AN APPROACH TO CREATING A SIMPLE DIGITAL TWIN FOR OPTIMIZING A SMALL ELECTRIC CONCEPT VEHICLE DRIVETRAIN

Tamás Dóka
Péter Horák, PhD
Department of Machine and Product Design
Faculty of Mechanical Engineering
Budapest University of Technology and Economics
1111, Műegyetem rkp. 3, Budapest, Hungary
Email: doka.tamas@gt3.bme.hu

KEYWORDS

3D Model-Based Simulation; Digital Twin; Electric Vehicle Drivetrain

ABSTRACT

Since modeling and simulation are integral tools in engineering, the question is not *if* they should be used in a design process, but rather *how* they should be used to deliver the best solutions. The objective of this paper is to outline an approach to creating a simple Digital Twin for a small electric vehicle drivetrain utilizing only parametric 3D CAD models, widely used simulation tools and some programming libraries. First, the concept of the Digital Twin, its benefits, then the possibilities of using Generative Design are briefly introduced, afterwards electric vehicles' advantages are reviewed. In an example project the properties and opportunities of the 3D CAD- and simulation models are demonstrated. Finally, future improvements and automated optimization opportunities are discussed.

INTRODUCTION

Before today's modeling and simulation technologies have been emerged, designing a system and ensure its proper behavior was expensive and time-consuming. The only way to test a system in operation was to build it physically and to subject it to effects and impacts that the designers thought would be necessary (Grieves and Vickers 2017). In the second half of the 20th-century, Computer-Aided Design (CAD) software solutions enabled to create different variations of a system relatively easy. Finally, with a Digital Twin, in theory, we could analyze any product's behavior in different environments without having the physical representation itself.

With the recently available computational power and cloud-based services, the use of complex simulations and detailed virtual prototypes are no longer privileges of the biggest companies, but they can be created and also run on personal computers. In this paper, an example is shown how a simple Digital Twin of a product can be set up and prepared to be used for optimization, using only open-source or student license software solutions. The subject of this example is a small electric concept car's drivetrain; therefore, the properties of a battery electric vehicle are briefly introduced.

Digital Twins

The term Digital Twin (DT) has many slightly different definitions which are slowly changing through the years. "The Digital Twin is a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level. At its optimum, any information that could be obtained from inspecting a physical manufactured product can be obtained from its Digital Twin." (Grieves and Vickers 2017). In that way, a DT could exist without physical representation and could serve as a substitute for real prototypes.

In applications like aerospace or automotive industries, any modification to the product could generate unforeseen consequences to the whole system, so these effects should be adequately modeled and simulated before applying them to the system in operation (Goossens 2017). Not long ago, Digital Twins have been developed in a bottom-up philosophy after the real system was specified. The Digital Twin concept was used to create a virtual environment where a detailed simulation model is running. Based on that model and sensor information from the real physical world, the behavior of the real-world twin could be estimated. This approach has many case studies where the existing system is monitored, the Digital Twin could easily detect occurring problems and practical solutions could be calculated to solve these issues.

Traditionally in the conceptual phase, possible design alternatives were explored by engineers, which required a lot of experience and time. CAD models and simulations were only involved in the process after the design space was narrow enough to analyze, validate, and fabricate the design (Khan and Awan 2018). A detailed model for a product or process is not only capable of examining the system based on real-life data, but it can be leveraged in the concept design phase to define, test, and evaluate different variants of the system. In this method, the virtual model is not only used for diagnosis but to find out which version of the system should be built in the first place. It is more common today to utilize methods that offer a more standardized description of the models (Rodič 2017), allowing to use optimization algorithms acting directly on the digital model by modifying its parameters and comparing the simulation results. The Digital Twin technology, combined with novel optimization algorithms using artificial intelligence, could generate feasible systems that not only correspond with the requirements but are optimal in the prescribed aspects.

Generative Design

It is usually hard to say how a specific parameter will affect the whole system without knowing the system itself. In many fields during the concept design phase, even experts are using best practices to set up the initial boundaries of the product. This method could lead to sub-optimal solutions even if very precise optimization takes place in later stages.

Generative Design systems are using parametric design, optimization and simulation techniques, which allow engineers to iterate through a large number of design alternatives. Taking a problem definition as input Generative Design systems could create a set of optimal solutions for the given problem (Khan and Awan 2018). An example is shown in Figure 1.

Commercially available Generative Design systems are promising tools for creating complex mechanical parts for a given load- and constraint set using artificial intelligence and topology optimization. This method assumes that the surroundings and functions of the element are known. However, the constraints and loads acting on the part usually depend on other members of the system, so these generated solutions are only suitable for situations where all other components of the system remain the same. Still, the idea that engineers should only carefully define the functions and coarse boundaries of a system and artificial intelligence could do the rest of the work could create previously unimaginable new inventions.

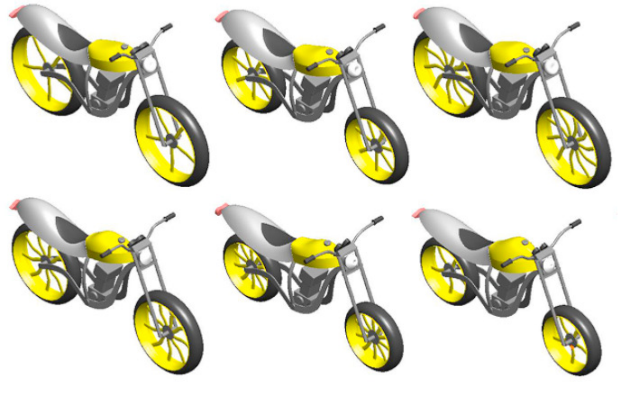

Fig. 1. Generated Design Variations of a Motorcycle (Khan and Awan 2018)

Generative Design combined with Digital Twin technology could allow the system-level optimization where components are heavily co-dependent, such as in electric vehicles.

Small Battery Electric Vehicle Concept

As the present trend suggests, electric vehicles are likely to replace internal combustion engine (ICE) vehicles in the near future (Un-Noor et al. 2017). This trend could be explained by electric vehicles (EVs) being more environmentally friendly, quiet, easy to operate, require less money for fuel and also, they provide instant torque from the startup. EVs are the unquestionably better choice for urban transport, but for longer journeys, two more factors come into question: power and range. Providing a bigger range needs more batteries; therefore, the overall mass of the vehicle grows; thus, the power consumption increases, limiting the achievable range of the vehicle. Finding the right size and arrangement of the electric powertrain components is not as evident as it is in ICE vehicles, because power can be transmitted through electrical wires, enabling to create very different configurations. Moreover, most of the components' parameters are determined by other parts (Un-Noor et al. 2017); thus, optimization becomes even more complicated.

In this paper, an approach for creating a Digital Twin is demonstrated on a small electric vehicle concept. The powertrain of an electric vehicle consists of an electrical and a mechanical subsystem; thus, it is necessary to accurately model and analyze them together to get a real insight into the vehicle's dynamic behavior (Park et al. 2014).

APPROACH TO CREATING A SIMPLE DIGITAL TWIN

As mentioned before, a Digital Twin is basically a set of information about an entity permitting to analyze it from different aspects accurately in a virtual environment. Although every area (mechanical engineering, electrical engineering, etc.) deals with the same product, each of these areas approach the parts that make up these components in a different way (Grieves and Vickers 2017). At present, computers with relatively high computing capacity are affordable, and many modeling, simulating, and optimizing tools and programs are available. Thus the opportunity of using Digital Twins is at hand for smaller businesses and smaller projects too. The design process could be even more improved if we could integrate the available software solutions which we are using, allowing the data exchange between them. Usually, the connection within these software products is not provided, so technically, it requires effort to implement such complex systems. The key to creating a consistent Digital Twin is to persist a homogeneous perspective of the information across functional boundaries. This can be realized by having an application that controls and manages data between different platforms and areas, as shown in Figure 2.

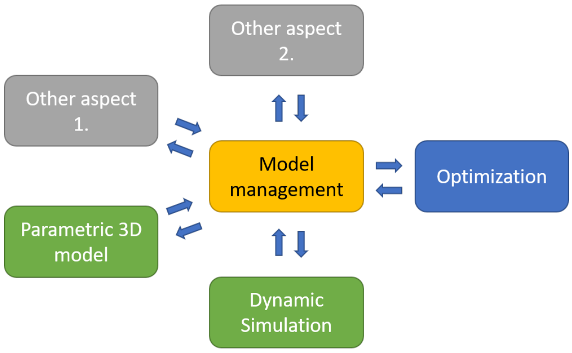

Fig. 2. Block Diagram of the Digital Twin Environment

In the example project, the vehicle was approximately modeled, focusing more on its drivetrain. A detailed, skeleton based top-down 3D model of the drive module (electric motor, fixed gear speed reducer and differential) was modeled and built (see in Figure 3) for further measurements, parameter identification and testing. A simple dynamic simulation was created, using parameters from the CAD models to analyze the vehicle's longitudinal behavior and to provide a basis for optimization in the future.

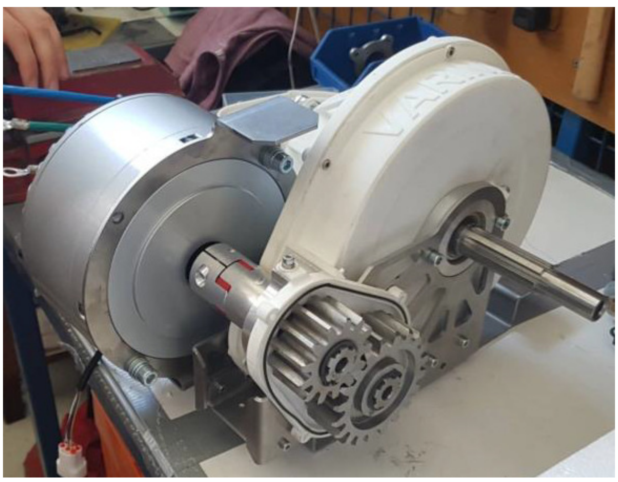

Fig. 3. Drive Module of the Small Electric Vehicle Concept

Parametric 3D model

A Computer-Aided Design (CAD) model is typically used to visualize the entity, how it will look like in its physical form. Almost every product design starts with an approximate 3D model after the main concept was laid out. These 3D models represent the product's mechanical and physical properties, such as total mass, material properties, or geometric boundaries. Today's advanced CAD systems are capable of performing many different tasks, such as FEM analysis, topology optimization, motion- and dynamic simulations. These integrated CAD systems provide pervasive solutions, but dynamic simulations usually do not require the detailed 3D representation of a product to deliver the desired results or, on the contrary only the aggregated properties of a body (mass, moment of inertia, center of gravity) should be taken into consideration to avoid long computational time.

In the project the vehicle was modeled in PTC Creo 4.0, because it is a high-level CAD system equipped with a wide variety of embedded modules, which can be automated. Make use of the skeleton-based top-down approach, all of the key parameters could be modified, and the assembly regenerated through the managing algorithm.

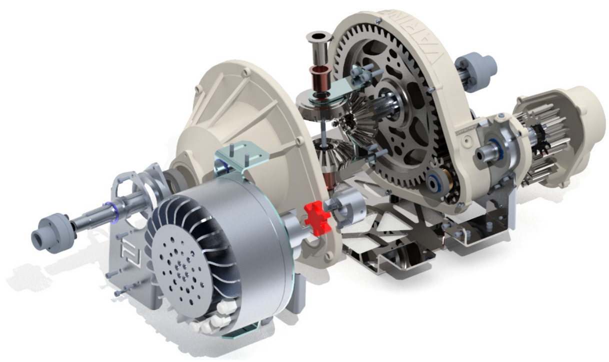

Fig. 4. Exploded View of the Drive Module

Simulation model

Using only aggregated parameters from the 3D CAD models (not the actual 3D mesh) enables us to simulate the system efficiently. Even without a detailed 3D model, simulation can be processed with approximate parameters for estimations. For example, in the early stages, the total mass of the vehicle is unknown, but the drive module could be tested with different scenarios. Later, when the actual vehicle model is available, approximate parameters could be replaced with precise ones. In the simulation model, a simplified, single mass point representation of the vehicle was implemented, which analyses only the longitudinal motion of the car. In general, there are two main approaches towards the longitudinal motion simulations of vehicles, kinematic and dynamic simulations. In case of a kinematic simulation, the actual state of the simulation components is calculated backward from a given driving condition, using the gear ratios and efficiencies to determine the required input values of the propulsion unit. In a dynamic simulation, where the calculations are forward-directed, a driver model calculates the torque demand of the vehicle from the current vehicle speed and provides the corresponding input to the propulsion unit. In the following blocks, the states of the drivetrain components are calculated, resulting in the vehicle's actual speed, which is connected to the driver model to close the simulation loop. Using MATLAB Simulink in the project, a dynamic simulation was implemented because this kind of model ensures a more realistic and accurate simulation of the vehicle's drivetrain than the kinematic approach (Winke and Bargende 2013). The simulation model is divided into three main blocks (see in Figure 5):

- *Driver module*
- *Drivetrain module*
- *Vehicle module*

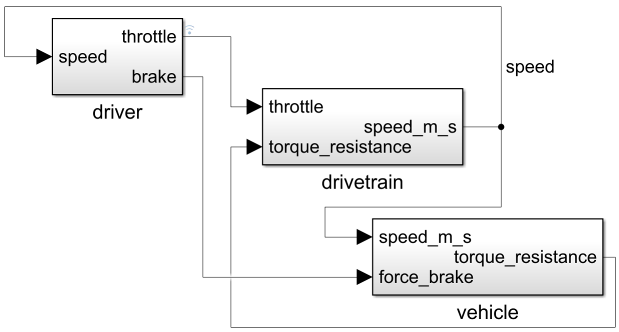

Fig. 5. The Basic Structure of the Dynamic Simulation Model

The *driver module* controls the electric motor's throttle based on the New European Driving Cycle (NEDC). If the current speed of the vehicle is lower than the desired, the module raises the throttle signal, if the opposite is true, it lowers the throttle signal, or even initiates braking.

The *drivetrain module* simulates the dynamics of the electric motor, the fixed-gear drive module, and the vehicle's wheel (see in Figure 6), using the throttle signal and the resistances acting on the vehicle as inputs to calculate the vehicle's speed. Each block calculates its inner state according to the input and output torques and angular accelerations, assuming that the system is totally stiff, and does not contain any non-linearity such as backlashes.

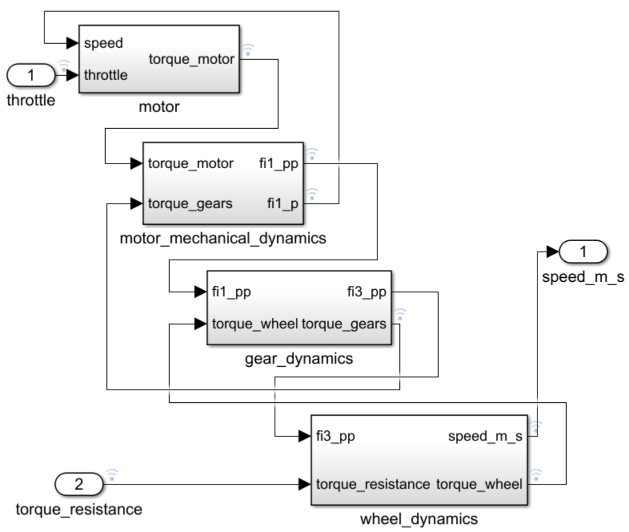

Fig. 6. Drivetrain Simulation Module's Block Diagram

Inside the *drivetrain module*, the *motor* block provides torque input to the *motor mechanical dynamics* block based on the current motor speed, the throttle signal (which can be between 0 and 1), and a look-up-table (LUT) which contains the motor's torque-speed characteristics. The LUT determines the maximal motor torque at the given speed; finally, this value is multiplied by the throttle signal value. This method enables to create the needed motor torque between zero and the nominal maximum torque of the motor for each motor speed. In the *motor mechanical dynamics* block, the angular acceleration of the motor is calculated as shown in Equation (1):

$$\ddot{\varphi}_m(t) = \frac{\tau_{motor}(t) - \tau_{gears}(t)}{J_{motor}} \quad (1)$$

where,

- φ_m is the motor's angular position,
- τ_{motor} is the motor torque,
- τ_{gears} is the input torque for the gearbox,
- J_{motor} is the moment of inertia of the motor's rotating parts

Integral of the angular acceleration over time gives the motor speed, which is connected to the *motor* block.

The *gear dynamics* block consists of the input-, intermediate- and differential shaft assemblies, which are connected together by gear pairs (see in Figure 7).

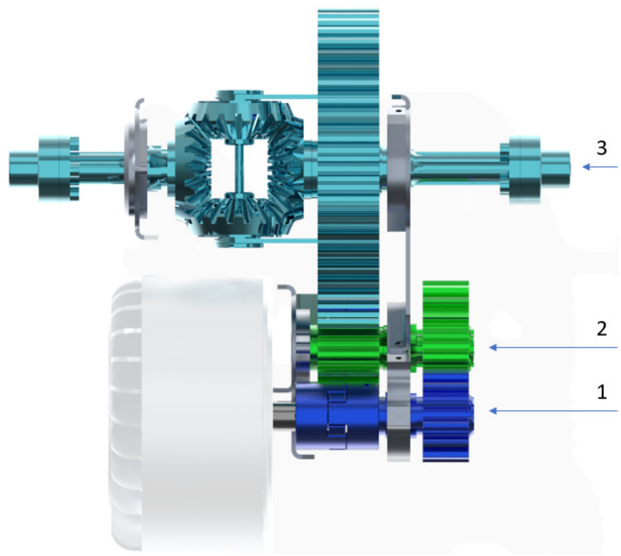

Fig. 7. 3D Model of the Fixed-gear Drive Module: Input- (1), Intermediate- (2), Differential Shaft (3) Assemblies

Since the motor is connected to the input shaft assembly, the angular acceleration of these two is equal. To ensure this the input torque for the gearbox (τ_{gears}) is calculated as shown in Equation (2):

$$\tau_{gears}(t) = \tau_{l1} + J_1 \cdot \ddot{\varphi}_m(t) + \frac{\tau_{l2} + J_2 \cdot \dfrac{\ddot{\varphi}_m(t)}{i_{1,2}} + \dfrac{\tau_{l3} + J_3 \cdot \dfrac{\ddot{\varphi}_m(t)}{i_{1,2} \cdot i_{2,3}} + \tau_{wheels}(t)}{i_{2,3}}}{i_{1,2}} \quad (2)$$

where,

- τ_{wheels} is the input torque for the wheels
- τ_{lj} is an estimated torque loss (e.g. from bearings) for the j-th shaft assembly,
- J_j is the moment of inertia of the j-th shaft assembly,
- $i_{j,k}$ is the gear ratio of the gear pair between the j-th and the k-th shaft assemblies.

The *wheel dynamics* block represents the connection between the drivetrain's rotational- and the vehicle's longitudinal movement. The torque transferred through the gearbox to the wheels is compensated by static friction on the ground; thus, the vehicle's center of gravity starts to accelerate in a longitudinal direction. Assuming that the contact points of the wheels are not sliding on the ground, the relation between the longitudinal and rotational measures can be expressed as shown in Equation (3) - (5):

$$a(t) = \ddot{\varphi}_{wheels}(t) \cdot r \tag{3}$$

$$m = \frac{J_{red}}{r^2} \tag{4}$$

$$F_P(t) = \frac{\tau_{red}(t)}{r} \tag{5}$$

where,

- a is the vehicle's longitudinal acceleration,
- φ_{wheels} is the wheels' angular position,
- r is the wheel radius,
- m is the vehicle's mass,
- J_{red} is the reduced moment of inertia of the vehicle's mass,
- F_P is the pulling force,
- τ_{red} is the torque that effectively accelerates the vehicle in longitudinal direction.

When the vehicle is moving straight, the wheels are rotating together at the same speed as the differential shaft assembly. Their angular acceleration can be calculated from the motor's angular acceleration (see in Equation (6)). Thus the torque acting on the wheels (τ_{wheels}) is responsible for the wheels' angular acceleration, the vehicle's longitudinal acceleration, and the compensation of resistance torques (see in Equation (7)).

$$\ddot{\varphi}_{wheels}(t) = \frac{\ddot{\varphi}_m(t)}{i_{1,2} \cdot i_{2,3}} \tag{6}$$

$$\tau_{wheels}(t) = (J_{wheels} + J_{red}) \cdot \ddot{\varphi}_{wheels}(t) + \tau_{res} \tag{7}$$

where,

- J_{wheels} is the moment of inertia of the wheels,
- τ_{res} is the resultant torque from resistance forces acting on the vehicle.

Integral of Equation (3) over time gives the vehicle's longitudinal speed, which serves as an input in the *vehicle module* to calculate resistances, and also in the *driver module* to determine the throttle and brake signals.

The separation of the blocks allows us to change the current model into more detailed versions, add more gear stages, and to read out the inner states of each component throughout the simulation.

In the *vehicle module*, the resistance forces acting on the vehicle are simulated (see in Figure 8).

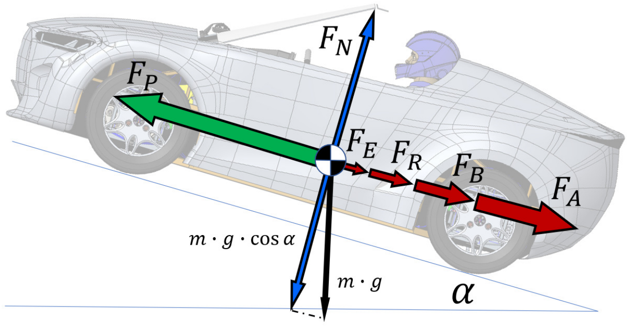

Fig. 8. Forces Acting on the Vehicle, at a Constant Velocity Equilibrium State

- F_P : pulling force from τ_{red},
- F_A : force from air resistance,
- F_R : force from rolling resistance,
- F_E : force from elevation on the slope,
- F_B : force from braking,
- F_N : normal force,
- g : gravitational acceleration,
- α : angle of slope.

For every equilibrium speed, the pulling force is equal to the summation of the resistance and elevation forces. If the pulling force is greater than the resistance forces, the vehicle is accelerating; else, it is decelerating.

Using this model a wide variety of valuable information can be extracted from the simulation results, such as the required motor torque to meet the prescribed speed profile (see in Figure 9) or the energy consumption during the examined time. This information can serve as input to other applications, which can check how good the final concept is. Modification of the parameters of mechanical parts is essential in the design process. Changes could be propagated through the simulation, and based on the results, optimal values could be calculated for the initial parameters. If we manipulate data in 3D models, a managing algorithm should keep track of the evaluated (or shared) parameters after the models are updated, to keep the Digital Twin consistent across CAD systems, simulation programs, or any other platforms.

Managing algorithm

Usually, data exchange between different applications from different fields (e.g. 3D modeling, simulation, optimization) is not provided. It requires some programming skills to extract data from one in a form that is useful to the other. Each of these software products rely heavily on graphic interfaces, however, if the models are appropriately set up, both of them can be managed from a third program. This managing program could enable the parameter optimization of the product; in our case, the drivetrain parameters can be optimized to minimize energy consumption.

If access to the model and the simulation results is provided, an optimizer algorithm could tune parameters in order to reach optimal properties of the modeled

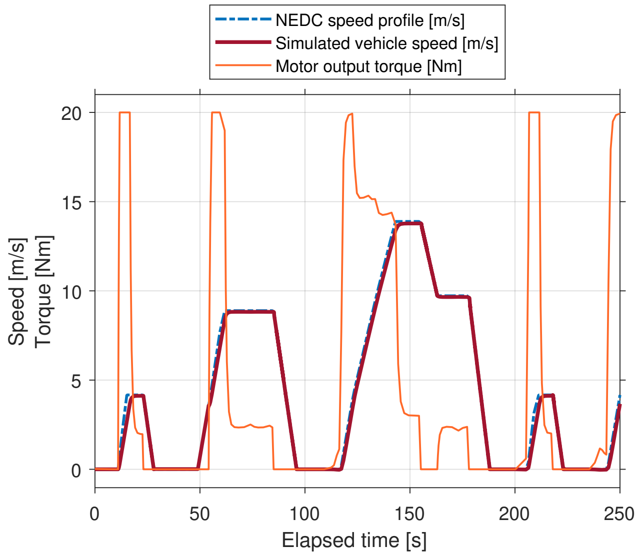

Fig. 9. Visualization of Simulation Results: Required Motor Output Torque to ensure the Prescribed Speed Profile for the Vehicle

system, however such an optimizer algorithm needs to be implemented in this project in the future. In the project, a simple Python script is used to ensure communication between Creo and Simulink.

CONCLUSIONS

The concept of the Digital Twin and Generative Design was introduced with its benefits and possibilities. Then small electric vehicles' advantages and disadvantages were reviewed. In the example project, the 3D CAD model, the simulation model, and the drive module's mathematical model were detailed.

FURTHER DEVELOPMENT

The detailed 3D model of the drive module and a simple simulation model of the powertrain is set up and ready to be utilized for measurements and tests. The model parameters and results could be accessed from a Python script, through which other different programs could be used to analyze the product from different aspects. As this managing algorithm could handle parameters consistently, it is capable of implementing an optimization algorithm to generate various optimal solutions for a defined problem. In the future, implementing a sensor network on the drive module to measure the actual torques and velocities, to identify the estimated loss parameters accurately, to test different scenarios and validate the simulation model is necessary. Based on these corrected parameters, the drivetrain module could be tested in dangerous situations virtually without damaging or breaking the physical twin. With a proper optimization algorithm (e.g. genetic algorithm), geometrical parameters could be optimized. Later, the model completed with the vehicle's suspension system could be used in a 3D physical engine for further optimization and testing.

ACKNOWLEDGEMENT

The research reported in this paper was supported by the Higher Education Excellence Program of the Ministry of Human Capacities in the frame of Artificial intelligence research area of Budapest University of Technology and Economics (BME FIKP-MI), and by the National Research, Development and Innovation Fund (TUDFO/51757/2019-ITM, Thematic Excellence Program).

REFERENCES

Goossens, P. (2017). Industry 4.0 and the power of the digital twin. Maplesoft Engineering Solutions.

Grieves, M., & Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. https://doi.org/10.1007/978-3-319-38756-7_4

Khan, S., & Awan, M. (2018). A generative design technique for exploring shape variations. *Advanced Engineering Informatics*, *38*, 712–724. https://doi.org/10.1016/j.aei.2018.10.005

Park, G., Lee, S., Jin, S., & Kwak, S. (2014). Integrated modeling and analysis of dynamics for electric vehicle powertrains. *Expert Systems with Applications: An International Journal*, *41*, 2595–2607. https://doi.org/10.1016/j.eswa.2013.10.007

Rodič, B. (2017). Industry 4.0 and the new simulation modelling paradigm. *Organizacija*, *50*, 193–207. https://doi.org/10.1515/orga-2017-0017

Un-Noor, F., Sanjeevikumar, P., Mihet-Popa, L., Mollah, M., & Hossain, E. (2017). A comprehensive study of key electric vehicle (ev) components, technologies, challenges, impacts, and future direction of development. *Energies*, *10*. https://doi.org/10.3390/en10081217

Winke, F., & Bargende, M. (2013). Dynamic Simulation of Urban Hybrid Electric Vehicles. *MTZ worldwide*, *74*, 56–63. https://doi.org/10.1007/s38313-013-0089-1

TAMÁS DÓKA is a PhD student at the Department of Machine and Product Design of Budapest University of Technology and Economics (BME), Hungary. He had received masters degree in Mechatronic Engineering in 2019. He is actively taking part in an Electric Vehicle development university project called BME Fuse, focusing on concept creating, 3D modeling, simulation and optimization. Email address: doka.tamas@gt3.bme.hu

PÉTER HORÁK, PhD is an associate professor at the Department of Machine and Product Design of Budapest University of Technology and Economics (BME), Hungary. He had received PhD degree in Mechanical Engineering in 2004. His research fields are tribology, kinematics and geometry of gears, design theory and methodology. Email address: horak.peter@gt3.bme.hu