Model-based Engineering of an Arial Working Platform with Trajectory Control

Benedikt Müller*, Markus Beckers**, Thomas Mock** and Heiko Baum*

FLUIDON Gesellschaft für Fluidtechnik mbH, Jülicher Straße 338a, D-52070 Aachen, Germany*
meastream GmbH, Gartenstraße 38, D-52249 Eschweiler, Germany**
E-mail: info@fluidon.com

Who is not faced with the challenges of developing more complex products or features under greater time and cost pressure? Digitization has given rise to tools that support these requirements for the product development process, and they are now also becoming accessible to small and medium-sized companies with limited engineering resources. The paper presents the model-based development of an electro-hydraulically driven aerial work platform with trajectory control. The real-time capable simulation model is coupled with the control hardware, human-machine interfaces, and visualization in FLUIDON's Virtual Engineering Lab.

Keywords: Trajectory Control, Virtual Commissioning, Hardware-in-the-Loop HiL, Digital Twin
Target audience: Mobile Hydraulics, System Integration, Control Development

1 Introduction

Complex products, especially the control and drive systems of multifunctional mobile machines, can only be developed efficiently if the system integration of the various domains such as mechanics, hydraulics, electronics and control technology takes place early and continuously in the development process. A large aerial working platform whose platform is moved by a total of 8 hydraulically driven axes serves as an example (**Figure 1**).

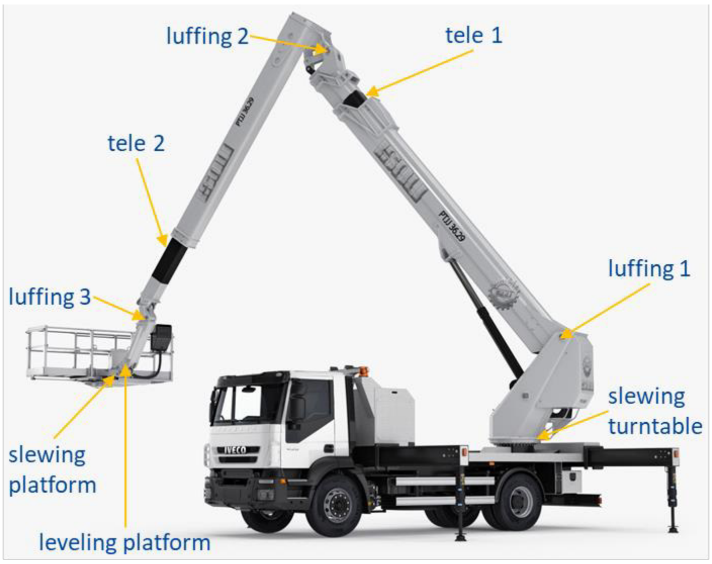

Figure 1: Motion axes of the mobile arial working platform

Even the basic design of such a machine raises many questions: The kinematics of the machine determine its movement possibilities. However, articulation points of cylinders, overlap of telescopic tubes, friction ratios and inertia forces also significantly influence the requirements on the hydraulics or the control system. These interfaces between domains often lead to problems in the development process when information is insufficiently synchronized or simply one side does not understand the other. One means of countering this interface problem is to integrate the system as early as possible.

Model-based development provides a toolbox for this purpose: The holistic system development is carried out based on a virtual representation (digital twin) of the machine, enabling the complex and interconnected tasks of system integration to be performed better and more efficiently. The example of the aerial working platform shows how the model-based approach holistically supports the development of complex machines (**Figure 2**).

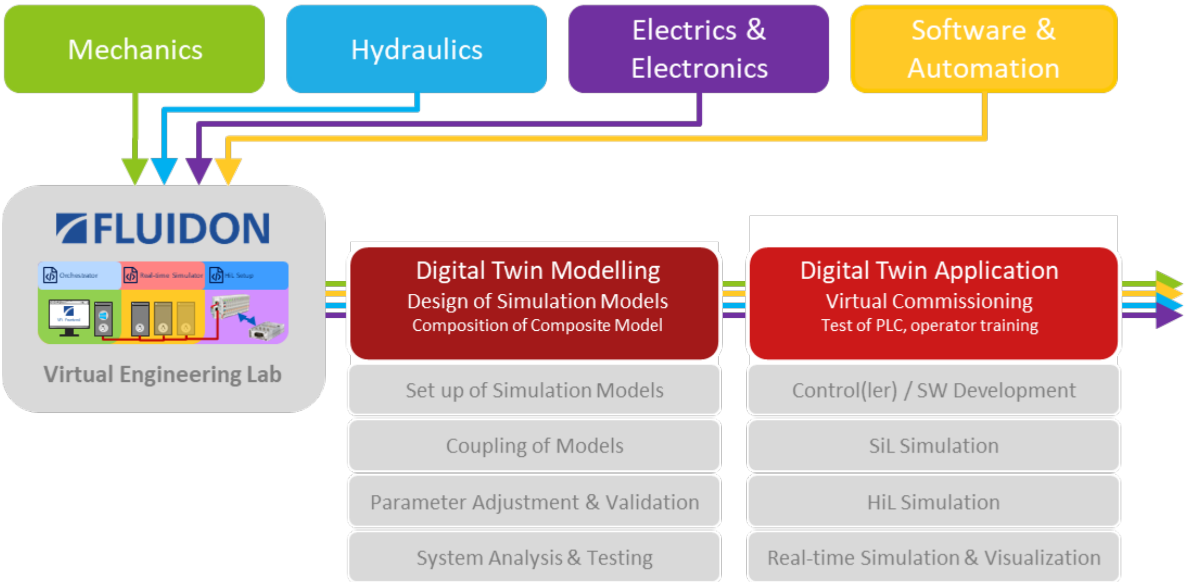

Figure 2: Cross-domain approach through model-based engineering

As the second major benefit, the model-based engineering supplies an easy-to-use environment for control development, which can be used to investigate the following questions, among others:

- Depending on the position and the desired direction of movement, up to 8 axes must be moved synchronously, making it difficult for the operator to directly actuate the axes. The control system should therefore convert a direction or path command of the operator into a combination of movement commands for the 8 axes.
- The required movements must be converted by the axis controllers of the 8 axes into control signals of the valves. High mass and friction forces and a large speed range from the slow continuous working movement to the fast infeed movement place high demands on these controllers. Their development must consider the complex interaction of mechanics (e.g. kinematics, friction conditions), hydraulics and control technology.
- Due to the limited installed power, not all movements can be executed simultaneously at their maximum possible speed. A module of the control system, operating between axis and path control, must therefore mediate sensible speeds between the participants when the power limit is reached.

2 Model based development

The first step in model-based development is to map the effects in the model that are decisive for its objectives. This should be done according to the principle "as simple as possible, as precise as necessary", since a more precise representation is usually also accompanied by a higher computing effort. On the other hand, real-time capability of the model is desired, if not a must, for control development.

2.1 From Model to Digital Twin

"A digital twin is a virtual representation that serves as the real-time digital counterpart of a physical object or process." [1]

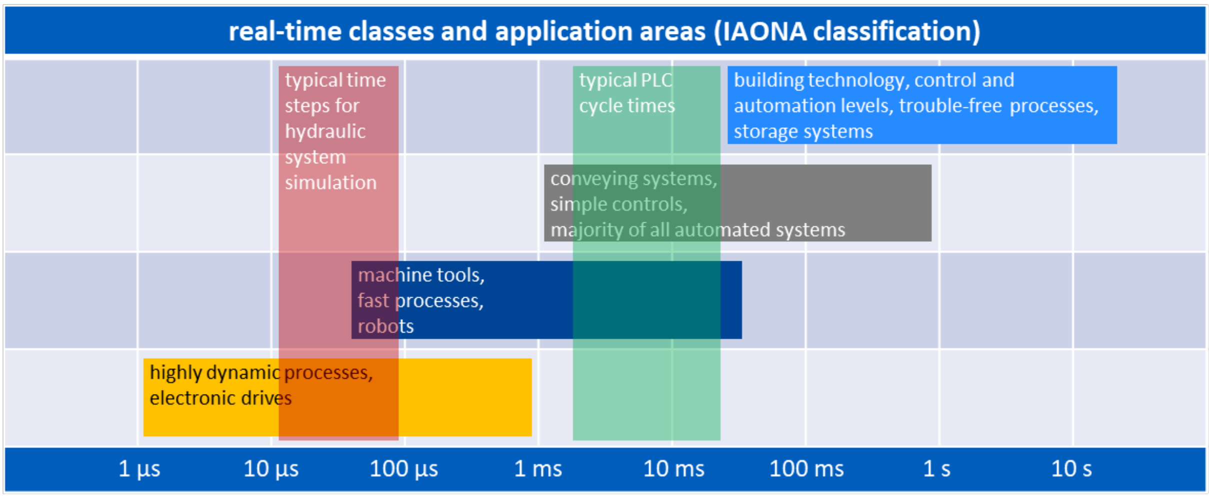

Figure 3: Real-time requirements for hydraulic system simulation vs. IAONA classes

This real-time capability is one of the major challenges in the development of a digital twin, particularly for complex hydraulically driven machines. Why is this especially true for hydraulic drives? The answer is found in the relatively small simulation step size (**Figure 3**). Typical step sizes for hydraulic simulation are in the order of 1e-6 to 1e-5 s, while mechanical multi-body simulation models, for example, often calculate with step sizes of 1e-4 to 1e-3 s with sufficient accuracy and numerical stability. The hydraulic systems, which are usually complex and highly interconnected, cannot therefore be easily simulated in real time, even with powerful computers.

One approach to making even complex models real-time capable lies in the parallelization of computational processes. However, it is important to use the right splitting: Since managing the threads also costs computational effort and time, parallel computation is not free. If parallelization is performed with a distribution that is too fine-granular, the time gained is very quickly eaten up by additional overhead.

The model shown below as an example is therefore not parallelized on the level of components or even individual computational processes, but at a module level. One module typically represents a functional unit, e.g. the hydraulic drive of an axis. Since the mechanical model of the aerial work platform, in contrast to the hydraulics, is already real-time capable as a complete model, it does not have to be modularized further, but is represented as one module in the overall model.

Figure 4 shows an example of the DSHplus model of a typical hydraulic drive axle used in aerial working platforms. The LS directional control valve, the counterbalance valves, the cylinder and the connecting lines can be seen. IO components exchange values with the other submodels, such as other hydraulic axes, the pressure supply or the mechanical model. A hydraulic connection is represented at this interface by the exchange variables pressure p and volume flow Q: The axis model receives the value of p as input and returns the calculated volume flow Q as output to the pressure supply. Similarly, a mechanical connection is described by the force output F and the inputs displacement x and velocity v, which are calculated in the mechanical model. [2]

The luffing axes of the boom parts generate rotational movements around the joints of the booms, combined with rotational movements in the joint eyes of the cylinders and, if necessary, additional levers. All these joints have in common that the frictional forces act on small lever arms. On the other hand, large masses and inertial torques act during these movements. Therefore, friction is neglected for these axes.

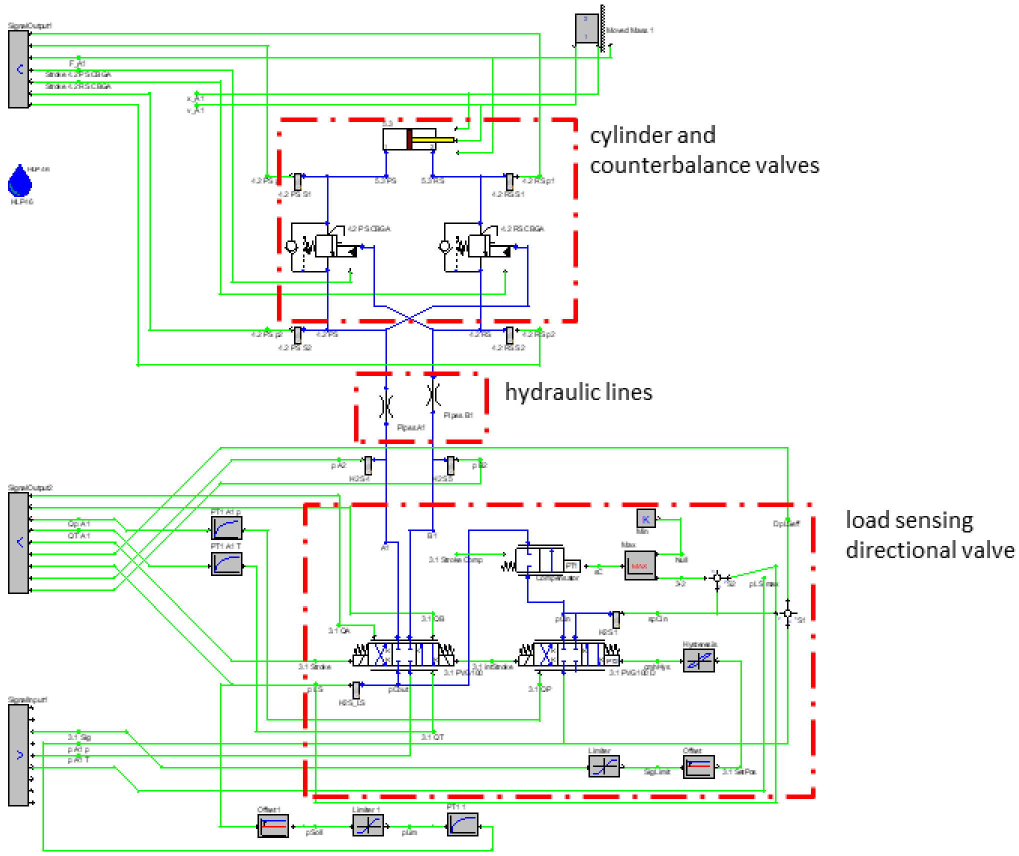

Figure 4: Submodel of a hydraulic axis

In the telescoping movements, on the other hand, the friction in the system has a significant influence on the design of the hydraulics and the axis controller and must therefore be considered in the model. The friction is not represented in the mechanical model, but in the submodels of the hydraulic axes. This has the advantage that their calculations are already performed with a small step size, thus the friction model does not lead to numerical instabilities. Friction is mapped according to the LuGre model [3]. This modeling approach represents the friction for small movements and direction changes better than, for example, a simple Stribeck model. At the same time, it offers better numerical stability and is therefore widely used in numerical simulation.

The partial model of an axis can be commissioned independently. This already reveals the first possible weaknesses of the technical realization, which can thus be corrected early in the development process and not only on the real prototype.

The partial models are exported as so-called Functional Mockup Units (FMU) according to the FMI 2.0 Co-Simulation standard from the respective simulation tool. "The Functional Mock-up Interface (FMI) is a free standard that defines a container and an interface to exchange dynamic models [...]" [4]. An FMU is a file containing a simulation model that provides the standardized FMI interface. An FMU created for co-simulation additionally contains its own solver.

The assembly of the complete machine from the modules created in this way takes place in the so-called composite model, **Figure 5**. The FMU of the mechanical submodel is marked, and the encapsulated mechanical model – created in OpdenModelica in the example – is shown. The other components labeled FMU in the figure each represent a hydraulic axis. The pressure supply is part of the overall model, but could also be encapsulated in a supplementary FMU.

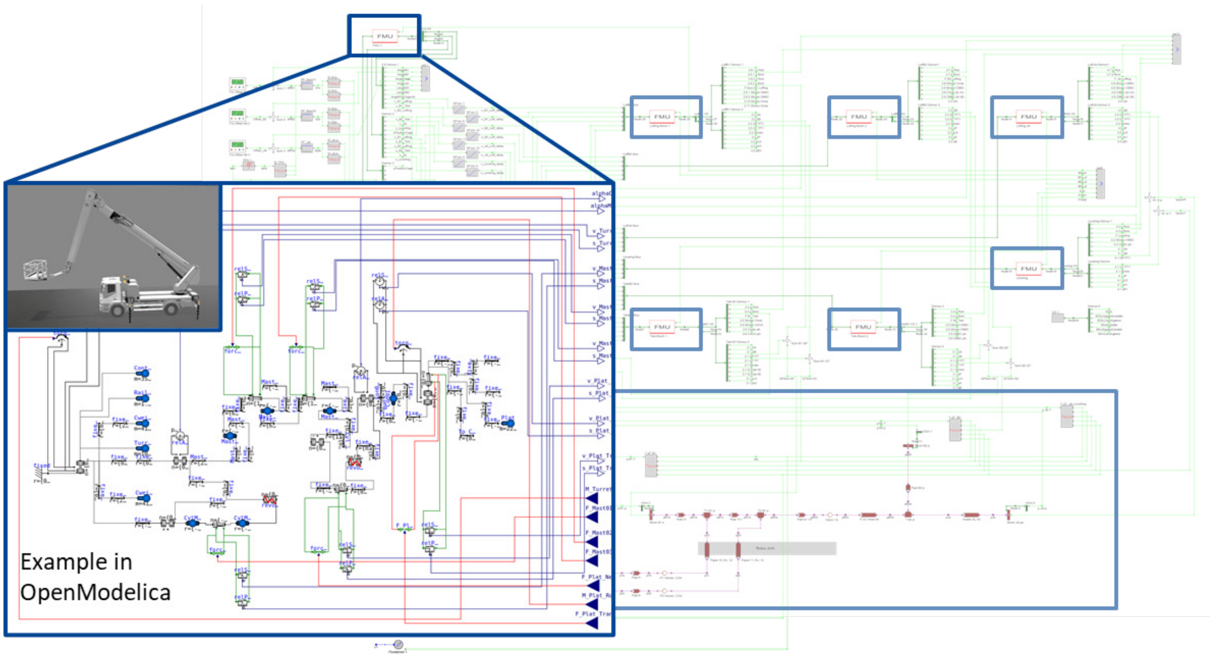

Figure 5: Composite model of the arial working platform

The FMUs are interconnected in DSHplus, which can perform their calculation in parallel. **Figure 6** shows once again how the modular, FMU-based model concept and parallelization save considerable computing time. This makes it possible to simulate the entire model of the aerial work platform shown in the example in real time.

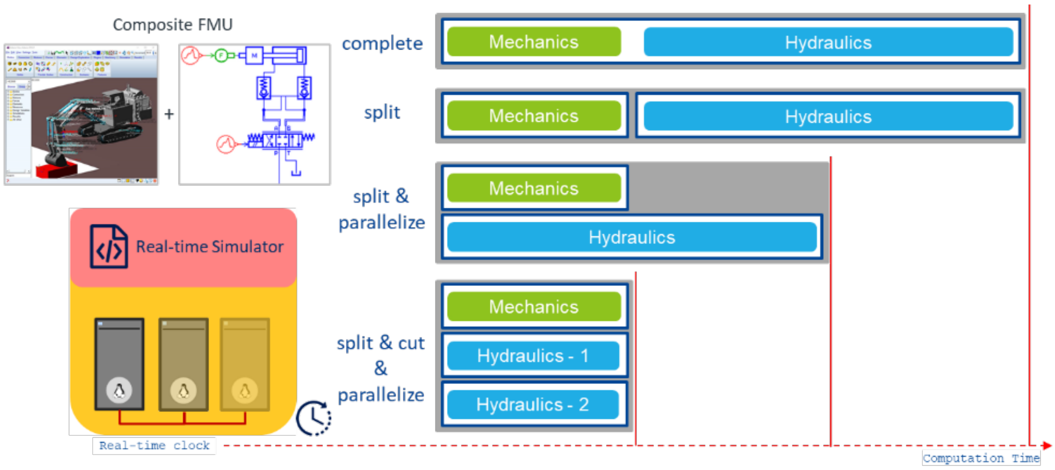

Figure 6: Real-time optimization through parallel computation of the modular model

The FMU-based modeling approach offers, besides the possibility of parallel computation, other important advantages:

- With FMI 2.0, a standardized interface is used. Therefore, all FMUs that comply with this standard can be included in the overall model, regardless of the tool used to create them. An example is the mechanical partial model of the aerial work platform created in OpenModelica and connected to the hydraulic FMUs in the DSHplus Composite Model. These could be generated in other tools as an alternative to DSHplus. In this way, each domain can work with the tool of its choice.
- Not only the calculation can be parallelized: The modular structure with a subdivision into functional units allows the submodels to be developed and put into operation simultaneously.

- Modules can be reused in other machines or variants. This creates a construction kit from which, for example, a new development can draw: The development effort is reduced, "unnecessary" variants are avoided and existing know-how is used more systematically.

2.2 Real-time simulation environment of the Virtual Engineering Lab

The composite model is again exported as FMU and loaded to the real-time environment of the Virtual Engineering Lab (VEL). The VEL makes the IOs of the model available via EtherCAT. Analog or digital input and output terminals or fieldbus terminals, e.g. CAN, connected to the EtherCAT enable very fast and flexible coupling with the control hardware in a Hardware-in-the-Loop (HiL) setup (**Figure 7**).

Figure 7: VEL real-time setup

As in the real machine, the joysticks are connected to the control hardware via CAN. A large number of sensors provide further input variables for the control system, such as the relative angles of the booms, the telescoping lengths or the deviation of the platform from the horizontal position. Within the HiL environment, the simulation model provides these quantities. In contrast to the real machine, however, these quantities are not output via CAN and transmitted to the machine controller. To simplify the setup, the coupling between simulation environment and control hardware is currently done via OPC-UA. Although communication via this interface does not guarantee hard real-time, it is sufficiently performant in the local network to ensure data transmission in the cycle time of the machine control and faster. In the next planned development step, this data transmission will take place via CAN - as in the real machine - so that the controller can be installed 1:1 in the real machine.

2.3 Visualization

Up to now, the visualization of simulation results has mostly been in the form of graphs representing the quantities of interest as a function of time. However, a much more intuitive access to the results of a simulation or the behavior of the virtual prototype in certain situations is provided by the 3D visualization, which can be coupled with the real-time simulation in the VEL. If the control system is linked with realistic operating elements, the virtual prototype already enables initial user tests during development.

Very simple visualization models that merely reflect the kinematics of the machine are already sufficient for the developer. These simple kinematics models can be displayed with limited effort in the early development phase, even before a CAD model is available. However, initial CAD drafts also form a good basis for a visualization model for system development.

Very simple visualization models that merely reflect the kinematics of the machine are already sufficient for the developer. These simple kinematics models can be generated with limited effort in the early development phase, even before a CAD model is available. However, initial CAD designs also form a good basis for a visualization model for system development.

For marketing purposes, the models are prepared in a more visually elaborate manner and, if desired, combined with a virtual environment. For this application, but also for virtual user tests of the machine and the man-machine interfaces, it is often useful to use VR glasses instead of the classic visualization on the screen, which can also be combined with the VEL.

The 3D visualization in the Virtual Engineering Lab is done with a Unity Engine [5], actually a runtime and development environment for games, which has been integrated into the web interface of the VEL. The variables required for the visualization are provided by the real-time simulation environment via OPC-UA.

3 Control development

Now that the tools for virtual control development and commissioning are available, the prototype control system for the aerial working platform is being developed with the help of the VEL.

3.1 Platform leveling

The control of today's aerial work platforms is still little automated in normal working operation. Depending on the operating situation, the control system limits the operating speeds or prevents certain movements, e.g. for safety reasons. Only the parallel guidance or leveling of the platform takes place in a closed control loop.

The required actuating speed of the cylinder for level control can be calculated from the angular changes of the boom sections if these are known. Based on this consideration, a level controller was developed that essentially determines the actuating signal from the angular velocities of the booms. A classical PI controller provides only minor necessary corrections. The control signals of the corresponding directional control valves are used to predict the angular velocities of the booms. Their characteristic curves and the kinematics of the booms (cylinder stroke vs. angle) are stored in the control system, which can thus predict the boom speeds and use them to determine the feedforward setpoint for the level control valve.

Figure 8 shows the results of the new levelling control in comparison with a classic PI controller. Despite its high control gain, which leads to instabilities at the beginning of the movement, the control deviations of a pure PI controller are comparatively high. In contrast, the new controller with feedforward control, whose PI controller has a significantly lower control gain, behaves stably in all situations with lower control deviation at the same time.

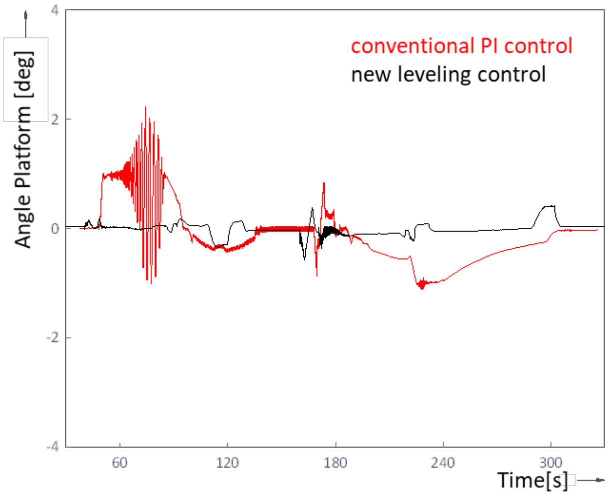

Figure 8: Leveling control results

3.2 Trajectory control

In today's standard operation of the aerial working platform, the user controls the individual axes separately. This is typically done using joysticks. In the case of a working platform with a multi-section boom, not all axes can be operated simultaneously simply because of the limited number of independent axes of a joystick. Coordinated simultaneous movements of the axes are therefore only possible to a limited extent.

Continuous and jerk-limited movement of the platform along a given trajectory is therefore difficult or even impossible for a human operator. The current development of a prototypical new control system therefore aims to provide the user with an easy-to-use working platform that translates a directional or trajectory command from the operator into a combination of motion commands for the axes. The movement along a fixed trajectory can be imagined, for example, as a rectilinear movement of the lifting platform along a window front.

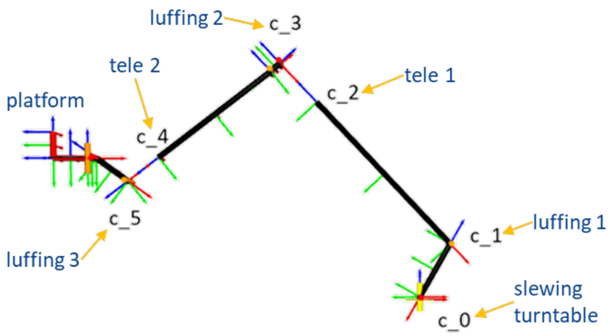

Figure 9: Kinematics model

Figure 9 shows the schematic representation of the kinematic model, which is the starting point for the calculation of the trajectory. The parameters c_i represent the rotation or translation of the axes. A parameter set C contains these parameters c_i and thus describes a position of the platform. With known parameter sets C it is a simple procedure to calculate the resulting position of the platform (forward calculation). However, for the desired application the reverse way is necessary: In order for an operator to move a working platform in space, a parameter set C must be determined for each position of the platform, which contains a parameter c_i for each axis. This is an inverse problem.

To solve this problem, the downhill simplex method of John Melder and Roger Mead was used for initial feasibility tests. [7]. This local optimization method, which does not require derivative information, iteratively computes for each point P_i = (x_i, y_i, z_i) of the nominal geometry the current parameter set c_i required for this position, using the configuration for the previous point P_(i-1) as the starting value of the optimization. With respect to the required real-time computation, in the future one will probably resort to "simpler" algorithms that can be better implemented in mobile controls, such as Newton minimization [8].

The following example shows how the time course of C results for a given trajectory, which is then provided to the axis controllers as a setpoint. The two left graphs in **Figure 10** show the initial position, the two right graphs the final position of a movement along a straight line in space.

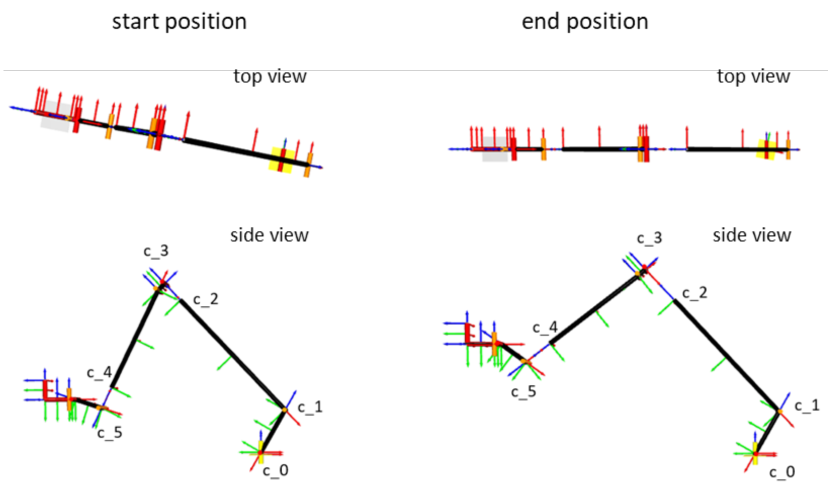

Figure 10: Start and end position of the trajectory

All parameters c_i must be adjusted simultaneously to realize this movement of the platform. The algorithm now calculates the time course of c_i, where various optimization criteria are possible, such as:

- Path optimization
- Optimization hydraulic system
- Jerk-limited movement

If one wants to move the platform from the start point to the end point of the trajectory in a straight line, the optimization calculates the parameters c_i, as shown in **Figure 11**.

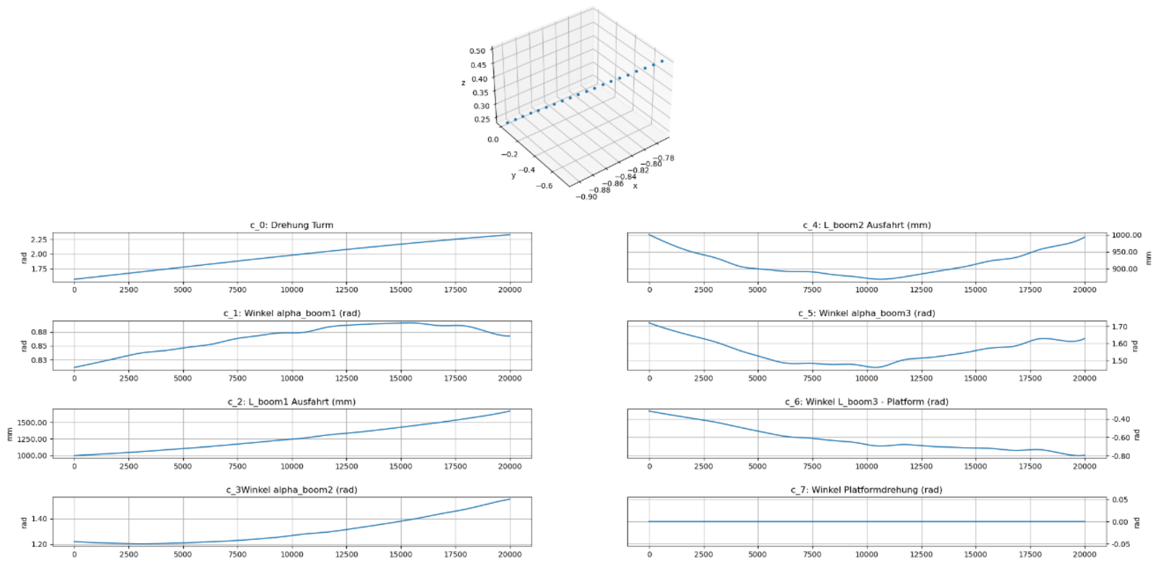

Figure 11: Time course of the parameters c_i for the desired linear trajectory

These desired positions and speeds for each axis are the setpoints for the axis controllers, which generate the control signals of the valves based on these values and the positions (or angles) currently measured on the machine.

The operation of the working platform can now be greatly simplified. If classic joysticks with two axes are used, the left joystick can, for example, specify the x and y directions in space, while the right joystick controls the movement in the vertical z direction and the rotation of the platform. With a 3D joystick, one-handed operation is also possible (**Figure 12**).

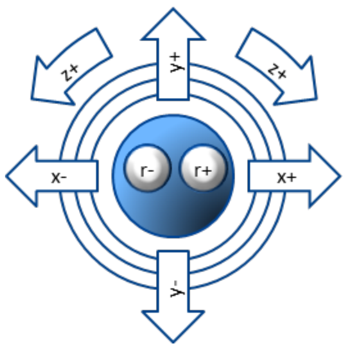

Figure 12: Possible joystick assignment for one-hand operation

With the help of the VEL-RT setup, the development and commissioning of the web control system can largely be carried out "at the desk" - with significantly reduced effort compared to development on the real prototype. In addition to the time and cost savings, one further advantage of virtual testing (**Figure 13**) is the ease of implementing reproducible test sequences, so that, for example, the control quality can be better compared for different parameters.

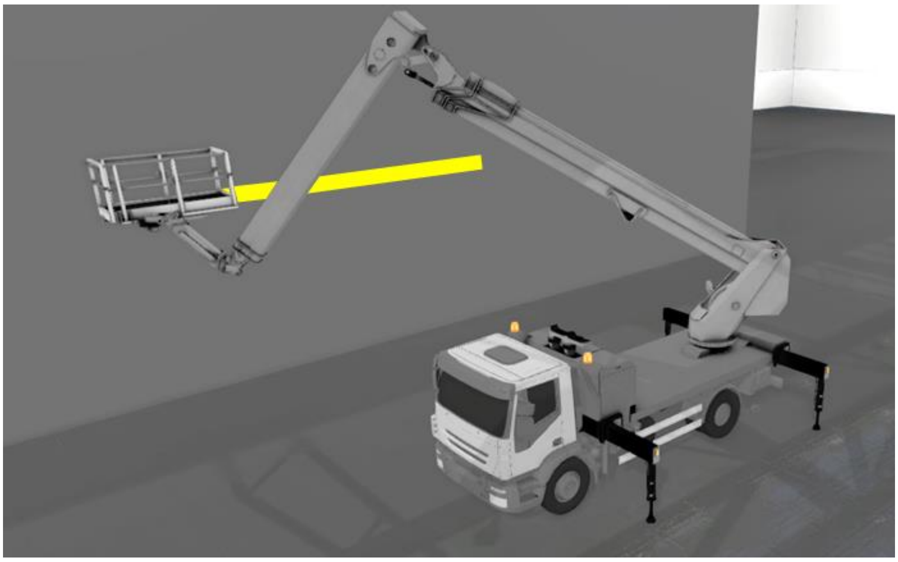

Figure 13: Visualization of a virtual trajectory test where a line is drawn on a wall

4 Summary and Conclusion

The modeling and use of a digital twin for control development was demonstrated using the example of the multi-axis boom of a large aerial work platform. The model-based development approach forces - in a positive sense - an early system integration of all technical domains involved. Thus, the mechanics, hydraulics, electronics and machine control are already tested in the virtual overall system during the concept phase and potential issues are identified early in the development process. **Figure 14** summarizes these advantages once again on the left.

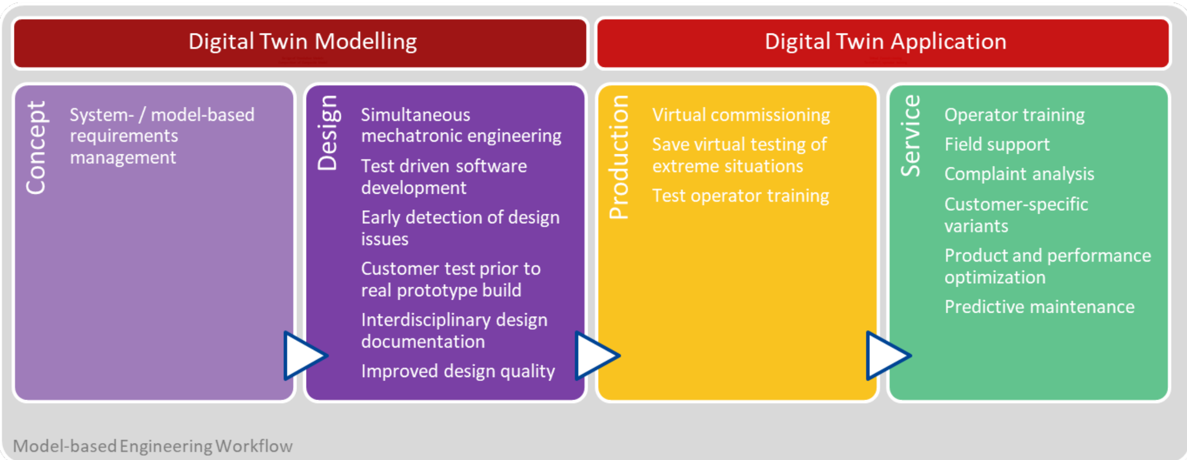

Figure 14: Model-based engineering workflow

The resulting digital twin also accompanies the further product life cycle (Figure 14, right part). Taking the aerial work platform as an example, the Virtual Engineering Lab (VEL) was used to develop a prototype of a 3D trajectory control system that considerably simplifies the operation of the aerial work platform. In the operation phase, the model supports the analysis of complex problems in the field or helps to prepare a retrofit of modernized controls for machines or plants. The virtual representative of the machine is (almost) always and everywhere available. This is particularly advantageous in the case of cross-site development or limited availability of real prototypes and reduces downtimes at the customer's site in the event of service.

References

[1] N. N., *Digital twin - Wikipedia*, https://en.wikipedia.org/wiki/Digital_twin, visited on March 15, 2022.

[2] Völker, L., *Investigation of the communication interval in coupled simulation*, PhD Thesis Karlsruhe Institute of Technology (KIT), 2011.

[3] de Wit, C. C., Olsson, H. , Aström, K.J.. , Lischinski, P., *A New Model for Control of Systems with Friction,* In: IEEE Transactions on Automation Control, Vol. 40, 1995.

[4] N. N., *Functional Mockup Interface*, https://fmi-standard.org/, visited March 15, 2022.

[5] N. N. , *Unity (game engine) - Wikipedia*, https://de.wikipedia.org/wiki/Unity_(game-engine), visited March 16, 2022.

[6] Tarantola. A.., *Inverse Problem Theory and Methods for Model Parameter Estimation*, Society for Industrial and Applied Mathematics, 2004.

[7] Nelder, J. A. , Mead, R. , *A Simplex Method for Function Minimization*, The Computer Journal, Vol. 7, 308-313 , 1965.

[8] Polyak, B. T. , *Newton's method and its use in optimization*, European Journal of Operations Research, 181(3), 1086-1096 , 2007.