DOI: 10.2195/lj_Proc_kivelae_en_201811_02
URN: urn:nbn:de:0009-14-47470

Simulation model for the verification of a safety-related control system in a hoist application

Simulationsmodell für die Verifikation einer sicherheitsbezogenen Hubwerksteuerung

Tommi Kivelä
Steffen Bolender
Markus Golder
Kai Furmans

Institut für Fördertechnik und Logistiksysteme (IFL)
Karlsruher Institut für Technologie (KIT)

Development of safety-related control systems requires extensive testing to verify correct functionality of the developed system in all circumstances. In this work, a simulation model of a novel hoist concept is shown. The simulation model consists of a complete electromechanical model of the drivetrain, including the pole-changing induction motor, slipping clutch and the brake, coupled with a two-mass spring-damped model of the hoist, the load and the crane bridge. The simulation model enables model-assisted testing of a hoist supervision unit currently being developed as part of a research project. We discuss the taken approach to software testing using the developed model.

[Keywords: Hoist, Simulation model, Functional Safety, Software testing]

Die Entwicklung von sicherheitsbezogenen Steuerungen erfordert umfangreiche Tests, um die korrekte Funktion des entwickelten Systems unter allen Umständen zu überprüfen. In dieser Arbeit wird ein Simulationsmodell eines neuartigen Hebezeugkonzepts gezeigt. Das Simulationsmodell besteht aus einem kompletten elektromechanischen Modell des Antriebsstrangs einschließlich polumschaltbarem Induktionsmotor, Rutschkupplung und Bremse, gekoppelt mit einem Zwei-Massen-Feder-Dämpfer-Modell des Hubwerks, der Last und der Kranbrücke. Das Simulationsmodell ermöglicht die modellgestützte Verifikation einer im Rahmen eines Forschungsprojektes entwickelten Hebezeug-Supervisionseinheit. Wir besprechen die Vorgehensweise beim Software-Testen anhand des entwickelten Modells.

[Schlüsselwörter: Hebezeug, Simulationsmodell, Funktionale Sicherheit, Software Tests]

1 INTRODUCTION

Development of safety-related systems, especially for the European marketplace, is subject to a multitude of legal and regulatory requirements. In Europe, the Machinery Directive [EC06] sets the base requirements for building safe machinery. The harmonized standard ISO 12100 provides guidance for the safety risk assessment and reduction for machinery. If the risks cannot be reduced to an acceptable level through inherently safe design measures, additional technical protective measures are needed. Different functional safety standards, such as the machinery-specific harmonized ISO 13849-series [ISO13849-1], the IEC 62061 [IEC62061], or the generic IEC 61508-series [IEC61508-1], provide guidance and requirements for the development of safety-related control systems.

Simulation models enable testing the functionality of a system under development without the complete machine or even the target hardware, for which the system is being developed. While the functional safety standards dictate strict verification requirements for new safety-related control systems, utilizing software- or hardware-in-the-loop testing with the help of simulation models enables testing the system under development extensively already in simulated environment, saving development time and increasing overall software quality.

In this work, we present a simulation model of a novel hoist concept, which includes a slipping clutch. The slipping clutch improves the inherent safety of the machine, but an electronic supervision unit can still be used for additional risk reduction. The simulation model includes a complete electromechanical model of the drive train, with an induction motor, the clutch, the brake, coupled with a two-mass spring-damper model of the hoist, the load and the crane bridge. Additionally, we present possible approaches for utilising the model to assist in verifying the safety-related software for a hoist supervision unit.

The remainder of the work is structured as follows. In chapter 2 we provide a background for the present work and summarize the relevant functional safety requirements for hoist applications. In chapter 3, we introduce the developed simulation model and its validation. The approach taken to model-based testing is discussed in chapter 4, with initial results shown in chapter 5. We summarize the present work and discuss further work in chapter 6.

2 BACKGROUND

Type-C product standards provide guidance on the safety requirements for specific types of machinery. In this section we shortly introduce the project, to which the current work is related to, as well as summarize the relevant base requirements for functional safety in hoist applications and for software verification.

2.1 PROJECT FOLSA

A new mixed-criticality controller is developed in ZIM-Project FOLSA ("Future-oriented Logistics Safety Application"). The controller includes safety-related and non-safety-related subsystems, the first to support the implementation of safety-related functions, the second for standard control functionality as well as for communication. The controller is targeted especially at material handling applications as a compact, embeddable and cost-efficient solution to enable safety and non-safety-related functions on the same platform, with secure IoT-capabilities. The first application prototype using the FOLSA-platform will be a hoist supervision unit. The controller concept was previously published in [HKV+16].

2.2 FUNCTIONAL SAFETY REQUIREMENTS IN HOIST APPLICATIONS

The type C standards [EN15011], [EN14492-2] and [EN13135] can be used to derive the functional safety requirements for hoist applications. The requirements can be summarised as follows, when considering a safety-related control system with electronic or programmable devices:

- Standard crane applications: PL c and at least category 2.
- High-risk applications: PL d and at least category 3.

Thus for the development of a hoist supervision unit, the target should be development according to the requirements for PL d as defined in [ISO13849-1]. In order to cover all markets, a device might also be typically designed to fulfil the requirements of the IEC 62061 for a SIL, which roughly corresponds to the required PL. In this case this would mean, that the unit should be developed targeting PL d and SIL 2.

For safety-related embedded software (SRESW) according to PL d, the [ISO13849-1] requires "extended functional testing, e.g. grey box testing, performance testing or simulation". For SRESW, the IEC 62061 requires fulfilling the requirements of the IEC 61508-3 according to the targeted SIL [IEC62061]. In IEC 61508-3, the following measures are highly recommended (required) for SIL 2:

- Software module testing and integration: Dynamic analysis and testing, data recording and analysis, functional and black box testing as well as the use of test automation tools
- Software aspects of system safety validation: Functional and black box testing
- For dynamic analysis and testing (Under module and integration testing): Test execution from boundary value analysis and 100% statement coverage.

The list above is not comprehensive, but contains the relevant requirements for the concept being discussed here. [IEC61508-3]

Similarities can be seen in the requirements from [ISO13849-1] and [IEC61508-3]. The main motivation for the development of the simulation model presented in this work is to enable functional and black box testing of the safety-related software against a realistic simulated process model coupled with automated testing tools. Simulative environment can ease the other goals as well, e.g. for easily testing corner cases to achieve high statement coverage. In addition to software-in-the-loop (SiL) testing in a simulative environment, the simulation model can later be used for hardware-in-the-loop (HiL) testing with the help of a HiL-Simulator. The approach taken in software testing with the help of the simulation model is discussed in section 4.

3 ELECTROMECHANICAL MODEL

In this section, we introduce the developed simulation model from the ground up, starting from the load interacting with the hoist and the crane bridge. Afterwards we introduce the drive train model with the brake, the clutch and the pole-changing induction motor.

3.1 THE BRIDGE, THE HOIST AND THE LOAD

The bridge, the hoist and the load are modelled as a spring damper model as described in [Gol04].

Hereby two phases of the system are considered. At the beginning of the simulation, the system is in rest. This usually means that the load is on the ground. In *Phase I* the motor is switched on and the chain is pre-stressed. When the force in the chain reaches the value of the gravitational force of the load, the load is lifted from the ground and

DOI: 10.2195/lj_Proc_kivelae_en_201811_02
URN: urn:nbn:de:0009-14-47470

Phase II is reached. The load than starts oscillating. (cf. Figure 1)

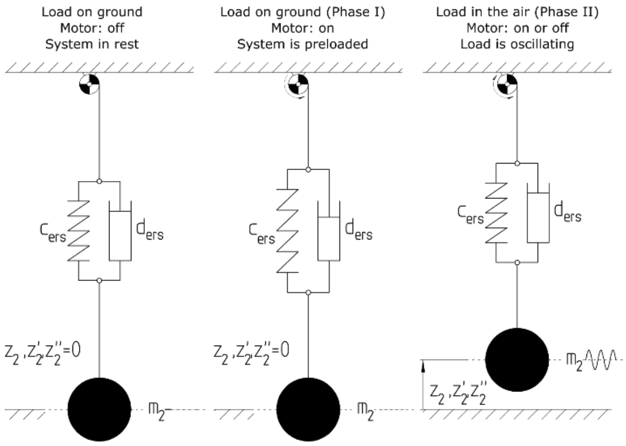

Figure 1. Different phases when lifting a load from the ground in a one-mass spring damper model. Translated version based on [Gol04].

For the simulation, a two-mass spring damper model is used. The system is reduced to two point masses. These are the pay load m_2 and the reduced equivalent mass of the bridge and the hoist m_1. The masses are coupled by elements which are considered as ideal springs and dampers. The values for the springs are derived from the elasticity of the crane bridge c_1 and the chain c_2. For the damping factors the values d_1 and d_2 are used. As an external force the chain drive force is induced into the system by the motor driving the chain. The acting forces for the Phases I and II are shown in Figure 2. Evaluating the balance of forces leads to the following equations of motion.

Phase I – System is preloaded:

The equation of motion for the equivalent load of crane bridge and hoist m_1 is shown in equation (1). As the pay load m_2 is still on the ground there is no equation of motion for m_2.

$$m_1 \cdot z_1''(t) + (d_1 + d_2) \cdot z_1'(t) + (c_1 + c_2) \cdot z_1(t) - c_2 \cdot S_I(t) = 0 \quad (1)$$

The force in the chain can then be calculated as given in equation (2).

$$F_1(t) = c_2 \cdot S_I(t) - c_2 \cdot z_1(t) - d_2 \cdot z_1'(t) \quad (2)$$

Phase II – Load is oscillating:

In Phase II the load is lifted off the ground, so there are two equations of motion for m_1 (Equation (3)) and m_2 (Equation (4)).

$$m_1 \cdot z_1''(t) + (d_1 + d_2) \cdot z_1'(t) + (c_1 + c_2) \cdot z_1(t) + d_2 \cdot z_2'(t) + c_2 \cdot z_2(t) - c_2 \cdot S_{II}(t) = 0 \quad (3)$$

$$m_2 \cdot z_2''(t) + d_2 \cdot z_2'(t) + c_2 \cdot z_2(t) + d_2 \cdot z_1'(t) + c_2 \cdot z_1(t) - c_2 \cdot S_{II}(t) + m_2 \cdot g = 0 \quad (4)$$

The equation for calculating the force in the chain is given in equation (5).

$$F_{II}(t) = c_2 \cdot S_{II}(t) - c_2 \cdot z_1(t) - d_2 \cdot z_1'(t) - c_2 \cdot z_2(t) - d_2 \cdot z_2'(t) \quad (5)$$

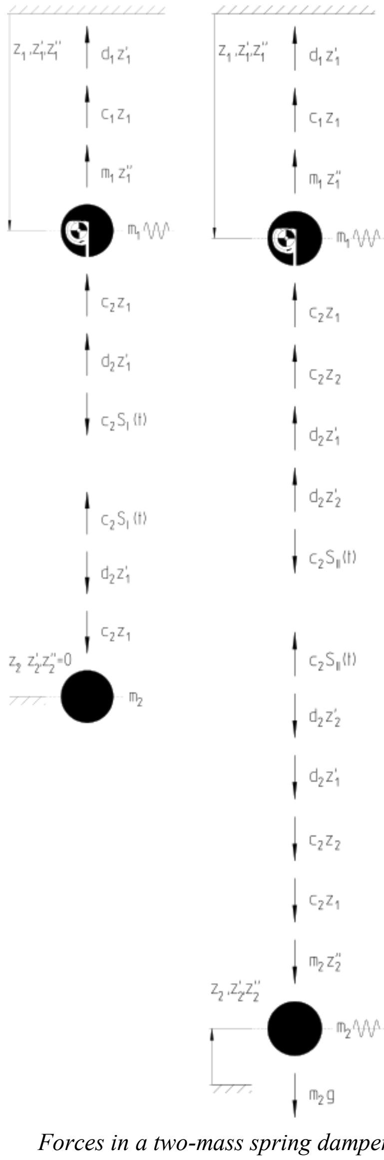

Figure 2. Forces in a two-mass spring damper model for Phase I (left) and Phase II (right) [Gol04]

DOI: 10.2195/lj_Proc_kivelae_en_201811_02
URN: urn:nbn:de:0009-14-47470

3.2 Drivetrain

The drivetrain, illustrated in Figure 4, consists of an induction motor, a slipping clutch, a brake and a gearbox. The coupling is positioned directly after the motor to separate the motor shaft and the gearbox shaft in case of an overload. The brake is positioned on the gearbox input shaft and is applied when the motor is off and the load needs to be held in position.

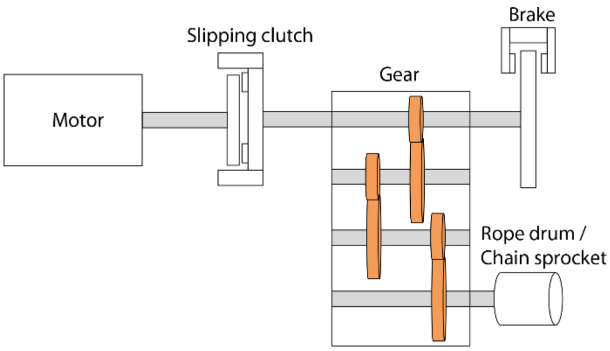

Figure 3. Model of the elements in the hoist drive train.

The model uses a total efficiency factor μ for the whole drive train and a total moment of inertia which is calculated based on the inertia of the single components and the translation ratio of the gearbox.

In the following section the modelling of the single components are explained in detail.

3.2.1 Induction motor

Even though frequency converters are commonly used in the industry, hoists are still more commonly equipped with a simple contactor control circuitry controlling a pole-changing (two-speed) induction motor due to high reliability, low costs, and ease of use. In this section, we present a model for a pole-changing induction motor drive, which will be used as a basis for this work. Detailed modelling of a frequency converter driven hoist motor was previously introduced in [KG16], and can also be used with the complete model as presented in this work.

The induction motor is modelled using space vectors. Space vector modelling allows for the modelling of a three-phase electrical system as a single-phase complex-valued system. For example, the space vector of the stator voltage in the stator coordinate system is shown in equation (6), where u_a. u_b and u_c are the instantaneous phase voltages and j is the imaginary unit.

$$\underline{u}_S^s(t) = \frac{2}{3}\left(u_a(t) + e^{\frac{j2\pi}{3}}u_b(t) + e^{\frac{j4\pi}{3}}u_c(t)\right) \tag{6}$$

The induction motor model shortly described here is the Inverse-Γ model [Sle89], where a single stator-rotor pair can be modelled as shown in the equations (7)-(8), where $\underline{i}_S^s$ and $\underline{\psi}_R^s$ are the stator current and rotor flux vector in the stator coordinate system and T_m the torque generated by the motor. The ω_r is the angular velocity of the rotor flux vector, which is related to the motor mechanical angular velocity ω_m: $\omega_r = p\omega_m$, where p is the number of motor pole pairs. R_s and R_R are the stator and rotor resistance, and L_M and L_σ are the magnetizing and total leakage inductance of the motor.

$$\frac{d\underline{i}_S^s}{dt} = \frac{1}{L_\sigma}\Big(\underline{u}_S^s - (R_s + R_R)\underline{i}_S^s - \left(j\omega_r - \frac{R_R}{L_M}\right)\underline{\psi}_R^s\Big) \tag{7}$$

$$\frac{d\underline{\psi}_R^s}{dt} = R_R\underline{i}_S^s + \left(j\omega_r - \frac{R_R}{L_M}\right)\underline{\psi}_R^s \tag{8}$$

$$T_m = \frac{3p}{2} Im\{\underline{\psi}_R^{s*}\underline{i}_S^s\} \tag{9}$$

According to Munoz and Lipo, dual-pole induction motors can be modelled as two uncoupled pairs of windings, if the number of poles in the winding pairs differ [ML00]. For a pole-changing motor the amount of poles per winding are different, since the motor is designed to switch between fast and slow mechanical speeds from a constant frequency grid, based on which stator coil is being supplied. In this work we thus modelled the pole-changing motor as two sets of stator-rotor-coil pairs, described in equations (7)-(9), but interacting with the same mechanical system, the torque of both summed as the complete motor torque in equation. For the two windings, the motor parameters (R_s, R_R, L_M, $L_\sigma$and p) differ. In normal hoist operation the two hoist windings are never energized simultaneously, this was also asserted in the motor simulation model.

3.2.2 Gearbox

The model uses a simple single step spur gearing and only uses the total translation ratio as the single translation steps have no impact on the model behaviour, they only need to be considered in calculating the moment of inertia. The gear is positioned directly next to the chain sprocket, so that the slipping clutch and the brake are impacted with the torque and rotational speed of the motor.

The gear transforms the revolution speed n and the torque T based on the given ratio i of the gears as given in equations (7) and (8) where n_1 and T_1 are on the motors side and n_2 and T_2 are on the chain side.

$$T_2 = i \cdot T_1 \tag{10}$$

DOI: 10.2195/lj_Proc_kivelae_en_201811_02
URN: urn:nbn:de:0009-14-47470

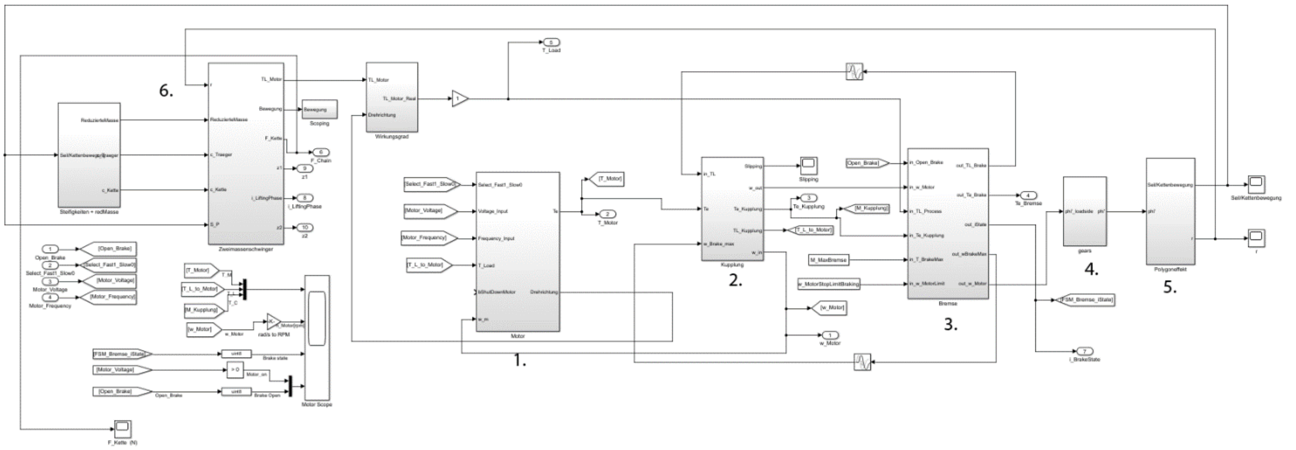

Figure 4. Electromechanical model overview. The numbered elements are 1: Motor, 2: Slipping clutch, 3: Brake, 4: Gear, 5: Polygon effect and 6: Two-mass spring-damper model.

$$n_2 = \frac{n_1}{i} \tag{11}$$

3.2.3 SLIPPING CLUTCH

The simulation model uses a slipping clutch to prevent damage to the system in case of an overload of the drive train.

The maximum torque for the slipping clutch can be controlled by the force acting on the friction plates. This slipping torque is based on the static friction in case of a locked clutch and on dynamic friction for a slipping clutch. As long as the transmitted torque is smaller than the slipping torque, the clutch will act as a simple connection between the two shafts. In case the motor torque is higher than the slipping torque, the maximum transmitted torque is limited to the slipping torque, the remaining torque will dissipate into heat and is removed from the model. The rotational speed of the input and output shaft can then be calculated based on the moment of inertia of the two parts. [HB14]

3.2.4 BRAKE

To hold the load in position while the crane is moving, a brake is needed. When the brake is closed it will hold the complete torque coming from the load and at the same time prevents the motor shaft from turning. The brake is usually synchronized with the motor control, so that it is only closed when the motor is turned off.

The brake is modelled with four phases:

- Brake is closed
- Brake is opening
- Brake is opened
- Brake is closing

Similar to the clutch, in the case of an open brake, the complete torque and speed will be transmitted through the clutch. In case the brake is closed or not completely open, it will apply a counter-torque, which eventually will slow down the system to a rotational speed of zero. The counter-torque is limited by the braking torque of the brake.

3.2.5 POLYGON EFFECT

When using a chain for lifting the polygon effect needs to be considered. In contrary to ropes, chains consist of single elements that are rotated by 90° to each other, so that a chain sprocket is needed to ensure a smooth winding of the chain. Due to the shape of the chain and the sprocket, the equivalent radius changes during the rotation of the sprocket. The simulation of this effect is based on the models developed in [Lan04].

3.3 MODEL OVERVIEW

An overview of the implemented electromechanical simulation model in Matlab Simulink is shown in Figure 4. The main inputs to the model are the induction motor control signals, which are the input voltage and the control signal for selecting the fast or the slow stator coils, as well as the brake control signal. The motor model (marked with 1) feeds the generated torque to the slipping clutch model (marked with 2), which also receives the torque generated by the brake model (marked with 3). The coupling model contains the solvers for the angular velocity on both sides of the coupling, which, alongside the translated torque on each side of the coupling is supplied back to the motor and brake models. The axis movement allowed by the brake model is translated through the gear and polygon effect models (marked with 4 and 5, respectively). The polygon effect can be turned off to simulate a rope hoist. The movement of the load is fed back to the two-mass spring-damper

DOI: 10.2195/lj_Proc_kivelae_en_201811_02
URN: urn:nbn:de:0009-14-47470

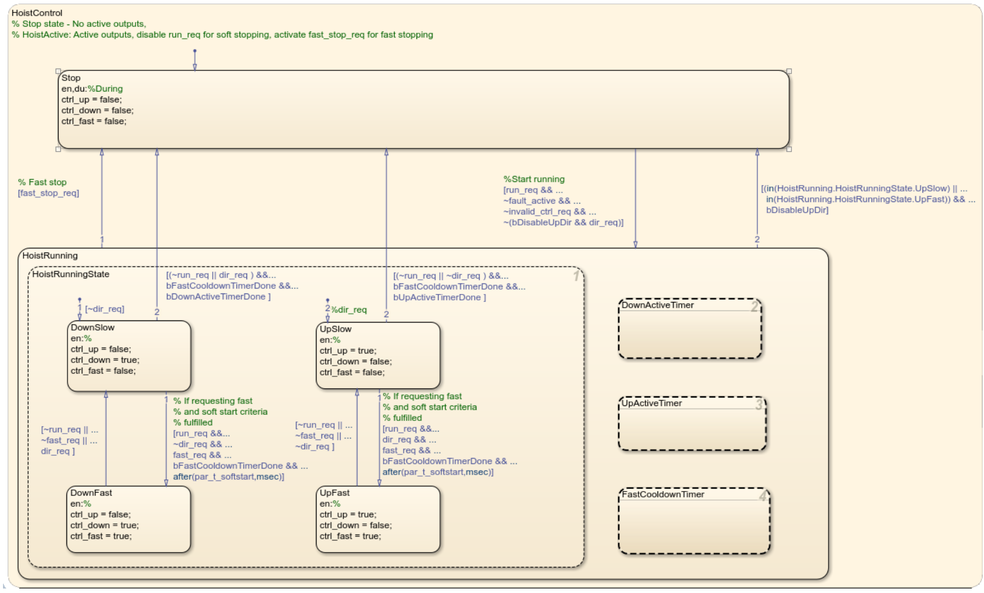

Figure 5. *Example stateflow model of the software for pole-changing motor control*

model (marked with 6), which supplies the load torque back to the motor-clutch-brake-section.

The implemented model allows for easily connecting the control system to drive the motor model, as well as to simulate different sensors, like for example a strain gauge signal (based on the force in the rope, if a rope hoist is simulated), or the speed signals from encoders connected to different axes. The hoist position can be used to generate e.g. limit switch signals.

3.4 Model validation

In a next step, the model needs to be validated against real hoist applications. Therefore, it is planned to do certain test runs with known parameters, so that the simulation results can be validated. Values that can be measured easily are the force in the chain, the rotational speed and the position of the load.

4 Software testing with the process model

In this section we describe different possibilities to use the model and the Matlab Simulink environment for testing the software of the hoist supervision unit.

4.1 Software modelling with finite state machines

With the addition of the Stateflow environment in Simulink, software systems can be modelled as hierarchical, finite state machines. Simulating discrete systems or logic circuits was possible also in the normal Simulink environment, but state machine modelling enables the creation of graphical formal models of software systems. [Har87], [Tiw02]

Using Stateflow, we can create an executable model of the safety-related software, which can later act as a specification of the handmade software implementation. An example of a hierarchical state machine model of a software module implemented in Stateflow is shown in Figure 5, where the control logic related to control of a pole-changing motor is shown. The system has two top level states, either the hoist is running or stopped. Within the hoist running state there are parallel states, where different timer functions are executed in parallel to the logic for activating different motor states depending on the direction and the requested and allowed speed.

State machine models can be used to create an executable model of the software specification quickly. The model can be used to check the software specification by simulation of the software logic with the developed electromechanical model, before moving forward with the manual implementation of the software running on the actual safety-related control system. The models can additionally be used for automated test generation, which can reduce the manual workload of creating module tests to achieve the statement coverage requirements. We are currently studying the suitability of this approach in project FOLSA.

4.2 Software-in-the-loop testing

In addition to the use of Stateflow models, we are studying SiL testing with the help of the process model. Simulink provides so called S-Functions, which enable the

DOI: 10.2195/lj_Proc_kivelae_en_201811_02
URN: urn:nbn:de:0009-14-47470

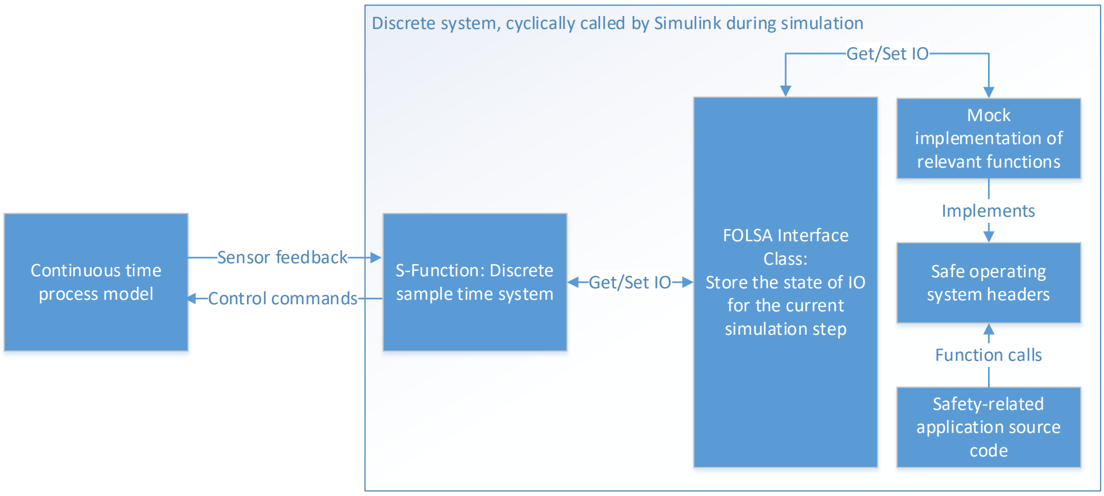

Figure 6. Interface to enable SiL testing

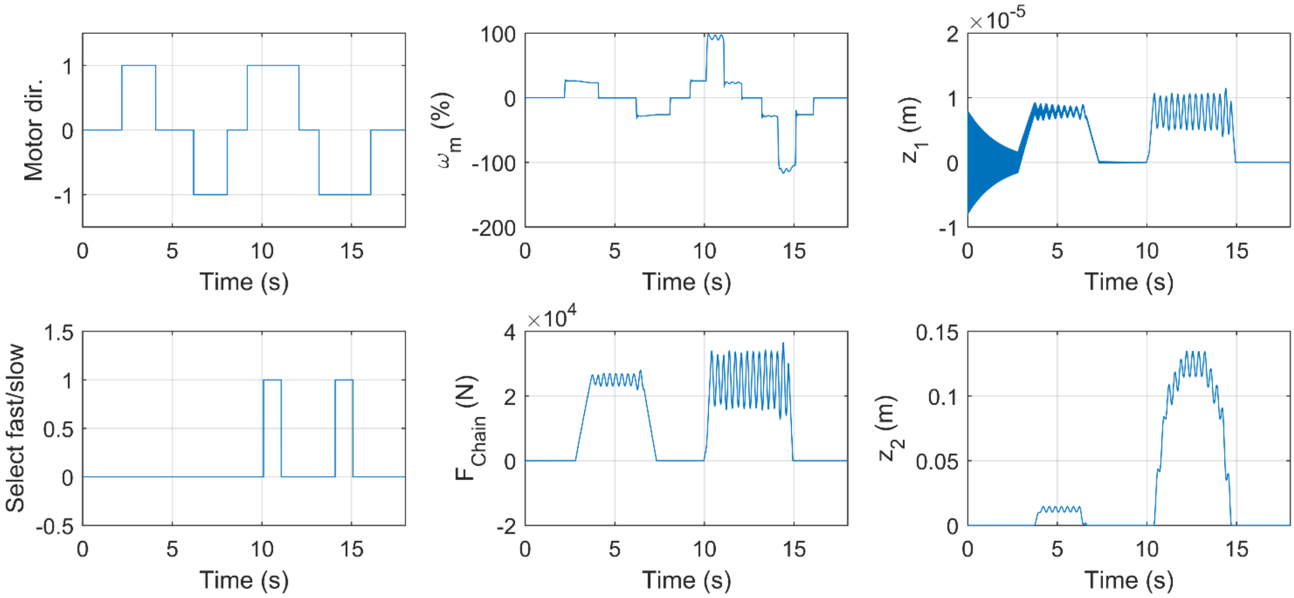

Figure 7. Test simulation with SiL: Control signals for motor direction and selecting fast or slow pole pair driven by the software on the left, the motor velocity and force in the chain in the middle, and the position of hoist and the load on the right

user to incorporate C/C++ into Simulink models. We have created an interface, which allows the safety-related application, written for the FOLSA platform, and thus for the HICore 1 safety system-on-chip (SoC) [HMS+14], to be executed within Simulink. The interface is illustrated in Figure 6. The handwritten source code for the safety system is compiled against the original safe operating system headers, for which mock implementations are provided, which communicate with a FOLSA interface class. The FOLSA interface class stores the controller IO values for the current simulation step, and is additionally also called through the S-Function to write the given control commands to the Simulink process model, as well as read back the sensor feedback from the model. The S-function, the FOLSA interface and the original safety application source code are compiled into a discrete Simulink system, which is called cyclically by Simulink.

A test simulation with SiL is shown in Figure 7. The software for the safety system is providing the control signals on the left side, for choosing the motor direction and the fast or the slow pole pair. The motor velocity and the force on the chain is shown in the middle. On the right the position of the hoist and the load are shown.

5 Conclusions

In this work we described a simulation model for a novel hoist concept, including the full electromechanical drivetrain combined with a two-mass spring-damper model

DOI: 10.2195/lj_Proc_kivelae_en_201811_02
URN: urn:nbn:de:0009-14-47470

of the crane bridge, the hoist and the load. We additionally described the requirements for functional safety in hoist applications, the related software verification requirements and the possibilities, how the developed simulation model can be used to help with the verification efforts.

In the future, the model will be validated against measurements from test runs with known parameters. We will additionally continue studying the different possibilities with model-based test generation, as well as using the simulation model for both SiL and HiL testing. For the latter, the simulation model needs to be extended with models of the relevant sensors.

6 Acknowledgements

We gratefully acknowledge that this research is funded by the German Federal Ministry for Economic Affairs and Energy (BMWi: Bundesministerium für Wirtschaft und Energie) under ZIM (Zentrales Innovationsprogramm Mittelstand) cooperation grant number ZF4251405DB7.

Literature

[EC06] European Parliament and The Council of the European Union: Directive 2006/42/EC Of The European Parliament And Of The Council of 17 May 2006 on Machinery, and Amending Directive 95/16/EC (Recast).

[EN13135] EN: Cranes - Safety - Design - Requirements for equipment.

[EN14492-2] EN: Cranes - Power driven winches and hoists - Part 2: Power driven hoists.

[EN15011] EN: Cranes - Bridge and gantry cranes.

[Gol04] Golder, M.: Ein Beitrag zur Kostenabschätzung für Brückenkranträger in Kastenbauweise auf Basis ihrer Bemessungsgrundlagen und Dimensionierungsnachweise. Dissertation, Universität Karlsruhe, Institut für Fördertechnik und Logistiksysteme, 2004.

[Har87] Harel, D.: Statecharts. A visual formalism for complex systems. In Science of Computer Programming, 1987, 8; pp. 231–274.

[HB14] Haberhauer, H.; Bodenstein, F.: Maschinenelemente. Gestaltung, Berechnung, Anwendung. Imprint: Springer Vieweg, Berlin, Heidelberg, 2014.

[HKV+16] Holzweissig, P. et al.: HiPi – A concept for a compact, IoT-enabled safety controller: Logistics Journal: Proceedings, 2016.

[HMS+14] Hayek, A. et al.: HICore1. “Safety on a chip” turnkey solution for industrial control: 2014 IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors. IEEE, 2014; pp. 74–75.

[IEC61508-1] International Electrotechnical Commission: EN IEC 61508-1:2010 - Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 1: General requirements, 2010.

[IEC61508-3] International Electrotechnical Commission: EN IEC 61508-3:2010 - Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 3: Software requirements, 2010.

[IEC62061] International Electrotechnical Commission: EN IEC 62061:2005 - Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems, 2005.

[ISO13849-1] International Organization for Standardization: EN ISO 13849-1:2015 - Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design.

[KG16] Kivelä, T.; Golder, M.: Simulation and analysis of an emergency lowering system for crane applications. Wissenschaftliche Gesellschaft für Technische Logistik, 2016.

[Lan04] Landschützer, C.: Analyse von Schwingungen an einsträngigen Elektrokettenzügen, 2004.

[ML00] Munoz, A. R.; Lipo, T. A.: Dual stator winding induction machine drive. In IEEE Transactions on Industry Applications, 2000, 36; pp. 1369–1379.

[Sle89] Slemon, G. R.: Modelling of induction machines for electric drives. In IEEE Transactions on Industry Applications, 1989, 25; pp. 1126–1131.

[Tiw02] Tiwari, A.: Formal semantics and analysis methods for Simulink Stateflow models, 2002.

M.Sc. Tommi Kivelä, is working as a Research associate at the Institute for Material Handling and Logistics (IFL), Karlsruhe Institute of Technology (KIT).

Email: Tommi.Kivelae@kit.edu

DOI: 10.2195/lj_Proc_kivelae_en_201811_02
URN: urn:nbn:de:0009-14-47470

M.Sc. Steffen Bolender, is working as a Research associate at the Institute for Material Handling and Logistics (IFL), Karlsruhe Institute of Technology (KIT).
Steffen Bolender studied Mechanical Engineering at Karlsruhe Institute of Technology (KIT) from 2009 to 2015. Since November 2015 he is a research assistant in the department of Warehouse and Material Handling Technology at IFL.
E-Mail: Steffen.Bolender@kit.edu

Prof. Dr.-Ing. Markus Golder, was the head of the chair of Safe mechatronic systems in intralogistics (SIMESI), Institute for Material Handling and Logistics (IFL), Karlsruhe Institute of Technology (KIT). He is now the head of the Professorship of Material Handling and Conveying Engineering, Technische Universität Chemnitz.

Prof. Dr.-Ing. Kai Furmans, is the head of the Institute for Material Handling and Logistics (IFL), Karlsruhe Institute of Technology (KIT).

Address: Institut für Fördertechnik und Logistiksysteme (IFL), Gebäude 50.38, Gotthard-Franz-Str. 8, 76131 Karlsruhe, Deutschland. Tel.: +49 721-608-48621, Fax: +49 721-608-48629