International Journal of Modern Manufacturing Technologies
ISSN 2067–3604, Special Issue, Vol. XIII, No. 3 / 2021
https://doi.org/10.54684/ijmmt.2021.13.3.87

GENERATING NUMERICAL MULTIBODY MODEL OF TOOTHED BELT IN TECHNOLOGICAL MACHINES WITH SIEMENS NX

Julian Malaka, Mariusz Hetmańczyk

Silesian University of Technology, Faculty of Mechanical Engineering
ul. Konarskiego 18A, 44-100 Gliwice, Poland

Corresponding author: Julian Malaka, julian.malaka@polsl.pl

Abstract: So far, in computer-aided modelling programs, there have not been implemented tools for the automatic generation of a dynamic model of an elastic belt transmission, created in the form of a multibody system. The publication presents reflections on this issue. There were developed scripts allowing one to automatically obtain the appropriate arrangement of links with respect to each other and to create the desired mechanical relations between or among them and other drive elements in the simulation. The programming bases of the proposed solutions were presented. Theoretical concepts were supported by application experiments in the Siemens NX environment. The method for the automatic generation of a multibody model of the toothed belt transmission, based on the geometric model of the latter, was developed. The work enabled the formulation of the conclusions on the development of modelling of the systems under consideration. The results of the research indicate high potential of the presented achievements. They constitute a basis for increasing the degree of computer technique aid for constructors or analysts.

Key words: multibody, modelling, Siemens NX, mechanics, drive, transmission.

1. INTRODUCTION

Currently, one of the methods of elastic transmission belt synthesis which is feasible in most computer aided engineering (CAE) environments is to present the selected component as a multibody system (MBS). In this case, the Kelvin-Voigt material flexibility model is applied, among others. A homogeneous object is divided into identical elements, characterised by the values of mass and moments of inertia. These can be, for instance, fragments with one tooth, if synchronous transmissions are taken into consideration. Their behaviour can be described by means of a defined number of state variables (space coordinates) [1].
The elasticity of individual links is not taken into consideration – they are perfectly rigid. As a result of connecting them by means of linear and angular springs and dampers (visible in Figure 1), individual bodies form a loop. Its line is subject to deformations caused by force actions, according to stiffness and damping parameters. In the simulation, the conditions of the contact between the transmission pulleys and the solids being belt components are defined. Thanks to such coupling of individual elements, the parts of the elastic object move on a path similar to the real one, observed during the operation of the mechanism [2, 3].

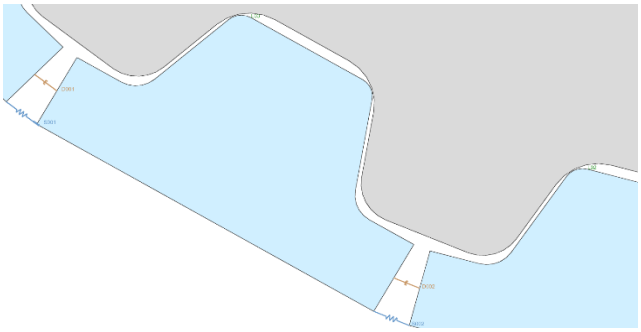

Fig. 1. Part of virtual multibody toothed belt based on Kelvin-Voigt model

The multitude and complexity of the equations describing the motion of the system under investigation make the computer aid necessary for the realisation of a multibody simulation in the circumstances under consideration. Contrary to the finite element method (FEM), in the MBS the elasticity of a given body does not result from its material-related features – appropriate connections between the solids being the components of the belt must be input. It makes the technique more time-consuming; yet, it allows one (in a way that is easier than in the case of the FEM) to show the condition of stress as early as the belt is in its initial position [4].

1.1 Procedure of multibody modelling of flexible toothed belt transmission in Siemens NX

In the Siemens NX application, a multibody model is built on the basis of a geometrical model. Data necessary for this purpose – obtained from the "Modelling" module, being a computer aided design (CAD) environment – mainly consists in the forms of

individual parts of the mechanism and their arrangement in space. Within the framework of further work, after launching the "Motion" multibody simulation tool, there can be distinguished 3 tasks, which constitute the subject matter of the considerations described in the present paper:
1) selecting objects which are to be used in the dynamic analysis (it is also indispensable to determine their mass parameters),
2) defining the conditions of contact of individual solids (in the case under consideration – the contact of all components of the belt with the pulleys),
3) putting bushing elements between the links (they make it possible to induce spring-like pulling and its damping in all axes between two rigid bodies – as assumed by Kelvin and Voigt) [5].

1.2 State of the art

The modelling procedure begins in the CAD environment. Regardless of the software used, the standard is that tools which facilitate and accelerate the generating and positioning of recurring instances of a given object are available (similar to the one presented in Figure 2). Parametric geometrical forms and their circular or linear arrays are sufficient for the application to automatically update the form of the transmission in the case of a change in the configuration of the latter (e.g. number of teeth of a given pulley or belt length) [5].

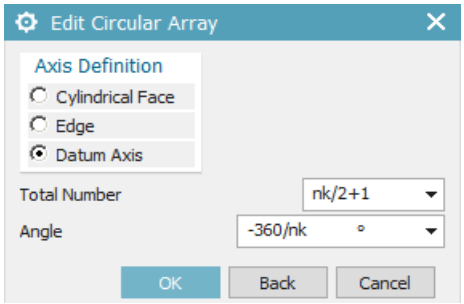

Fig. 2 . Parametric array generator

There is therefore no need to develop new solutions in this area. There also exists support for task no. 1 performed after switching to the "Motion" module, which is "Joint Wizard" tool show in Figure 3 [5].

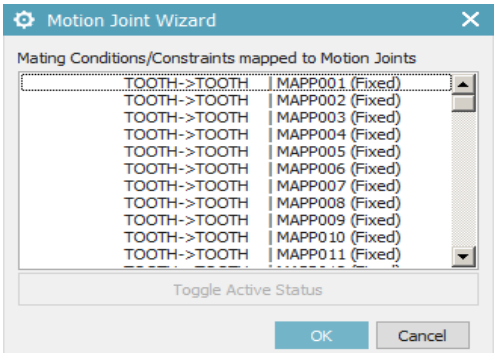

Fig. 3. "Motion Joint Wizard" window

Its purpose is, among others, to identify solids and characterise them in terms of size and mass distribution on the basis of their shape and density of the material declared in the "Modelling" module. However, in the finally proposed procedure, it was not introduced due to the dependencies discovered in the research and the guidelines for naming individual elements which condition the automation of further stages of modelling. Each of these stages consists of a number of repetitive actions, the objective of which is to assign the appropriate relations to all rigid bodies, and place the bushings (characterised in Figure 4) in the selected places of the system.

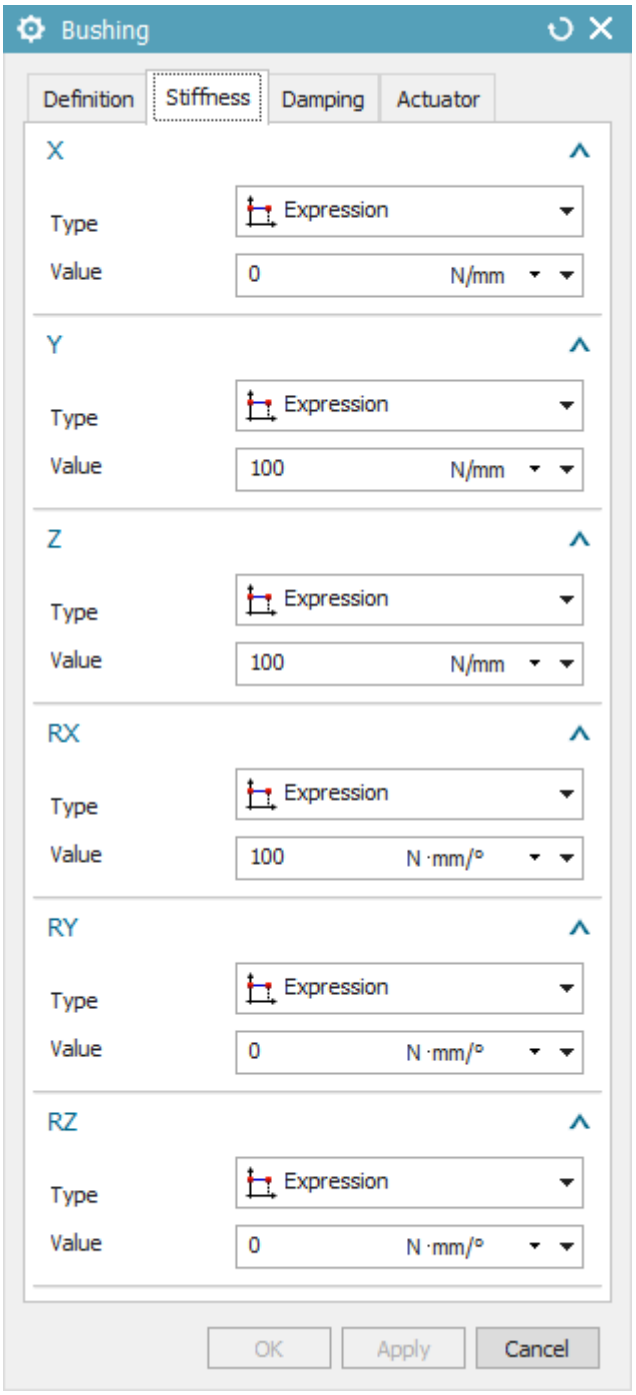

Fig. 4. Bushing configuration window

The standard command directory does not contain any commands that would help the user out in the desired area. However, the Siemens NX software enables the creation and running of scripts, invoking specific actions. These are called "journals", and several programming languages are available for the editing thereof, including Visual Basic. Therefore, standard commands apply, including, for instance, looping a given operation according to the adopted logical condition, with some changes in each iteration [5].

1.3 Research objectives and hypothesis

The objective of the research was to verify the possibility of the automation of all 3 tasks executed in the "Motion" module. What was suggested as a potential solution was the use of scripts created by means of the "journaling" tool and an appropriately prepared basic geometric model. It was assumed that objects selected while defining the elements of the multibody system (for example, the bushing origins on the arrayed solids) have sequentially numbered signatures, allowing one to refer to their location in the system. The work was expected to discover knowledge in the fields of:

1) the methods of referring to individual objects which are necessary in the definition of a given MBS element,
2) the conditions of the loop encompassing all instances necessary to be defined,
3) the characteristic variables in the case of a particular instance of a given MBS element,
4) the preparation of a basic geometric model or other previously created objects so that the desired references are possible,
5) the correctness of generating the elements of the model using the developed script.

2. METHODS & MATERIALS

The research case under consideration was a transmission consisting of two identical pulleys and a belt with 56 teeth, presented in Figure 5.

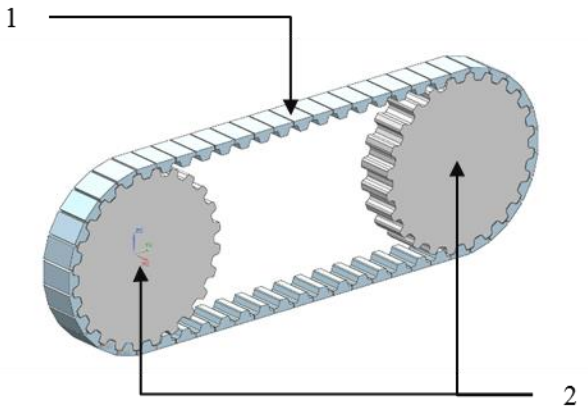

Fig. 5. View of object under consideration, where: 1 – teeth, 2 – pulleys

Attempts were made to create scripts to automate the definition of the transmission as a multibody system in the "Motion" module, using a previously prepared (in line with state of the art) geometric model. This was done within the framework of the procedure shown in Figure 6. The analysis was performed 3 times, in relation to 3 tasks from the chapter 1.1.

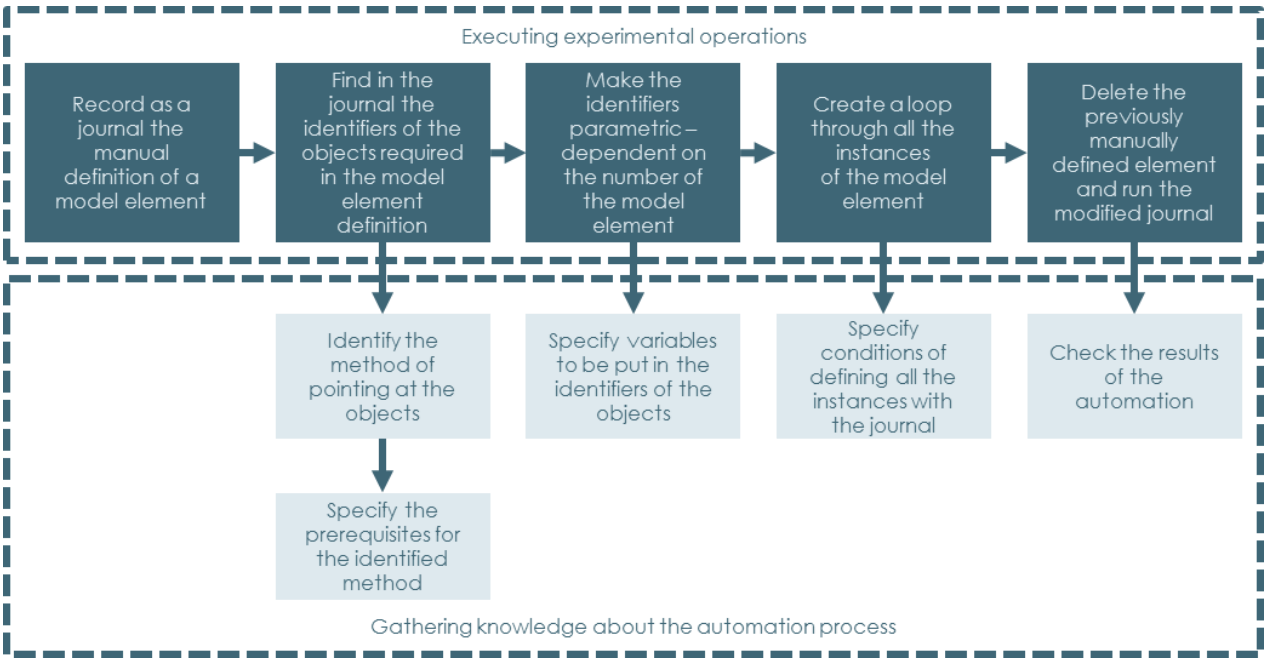

Fig. 6. Procedure of modelling and research

The whole led to acquiring information on the 5 issues mentioned in the chapter 1.3.

3. RESULTS

The present chapter, describing the results of the research on the automation, is divided into 3 parts corresponding consecutively to 3 tasks indicated in chapter 1.1. Each part introduces further division into 5 smaller units, in order to present the knowledge discovered in 5 areas mentioned in the chapter 1.3.

3.1 Definition of toothed belt links

The model comprises 56 instances – links containing one tooth, characterised in Figure 7.

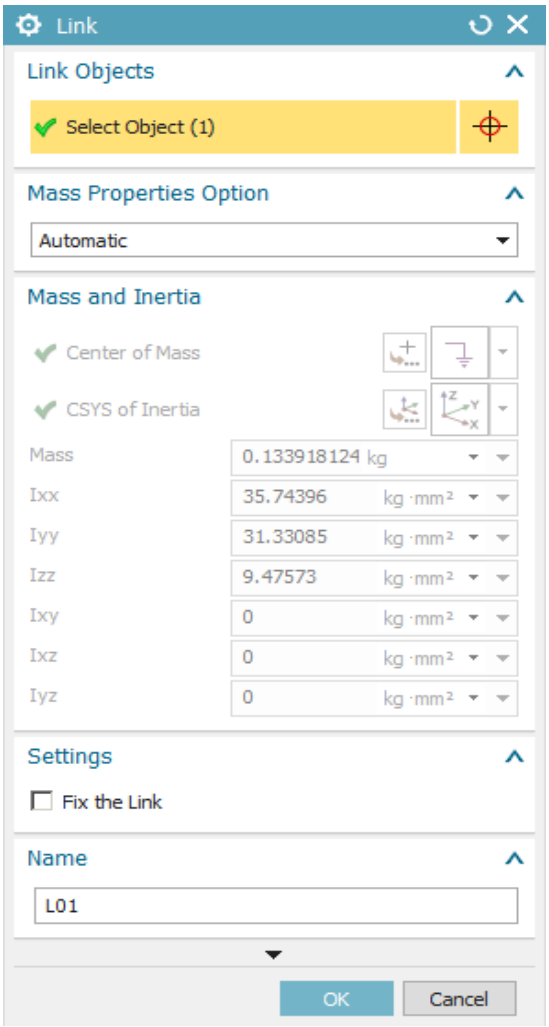

Fig. 7. Link definition window

The creation of a single instance requires selecting an object, its mass parameters and the name of a link.

3.1.1 Referring to individual objects

The object that forms a link ("body1") can be selected by means of the command to find it by its signature ("PARTIAL_PROTO#.Bodies|Body144") inside the component ("COMPONENT Tooth 1"), and the latter inside the assembly ("COMPONENT Transmission 1"). The proposition of a script for that operation is presented in Figure 8.

```
Dim component1 As Assemblies.Component =
 CType(
workPart.ComponentAssembly.RootComponent.
FindObject("COMPONENT Transmission 1"),
Assemblies.Component)

Dim component2 As Assemblies.Component =
 CType(component1.FindObject("COMPONENT
Tooth 1"), Assemblies.Component)

Dim body1 As Body = CType(
component2.FindObject(
"PARTIAL_PROTO#.Bodies|Body144"), Body)

Dim added1 As Boolean
added1 = linkBuilder1.Geometries.Add(
body1)
```

Fig. 8. References to link objects

It is possible to choose the automatic selection of mass parameters on the basis of the geometric model data. The proposition of a command for that operation is presented in Figure 9.

```
linkBuilder1.MassProperty.MassType =
Motion.LinkMassProperty.MassPropertyType.
Automatic
```

Fig. 9. Link parameter definition

3.1.2 Loop conditions

1 loop with the number of iterations equal to the number of teeth is required. The proposition of the loop statement is presented in Figure 10.

```
For nr_t As Integer = 1 To 56

    ...

Next
```

Fig. 10. Link loop condition

3.1.3 Characteristic variables

The signature of the geometric model object, which is selected when creating a given instance, may be correlated with the number of the iteration in the tooth loop ("nr_t"). The proposition of a command for that operation is presented in Figure 11.

```
Dim component2 As Assemblies.Component =
 CType(component1.FindObject("COMPONENT
Tooth " & nr_t), Assemblies.Component)
```

Fig. 11. Variables in link object reference

The name of the instance may also be correlated with the number of the iteration in the tooth loop ("nr_t"). The proposition of a command for that operation is presented in Figure 12.

```
linkBuilder1.Name = name_prefix & nr_t
```

Fig. 12. Variables in link name

The prefix of the name of the link changes after the given iteration due to the desired order in the list of model elements. The proposition of the conditional

statement is presented in Figure 13.

```
If nr_t > 9 Then
name_prefix = "L"
Else
name_prefix = "L0"
End If
```

Fig. 13. Condition of variable in link name

3.1.4 Preparation of previously created elements
Objects in the geometrical model must bear names provided in the reference, which is shown in Figure 14.

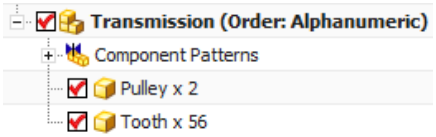

Fig. 14. Link object naming

The arrays must be created one after another, from the first link the last one, in order to maintain the order of numbering the solids forming the link, as visible in Figure 15.

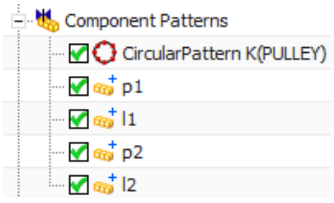

Fig. 15. Link object structure

In the geometrical model, an appropriate material needs to be assigned (in the window presented in Figure 16) to all solids for the program to correctly estimate their mass parameters.

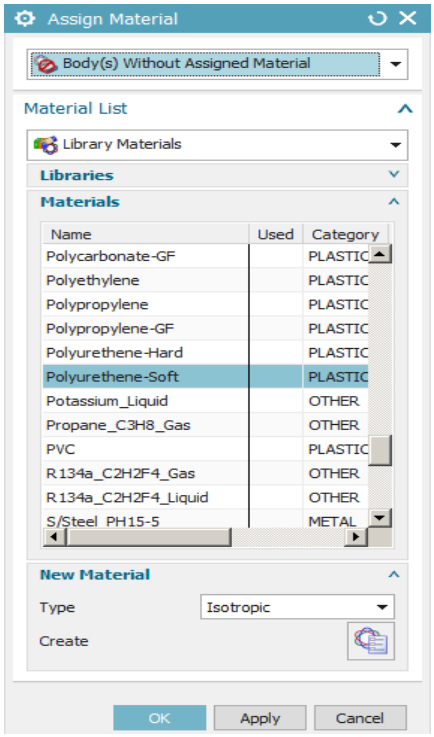

Fig. 16. Link object material parametrising

3.1.5. Correctness of generating model elements
After running the script, all the links were defined correctly in the order corresponding to the geometric model objects – the result is shown in Figure 17.

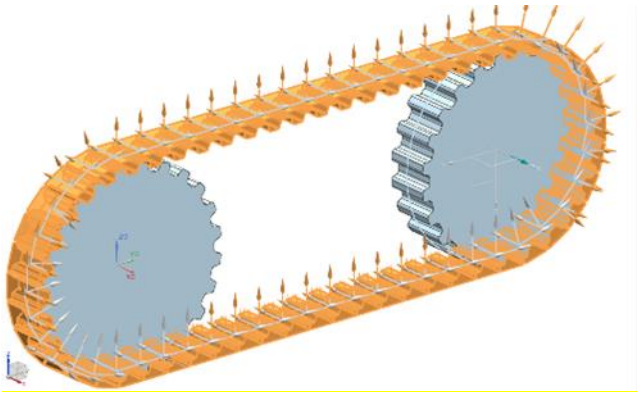

Fig. 17. Model after script-based definition of links

3.2 Definition of body contact conditions
The model comprises 112 instances of "CAD contact", each of which connects two objects – one of the 56 teeth with one of the 2 pulleys – as one can see in Figure 18.

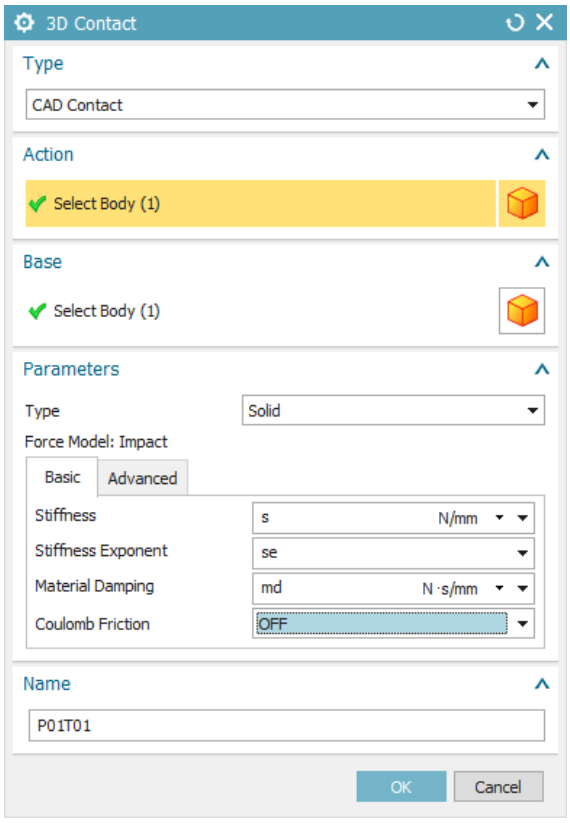

Fig. 18. Contact definition window

The creation of a single instance requires selecting 2 bodies and defining contact parameters.

3.2.1 Referring to individual objects
Bodies coming into contact ("body1" and "body2") can be selected by means of the command to find the object by its signature

("PARTIAL_PROTO#.Bodies|Body144" as well as "PARTIAL_PROTO#.Bodies|Body8") inside the components ("COMPONENT Tooth 1" as well as "COMPONENT Pulley 1"), and the latter inside the assembly ("COMPONENT Transmission 1"). The proposition of a script for that operation is presented in Figure 19.

```
Dim component1 As
NXOpen.Assemblies.Component = CType(
workPart.ComponentAssembly.RootComponent.Find
Object("COMPONENT Transmission 1"),
NXOpen.Assemblies.Component)

Dim component2 As
NXOpen.Assemblies.Component = CType(
component1.FindObject("COMPONENT Tooth 1"),
NXOpen.Assemblies.Component)

Dim body1 As NXOpen.Body = CType(
component2.FindObject(
"PARTIAL_PROTO#.Bodies|Body144"),
NXOpen.Body)

Dim added1 As Boolean = Nothing
added1 =
bodyContactBuilder1.FirstContactGeometry.Add
(body1)

Dim component3 As
NXOpen.Assemblies.Component = CType(
component1.FindObject("COMPONENT Pulley 1"),
 NXOpen.Assemblies.Component)

Dim body2 As NXOpen.Body = CType(
component3.FindObject(
"PARTIAL_PROTO#.Bodies|Body8"), NXOpen.Body)

Dim added2 As Boolean = Nothing
added2 =
bodyContactBuilder1.SecondContactGeometry.Add
(body2)
```

Fig. 19. References to contact objects

In the numerical parameter fields, there can be entered expressions, which enables the user to regulate all occurrences of a given quantity by means of a single change of the numerical value.
The proposition of the parametrising statements is presented in Figure 20.

```
bodyContactBuilder1.RecurdynParameters.
StiffnessExpression.RightHandSide = "s"

bodyContactBuilder1.RecurdynParameters.
ForceExponentExpression.RightHandSide
= "se"

bodyContactBuilder1.RecurdynParameters.
MaterialDampingExpression.RightHandSide
 = "md"
```

Fig. 20. Parametrising contact

3.2.2 Loop conditions
2 loops are required: one with the number of iterations equal to the number of pulleys; the other with the number of iterations equal to the number of teeth. The proposition of the loop statement is presented in Figure 21.

```
For nr_p As Integer = 1 To 2

    For nr_t As Integer = 1 To 56

        ...

    Next

Next
```

Fig. 21. Contact loop conditions

3.2.3 Characteristic variables
The signatures of the geometric model objects, which are selected when creating a given instance, may be correlated with the numbers of iterations in the pulley loops ("nr_p") and tooth loops ("nr_t"). The proposition of a script for that operation is presented in Figure 22.

```
Dim component2 As
NXOpen.Assemblies.Component = CType(
component1.FindObject("COMPONENT
Tooth " & nr_t),
NXOpen.Assemblies.Component)

Dim component3 As
NXOpen.Assemblies.Component = CType(
component1.FindObject("COMPONENT
Pulley " & nr_p),
NXOpen.Assemblies.Component)
```

Fig. 22. Variables in references of contact objects

The name of the instance may also be correlated with the numbers of iterations in the pulley loops ("nr_p") and tooth loops („nr_t"). The proposition of a command for that operation is presented in Figure 23.

```
bodyContactBuilder1.Name =
"P0" & nr_p & name_prefix &
nr_t
```

Fig. 23. Variables in contact name

3.2.4 Preparation of previously created elements
Expressions input as parameter values must be declared beforehand (in the window presented in Figure 24).

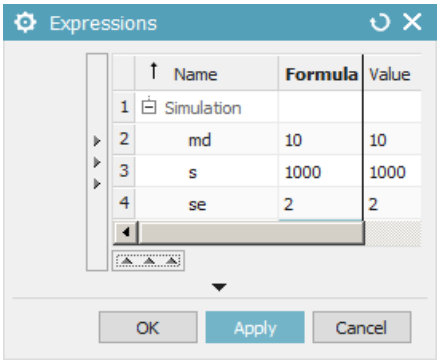

Fig. 24. Expressions with contact parameters

3.2.5. Correctness of generating model elements

After running the script, all the "CAD contacts" were defined correctly in the order corresponding to the geometric model objects – the result is shown in Figure 25.

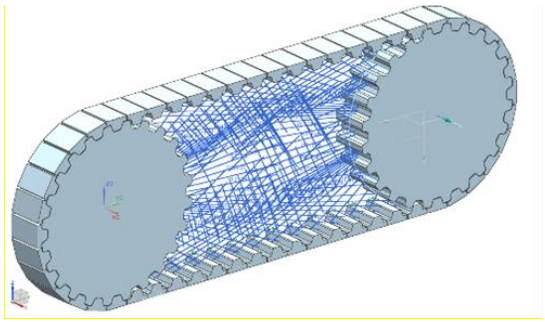

Fig. 25. Model after script-based definition of contacts

3.3 Bushing putting

The model comprises 56 instances of bushing – between each of 56 links – as one can see in Figure 26.

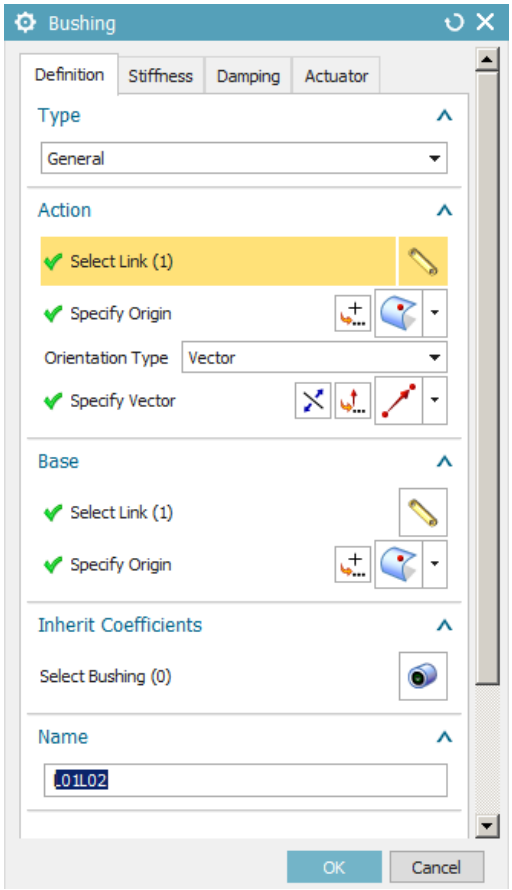

Fig. 26. Bushing definition window

The creation of a single instance requires selecting 2 links which are attracted to each other, the places of origins of bushing in them, the direction of its action and the parameters of stiffness and damping in selected axes.

3.3.1 Referring to individual objects

The links to which bushing are anchored ("link1", "link2") can be selected by means of the command to find the object by its signature ("L01" and "L02") The proposition of a script for that operation is presented in Figure 27.

```
Dim link1 As
NXOpen.Motion.Link = CType(
workPart.MotionManager.Links.
FindObject("L01"),
NXOpen.Motion.Link)

bushingBuilder1.BushingDefine
.ActionLink.Value = link1

Dim link2 As
NXOpen.Motion.Link = CType(
workPart.MotionManager.Links.
FindObject("L02"),
NXOpen.Motion.Link)

bushingBuilder1.BushingDefine
.ReactionLink.Value = link2
```

Fig. 27. References to bushing objects (1)

As bushing origins ("point 1" and "point2"), one can select the points located in the middle ("scalar1") of the length and width of given surfaces ("face1" and "face2") in the components searched by name ("Tooth 1" and "Tooth 2") inside the assembly ("Transmission 1"). The proposition of a script for that operation is presented in Figure 28.

```
Dim component1 As NXOpen.Assemblies.Component =
CType(
workPart.ComponentAssembly.RootComponent.FindObject
("COMPONENT Transmission 1"),
NXOpen.Assemblies.Component)

Dim component2 As NXOpen.Assemblies.Component =
CType(component1.FindObject("COMPONENT Tooth 1"),
NXOpen.Assemblies.Component)

Dim face1 As NXOpen.Face = CType(
component2.FindObject(
"PARTIAL_PROTO#.Bodies|Body144|HANDLE R-179588"),
NXOpen.Face)

Dim scalar1 As NXOpen.Scalar = Nothing
scalar1 = workPart.Scalars.CreateScalar(0.5,
NXOpen.Scalar.DimensionalityType.None,
NXOpen.SmartObject.UpdateOption.AfterModeling)

Dim point1 As NXOpen.Point = Nothing
point1 = workPart.Points.CreatePoint(face1,
scalar1, scalar1,
NXOpen.SmartObject.UpdateOption.AfterModeling)

bushingBuilder1.BushingDefine.ActionPoint = point1

Dim component3 As NXOpen.Assemblies.Component =
CType(component1.FindObject("COMPONENT Tooth 2"),
NXOpen.Assemblies.Component)

Dim face2 As NXOpen.Face = CType(
component3.FindObject(
"PARTIAL_PROTO#.Bodies|Body144|HANDLE R-687897"),
NXOpen.Face)

Dim point2 As NXOpen.Point = Nothing
point2 = workPart.Points.CreatePoint(face2,
scalar1, scalar1,
NXOpen.SmartObject.UpdateOption.AfterModeling)

bushingBuilder1.BushingDefine.ReactionPoint =
point2
```

Fig. 28. References to bushing objects (2)

The direction of the bushing action can be defined by

selecting the points used beforehand. The proposition of a script for that operation is presented in Figure 29.

```
Dim direction1 As NXOpen.Direction = Nothing
direction1 = workPart.Directions.CreateDirection(
point1, point2,
NXOpen.SmartObject.UpdateOption.AfterModeling)

bushingBuilder1.BushingDefine.Direction =
direction1
```

Fig. 29. References to bushing objects (3)

3.3.2 Loop conditions

1 loop with the number of iterations equal to the number of links is required. The proposition of the loop statement is presented in Figure 30.

```
For nr_l As Integer = 1 To 56

    ...

Next
```

Fig. 30. Bushing loop condition

Additionally, on the basis of the number of the iteration in the link loop, the number of the second link to select in the bushing definition is determined. The proposition of the conditional statement is presented in Figure 31.

```
If nr_l < 56 Then
nr_l_2 = nr_l+1
Else
nr_l_2 = 1
End If
```

Fig. 31. Bushing "base" object numeration

3.3.3 Characteristic variables

The signatures of the objects, which are selected when creating a given instance, may be correlated with the prefixes matched to the numbers of links (see: chapter 3.1.3), number of the "action"-type link ("nr_l") – identical to the number of the iteration in the link loop – and to the number of the "base" type link ("nr_l_2"). The proposition of a script for that operation is presented in Figure 32.

```
Dim link1 As NXOpen.Motion.Link = CType(
workPart.MotionManager.Links.FindObject(
name_prefix & nr_l), NXOpen.Motion.Link)

Dim link2 As NXOpen.Motion.Link = CType(
workPart.MotionManager.Links.FindObject(l_prefix_2
 & nr_l_2), NXOpen.Motion.Link)
```

Fig. 32. Variables in references of bushing objects (1)

The signatures of the geometric model objects, which are selected when creating a given instance, may be correlated with numbers of links in the given iteration. The proposition of a script for that operation is presented in Figure 33.

```
Dim component2 As NXOpen.Assemblies.Component =
CType(component1.FindObject("COMPONENT Tooth " &
nr_l), NXOpen.Assemblies.Component)

Dim component3 As NXOpen.Assemblies.Component =
CType(component1.FindObject("COMPONENT Tooth " &
nr_l_2), NXOpen.Assemblies.Component)
```

Fig. 33. Variables in references of bushing objects (2)

The name of the instance may also be correlated with the prefixes matched to the numbers of links and on the numbers of links in a given iteration.
The proposition of a command for that operation is presented in Figure 34.

```
bushingBuilder1.BushingDefine.Name = name_prefix &
 nr_l & name_prefix_2 & nr_l_2
```

Fig. 34. Variables in bushing name

3.3.4 Preparation of previously created elements

Bushing origins must have unique coordinates. It is guaranteed when the single-toothed solids in the geometric model do not come into contact with each other at any point – as visible in Figure 35.

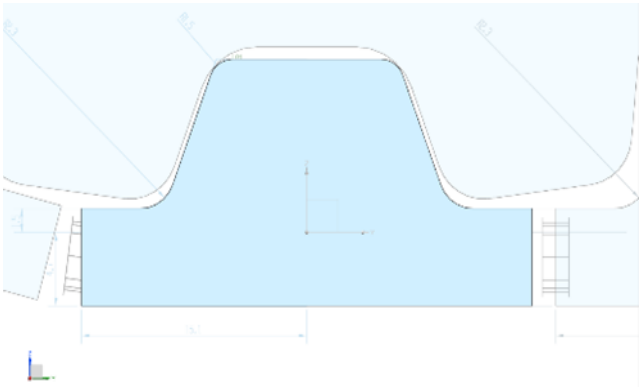

Fig. 35. Providing distance between bushing origins

Potentially problematic areas are the rectilinear sections, where the belt components may be adjacent to each other. The solution is to set their length as the value lower than the one resulting from dividing the total length of the "chain" by the number of its links. This introduces the drawing away of the walls of adjacent objects, thus avoiding potential errors in the proposed procedure of the automatic generation of the elements under consideration.

3.3.5. Correctness of generating model elements

After running the script, all the bushings were defined in the correct places, in the order corresponding to the geometric model objects – the result is shown in Figure 36.

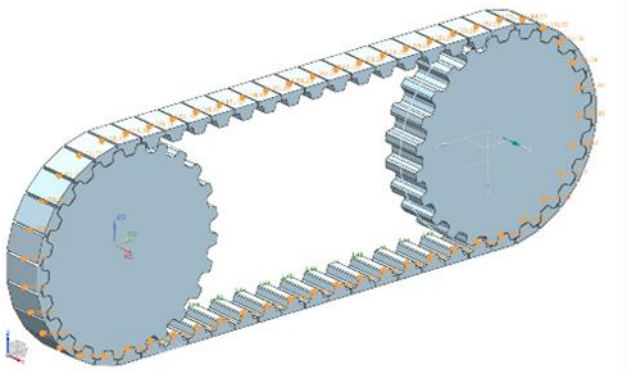

Fig. 36. Model after script-based definition of bushings

4. CONCLUSIONS

Within the framework of the research, there was developed the method for the automatic generation of a multibody model of the toothed belt transmission, based on the geometric model of the latter, in the Siemens NX environment. When the appropriate signatures and structures are applied in the "Modelling" module, the proposed procedures are feasible regardless of the form and number of the mechanism elements. By using the chosen method – scripts invoking and repeating certain operations according to the selected pattern – the automation can be extended to all activities necessary in the modelling process. This way, a comprehensive generator of the desired MBS would be created, which can be operated without the need for the manual preparation of the elements beforehand. All requirements would then be met, with no dependence on the user and their supervision. Such an approach eliminates the risk of making mistakes and allows one to achieve the desired effect even in the absence of expert knowledge. The actions declared in the "journals" beforehand ensure the correct construction of the virtual transmission, allowing it to be configured to some extent. The changes which would follow would update themselves immediately after the introduction thereof (in the case of parameters regulated by expressions) or would require the re-creation of the given elements (in the case of features indicated directly in the script). Such a tool can be operated from the graphical user interface (GUI) level, which in the environment under consideration is also created within the script. It would be used to invoke the desired commands with parameters determined manually or selected from a catalogue of predefined values (e.g. subassembly series of types). Most of the automation commands are not complicated and require at most entering numerical values in the right places. Those related to algorithmic tasks and to referring to objects at various levels of the model structure were taken into consideration within the framework of the study described in the present publication. Programming solutions were proposed in this area and their effectiveness was confirmed by analysing the result of their application. They constitute an important contribution to accelerating and facilitating computer-aided analysis of toothed belt transmissions.

5. REFERENCES

1. Kubas K., (2015). *A model for analysing the dynamics of belt transmissions with a 5pk belt*, The Archives of Automotive Engineering, **67**(1).
2. Callegari M., Cannella F., (2001). *Lumped parameter model of timing belt transmissions*, 5th AIMETA Congress of Theoretical and Applied Mechanics, Taormina, Italy, Sept.
3. Čepon G., Manin L., Boltežar M., (2009). *Introduction of damping into the flexible multibody belt-drive model: A numerical and experimental investigation*, Journal of Sound and Vibration, **324**.
4. Čepon G., Manin L., Boltežar M., (2011). *Validation of a Flexible Multibody Belt-Drive Model*, Journal of Mechanical Engineering, **57**, 7-8.
5. Siemens NX software documentation, https://docs.plm.automation.siemens.com/tdoc/nx/12.0.2/nx_help, [Accessed 24.05.2020/05].

Received: March 15, 2021 / Accepted: December 20, 2021 / Paper available online: December 25, 2021