Introduction of the 3D Geometrical Constraints in Modelica

Régis PLATEAUX Olivia PENAS Faïda MHENNI
Jean-Yves CHOLEY Alain RIVIERE
LISMMA – EA2336 (SUPMECA)
3, rue Fernand Hainaut - 93407 Saint-Ouen Cedex France

{firstname.lastname}@supmeca.fr

Abstract

Mechanical Modelica Libraries enable one to model instances of geometrical objects and their absolute positioning. But in order to facilitate the design by taking into account the size and the position of objects, we need relative positioning variables and parameters. In this paper, we propose to meet this challenge by adding in Modelica a set of 13 geometrical constraints and by implementing them.

Keywords: Modelica 3D, Geometrical Constraints, TTRS, Relative positioning

1 Introduction

In previous studies [1], to facilitate the design of a mechatronic product [2], we proposed to integrate the entire downward design cycle in order to achieve a modelling continuity. To do this, we proposed a hybrid design methodology based on several tools, languages and methodologies including Modelica [3], in Dymola [4] environment.

Modelica models integrate some geometrical elements such as position of gravity point, mass, volumes and inertia matrix elements. However, the current environments don't allow until now to improve entirely our method. Figure 1 presents the desired 3D Modelica framework enabling a simultaneous representation in the "modelling" design window of the 2D logical diagram and of the 3D geometric class and keeping the 3D view for the simulation

Actually they usually propose:

- 2D icons containing only geometrical parameters (absolute positions and dimensions) in the modelling interface,
- realistic 3D representations of Modelica objects in the simulation interface with the same geometrical parameters,

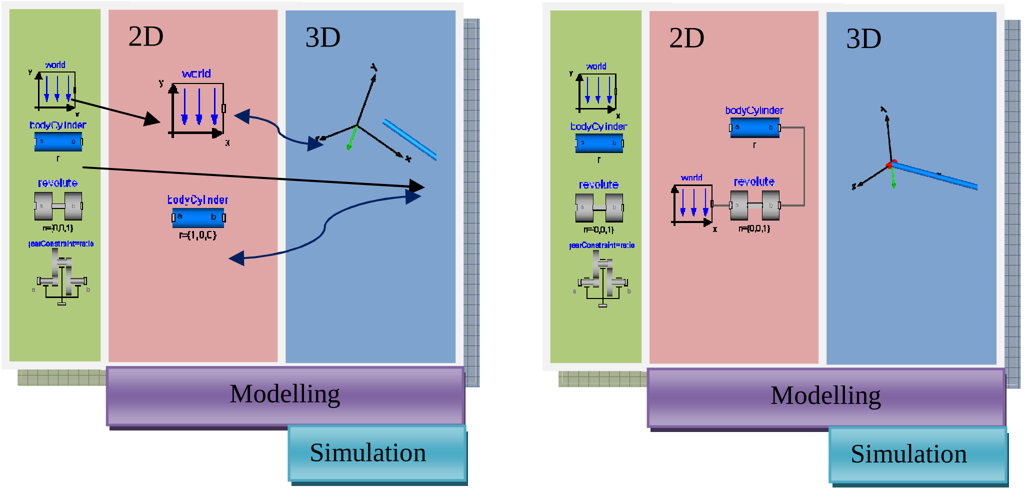

Figure 1. 3D Modelica Environment

But:

- no geometrical variables, and
- a framework without any dynamic link between both of these interfaces.

For example, the geometrical structure of the model in Figure 2 cannot easily be anticipated in 3D. We must wait for the simulation to know which of structures has been defined.

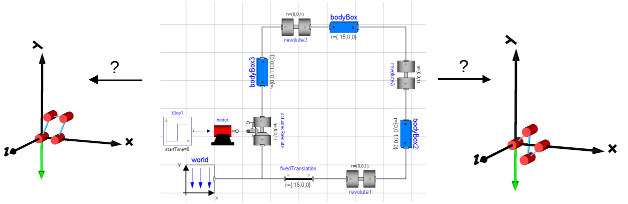

Figure 2. Two possible representations of the geometry of a model: parallelogram or twisted parallelogram.

That is why our aim is to transform geometrical parameters into geometrical variables in both interfaces and to propose for the design one new modelling interface with both views (2D and 3D). Thus today we tackle the first problem by modifying libraries of Modelica objects in order to switch geometrical instances in objects modelled by constraints. It will later help the transition from 2D to 3D modelling for the second step [5].

To do this, the first point is to introduce explicitly geometrical constraints as Modelica objects.

2 Towards an explicit relative positioning in Modelica

Considering the model already developed in our laboratory for the geometrical tolerancing [6], we implement geometric constraints with the « Topologically and Technologically Related Surfaces » (TTRS) theory [7].

2.1 TTRS Objects in Modelica

Any surface or association of real surfaces of an object is related to a kinematic invariance class named TTRS. There are 7 classes of TTRS classified according to increasing degrees of freedom (DOF) in Table 1.

Table 1 7 Classes of TTRS

Classes	Symbol	DOF
Identity	{E}	0
Revolute	$\{R_{D,P}\}$	1
Prismatic	$\{T_D\}$	1
Helical	$\{H_{D,P}\}$	1
Cylindrical	$\{C_D\}$	2
Spherical	$\{G_P\}$	3
Planar	$\{S_O\}$	3

Kinematic joints can be expressed by TTRS.

Each TTRS is characterised by a MRGE (Minimal Reference Geometric Element). Each MRGE is made up of a combination of one point, one line and/or one plan, but does not take into account the intrinsic dimensional aspect of the object.

Figure 3 shows the example of a cone, which is represented by the TTRS "Revolute Surface" whose MRGE representation is one point and one line.

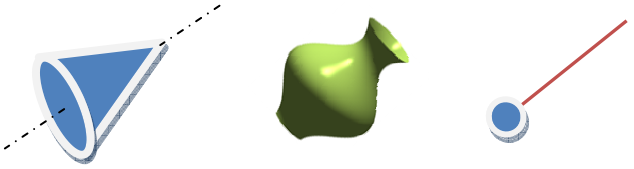

Figure 3. Cone representation: TTRS Revolute & MGRE point/line.

In order to assemble two geometrical objects, i.e. to define geometrical constraints between two TTRS, 44 associations are identified depending on the relative orientations and positions with regards to the other. In turn each association forms TTRS. They correspond to the most elementary formulation of a kinematic connection between objects.

Example: association of two Revolute TTRS

Given $\{R_{D1,P1}\}$ and $\{R_{D2,P2}\}$:

If $D1 \neq D2$, $\{R_{D1,P1}\} \cup \{R_{D2,P2}\} = \{E\}$

If $D1 = D2$, $\{R_{D1,P1}\} \cup \{R_{D2,P2}\} = \{R_{D1,P1}\}$

Finally the passage to its MRGE enables us to have only 13 possible cases of constraints (Table 3 p.5).

These constraints between MRGE numbered from C1 to C13 are expressed by means of algebraic expressions and parameters.

For example: the association of 2 beams by application of the C12 constraint (Figure 4) corresponds to:

Given (M_1, $\mathbf{u_1}$) and (M_2, $\mathbf{u_2}$) two sliding points and vectors which belong respectively to D1 and D2 lines, we have:

$$C12 = \left\{ \begin{array}{c} \mathbf{u_1} = \mathbf{u_2} \\ \mathbf{u_1}.\overrightarrow{\mathbf{M_1M_2}} = 0 \\ d = \left\|\overrightarrow{\mathbf{M_1M_2}}\right\| \end{array} \right\}$$

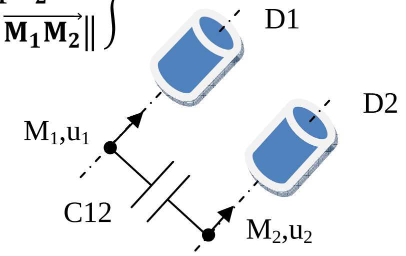

Figure 4. C12: line-line, parallel, distance.

2.2 MRGE Expression in Modelica/Dymola

The set of related MRGE may be implemented with Modelica: each MRGE is an object (point, line and plan).

In Modelica language, each object is associated to another by means of its topological connexion performed through its connectors. The elementary connector for the MRGE is the point. The implementation choice was to represent the MRGE line by its affine view using the previous elementary MRGE point.

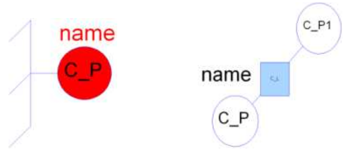

Figure 5. Modelica implementation of the MRGE Point connector and of the MRGE Line with 2 point connectors and 1 line connector.

2.3 The 13 Constraints

With MRGE, the 13 constraints are generated thanks to their algebraic expressions. The model is now designed by constraints: they are no longer instances of geometrical objects but their equivalent constraints (e.g. Figure 5) with real geometrical variables and parameters.

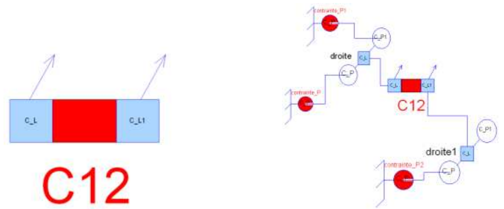

Figure 6. Example of the C12 Constraint in Modelica: two lines constrained parallel.

3 Academic Application

Models currently developed take into account only geometric variables. Further improvements will integrate mechanical ones.

One goal is to come closer to the design habit. *What you see is what you mean* is another one. So when we express perpendicularity, parallelism or any geometric constraint, our point of view is to declare it.

After a short description of the treated example, we will study the method used with current models. Then we will build it step-by-step with the new constraint objects.

3.1 Design Point of View

The studied mechanical system is an automatic rising barrier called “Sinusmatic”. The SINUSMATIC barrier is adaptable to the dimensions and speed to most applications. Its particularity results from patented kinematics for his bellcrank that transforms the continuous circular movement into an approximately sinusoidal 1/4 round one.

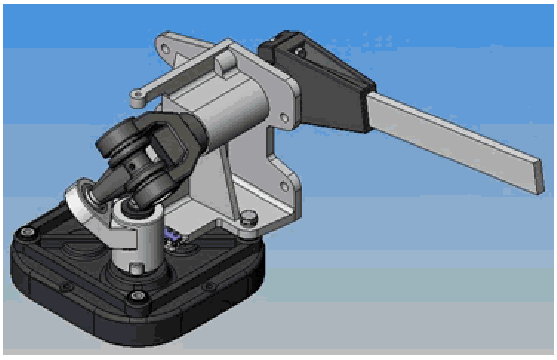

Figure 7 Currently Sinusmatic

Its mechanical structure is schematized in Figure 8.

It is composed of S0: frame, S1: plateau, S2: socket, S3: ball, S4: crosspiece and S5: fork axis.

Joints are expressed in Table 2.

Table 2 Sinusmatic Joints

	S1	S2	S3	S4	S5
S0	{$R_{D,P}$}				{$R_{D,P}$}
S1		{E}			
S2			{G_P}		
S3				{C_D}	
S4					{$R_{D,P}$}

Some design requirements are specified:

- Axis of the cylindrical joint go through the centre of the ball-and-socket (S3/S4),

- Centre D of S4/S5 joint is the intersection of S1/S0 et S5/S0 axes,
- Axes of (S3/S4) and (S4/S5) joints are converging.

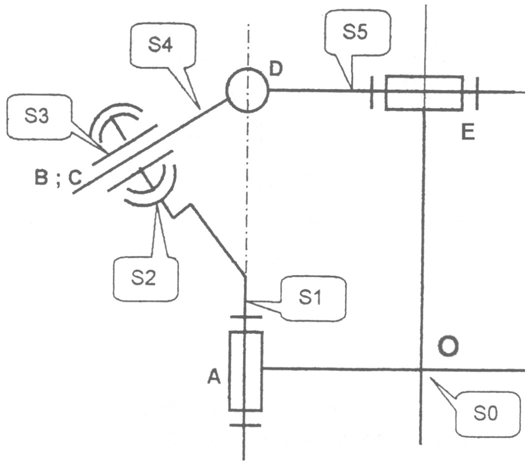

Figure 8 Kinematic Diagram

3.2 Current approach

We need to express the structure by means of vectors as Figure 9.

The direct impact is that we need all values or a set of consistent values.

Moreover the sense depends on the choice of the connected initial point.

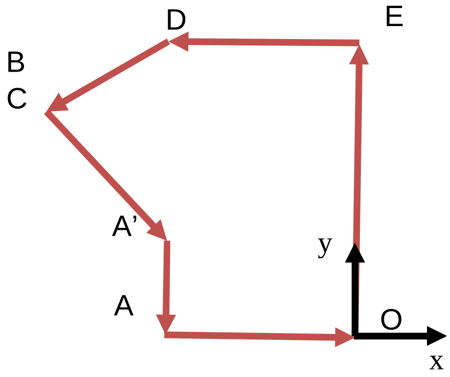

Figure 9 Vectorial Model

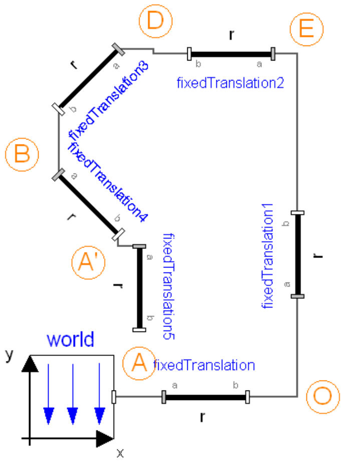

Figure 10 Modelling with Vectorial Approach

3.3 Method with constraint objects

With this approach we may draw (Figure 11 in Dymola framework) all characteristic nodes (green circles) with lines (blue rectangles).

The entire model is non-simulable. Thus we add constraints that express the design needs. Whatever expressed now is definite.

The geometric loop may be simulated to obtain the result in Figure 12.

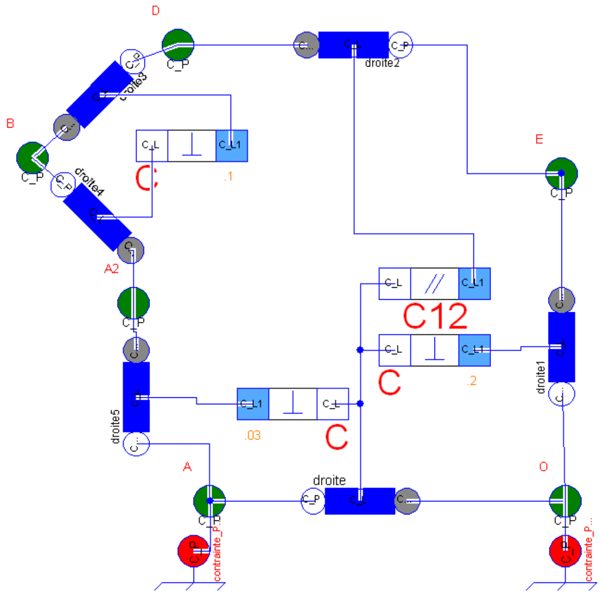

Figure 11 Geometric Constraint Modelling

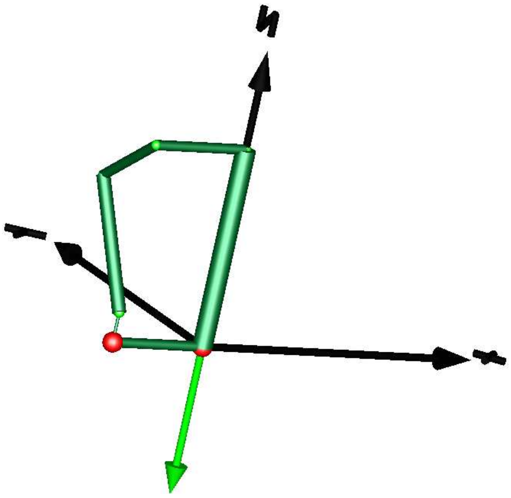

Figure 12 Result of Constraint Modelling

4 Conclusions

We use the interface properties with 3D environment of Modelica/Dymola framework to add design functionalities for geometrical integration.

This modelling by constraints in Modelica is not only used to transform geometrical instances to geometrical variables but also to integrate multi-physics and geometry variables, relying on the requirements data.

It will be developed more deeply in further papers.

References

[1]. **Plateaux, Régis, et al.** *Towards an Integrated Mechatronic Design Process.* Málaga : IEEE, 2009. ISBN 978-1-4244-4195-2.

[2]. **Ferretti G., Magnani G., Rocco P.** Virtual prototyping of mechatronic systems. 2004, Vol. 28, (2), pp. 193–206. doi:10.1016/j.arcontrol.2004.02.002.

[3]. **Modelica Association.** Modeling of Complex Physical Systems. *Modelica.* [Online] http://www.modelica.org/.

[4]. Dymola – DYnamic MOdeling LAboratory with Modelica (Dynasim AB). [Online] http://www.dynasim.com.

[5]. **Hadj-Amor, H.J.** *Contribution au prototypage virtuel de systèmes mécatroniques basé sur une architecture distribuée HLA - Expérimentation sous les environnements OpenModelica-OpenMASK.* Toulon : Supmeca, 2008. LISMMA (EA 2336) thesis.

[6]. **Clément, André, Rivière, Alain and Temmerman, Michel.** *Cotation tridimensionnelle des systèmes mécaniques.* Paris : PYC Edition, 1994.

[7]. **Clément A., Rivière A., Serré P., Valade C.** *The TTRS: 13 Constraint for Dimensionning and Tolerancing ", 5th CIRP Seminar on Computer Aided Tolerancing,.* The University of Toronto, Canada : s.n., April 27-29, 1997.

[8]. **Fritzson, P.** Introduction to Object-Oriented Modeling and Simulation with OpenModelica. *IDA - The Department of Computer and Information Science.* [Online] [Cited: 03 01, 2009.] http://www.ida.liu.se/~pelab/modelica/OpenModelica/Documents/ModelicaTutorialFritzson.pdf.

[9]. **Plateaux, R., et al.** *Méthodologie intégrée de conception d'un produit mécatronique.* Marseille : CFM, 2009. (in reviewing).

Table 3 The 13 constraints

	●	╱	▱
●	$C1 : O_1 = O_2 \rightarrow \{S_{O_1}\}$	$C4 : O_1 \in D_2 \rightarrow \{R_{D_2}\}$	$C3 : \{R_D\}$
	$C2 : O_1 \neq O_2 \rightarrow \{R_{O_1 O_2}\}$	$C5 : O_1 \notin D_2 \rightarrow \{E\}$	
╱		$C11: D_1 = D_2 \rightarrow \{C_{D_1}\}$	$C8: D_1 \perp P_2 \rightarrow \{R_{D_1}\}$
		$C12: \begin{Bmatrix} D_1 \parallel D_2 \\ D_1 \neq D_2 \end{Bmatrix} \rightarrow \{T_{D_1}\}$	$C9: D_1 \parallel P_2 \rightarrow \{T_{D_1}\}$
		$C13: \begin{Bmatrix} D_1 \nparallel D_2 \\ D_1 \neq D_2 \end{Bmatrix} \rightarrow \{E\}$	$C10: D_1 \angle P_2 \rightarrow \{E\}$
▱			$C6: P_1 \parallel P_2 \rightarrow \{G_{P_1}\}$
			$C7: P_1 \nparallel P_2 \rightarrow \{T_{P_1 \cap P_2}\}$