IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 5 December 2023, accepted 27 December 2023, date of publication 2 January 2024, date of current version 10 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3349247

METHODS

Multibody Simulation Model as Part of Digital Twin Architecture: Stewart Platform Example

DOMINIK WALICA AND PETR NOSKIEVIČ
Department of Control Systems and Instrumentation, VSB—Technical University of Ostrava, 708 00 Ostrava, Czech Republic

Corresponding author: Dominik Walica (dominik.walica@vsb.cz)

This work was supported in part by the European Regional Development Fund in the Research Centre of Advanced Mechatronic Systems Project within the Operational Programme Research, Development, and Education under Grant CZ.02.1.01/0.0/0.0/16_019/0000867; and in part by the Application of Machine and Process Control Advanced Methods through the Ministry of Education, Youth and Sports, Czech Republic, under Project SP2023/074.

ABSTRACT The digital twin is considered a new and promising concept whose added value is seen mainly in end applications. However, the benefit of considering the digital twin application since the early development of the system seems not to be stressed enough. The ability to choose system components, methods, and tools can have a symbiotic effect during system integration. This work describes a development process of a Stewart platform digital twin based on its multibody simulation model created in Matlab/Simulink. Although the multibody simulation model is useful in the design phase, after adjustments and verification, it can also be reused as a virtual entity of the digital twin. This integration is enabled by the methods, tools, and system architecture selected for the purpose of including a digital twin. This is considered to be the main contribution of this paper to emerging methodologies for the development of mechatronic systems and its digital twins. However, practical integration comes with challenges that are related to the model fidelity and synchronisation of the virtual and physical entities and are important to overcome in order to employ the system in the real applications.

INDEX TERMS Digital twin, machine design, multibody simulation, Stewart platform.

I. INTRODUCTION

The development of a mechatronic system is a task that requires the involvement of disciplines of mechanical, electrical, and software engineering. The literature describes many mechatronic system development methodologies that begin with the requirements and end with the final product. Many of them are variations of the V-Model. The V-Model describes phases of machine development supported by systems engineering, modelling, and simulation tools. The modelling and simulation tools are used for the analysis and synthesis of the mechatronic system [1], [2].

An example of a mechatronic system is a generally known parallel manipulator, a Stewart platform. In Fig. 1 there is a Stewart platform developed and built at VSB - Technical University of Ostrava. It is a parallel robot with six degrees of freedom. It has a wide range of applications in load stabilisation, vibration isolation and generally in areas where spatial motion needs to be simulated in laboratory conditions for testing (e.g. earthquake, flight, driving, or oceanic wave simulation) [3], [4], [5], [6], [7]. However, one of its limitations is the relatively small workspace. Stewart platform generally consists of a base plate, actuators, a moving platform, and upper and lower joints that connect the actuators to the base and the moving platform.

When designing a Stewart platform, we are interested in its typical characteristics given by its payload, dynamics, workspace, etc. Some works describe the optimisation of a single aspect of the Stewart platform [8], [9], others focus on the design of a given application [10], [11] or present a general design [12], [13], [14]. To achieve the desired characteristics, we must find the right combination of components that will satisfy our requirements. However, there can also rise nontypical requirements related to the

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad AlShabi.

FIGURE 1. Stewart platform.

current trends of digitalisation that can be taken into account at the beginning of the development process.

One of the digitalisation trends is a digital twin. There are many definitions of the digital twin. Nevertheless, recent publications show the opinion seems to be converging to an agreement that the digital twin should consist of a physical entity, virtual entity and a bi-directional connection between them that ensures data/information exchange with the purpose of enhancing the application's value.

The virtual entity can be a representation of the physical entity or its part. Among the characteristics of the digital twin belong its purpose, the level of fidelity of the virtual entity, and the way of data/information exchange [15], [16], [17], [18], [19], [20]. The purpose of the digital twin is often related to fault diagnostics, predictive maintenance, optimisation, etc. Although the concept of digital twin is usually associated with a 3D visualisation, which can provide significant support for decision making, the publications [21], [22] show that it is not a necessary feature.

There are many publications that are concerned with the methodology for the development of digital twins [21], [23], [24] or applications [22], [25], [26] related to mechatronic systems.

The virtual entity of the digital twin can be physics-based, data-based, or a combination of both [27]. Physics-based models are usually based on simulation models and rely on knowledge of the structure and parameters of the system. Data-based models, on the other hand, often use machine learning methods based on historical and present data. The machine learning model must be trained on the data set that might be provided either by the real measurements, which implies the need for a real system or a physics-based model. Therefore, the combined approach uses physics-based models to train the data-based model [22].

Based on the reviewed literature, the methods and tools used depend on the individual use case. Among the used tools belong software platforms such as Matlab/Simulink, Ansys, Microsoft Azure, etc. [28]. Although the comparison of physics-based and data-based methods is represented in the literature [28] and [29], the comparison of individual software platforms seems to be missing. Carrying out such a study would require access to various software tools and sufficient technical prowess to apply them.

There are not many works which concern a digital twin of a Stewart platform or parallel manipulators in general. Huynh et al. [30] present a universal methodology to create serial and parallel manipulator digital twins. The presented methodology shows a uni-directional (one-way) connection from a real robot to the application that reflects its status. It also discusses the possible extension to a bi-directional connection, which would allow responding to anomalies.

On the contrary, the works that present serial manipulators are represented more frequently. The work presented by Aivaliotis et al. [21] proposes an approach to enable the implementation of digital twins of industrial robots and complex machines. It also reports on the integration of the approach in a real industrial setting where the digital twin is used for a predictive maintenance application.

The methodologies and applications of digital twins are widely represented in the literature. Most of them focus on the development or application of the digital twin concept in an existing system. However, there seems to be a lack of publications that focus on the practical development of the digital twin of the mechatronic system from the early stages of development.

Considering the digital twin since the early stages of machine development can bring potential benefits in the system integration stage. On the other hand, it also requires us to take into account considerations that might not be typical in machine development. Therefore, one of the goals of this article is to highlight and discuss these benefits and considerations.

This paper presents a development of the Stewart platform digital twin architecture from its early stages. It describes tools and methods that streamline digital twin development. The core of this work are the steps that describe the transformation of the Stewart platform multibody model to the virtual entity of the digital twin. The steps are described to the point where the possibility of bi-directional communication between both entities is presented. This paper does not present the application of the digital twin, but outlines the challenges and benefits of its future implementation.

The paper is organised as follows. Section I-A extends this section and introduces the system architecture. Section II presents the design of the kinematic structure of the Stewart platform. Section III describes the creation of the multibody dynamics model for the purpose of machine design. Section IV discusses adjustments of the multibody dynamics model for its verification and shows the results of the multibody dynamics model verification. Section V shows

TABLE 1. Stewart platform requirements.

Requirement	Description
Payload Workspace Motion dynamics	Application dependent.
Placement	Must fit into the laboratory.
Maneuverability	Must be movable around the laboratory.
Fitting	Must have mounting holes to fit a load and for an assembly.
Workstation	Must have a PC with a user control interface.
User control interface	Must be capable to issue basic motion commands.
Safety button	Must have a safety button that immediately stops the mechanism.
Workspace check	Must have a function that prevents the Stewart platform to leave its workspace.
Overload check	Must have a function that prevents overloading of the Stewart platform.
...	...

the integration of the Stewart platform multibody dynamics model into the digital twin architecture and describes the challenges for the development of digital twin functionalities. Section VI discusses the results, and Section VII summarises the work and proposes future research.

A. SYSTEM ARCHITECTURE

The design of a system begins with a definition of requirements. In our scenario, the goal was to design a Stewart platform for an experimental mechatronics lab to simulate the spatial motion of a real phenomenon (i.e., vibration). The typical constraints of this scenario are budget and time. A snippet of the general requirements of the Stewart platform itself is shown in Table 1.

We decided to not present specific requirement values, as the article is focused more on the digital twin topic than the design of the Stewart platform itself, which has already been documented in the literature.

After the initial definition of requirements, it was necessary to make the first design decisions and propose a general system architecture. To have an overview of the requirements and the proposed design, we employed the model-based systems engineering tool System Composer™ which is Matlab/Simulink's toolbox, as illustrated in Fig. 2. This tool allows us to link the requirements with the interconnected blocks representing system objects and functions, which should fulfil the requirements. At the beginning of the design phase, the system model can be very general, but becomes more precise with time.

Shortly after we get concerned with the type of structure of the Stewart platform, the use of electric or hydraulic actuators, and its control system, we can think about workspace and overload checking functions. At this point, the idea of involving the digital twin for predicting fault states can be included in the solution variants. Therefore, we do not define the digital twin as a requirement, but we define

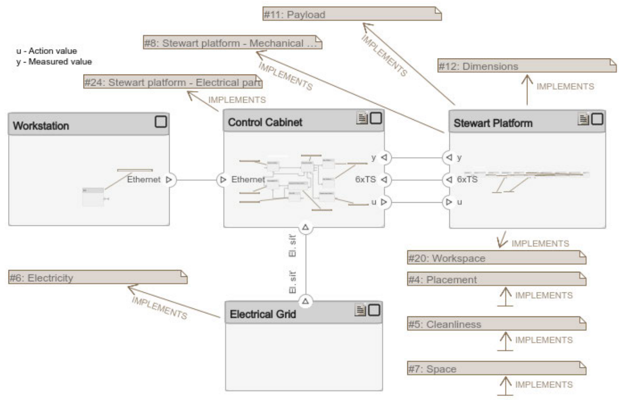


FIGURE 2. The system architecture described in the system composer™.

the required functions of the system that can be fulfilled by employing the digital twin. If we decide to develop this idea, we need to augment our system design with a hardware containing the virtual entity of the digital twin. It also must be decided what kind of model will represent the virtual entity and how the data/information will be exchanged.

Multibody simulation in machine design is standard practice in model-based approaches. In our scenario, we could use it not only to verify our design, but also as a virtual entity for the digital twin [31], [32], [33]. From our point of view, the virtual entity could be an augmented simulation model running in parallel to the real machine with an automatic data/information exchange capability and related decision-making functions.

The verification experiments of the Stewart platform described in the literature used a camera [34], force measurements in the struts [35] or inertial measurement sensors [36], [37]. Camera and inertial measurement sensors are used for an estimation of the kinematic variables. Meanwhile, force measurements are used for the verification of the dynamic properties.

Since the authors have experience with modelling and simulation in the Matlab/Simulink environment, the first design variant proposes to create a multibody model with its Simscape Multibody toolbox [38]. Although it should be noted, different options to create a simulation model of the Stewart platform are possible, e.g., using different software for multibody simulation or employing models described in the literature [39], [40], and [41].

A hardware containing the virtual entity in our case could be a real-time target [42], a PC [43], or a machine controller [44]. A necessary condition for the hardware is a sufficient computational power, a communication interface that enables connection to the other devices of the architecture, and the possibility to utilise a Matlab/Simulink model on it. The advantage of using a real-time target is its capability to perform the simulation in real time and its I/O interface. Its disadvantage is the cost and limitation of a fixed-step solver. A PC in contrast can provide a cheaper solution with various options for simulation of a real system. However, at the price of limited real-time simulation capabilities. Employing a machine controller that could also integrate the virtual entity seems to be a perfect solution. The experiment described in [44] and [45] where the simulation of a single hydraulic axis is performed on the PLC (Programmable Logic Controller) for the purpose of virtual commissioning is close to this approach.

A design of a machine is an iterative process, and the architecture is refined until the solution variant that meets all the requirements is found. The selection and design process of most system components is not described in this article, since it would exceed its scope. The resulting choices are captured in the designed digital twin architecture shown in Fig. 3. The methods and tools for developing the physical and virtual entity of the Stewart platform digital twin are described in the following sections.

The Stewart platform as a physical entity is controlled via a Bosch MLC XM21 machine controller that allows synchronised motion control of six Bosch EMC-063 linear electric actuators through Bosch HCS01 compact converters and the SERCOS communication interface. An important tool that the controller offers is its Open Core Interface (OCE). The OCE allows the user to control the linear electric actuators through the Matlab and Simulink environment [46], [47]. This is beneficial for the integration of the multibody model as a virtual entity, as shown in Section V.

A PC performed well in solving the Stewart platform multibody model, as shown in Section IV. Therefore, we decided to use it as both a platform for a virtual entity and an operator workstation with a Matlab-based user control interface. The PC configuration includes CPU Intel(R) Core(TM) i5-13600K at 3.5 GHz, GPU NVIDIA GeForce RTX 3080 Ti, 32 GB RAM, and Windows 10 Pro 64-bit.

For the purpose of the verification experiment and the development of digital twin-based functions, Comforia MCF150-5kN load cells for force measurement have been included in the design. The load cells are connected to the amplifiers that adjust the measured signals to the ±10 V range suitable for the analog inputs of the dSPACE MicroLabBox measurement unit. The signals are processed, and the resulting data are sent over the local area network (LAN) to the PC. The central node of the LAN is a router that connects the other devices to the star topology.

II. STEWART PLATFORM—KINEMATIC STRUCTURE

A design of the mechanical part of the Stewart platform starts with its kinematic structure. In our case, we decided to select the structure with linear actuators that are represented as struts as indicated in Fig. 4. The pose of the Stewart platform is determined by the position and orientation of its moving plate. The position and orientation depend on the length of the struts and joint positions of both plates. The leg vector $\mathbf{l}_i$ is determined by the inverse kinematics (1) and consequently the length of the strut is given by (2).

$$\mathbf{l}_i = \mathbf{r} + \mathbf{R}\mathbf{a}_i^p - \mathbf{b}_i \tag{1}$$

$$|\mathbf{l}_i| = \sqrt{\mathbf{l}_i \cdot \mathbf{l}_i}$$
$$i = 1, 2, \ldots, 6 \tag{2}$$

where $\mathbf{r}$ is a position vector of the reference frame of the moving plate with respect to the reference frame of the base plate, $\mathbf{R}$ is the rotation matrix, $\mathbf{a}_i^p$ is the vector related to the reference frame of the moving plate representing the joint position of the moving plate, similarly $\mathbf{b}_i$ is the vector related to the reference frame of the base plate representing the joint position of the base plate.

As depicted in Fig. 5 the points $\mathbf{B}_i$ and $\mathbf{P}_i$ represent the joint coordinates of the base plate and the moving plate, respectively. The designer specifies the radiuses r_b and r_p, and the offset angles β and α. Without offset angles, the symmetry axes of each respective plate are shifted by 120°. The offset between the symmetry axes of the base and the moving plate is 60°.

The initial configuration of the Stewart platform is determined by the minimal length of the linear actuator and joint positions. The specification of the minimum leg length $l_{\min}$ allows us to calculate the minimum height of the Stewart platform $Z_{\min}$ by (3).

$$Z_{\min} = \sqrt{l_{\min}^2 - (P_{x_i} - B_{x_i})^2 - (P_{y_i} - B_{y_i})^2} \tag{3}$$

where P_{x_i}, B_{x_i}, P_{y_i}, and B_{y_i} are the coordinates x and y of the moving and base plate joints. Leg length $l_{\min}$ is the distance between the pair of joints in the default position of the Stewart platform. This implies that an arbitrary platform and base joint pair ($\mathbf{B}_i$, $\mathbf{P}_i$) can be selected for the calculation.

The length of the strut is limited by its minimum and maximum length, as given by (4). Where $l_{\text{extension}}$ is the maximum leg extension.

$$l_{\min} \leq |\mathbf{l}_i| \leq l_{\max}$$
$$l_{\max} = l_{\min} + l_{\text{extension}} \tag{4}$$

Another constraint is imposed by the joint angles between the leg, the moving platform, and the base plate. The joint angle between the leg and the moving plate can be calculated according to (5).

$$\gamma_i = \arccos\left(\frac{\mathbf{l}_i \cdot \mathbf{v}}{|\mathbf{l}_i||\mathbf{v}|}\right) \tag{5}$$

where γ_i is the angle between the leg vector $\mathbf{l}_i$ and the vector $\mathbf{v}$ that is perpendicular to the moving plate. Equation (5) can be applied to calculate the angle between the leg vector and the static base plate if we substitute $\mathbf{v}$ with $\mathbf{n}$ which represents the normal vector to the base plate. This angle will be represented by η_i. The constraints of the joint angles γ_i and η_i are given by (6).

$$|\gamma_i| \leq \gamma_{\max}, |\eta_i| \leq \eta_{\max} \tag{6}$$

To obtain a larger workspace, it is often beneficial to align the joint axis with the leg vector $\mathbf{l}_i$ in the default position of

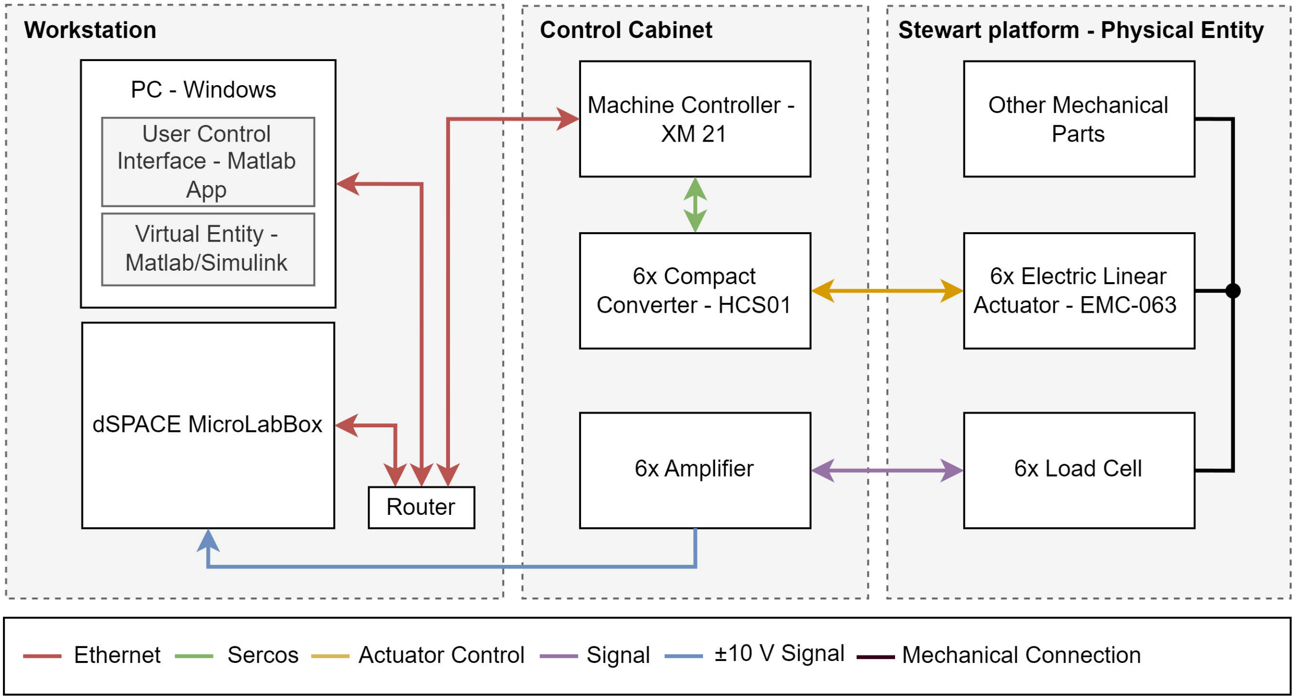


FIGURE 3. The designed Stewart platform digital twin architecture.

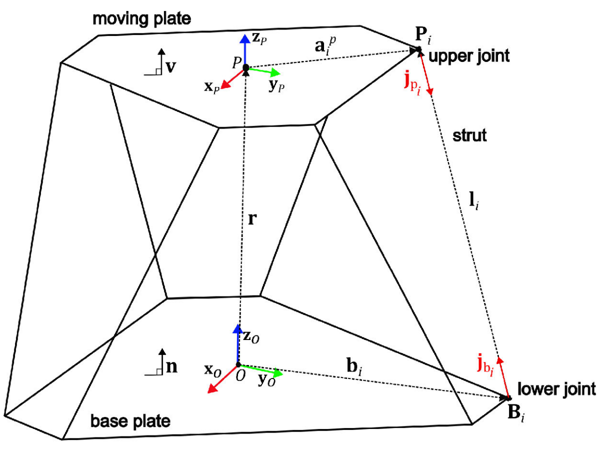


FIGURE 4. General kinematic structure of the Stewart platform.

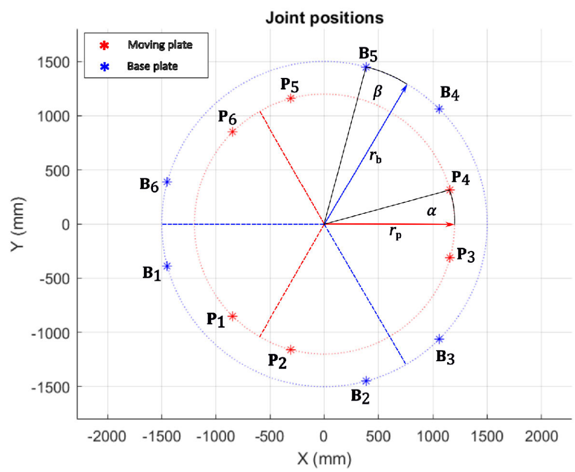


FIGURE 5. Stewart platform base and moving plate joint positions.

the Stewart platform. The lower joint axis is $\mathbf{j}_{\mathrm{b}_i}$. As a result, we will get zero initial joint angles. For the base joint angle η_i the (5) is slightly modified to (7) [48].

$$\eta_i = \arccos(\frac{\mathbf{l}_i \cdot \mathbf{j}_{\mathrm{b}_i}}{|\mathbf{l}_i||\mathbf{j}_{\mathrm{b}_i}|}) \tag{7}$$

To calculate the angle γ_i between the leg vector $\mathbf{l}_i$ and the upper joint axis is $\mathbf{j}_{\mathrm{p}_i}$ we must rotate $\mathbf{j}_{\mathrm{p}_i}$ with the moving plate as shown in (8).

$$\gamma_i = \arccos(\frac{\mathbf{l}_i \cdot (-\mathbf{R}\mathbf{j}_{\mathrm{p}_i})}{|\mathbf{l}_i||\mathbf{j}_{\mathrm{p}_i}|}) \tag{8}$$

A. STEWART PLATFORM DESIGN APPLICATION

The kinematic structure of the Stewart platform was synthesised and analysed in an iterative manner using an application created in Matlab App Designer shown in Fig. 6. Among the main features of the application belong the parameterisation of the kinematic structure, investigation of the workspace, and export of the structure parameters to the file. The application also allows saving current and loading past kinematic structure parameters.

The application is based on the equations presented earlier in this section. On the left side of Fig. 6 is a control panel with input fields and buttons that allows the user to parameterise the general kinematic structure of the Stewart platform by specifying the base and platform radiuses r_b and r_p, the offset angles β and α, the maximum actuator extension $l_\mathrm{extension}$ and the minimum length of the strut l_min which is given by the length of the actuator body with a fully retracted piston. By entering these parameters, the application can calculate the minimum height Z_min of the Stewart platform.

Based on the input parameters provided, the Stewart platform configuration can be plotted in an arbitrary position and orientation, as shown on the upper right side of Fig. 6. Below the graph are the values of actual leg lengths $|\mathbf{l}_i|$ and joint angles γ_i and η_i. The joint angles can be evaluated with respect to either the joint axis perpendicular to the plate or the joint axis aligned with the leg in the default position, as was explained earlier.

By this means, we can explore either a single position and orientation or a predefined set. A predefined set is a variable that contains workspace points that are to be investigated. These are the red points on the plot on the upper right side of Fig. 6. Although this allows us to evaluate different configurations, we cannot estimate the mechanism workspace without defining constraints.

To perform a workspace analysis, we need to select constraints l_min, l_max, γ_max, and η_max. Here l_max is given by the sum of the actuator body length and its maximum extension. In our case, we determined the constraints from the parameters of the strut configuration. The strut configuration consists of a lower and upper joint, a linear actuator, a load cell, and mounting parts. Parameters were taken from the datasheets of the preselected equipment. The selection of the linear actuator and load cell was based on the payload and required dynamic properties. The joint positions **B** and **P** were determined on the basis of the analysis of the workspace.

Workspace analysis is performed by determining the constant-orientation workspaces of the Stewart platform. This is accomplished by evaluating a set of desired workspace positions with constant-orientation of the Stewart platform. If the constraints are not violated, the point belongs to the constant-orientation workspace, as shown in Fig. 7. It helped us find constant-orientation workspaces for boundary values and decide if it is viable to proceed with the configuration. In the application, we do not check for body collisions and singularities. Body collisions are checked with both basic calculations based on the shape of the selected linear actuator and visually in the CAD software. Due to the selected configuration of the Stewart platform, we assume that the singularities are not within the specified workspace.

After the initial analysis, the data files describing the kinematic configuration are exported and used in Autodesk Inventor CAD software. The output files contain the joint coordinates **B** and **P** of the base and moving plate, the body length of the actuator l_min and the maximum extension of the actuator $l_\mathrm{extension}$ to parameterize the 3D models of the base and moving plate, and linear actuators that are combined in the assembly model of the Stewart platform. Based on joint alignment, it can also contain data describing the upper and lower joint axis vectors $\mathbf{j}_{\mathrm{p}_i}$ and $\mathbf{j}_{\mathrm{b}_i}$ that are crucial for the configuration of a proper joint orientation.

A sketch of the joint positions of the base plate is shown in Fig. 8. The dimensions of the sketch are parameterized by the base joint coordinates **B**. In case the output data files are changed, the sketch is updated, and therefore the whole assembly. By this approach, we could evaluate multiple variants relatively quickly.

After assigning realistic physical properties to the bodies or its direct replacement by CAD models of its real counterparts, the selected variant is transformed into the multibody simulation model for further analysis. The presented method can also be extended for parameterisation of the multibody simulation model [49].

III. STEWART PLATFORM—CREATING A MULTIBODY DYNAMICS MODEL

Unfortunately, the kinematic structure will not help us evaluate the dynamics requirements on the Stewart platform. The dynamics of the Stewart platform will be mainly determined by the choice of actuators and its control system. However, the dimensions of the selected equipment must either fit our kinematic structure or be adjusted. To support the development process, it is essential to employ tools for modelling and simulation.

As described in Section II-A we first chose the linear actuators and built the structure around them (Fig. 9). The presented solution variant meets the requirements described in Table 1 that concern the mechanical design. Assembly parts are either commercially available (e.g., linear actuators, load cells, spherical joints, etc.) or manufactured (e.g., base and moving plate, etc.). The manufactured parts are based on the drawings generated from the CAD model.

Linear actuators have been selected on the basis of experiments with multiple iterations of a multibody model. The problem with the multibody model is that we cannot verify it until the real machine is built. In our case, the motivation to build and verify a multibody model stems both from the design and from the perspective of the digital twin application.

The CAD model in Fig. 9 describes the real Stewart platform shown in Fig. 1. However, generating a multibody model directly from this configuration would lead to a complex and computationally demanding model.

The CAD model consists of parts (e.g., bolts, wheels, mounting equipment, etc.) that are essential for the function of the real Stewart platform but unimportant for the multibody simulation model. Another issue represent parts that form a functional unit but are interconnected by fixed constraints, e.g., piston parts, linear actuator mechanical drive and electrical motor, etc. The number of bodies and joints increases the complexity of the model. Therefore, to obtain the best possible performance, it is necessary to reduce the number of these elements to the minimum. The mass of the removed parts is added to the relevant bodies. The remaining bodies are considered homogeneous.

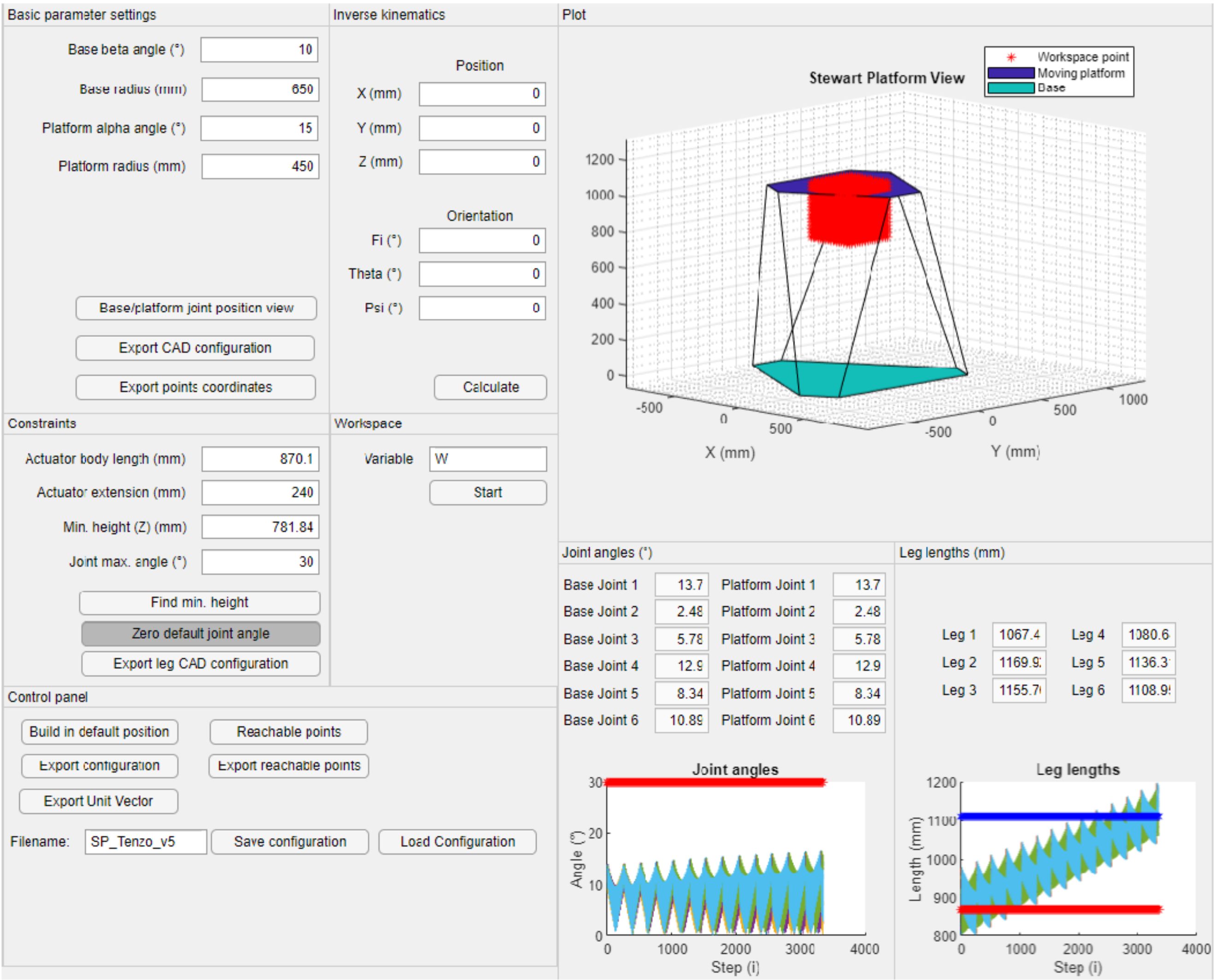


FIGURE 6. Stewart platform design application.

After the preparation of the CAD model, the next step is the generation of the multibody model with the use of the Simscape Multibody Link plug-in. It generates the XML multibody description file and a set of body geometry files. The files are then imported to Matlab and based on them a reduced multibody simulation model of the Stewart platform is generated.

The structure of the multibody model must be checked to ensure that it was generated correctly. To verify this, Matlab/Simulink offers diagnostic tools that report some of the model problems. The multibody simulation model can also be visually inspected (Fig. 10) to see whether it resembles the modelled system. The visualisation tool is also viable during the digital twin experiments presented in Section V-B.

After setting up a gravity vector and constraints, it can be checked if the model does not fall apart by the effect of the gravitational force or violated constraints. Some of the parameters, such as joint limits, must be set as well, since they may not be transferred directly from the CAD software. The joints that represent linear actuators must be set in a forward or inverse dynamics regime to analyse motion profiles and acting forces.

The refined multibody simulation model is the main part of a block diagram in Fig. 11. The measurement block encompasses all scopes for tracking values related to the Stewart platform. It ranges from leg extensions, moving plate position and orientation to the joint angles and forces exerted by the linear actuators.

The trajectory generator block shown in Fig. 11 serves to plan a trajectory of the moving plate. The inverse kinematics block transforms the trajectory point into the required leg lengths. The output signals that carry the leg length values lead to the blocks that convert the Simulink signal into the Simscape physical signal. This block also functions as a second-order low-pass filter that can provide the first and

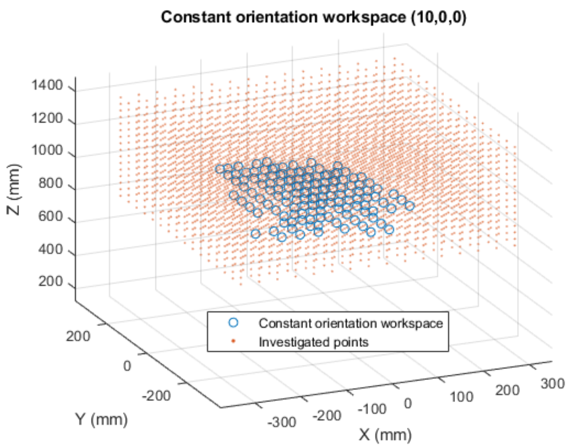


FIGURE 7. Workspace analysis for the constant-orientation (10°, 0°, 0°) workspace of the Stewart platform.

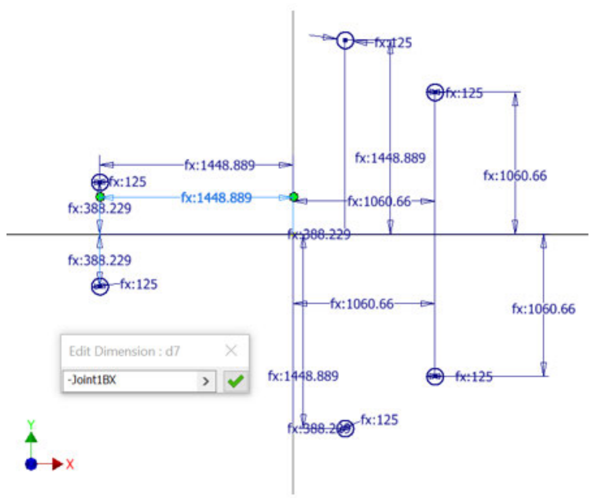


FIGURE 8. Parameterized joint positions of the base plate.

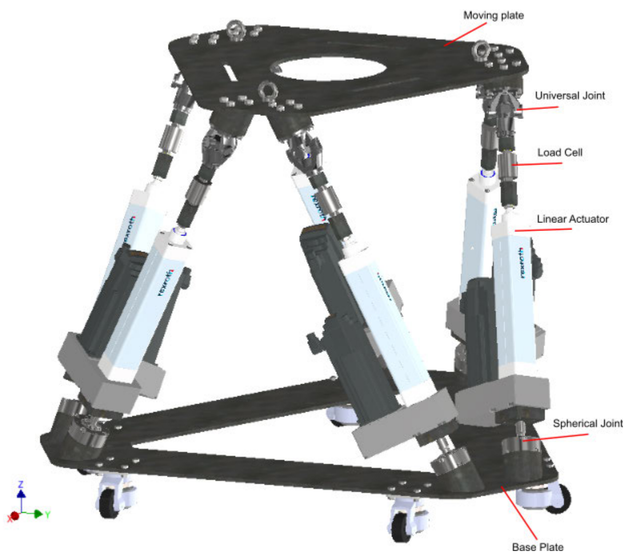


FIGURE 9. Stewart platform CAD model.

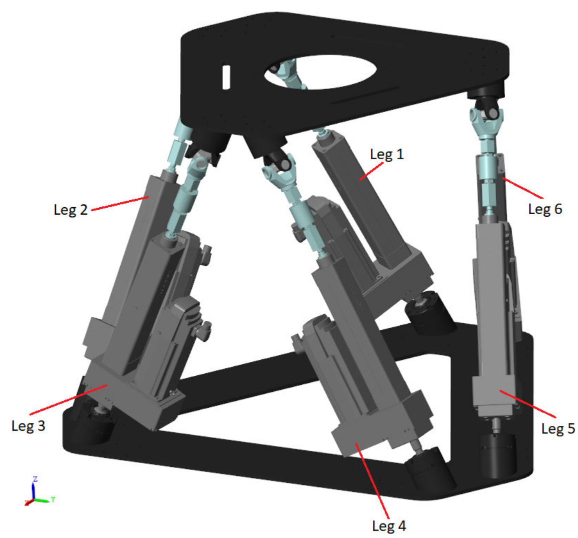


FIGURE 10. Multibody model of the Stewart platform.

second derivatives of the input. The converted signals lead to cylindrical joint blocks set into the inverse dynamics regime. The cylindrical joint represents a linear actuator that consists of its body and an extensible piston. The extensible piston changes the length of the strut.

At this phase, the multibody model allows us to test arbitrary motion profiles of the moving plate (Fig. 12) and determine if the piston extensions (Fig. 13), velocities (Fig. 14), accelerations and forces (Fig. 15) are in the limits given by the linear actuator manufacturer.

IV. STEWART PLATFORM—MULTIBODY DYNAMICS MODEL VERIFICATION

As foreshadowed earlier, since we plan to employ the multibody model as a virtual entity, we need to verify it. To verify the multibody model, we must compare relevant machine states with its simulated counterparts. In our case, the model is verified based on the force measurements obtained from the load cells whose location can be seen in Fig. 9.

In the previous Section III we have reduced the linear actuator to two parts, the body and the piston. Since the piston is reduced to the single body, we cannot obtain the force acting on the load cell but only the force exerted through the cylindrical joint (Fig. 16(a)).

In practice, it is possible to obtain the acting force of a linear actuator by measuring the motor currents and knowing the relevant parameters of the motor and linear drive. However, these parameters can be difficult or even impossible to obtain. The accuracy of such an approach can also be insufficient. Therefore, the reason the load cell is included in our design is better measurement accuracy. The used load cells can measure forces in both push and pull directions up

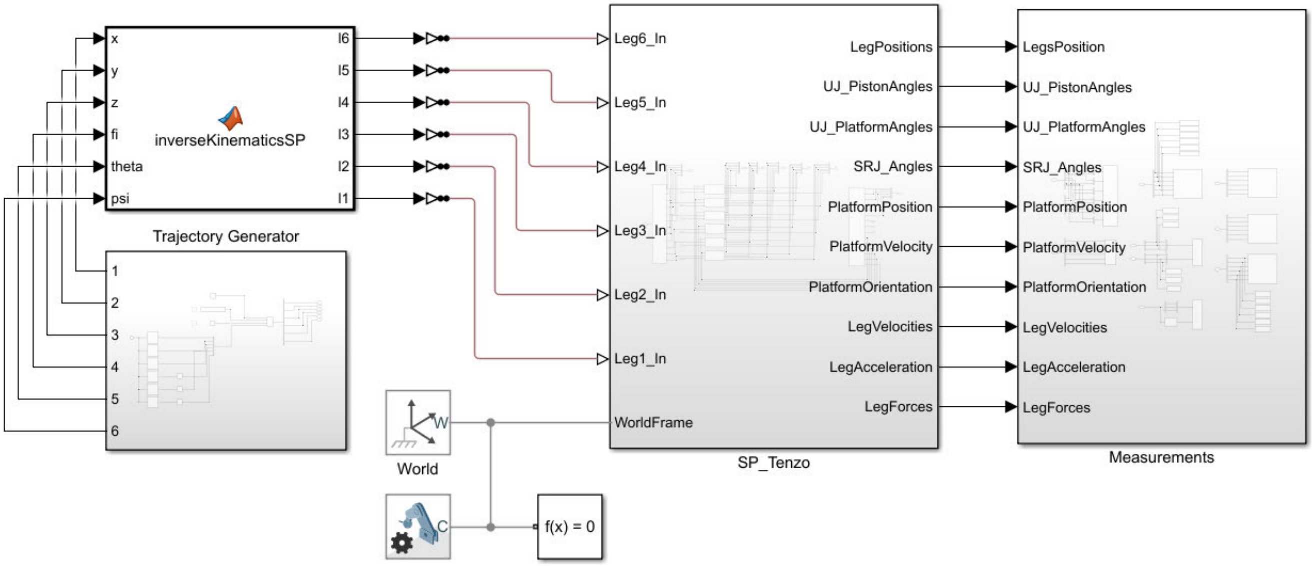


FIGURE 11. Stewart platform multibody Simulink diagram.

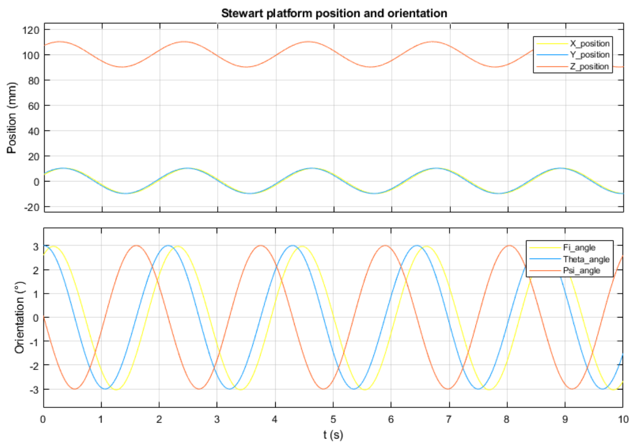


FIGURE 12. Stewart platform motion profile.

TABLE 2. Maximum error of Comforia MCF150-5kN load cell.

Maximum error	Value
Zero offset	2% of full scale
Linearity	0.1% of full scale
Hysteresis	0.1% of full scale
Creep (30 minutes)	0.05% of full scale

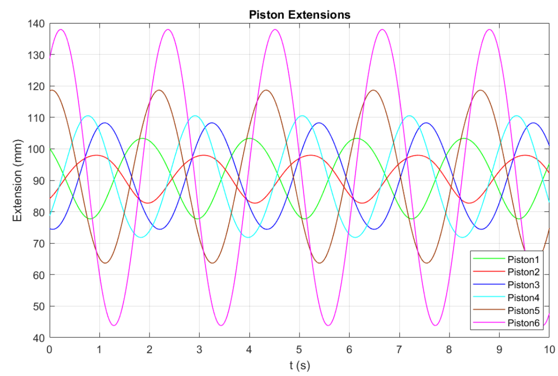


FIGURE 13. Piston extensions.

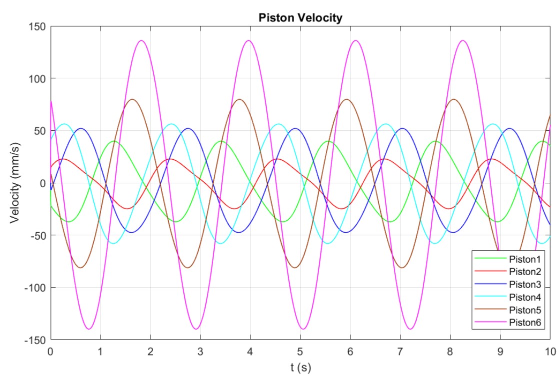


FIGURE 14. Piston velocities.

to the 5 kN range. Its maximum error given by a manufacturer is shown in Table 2.

The single-body piston model variant was sufficient for machine design. However, for the purpose of model verification, we developed a second version of the model. In the second version, we divide the piston into two bodies. The piston is split in the position of the load cell. The acting force is measured with respect to its reference frame shown in Fig. 16(b). The two piston bodies are connected by a fixed constraint (Weld joint).

This modification will add six more bodies to the system, and therefore it increases the computational load in exchange

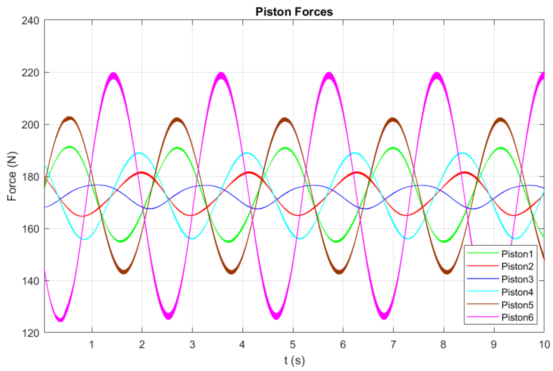
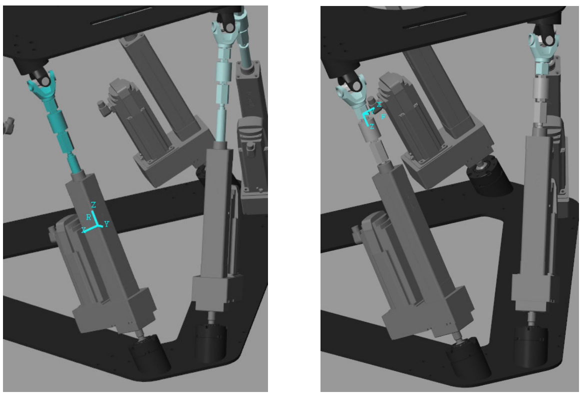


FIGURE 15. Piston forces.



(a) Single body piston (b) Piston split into two bodies

FIGURE 16. Force measurement reference frames.

for the increased fidelity of the multibody model. Since an optimal scenario would be the highest possible model fidelity with the least computational load, this step represents the dilemma of building the virtual entity of the digital twin.

During the machine design phase and the verification experiment, we are not concerned with the computational load as long as we are able to obtain the required result in a reasonable time. However, for the digital twin, the computational time is essential since it directly affects synchronisation of the virtual and physical entities.

With Simulink's Solver Profiler tool (Fig. 17), we can analyse the performance of the Stewart platform multibody simulation model for an arbitrary motion profile. In Fig. 17 is visible that we used the variable-step solver ode45. The important marker is the run/simulation time ratio. It gives us information on whether a current model configuration can be run in a near real-time scenario or not. In our testing scenario, we reached the run/simulation time ratio lower than one, which signals that the simulation model is capable of running in the real-time scenario. Therefore, at this stage, we estimate that the multibody simulation model in combination with computational hardware is usable for integration in the digital twin architecture. If the run/simulation time ratio was higher than one, either the simulation model must be further refined or the computational hardware needs to be upgraded.

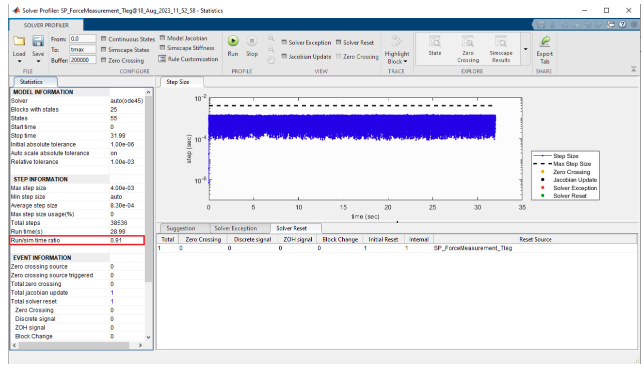


FIGURE 17. Performance evaluation of the Stewart platform multibody model.

A. VERIFICATION EXPERIMENT

After building the machine, setting up a control and measurement instrumentation and preparing the basic user interface to issue commands to the Stewart platform, we performed the verification experiment. At this point, the system architecture is almost identical to the one presented in Fig. 3, with the exception that the LAN does not yet need to be established.

To verify the model, we obtained the force and leg extension measurements during the monitored maneuver of the real Stewart platform. Leg extension measurements were obtained through the development environment of the control system and its oscilloscope tool. Simultaneously, the measured forces were processed by MicroLabBox and postprocessed in the Matlab environment. Leg length measurements were used as input to the cylindrical joints block in the inverse dynamics regime, which allowed us to replicate the maneuver and obtain the calculated forces. The measured and calculated forces were then compared. Therefore, in this experiment, the trajectory generator and the inverse kinematics blocks (shown in Fig. 11) were bypassed.

The real and calculated signals did not have the same length and sample rates. To correct for this, we had to resample the force measurement and find a time lag between the real and calculated force signal. Therefore, to compare the signals, we used a cross-covariance [50].

Fig. 18 shows the result of the verification experiment. The six graphs in column (a) show the result of a cross-covariance between the calculated and measured force signals for each Stewart platform leg. The graphs signal that there is a match between each pair of signals and that the time lag is approximately 2 seconds. The column (b) plots the compared signals with its mean values subtracted and shifted by a value of time lag. In this view, we can compare the signals without the offset error. In column (c) the original shifted signals are plotted.

The differences between the real and simulated signals may seem insignificant in this case. However, the possible sources of an error must be taken into account. In Section III we described simplifications that traded model fidelity for computational speed. Other simplifications include omitting

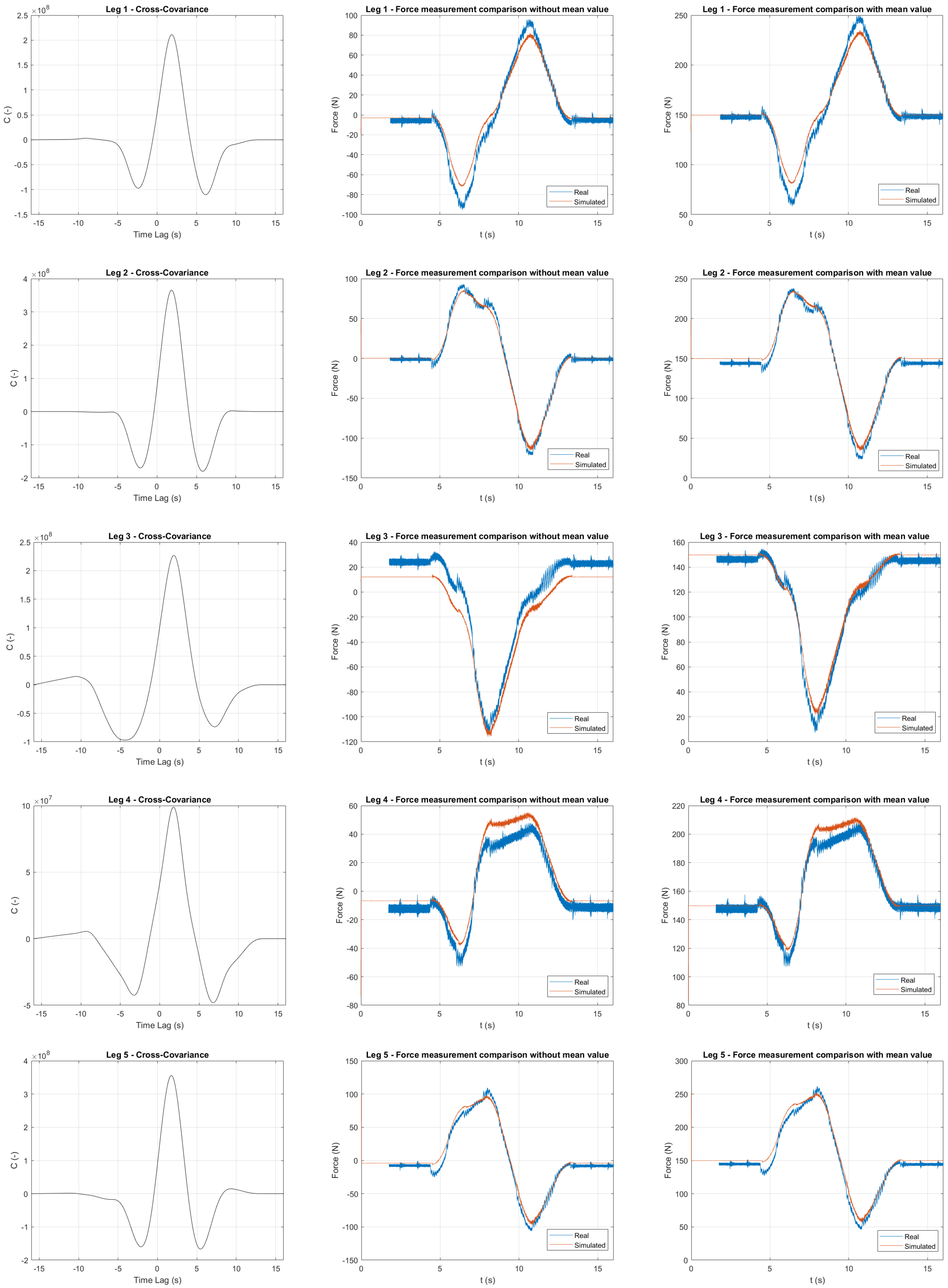
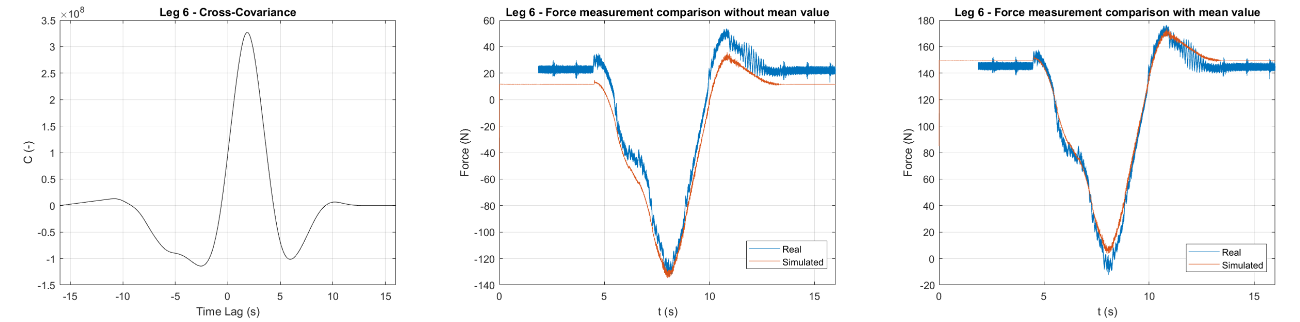


FIGURE 18. **Comparison of the calculated and measured force signals for each leg of the Stewart platform.**



(a) Cross-covariance and a time lag between signals (b) Comparison without a mean value (c) Comparison with a mean value

FIGURE 18. ***(Continued.)*** **Comparison of the calculated and measured force signals for each leg of the Stewart platform.**

damping and stiffness in joints and ignoring the flexibility of the machine bodies. The errors of the numerical method used (i.e. truncation, round-off) are to be considered as well.

Another source of error is a conversion of the machine from the blueprints to the real world. The quality of materials, manufacturing tolerances, backlashes, quality of assembly, etc. can all be factors that might cause a deviation of the model. And last but not least, the measurement itself contains systematic and random errors.

All of these considerations are related to the fidelity of the digital twin and are partially the reason why the achievement of a perfect virtual replica is very challenging or almost unattainable. The previous information implies that even if we consider the simulation model as verified, there will be some uncertainty. If this amount of uncertainty is considered acceptable, it needs to be taken into account in the digital twin functions.

V. STEWART PLATFORM–DIGITAL TWIN

After the multibody model verification we have obtained the first piece to develop the virtual entity of the digital twin. The development of the digital twin will not be described to the point of application in this paper. In this section, we will present basic features enabled by the bi-directional communication between the physical and virtual entities and outline the challenges brought by this integration. The features are enabled by the architecture choices presented in Fig. 3.

To setup the bi-directional communication between the virtual and physical entities, we need to decide which data/information will be transferred. In our case, the data flow from the physical entity will be represented by measured forces. The information that will be sent from the virtual entity is represented by the issued commands (i.e. start/stop motion, move to, etc.) for the Stewart platform.

To put used terms in context, we present a reduced block diagram describing the Stewart platform digital twin (Fig. 19). Here, the virtual entity represents a single instance of the existing physical entity with features for the communication and workspace and overload checking functions. The physical entity of the Stewart platform is limited to the mechanical part of the system, which is described by the virtual entity. The interface incorporates the control, measurement, and communication hardware. It ensures the bi-directional communication between entities and transforms the issued commands to the actions of the physical entity.

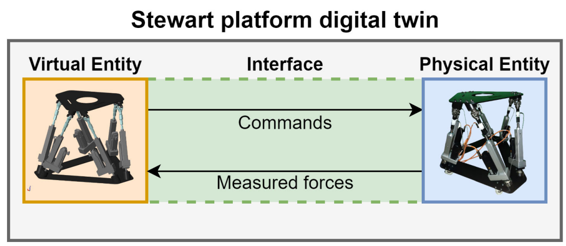


FIGURE 19. **Stewart platform digital twin.**

A. INTEGRATION OF THE MULTIBODY MODEL

In Section IV-A we used MicroLabBox to record the force measurement. The measurement was then post-processed for the purposes of model verification. In the presented digital twin architecture, the measured data are directly fed from the MicroLabBox to the PC that is a platform for the virtual entity. Therefore, to reuse the multibody model described in the previous sections in the digital twin architecture, we had to further modify it.

The modified Simulink diagram of the multibody model is shown in Fig. 20. It is augmented with basic functions for bi-directional communication and workspace and overload checking. The idea is that the simulation model runs ahead of the real machine and issues a stop command if a variable of interest exceeds predefined limits. The inputs to the multibody model are six predefined linear actuator trajectories that must be known in advance.

The first assumption of running the simulation ahead of the real machine in parallel is the real-time capability of the simulation model. This is the reason why we evaluated the run/simulation time ratio in Section IV. Since the resulting

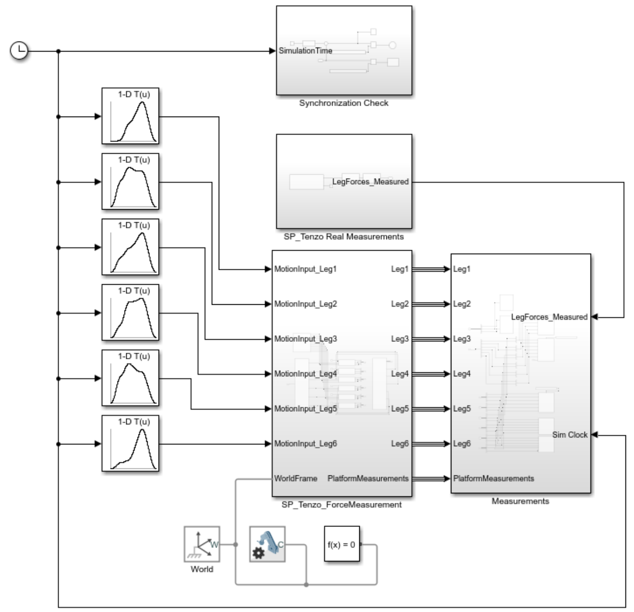


FIGURE 20. Modified Stewart platform multibody Simulink diagram for the reuse in the digital twin architecture.

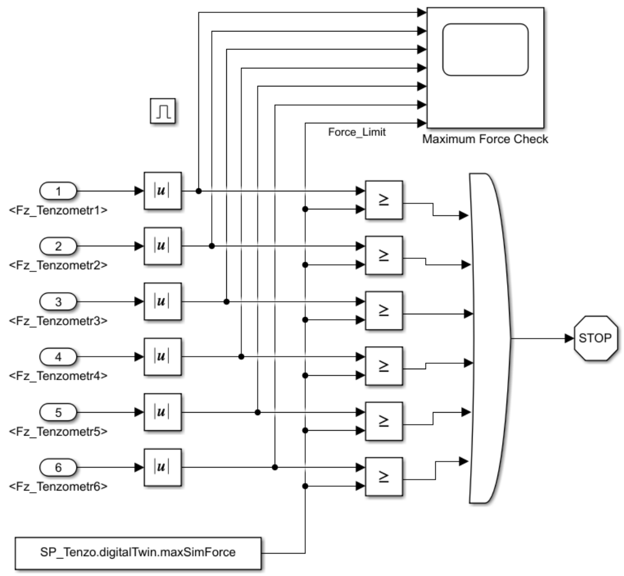


FIGURE 21. Simulink diagram that represents function to issue stop command to the real Stewart platform if the maximum force is exceeded.

run/simulation ratio was less than one, it suggests that the first assumption is fulfilled.

However, since we want to incorporate the online measurements within the simulation model, it would be optimal if it were perfectly synchronised with the real-time. In other words, the run/simulation ratio would have to be equal to one throughout the whole experiment. Otherwise, the measurements will be distorted as shown in the following Section V-B. To keep the run/simulation ratio as close as possible to one, we used the simulation pacing option offered by the Simulink environment. In our case, the simulation pacing option slows down the simulation near the pace of the wall clock time.

To incorporate real force measurements into the simulation model, we used the UDP Receive block. It can process UDP packets that carry data that represent the measured forces. The most important parameter of the UDP Receive block is the sample rate. To obtain an accurate measurement, it is necessary to have a sufficiently high sampling rate. The reception of packets during the course of the simulation directly impacts its performance and reveals another complexity related to the concept of a digital twin. From the authors' point of view, these problems are closely related to the co-simulation area [51], [52].

UDP allows data to be streamed over the network without the need of a handshake with a target device [51]. This is beneficial from the data speed transfer point of view and is also the reason why UDP is often used in real-time applications. Unfortunately, the connection cannot be considered reliable since the protocol does not check if all sent packets were received.

On the other hand, to issue commands to the physical entity, the mlpi4MATLAB toolbox based on Open Core Engineering is used. This toolbox allows for connection to the controller of the Stewart platform and issue commands to it through the Matlab environment. Unfortunately, the command functions cannot be called directly from Simulink during the course of the simulation. To deal with this issue, we used a Simulink feature - model callbacks.

The model callbacks allow us to execute command functions in the specified situations during the course of the simulation, such as start, stop, continue, or pause events. Some of these events are performed during each simulation run. In other scenarios, it can be triggered by specific conditions during the course of the simulation, such as exceeding a threshold value of a variable (e.g., joint or force limits).

An example of integration of this function is shown in the Simulink diagram in Fig. 21. If the maximum leg force is exceeded, the simulation is stopped and the callback function, which issues the motion stop command, is called. If the simulation is ahead of the real machine and the pacing constraint is not violated, the physical entity stops before it reaches the set limit.

However, when configuring the threshold values, we must remember that the simulation model has some level of fidelity and related uncertainty, as discussed in Section IV-A. In the application, that should be taken into account by multiplying the threshold value by a safety coefficient. The determination of the safety coefficient could be based on a statistical evaluation of the differences between the calculated and measured leg forces obtained during the verification experiment. This should include measurement errors of the measured quantity.

It should be noted that all of the introduced functions also increase the computational complexity of the system, which can cause technical difficulties. The technical difficulties and their potential solutions will be explained in the following subsection.

B. DIGITAL TWIN EXPERIMENT

The graphs shown in Fig. 22 depict the results of an experiment in which the simulation of a given trajectory was started and after 2 seconds the command for the real machine was issued. The measured forces were processed throughout the experiment. From the graphs, it can be seen that we started to get real measurements approximately 3 seconds after the command was issued. Although what is not visible from the graph is the fact that the machine starts to move almost immediately after the issued motion command.

Therefore, the delay can be divided into two terms. The first term is the delay between the issuing of the command and the physical action. On the basis of the observation, this delay is estimated to be in the order of fraction of a second. The second term is the delay between the measurement and its processing in the simulation environment, which is the rest of the total delay. The total delay remains constant during the digital twin experiments.

This technical issue is related to the area of computer networking. Currently, it is unknown whether the problem is related to hardware or software. However, since UDP is used in real-time applications [51], [53], we believe it is the correct choice for the architecture and the problem is solvable. It emphasises the need for an interdisciplinary approach to the development of digital twins.

Another issue is related to the possible distortion of the measured force due to the non real-time simulation of the model. The graph in Fig. 23 shows a significant distortion of the measured force signal. The calculated leg force is shifted on the time axis so that it roughly matches the start of the real maneuver measurement.

The distortion is the result of the run/simulation ratio being larger than one. Although the simulation pacing option tries to keep the run/simulation time ratio equal to one, it can be violated if the computational load is too heavy. In this case, we increase the computational load by enabling universal joint angle, maximum force, and force difference checking functions. The selected solver was variable-step solver ode45 which is the same configuration as tested in Section IV. The computational time can also be increased by increasing the sampling rate of the UDP Receive block and the general load of the PC.

In contrast, the graph in Fig. 24 shows that there is a match between both signals. In this case, we used the fixed-step solver ode3 with a fixed step size of 0.002 seconds. Although in this configuration the fixed-step yields better results, it could be possibly different for trajectories with a different dynamics.

The potential prevention of this issue is the continuous check of the run/simulation time ratio (as depicted by the block "Synchronization Check" in Fig. 20), and in case it gets significantly violated, the callback function will be issued to stop the machine. However, it must be noted that extending the digital twin with both safety and application functions increases its computational load, and therefore maintaining the pacing constraint is even more difficult.

The solution of the problem would be to design the virtual entity of the digital twin in such a way that it manages to perform all required tasks near real-time. This is also one of the basic assumptions for future applications.

In this section, we describe the integration of a multibody model as a virtual entity of a digital twin. The benefit of this integration is that we could reuse the developed multibody simulation model. This was enabled by the system architecture and the selected development methods and tools. However, as shown in this section, this integration poses challenges such as communication delay, measurement distortion, and threshold value setting. Although potential solutions were presented, it is obvious that robustness and model fidelity are key factors for successful future applications of the digital twin.

VI. DISCUSSION

This paper highlights the potential benefits of considering the digital twin since the early stages of system development by using the example of the Stewart platform. Potential benefits are represented by the ability to choose system components, methods, and tools that have a symbiotic effect during system integration that includes the concept of a digital twin. The proposed approach can contribute to the emerging methodologies for the development of the digital twin of the parallel manipulator but also to mechatronic systems in general.

In the presented example, we used modelling and simulation techniques to support the development process of the Stewart platform that allowed us to build its multibody dynamics model with the help of the Matlab/Simulink tool. Since we designed the machine, we had access to the physical properties (e.g., masses, dimensions, etc.) of the individual components, making the creation of the multibody dynamics model easier.

Although this could be considered typical design practice, the decision to include the load cells stemmed solely from the purpose to verify the multibody dynamics models which is necessary in order to use it as a part of a virtual entity of the digital twin. Although it would be possible to incorporate the load cells in the existing mechanism, it would result in a change in its properties.

By including load cells in the machine structure, we could perform a verification experiment of the dynamic properties of our Stewart platform. However, this required us to modify the multibody model. We had to split the piston into two bodies so that we could obtain the calculated forces in the same positions as the measured ones. Although this step negatively affects the computational complexity of the model, it increases the fidelity of the model. This step represents a typical dilemma in digital twin applications.

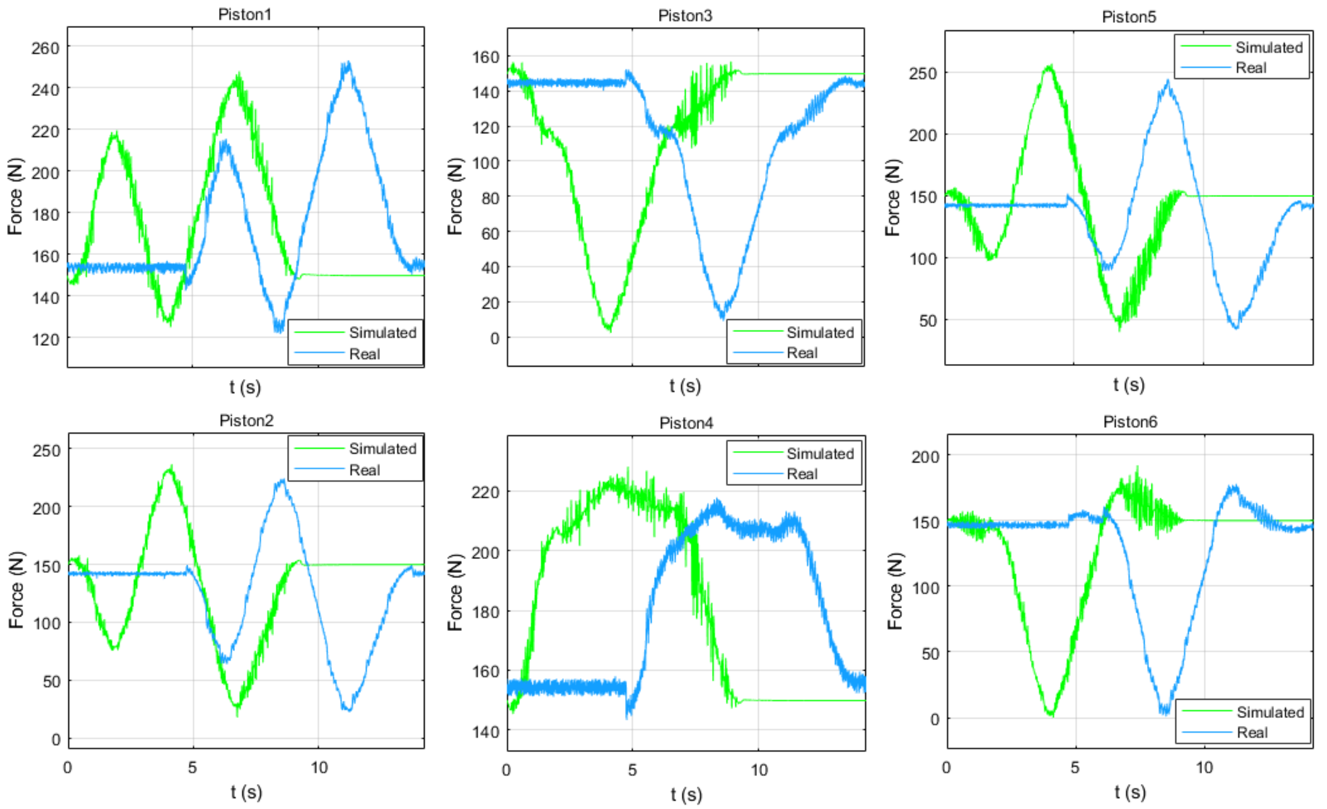


FIGURE 22. Forces obtained during a digital twin experiment.

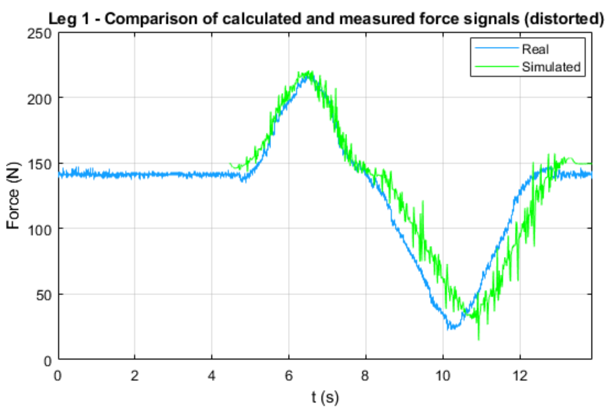


FIGURE 23. The comparison of calculated and measured leg force signals (distorted).

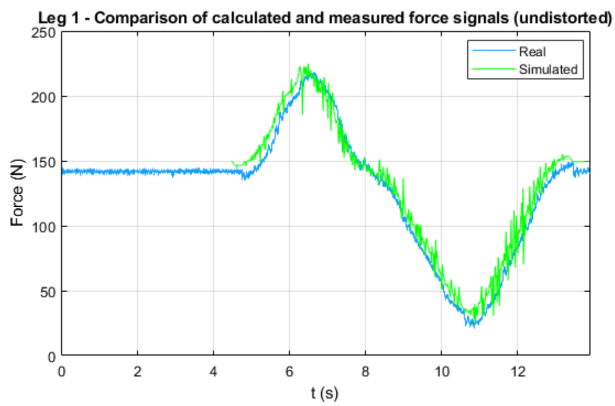


FIGURE 24. The comparison of calculated and measured leg force signals (undistorted).

The verification experiment showed that the simulated and measured forces were in good agreement. Nevertheless, we must note that we verified only the dynamical properties. To verify the kinematic properties, we would have to include the instrumentation to measure the position and orientation of the moving plate. For the demonstration of the proof of concept, we consider the fidelity of the model to be sufficient.

The multibody dynamics model developed in Matlab/Simulink can be integrated as the virtual entity of the digital twin by augmenting it with a bi-directional communication capability with the physical entity and required functions. As shown in this work, the Simulink diagram can be augmented by communication functions that allow receiving force measurement data over UDP and callback functions that issue commands to the Stewart platform controller over the mlpi4MATLAB toolbox based on Open Core Engineering.

However, this integration also presents challenges given by model fidelity, related measurement errors, and the synchronisation of virtual and physical entities (i.e., communication delay, measurement distortion). These properties can directly affect the overload and workspace check functions. This implies that the virtual entity must be up to date representation of the real entity, otherwise the measurements will differ significantly from its virtual counterparts, which will result in digital twin not fulfilling its purpose properly. In other words, it can result in false detection of a critical condition, non-detection of a critical condition, or late response to a critical condition. Although we presented possible solutions for these challenges, it is evident that using the presented digital twin concept in a real application requires increasing the robustness of the system. The first step towards this would be to include a dedicated hardware platform for the virtual entity (e.g., real-time target as mentioned in I-A) or include the virtual entity into the machine controller which currently seems to be more futuristic variant.

It could also be argued that the prediction of exceeding maximum forces and joint angles could be performed offline solely with known inputs and a simulation model that would be periodically updated. Especially if we consider the issues with the synchronisation of virtual and physical entities. However, this configuration would disable a valuable feature, the online evaluation of the difference between measured and calculated force. The difference between measured and calculated force can indicate the change in model parameters (change of load, mechanical faults, etc.) but also collisions with objects. Exceeding the force difference can result in issuing a command to stop the machine.

The Stewart platform fault prediction concept as presented in this paper can be useful in applications where the trajectory is known in advance (e.g., life cycle testing). On the other hand, in use cases where the trajectory is not known beforehand (e.g., vehicle simulators), it is not viable. In such cases, only actual and past states could be compared, which can still be valuable, as explained in the previous paragraph. However, the virtual entity would have to be modified to receive online inputs and issue motion commands throughout the simulation, not only at the beginning.

VII. CONCLUSION

In this paper, a development process of the Stewart platform digital twin is presented from its early stages. The idea of employing a digital twin is shown to originate from the requirements of the system. Based on the requirements, suitable tools and methods that consider the digital twin as well as the system architecture are selected and briefly explained.

The next step, which is concerned with the kinematic structure of the Stewart platform, does not really differ from the typical development process. However, it cannot be omitted since it is a preliminary step in the creation of the multibody model of the Stewart platform. In Sections III and IV we present that in this phase we must take into account both the model fidelity and the computational complexity of the multibody dynamics model. This is for the sake of its subsequent use as a virtual entity of the digital twin. To use the multibody model in the model verification experiment, we created a model variant with a piston divided into two bodies.

In the verification experiment, we compared measured and calculated forces with the use of cross-covariance. This was enabled by introducing load cells into the system architecture. The results of the verification were sufficient. Although it has been stated that differences between the signals can be caused by multiple factors ranging from making too many simplifications to the imprecise transformation of the blueprints to the reality. Additionally, the measurement error also represents an intervening factor.

In combination with computational limitations, these are the reasons why achieving a perfect digital representation of the real object is very difficult or almost unattainable. Therefore, uncertainty must be taken into account.

The last section describes the integration of the multibody model as a virtual entity of the digital twin that runs ahead in parallel to the physical entity. It presents the capability of the bi-directional communication between both real and virtual entities and basic overload and workspace check functions. Through the virtual entity, we can evaluate measured and simulated forces of the real machine and also issue commands to the control system of the Stewart platform as described in the presented experiment results. This experiment revealed some of the challenges given by model fidelity, related measurement errors, and the synchronisation of virtual and physical entities (i.e., communication delay, measurement distortion) that need to be addressed in future work.

The main contribution to the development of the digital twin technology presented in this work is the idea of aligning tools and methods that allow reuse of the developed multibody simulation model and its integration into the digital twin architecture. This is demonstrated on the example of the Stewart platform with the intended use of a digital twin for fault prediction applications.

This work lays the foundation for the next series of experiments that will focus on the robustness and implementation of the digital twins based on multibody simulation models in real applications. This can include evaluating different approaches and software platforms to develop the multibody dynamics-based virtual entity of a digital twin.

REFERENCES

[1] I. Graessler, J. Hentze, and T. Bruckmann, "V-models for interdisciplinary systems engineering," in *Proc. Design 15th Int. Design Conf.*, 2018, pp. 747–756.

[2] G. Barbieri, C. Fantuzzi, and R. Borsari, "A model-based design methodology for the development of mechatronic systems," *Mechatronics*, vol. 24, no. 7, pp. 833–843, Oct. 2014.

[3] X. Liu and R. D. Wiersma, "Optimization based trajectory planning for real-time 6DoF robotic patient motion compensation systems," *PLoS ONE*, vol. 14, no. 1, Jan. 2019, Art. no. e0210385.

[4] A. Preumont, M. Horodinca, I. Romanescu, B. de Marneffe, M. Avraam, A. Deraemaeker, F. Bossens, and A. Abu Hanieh, "A six-axis single-stage active vibration isolator based on stewart platform," *J. Sound Vibrat.*, vol. 300, nos. 3–5, pp. 644–661, Mar. 2007.
[5] C. Du, T. Wang, H. Zhou, Z. Sun, and Y. Lei, "Development of an integrated load testing device for a substructure hybrid test of a curved bridge," *Experim. Techn.*, vol. 46, no. 3, pp. 395–411, Jun. 2022.
[6] V. S. Arconada, J. García-Barruetabeña, and R. Haas, "Validation of a ride comfort simulation strategy on an electric stewart platform for real road driving applications," *J. Low Freq. Noise, Vibrat. Act. Control*, vol. 42, no. 1, pp. 368–391, Mar. 2023.
[7] D. Silva, J. Garrido, and E. Riveiro, "Stewart platform motion control automation with industrial resources to perform cycloidal and oceanic wave trajectories," *Machines*, vol. 10, no. 8, p. 711, Aug. 2022.
[8] M. Toz and S. Kucuk, "Dexterous workspace optimization of an asymmetric six-degree of freedom Stewart–Gough platform type manipulator," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1516–1528, Dec. 2013.
[9] H. Tourajizadeh, M. Yousefzadeh, and A. Tajik, "Closed loop optimal control of a stewart platform using an optimal feedback linearization method," *Int. J. Adv. Robotic Syst.*, vol. 13, no. 3, p. 134, May 2016.
[10] H. Yun, L. Liu, Q. Li, W. Li, and L. Tang, "Development of an isotropic Stewart platform for telescope secondary mirror," *Mech. Syst. Signal Process.*, vol. 127, pp. 328–344, Jul. 2019.
[11] X. Song, Y. Pan, and Y. Chen, "Development of a low-cost parallel kinematic machine for multidirectional additive manufacturing," *J. Manuf. Sci. Eng.*, vol. 137, no. 2, Apr. 2015, Art. no. 021005.
[12] M. Hiller, S. Fang, S. Mielczarek, R. Verhoeven, and D. Franitza, "Design, analysis and realization of tendon-based parallel manipulators," *Mechanism Mach. Theory*, vol. 40, no. 4, pp. 429–445, Apr. 2005.
[13] T.-L. Wu, J.-H. Chen, and S.-H. Chang, "A six-DOF prismatic-spherical-spherical parallel compliant nanopositioner," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 55, no. 12, pp. 2544–2551, Dec. 2008.
[14] H. C. G. Pritschow and K.-H. Wurst, "Systematic design of hexapods and other parallel link systems," *CIRP Ann.*, vol. 46, no. 1, pp. 291–295, 1997.
[15] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*. Cham, Switzerland: Springer, 2016, pp. 85–113.
[16] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE Access*, vol. 8, pp. 108952–108971, 2020.
[17] Y. K. Liu, S. K. Ong, and A. Y. C. Nee, "State-of-the-art survey on digital twin implementations," *Adv. Manuf.*, vol. 10, no. 1, pp. 1–23, Jan. 2022.
[18] J.-F. Yao, Y. Yang, X.-C. Wang, and X.-P. Zhang, "Systematic review of digital twin technology and applications," *Vis. Comput. Ind., Biomed., Art*, vol. 6, no. 1, p. 10, May 2023.
[19] E. VanDerHorn and S. Mahadevan, "Digital twin: Generalization, characterization and implementation," *Decis. Support Syst.*, vol. 145, Jun. 2021, Art. no. 113524.
[20] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP J. Manuf. Sci. Technol.*, vol. 29, pp. 36–52, May 2020.
[21] P. Aivaliotis, Z. Arkouli, K. Georgoulias, and S. Makris, "Methodology for enabling dynamic digital twins and virtual model evolution in industrial robotics—A predictive maintenance application," *Int. J. Comput. Integr. Manuf.*, vol. 36, no. 7, pp. 947–965, Jul. 2023.
[22] L. Wang, Y. Liu, H. Yin, and W. Sun, "Fault diagnosis and predictive maintenance for hydraulic system based on digital twin model," *AIP Adv.*, vol. 12, no. 6, Jun. 2022, Art. no. 065213.
[23] X. Zhang, B. Wu, X. Zhang, J. Duan, C. Wan, and Y. Hu, "An effective MBSE approach for constructing industrial robot digital twin system," *Robot. Comput.-Integr. Manuf.*, vol. 80, Apr. 2023, Art. no. 102455.
[24] Y. Wei, T. Hu, P. Yue, W. Luo, and S. Ma, "Study on the construction theory of digital twin mechanism model for mechatronics equipment," *Int. J. Adv. Manuf. Technol.*, pp. 1–19, Apr. 2022.
[25] M. Matulis and C. Harvey, "A robot arm digital twin utilising reinforcement learning," *Comput. Graph.*, vol. 95, pp. 106–114, Apr. 2021.
[26] K. Classens, W. P. M. H. Heemels, and T. Oomen, "Digital twins in mechatronics: From model-based control to predictive maintenance," in *Proc. IEEE 1st Int. Conf. Digit. Twins Parallel Intell. (DTPI)*, Jul. 2021, pp. 336–339.
[27] D. Adamenko, S. Kunnen, R. Pluhnau, A. Loibl, and A. Nagarajah, "Review and comparison of the methods of designing the digital twin," *Proc. CIRP*, vol. 91, pp. 27–32, Jan. 2020.
[28] M. Ibrahim, V. Rjabtšikov, and R. Gilbert, "Overview of digital twin platforms for EV applications," *Sensors*, vol. 23, no. 3, p. 1414, Jan. 2023.
[29] A. Regis, S. Arroyave-Tobon, J.-M. Linares, and E. Mermoz, "Physic-based vs data-based digital twins for bush bearing wear diagnostic," *Wear*, vols. 526–527, Aug. 2023, Art. no. 204888.
[30] B. H. Huynh, H. Akhtar, and M. K. Sett, "A universal methodology to create digital twins for serial and parallel manipulators," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 3104–3109.
[31] E. Kurvinen, "Physics-based digital twins merging with machines: Cases of mobile log crane and rotating machine," *IEEE Access*, vol. 10, pp. 45962–45978, 2022.
[32] Q. Khadim, Y. S. Hagh, L. Pyrhönen, S. Jaiswal, V. Zhidchenko, E. Kurvinen, J. Sopanen, A. Mikkola, and H. Handroos, "State estimation in a hydraulically actuated log crane using unscented Kalman filter," *IEEE Access*, vol. 10, pp. 62863–62878, 2022.
[33] V. Zhidchenko, I. Malysheva, H. Handroos, and A. Kovartsev, "Faster than real-time simulation of mobile crane dynamics using digital twin concept," *J. Phys., Conf. Ser.*, vol. 1096, Sep. 2018, Art. no. 012071.
[34] C. Schempp and S. Schulz, "High-precision absolute pose sensing for parallel mechanisms," *Sensors*, vol. 22, no. 5, p. 1995, Mar. 2022.
[35] Z.-F. Shao, X. Tang, and L. Wang, "Dynamics verification experiment of the Stewart parallel manipulator," *Int. J. Adv. Robotic Syst.*, vol. 12, no. 10, p. 144, Oct. 2015.
[36] S. A. Maged, A. A. Abouelsoud, A. M. R. F. E. Bab, and T. Namerikawa, "Stewart platform manipulator: State estimation using inertia sensors and unscented Kalman filter," in *Proc. 3rd Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Jul. 2016, pp. 1136–1140.
[37] I. Miletović, D. M. Pool, O. Stroosma, M. M. van Paassen, and Q. P. Chu, "Improved Stewart platform state estimation using inertial and actuator position measurements," *Control Eng. Pract.*, vol. 62, pp. 102–115, May 2017.
[38] P. Noskievic and D. Walica, "Design and realisation of the simulation model of the Stewart platform using the MATLAB-simulink and the simscape multibody library," in *Proc. 21st Int. Carpathian Control Conf. (ICCC)*, Oct. 2020, pp. 1–5.
[39] H. Hajimirzaalian, H. Moosavi, and M. Massah, "Dynamics analysis and simulation of parallel robot stewart platform," in *Proc. 2nd Int. Conf. Comput. Autom. Eng. (ICCAE)*, vol. 5, Feb. 2010, pp. 472–477.
[40] W. Zhu, M. Jiang, F. Yang, and X. Rui, "Modeling of a hexapod piezo-actuated positioning platform," *Multibody Syst. Dyn.*, pp. 1–24, Jun. 2023.
[41] G. Kazezkhan, B. Xiang, N. Wang, and A. Yusup, "Dynamic modeling of the Stewart platform for the NanShan radio telescope," *Adv. Mech. Eng.*, vol. 12, no. 7, Jul. 2020, Art. no. 168781402094007.
[42] F. Toso, R. Torchio, A. Favato, P. G. Carlet, S. Bolognani, and P. Alotto, "Digital twins as electric motor soft-sensors in the automotive industry," in *Proc. IEEE Int. Workshop Metrol. Automot. (MetroAutomotive)*, Jul. 2021, pp. 13–18.
[43] C.-J. Liang, W. McGee, C. C. Menassa, and V. R. Kamat, "Real-time state synchronization between physical construction robots and process-level digital twins," *Construct. Robot.*, vol. 6, no. 1, pp. 57–73, Mar. 2022.
[44] N. Menager, N. Worschech, and L. Mikelsons, "A toolchain for rapid control prototyping using rexroth controllers and open source software," in *Proc. Linköping Electron. Conf.*, Mar. 2014, pp. 371–380.
[45] N. Menager, R. Kampfmann, N. Worschech, and L. Mikelsons, "Suitability of different real-time solvers for a model-based engineering toolchain using industrial rexroth controllers," in *Proc. Linköping Electron. Conf. Proc.*, Sep. 2015, pp. 883–891.
[46] MathWorks. *Open Core Interface SDK for MATLAB and Simulink*. Natick, MA, USA. Accessed: Dec. 30, 2023. [Online]. Available: https://www.mathworks.com/products/connections/product_detail/open-core-interface.html.
[47] Bosch Rexroth AG. *MATLAB Reference (MLPI4MATLAB)*. Lohr am Main, Germany. Accessed: Dec. 30, 2023. [Online]. Available: https://apps.boschrexroth.com/docs/oci/mlpi4matlab/
[48] I. A. Bonev and J. Ryu, "A new approach to orientation workspace analysis of 6-DOF parallel manipulators," *Mechanism Mach. Theory*, vol. 36, no. 1, pp. 15–28, Jan. 2001.

[49] M. Mohammadi, E. Kurvinen, and A. Mikkola, ''A design process to parameterize a real-time simulation model of a commercial vehicle,'' *Int. Rev. Mech. Eng. (IREME)*, vol. 13, no. 12, p. 673, Dec. 2019.

[50] J. Czebe, P. Šuránek, J. Tuma, and D. Fojtík, ''Finding similarities in laser micrometers scanned data using cross-covariance,'' in *Proc. 20th Int. Carpathian Control Conf. (ICCC)*, May 2019, pp. 1–4.

[51] H. Palahalli, E. Ragaini, and G. Gruosso, ''Real-time smart microgrid simulation: The integration of communication layer in electrical simulation,'' in *Proc. 22nd IEEE Int. Conf. Ind. Technol. (ICIT)*, vol. 1, Mar. 2021, pp. 631–636.

[52] R. Eskola, H. Korpilahti, B. Bozorgmehri, M. K. Matikainen, and A. Mikkola, ''Real-time multi-body co-simulation model of a veneer peeling lathe,'' *Int. J. Comput. Integr. Manuf.*, vol. 36, no. 4, pp. 634–656, Apr. 2023.

[53] Y. Zan, D. Han, L. Yuan, M. Liu, and Z. Wu, ''Research on real-time simulation system of ship motion based on simulink,'' *Open Mech. Eng. J.*, vol. 8, no. 1, pp. 820–827, Dec. 2014.

DOMINIK WALICA received the M.Sc. degree in automatic control and engineering informatics from the VŠB—Technical University of Ostrava (VŠB-TUO), Czech Republic, in 2019. Since 2019, he has been a Junior Researcher with the Department of Control Systems and Instrumentation, Faculty of Mechanical Engineering, VŠB-TUO. His research interests include modelling and simulation of mechatronic systems, Stewart platform, and digital twin.

PETR NOSKIEVIČ was born in Ostrava, Czech Republic, in 1959. He received the Doctor (Associate Professor) degree, in 1992. After dissertation the VŠB-Technical University of Ostrava (VŠB-TUO), he worked for eight years in the industry in control of hydraulic drives, in 1987, where he has been an Associate Professor with the Faculty of Mechanical Engineering, since 1994. In 2001, he was a Full Professor in control of machines and processes and continued with VŠB-TUO. He delivers lectures from modelling and simulation of mechatronic systems, system identification, and control of fluid power actuators. He introduced the bachelor's and master's study programme mechatronics with VŠB-TUO. He is a supervisor and a guarantor of the Ph.D. Study Programme Control of Machines and Processes with the Faculty of Mechanical Engineering. His main research interests include modelling, simulation, and control of hydraulic drives. He was the Chairperson of the Organization Committee of the Czech Association for Hydraulics and Pneumatics, from 2001 to 2023, and a member of the REM Network for Research and Education in Mechatronics and Fluid Power Net International—FPNI, from 2003 to 2018 and from 2000 to 2014.