

The Embedded Simulation via FMI and its Application to the Simulation of Lifetime Tests Including Wear

Julia Gundermann¹ Matthias Thiele² Sebastian Fraulob³ Susanne Walther¹ Karsten
Todtermuschke¹ Uwe Schnabel¹

¹ESI ITI GmbH, julia.gundermann@esi-group.com

²Institute of Electromechanical and Electronic Design, Technische Universität Dresden

³Johnson Electric Germany GmbH & Co. KG Dresden

Abstract

The "Embedded Simulation via FMI" is a new modeling approach which allows for efficient and fast computation of systems with a clear separation of time axis or time scale. For its application the so-called "inner model" is wrapped into an FMU and embedded into an outer model, whose dynamics control the integration. The computation of the embedded model is only utilized on demand. In this way the "Embedded Simulation via FMI" uses the Functional Mock-up Interface for Co-Simulation in a different way than it was provided for. The functionality has been realized within SimulationX prototypically. It is applied to the simulation of the lifetime test of a linear stepper motor including wear in the screw drive, for which the axial play after several months' runtime shall be determined. A significant reduction of computing time while preserving considerable accuracy can be shown.

Keywords: FMI, wear, SimulationX, simulation, mechatronics

1 Introduction

The functional mock-up interface (FMI) defines a tool independent standard interface used for the coupling of different simulation tools (FMI, 2014). It allows for the simulation of multi-component systems. Components of the system are packed into an functional mock-up unit (FMU), which is a .zip-file. It contains an xml-file with a model description and C-functions covering the functionality and behavior of the model by functions defined by the FMI-standard. The acceptance and application of the FMI-standard is increasing, there are about 90 tools which support FMI. Depending on the simulation task a model can be exported with (FMI for Co-Simulation) or without its solver (FMI for Model exchange).

The FMI-standard for Co-simulation is used for the coupling of component models, which run independently besides the communication after predefined time-intervals. There are examples for which the simulation of all components is not necessary throughout the whole simulation time. This is the case, if the system's topology

can be divided into an inner and an outer model, which have a separated time axis or time scale. In such cases the pure FMI for Co-Simulation enforces unnecessary computations of the inner model. The "Embedded Simulation via FMI" avoids this by calling the computation of the inner model only on demand.

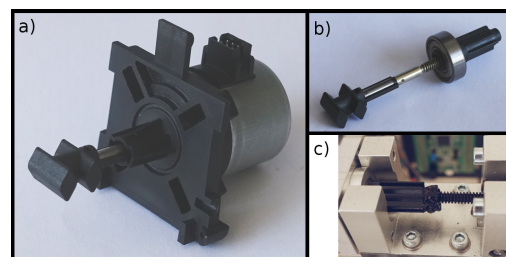


Figure 1. A linear stepper motor based on a screw drive. a) Complete assembly of the motor to be used as adjusting drive. b) Dismantled nut thread with ball bearing. These two components are susceptible to mechanical wear. c) Screw drive without motor in a setup for lifetime testing. The manufacturer of this and other mechatronic components is interested in the simulation of these lifetime tests in order to save time during the development of new components.

One application of this concept is the simulation of lifetime tests of mechatronic (and other) components. Figure 1 shows an linear stepper motor, which underlies wear effects due to abrasion in the screw drive. The manufacturer is faced with the following problem: the development period of a new component is comparable to the duration of lifetime tests. The latter are necessary since they assess the compliance of predefined criteria during the lifetime of the component. One way out of this problem is the virtual simulation of these lifetime tests. Such a simulation consists of a high number of repetitions of nearly identical cycles, i.e. rotating a screw drive back and forth. During the lifetime the component is subject to wear and aging effects. These alter the behavior of the component slowly, i.e. on a long time scale, however barely within the duration of one cycle. However, the current behavior of a component and the wear phenomena cannot be considered separately, since wear is dependent on

the usage of the component, but vice versa wear and aging modify the response of a system, e.g. debris increases friction. The manufacturer is interested in a fast extrapolation of the wear effects, yet not in the computation of each cycle. The virtual simulation of a whole lifetime test including wear and aging can help to derive information about the accumulated effects of loads, temperature, etc., which in turn can be used for the re-planning, shortening or even circumvention of real lifetime tests.

The simulation of the wear effect in a lifetime test consisting of many nearly identical cycles is slowed down by the calculation of the fast degrees of freedom such as the position of the screw drive. Wrapping these fast degrees of freedom into an inner model and embedding this inner model into an outer extrapolation enables to substantially speed up the simulation. In the "Embedded Simulation via FMI" the inner model containing the simulation of one cycle is wrapped into a functional mock-up unit, and then embedded into an outer integration, which extrapolates the wear quantities such as the abraded volume.

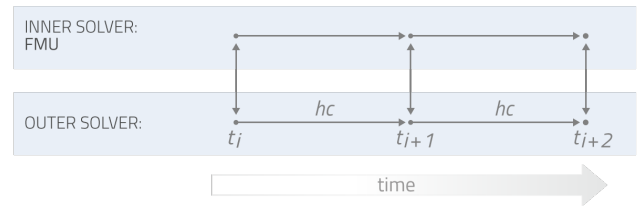
The remainder of this paper is structured as follows: In section 2 the concept of the Embedded Simulation via FMI is illustrated. Section 3 introduces the linear stepper motor. The motor is modeled with components from SimulationX libraries. The screwDrive - component is extended by a wear model, which is also presented here. The preparation of the model for the Embedded Simulation is described in section 4, and its performance is assessed in section 5. The article closes with a summary.

2 Embedded Simulation via FMI

The Embedded Simulation via FMI allows for an accelerated computation of certain types of coupled simulations, where one model's time scale or time axis is separated from the time scale or axis of the other models. The following small examples shall illustrate the idea and what is meant by time scale or time axis separation. One of the examples - the linear stepper - will be illustrated explicitly in the following section.

- Equation-free modeling / molecular dynamics (Kevrekidis and Samaey, 2009) - time scale separation: Given is a system with many fast degrees of freedom (such as molecules), for which the time-like evolution is defined by equations. However, one is interested in the long-term dynamics of averaged quantities such as the evolution of the energy or temperature of the system.
- Simulation of lifetime tests - time scale separation: The example of lifetime tests including wear and aging was introduced above. These tests - and hence its simulation - contain a high number of repetitions of nearly identical cycles. However, one is only interested in the slowly varying quantities such as the amount of wear debris and the increase of axial backlash, but not in the fast degrees of freedom.

FMI for Co-Simulation



Embedded Simulation via FMI

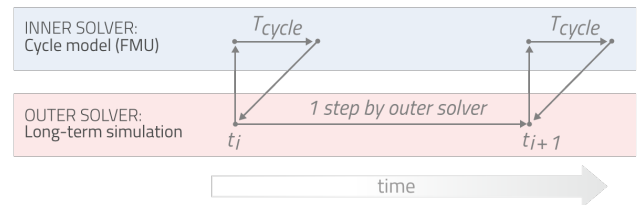


Figure 2. Reuse of the Functional Mock-up Interface (FMI). The upper figure illustrates the usage of FMI in the way it was designed for. Two models are connected and are simulated in parallel independently of each other. They exchange variables at previously defined communication times. The Embedded Simulation via FMI is shown in the lower figure. In contrast to the Co-Simulation only the outer solver runs for the whole simulation. It calls the inner solver at each of its time steps but not at predefined communication steps. The inner solver simulates for the duration of one cycle only. It returns output variables to the outer solver afterwards. Since the time steps of the outer solver are expected to be much longer than T_{Cycle} , the outer solver continues its calculation at t_i , but not at $t_i + T_{Cycle}$.

- Model predictive control (Dittmar and Pfeiffer, 2004) - time axis separation: The model predictive control uses a time-discrete dynamical model of the process, which is to be controlled. This model shall compute the future behavior of the process depending of the input signals based on an iterative, finite-horizon optimization of the underlying model.

Common to all three examples is the possibility to split the system into an inner model - the fast molecules, one cycle in the lifetime test, or the model predictive control - and an outer model - the evolution of energy or temperature, the extrapolation of accumulated wear debris, or the outer system which the controller is part of. The crucial point is that the inner system needs only to be computed on demand, but not for the whole simulation time of the outer system. For the Embedded Simulation the inner model is wrapped with its own solver into a FMU and embedded into an outer model. The latter determines the time-integration of the overall system. In contrast to the common usage of the FMI for Co-Simulation the inner model does not run independent from the outer model and communicates only at previously fixed points in time. Instead it is run for a predefined (short) time interval, and only on demand given by the outer model. The number of calls of the inner model depends on the simulation task. For systems with time scale separation this number will be

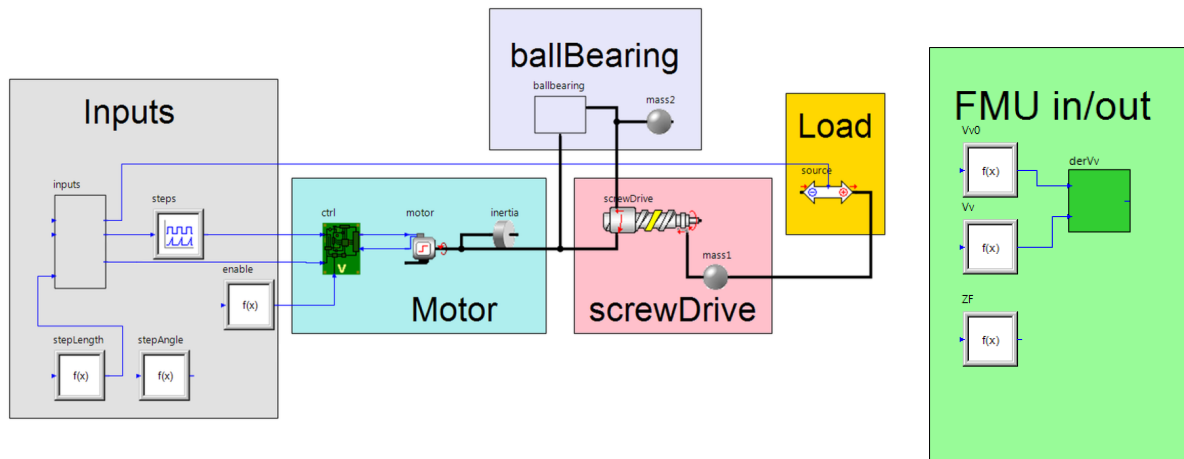


Figure 3. SimulationX model of the linear stepper motor. The components affected by wear are the screwDrive due to abraded volume and the ballBearing due to an increasing backlash. The latter effect is negligibly small and hence not considered further. The FMUin/out block on the right is added in preparation of the FMU-Export.

considerably low compared to the ratio between the simulation times of the outer and inner model. In contrast, for systems with time axis separation no general statement can be made about the number of calls of the inner model besides that it depends on the variables exchanged between the embedded inner and the outer model.

Every time it is initiated, it receives input values from the outer model (such as previous wear debris), while some other variables have default start values (such as spindle position). The inner model is reset, initialized, run, and returns output values to the outer model, and is reset (Technically it is reset at the beginning of the FMU-call). Figure 2 illustrates the difference to the standard of FMI for Co-Simulation.

With the Embedded Simulation via FMI it is possible to speed up simulations preserving considerable accuracy. Furthermore the inner model is replaceable more easily. The presented method was realized and tested in SimulationX for an example with time scale separation. The following section shall illustrate the application of the Embedded Simulation via FMI to the lifetime test of a linear stepper motor.

3 The Linear Stepper

3.1 The SimulationX model

Figure 1 shows a picture of the linear stepper model. In preparation of the lifetime-simulation it was modeled in SimulationX, as shown in Figure 3. It is parametrized such that during one simulation the motor drives the nut to rotate by a predefined angle forth and back. This rotation leads to a translation of the spindle a few millimeters forward and backward. The translation is acting against a constant force (Load) (in the example, $F = 1$ N). By construction the angle of rotation increases stepwise, but not continuously. Each jump triggers an event, which slows down the simulation. The step frequency of 50 Hz on av-

erage for 1000 steps in one cycle leads to a computation time of 1.6 s, versus 20 s of real cycle length.

3.2 The wear in the screw drive

The centerpiece of wear processes is the screw drive, which is modeled by the screwDrive element from the PowerTransmission library in SimulationX. The linear stepper model does not contain other wear or aging phenomena such as the increase of backlash in the ball bearing, since the increasing amount of wear debris in the screw drive is considered as the major effect leading to system failure.

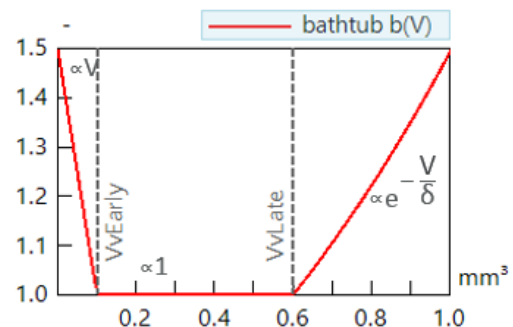


Figure 4. Bathtub curve determining the value of the wear coefficient k (Eq. (2)) and the friction coefficient $\mu = \mu_0 b(V)$. The parameter b linearly decreases for $V < V_{vEarly}$, stays constant until $V = V_{vLate}$ and increases exponentially afterwards. The values for the regime-changes and the exponential increase are optimally derived from experiments, but in general they have to be estimated.

The underlying wear model bases on Archard's law,

$$dV = \frac{k F s}{h}, \quad (1)$$

where V is the volume of wear debris, dV its increase, F the total normal load, s the sliding distance. The material

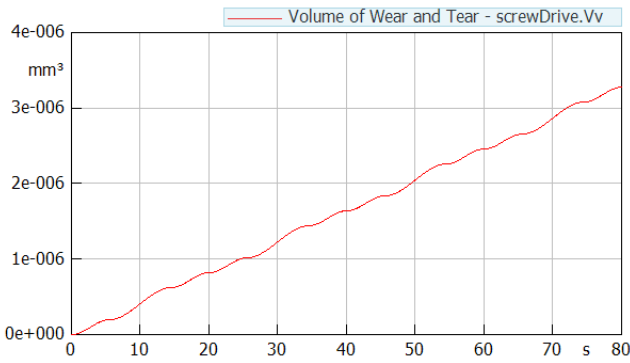


Figure 5. Simulation results of the linear stepper motor, original model. Abraded volume in the screw drive vs. simulation time. The overlaying oscillation is easy to see. Computation step size was $\ll 1s$.

quantities h and k are the Vicker’s hardness of the plastics and the wear coefficient, respectively. In the model here the wear coefficient is assumed to be dependent on the wear debris V , and to run through a bathtub curve (Stachowiak, 2006),

$$k = k_0 b(V). \tag{2}$$

The bathtub is shown in Figure 4.

The axial play in the screw drive increases due to the abraded layer on the flanks. It can be calculated by geometric construction. The wear debris also feeds back on the current wear process, since the friction coefficient is not constant, instead, $\mu = \mu_0 b(V)$.

Technically the wear is modeled by an extension of the ScrewDrive component.

4 Preparation for the Embedded Simulation

Per cycle the linear stepper model computes a change in wear volume. Figure 5 shows the wear debris calculated during four cycles. The increasing curve is superimposed by an oscillation with period equal to the cycle length. For a simulation of approximately 60 days, i.e. rotating forth and back 259.200 times, one is not interested in this oscillation, but more in the long term dynamics. Therefore it suffices to calculate the average change in wear debris per cycle, i.e. $derVv = (screwDrive.Vv0 - screwDrive.Vv) / TCycle$, where $Vv0$ and Vv are the values of wear debris at the beginning and end of the cycle.

For the preparation of the Embedded Simulation the averaged derivative $derVv$ is calculated in an additional element, cf. Figure 3. Furthermore, a function element ZF is added, which contains the regime changes of the bathtub, i.e., $ZF.y = \{screwDrive.Vv - screwDrive.VvEarly, screwDrive.Vv - screwDrive.VvLate\}$. Triggering events in the Embedded Simulation by adding if-conditions containing $ZF.y$ forces the solver to

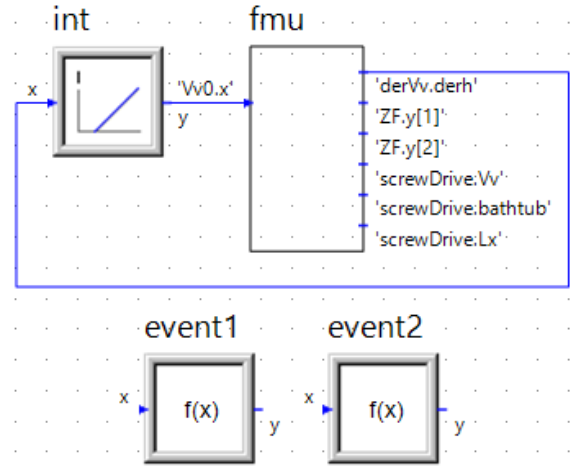


Figure 6. Embedded Simulation via FMI of the linear stepper motor. The wear debris Vv is integrated by the outer solver. The function blocks `event1` and `2` are used to let the outer solver catch the transition points in the bathtub curve correctly (cf. section 4).

integrate more accurately at the regime changes.

The linear stepper model is exported as FMU for Co-Simulation 2.0. The wear debris $Vv0$ at the beginning of the cycle is set as input, the derivative $derVv$ and the bathtub-differences ZF as output. Further output variables or parameters are optional.

The FMU is imported into a new SimulationX model and the modelica-component containing it is modified to make the FMU run in the embedded mode (cf. Figure 2). The main modifications are the replacement of the communication step size hc by the cycle length $TCycle$, and the altered calling of the `fmu` in order to be re-set, re-initialized and run for one cycle everytime it is called. Technically the latter was realized in SimulationX by wrapping all these steps into a single function call.

The output of the variable of interest, i.e. `derVv.derh` is connected to an integral element, whose output, in turn, is connected to the FMU’s input. For each component of ZF a function block is added in order to trigger an event, e.g. `event1.y = if (fmu.ZF.y[1]>0) then 1 else 0`, cf. Figure 6.

5 Performance and Validation

To validate the approach, the Embedded Simulation via FMI of the linear stepper is compared with a longterm-simulation of the original model. The simulation time is set to 60 days.

Using the wear coefficient from experiment, $k0 = 2 \cdot 10^{-10}$, there is little wear and tear within 60 days. The error tolerances of the outer solver have been chosen such that a reduction would not improve the accuracy significantly, but would slow down the computation speed. The deviation of calculated wear debris between original model and Embedded simulation is smaller than 0.05%. The computation time in SimulationX was reduced from

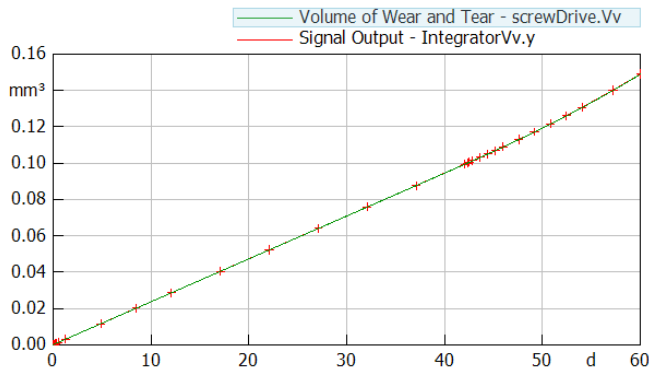


Figure 7. Comparison of Embedded Simulation with FMI (pluses, red) and original model (solid line, green) with accelerated effect of wear and tear. The wear behavior switches between regions after some hours, where $V_v = 10^{-3} \text{ mm}^3$, and after 42 d, where $V_v = 0.1 \text{ mm}^3$, respectively. These behavior changes are passed to the outer solver via triggering events. In the second region, where the wear coefficient stays constant, the time step size of the outer model reaches its maximal value of 5d.

about 2.5 days to 1 minute or by the factor 4000, respectively.

To reveal the dynamics within the region of exponential increase longer simulation times (with the original model) are needed. On the other hand it is also possible to accelerate the wear by increasing the wear coefficient k_0 to $6 \cdot 10^{-8}$. Then the simulation time of 60 days is sufficient to reach all regions of the bathtub curve. The maximum difference between the result values of both simulations is smaller than 0.1%, cf. Figure 7, which also reveals that the solver is able to catch the regime changes in the bathtub. In this case the computation time was reduced from about 2.3 days to 2.5 minutes or by the factor 1300, respectively.

6 Summary and Outlook

The Embedded Simulation via FMI can speed up certain types of simulation tasks by preserving considerable accuracy. The new simulation approach was introduced and motivated by three potential applications. Its successful implementation was proven exemplary for the lifetime simulation of a mechatronic component including wear. The application of the Embedded Simulation to the other two of the named examples or further systems is future work. It remains to be examined whether other integration approaches such as Quantized State Systems methods (Cellier and Kofman, 2006; Casella, 2015) could serve as an alternative.

Acknowledgements

The project Robustness and Reliability Simulation of Mechatronic Systems including Aging and Wear (ROMESA (BMBF, 2015)) runs in cooperation of ESI ITI GmbH Dresden, Institute of Electromechanical and Electronic Design from Technische Universität Dresden,

Dynardo GmbH Weimar and Johnson Electric Germany GmbH & Co. KG Dresden (associated partner). Especially the authors but also the involved companies and institutes like to thank the German Federal Ministry of Education and Research (BMBF) represented by the project coordinator German Aerospace Center (DLR) for supporting financially.

References

- Federal Ministry of Education and Research BMBF. *Projektblatt Romesa*, 2015. URL http://www.pt-sw.de/media/content/Projektblatt_ROMESA.pdf.
- Francesco Casella. Simulation of large-scale models in modelica: State of the art and future perspectives. In *Proceedings of the 11th International Modelica Conference*, pages 459–468. The Modelica Association, September 21–23 2015. doi:10.3384/ecp15118459.
- François E Cellier and Ernesto Kofman. *Continuous system simulation*. Springer Science & Business Media, 2006. ISBN 0387261028.
- Rainer Dittmar and Bernd-Markus Pfeiffer. *Modellbasierte prädiktive Regelung: Eine Einführung für Ingenieure*. Walter de Gruyter, 2004. ISBN 3486275232.
- Functional Mock-up Interface FMI. *FMI 2.0*, 2014. URL <https://www.fmi-standard.org/>.
- Ioannis G. Kevrekidis and Giovanni Samaey. Equation-free multiscale computation: Algorithms and applications. *Annual review of physical chemistry*, 60:321–344, 2009.
- G. W. Stachowiak. *Wear: materials, mechanisms and practice*. John Wiley & Sons, 2006. ISBN 0-262-16209-1.