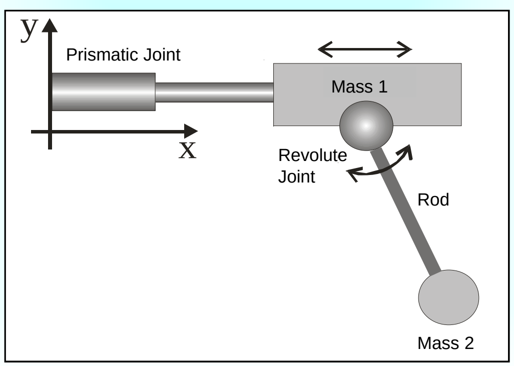
Planar Mechanics

- We shall now look at a first application of multi-bond graphs: ***planar mechanics***.
- We shall notice that a mechanical model composed of multi-bond graphs grows quickly in size and becomes poorly readable.
- For this reason, it is important to wrap multi-bond graph models of mechanical components in a framework that is more suitable to a modular description of mechanical systems.

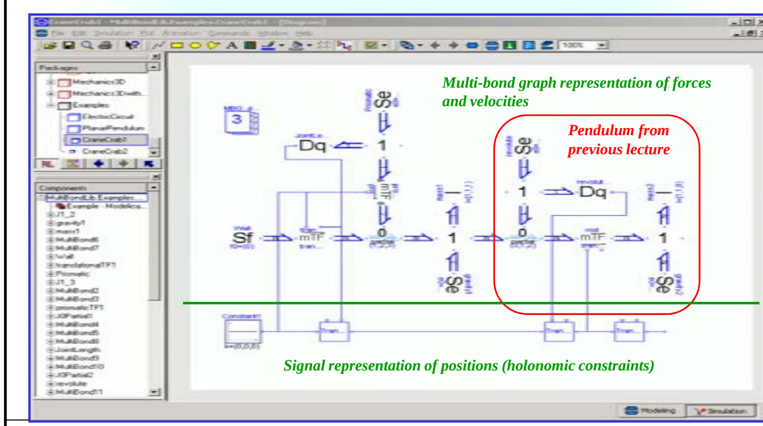
Table of Contents

A Crane Crab

- Let us start by modeling the following crane crab:

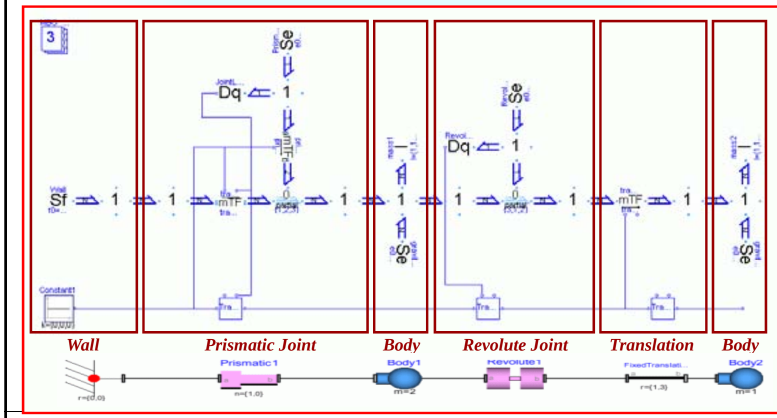


A Crane Crab II



ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

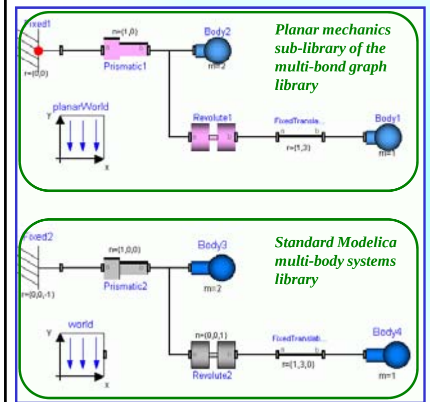
A Crane Crab III



ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

Mathematical Modeling of Physical Systems

A Crane Crab IV



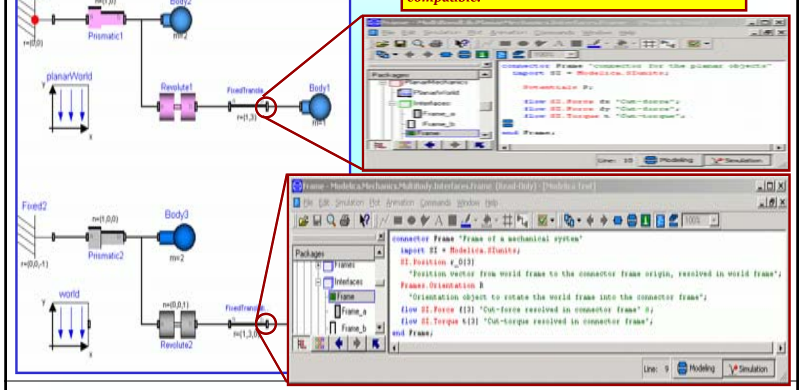
- The standard Modelica multi-body systems library is a general-purpose 3D mechanics library. No separate support for planar mechanics is currently being offered.
- The multi-bond graph library contains separate sub-libraries for planar mechanics and 3D mechanics, as well as for modeling hard collisions between mechanical bodies and for modeling gravitational pools (celestial mechanics).

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

Mathematical Modeling of Physical Systems

Mechanical Connectors (Frames)

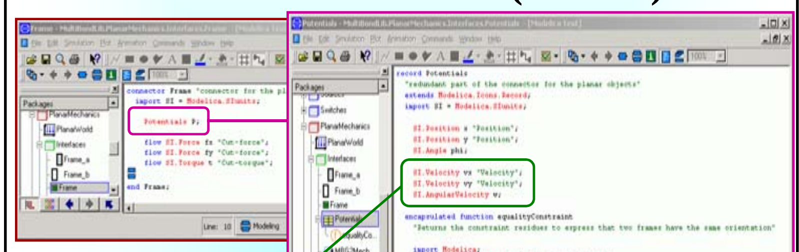
Although the connectors look the same, they are not compatible.



ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

Mathematical Modeling of Physical Systems

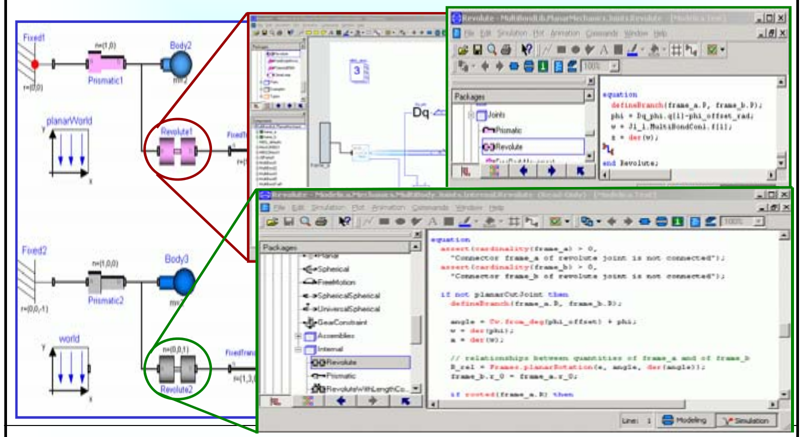
Mechanical Connectors (Frames) II



Redundant connections needed because of the bond graph approach.

The component models of the standard multi-body systems library and the wrapped multi-bond graph library cannot be mixed.

Revolute Joints



Revolute Joints II

- Using the multi-bond graph library, almost the entire model of the revolute joint has been coded graphically. There are only very few equations to be coded in the equation window. (There is still quite a bit of code there, because the object is being animated, and ***Dymola*** doesn't offer graphical support yet for coding animation models.)
- Using the multi-body library of the standard ***Modelica*** library, the entire revolute joint had to be coded by means of equations, leading to a fairly large equation model that is difficult to understand and even harder to maintain.

Rationale for Multi-bond Graphs

- It is important to keep the distance between the lowermost graphical layer and the equation layer small, such that as few equations as possible need to be maintained in alphanumerical form.
- ***Bond graphs and multi-bond graphs provide the most primitive graphical interface that is still fully object-oriented.*** Hence, when using bond graphs, the distance between the lowermost graphical layer (the bond graph layer) and the equation layer is minimized.

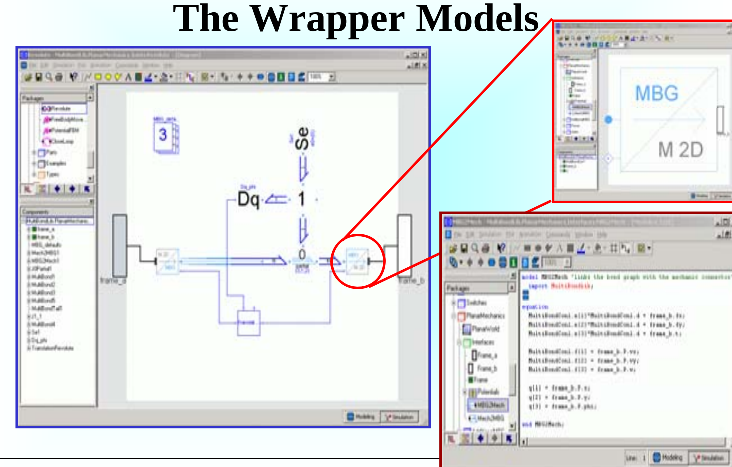
Rationale for Multi-bond Graphs II

- However, this does not imply that bond graphs offer an optimal user interface. For mechanical systems, this is certainly not the case.
- ***Wrapping bond graphs*** enables the modeler to map *any* graphical object-oriented modeling paradigm onto a lower-level bond graph layer that simplifies the maintenance of the resulting application libraries.

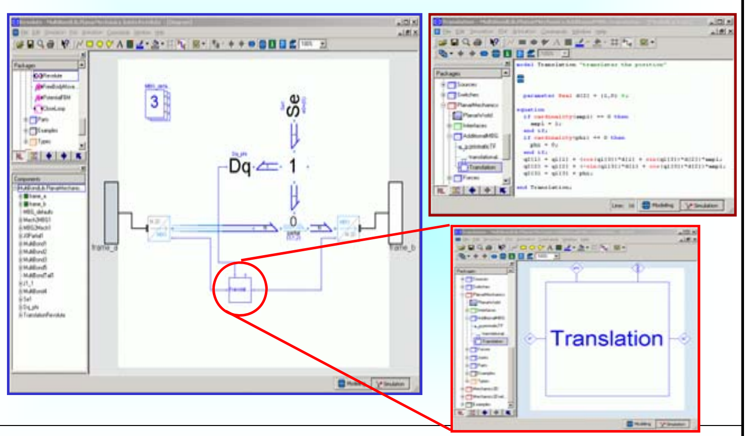
Animation

- In ***Dymola***, mechanical models can be ***automatically animated***. The end user of the models doesn't need to be concerned about this facet of modeling.
- However, individual bonds cannot be animated. The animation must take place at a higher conceptual level, namely that of multi-body system components, such as masses and joints.
- For this reason, a wrapping of multi-bond graphs is ***necessary*** if the resulting models are to be animated.

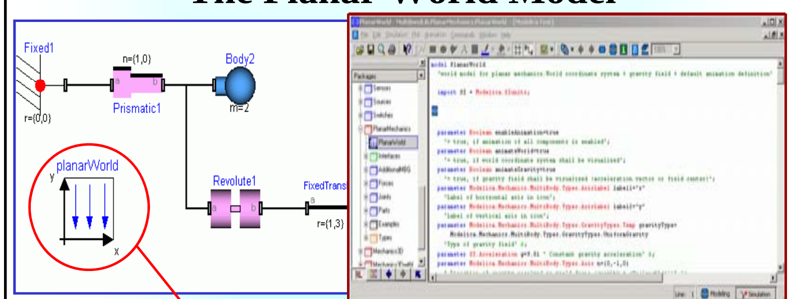
The Wrapper Models



The Position Translation Model



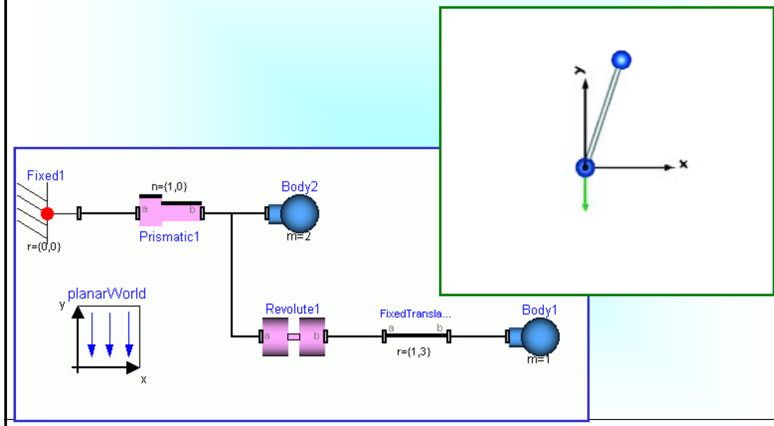
The Planar World Model



Every planar mechanical model must invoke the "planarWorld" model.

The primary purpose of the world model is to set up the animation.

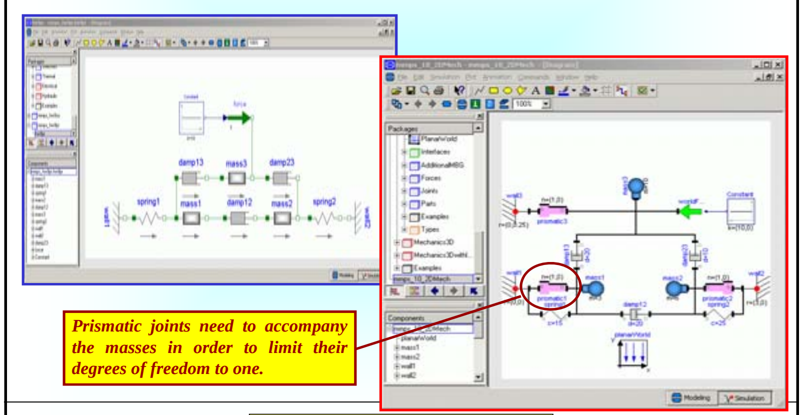
Crane Crab Simulation Results



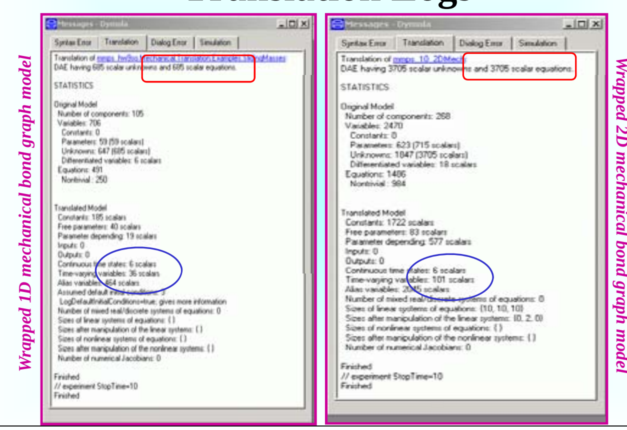
2D Simulation of 1D Models

- It is of course always possible to make use of the planar library also for the simulation of 1D models.
- Let us investigate, what the overhead of such an approach would be.
- To this end, we shall simulate the sliding mass model now using the planar mechanics library.

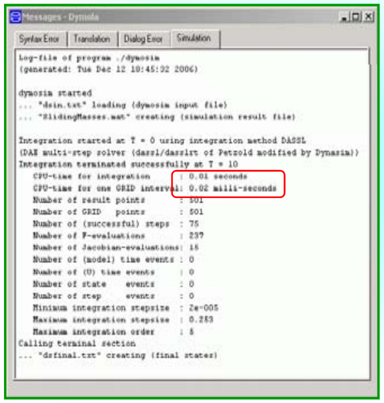
2D Simulation of 1D Models II



Translation Logs



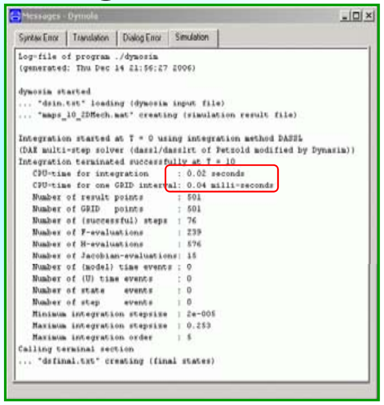
Simulation Logs



```
Log-file of program ./dymosim
(generated: Tue Dec 12 10:45:32 2006)

dymosim started
... "dsin.txt" loading (dymosim input file)
... "SlidingMasses.mat" creating (simulation result file)

Integration started at T = 0 using integration method DASSL
(DAE multi-step solver (dassl/dasslrt of Petzold modified by Dynasim))
Integration terminated successfully at T = 10
   CPU-time for integration      : 0.01 seconds
   CPU-time for one GRID interval: 0.02 milli-seconds
   Number of result points       : 501
   Number of GRID   points       : 501
   Number of (successful) steps  : 75
   Number of F-evaluations       : 237
   Number of Jacobian-evaluations: 15
   Number of (model) time events : 0
   Number of (U) time events     : 0
   Number of state    events     : 0
   Number of step     events     : 0
   Minimum integration stepsize  : 2e-005
   Maximum integration stepsize  : 0.253
   Maximum integration order     : 5
Calling terminal section
... "dsfinal.txt" creating (final states)
```

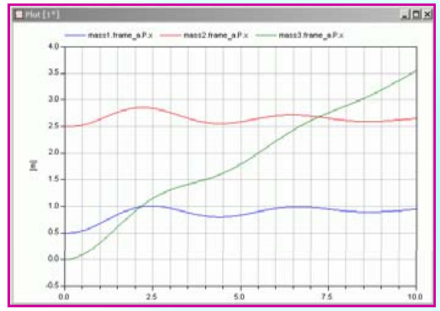


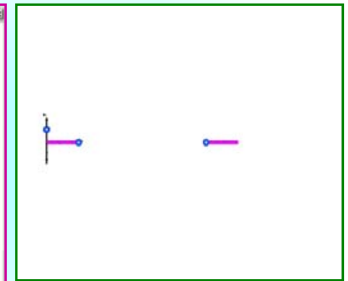
```
Log-file of program ./dymosim
(generated: Thu Dec 14 21:56:27 2006)

dymosim started
... "dsin.txt" loading (dymosim input file)
... "maps_10_2DMech.mat" creating (simulation result file)

Integration started at T = 0 using integration method DASSL
(DAE multi-step solver (dassl/dasslrt of Petzold modified by Dynasim))
Integration terminated successfully at T = 10
   CPU-time for integration      : 0.02 seconds
   CPU-time for one GRID interval: 0.04 milli-seconds
   Number of result points       : 501
   Number of GRID   points       : 501
   Number of (successful) steps  : 76
   Number of F-evaluations       : 239
   Number of H-evaluations       : 576
   Number of Jacobian-evaluations: 15
   Number of (model) time events : 0
   Number of (U) time events     : 0
   Number of state    events     : 0
   Number of step     events     : 0
   Minimum integration stepsize  : 2e-005
   Maximum integration stepsize  : 0.253
   Maximum integration order     : 5
Calling terminal section
... "dsfinal.txt" creating (final states)
```


Simulation Results





References I

- Zimmer, D. (2006), *A Modelica Library for MultiBond Graphs and its Application in 3D-Mechanics*, MS Thesis, Dept. of Computer Science, ETH Zurich.
- Zimmer, D. and F.E. Cellier (2006), "The Modelica Multi-bond Graph Library," *Proc. 5th Intl. Modelica Conference*, Vienna, Austria, Vol.2, pp. 559-568.

References II

- Cellier, F.E. and D. Zimmer (2006), "Wrapping Multi-bond Graphs: A Structured Approach to Modeling Complex Multi-body Dynamics," *Proc. 20th European Conference on Modeling and Simulation*, Bonn, Germany, pp. 7-13.