ITEA Office
High Tech Campus 69 - 3
5656 AG Eindhoven
The Netherlands
T + 31 88 003 6136
E info@itea3.org
W www.itea3.org

ITEA 3 is a EUREKA strategic ICT cluster programme

EMPHYSIS – D2.1 Demonstrator scenario for test cases

ITEA 3

Version 2.0, February 27th, 2019

Project Acronyms

eFMI	Functional mock-up interface for embedded systems
eFMU	Functional mock-up unit for embedded systems
DAE	Differential algebraic equation

From Full Project Proposal

Task ID	Task Description	Task Leader	Contributor
T2.1	*Demonstrator scenario for test cases* In this task simple, generic models and controllers are provided in Modelica, Simulink, Amesim or other relevant forms. They will be used to evaluate the tool chains capabilities for eFMI to fulfil the WP1 requirements. It is intended to provide these models in open source (Modelica) or on a free access (Amesim, others) to support eFMI testing and evaluation out of the EMPHYSIS consortium. The Modelica models will be published in an open source Modelica library, in order to be used as benchmarks for Modelica tool vendors.	DLR	Bosch, CEA, DS-SE, FH, GIPSA, LiU, Maplesoft, Siemens-SAS, UAntwerp
D2.1	This deliverable consists of open source models developed in T2.1 that shall be used in T7.1 for evaluation of the tools for eFMI code generation.		

Table of Contents

1. Introduction

In this scenario generic models and controllers provided in Modelica, Simulink, Amesim or other relevant forms are used to evaluate the tool chain capabilities for eFMI and fulfilment of the WP1 requirements. The models are provided in open source (Modelica) or on a free access (Amesim, others) basis to support eFMI testing and evaluation outside of the EMPHYSIS consortium. The Modelica models are available in the attached open source Modelica library EMPHYSIS_TestCases, in order to be used as benchmarks for Modelica tool vendors.

In order to make sure that all relevant advanced model/controller structures can be handled by eFMI, as simple as possible benchmarks for the intended models/controllers are provided.

2. Models for the Test Cases

The models of the test case scenarios are listed in the following table. Each test case has a unique identification number to clearly mark and recognize the different cases within this document and within the model libraries.

No	Problem	Source	Control/Estimation Methods
1	Velocity Control 1	Modelica	PI Controller
2	Velocity Control 2	Modelica	PID Controller with anti-windup
3	Velocity Control 3	Modelica	PI with anti-windup
4	Reduction of Drivetrain Oscillations	Modelica	Linear inverse model (feedforward) + PI with anti-windup (feedback)
5	Mixing Reactor Control 1	Modelica	Nonlinear inverse model (feedforward) + P controller (feedback)
6	Semi-active suspension control	Modelica	Skyhook/Groundhook controller
7	Crab Estimation	Modelica	Extended Kalman Filter
8	Academic examples	Modelica + C	Root finding of nonlinear functions
9	Mixing Reactor Control 2	Modelica	Input/output linearization
10	Slider Crank Mechanism	Modelica	Nonlinear inverse model (feedforward) + P Controller (feedback) leading to nonlinear algebraic equations in the model
12	Fault Diagnosis	Modelica	Residual generator
13	Drivetrain Virtual Sensor	Amesim	Virtual sensor (PID)
14	Compact Code	Modelica	Performance Assessment
15	Stiff non-linear system	Modelica	Performance Assessment
16	Large multi-dim. maps	Modelica	Performance Assessment
17	Sparse	Modelica	Performance Assessment
18	Speed control	Amesim/Matlab	Model Predictive Control

The properties of the test cases are listed in the following table:

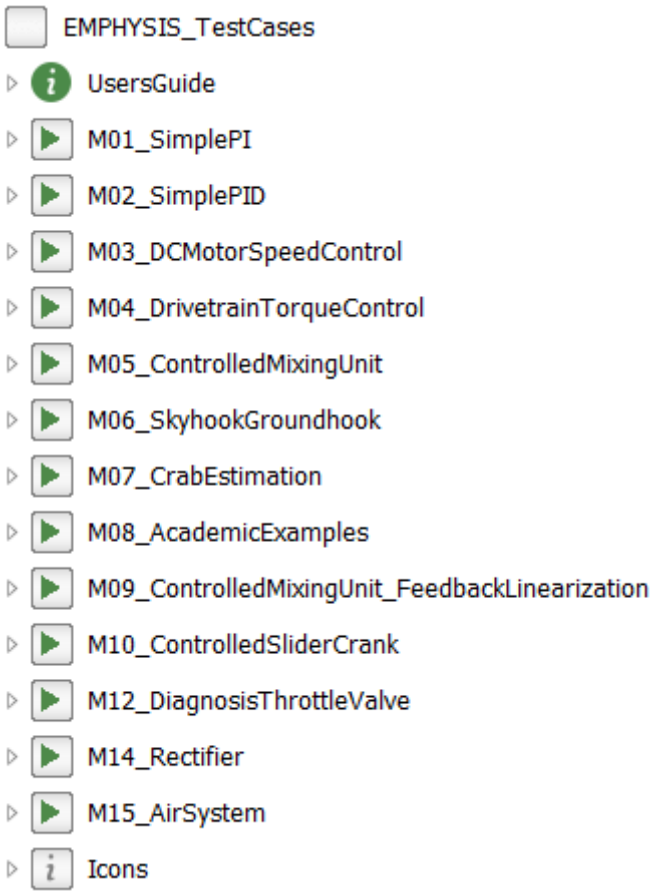
No	Continuous time	Discrete time	Linear	Non-Linear	Lookup-Tables	Events	DAE	Optimization
1	X	X	X					
2	X	X		X		X		
3	X	X		X		X		
4	X	X		X		X		
5		X		X				
6		X		X		X		
7	X			X				
8				X				
9				X				
10	X	X		X		X	X	
12	X		X					
13	X	X		X	X	X		
14	x	x		x	x	x	x	
15				X				
16				X	X			
17				X	X			
18	X	X	X	(X)		X	X	X

The different properties of models are defined as follows:

Property	Explanation
Continuous time	The mathematical description of the model equations is represented in continuous time.
Discrete time	The mathematical description of the model equations is represented in discrete time, e.g., discretization of a continuous-time representation by an numerical integration scheme.
Linear	The controller model is a linear model.
Nonlinear	The controller model is a nonlinear model.
Lookup-Tables	The model contains “lookup tables”, i.e., tables / matrices of numerical data, that are used to model functional dependencies between variables.
Events	The model equations contain non-smooth functions that require separate handling.
DAE	The representation of the dynamic model is a differential algebraic equation system and not an ordinary differential equation system.
Optimization	The controller model contains numerical optimization methods.

2.1. Modelica models

Many models of the test cases are Modelica models. Therefore, these models are collected within one Modelica library “EMPHYSIS_TestCases”:

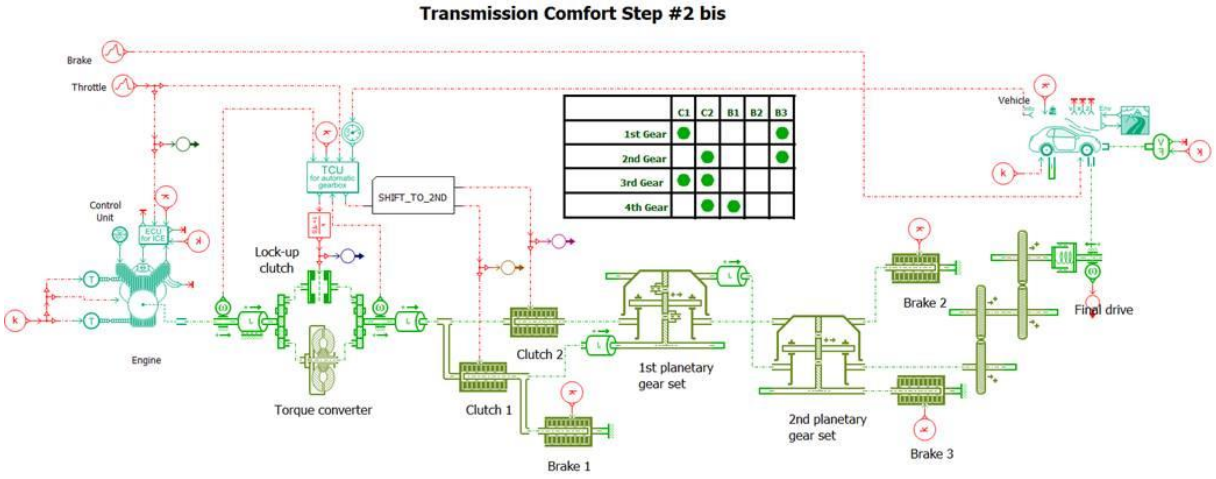


Each of the Modelica test cases is contained in a separate sub-package of the library. The background and the intention of the model is described within each of the sub-packages in the Modelica library. An html-documentation of the library is available here:
Modelica/EMPHYSIS_TestCases/help/EMPHYSIS_TestCases.html

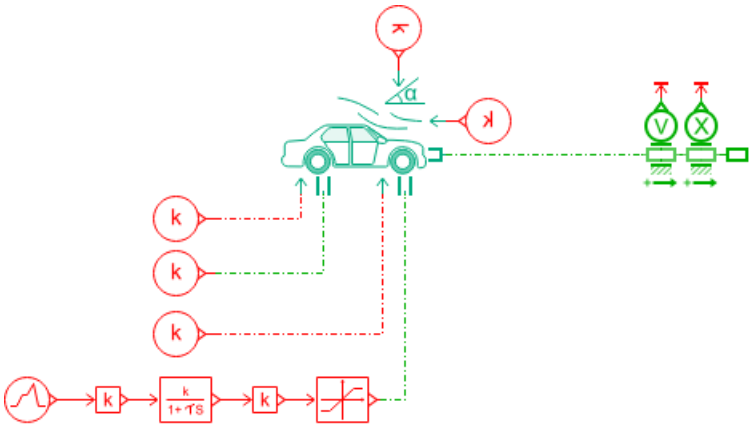
2.2. Amesim models

The Amesim test cases models are available and controls will be developed in the following EMPHYSIS activities. These will be built in collaboration with the Amesim development team during work in WP4 and the evolution of the implementation of the eFMUs in Amesim.

Virtual sensor of a drivetrain to estimate engine speed. This sensor will be used together with a Kalman filter. This sensor will feed a simple map-based controller of the automatic gearbox.
For that a map-based engine model is used. The virtual sensor will be fed with the control input to estimate the state of the system. The Kalman filter will correct the state based on the sensor input of the other sensors.



Model predictive controller
A model to act on the speed of the car when an object is crossing its path. For now, a simple linear drivetrain is used. This can be later upgraded with a non-linear model for the drivetrain. The model only allows movement of the car in one direction.



3. Conclusion

Once finalized and jointly assessed by EMPHYSIS partners with first eFMI implementations and demonstrations, these models will serve as development and testing support for the tools providers and will be made accessible along the eFMI standard for the eFMI community.