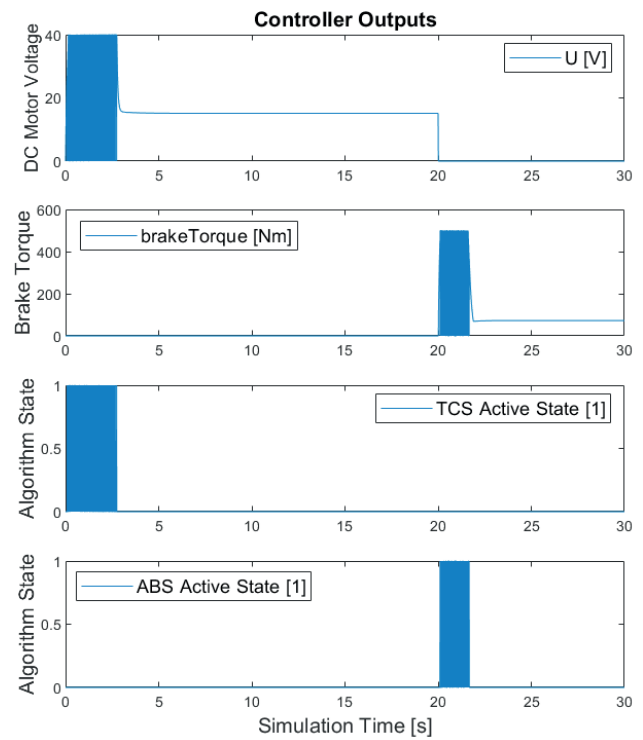


Fig. 10 Infeasible driving cycle, control algorithm states

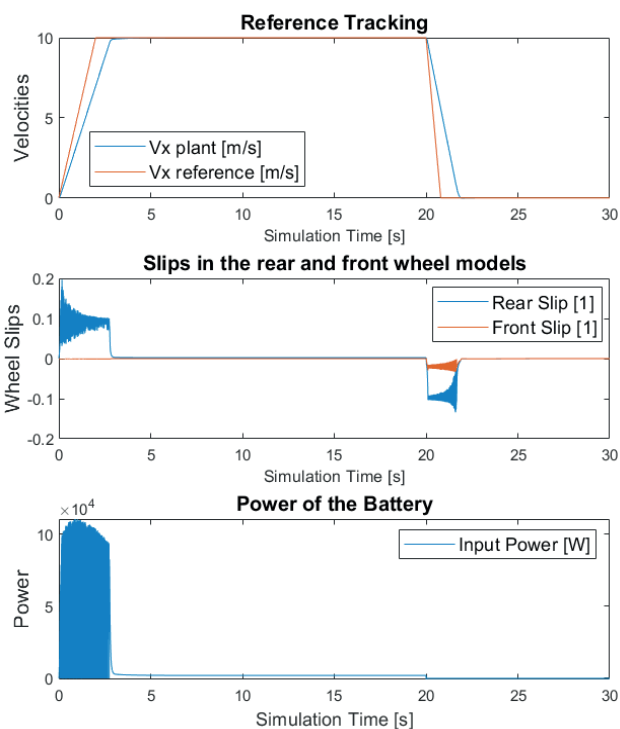


Fig. 9 Infeasible driving cycle, plant states

4.2 Infeasible driving cycle input

An infeasible driving cycle input cannot be tracked, because the friction coefficient is limited: it has its maximum at the slip value of the adhesion limit (0.1 in this case).

The idealized ABS-TCS algorithm is activated above this value (described in Section 3.2). The results are presented in Figs. 9 and 10.

5 Conclusion

Modeling environments provide tools to help to solve complex engineering problems by the means of modeling and simulation, which are the main methods of MBD. The modeling of the physical plants is performed by acausal modeling methods, and the development of the control algorithms is performed by causal ones. Besides the modeling environment, a programming environment (MATLAB) is used for analysis, optimization, and data management purposes.

The objective of this article is completed, the created models meet the requirements. The vehicle model is verified, the controller for the driving cycle input is validated. The created methods are ready to be used for powertrain optimizations with real data.

Acknowledgment

The research was financed by the Thematic Excellence Programme of the Ministry for Innovation and Technology in Hungary (ED_18-1-2019-0028), within the framework of the (Automotive Industry) thematic programme of the University of Debrecen.

References

- Demuynck, J., Bosteels, D., De Paepe, M., Favre, C., May, J., Verhelst, S. (2012) "Recommendations for the new WLTP cycle based on an analysis of vehicle emission measurements on NEDC and CADC", *Energy Policy*, 49, pp. 234–242.
<https://doi.org/10.1016/j.enpol.2012.05.081>
- Dizqah, A. M., Maheri, A., Busawon, K., Fritzson, P. (2013) "Acausal Modelling and Dynamic Simulation of the Standalone Wind-Solar Plant Using Modelica", In: 2013 UKSim 15th International Conference on Computer Modelling and Simulation, Cambridge, UK, pp. 580–585.
<https://doi.org/10.1109/UKSim.2013.145>
- Friedman, J. (2006) "MATLAB/Simulink for Automotive Systems Design", In: Proceedings of the Design Automation & Test in Europe Conference, Munich, Germany, pp. 1–2.
<https://doi.org/10.1109/DATE.2006.243988>
- Jensen, J. C., Chang, D. H., Lee, E. A. (2011) "A Model-Based Design Methodology for Cyber-Physical Systems", In: 7th International Wireless Communications and Mobile Computing Conference, Istanbul, Turkey, pp. 1666–1671.
<https://doi.org/10.1109/IWCMC.2011.5982785>
- Kofránek, J., Mateják, M., Privitzer, P., Tribula, M. (2008) "Causal or acausal modeling: labour for humans or labour for machines", [pdf] Technical Computing Prague 2008: 16th Annual Conference Proceedings, Prague, Czech Republic, pp. 1–16. Available at: <http://www.physiome.cz/references/tcp2008.pdf>
- Mahapatra, S., Egel, T., Hassan, R., Shenoy, R., Carone, M. (2008) "Model-based design for hybrid electric vehicle systems", *SAE Technical Papers*, 2008(724), Paper number: 2008-01-0085.
<https://doi.org/10.4271/2008-01-0085>
- Maurice, J. P., Pacejka, H. B. (1997) "Relaxation length behaviour of tyres", *Vehicle System Dynamics*, 27(Sup001), pp. 339–342.
<https://doi.org/10.1080/00423119708969668>
- Pacejka, H. B. (2012) "Tire and Vehicle Dynamics", Butterworth-Heinemann, Oxford, UK.
<https://doi.org/10.1016/C2010-0-68548-8>
- Pérez, L. V., Bossio, G. R., Moitre, D., García, G. O. (2006) "Optimization of power management in an hybrid electric vehicle using dynamic programming", *Mathematics and Computers in Simulation*, 73(1–4), pp. 244–254.
<https://doi.org/10.1016/j.matcom.2006.06.016>
- Sayers, M. W., Han, D. (1996) "A generic multibody vehicle model for simulating handling and braking", *Vehicle System Dynamics*, 25(sup1), pp. 599–613.
<https://doi.org/10.1080/00423119608969223>
- Szántó, A., Hajdu, S. (2019) "Vehicle Modelling and Simulation in Simulink", *International Journal of Engineering and Management Sciences*, 4(1), pp. 260–265.
<https://doi.org/10.21791/IJEMS.2019.1.33>
- Szíki, G. Á., Sarvajcz, K., Szántó, A., Mankovits, T. (2020) "Series wound DC motor simulation applying MATLAB Simulink and LabVIEW control design and simulation module", *Periodica Polytechnica Transportation Engineering*, 48(1), pp. 65–69.
<https://doi.org/10.3311/PPtr.12908>
- Wan, J., Canedo, A., Al Faruque, M. A. (2015) "Functional Model-Based Design Methodology for Automotive Cyber-Physical Systems", *IEEE Systems Journal*, 11(4), pp. 2028–2039.
<https://doi.org/10.1109/jsyst.2014.2387487>