

# Implementation of a Modelica Library for Simulation of Electromechanical Actuators for Aircraft and Helicopters

Franciscus L.J. van der Linden  
German Aerospace Center (DLR),  
Institute of System Dynamics and  
Control, Münchner Straße 20  
82234 Weßling, Germany  
Franciscus.vanderlinden@dlr.de

Clemens Schlegel  
Schlegel Simulation GmbH  
Meichelbeckstr. 8b  
85356 Freising, Germany  
cs@schlegel-simulation.de

Markus Christmann  
EADS Innovation Works, TCC6  
81663 Munich, Germany  
markus.christmann@eads.net

Gergely Regula  
MTA SZTAKI  
13-17 Kende St.  
1111 Budapest, Hungary  
gergely.regula@sztaki.mta.hu

Chris I. Hill  
University of Nottingham  
Aerospace Technology Centre,  
Innovation Park  
Nottingham. NG7 2TU. UK  
c.hill@nottingham.ac.uk

Paolo Giangrande  
University of Nottingham  
Tower Building,  
University Park  
Nottingham. NG7 2RD. UK  
p.giangrande@nottingham.ac.uk

Jean-Charles Maré  
Institut National des Sciences Appliquées  
135 Avenue de Rangueil  
31077 Toulouse Cedex 4, France  
jean-charles.mare@insa-toulouse.fr

Imanol Egaña  
IK4 TEKNIKER  
Calle Iñaki Goenaga, 5  
20600 Eibar, Gipuzkoa, Spain  
imanol.egana@tekniker.es

## Abstract

The goal of the A2015 library presented in this paper is to develop a Modelica based, tool-independent standard for electromechanical actuators (EMA). This will contribute to the establishment of a “common language” throughout the development of EMAs for aircraft and helicopters and through the supply chain. All stages of the design and validation process (conceptual design, specification, development and validation) are covered. The modeling approach addresses specific aspects of the EMA design process not covered by existing tools. The library scope, engineering need and implementation are described. Modeling of selected EMA components is discussed in more detail. An application example of the library is given (linear actuator, A320 aileron)

*Keywords: Actuator, EMA; Library; Multi Physics*

## 1 Library scope

### 1.1 Introduction

Protecting the environment and providing efficient onboard energy supply is one of the top goals in the

development of future aircraft. A key technology towards the realization of these goals is the “More Electric Aircraft”. Various research initiatives have been launched in recent years to get closer to this goal. The ACTUATION2015 project [1] will complete this approach by focusing on Electro Mechanical Actuator (EMA) technologies. EMAs are mandatory in order to substitute hydraulic circuits, pumps and reservoirs. The objective of the ACTUATION 2015 project is to develop and validate a common set of standardized, modular and scalable EMA modules that address cost, reliability and weight requirements from the air framers. In the context of that project the A2015 simulation model library is developed to support and streamline the EMA design process.

### 1.2 Engineering needs

The focus of the ACTUATION2015 project is on aircraft families including business, regional and large commercial aircraft as well as helicopters. The actuation systems covered include primary and secondary flight controls, main landing gears and breaks. A further major focus is on modularization and scalability of EMA modules. All the needs and architecture differences of such systems and all dif-

ferent stages of the development process (conceptual design, specification, development, and validation) through the whole supply chain must be covered by a universal simulation model library.

Apart from covering the relevant physical effects and general system dynamics aspects, the library should support the development in terms of concept and performance assessments, sizing (as far as system aspects are involved), component requirements definition and fail case assumptions, system reaction and performance in case of failures (static and transient), and virtual design validation. Electromechanical actuators have mostly a reduced reliability compared with conventional hydraulic actuators. Therefore the library must allow assessing redundant actuator concepts.

Since simulations on system and subsystem level can't cover all computational needs of an EMA development, interoperability with established tools is vital. Because many tools support it, the FMI standard [13] is the natural choice for model exchange of Modelica libraries. The library does not provide distributed parameter models (for e.g. stress computations) and monitoring and state of health algorithms, because specialized tools for these tasks are already available.

### 1.3 Library architecture

In order to cover the described range of applications several models of different scope and level of detail are implemented for each of the core EMA components (multi-level approach, see e.g. [14]). Since each group of models shares the same interface they are easily replaceable. In the A2015 library five modeling levels are predefined, mostly associated with nonlinearities included and events triggered (for easier handling each level has an associated icon color):

1. Perfect (linear, no losses)
2. Linear, invertible
3. Nonlinear, invertible (e.g. using tanh instead of sign functions)
4. Hard nonlinear (state events are triggered by nonlinearities)
5. Fully switched (e.g. switching inverter, based on state events)

This scheme does not imply that for all components models of all levels are included, or for each level only a single model is included. Instead, the modeling level scheme is introduced to give the user a quick idea what kind of effects a certain model does include.

Naturally, the A2015 library follows a standardized model breakdown structure with common interfaces based on the Modelica Standard Library (MSL) and uses its predefined implementations of thermal flows and heat exchange, and mechanical support flanges [9]. Parameters are organized in records, signals are bundled in busses. In order to unify failure injection, a dedicated library is used [12].

## 2 Selected library components

The A2015 library contains model components from the following domains: Electrical (inverters, motors), mechanical (rotation to rotation and rotation to translation transformers), sensors (position, speed, force, etc.), thermal (heat sinks, housings), and control (e.g. force fight compensator). Selected components are described in this paragraph.

### 2.1 Fault Triggering Library

The development of health monitoring algorithms, but also the design of fault tolerant redundant actuators is aided by the possibility to easily trigger faults in the models.

The `FaultTriggering` library [12] is used as a standardized approach to trigger all failures implemented in the library. At the moment the motor, inverter, gear and nut screw models have predefined faults that can be enabled. Further model components will follow.

### 2.2 Inverters

This section gives an overview of the implementation of the multi-level Power Electronic Inverter modeling provided within the A2015 library. For further technical details see [3].

#### 2.2.1 Package Structure

As described previously, a multi-level modeling approach is used within the A2015 library. For the Power Electronic Inverters 5 levels of complexity are provided. The Inverters package is split into the `NonSwitching` and the `Switching` subpackages, see figure 1. The `NonSwitching` package provides models for levels 1 to 4 and the `Switching` package provides a level 5 model. A core feature of the A2015 library is that for each individual component every modeling level is fully replaceable with one another. In order for each modeling level within the Inverters library to be fully exchangeable a common interface is used for all 5 modeling levels.

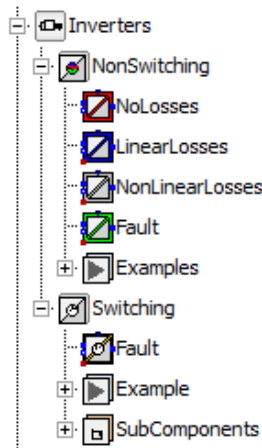


Figure 1: Structure of the Inverters package as part of the Actuator Library.

### 2.2.2 Losses

It is important to note that all the Power Electronic Inverter models are fully multi-directional. Therefore losses are fully implemented, regardless the direction of the power flow. Hence, if power flows from DC to AC the AC side power will be lower than the DC side power and vice versa. This is inherent within the model and will be calculated automatically even when the direction of power flow changes during simulation. Table 1 summarizes the losses included in each level of the Inverter models.

Level	Losses
1	None
2	Linear Losses
3	Non-Linear Losses
4	Non-Linear Losses
5	Conduction Losses

Table 1: Losses modeled within each level of the Inverters package.

Using the level 2 model the user may specify a constant efficiency for the Power Electronic Inverter under all operating conditions (a default value is provided, if no data is available). Levels 3 and 4 both use non-linear loss characteristics. The user can specify the power range of the inverter and the losses within the inverter over the specified power range. A default characteristic is also given, as shown in figure 2 below, however accurate characteristics are recommended to be added by the user if known. The default characteristics are per unit and are scaled within the model by the specified, or default, power range.

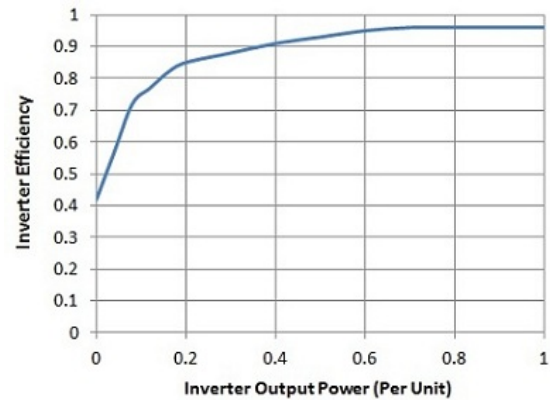


Figure 2: Default non-linear loss characteristics

Level 5 includes switches and Pulse Width Modulation in order to create the required electrical outputs. The losses included are the conduction losses due to the switch and diode resistances when conducting. Energy losses due to switching are neglected.

### 2.2.3 Faults

Another core aspect of the A2015 library is the inclusion of fault conditions within all component models. In the case of the Power Electronic Inverters these are mainly included in the level 5 model because failure of switches are the most likely failure conditions and no switches exist within the lower level models. Table 2 summarizes the fault conditions included in the Inverter models.

Level	Faults
1	None
2	None
3	None
4	Full-Bridge
5	Single Switch, Single Phase, Multi-Phase and Full Bridge

Table 2: Fault conditions modeled within each level of the Inverters package.

For level 4 a full bridge fault is implemented. When triggered the Power Electronic Inverter output becomes zero. This represents a full-bridge open circuit fault or deactivation of the Inverter. The level 5 model includes the ability to activate open circuit and short circuit faults. These can be injected for a single switch, multiple switches, a single phase, multiple phases and full bridge. The Fault Triggering library described in [12] is used in order to trigger the implemented faults. The actual faults can be injected at any time instance but must be predefined.

## 2.3 Motors

### 2.3.1 Package Structure

According to the machine topologies considered, the Electrical Machines package is divided in four sub-packages:

- Basic, which considers a generic AC machine
- PermanentMagnet, which includes the permanent magnet synchronous machine (PMSM) models
- Reluctance, which models synchronous reluctance machines (SRM)
- DirectCurrent, which contains the models of DC machines with independent excitation

An Example package is provided for each of the described sub-packages, in order to highlight the model features. The Electrical Machines package has been developed adopting a multi-level approach as described in chapter 1. Several model levels of the same machine type, with different physical effects taken into account have been implemented.

### 2.3.2 Model Interfaces

The interchangeability among model levels is ensured using common interfaces. Considering AC electrical machines `PositivePlug` and `NegativePlug` have been used as electrical interface. In case of DC machine models, `PositivePin` and `NegativePin` have been adopted as electrical interfaces, for both armature and field windings. All the machine models adopt `Flange` and `Support` as mechanical interfaces and contain a heat port providing housing temperature and heat flow.

### 2.3.3 Basic Model

The basic model is the simplest model for a generic AC machine. The machine behavior is described by a torque constant, which establishes the relationship between current and electromagnetic torque. Losses, non-linear effects and thermal behavior are not considered at this modeling level. The basic model, as all the other machine models, can work reversibly representing both motor and generator.

### 2.3.4 Standard Models

Standard models are based on a lumped parameter approach. They describe the behavior of the machine considering losses and thermal behavior, but neglecting magnetic saturation, torque ripple and fault conditions. Standard models have been developed for PMSM, SRM and DC machine. These models are power balanced. The losses taken into account are:

copper, iron [4], mechanical and for DC machine brush losses. Finally, standard models provide a machine thermal model, which gives the winding and housing temperatures, and the heat flow.

### 2.3.5 Saturation Models

Saturation models for PMSM, SRM and DC machines have been obtained by extending the standard models. The main features of the Saturation models are magnetic saturation and torque ripple. Magnetic saturation is taken into account using two approaches (which can be user-selected): look-up table and analytical implementation. In the former case, the torque-current curve data (may be obtained by a finite element simulations) are stored in a look-up table, within the model. In the latter case, the torque-current curve is implemented using three given points (current and torque at the knee and torque at the saturation region). Finally, Saturation models allow superimposing a sinusoidal torque ripple on the computed electromagnetic torque. The torque ripple parameters (amplitude and frequency) must be given by the user.

### 2.3.6 Fault Models

Models including fault conditions have been implemented only for PMSM, since the presence of permanent magnets (PMs) is source of concerns in the case of failures. Fault conditions such as winding short-circuits, winding open-circuits and PM demagnetization have been considered. For each of these cases a specific PMSM model has been developed in order to minimize the model complexity. The fault models are:

- ShortCircuit, which implements single- and three-phase short-circuits
- OpenCircuit, which considers single- and three-phase open-circuits
- Demagnetization, which takes into account the PM demagnetization.

As suggested in [5], the winding faults have been implemented by means of ideal electrical switches. In the demagnetization model, the fault condition is taken as a consequence of the fault (i.e. PM flux reduction) neglecting the causes that led to the fault.

## 2.4 Mechanical rotational (reducers)

The rotational mechanical models include gear reducers and torque limiters. As with the other components, multiple modelling levels are included for different accuracy needs.

### 2.4.1 Package Structure

The package structure is given in figure 3. Only selected packages are expanded to give a good overview of the used models.

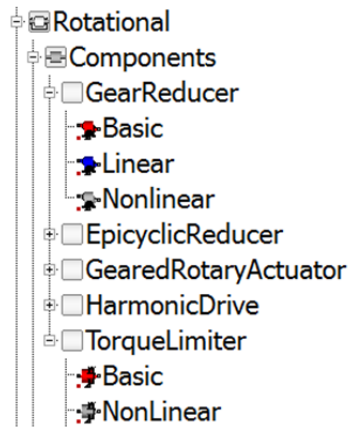


Figure 3: Structure of the Rotational package.

### 2.4.2 Model properties

All rotational models are fully balanced models with optional support flange. Using this approach, the support torques can be modelled and used for housing design. Also all models are energetically balanced. All losses are available in the (optional) heat ports.

### 2.4.3 Gear reducers

The `GearReducer`, `EpicyclicReducer`, `GearedRotaryActuator` and `HarmonicDrive` packages model geared transmissions with following levels of detail:

- Basic: no losses
- Linear: losses dependent on speed
- Nonlinear: losses dependent on the speed using a hyperbolic tangent approach to approach the nonlinear friction effect at zero velocity.

### 2.4.4 Torque limiter

The torque limiter restricts the torques on a shaft to a safe level by transferring the rest torque to a support flange.

## 2.5 Mechanical translational (nut screws)

As illustrated by table 3, the most simple models of nut screws (NS) start from the MSL `IdealR2T` model to reproduce the functional effect of perfect power transformation between rotational and translational domains.

For the most simple case, end-stops and bearings are not explicitly considered in the NS model:

- When the displacement is out of physical stroke, a warning is generated or the simulation is stopped.
- Anti-rotation and anti-translation devices are included implicitly, assuming no translation of the drive shaft and no rotation of the actuator rod.

When using the support options, the NS model becomes a mechanical quadriport model which combines the rotation and translation of nut and screw. This enables the explicit modelling of anti-rotation and anti-translation outside the NS model [6]. Inertial effects are also to be added outside of the NS model.

Technological imperfections are progressively introduced and modelled with increasing complexity. Among all candidate topologies, the NS model structure is selected in order to meet major engineering needs for aerospace actuators. For this reason, it associates in series (from drive to load flanges):

- a perfect rotational to translational power transformer, with warning or error generated when the position is out of stroke
- an optional friction loss that applies to a rigid rod
- an optional and generic compliance that can represent in addition either backlash or preloading, depending on parameterisation.

### 2.5.1 Friction models

There are a lot of candidate options for friction modelling and parameterization [7]: efficiency (direct/indirect), parametric model of friction force (e.g. Coulomb or Stribeck), coefficient of friction with rod diameter and lead, etc. Using efficiency (direct and indirect) has been selected for non linear friction models because it is widely used by engineers. As a drawback, using efficiency exclusively does not allow to compute the no-load driving force or no-drive back-driving force. For non preloaded designs, a lossy rod model has been developed with the same structure as the MSL `LossyGear` [8].

For continuous models, the transition between sticking and sliding has been approximated by a hyperbolic tangent function in order to avoid triggering state events. In addition, removing any conditional statements has contributed to typically reduce the simulation time of the friction model by 25%. By setting negative values for the indirect efficiencies, irreversibility (impossible to back drive the NS) is parameterized without introducing an additional pa-

parameter for that case. When activated the thermal port of the friction model outputs the heat generated by friction.

### 2.5.2 Compliance models

In order to enable failure injection or design exploration, a non-linear model has been developed to reproduce backlash, pure compliance or preloading, depending on its parameterization. NS compliance has been modelled as a transmission device involving two unilateral elastic contacts that act in opposition. The compliance model structure was defined in order to maintain consistency with the MSL, as illustrated by table 3.

The MSL `Elastogap` model has been modified for non-linear continuous models in order to avoid any event triggering or conditional statements, even for structural damping. Each direction of contact between nut and screw has been modelled by a serial combination of a lossy-rod and an elasto-gap, with one elasto-gap being flipped. This choice has the major advantage of reproducing the friction force

due to preload without any modification of the friction model. For non continuous models, the strengthening length of contacts has been used as the unique parameter of preload and backlash, enabling continuous transition between these characteristics. Positive values correspond to backlash, zero values to pure elastic and negative values to preload conditions.

### 2.5.3 Fault triggering

The two major types of NS failure are jamming and free-run. Jamming has been modelled by increasing friction through modification of the friction model parameters. Free-run has been modelled by manipulation of the compliance model parameters: decreasing stiffness (linear model) or preload, then increasing backlash. The failure injection library can also be used to introduce progressive degradation through continuous evolution of friction and compliance parameters. This is of particular interest for virtual assessment of health monitoring strategies.

Properties of different nut screw models	MSL	Basic	Linear	Nonlinear Continuous	Nonlinear non continuous
Perfect power transformation					
Optionally activated ports of mechanical support	✓	✓	✓	✓	✓
Replaceable models and data records					
Warning or stop if out of stroke		✓	✓	✓	✓
Offset of displacement transformation		✓	✓	✓	✓
Optionally activated heat port	✓	✓	✓	✓	✓
Optional version for Failure injection (jamming or free-run)		✓	✓	✓	✓
Compliance			Linear spring	✓	✓
Friction loss	Direct/Inverse efficiency		Viscous damper	Direct/Inverse efficiency	Direct/Inverse efficiency
Irreversibility				✓	✓
Preload				✓	✓
Backlash				✓	✓
Effect of velocity on friction				Planned	Planned
Effect of temperature on friction				Planned	Planned

Table 3: Properties of A2015 library nut screw models

### 2.5.4 Model verification

All models have been designed to accept implementation in causal simulation software. In addition, their ability to run correctly for any causality case in the Modelica environment (any combination of two imposed variables out of drive torque, drive velocity, load force, and load velocity) has been addressed in detail.

### 2.5.5 Future work

The last part of the model development will include the consideration of the dependency of friction on velocity and temperature. Unfortunately, these effects are rarely documented in supplier's catalogues and are generally considered as confidential. However, it is intended to enable varying efficiencies versus velocity and temperature, either through a parametric equation or by a 3-D table.

## 2.6 Sensors

The A2015 library includes models for linear and angular position sensors, force sensors, torque sensors, current sensors, and temperature sensors.

They extend ideal sensor models from the MSL [9] and add effects and failures to the sensor output that are present in real sensor devices. Offset, non-linearity, saturation, and temperature dependency, as well as hysteresis, drift, and signal discretization are the effects taken into account. Most common sensor failures, such as open-circuit and short-circuit, will also be implemented.

The sensor models are parameterized according to typical data sheets in order to provide the end user with a practical library. This approach results in very general sensor models, independent from the sensor technology, which can be easily customized.

Figure 4 shows the structure of the Sensors package, which follows the general modeling approach for the whole library: different model levels (here: `Ideal`, `Linear`, and `NonLinear`) sharing a common interface, parameters included in records, and a bus connector predefined with the ideal and real signals of the sensor, and a status flag.

## 2.7 Controllers

### 2.7.1 Package Structure

This package contains blocks suitable for designing the control system of actuators [10]. The blocks can be combined with actuators of different modelling levels.

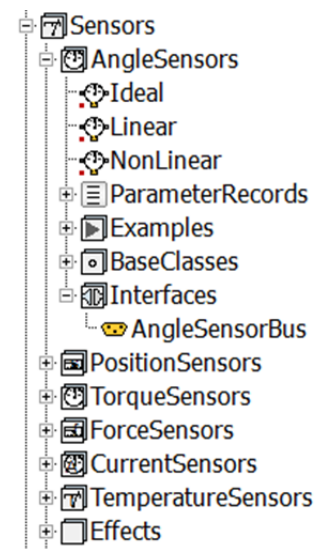


Figure 4: Structure of Sensors package

The structure of the Controllers package is shown in figure 5. Due to the function of the models within the package, the structure of the Controllers package follows a slightly different approach than other packages of the A2015 library.

The package consists of four sub-packages. The Continuous and Discrete sub-packages contain basic controller blocks built up from proportional, integrator and derivative blocks. The main purpose of the blocks in the Filters sub-package is to smooth control demand signals or filter measurements. The last sub-package contains blocks performing direct and inverse Park transformations. The package contains wrapper classes that allow the user to conveniently exchange blocks with similar functionality. The blocks in the library are parameterised via dedicated parameter records, organized in a separate sub-package.

### 2.7.2 Description of selected blocks

The PID controller variants include a symmetrical output saturation and integrator anti-windup (when applicable) and can be parameterised according to both the serial and parallel convention. They differ from the counterpart in the Modelica Standard Library in the implementation of the integrator anti-windup. Within the A2015 library, the integrator anti-windup is based only on the output of the integrator, not on the overall output. The output saturation is therefore implemented separately.

The filter blocks include both continuous and discrete time blocks, as well as general blocks that can be parameterised by specifying the numerator and

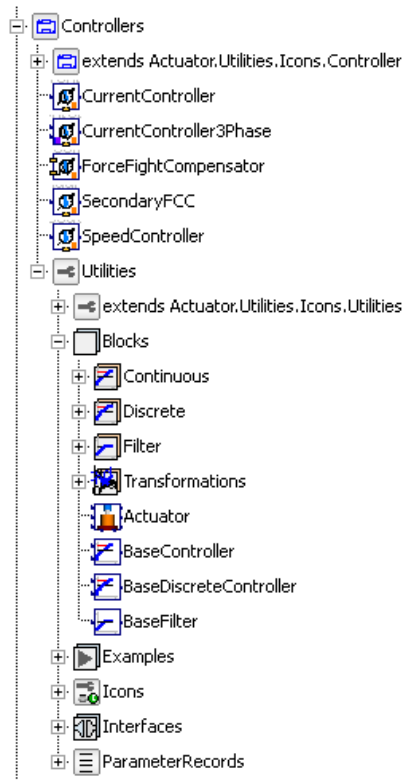


Figure 5: Structure of the Controllers package.

denominator of their transfer functions and also the well-known continuous time filter types available in the Modelica Standard Library (Butterworth, type I Chebyshev, etc). These model classes are wrappers for the blocks from the Modelica Standard Library in order to allow the user to switch between the different filter types.

A specialised force-fight compensator, the position, speed and current controllers are available as distinct standalone blocks. The measurements required by these blocks are collected via specialised sensor buses (see chapter 2.5). The user is free to select the controller and control demand filter types. These blocks can be connected to motor models via a three-phase interface and also to simplified motor models with one input. This is used in the two variants of the current controller, one of which performs space vector control.

### 2.7.3 Failure modes

The actuator control library also provides the user with a means for simulating the effects of various control failure scenarios. The following failure modes will be implemented: control output freeze, controller reset, short spikes in the controller output, non-return to zero error, runaway and no valid data faults.

## 3 Application example

To test the library and give a realistic example, a linear actuator based on an A320 aileron actuator has been modeled using the described library. The actuator model is set up using the base class `LinearGearedEMA`. This is a predefined base class for linear actuators which fully supports all redeclaration schemes of the library. It is extended by a position controller and an extra position sensor.

The final model can be seen in figure 6. In the model, the multi disciplinary approach of the library is directly visible. Electrical, mechanical and thermal effects are taken into account as well as the actuator control.

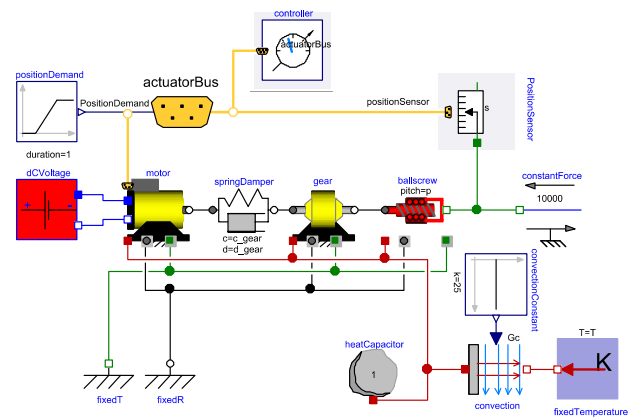


Figure 6: Overview of complete Linear EMA

### 3.1 Model structure

The mechanical model is built using a 2x1DOF approach. The rotational as well as the translational forces and support forces generated by each part are accessible outside the model. This may be used for the design of anti-rotation devices. The parameter data of the motor, gear, inverter, nut screw and controller are stored in parameter records. They extend from base records of these data to facilitate easy re-declaration in the models.

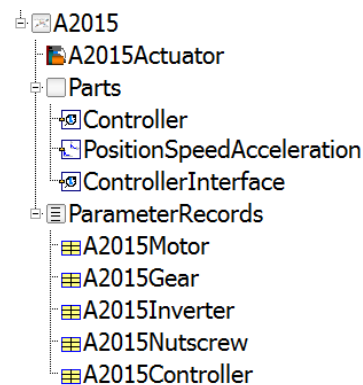


Figure 7: Overview of the library with the components for a specific linear EMA



In figure 7 an overview of the library for this specific linear EMA is given. All specific parts are combined in the package `Parts` and all parameters in the package `ParameterRecords`.

### 3.2 Simulation results

Simulation results of the described actuator can be seen in figure 8. The shown current response is the result of a position step input of 4.3 mm. In the Modelica model, the aerodynamic damping is neglected and a continuous controller is used. For comparison the simulation result of a discrete Simulink reference model is also plotted. Both simulations show a comparable dynamic response.

By modelling the controllers using the existing discrete blocks, it is possible to also model the effects of discrete controllers and increase the match between the model results.

To showcase the more advanced features of the library, an advanced model, including motor saturation and motor ripple is included. The effect of a motor ripple is visible from 1.15 to 1.3 seconds.

The shown simulation run takes 0.2 seconds of simulation time on an Intel® Xeon® E5-1620 Processor. The reference model in Simulink takes 6.5 seconds simulation time on the same machine. Including a sampled model using the same sample time as the Simulink model increases the simulation time to 1.7 seconds.

The frequency response of an actuator is also an important design criterion. It can be calculated using the `FrequencyResponse` package [11] by T. Bunte. The amplitude and phase response for 3 amplitude values (1, 5 and 10 mm) is calculated. The response from the input reference to the output rod is shown in figure 9. The frequency response of the same system with different amplitude changes depending on the stroke. This clearly shows the nonlinearity of the system. As expected, the system cannot follow the high frequencies at high amplitudes due to limitations on the motor current.

## 4. Conclusion and outlook

In this paper we have presented a library for modeling Electromechanical Actuators (EMAs). In the library, electrical machines, inverters, controllers, mechanical parts (rotational and translational) as well as sensors have been included. All model components have been set up using replaceable parameter sets and replaceable parts. The standardization of the interfaces of the components using partial classes

enables the user to build an EMA model as a modular system. This supports the exchange of models between partners and makes it possible to reuse these models for different simulation tasks without a structural redesign. The integration of component faults helps to quickly assess the actuator performance for failure cases.

To test the library, an existing Simulink model has been reworked into a Modelica model using mostly the components from the library. The results of the Modelica model show a good agreement with the Simulink reference model.

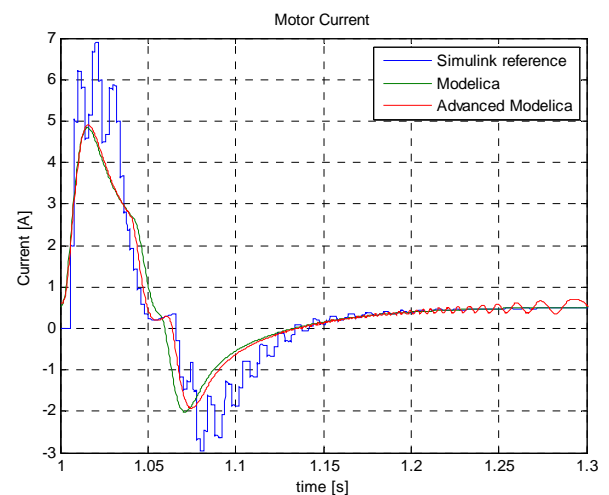


Figure 8: Simulation of actuator system

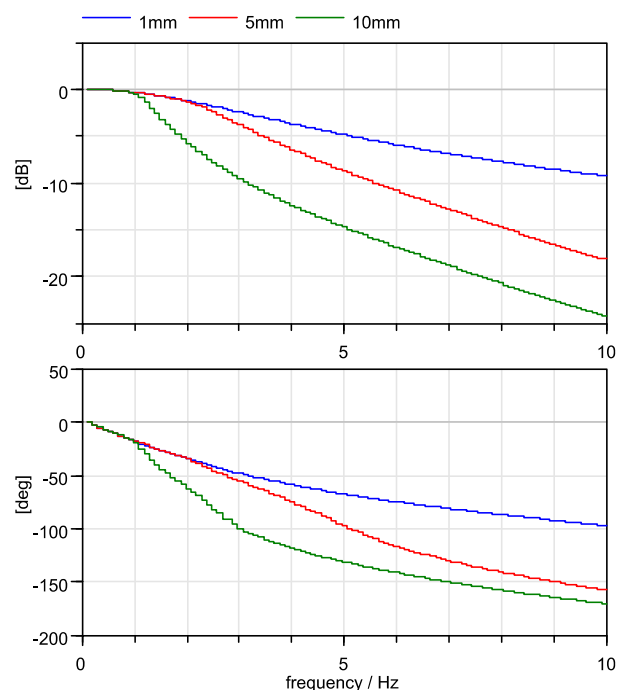


Figure 9: Frequency response of controlled actuator from position reference input to rod output.

The presented library is developed with a strong focus on Electromechanical Aircraft and Helicopter Actuators. This is not the only field the library can be used; It can help developers of all electromechanical actuators like automotive actuation (air path actuators for diesel engines, automatic hatch door actuation) , automation systems (packaging machines) or printing systems.

Possible future library extensions include mass, volume, cost, etc. design optimization (by providing specialized modeling levels while maintaining the EMA model topology) and automated safety and reliability analysis.

The availability of the library in- and outside of ACTUATION 2015 is currently under clarification by the partners of ACTUATION2015.

## Acknowledgement

The research leading to these results has received funding from the European Union's Seventh Framework Program (FP7-284916) for ACTUATION 2015 under grant agreement no. 284915.

## References

- [1] [www.actuation2015.eu](http://www.actuation2015.eu)
- [2] [www.modelica.org](http://www.modelica.org)
- [3] C.I. Hill, P. Giangrande, C. Gerada and S.V. Bozhko, Implementation of a Multi-Level Power Electronic Inverter in Modelica. *In proceedings of the 10<sup>th</sup> Modelica Conference*, 2014.
- [4] N. Urasaki, T. Senjyu and K. Uezato, A Novel Calculation Method for Iron Loss Resistance Suitable in Modeling Permanent-Magnet Synchronous Motors, *IEEE Transactions on Energy Conversion*, vol. 18, no. 1, pp. 41-77, March 2003.
- [5] D. Winkler and C. Gühmann, Modelling of Electrical Faults in Induction Machines Using Modelica, *Proceedings in 48th Scandinavian Conference on Simulation and Modeling (SIMS)*, 2007.
- [6] Maré J-C., 2-D Lumped parameters modelling of EMAs for advanced virtual prototyping of EMAs, *Proceedings of Recent Advances in Aerospace Actuation Systems and Components*, pp 122-127, Toulouse, France, June 13-14, 2012
- [7] Maré J-C., Friction modelling and simulation at system level: a practical view for the designer, *ImechE part I, Journal of Systems and Control Engineering*, Volume 226 Issue 6, pp. 728 - 741, July 2012
- [8] Otter M., Elmqvist H., Mattson S. E., Hybrid modelling in Modelica based on the synchronous data flow principle, *1999 IEEE Symposium on Computer-Aided Control System Design*, pp. 151-157, August 22-26 Hawaii, USA, 1999
- [9] MSL: Modelica Standard Library, [www.modelica.org/libraries](http://www.modelica.org/libraries)
- [10] I. Réti, M. Lukátsi, B. Vanek, I. Gőzse, Á. Bakos, J. Bokor, Smart mini actuators for safety critical unmanned aerial vehicles, *2nd International Conference on Control and Fault-Tolerant Systems (SysTol)*, Nice, France, 2013.
- [11] Bünte, T., Recording of Model Frequency Responses and Describing Functions in Modelica. In: *Proceedings of the 8<sup>th</sup> International Modelica Conference*, Technical University Dresden, Germany (pp. 686–696), 2011.
- [12] Linden, F. L. J. Van der, General fault triggering architecture to trigger model faults in Modelica using a standardized blockset. *In proceedings of the 10<sup>th</sup> Modelica conference, 2014*
- [13] Functional Mock-up Interface [www.fmi-standard.org](http://www.fmi-standard.org)
- [14] M. Kuhn, M. Otter, and L. Raulin. (2008): A Multi Level Approach for Aircraft Electrical Systems Design. In: *Proceedings of the 6<sup>th</sup> International Modelica Conference*, Bielefeld, Germany, pp. 95-101, 2008