

Towards an FMI Layered Standard for DAE

Applications for Simulation and Optimization

American Modelica & FMI Conference 2026

October 12th-14th, 2026 in Atlanta, GA, USA

Elmir Nahodovic, Andreas Heuermann, Joel A. E. Andersson, Adwait Verulkar, Srikanth Sivaramakrishnan, Masoud Najafi, Linus Langenkamp, Christian Bertsch, Torsten Sommer, Erik Henningsson, Hans Olsson, Bernhard Bachmann

Presenters: Joel A. E. Andersson, Adwait Verulkar

© 2026 FMI Modelica Association Project | www.fmi-standard.org

Modelica Association

Motivation

Physical models, e.g. in Modelica, are often DAEs. But FMI Model Exchange exposes only ODEs.

STANDARD FMI MODEL EXCHANGE

Algebraic loops solved inside the FMU

$$\dot{x} = f(x, g^{-1}(x, u, t), u, t)$$

where $w = g^{-1}(x, u, t)$ is the solution to

$$g(x, w, u, t) = 0$$

FMU internal solves; algebraic structure hidden.

LS-DAE

Expose the equations to the importer

$$\dot{x} = f(x, w, u, t)$$

$$0 = g(x, w, u, t)$$

Importer sets *w* and drives residuals to zero.

Accuracy

Remove inner-solver tolerances from residual evaluations.

Efficiency

Avoid nested Newton solves; exploit the coupled Jacobian.

Robustness

Avoid hidden solver state; enable global error recovery.

© 2026 FMI Modelica Association Project | www.fmi-standard.org

LS-DAE Working Group

Started in 2025, bi-weekly meetings lead by Joel Andersson, Andreas Heuermann

Members include: Claytex, Dassault Systèmes, MapleSoft, Modelon, OpenModelica Altair, AVL, CasADi/FMIOPT, DLR, Dynamica, TUE, GM Motorsports (collaborator)

GitHub: modelica/fmi-ls-dae

Alpha.1 released 11 September 2026:

https://github.com/modelica/fmi-ls-dae/releases/tag/v1.0.0-alpha.1

Use Cases

- Optimization
- DAE simulation
- Steady State
- Connecting DAE FMUs

Tool Prototypes/implementations

- Dymola
- CasADi/FMIOPT
- SimCenter TwinActive
- MOO
- Fmusim
- OpenModelica

Layered Standard Alpha.1

- Mathematical Formulation
- LS Manifest
- Examples

© 2026 FMI Modelica Association Project | www.fmi-standard.org

Design overview

- Starting point: 2018 proposal FMI for DAE
- Semi-explicit DAE Formulation
  - Residuals are represented as variables
  - Index-1 assumed (for now)
- Layered Standard
  - Extension of FMI 3 Model Exchange
  - Still a valid standard (ODE) FMU:
    - DAE is an **opt-in** via structural parameters and (re)configuration
  - Addition to XML (next slide)
  - No additional functions in C-API
    - But now permitted to set algebraic variables (that are *local*)
    - Extended derivative calculations to algebraic variables

Algebraic variables

$$\dot{x} = f(x, w, u, t)$$

$$0 = g(x, w, u, t)$$

Residual equations

$$y = h(x, w, u, t)$$

Details in the paper:

Towards an FMI Layered Standard for DAE – Applications for Simulation and Optimization

Elmir Nahodovic, et. al

© 2026 FMI Modelica Association Project | www.fmi-standard.org

FMI-LS-MANIFEST

- \extra\org.fmi-standard.fmi-ls-dae\fmi-ls-manifest.xml
- New structural parameter for switching between ODE and DAE
- List of algebraic variables
  - Local variables in ModelDescription.xml
- New ModelStructure section, including:
  - Residual variables <**Residual** ...>
  - Updated **dependencies**, **dependenciesKind**
    - Direct dependencies of algebraic variables included
    - Indirect dependencies of algebraic variables removed

```
<EnableDAEParameter valueReference="0" />
<AlgebraicVariables>
  <AlgebraicVariable valueReference="1" />
</AlgebraicVariables>
<ModelStructure>
  <!-- y -->
  <Output
    valueReference="2"
    dependencies = "..."
    dependenciesKind = "..."/>
  <!-- der(x) -->
  <ContinuousStateDerivative
    valueReference="3"
    dependencies = "..."
    dependenciesKind = "..."/>
  <Residual
    valueReference="4"
    dependencies = "..."
    dependenciesKind = "..."/>
</ModelStructure>
```

© 2026 FMI Modelica Association Project | www.fmi-standard.org

Example XML – Roberts model

- Reference FMU: https://github.com/modelica/reference-fmus
- Index-1 problem from chemical kinetics, included in the example programs for Sundials IDA

$$\frac{dy_1}{dt} = -0.04y_1 + 10^4 y_2 y_3$$

$$\frac{dy_2}{dt} = 0.04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2$$

$$0 = y_1 + y_2 + y_3 - 1$$

$$y_1(0) = 1, \qquad y_2(0) = y_3(0) = 0.$$

```
<EnableDAEParameter valueReference="1" /> <!-- Enable
<AlgebraicVariables>
  <AlgebraicVariable valueReference="6" /> <!-- y3 -->
</AlgebraicVariables>
<ModelStructure>
  <!-- derivative of y1 -->
  <ContinuousStateDerivative
    valueReference="2"
    dependencies = "2 4 6"
    dependenciesKind = "fixed fixed fixed"/>
  <!-- derivative of y2 -->
  <ContinuousStateDerivative
    valueReference="5"
    dependencies = "2 4 6"
    dependenciesKind = "fixed fixed fixed"/>
  <!-- algebraic equation -->
  <Residual
    valueReference="7"
    dependencies = "2 4 6"
    dependenciesKind = "fixed fixed fixed"/>
</ModelStructure>
```

© 2026 FMI Modelica Association Project | www.fmi-standard.org

Example XML: DAE vs. ODE

Original (**ODE**)

```
<!-- derivative of y1 -->
<ContinuousStateDerivative
  valueReference="2"
  dependencies = "2 4"
  dependenciesKind = "dependent dependent"/>
<!-- derivative of y2 -->
<ContinuousStateDerivative
  valueReference="5"
  dependencies = "2 4"
  dependenciesKind = "dependent dependent"/>
```

Dependencies are non-linear

$$\frac{dy_1}{dt} = -0.04y_1 + 10^4 y_2(1 - y_1 - y_2)$$

$$\frac{dy_2}{dt} = 0.04y_1 - 10^4 y_2(1 - y_1 - y_2) - 3 \times 10^7 y_2^2$$

LS-DAE (**DAE**)

```
<!-- derivative of y1 -->
<ContinuousStateDerivative
  valueReference="2"
  dependencies = "2 4 6"
  dependenciesKind = "fixed fixed fixed"/>
<!-- derivative of y2 -->
<ContinuousStateDerivative
  valueReference="5"
  dependencies = "2 4 6"
  dependenciesKind = "fixed fixed fixed"/>
```

Linearity is preserved

Algebraic variable is “known”

$$\frac{dy_1}{dt} = -0.04y_1 + 10^4 y_2 y_3$$

$$\frac{dy_2}{dt} = 0.04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2$$

$$0 = y_1 + y_2 + y_3 - 1$$

© 2026 FMI Modelica Association Project | www.fmi-standard.org

Tool prototypes and examples

OpenModelica

Export and **import**

OpenModelica on the Web

playground.openmodelica.org/demo/simulator

Dymola

Export for v.1.0.0-alpha.1 with 2027x

SimCenter Twin Activate

Import

CasADi

Export and **import** (v. 3.8.1)

fmusim

Import for v.1.0.0-alpha.1

Dynamic optimization example: fmu_collocation.py

FMIOPT

MOO: Modelica / Model Optimizer

Import

© 2026 FMI Modelica Association Project | www.fmi-standard.org

Usage examples from GM Motorsports

- Dynamic optimization for a multilink suspension model
  - FMI-LS-DAE exported from Dymola
  - Dynamic optimization with MOO or CasADi/FMIOPT
  - Details in paper
- Minimal lap-time optimization for a production F1 model
  - Next slides

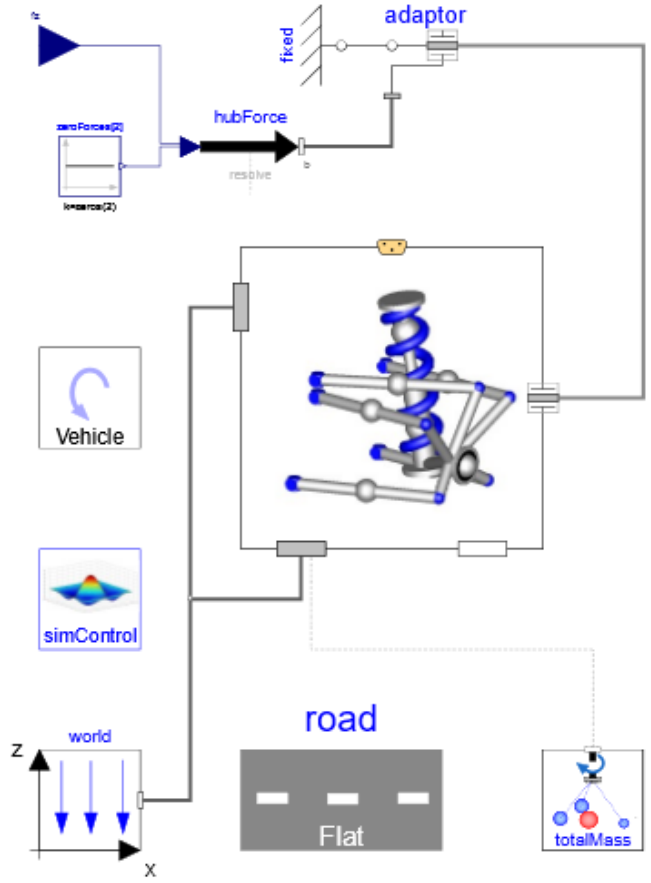


Figure 6. Multilink rear suspension corner model. The actuator force f_z is the control input and the joint angle ϕ is the tracked output.

© 2026 FMI Modelica Association Project | www.fmi-standard.org

LS-DAE Formula 1 Example

© 2026 FMI Modelica Association Project | www.fmi-standard.org

Why lap-time optimization in Formula 1?

Track time is scarce, so most setup and strategy decisions are made in simulation first.

1. **Limited testing**
   Few test days, cost cap and restricted aero testing: the car must arrive at each circuit close to its best setup.

2. **The “perfect driver”**
   Minimum-lap-time optimal control finds the fastest lap a car can do. No driver model to tune; setups are compared on equal terms.

3. **2026 rules raise the stakes**
   350 kW MGU-K ≈ ICE power, 8.5 MJ harvest per lap, active aero. Energy use is now as big a lap-time lever as setup.

4. **One model for simulation and optimization**
   The team’s Modelica vehicle model already drives the simulators. Optimizing a second, simplified copy causes drift.

USE CASE 1 · SETUP PER CIRCUIT

Which car for which track?

- Wing level, ride heights, balance, differential, brake bias
- Lap-time sensitivities to mass, CoG, aero and grip

USE CASE 2 · ENERGY MANAGEMENT (EMS)

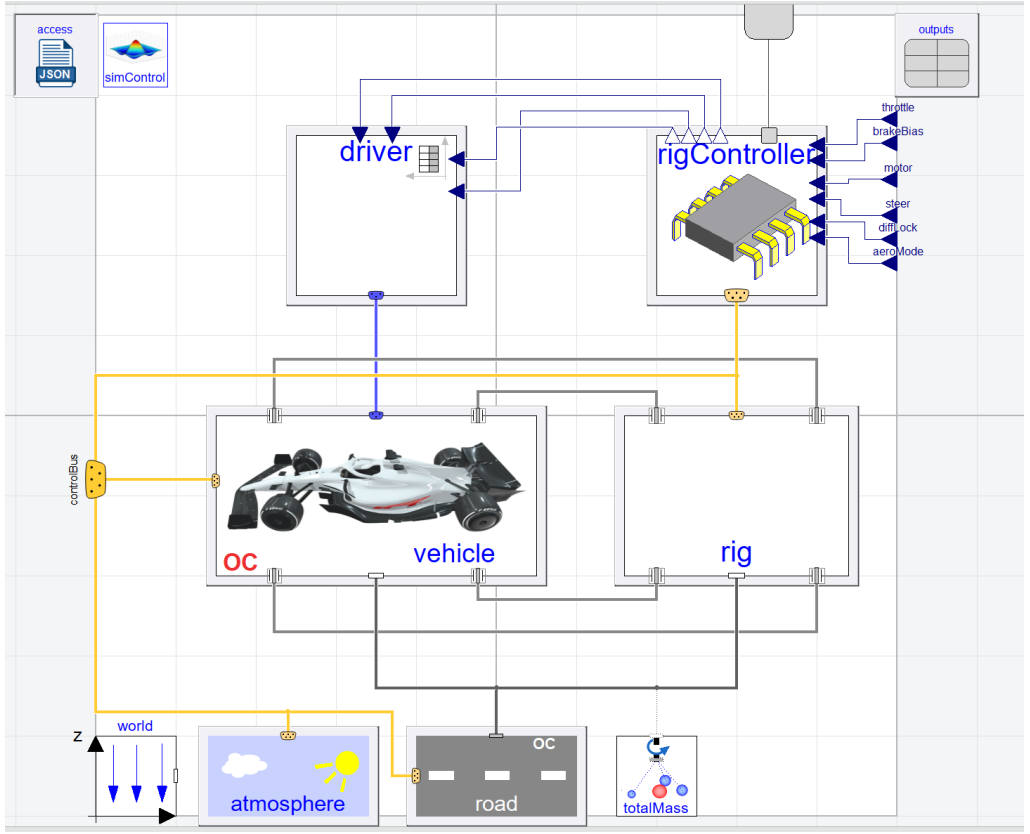
Where to deploy, where to harvest?

- 350 kW MGU-K within a 4 MJ battery window
- Harvest cap per lap, speed derate, active-aero zones

Goal: optimize the production Modelica model directly
Exported once from Dymola as FMI 3.0 Model Exchange with LS-DAE; run on all 23 circuits.

© 2026 FMI Modelica Association Project | www.fmi-standard.org

F1 vehicle model as an FMI 3 LS-DAE FMU



Dymola test rig: vehicle, driver inputs, controller, road and atmosphere

33	16	6	12
states x	algebraic z	controls u	fixed inputs p

Chassis & suspension	multibody; multilink corners as closed kinematic loops
Tyres	combined slip, sliding-power limits
Aerodynamics	ride-height maps, active aero (corner / straight mode)
Power unit	ICE + 350 kW MGU-K, battery with losses
Track	3D road surface r(s, w), track limits

WHERE LS-DAE COMES IN

Each corner closes a loop through universal + spherical joints: 4 link angles per corner are algebraic. LS-DAE exposes them and their residuals instead of solving them inside the FMU.

© 2026 FMI Modelica Association Project | www.fmi-standard.org

Minimum-lap-time optimal control problem

Formulated in track distance s, so the lap length is fixed and lap time t(L) is a state.

$$\min_{x(\cdot),\, z(\cdot),\, u(\cdot)} \quad t(L) \;+\; \varepsilon\, R(u)$$

lap time + small control regularization

$$\text{s.t.} \quad \frac{dx}{ds} = \frac{1}{\dot{s}(x)}\, f(x, z, u, p)$$

vehicle dynamics from the FMU, s = track distance

$$0 = g(x, z, u, p)$$

LS-DAE residuals: closed suspension loops

$$w_R(s) \le w(s) \le w_L(s)$$

track limits

$$h(x, z, u, p) \le 0$$

tyre saturation, MGU-K power & speed derate

$$0 \le \mathrm{SoC}(s) \le 1, \quad E_{\mathrm{harv}}(L) \le 1$$

battery window 4 MJ, harvest 8.5 MJ / lap

$$0 \le a(s) \le a_{\max}(s)$$

active-aero zones

$$x(0) \approx x(L)$$

periodic, flying lap

$$u_{\min} \le u \le u_{\max}, \quad p = \bar{p}$$

actuator limits; setup offsets fixed

TRANSCRIPTION (FMIOPT + CasADi)

- Direct collocation, Radau order 3, 305 intervals
- IPOPT + MUMPS, quasi-Newton Hessian
- FMU Jacobians via FMI 3 directional derivatives, in parallel
- Barcelona: 193 iterations, 6.1 min solve time

WITH LS-DAE

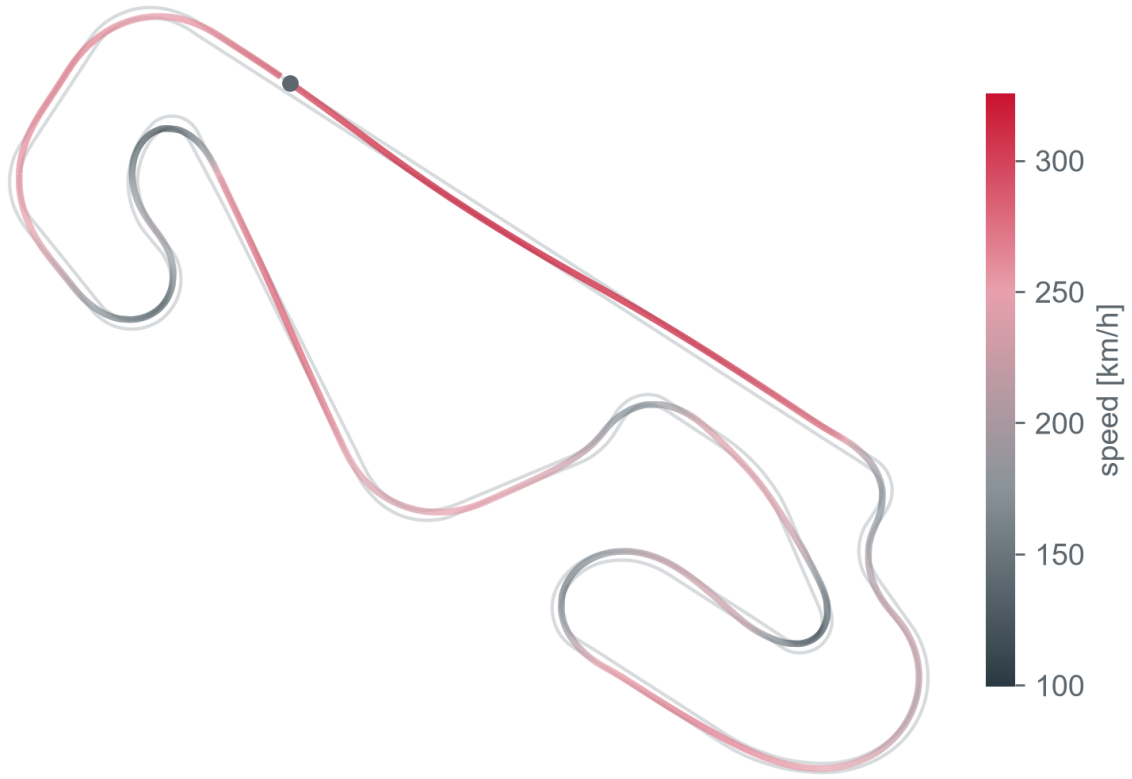
- z at every collocation point is an NLP variable
- g(x, z, u, p) = 0 becomes NLP constraints
- Sparsity read from the manifest; no inner Newton solves

One NLP per circuit and setup, solved from a static initial guess

Warm starts and batch mode run the full calendar, for setup studies and EMS strategy.

© 2026 FMI Modelica Association Project | www.fmi-standard.org

Result: optimal lap, Circuit de Barcelona-Catalunya



Spec car, Barcelona setup; colours show speed along the optimal path.

74.27 s
optimal lap time

6.1 min
wall-clock solve time

325 km/h
top speed

100 km/h
slowest corner

- Converged from a static guess: 20 m/s, mid-track
- Racing line, speed, aero mode and MGU-K use from one solve
- Full channel set exported to .mat for standard analysis tools

© 2026 FMI Modelica Association Project | www.fmi-standard.org

Summary LS-DAE

- Expose algebraic equations in FMI model exchange
- Long known benefits:
  - Accuracy
  - Efficiency
  - Robustness
- Novel: *Standardized* support in Dymola, OpenModelica etc. with realistic models
- Current status: Alpha.1: Ready to be used
- Encouraging industrial results, e.g. for F1 lap-time optimization

Outlook

- Agree on
  - Initialization
  - Event support
  - Coupled FMUs
  - Higher index DAEs
- Extensions to co-simulation?

More in the issue tracker

https://github.com/modelica/fmi-ls-dae/issues

© 2026 FMI Modelica Association Project | www.fmi-standard.org