Control of Mobile Robots

Introduction to multi-body simulation

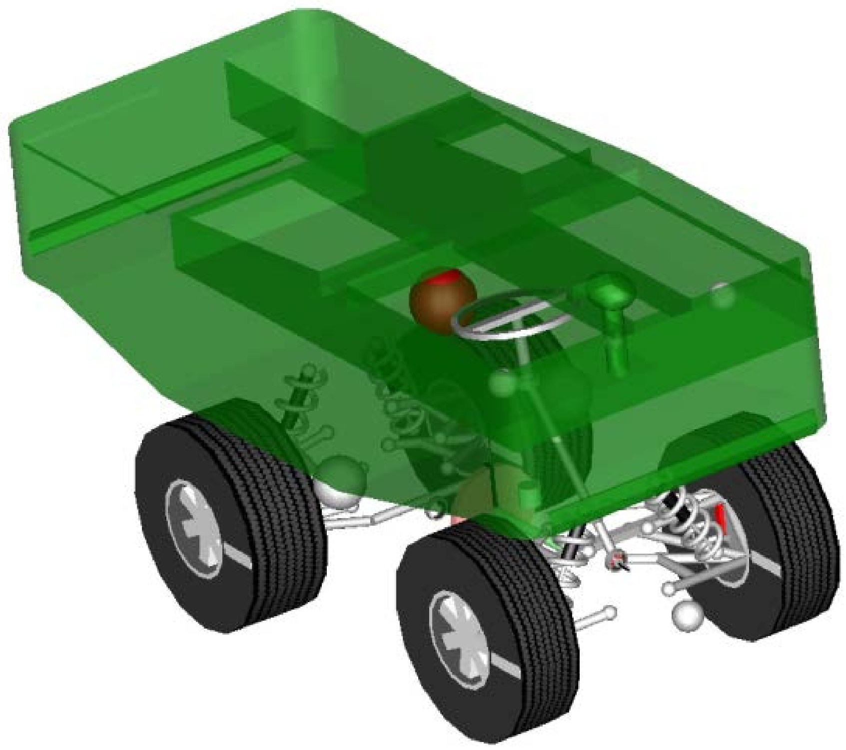
Prof. Luca Bascetta (luca.bascetta@polimi.it)

Politecnico di Milano – Dipartimento di Elettronica, Informazione e Bioingegneria

Now that we know the tools to develop a dynamic model of a mobile robot, why should we consider a multi-body simulator?

In a multi-body simulator a complex physical system is modelled as a composition of rigid or flexible bodies, exploiting modularity.

A bicycle model is a simplified tool to represent the dynamics of a four-wheel vehicle for model-based control purposes, but it could be not accurate enough to simulate the behavior of the vehicle for dynamic analysis and control system validation.



Modelica

Modelica is a language for modelling complex physical systems coming from different applications domains

- robotics
- automotive
- aircrafts
- satellites
- power plants...

It has been primarily designed for simulation of complex systems.

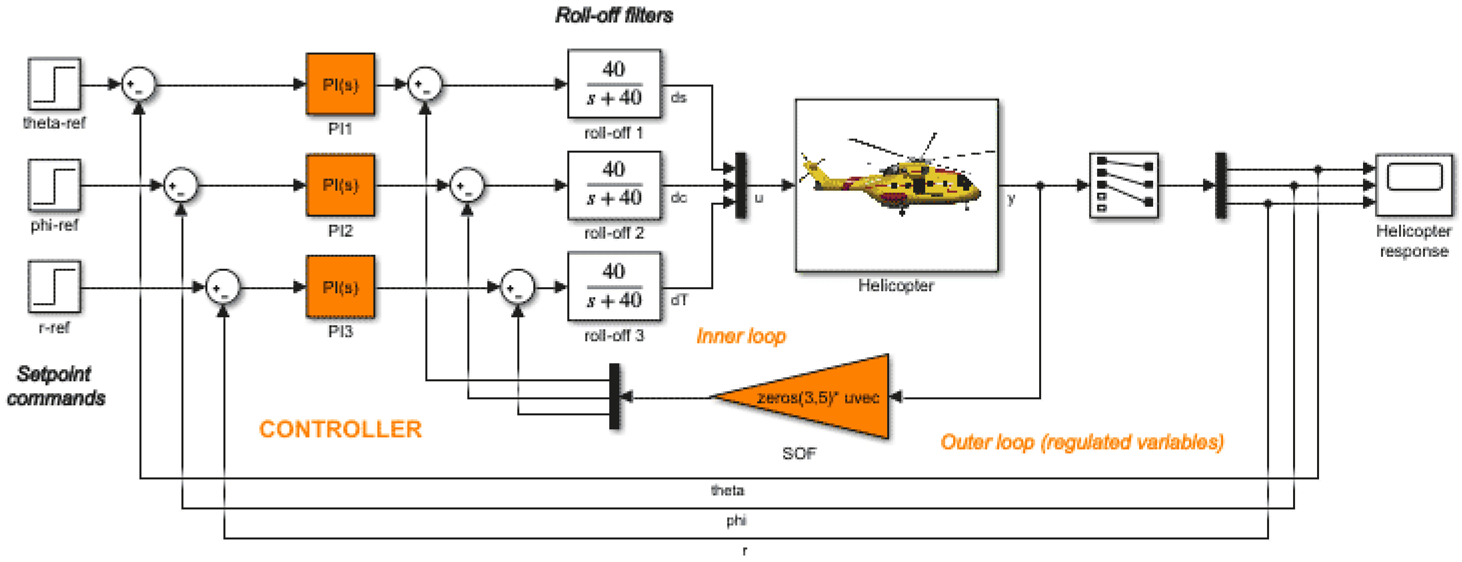
Modelica is free and the language specification is open (www.modelica.org).

There are several free and commercial tools that support modelling with Modelica (e.g., OpenModelica and Dymola)

Main features of Modelica language

- Declarative language
  - Equations and mathematical functions allow a-causal modelling
- Multi-domain modelling
  - Combine electrical, mechanical, thermodynamic, hydraulic, control, real-time,…
- Object-oriented modelling
  - Strongly typed object-oriented language
- Visual component programming
  - Visual programming ease to set up modular and hierarchical system architectures
- Efficient simulation
  - Efficiency comparable to C

Classical Simulink networks represent *causal models*.

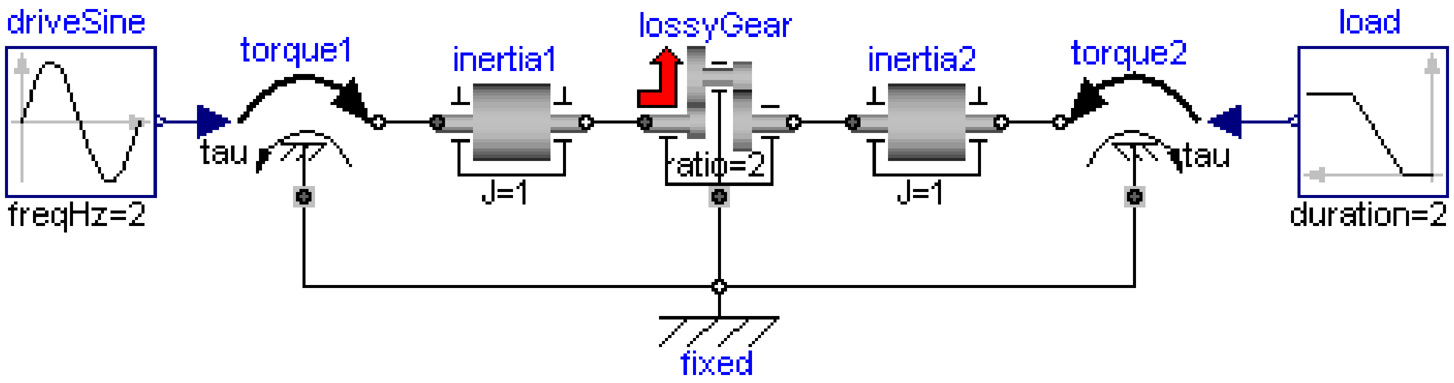


Signals flow in connections between blocks, transmitting values of individual variables from the output on one block to inputs to other blocks.

Processing of input information to output information takes place in blocks.

A-causal modelling

On the contrary, in a-causal declarative languages individual components of the model describe the equations directly and not the algorithm to solve them.



Interconnection of the components does not define the calculation procedure but the structure of the modelled physical system. The way of solving the equations is then “left up to the simulator”.

A-causal declarative modelling increase reusability of Modelica models with respect to models where the input-output causality is fixed.

A-causal modelling: a simple example

A resistor is modelled by the following equation

R*i=v;

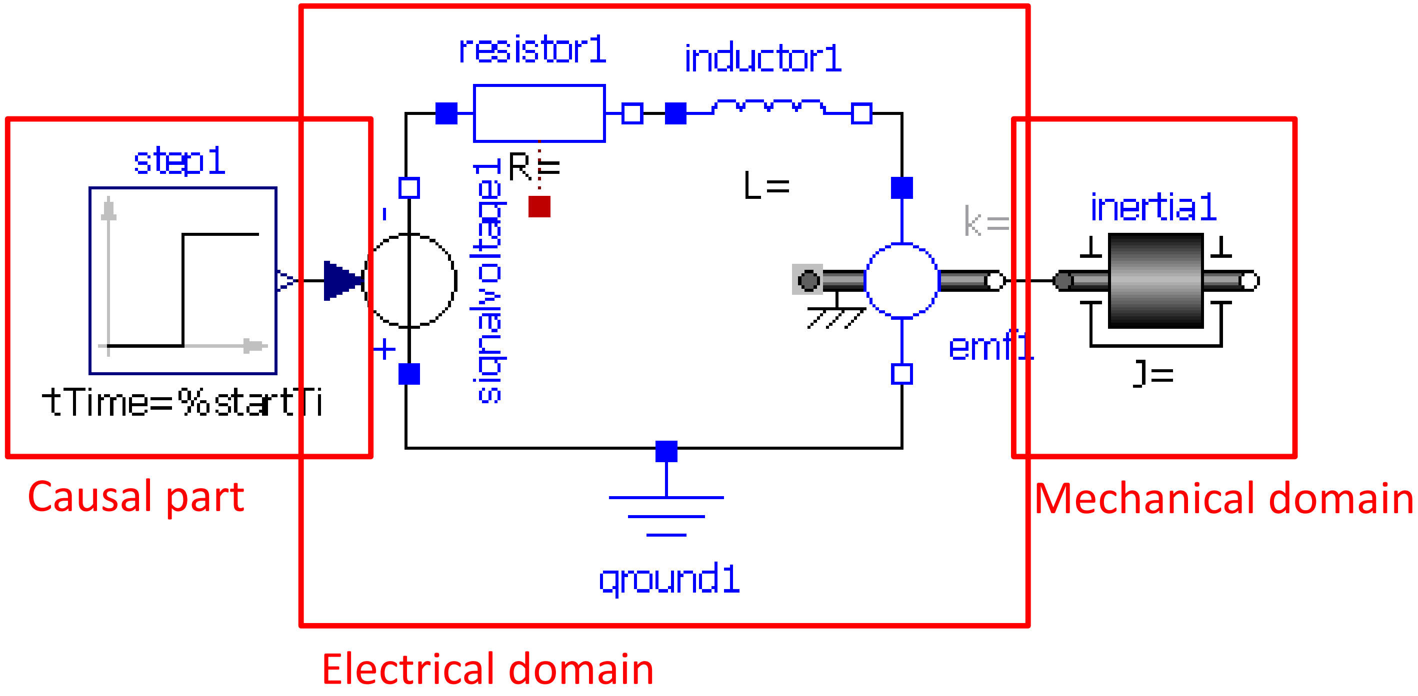
and can be used in three different ways

i:=v/R;

v:=R*i;

R:=v/i;

Another important feature of Modelica is the capability of modelling system coming from different physical domains.

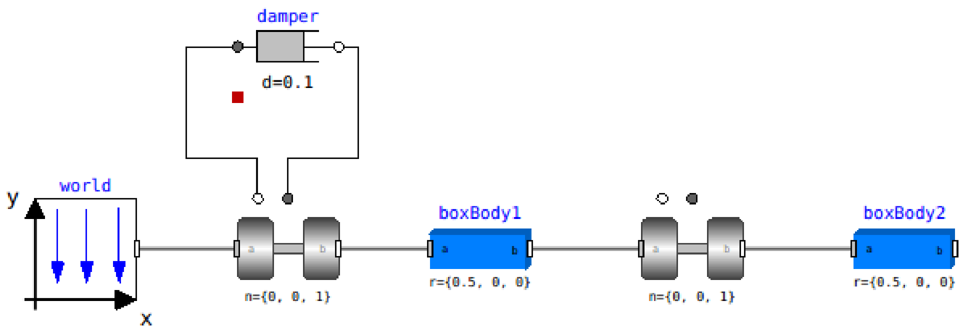


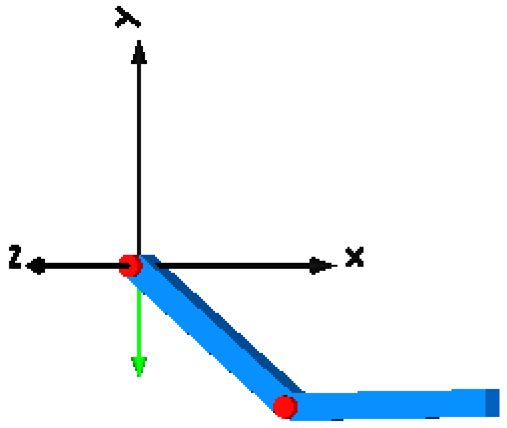
Modelica standard library includes many basic blocks to set up models in different domains.

We are particularly interested to consider the mechanical domain that includes

- Rotational, a library of 1-dimensional rotational mechanical systems
- Translational, a library of 1-dimensional translational mechanical systems
- MultiBody, a library of 3-dimensional mechanical systems

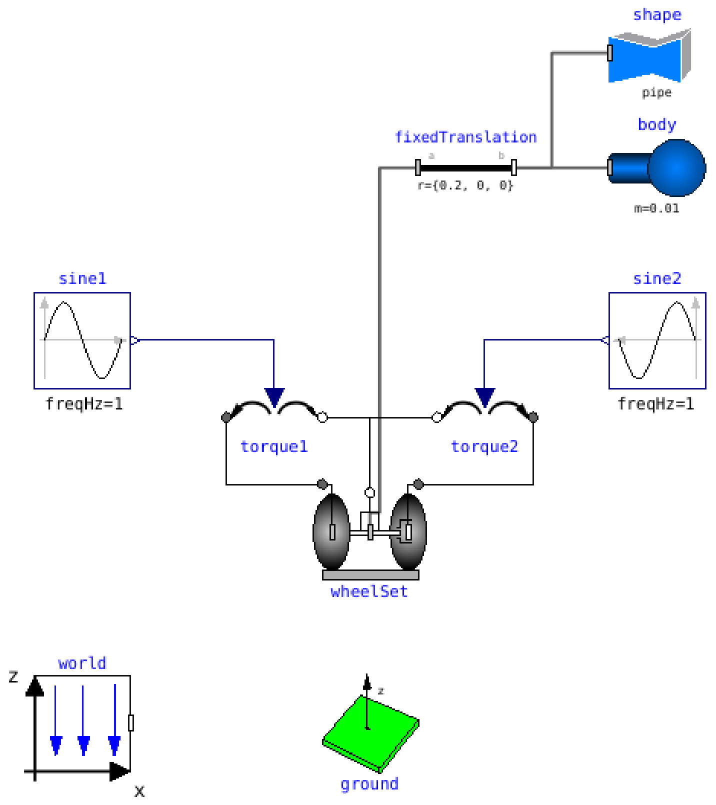
Consider, for example, a simple pendulum with two revolute joints

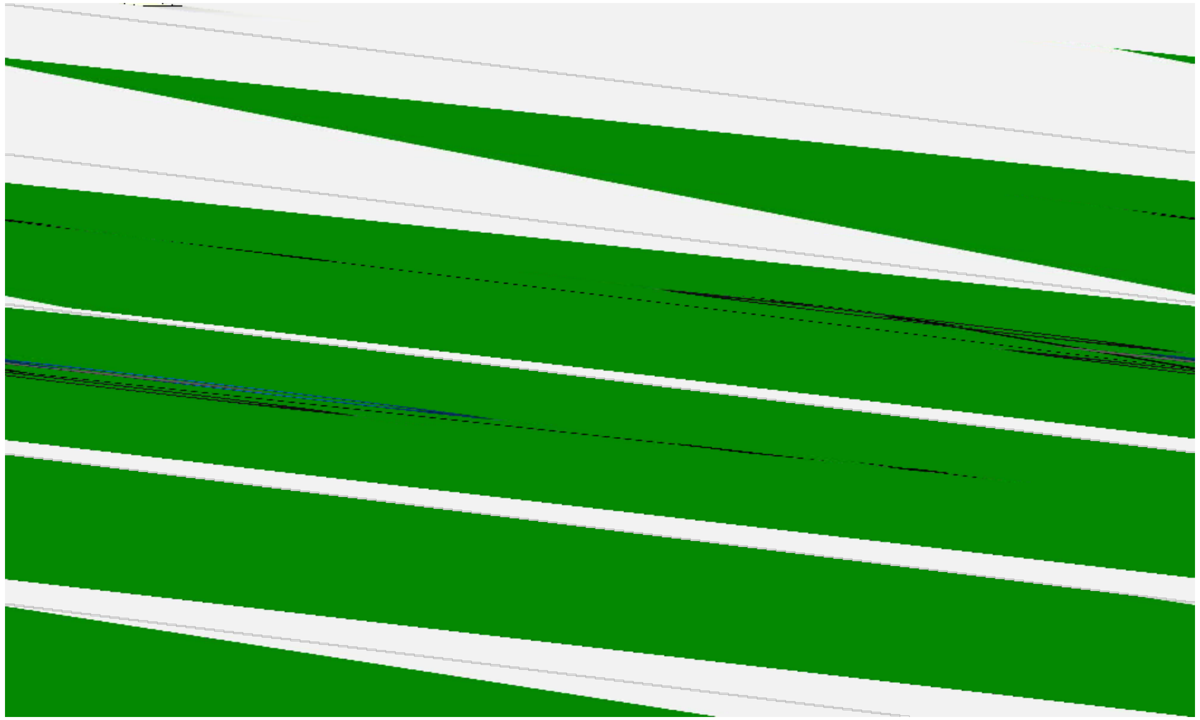




Multi-body model of a simple mobile robot

An example of a rolling wheel set driven by two torques at the wheels



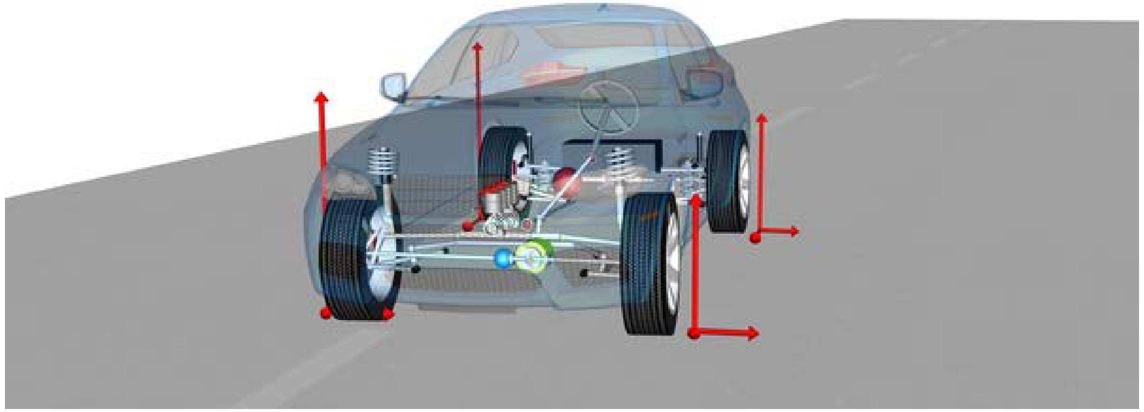


Vehicle Dynamics Library

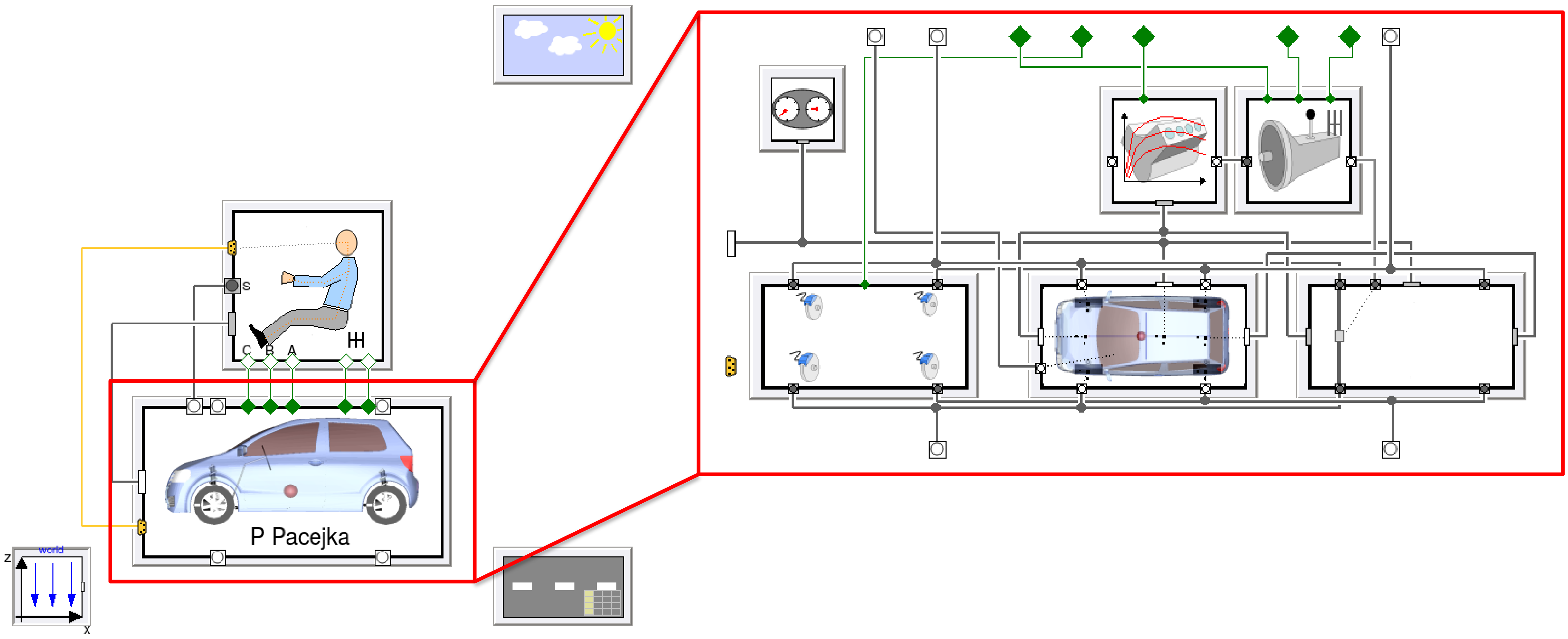
A mobile robot can be modelled starting from scratch or exploiting existing blocks / libraries.

The Vehicle Dynamics Library is a commercial library developed by Modelon that provides an open and user-extensible environment for full vehicle and vehicle subsystem simulation.

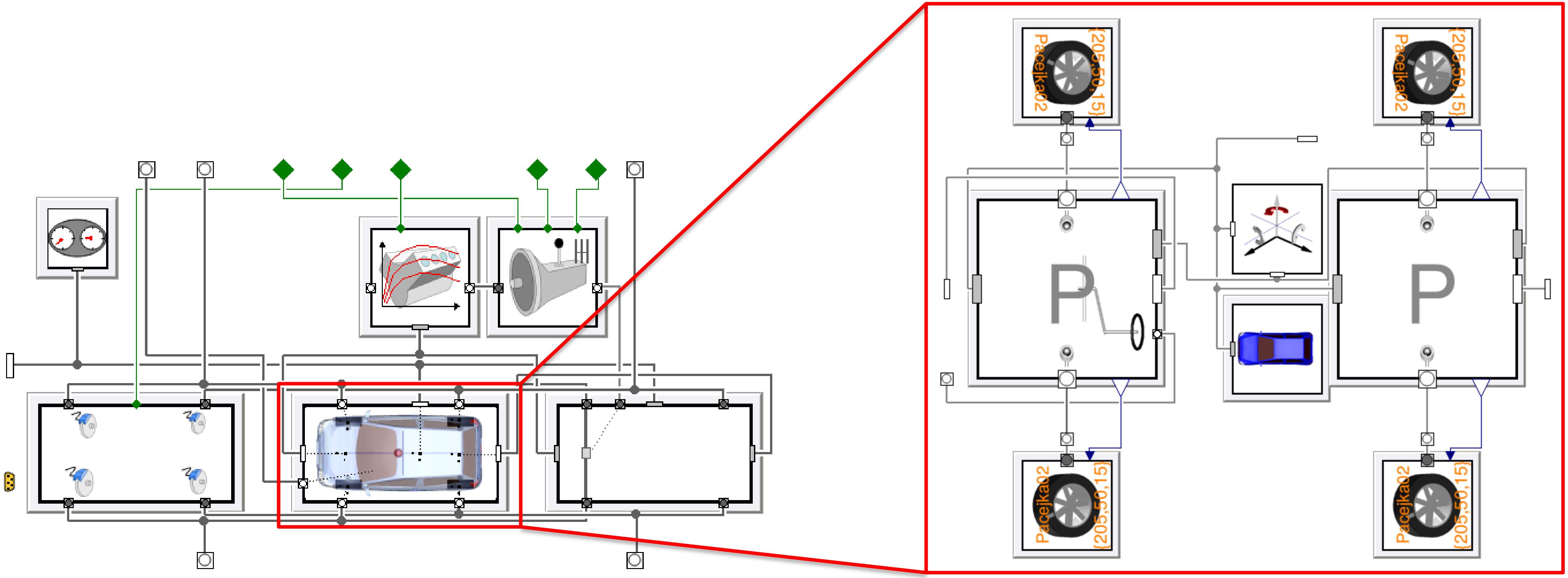
A webinar describing the library is available on YouTube.



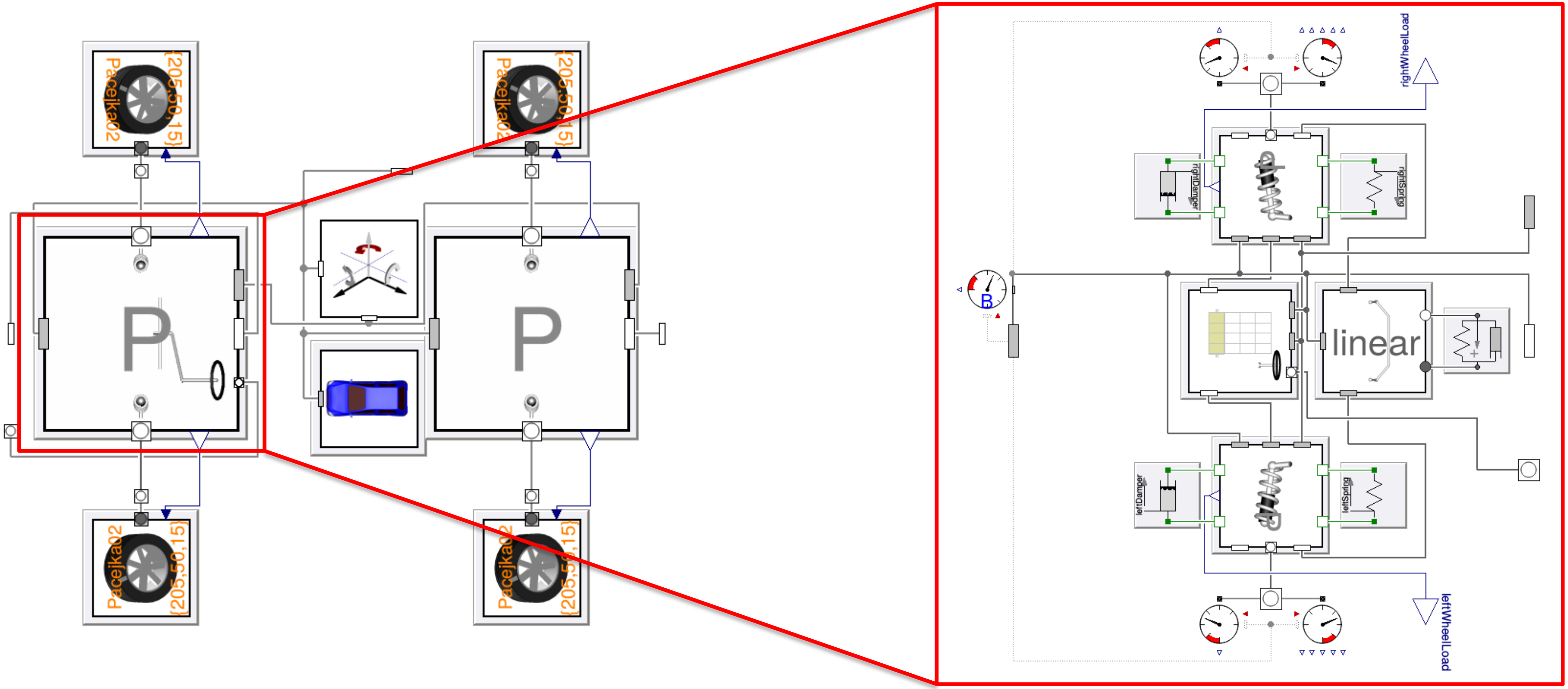
Vehicle Dynamics Library: the double lane change example



Vehicle Dynamics Library: the double lane change example



Vehicle Dynamics Library: the double lane change example



Vehicle Dynamics Library: the double lane change example

A navigation system is composed of different functionalities:

- localization and mapping
- environment perception
- path planning and following

Models we have considered so far, however, do not represent sensing for environment perception.

Modelling a lidar involves:

- developing a model of the sensor and the sensing process
- developing a geometric model of the environment

Simulink and Modelica are not particularly suitable to develop these kind of models, instead…

Other simulators, like Gazebo and v-rep, represent the right choice to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments.

Exploiting the same techniques and physics engines used in video games they are able to efficiently simulate the environment in a very realistic way (that does not mean physically consistent!).

This is exactly what is needed to test perception and perception-based algorithms!

These simulator are also well integrated in ROS and can be thus used for software testing.

Simulating a navigation system: Gazebo and v-rep

Simulating a navigation system: Gazebo and v-rep

Simulating a navigation system: conclusions

Summarizing:

- Matlab / Simulink represent the best solution to design and simulate the control algorithms
- Modelica represents the best solution to simulate in a physically consistent way vehicle dynamics, and thus for control system testing
- Gazebo / v-rep represent the best solution for perception algorithm design and validation, and for software testing

The only solution is the integration of different software environments!