

Rotational3D — Efficient modelling of 3D effects in rotational mechanics

Johan Andreasson
Magnus Gäfvert
Modelon AB

Ideon Science Park
SE-223 70 Lund, Sweden

E-mail: {johan.andreasson,magnus.gafvert}@modelon.se

Abstract

In this work, it is described how effects of rotational mechanics represented in three dimensions can be described in an efficient way. The ideas have been implemented in the Rotational3D library which is introduced in this paper. Special attention is paid to the influence of joints and how 1D rotational and multi-body representations can be combined. Comparison of accuracy and efficiency with MultiBody is shown as well as typical application examples.

Keywords: Rotational mechanics, Rotational3D, Multi-Body, Driveline, Shaft

1 Introduction

There is large class of rotational mechanical systems where 1-dimensional revolution, or spin, is the dominating motion. Examples of such systems are automotive drivelines, transmissions, and steering mechanisms [1, 2]. These rotational systems are often part of general 3-dimensional multi-body systems with which they interact. This may be reaction forces and torques from brakes or motors, or gyroscopic moments resulting from motion orthogonal to the spin direction. It can also be kinematic effects on the spin motion from universal joints with bending angle coupled to 3-dimensional motion of multi-body parts.

Systems of this class may in parts be described with the Modelica Standard Library (MSL) [3] by using the the 1-dimensional formalism of the Rotational library, or by the full multi-body formalism of the MultiBody library [4]. The 1-dimensional approach gives a low-complexity representation that allows for fairly straightforward modeling of components such as clutches and gears. Simple reaction torques can be applied to multi-body parts by using the Mounting1D

component. Gyroscopic effect may be included by using the Rotor1D component instead of the normal rotational inertia. These composite Rotational and Multi-Body models in the Modelica Standard Library were introduced in [5]. They were designed for modeling of automatic transmissions, and are useful in contexts where the complete rotational mechanism is mounted on one multi-body part. In more general cases they have the limitations of not describing all interaction effects. Other drawbacks are the lack of information on mechanism geometry and the very rudimentary visualization.

Using MSL, a full model that includes all interaction effects requires that also the spin motion is described with MultiBody models. This results in overly complex and inefficient models where the 1-dimensional rotation is hidden in transformation matrices. The rotation angle is difficult and computationally expensive to extract. It is also only available in the range $[-\pi, \pi]$ and it is therefore tricky to track revolutions. Components such as shafts introduces a lot of constraint equations when defined with standard multi-body joints.

The Rotational3D library was designed to combine the advantages of the Rotational and MultiBody approaches. In particular, this means:

- Efficient description of the spin motion
- Reaction torques and forces
- Geometry and kinematic effects
- Visualization
- Interfacing to both MultiBody and Rotational

As a result, Rotational3D is very suitable for e.g. automotive applications and it is an established part of the VehicleDynamics Library [6].

2 Modelling principles

The main idea is to describe the spin motion similarly as in the Rotational library, and provide references that are coupled to MultiBody frames.

2.1 Connector definition

To represent the reference and the shaft motion, the following information must be communicated in a connector.

1. The rotation angle around the axis of rotation
2. The torque around the axis of rotation
3. The forces and torques acting on the reference frame
4. A reference frame with position and orientation
5. The direction of the axis of rotation, n
6. The axis that defines zero rotation, q

Items 1 and 2 can be described by a Rotational flange and items 3 and 4 by a MultiBody frame. In addition, the reference axes n and q are required. The n and q axes were explicitly defined in the connector in early versions of Rotational3D. Now they are instead implicitly defined as the x and y axes of the reference frame, resulting in the following connector definition¹:

```
connector Flange
  MB.Interfaces.Frame frame
  "Reference frame";
  Rt.Interfaces.Flange flange
  "Rotation around frame x-axis relative y-axis";
end Flange;
```

In Figure 1, the representation of the connector is seen.

2.2 Coupling 1D and 3D effects

Consider the fundamentals of an inertia, the angular acceleration of the inertia is a sum of the contributions of the flange and the frame so that the resulting torque at the flange, τ_{flange} is defined² as

$$\tau_{flange} = J(\dot{\omega}_{flange} + \hat{n} \cdot \dot{\bar{\omega}}_{frame}) \quad (1)$$

¹A similar connector definition is also used in [7] and has been introduced in the Modelica Standard Library as "FlangeWithBearing". The scope of this is different as it essentially is intended as a Rotational connector with an optional "bearingFrame" that allow for reaction torques to be applied.

²This can also be found in [5].

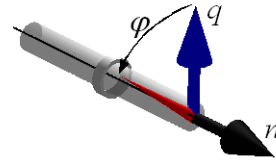


Figure 1: Animation view of an axle in a bearing with a connector reference shown. Axis-of-rotation (n), reference rotation (q), actual rotation (red arrow) and the angle φ .

and the torque at the frame, $\bar{\tau}_{frame}$ is defined by

$$\bar{\tau}_{frame} + \hat{n}\tau_{flange} = J(\hat{n}\dot{\omega}_{flange} + \bar{\omega}_{frame} \times \hat{n}\omega_{flange}). \quad (2)$$

Here ω_{flange} , $\bar{\omega}_{frame}$, \hat{n} , and J refer to the angular velocity of flange and frame, spin axis, and inertia, respectively. As a result, the torque at the flange, depends on the motion of the frame and vice versa.

Another example of how information from both connectors are required to formulate a model is a tyre model. Typically, the longitudinal force generated by the tyre is dependent on the wheel's spin velocity. Unlike for example a wheel speed sensor that measures the speed relative to the hub, the total rotational velocity around the spin axis, ω_{tot} , is required.

$$\omega_{tot} = \omega_{flange} + \hat{n} \cdot \bar{\omega}_{frame} \quad (3)$$

2.3 Avoiding over- and under-determined systems

Consider Figure 2: With two mounted bearings (1), each bearing is mounted to a MultiBody frame and by supplying parameters for n and q , the rotation axis and reference is defined in the Rotational3D connector. This model is over-determined since two connectors, both specifying the reference is connected. Correspondingly, if none of the connectors in a set would have the reference specified, the model would be under-determined (2).

This is avoided by a connection rule using two connectors with different colours, the grey is unconstrained and the white is constrained: *Each connection set must have one and only one white connector and there must never be a loop containing only white connectors.*

This rule is somewhat simplified, consider for example the under-determined example from Figure 2 (3). Here, both ends of the shaft are attached to white connectors and there is no loop consisting on only white connectors. Still, the model is over-constrained

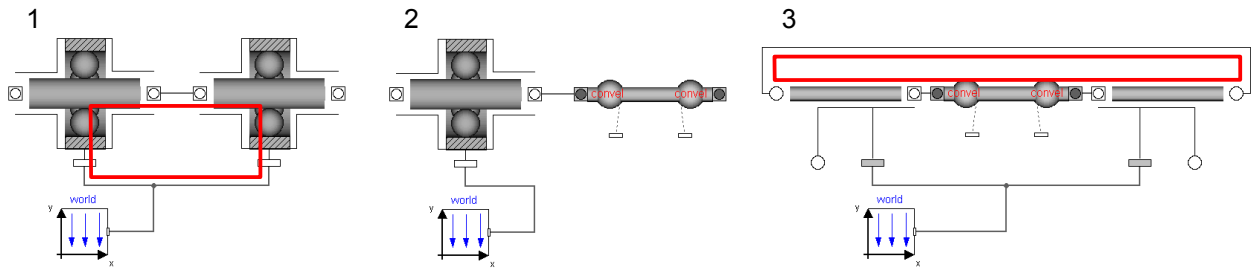


Figure 2: Example of under- and over-determined systems: Loop consisting on only white connectors (1). The right-hand flange of the shaft is un-connected and under-determined (2). A model that is over-constrained through the Rotational part (3).

since the Rotational part is forming a loop. This of course could be handled by the rule above. It would require that there were four Rotational3D connectors to cover the combinations of constrained and unconstrained connectors.

Still, however, this would not be sufficient for the general case since there are cases where some of the MultiBody variables are constrained and some not, which ultimately would lead to an unreasonable amount of connectors.

3 Library contents

Unlike for many other libraries, Rotational3D components are not intended to be used entirely on its own, but together with Rotational and MultiBody components. Only parts that cannot be modelled in Rotational are implemented. This includes shafts, inertias, visualizers, and other models where either the geometry or inertial effects are considered. In addition, there are models that allow both Rotational and MultiBody models to be used with the library, especially mounts and housings.

Figure 3 shows some highlighted components: The MultiBodyMount (1) translates a Rotational3D connector to a rotating MultiBody frame and the RotationalMount (2) allow Rotational models with action and reaction torques to be connected. The SupportedRotationalMount (3) is a version of (2) including a bearing and thus has an additional MultiBody connector to define the orientation of the flange. The Flange visualization (4) visualizes the vectors n and q as well as the rotation angle ϕ as seen in Figure 1 which is useful both for debugging and comprehension. The Shaft component (5) defines the motion between two flanges without reference. It imposes no constraints between

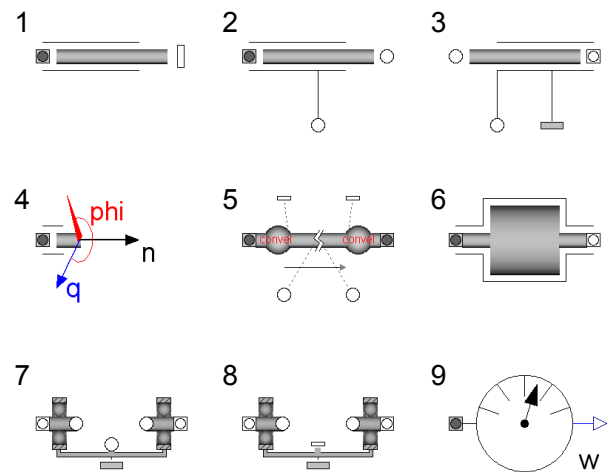


Figure 3: A selection of components from Rotational3D.

the flanges since it is assumed to be flexible also in the length direction. The twist characteristics is defined by connecting components to the Rotational connectors and additionally, inertial effects can be added via the MultiBody connectors. The rotation angle is defined based on the choice of joint type made, either universal and constant velocity. The Inertia (6) corresponds to the Rotor component but additionally contains geometric information as well as mass properties and the corresponding visualization.

As an extension to the basic mounts (1-3), there are also housings suitable for components such as gear boxes. The SupportedHousing (7) allow for ideal gears to be directly connected to the Rotational flanges. The SupportedHousing2 (8) is an adaption to handle gearboxes from e.g. the PowerTrain library [8] requiring a MultiBody support frame for internal in-

erties and animation. There are also sensors, such as the AbsoluteRotationalVelocity (9), described in Section 2.2.

4 Examples

As explained earlier, Rotational3D is intended to be used together with both Rotational and MultiBody components. This section explains and exemplifies when Rotational3D is suitable and how component and subsystem models can be designed.

4.1 Bevel gear

The first example shows two bevel gear components. Consider first the upper diagram layer in Figure 4, showing a bevel gear with an external support frame (1). To this frame, two SupportedRotationalMount components (2) are connected, each containing a bearing that specifies the orientations of the two axes. These mounts are connected to a Rotational.GearR2R component (3) that contains the 1D gear characteristics. Each mount is also connected to a connector (4) and visualization for gear wheel and axle (5). Both connectors are white since their orientation is fully specified and as seen in the animation view (6), there are two bearings except for the axles and gear wheels.

Now consider the lower diagram, showing the same bevel gear but without the support frame. Instead, the left connector is grey, indicating that this component has to be connected to an outside bearing. The left mount is here a RotationalMount, and as a consequence, no bearing is present in the visualization.

4.2 Steering system

Steering systems for passenger cars are typically designed as illustrated in Figure 5. The steering wheel is attached to the steering column (1) which in turn is connected to a shaft that is connected to the pinion of the rack-and-pinion mechanism (3). Each connection requires a joint (4) with two degrees of freedom to allow the shaft to rotate around its length axis. The universal joint is the most common type, consisting of two revolute joints in series forming a cross. Unlike more advanced joints, the universal joint has a varying ratio over a revolution, depending on the bend angle (α).

This requires careful design of the geometric layout as well as the rotation of the cross and depending on these two factors, the resulting gear ratio of the mechanism will vary. Figure 6 shows the difference in rack

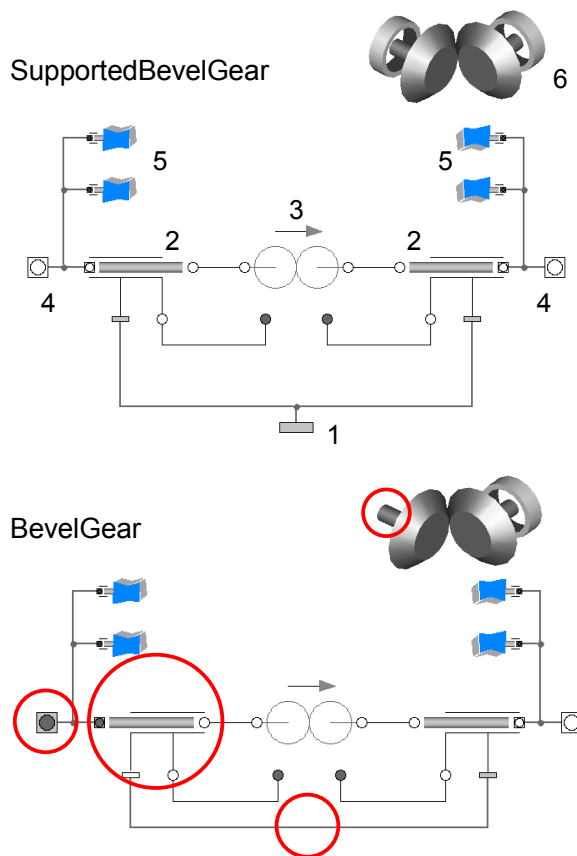


Figure 4: Diagram layer and visualization of two bevel gear models.

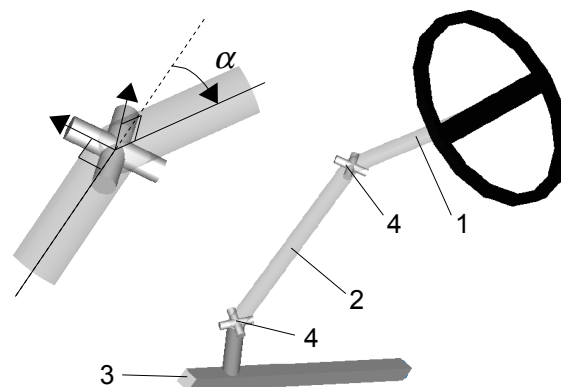


Figure 5: Typical steering system layout (right) and universal joint geometry (left).

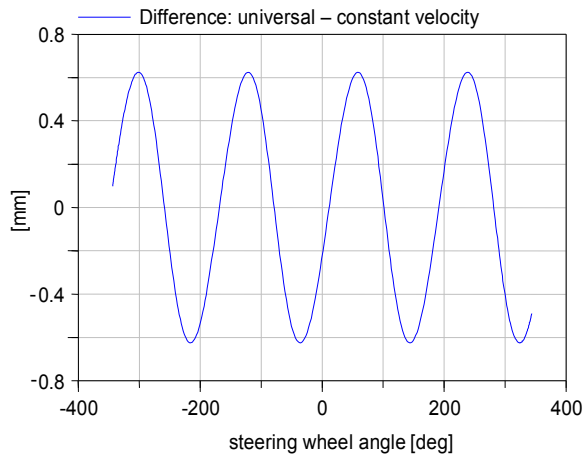


Figure 6: Difference in rack position as function of steering wheel angle.

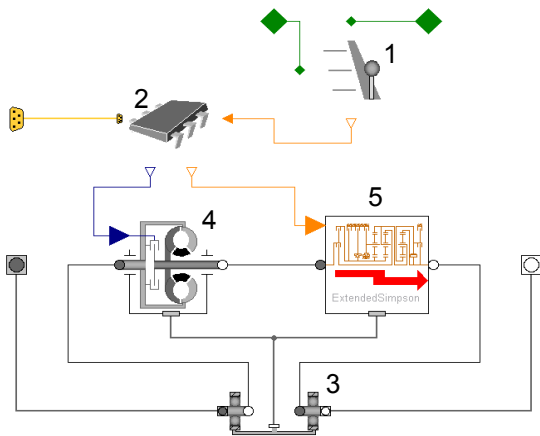


Figure 7: Transmission model for the VehicleDynamics library using PowerTrain library components.

position for the steering system in Figure 5, compared to the same system but with constant velocity joints.

4.3 Automatic Transmission

Another advantage of Rotational3D is its compatibility with other Modelica libraries. Figure 7 shows an example of a seamless integration of two libraries for an automatic transmission. It is defined from a standard interface in the VehicleDynamics library and completed with shift mechanism (1), shift controller (2) and a housing (3). The torque converter (4) and the gear box (5) are from the PowerTrain library [8]. Note that the PowerTrain components use a MultiBody support frame as discussed in Section 3.

Table 1: Translation and simulation results from Dymola [9].

	MB	Rt3D
linear systems	{10,3,3,2}	{3,3}
non-linear systems	{10}	{1}
simulation time	1.91	0.047

4.4 Comparison with MultiBody

Two implementations of a drive shaft are compared. A drive shaft is typically used to transfer axial rotational motion between two moving bodies. Using MultiBody primitives to define a non-elastic shaft with universal joints yields the series R-U-P-U-R. R denotes a revolute joint, here representing the bearings, P is a prismatic and U is a universal (two orthogonal revolute joints).

Figure 9, shows an animation screen shot of the performed simulation where both the MultiBody (MB) and the Rotational3D (Rt3D) representations are included on top of each other. The trajectory of the first universal joint is shown as well as vectors for the resulting forces and torques. The first shaft end is driven with a constant speed relative to its bearing while the required drive torque as well as the speed at the other shaft end is shown, Figure 9.

Table 1 shows the number of linear and non-linear systems of equations and the simulation time for the two cases described above. By avoiding the constraints imposed by the MultiBody primitives, the number of non-linear equations are drastically decreased which makes the Rotational3D implementation about a factor 40 faster and less sensitive to the specified accuracy of the integrator.

5 Limitations

Although the library concept is proven to be efficient, it requires that some fundamental rules are followed. As already stated, there is a connection rule to avoid over- and under-constrained models. Additional, relating to the problems with the representation of multiple revolution in MultiBody, it is required that the relative rotation between two reference frames is less than one revolution.

6 Conclusions

This work presents a new library that combines the advantage with Rotational and MultiBody representation

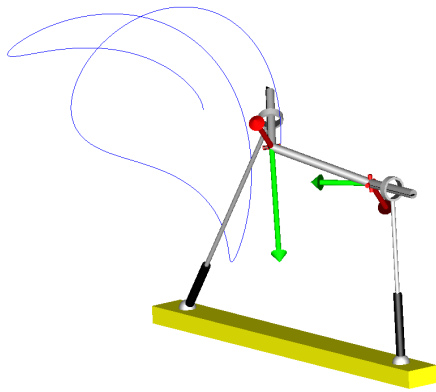


Figure 8: Animation view of the comparison and validation example with motion, forces and torques indicated. The MultiBody and the Rotational3D models are overlaid.

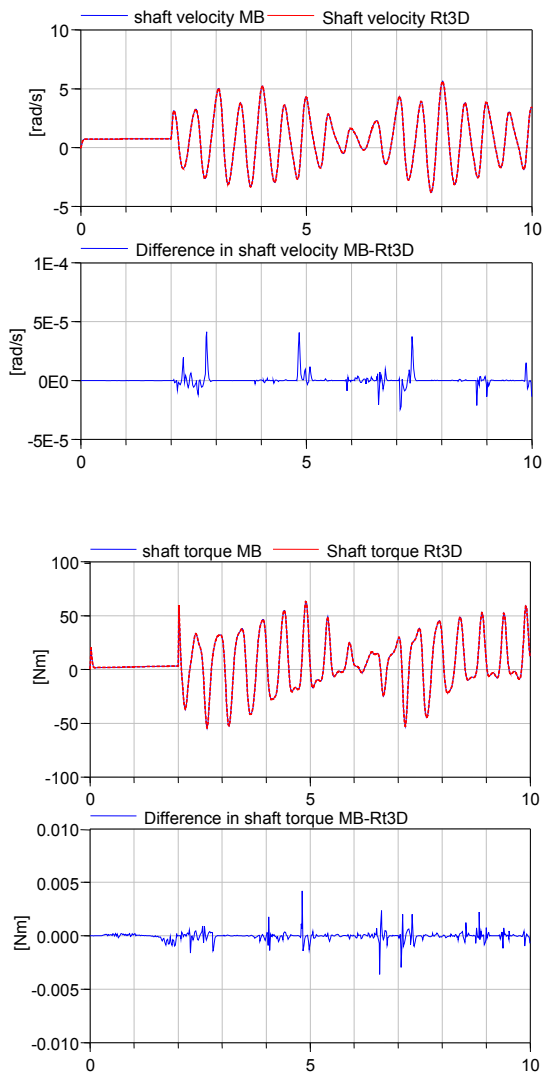


Figure 9: Comparison of the MultiBody and the Rotational3D implementations. Speed at the second shaft end (upper two plots) and required drive torque at first shaft end (lower two plots).

of rotational mechanics. It gives the same simulation results as MultiBody representations, often at a fraction of the cost. An example shows a factor of around 40. More complex mechanisms such as drivelines are often difficult/impossible to get to work in MultiBody. These are built seamlessly in Rotational3D.

References

- [1] Andreasson, J., Gäfvert, M., The VehicleDynamics Library - Overview and Applications. In: *Proceedings of the 5th Modelica Conference*, Vienna, Austria, Modelica Association, 4-5 September 2006.
- [2] Andreasson, J., On Generic Vehicle Motion Modelling and Control, Ph.D. thesis, ISBN 91-7178-527-2, 2006.
- [3] Modelica Association. The Modelica Standard Library, <http://www.modelica.org>
- [4] Otter, M. et al. The New Modelica MultiBody Library. In: *Proceedings of the 3rd Modelica Conference 2003*, Linköping, Sweden, Modelica Association, October 2003.
- [5] Schweiger, C., Otter, M., Modelling 3D Mechanical Effects of 1D Powertrains. In: *Proceedings of the 3rd Modelica Conference 2003*, Linköping, Sweden, Modelica Association, October 2003.
- [6] Modelon AB, Lund, Sweden. The VehicleDynamics library, User's Guide, Version 1.2, 2007.
- [7] Dempsey M. et al. Coordinated automotive libraries for vehicle system modelling. In: *Proceedings of the 5th Modelica Conference 2006*, Vienna, Austria, Modelica Association, September 2006.
- [8] Schweiger, C. et al. The PowerTrain Library: New Concepts and New Fields of Application. In: *Proceedings of the 4th Modelica Conference 2005*, Hamburg, Germany, Modelica Association, March 2005.
- [9] Dymola - Dynamic Modelica Laboratory, <http://www.dynasim.se>