SIEMENS

Learn-/Training Document

Siemens Automation Cooperates with Education (SCE) | NX MCD V12/TIA Portal V15.0 or higher

DigitalTwin@Education Module 150-005
Creation of a Dynamic 3D Model Using the Mechatronics Concept Designer CAE System

siemens.com/sce

SIEMENS
Global Industry Partner of WorldSkills International

worldskills

Matching SCE trainer packages for this Learn-/Training Document

SIMATIC STEP 7 Software for Training (incl. PLCSIM Advanced)

- **SIMATIC STEP 7 Professional V15.0 - Single License**
  Order no.: 6ES7822-1AA05-4YA5
- **SIMATIC STEP 7 Professional V15.0 - Classroom License for 6 Users**
  Order no.: 6ES7822-1BA05-4YA5
- **SIMATIC STEP 7 Professional V15.0 - Upgrade License for 6 Users**
  Order no.: 6ES7822-1AA05-4YE5
- **SIMATIC STEP 7 Professional V15.0 - Student License for 20 Users**
  Order no.: 6ES7822-1AC05-4YA5

SIMATIC WinCC Engineering/Runtime Advanced software in the TIA Portal

- **SIMATIC WinCC Advanced V15.0 - Classroom License for 6 Users**
  6AV2102-0AA05-0AS5
- **Upgrade SIMATIC WinCC Advanced V15.0 - Classroom License for 6 Users**
  6AV2102-4AA05-0AS5
- **SIMATIC WinCC Advanced V15.0 - Student License for 20 Users**
  6AV2102-0AA05-0AS7

NX V12.0 Educational Bundle (Schools, universities, not for in-company training centers)

- **Contact**: academics.plm@siemens.com

Additional information regarding SCE

siemens.com/sce

Information regarding use

The SCE Learn-/Training Document for the integrated automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for training purposes for public educational and R&D institutions. Siemens does not guarantee the contents.

This document is only to be used for initial training on Siemens products/systems. This means it can be copied in whole or in part and given to trainees/students for use within the scope of their training/course of study. Disseminating or duplicating this document and sharing its content is permitted within public training and advanced training facilities for training purposes or as part of a course of study.

Exceptions require written consent from Siemens. Send all related requests to scesupportfinder.i-ia@siemens.com.

We wish to thank the HS Darmstadt, particularly Mr. Heiko Webert and Mr. Prof. Dr.-Ing. Stephan Simons, and all other persons involved for their support in the preparation of this SCE learning/training document.

Table of contents

List of figures

List of tables

Creation of a Dynamic 3D Model Using the Mechatronics Concept Designer CAE System

1 Goal

In Module 4 of the DigitalTwin@Education workshop series, you have performed the first steps in creating 3D models. You were able to successfully model all the individual models needed for the sorting plant. You then inserted and positioned them in an assembly in such a way as to resemble the appearance of the model provided in Module 1.

The goal of this module is to provide your static models with dynamic properties in order to allow physical simulations. In accomplishing this, you will become acquainted with the basic operations and functionalities of the Mechatronics Concept Designer (MCD) NX add-on.

2 Requirement

For this module, you should brush up on your knowledge about static models. It is therefore recommended that you complete Module 4 of this workshop series beforehand. To successfully understand the dynamic processes of the model, you should be familiar with how the sorting plant works. Refer to Module 1 of this workshop series, in particular for more detailed descriptions of this.

3 Required hardware and software

The following components are required for this module:

1 **Engineering station**: Requirements include hardware and operating system (for additional information: see Readme on the TIA Portal Installation DVDs, and in the NX software package)

2 **NX software with Mechatronics Concept Designer add-on** – V12.0 or higher

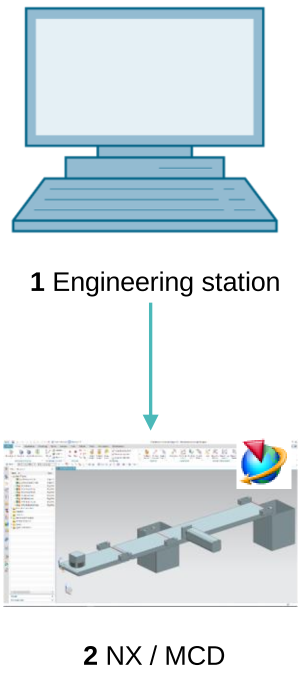


Figure 1: Overview of required software and hardware components in this module

Figure 1 clearly shows that the engineering station is the only hardware component of the system. The remaining components are based exclusively on software.

4 Theory

4.1 Dynamic 3D model

A dynamic 3D model can be created on the basis of a static 3D model in Mechatronics Concept Designer. This is done by defining dynamic properties in a static model. For example, dynamic properties describe the behavior of bodies under the influence of gravity and the reaction of a model under the action of forces. The dynamization thus enables you to run a simulation, as you have used previously in Modules 1 - 3 of this workshop series.

However, it is NOT possible to create a dynamic model without having a static 3D model.

When it comes to dynamization, the level of detail in the static 3D model is an essential criterion in determining the quality of a digital twin. The more detailed the static model, the closer it can approximate the behavior of a real plant when subsequently dynamized. As already mentioned, however, it is not possible to assign dynamic properties to non-static objects.

The dynamic behavior of a 3D model itself is also a definitive criterion for the level of detail. Even if the static model you created is extremely precise, an appropriate level of dynamization must be present. At the same time, it is not necessary to provide every static model with all possible dynamic properties. On the contrary, you should be aware of what exactly is to be simulated in the digital twin and only introduce dynamizations in the model that are relevant for this application. The more dynamic properties that are defined, the greater the required computing capacity will be for the simulation.

Before creating a 3D model, it is therefore important to define clear specifications regarding which tasks and functions of the plant or components to be modeled are to be covered. Only then can the effort for creating the dynamic model and the computing capacity for running the simulation be realistically estimated.

4.2 Dynamic properties in Mechatronics Concept Designer

The Mechatronics Concept Designer is an add-on module for NX. It allows you to assign dynamic properties to static models created previously in NX so that these models display a defined physical behavior. This is made possible by using of an integrated physics engine that calculates the physical and kinematic properties. Chapters 4.2.1 and 4.2.2 identify and briefly explain some of the possible dynamic properties of the program, to the extent these are required for completing this module.

The workspace of Mechatronics Concept Designer is shown in Figure 2. To open this application, you can use the familiar Command Finder at the top right of the screen to search for the "Mechatronics Concept Designer" application.

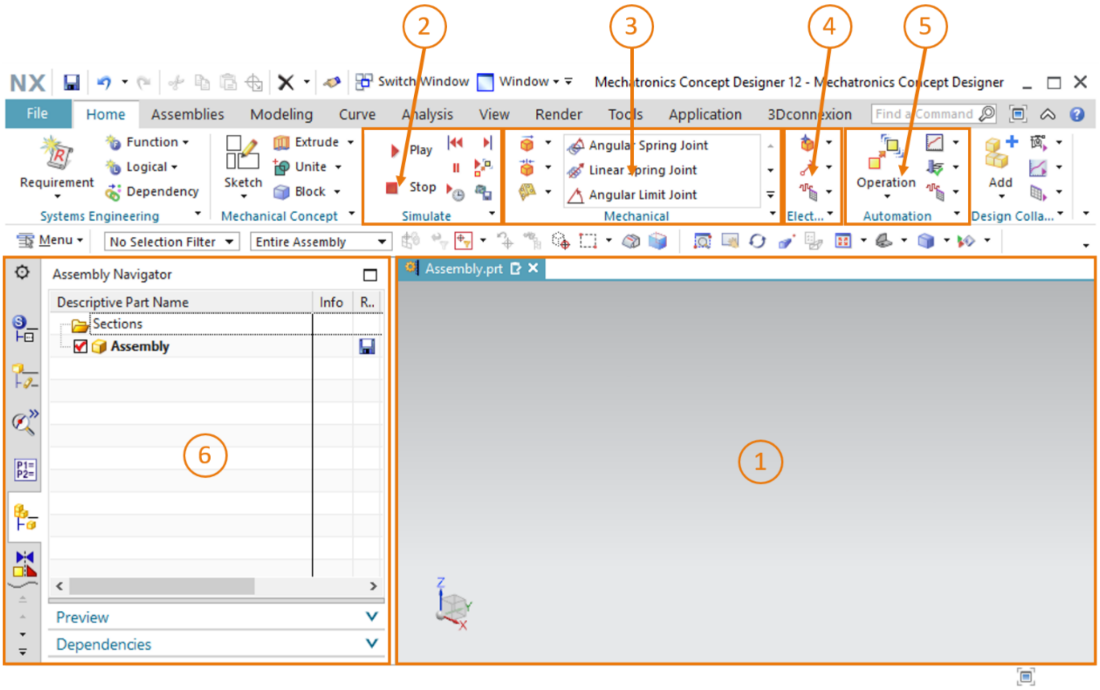


Figure 2: "Mechatronics Concept Designer" application in NX with markings for explaining the areas in the text

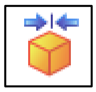
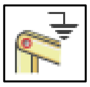
The following windows are used in this application to define dynamic properties for a model:

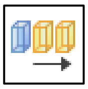
- The central screen again contains the three-dimension graphics window (see Figure 2, area 1). Here, for example, model surfaces can be selected for assigning dynamic properties.
- You can control the simulation of your model in the application from a group located in the middle of the menu bar (see Figure 2, area 2).
- Another group in the middle of the menu bar (see Figure 2, area 3) lists all the dynamic properties of Mechatronics Concept Designer from the field of mechanics. These include rigid and collision bodies, which are explained in more detail in Chapter 4.2.1.
- You will find the dynamic properties from the field of electrics in the menu group next to the dynamic mechanical properties (see Figure 2, area 4). Sensors and controls are mainly listed here. The dynamic electrical properties that are relevant for this module are described in Chapter 4.2.2.

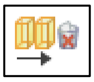
- The menu bar also has a group for the dynamic properties from the field of automation (see Figure 2, area 5). These properties include motion profiles as well as signal assignments for control by external programs such as PLCSIM Advanced. These functionalities are not used in this module.
- From the Resource bar on the left part of the screen (see Figure 2, area 6), you can open the Physics Navigator tab, among others, where all physical properties of an assembly or a model can be displayed. In addition, the Runtime Inspector allows you to change values of the physical properties during an active simulation. The Runtime Inspector is explained in Chapter 4.3.

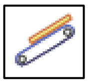
4.2.1 Dynamic and mechanical properties in Mechatronics Concept Designer

This chapter describes several dynamic properties from the field of mechanics that are required in this module for dynamizing the sorting plant. This set of properties is described here in order to give a brief, up-front overview of the types of mechanical dynamics in Mechatronics Concept Designer and their functions.

- The **Rigid Body** function  can be used to define a static model as a movable body. For this, the model is assigned a rigid body with mass that can react to actions of external forces. If a body is not assigned a rigid body, it remains immovable.
- A model or a surface of a model can be specified as a **Collision Body** . The model or the surface of a model is then able to collide with other models that have also been defined as collision bodies. The manner in which they collide mainly depends on the collision shape used for the model. You can find a list of the possible collision shapes along with a brief description in the online help of NX (see Chapter 9, Link [1]). Note that a rigid body of the model does not have to exist to create a collision body.
- The **Fixed Joint** function  can be used to prevent a rigid body from leaving a prescribed position in space. A fixed joint sets all the degrees of freedom of a rigid body, thereby preventing any motion.

- The **Object Source** function  allows a rigid body to be automatically generated as a new replica of the body during the course of a simulation. As a result of this, multiple replicas of a rigid body can exist in parallel and as fully independent bodies within a simulation. The generation of a new replica can be triggered on a time-driven or event-driven basis.

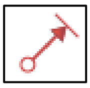
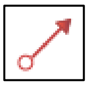
- A collision body can be selected in the **Object Sink** function . If a body from an object source comes in contact with this collision body, this object is removed again. This causes only this particular replica to disappear from the object source. All other replicas are retained.

- A **Transport Surface**  can be used to turn any planar surface into a conveyor belt. Bodies with collision surfaces can be transported on them in a specified direction. Motion can be along a straight path or a curved path.

- The **Sliding Joint**  allows a rigid body to execute a movement in relation to another rigid body along a vector. Other motions in other directions are prevented.

4.2.2 Dynamic and electrical properties in Mechatronics Concept Designer

An overview of the dynamic properties from the field of electrics that are relevant for dynamization of the sorting plant in this module is presented below.

- By defining a **Collision Sensor** , a component in an assembly can detect collisions with a collision body. This signal, which is usually a Boolean signal, enables reactions to certain situations.
- When a **Position Control**  is created, an actuator can be moved along a specified axis up to a defined position. For this, an existing model with a kinematic component, such as a sliding joint or transport surface, must be selected as the actuator.
- The **Speed Control** function  can be used to move an actuator along a specified axis at a specified speed. For this, you need to select an existing model with a kinematic component, such as a sliding joint or transport surface, as the actuator.

NOTE

For further information on other dynamic properties in Mechatronics Concept Designer, you can search for corresponding entries in the online help (see Chapter 9, Link [2]).

It is recommended that the search be conducted using English terms since the online help contains only an incomplete set of German terms.

4.3 Simulation capability of Mechatronics Concept Designer

With the help of a physics engine, it is possible to run simulations of models and bodies with physical and kinematics attributes in Mechatronics Concept Designer. Several functions are available for controlling a simulation. The most important commands include:

- **Play simulation** , which causes the models and bodies to act according to their respective, defined dynamic properties. This also includes interaction with other models that are provided with dynamic attributes.
- **Stop simulation** for exiting simulation mode.

It should be noted that a simulation can have a highly adverse effect on the performance of your engineering PC, depending on the scope and level of detail of the physical properties included. Therefore, you should attempt to simulate only as many properties as you need for testing your dynamic 3D model.

For checking your inserted dynamization, use of the **Runtime Inspector** in Mechatronics Concept Designer is recommended as the first step. This allows you to modify input parameters of physical properties and check the changes of output parameters during an active simulation. The destination of a position control is an example of an input parameter you can specify. Detection of a collision by a collision sensor is an example of an output parameter you can see.

Section: Adding and controlling a property in the simulation

To add a physical property to the Runtime Inspector, open the "**Physics Navigator**" tab in the Resource bar  (see Figure 3, step 1). Right-click on the desired property and select the "**Add to Inspector**" command (see Figure 3, step 2).

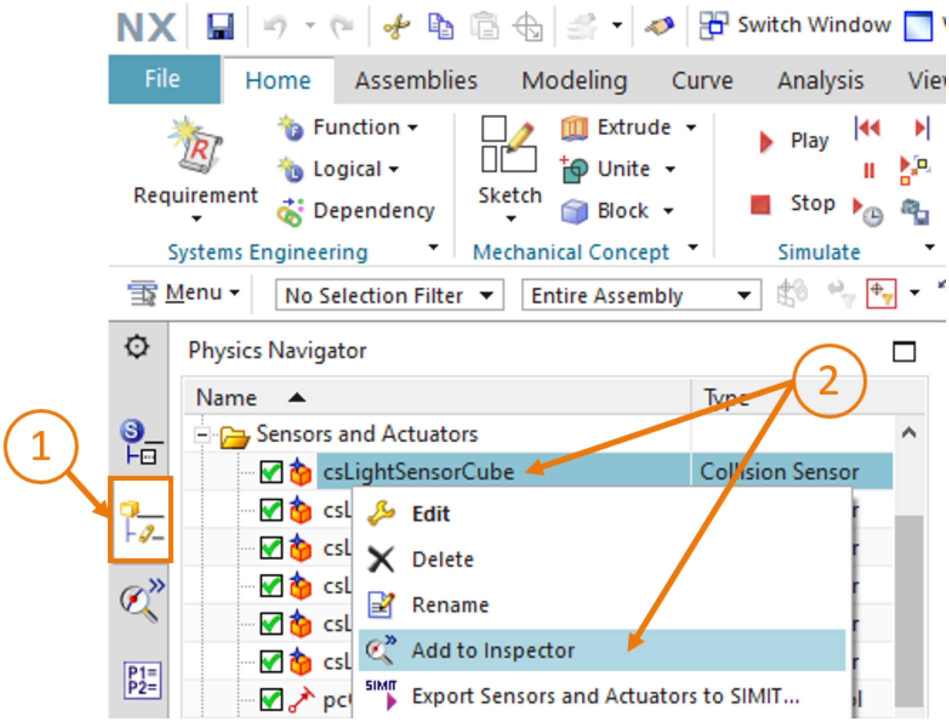


Figure 3: Adding a dynamic attribute to the Runtime Inspector

Go to the "**Runtime Inspector**" tab  (see Figure 4, step 1). There you will see an overview of all the added dynamic properties you want to monitor. Input parameters can be changed during a simulation. These can be Bool or Real data type parameters (see Figure 4, step 2).

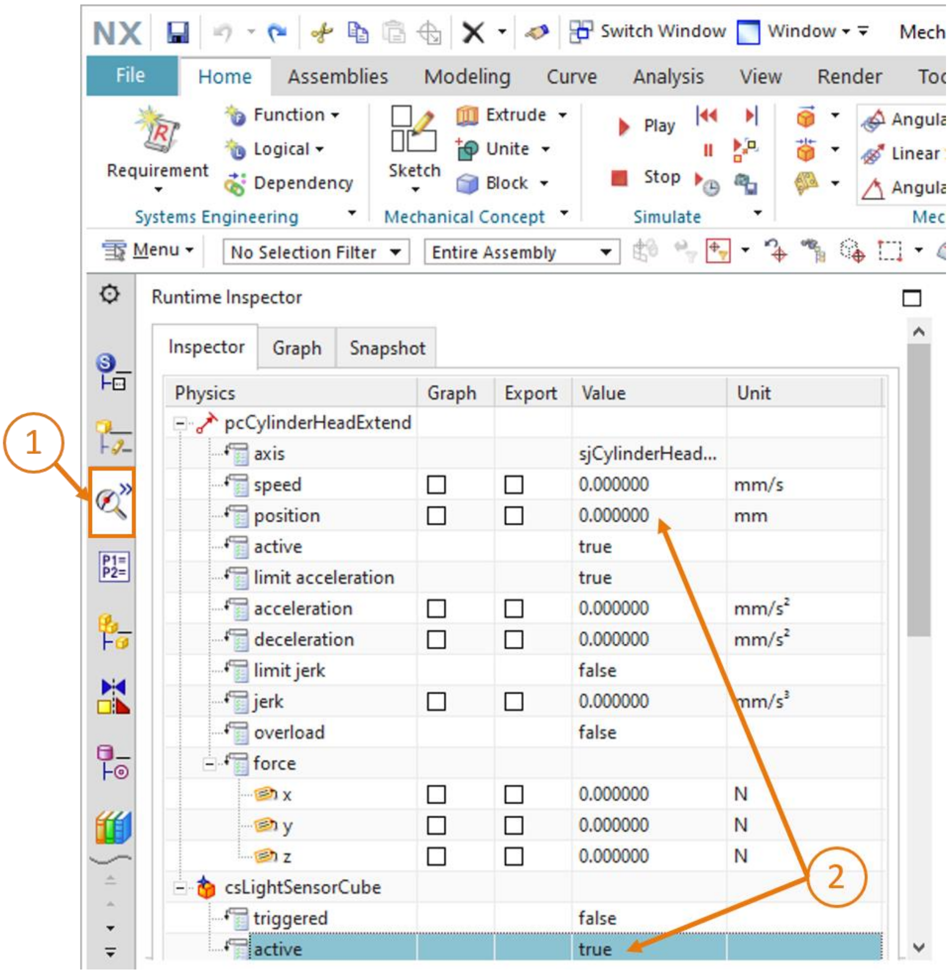


Figure 4: Runtime Inspector with option for changing and monitoring parameters

To remove information from the Runtime Inspector, right-click on the desired property and click "Remove".

5 Task

In this module, you are to expand the static 3D model of the sorting plant you created in Module 4 to include dynamic properties that are required for virtual commissioning.

You will use the Mechatronics Concept Designer (MCD) NX application for this. You can use this application to define physical attributes of individual models and to specify interactions with other models. In so doing, you will become acquainted with the functionality of various dynamic elements in MCD. Using the simulation environment integrated in MCD, you can then test the behavior of your model.

6 Planning

This dynamic 3D model requires at least Version **V12.0** of the **NX** CAD system. The **Mechatronics Concept Designer (MCD)** add-on module must also be available in NX.

You require an understanding of static 3D models, for example, by having completed Module 4.

If you are unclear about how the sorting plant works, you should review the theory section of Chapter 4.2 of **Module 1** again.

The Siemens "**Guide to Standardization**" has been followed for naming the various dynamic properties. You can find this guide by going to Chapter 9 and selecting the link there [2].

Programming of the PLC, visualization and creation of a virtual PLC for simulation purposes is not included in this module.

7 Structured step-by-step instructions

The project "**150-005_DigitalTwinAtEducation_NX_dynModel**" is made available with this module. The project consists of two folders:

- "**fullStatModel**" contains the complete static 3D model of the sorting plant from Module 4. You can use this model for this module if your results from Module 4 are incomplete.
- "**fullDynModel**" contains the solution for this module in case you need help completing a step.

As a reminder, use the Command Finder if you are unable to find a command or an application in the development environment at any time in this module. This is located in the upper right part of the NX user interface screen, as shown in Figure 5.

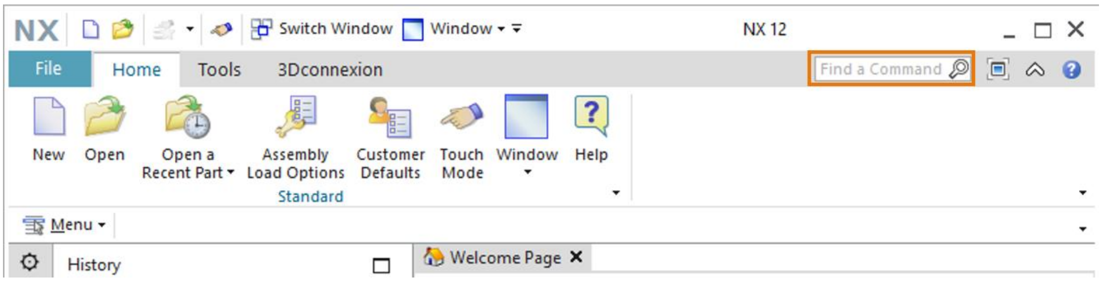


Figure 5: Command Finder in the NX menu, highlighted in orange

You can select the appropriate command from the search results. NX also shows you where the command is located so that you can also select it directly from the menu in the future.

IMPORTANT: The user interface and the arrangement of various commands in the menus change with new versions of NX. In addition, users can define their own user interfaces. While the following descriptions depict the standard user interface of NX12.0, the interface in your version may be different. **For this reason, if you are unable to find a command in the window at the positions described, use the Command Finder.**

Also bear in mind that this description serves only as a suggested solution. There, are countless ways to represent dynamic behavior in MCD. The aim here was to describe a procedure that is easy to follow and can interact with a virtual PLC from Modules 1-3 without complications. Of course, you are also free to try out various other ways here.

Note that certain passages in this module are labeled as "Sections". Because frequent reference is made to these passages in the course of the description, this labeling is intended as an orientation guide.

7.1 Opening an assembly in the Mechatronics Concept Designer application

In this chapter, you are to open your assembly from Module 4 in NX and start the Mechatronics Concept Designer (MCD) application.

Proceed as follows for this:

→ Create a copy of the models you created in Module 4 on your operating system and save them in a new folder on your file system. As mentioned in Chapter 7, if you have an incomplete model, you can also access the provided "**fullStatModel**" project and create a working copy from this folder.

→ Start NX and wait until the program opens and you see the Home page. Click the "**Open**" button (see Figure 6, step 1) and navigate to the folder you created beforehand. You now see the parts created in Module 4. Select the "**assSortingPlant**" assembly, which contains the complete static 3D model of the sorting plant (see Figure 6, step 2). Select the "**Partially Load**" option (see Figure 6, step 3) so that only the models of the individual components of the assembly are loaded but not any additional drawings or coordinate systems. Confirm your selection by clicking the "**OK**" (see Figure 6, step 4).

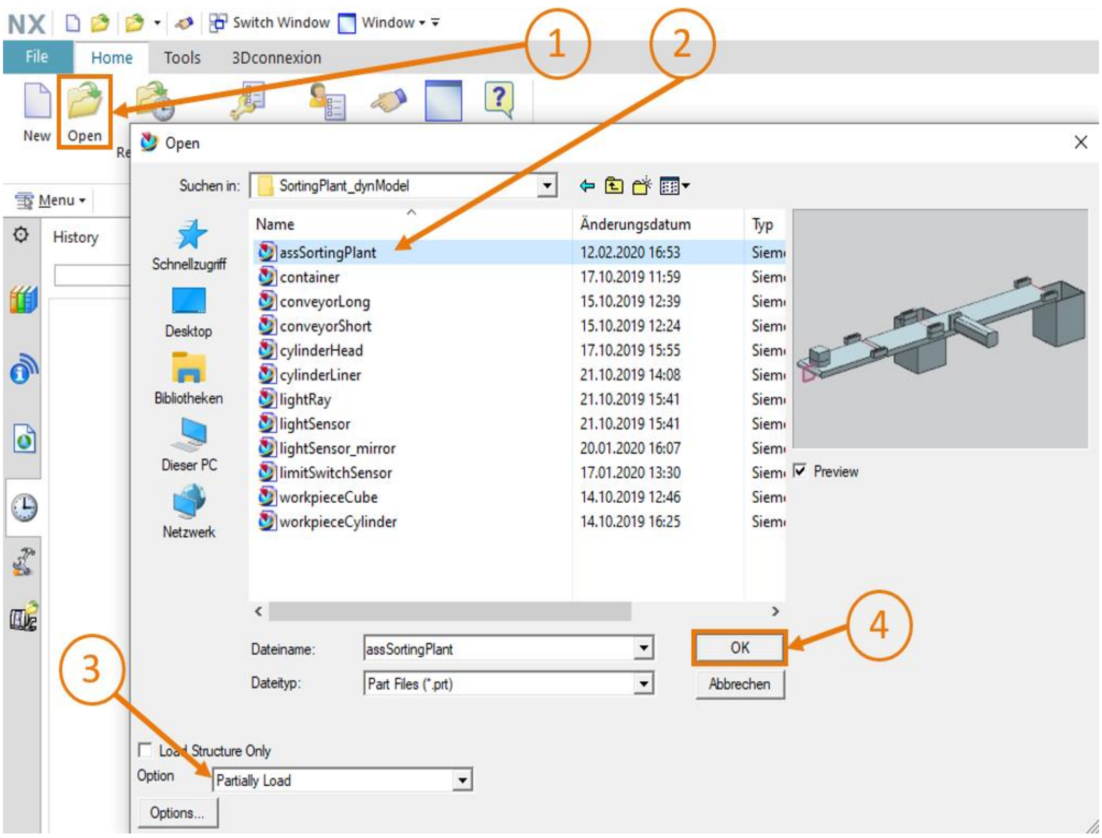


Figure 6: Opening an assembly in NX

→ Once the assembly has opened, you should see the image of the sorting plant in the three-dimensional graphics window. You can see in the header of the program that the NX "Modeling" application is still active (see Figure 7, orange frame). For dynamization of the sorting plant, you need to change to the "**Mechatronics Concept Designer**" application. Search for this add-on in the Command Finder and confirm the change of application with a click (see Figure 7, step 1).

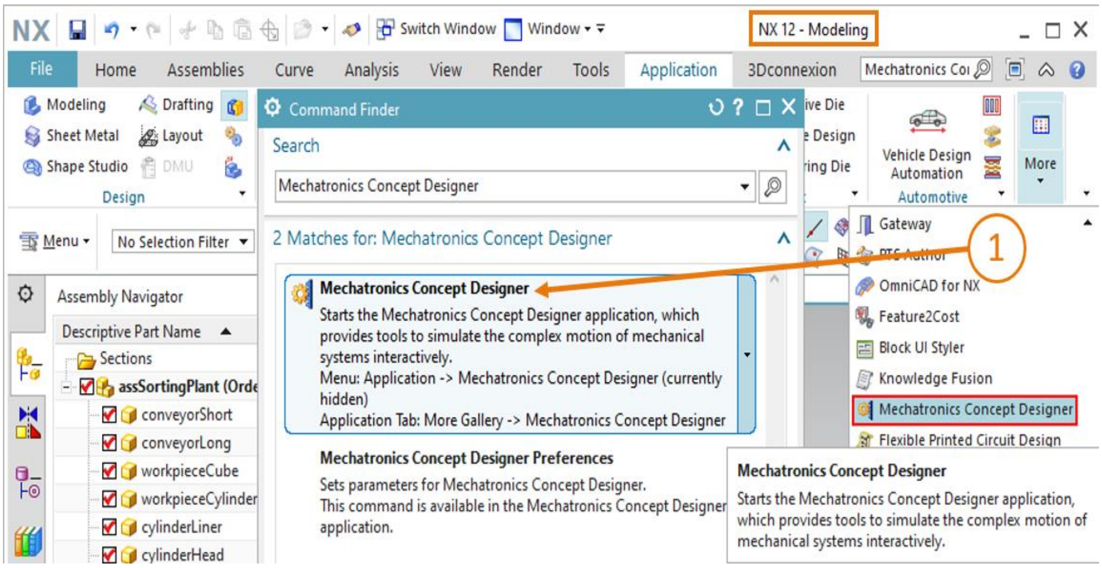


Figure 7: Opening MCD in NX

Section: Starting and stopping a simulation in MCD

→ You can then see in the header that the "Mechatronics Concept Designer" application is active. Go to the "**Home**" tab (see Figure 8, step 1). A development environment appears, which you were already introduced to in Chapter 4.2. Click the "**Play**" button in the "Simulate" menu group to start the simulation of the sorting plant (see Figure 8, step 2).

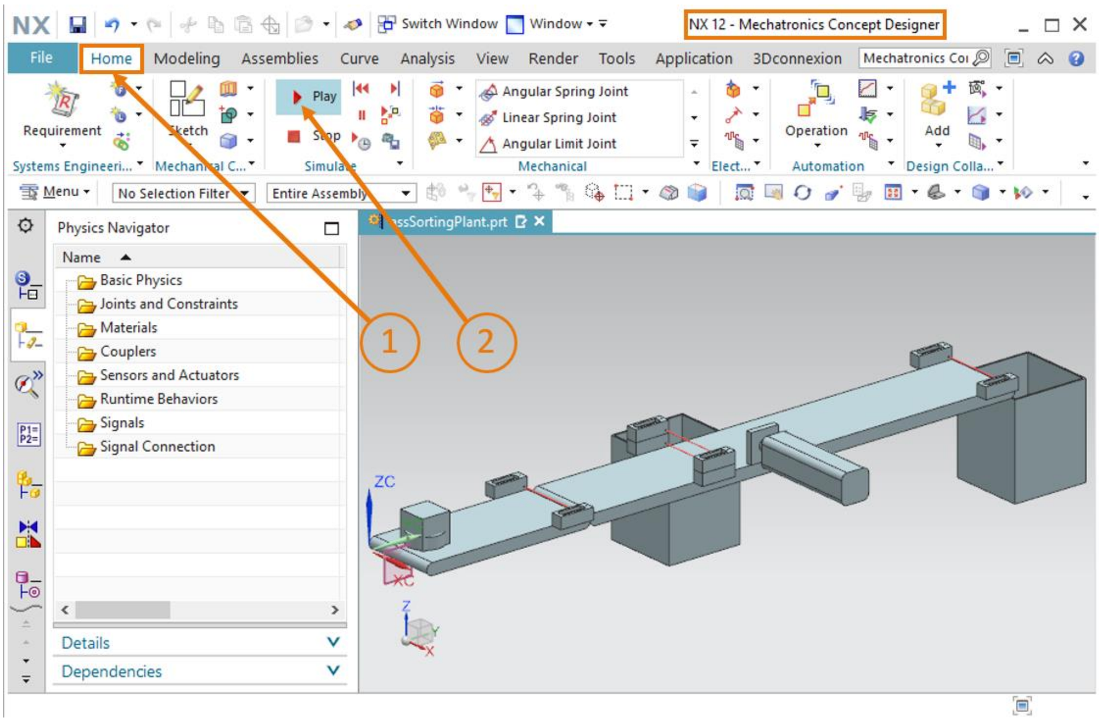


Figure 8: Starting a simulation in MCD

→ You are able to tell that a simulation is running by a display of the elapsed simulation time in the footer (see Figure 9, orange frame). You can also see in this case that the assembly in the three-dimensional graphics window does not change. Although you have opened MCD, no physical or kinematic properties have been defined yet. Stop the simulation again by clicking the "**Stop**" button  (see Figure 9, step 1).

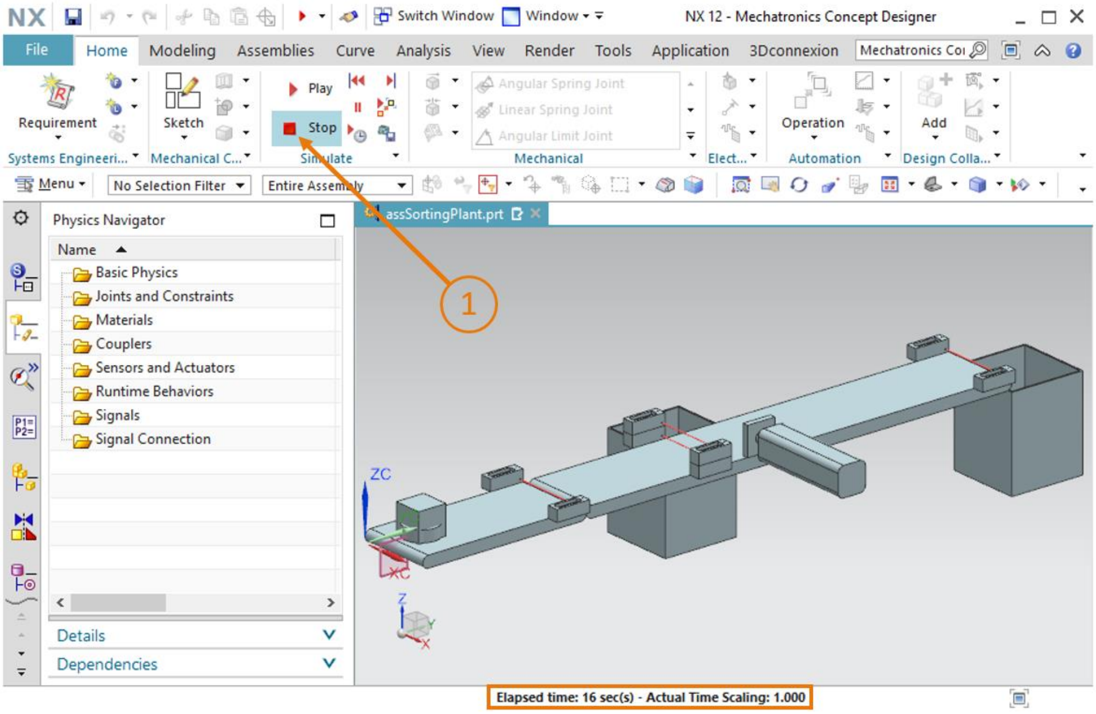
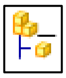


Figure 9: Stopping a simulation in MCD

7.2 Definition of rigid bodies

As the first basic physical property, you are to define your individual components as rigid bodies.

→ Start by assigning the "Rigid Body" property to the "**conveyorShort**" component. To do this, select the "**Rigid Body**" command in the "Mechanical" menu group (see Figure 10, step 1). Of course, you can also call up this command via the Command Finder. The "Rigid Body" window opens. In this window, you first must select the object that is to be defined as a rigid body. To do this, click the "**Select Object**" button in the "**Rigid Body Object**" group (see Figure 10, step 2). In the Resource bar on the left of the screen, navigate to the **Assembly Navigator**  tab. Select the "**conveyorShort**" model from the selection menu under the "assSortingPlant" assembly (see Figure 10, step 3). Leave the mass properties set to "**Automatic**" in "**Mass and Inertia**" group (see Figure 10, step 4).

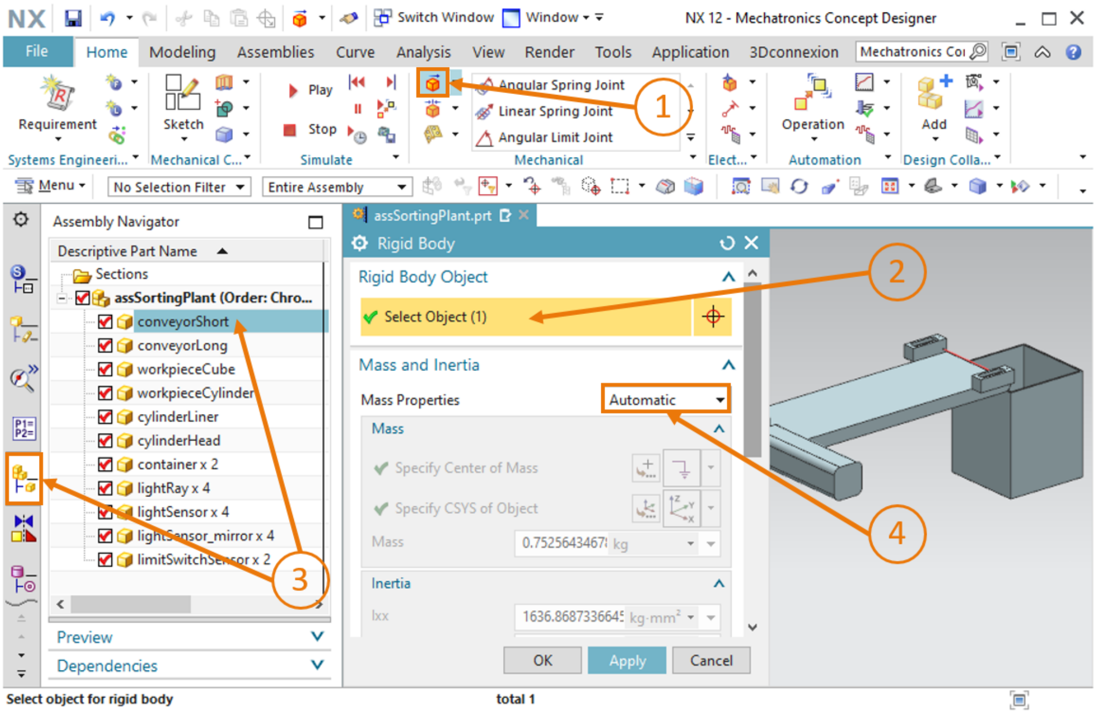


Figure 10: Creating a rigid body in MCD – Object selection, mass and inertia

→ Enter "**rbConveyorShort**" as the name (see Figure 11, step 1) and confirm your settings by clicking the "**OK**" button (see Figure 11, step 2). The prefixed abbreviation "rb" stands for "rigid body".

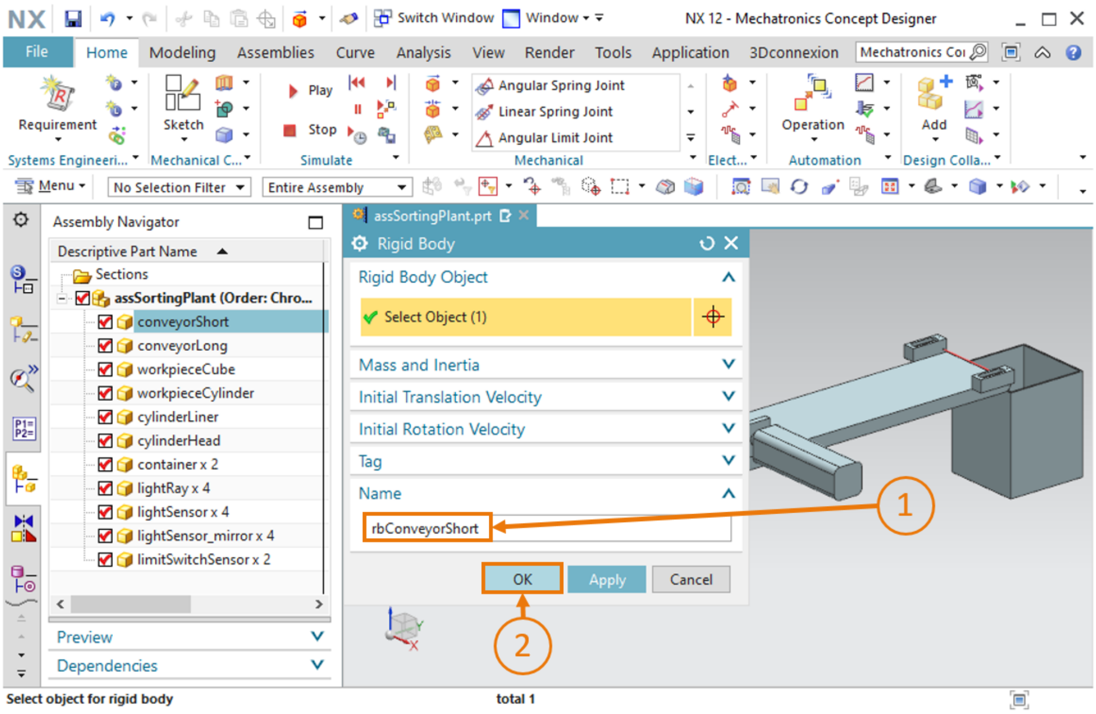


Figure 11: Creating a rigid body in MCD – Name assignment

→ Start a simulation as explained in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**". You should notice that the "conveyorShort" conveyor belt falls down when the simulation starts. By defining the conveyor belt as a rigid body, it is assigned a mass. Due to gravitational forces, the conveyor belt falls down in the simulation, as shown in Figure 12. Stop the simulation again.

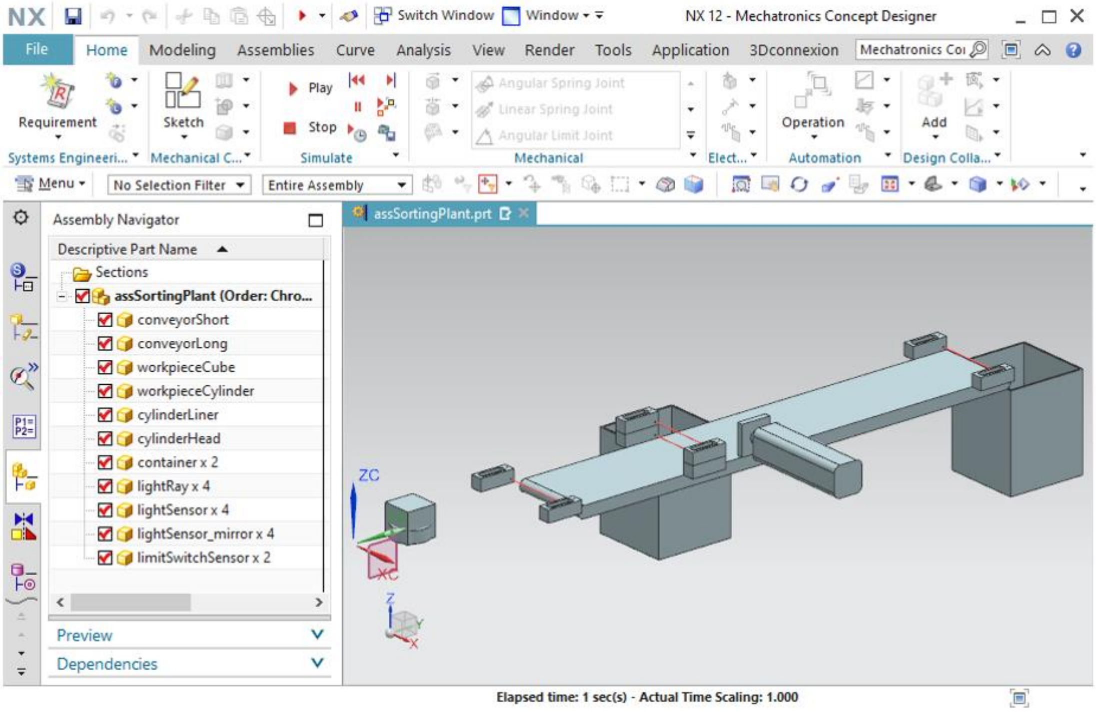
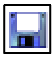


Figure 12: Simulation of a rigid body in MCD

You have now assigned your first dynamic property to the static 3D model of the sorting plant. Save your project by clicking the Save icon .

→ Follow the preceding instructions in this chapter to create additional rigid bodies for the following components:

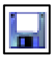
- "**conveyorLong**" as a rigid body named "**rbConveyorLong**"
- "**workpieceCube**" as a rigid body named "**rbWorkpieceCube**"
- "**workpieceCylinder**" as a rigid body named "**rbWorkpieceCylinder**"
- "**cylinderLiner**" as a rigid body named "**rbCylinderLiner**"
- "**cylinderHead**" as a rigid body named "**rbCylinderHead**"
- "**container**" as a rigid body named "**rbContainer**"

Because the light sensors in the sorting plant serve only as pure sensors and should not exert any mechanical effect on other components, the definition of these components as a rigid body is omitted. By leaving out unnecessary physical properties, the performance of your dynamic model can remain as high as possible.

NOTE

Most of the dynamic commands in NX have an "Apply" button in addition to the "<OK>" button.

- If you click on "<OK>", the last settings are applied and the corresponding command window is then closed.
- If you click on "Apply", the last settings are applied, but the window remains open.

→ Start a simulation as described in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**". All objects, except the light sensors, should have mass and thus fall out of the graphics window. Stop the simulation and save the project by clicking the Save icon .

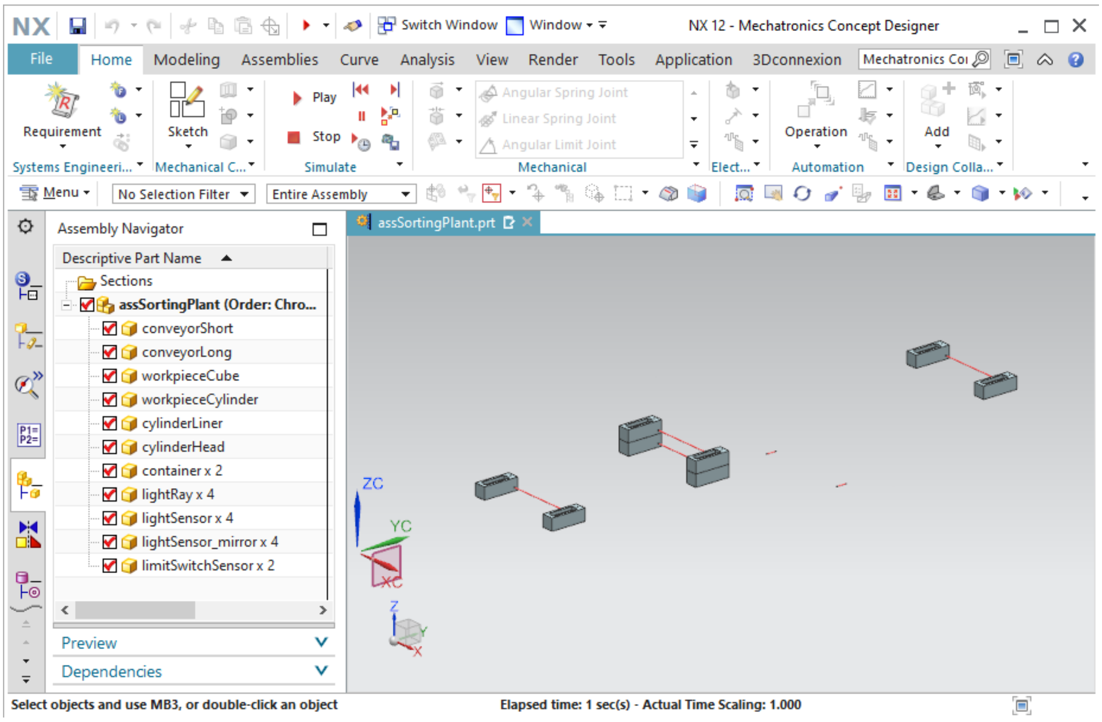


Figure 13: Simulation of all rigid bodies in MCD

NOTE

In this solution approach, a standard naming convention was observed for each dynamic property.

- The names are written in "camelCase" as stipulated in the Siemens Guide to Standardization (see Chapter 9, Link [3]).
- Each name begins with an abbreviation of the corresponding dynamic property in English (e.g. "rb" = rigid body; "ts" = transport surface).
- This abbreviation is directly followed by the primary element, which is indicated by the component name, for example, "conveyorShort".
- If relationships between various components are defined, these are to be separated by an underscore "_" (e.g. "cylinderLiner_ cylinderHead").

7.3 Specifying the fixed joints

Obviously, it is not desirable for certain bodies to fall from the conveyor belt. For this reason, the conveyor belts, the containers and the ejector must remain at one and the same position. This is possible through definition of another dynamic property: the "Fixed Joint".

The following steps must be performed to create a fixed joint.

→ Navigate to the "**Fixed Joint**" command in the "Mechanical" menu group and click on it (see Figure 14, step 1).

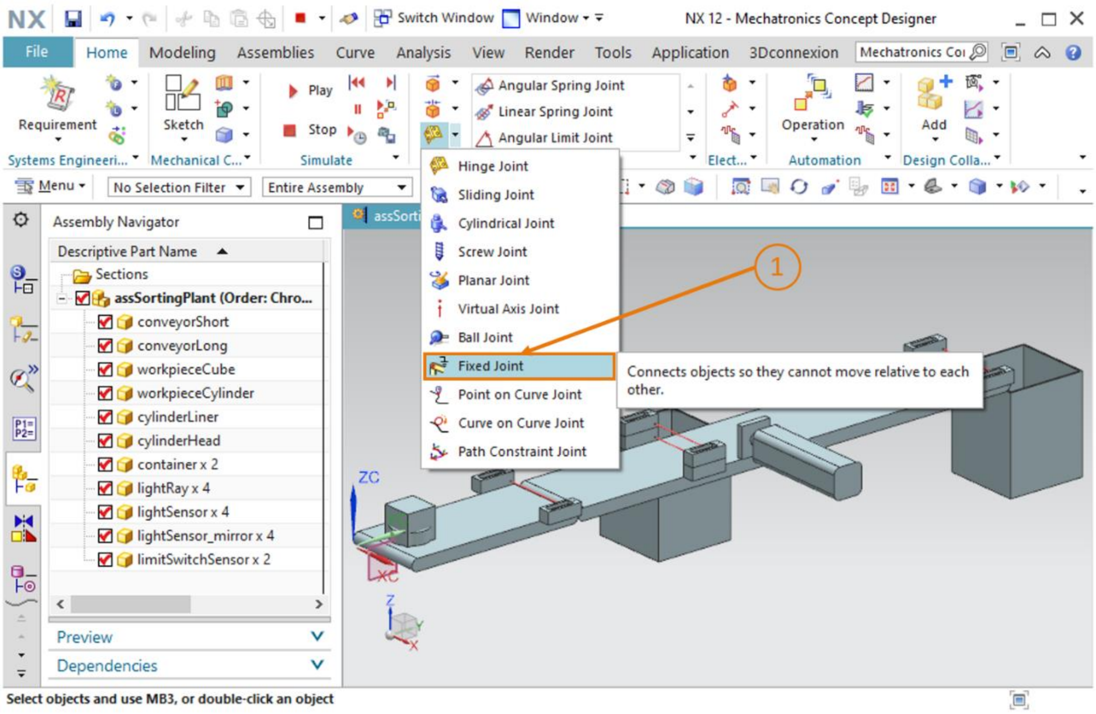


Figure 14: Creating a fixed joint in MCD – Calling up the command

→ The "Fixed Joint" window opens. At least one underlying rigid body that is fixed in space is required for this property. Click inside the "Rigid Bodies" group and select the "**Select Attachment**" button (see Figure 15, step 1). Navigate in the Resource bar to the "**Physics Navigator**"  tab and select the "**rbConveyorShort**" rigid body you created in Chapter 7.2 (see Figure 15, step 2). Then assign the name "**fjConveyorShort**" for this new property (see Figure 15, step 3) and confirm your settings by clicking the "**OK**" button (see Figure 15, step 4). The prefix "fj" stands for "fixed joint".

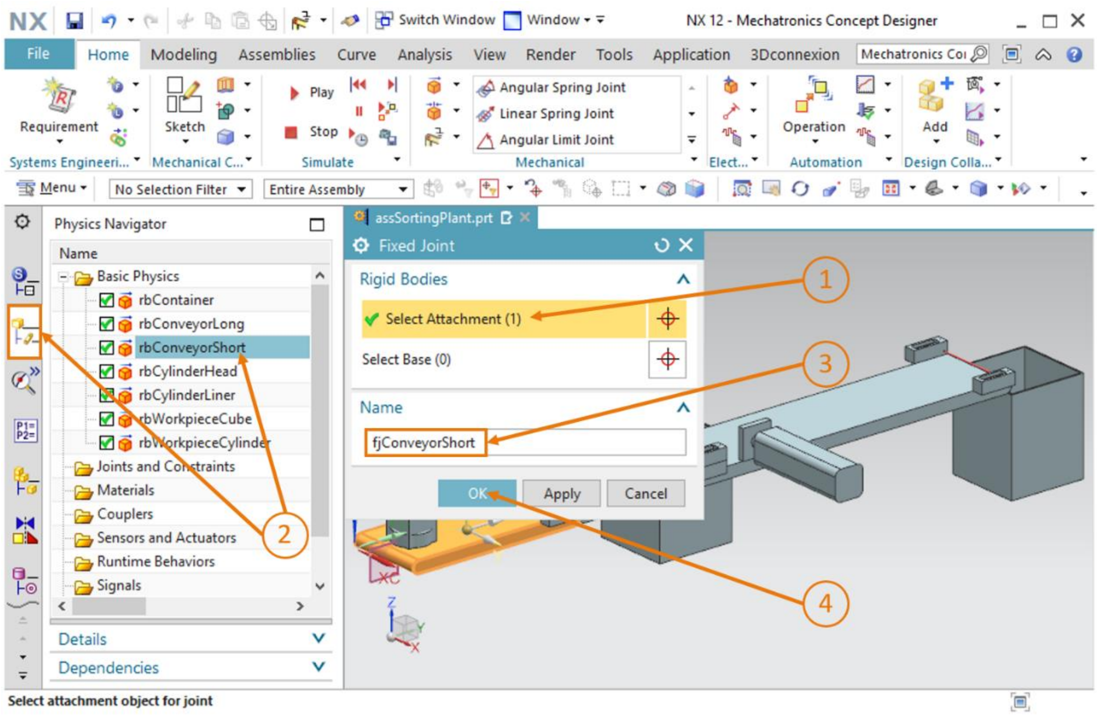
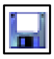


Figure 15: Creating a fixed joint in MCD – Selecting a rigid body and name

NOTE

With the "Fixed Joint" property, selecting a base means that the fixed joint only refers to the connection to the other selected rigid body. If no base is selected (as in the above case), the fixed joint links to the background.

→ Start a simulation as described in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**". You can see that the rigid body of the "conveyorShort" conveyor belt remains in the same position (see Figure 16). Stop the simulation. Save the project by clicking the Save icon .

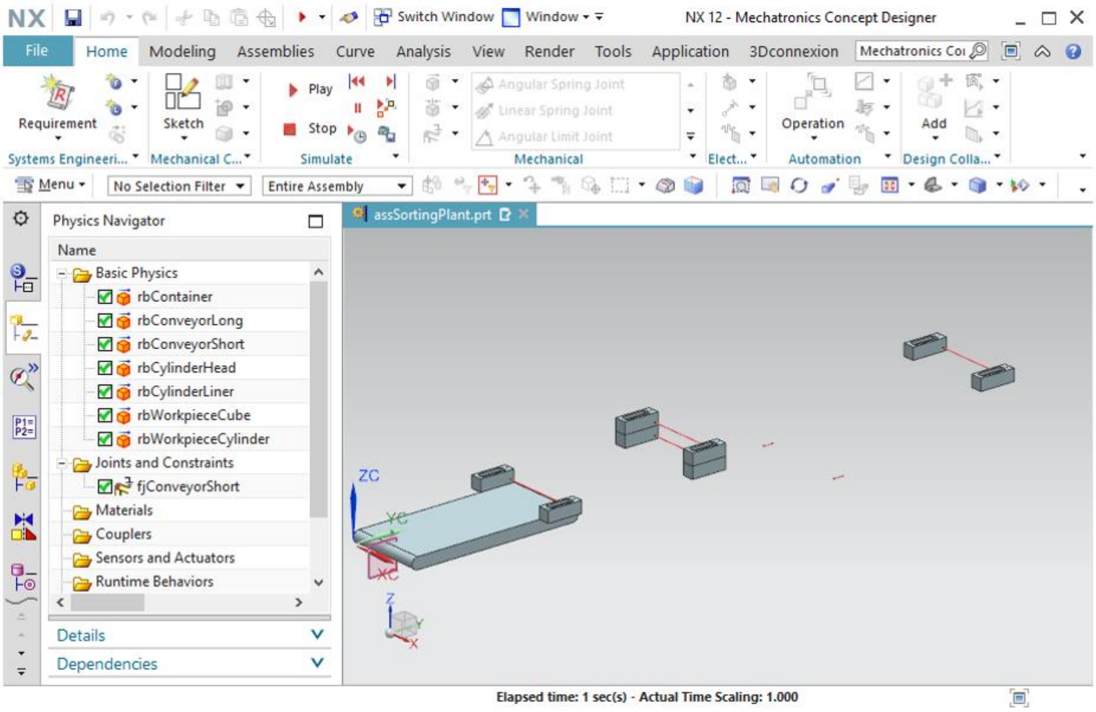


Figure 16: Simulation of a fixed joint in MCD

→ Insert the other required fixed joints in your assembly.

- For "**rbConveyorLong**" a fixed joint named "**fjConveyorLong**"
- For "**rbCylinderLiner**" a fixed joint named "**fjCylinderLiner**"
- For "**rbContainer**" a fixed joint named "**fjContainer**"

The two workpieces and the ejector head are to remain movable parts, so these models are not assigned a fixed joint.

→ Start a simulation again as explained in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**". The two conveyor belts, both containers and the ejector base remain fixed at their position (see Figure 17). Stop the simulation and save the project by clicking the Save icon .

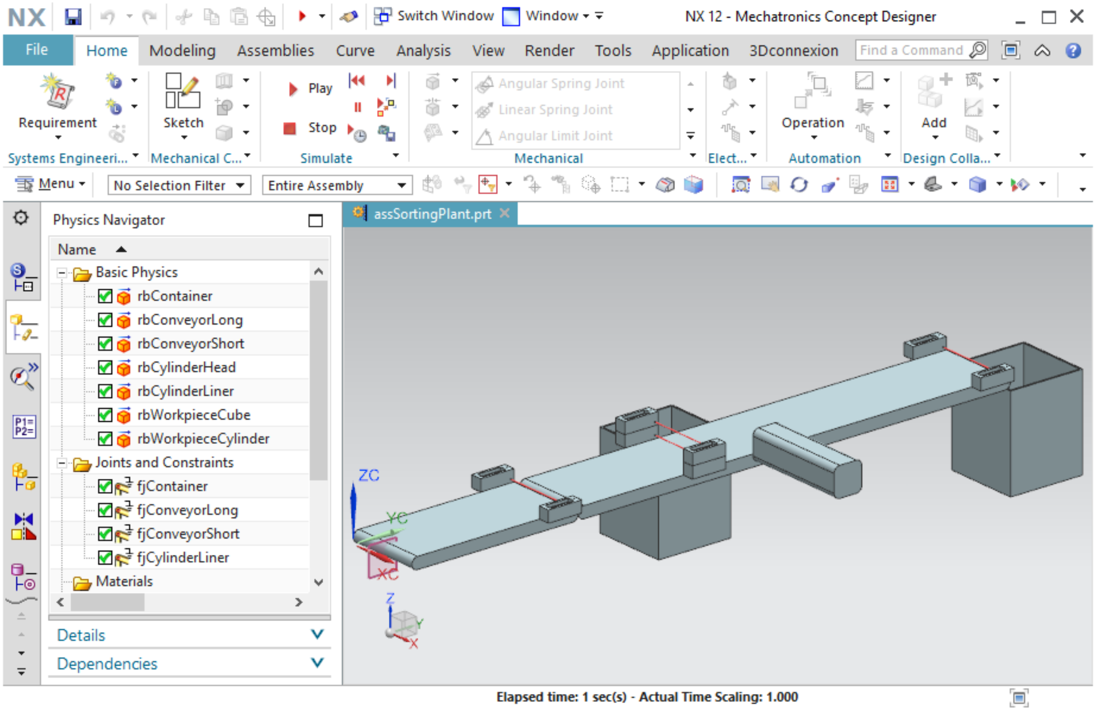


Figure 17: Simulation of all fixed joints in MCD

7.4 Assignment of collision surfaces through collision bodies

In the current intermediate status of the assembly, interactions between the various models have not yet been defined. The most basic and important interaction property in MCD is the collision body. A collision surface on a collision body results in it being able to react appropriately to another collision surface. Most of the time, the reaction is to repel it. The following chapters will describe the creation of the collision bodies required for the sorting plant.

7.4.1 Creating a collision body for workpieceCube

To create a collision body for "workpieceCube", proceed as follows:

Section: Displaying/hiding components and assemblies

→ First, hide all components except the "workpieceCube" component. Go the to "**Assembly Navigator**" tab in the Resource bar (see Figure 18, step 1). Click on the **red check mark** in front of the "**assSortingPlant**" assembly in order to hide all models in the graphics window (see Figure 18, step 2). Now, every component should have a grayed-out check mark in front of it, and no bodies should be displayed in the three-dimensional graphics window. Activate the view of the "workpieceCube" workpiece by clicking on the **gray check mark** of this individual component (see Figure 18, step 3). The check mark should turn red and the selected workpiece should appear as the only model in the graphics window. Change to the trimetric view, which allows you to see the body in its entirety, as shown in Figure 18, step 4.

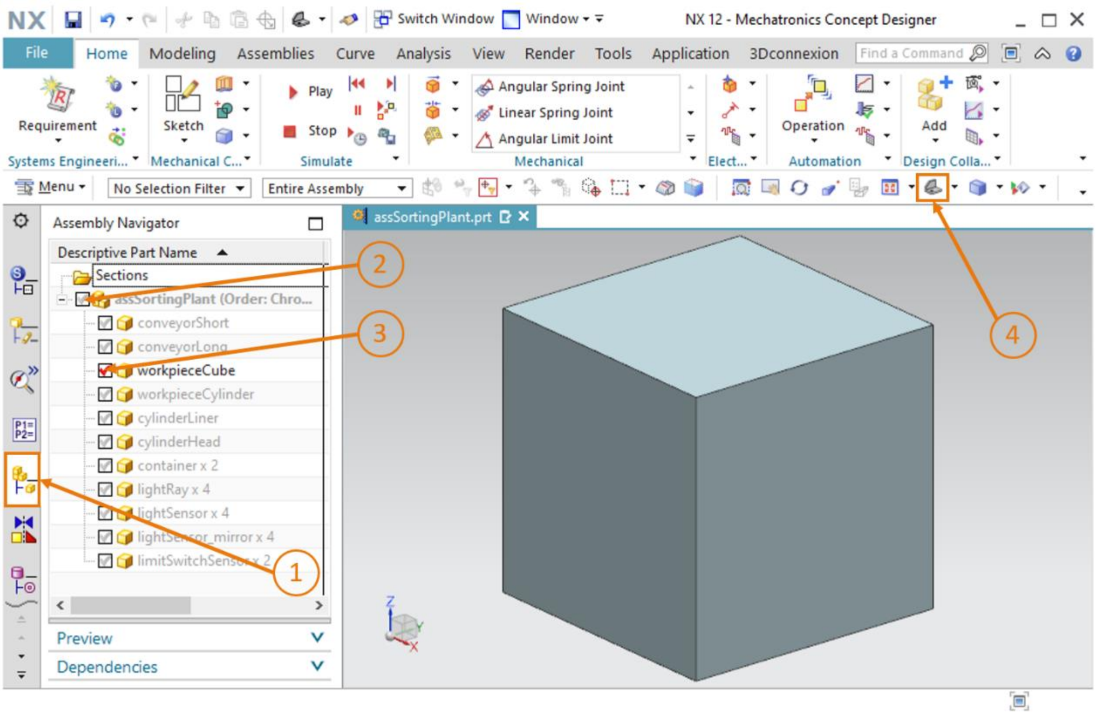


Figure 18: Hiding all components and displaying a single component

→ Select the "**Collision Body**" command from the "Mechanical" menu group or via the Command Finder (see Figure 19, step 1). The "Collision Body" window opens. In the first step, you need to select all objects that are to represent the collision body. For example, these can be various surfaces of a body. To do this, click the "**Select Object**" button in the "Collision Body Object" group (see Figure 19, step 2). Navigate in the three-dimensional graphics window to the first surface of the body (see Figure 19, step 3).

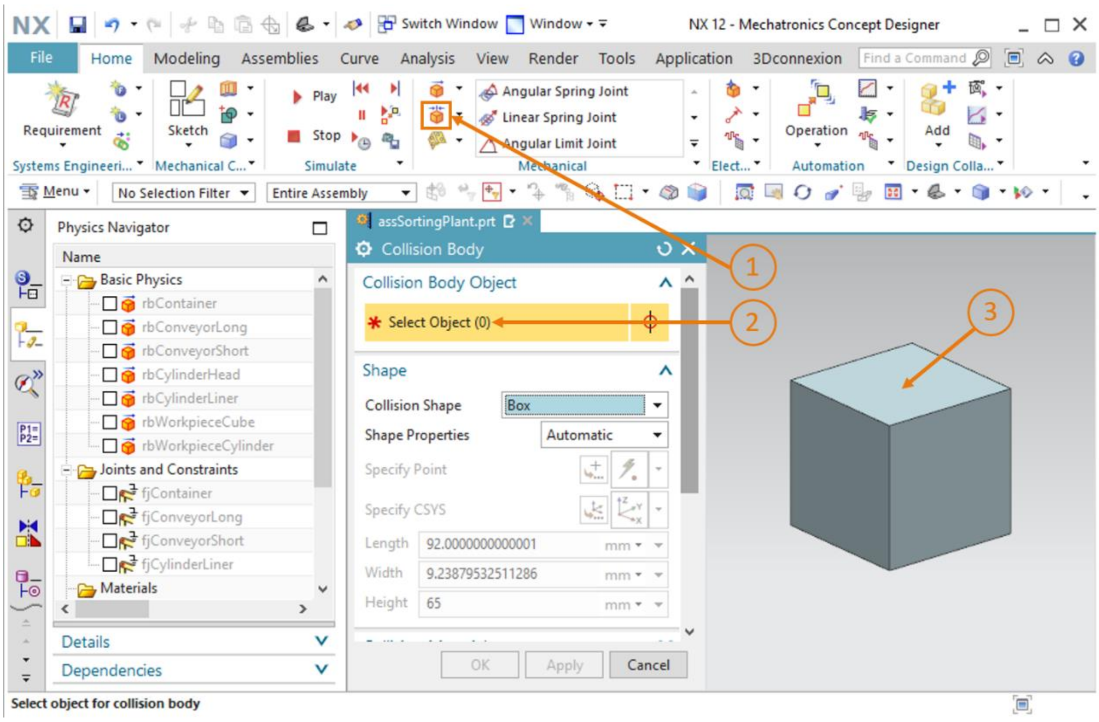


Figure 19: Creating the collision body for workpieceCube – Selecting collision objects

→ If your mouse is not pointing to a part of the body, the part is displayed in the gray color typical of NX (see Figure 20, on left). When the mouse pointer lands on a surface, the surface turns red (see Figure 20, at center). Click on this surface. The selected surface is then displayed in orange (see Figure 20, on right).

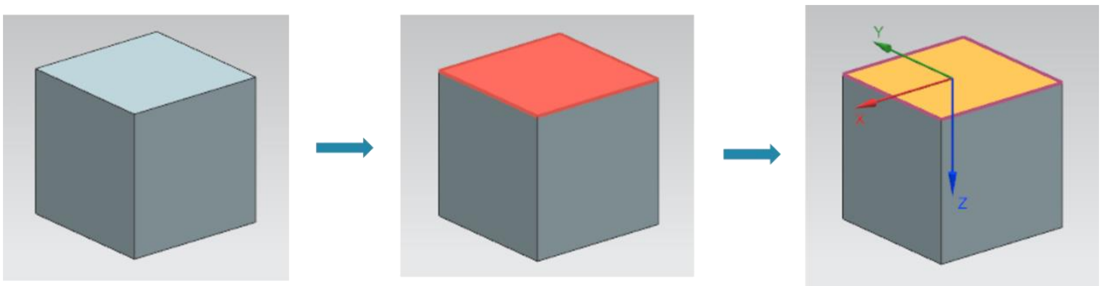


Figure 20: Selecting a surface in MCD

Section: Rotating a model in MCD

→ Select the two other visible surfaces of the cuboid (see Figure 21, step 1). You should have a total of three surfaces (see parenthesized information in the "Select Object" button). In order to also see the remaining surfaces of the body, you need to change the view. To do this, click the "**Rotate**" button  to rotate the model (see Figure 21, step 2).

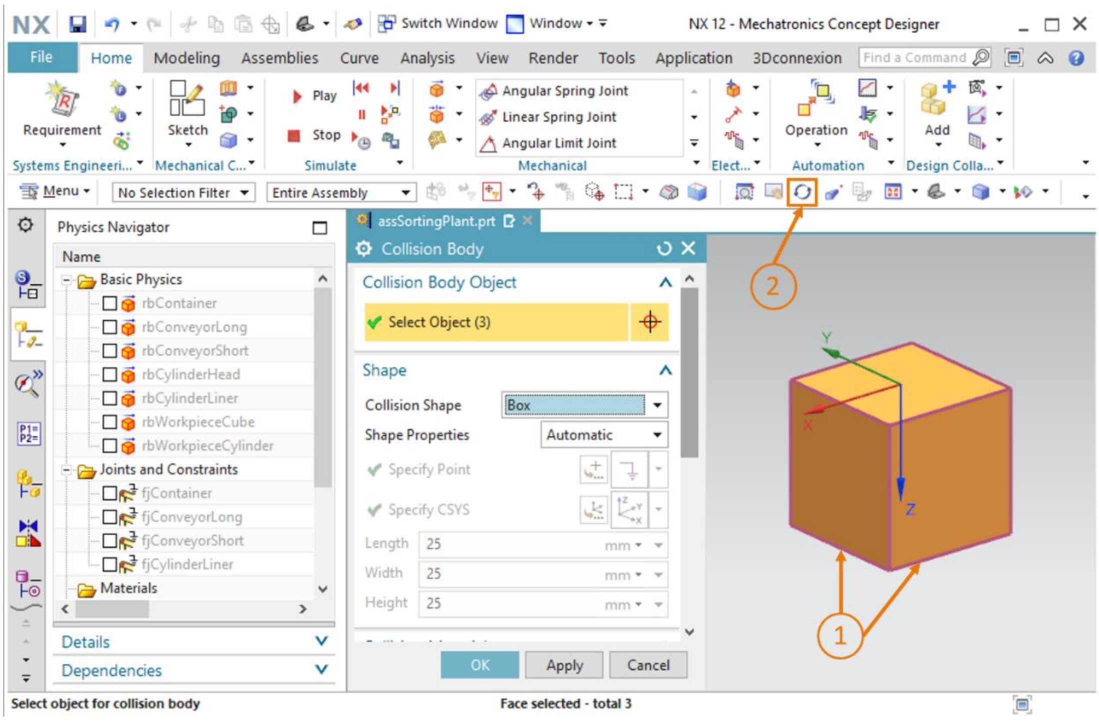


Figure 21: Creating the collision body for workpieceCube – Selection of additional surfaces

→ Now, rotate the body by pressing and holding down the left mouse button in the middle of the graphics window and dragging downward (see Figure 22, step 1). After a while you can see the three non-selected surfaces, as shown in Figure 22. Exit Rotate mode by clicking the "Rotate" button again (see Figure 22, step 2). Select the remaining three surfaces shown in Figure 22, step 3. Then, switch back to the trimetric view (see Figure 22, step 4).

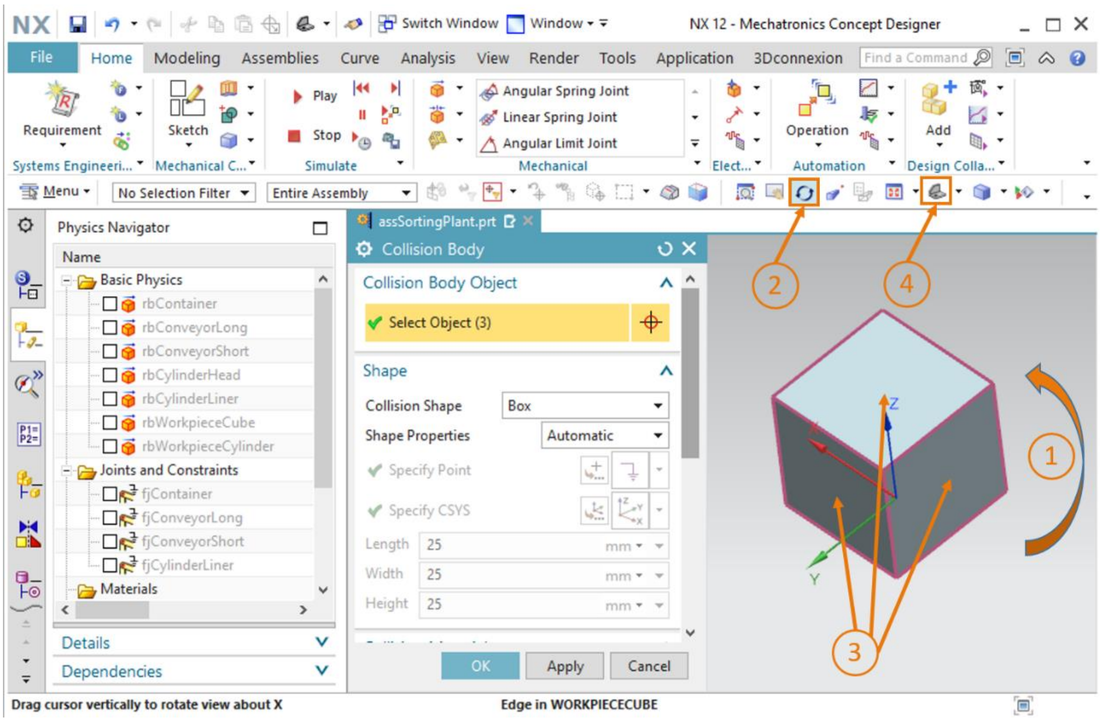


Figure 22: Creating the collision body for workpieceCube – Rotating the view and selecting the remaining collision objects

→ Various collision shapes can be selected in the "Shape" group of the "Collision Body" window. You can find an explanation of these in Chapter 4.2.1. Select "**Box**" here as the collision shape for the cube, because MCD can simulate the collision body with this shape with only a minor loss in performance (see Figure 23, step 1).

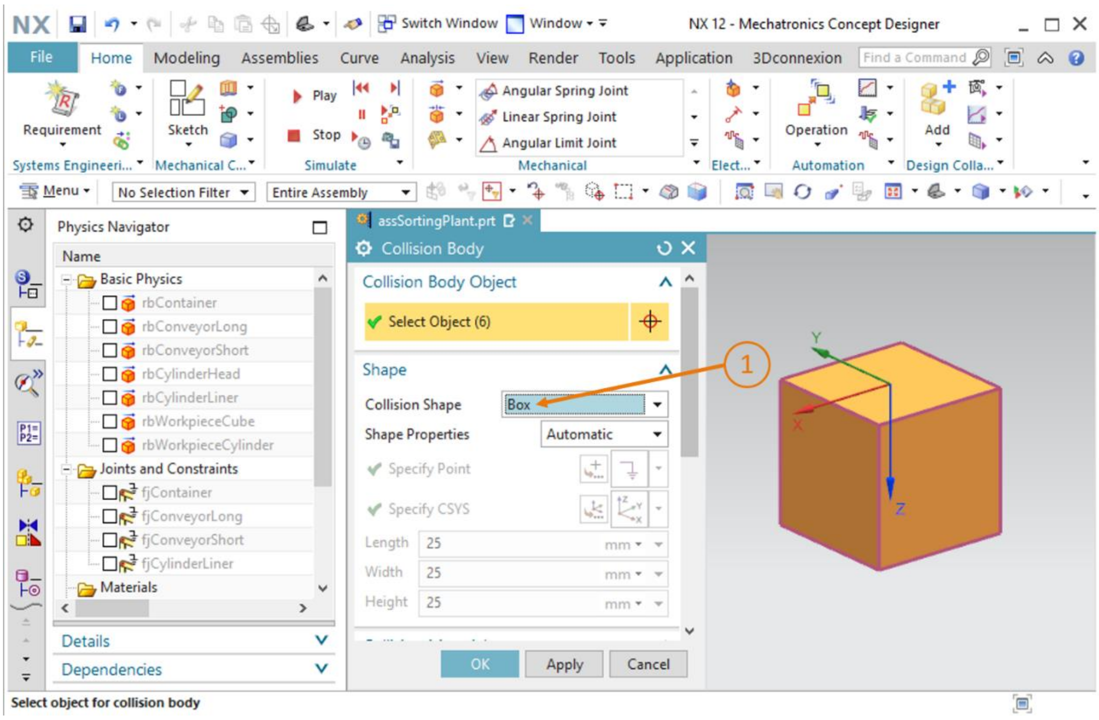


Figure 23: Creating the collision body for workpieceCube – Defining the collision shape

→ Scroll down in the command window to see additional groups. Leave "**Default Material**" as the selected material in the "**Collision Material**" group (see Figure 24, step 1). The **category** that is specified in the "Category" group remains set to the value "**0**" (see Figure 24, step 2). In the "Collision Setting" group, ensure that the "Highlight on Collision" and "Stick when Collision" check boxes are **cleared** (see Figure 24, step 3). Once you assign the name "**cbWorkpieceCube**", as shown in Figure 24, step 4, you can finish the creation of the collision body by clicking the "OK" button (see Figure 24, step 5). The prefix "cb" stands for "collision body".

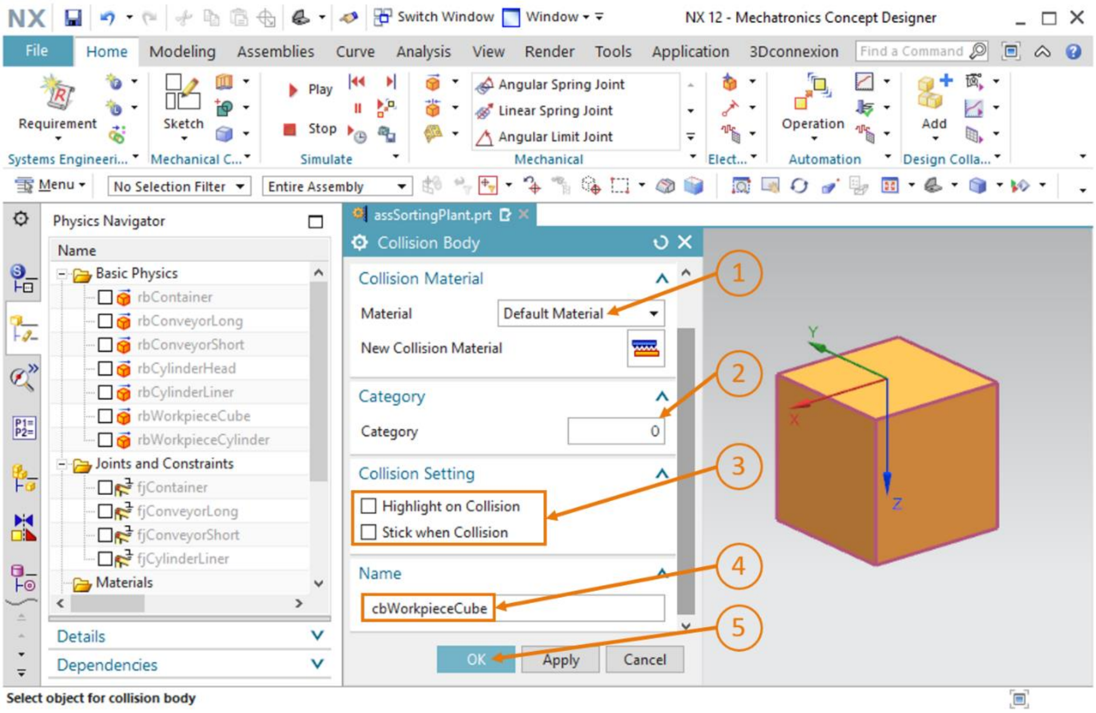


Figure 24: Creating the collision body for workpieceCube – Specifying additional settings and the name

→ As described previously in "**Section: Hiding/displaying components and assemblies**", go to the "Assembly Navigator" tab in the Resource bar and activate the "**assSortingPlant**" assembly by clicking on the gray check mark (see Figure 25, steps 1 + 2). Then, switch back to the trimetric view so that you see all of your model again (see Figure 25, step 3).

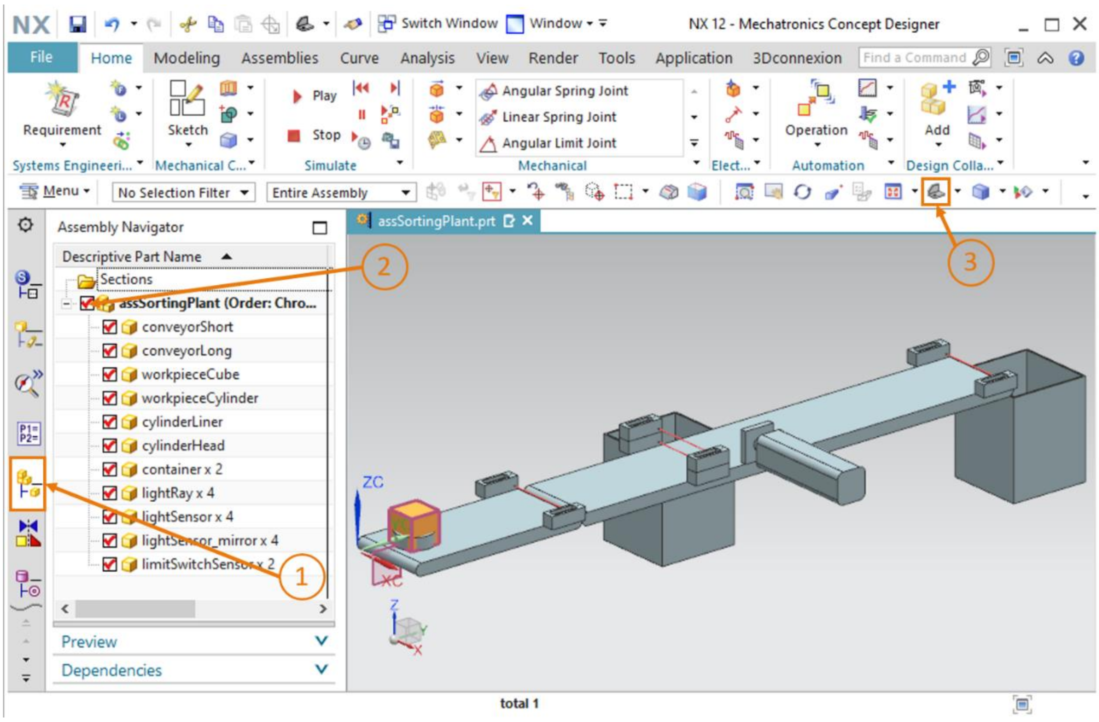
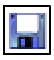


Figure 25: Creating the collision body for workpieceCube – Displaying the assembly

You have now created your first collision body. Save your assembly by clicking the Save button .

7.4.2 Creating a collision body for workpieceCylinder

You can use a procedure similar to the one described in Chapter 7.4.1 to create a collision body for "workpieceCylinder".

→ First, all components except "workpieceCylinder" must be hidden. Use the procedure described in Chapter 7.4.1, "**Section: Displaying/hiding components and assemblies**" for this.

→ You must then select the "**Collision Body**" command. Select all surfaces of the "workpieceCylinder" model as collision body objects following the same principle as described in Chapter 7.4.1. To rotate the component, follow the procedure described in Chapter 7.4.1, "**Section: Rotating a model in MCD**". You should have a total of **three surfaces**.

→ Because this is a cylindrical workpiece, "**Cylinder**" must be selected as the collision shape (see Figure 26, step 1).

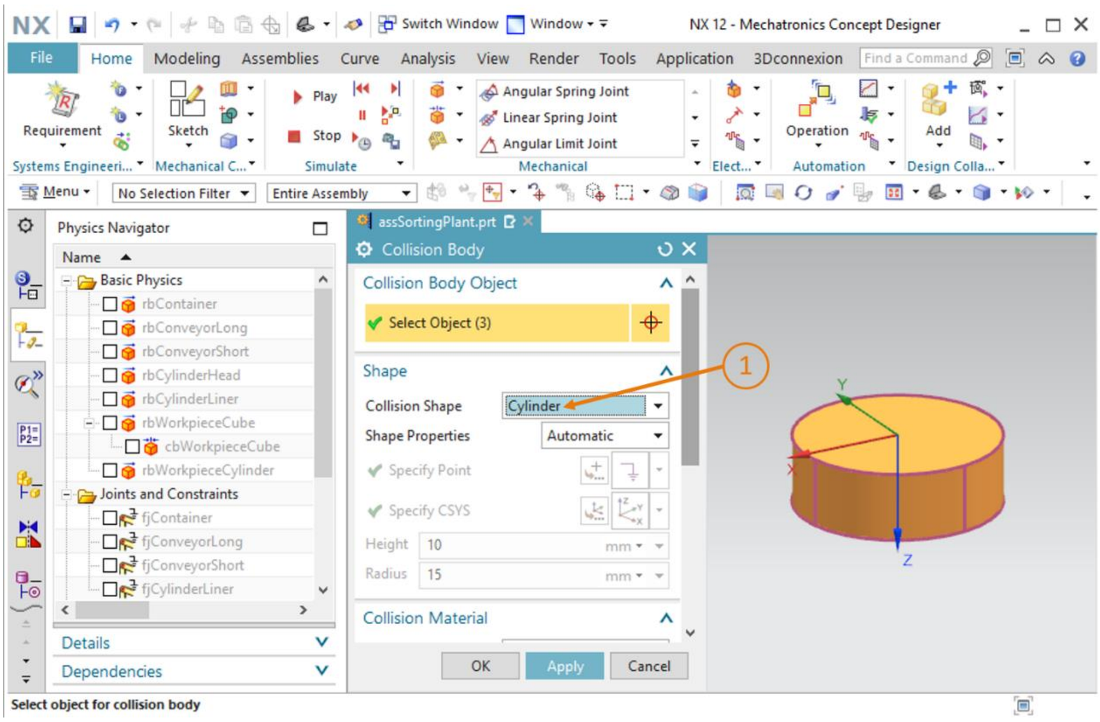


Figure 26: Creating the collision body for workpieceCylinder

→ Follow the procedure explained in Chapter 7.4.1 for the remaining settings. However, assign the name "**cbWorkpieceCylinder**" for the collision body.

→ Finish by displaying the entire assembly again as described in Chapter 7.4.1, "**Section: Displaying/hiding components and assemblies**" and switching to the trimetric view. Save the sorting plant by clicking the Save icon .

7.4.3 Creating collision bodies for conveyorShort

In this chapter, you are to create the collision bodies for the "conveyorShort" transport surface. Compared to the workpieces in Chapters 7.4.1 and 7.4.2, this component is not a simple geometrical body and multiple collision bodies must therefore be created for this model. More than one collision body can thus exist for any model.

→ First, you should hide all models of the assembly except for "conveyorShort" as described in Chapter 7.4.1, "**Section: Displaying/hiding components and assemblies**". Also change the display of the model to "**Top**" view .

Defining a collision body for the planar transport surface:

→ Begin by defining a collision body for the planar transport surface. The procedure is very similar to that described in Chapter 7.4.1. Open the "**Collision Body**" command. Navigate in the "Collision Body" command window to the "Collision Body Object" group and click the "**Select Object**" button (see Figure 27, step 1). Select the **top planar transport surface** in the three-dimensional graphics window (see Figure 27, step 2). Select "**Box**" as the collision shape, as shown in Figure 27, step 3. Leave the default values for the remaining settings as described in Chapter 7.4.1. Finish by assigning the name "**cbConveyorShortPlane**" for this collision body and confirm the settings by clicking the "**OK**" button.

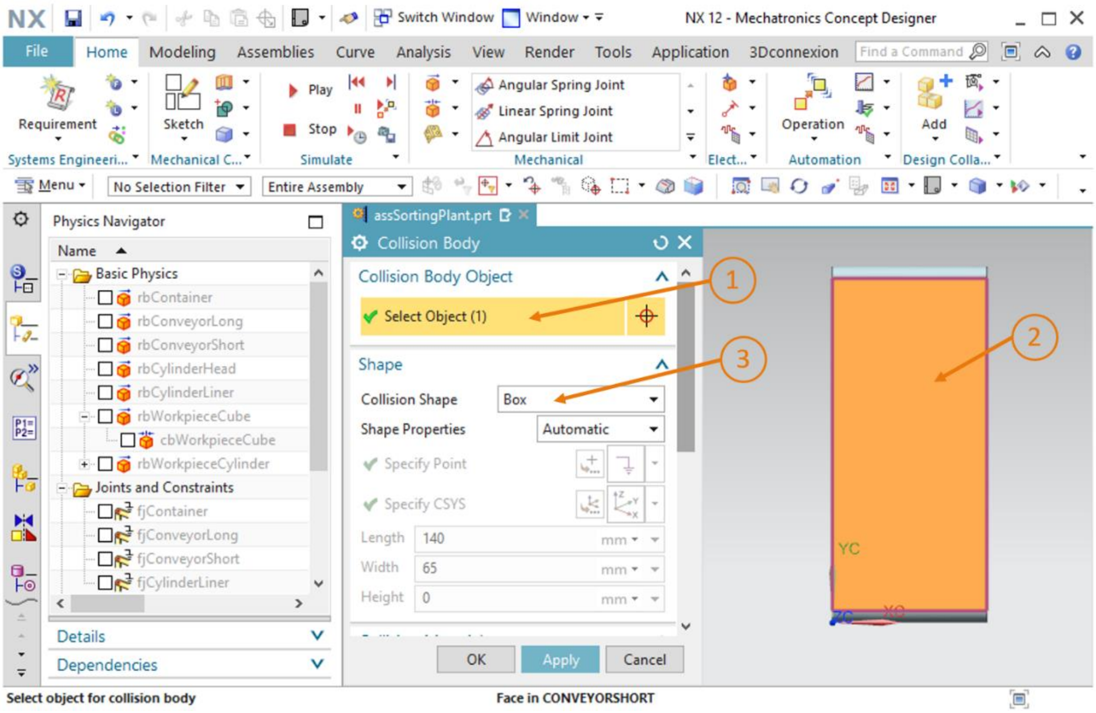


Figure 27: Creating a collision body for the planar surface of conveyorShort

Defining collision bodies for rollers of the conveyor belt:

Other collision bodies are the rollers at the front and rear ends of the conveyor belt, as illustrated in Figure 28. These end rollers are basically cylindrical in shape. Note that a separate collision body must be created for each end.

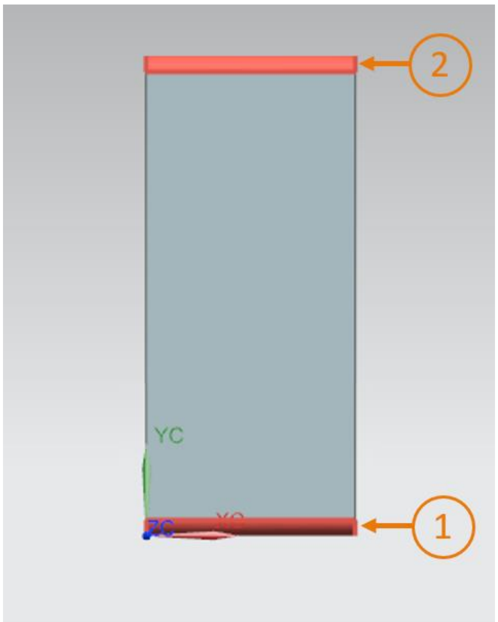
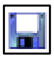


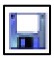
Figure 28: Conveyor belt with end rollers highlighted in red

→ Start with the front-end roller. Open the "**Collision Body**" command window again for this. Select the **front end roller** as the object (see Figure 28, step 1) and "**Cylinder**" as the collision shape. Name this collision body "**cbConveyorShortStart**" and finish the creation of the collision body.

→ Now continue with the rear end roller. Open the "**Collision Body**" command. Select the **rear end roller** as the object (see Figure 28, step 2) and "**Cylinder**" as the collision shape. Name this collision body "**cbConveyorShortEnd**" and finish the creation of the collision body.

→ You have now created a total of three collision bodies for this conveyor belt. Follow the procedure in Chapter 7.4.1, "**Section: Displaying/hiding components and assemblies**" to display the entire assembly again. Switch back to the trimetric view. Save your project by clicking the Save icon .

7.4.4 Creating collision bodies for conveyorLong

As already explained for conveyorShort in Chapter 7.4.3, the "conveyorLong" component is also to consist of three collision bodies: the planar transport surface and the two end rollers. Use the same procedure you used in Chapter 7.4.3. Display only conveyorLong in the three-dimensional graphics window. As names of the collision bodies, use "**cbConveyorLongPlane**" for the planar transport surface and "**cbConveyorLongStart**" and "**cbConveyorLongEnd**" for the two end rollers. Finally, switch back to the trimetric view and save your project by clicking the Save icon .

7.4.5 Creating collision bodies for the ejector head

The ejector head consists of two assembled bodies, namely a cuboid and a cylinder. For this reason, two collision bodies that can take on simple geometrical shapes are required for this component.

Proceed as follows for this:

→ Hide all models of the assembly except the "cylinderHead" component. Make use of the descriptions in Chapter 7.4.1, "**Section: Displaying/hiding components and assemblies**" for this.

→ In the first step, create a collision body for the arm, that is, the cuboid of the ejector head that will eject the workpieces. Open the "**Collision Body**" command. Select all six surfaces of the cuboid as collision body objects (see Figure 29, steps 1 + 2). Make use of the ability to rotate the object, as explained in Chapter 7.4.1, "**Section: Rotating a model in MCD**". Select "**Box**" as the collision shape, as shown in Figure 29, step 3. All other settings are to be chosen exactly the same as in the preceding chapters. Specify "**cbCylinderHead Workpiece**" as the name. Confirm your inputs by clicking the "**OK**" button.

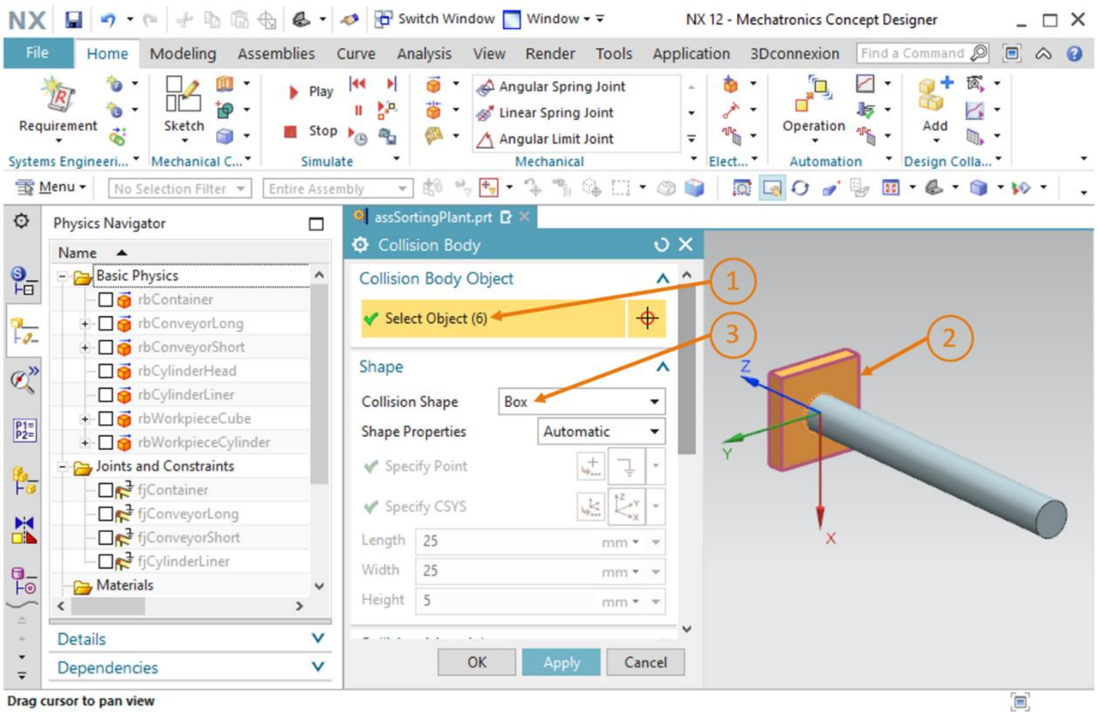


Figure 29: Creating the collision body for the arm of the ejector head

→ Also create a collision body for the cylindrical guide of the ejector head, because a collision of the guide with a workpiece is possible in principle. Open the "Collision Body" command for this, and select the **cylindrical surface** as collision object (see Figure 30, steps 1 + 2). Assign the collision shape "**Cylinder**" to this body (see Figure 30, step 3) and assign the name "**cbCylinderHeadLiner**". Confirm these settings by clicking the "**OK**" button.

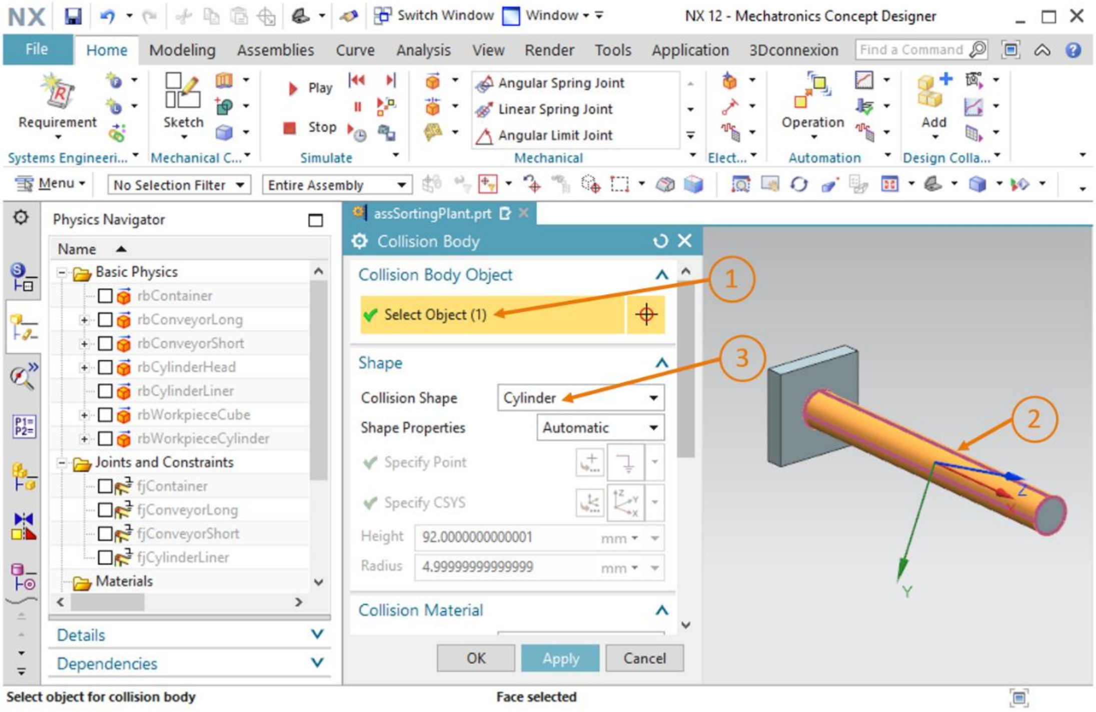
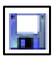


Figure 30: Creating the collision body for the guide cylinder of the ejector head

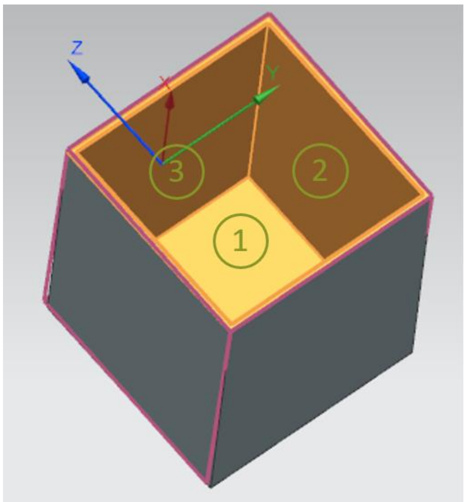
→ The collision bodies for the ejector head are now defined. Finish by displaying all components of the assembly again as described in Chapter 7.4.1, "**Section: Displaying/hiding components and assemblies**". Switch back to the trimetric view and save your project by clicking the Save icon .

7.4.6 Creating collision bodies for the containers

The two containers also need collision surfaces for containing the sorted-out workpieces. These are limited to the inside of both bodies. Proceed as follows to create the two collision bodies:

→ Apart from the container, all other parts of the assembly are to be hidden. Make use of the explanations in Chapter 7.4.1, "**Section: Displaying/hiding components and assemblies**" for this. You then see the two containers of your assembly in the three-dimensional graphics window.

As mentioned previously, the sole purpose of this model is to hold the bodies inside the container. Therefore, you only have to select the inside surfaces of the container that can come into contact with the workpieces. Figure 31 shows that six surfaces must be selected for this. Five surfaces represent the inside of the container (see Figure 31, surfaces 1 – 5), while one surface forms the top frame (see Figure 31, surface 6).



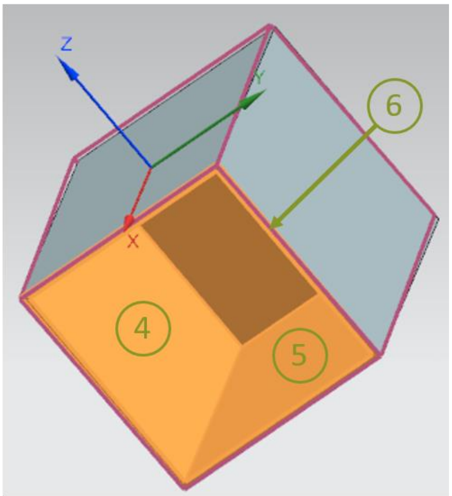


Figure 31: Collision surfaces of the containers from different viewing angles

→ First, create a collision body for the first container into which the "workpieceCylinder" workpieces will be sorted out by the ejector. To do this, open the "Collision Body" command. Once you have clicked the "**Select Objects**" button in the command window, select the **six surfaces**, as shown in Figure 31 (see Figure 32, steps 1 + 2). Select "**Mesh**" as the collision shape and a convex factor of "**1.00**", as specified in Figure 32, step 3. Use of the mesh collision shape is required here because you want to represent an interior body that cannot be represented with a simple geometrical shape. However, this collision shape needs more computing capacity during a simulation than simple shapes. Specify "**cbContainerCylinder**" as the name and confirm the settings by clicking the "**OK**" button.

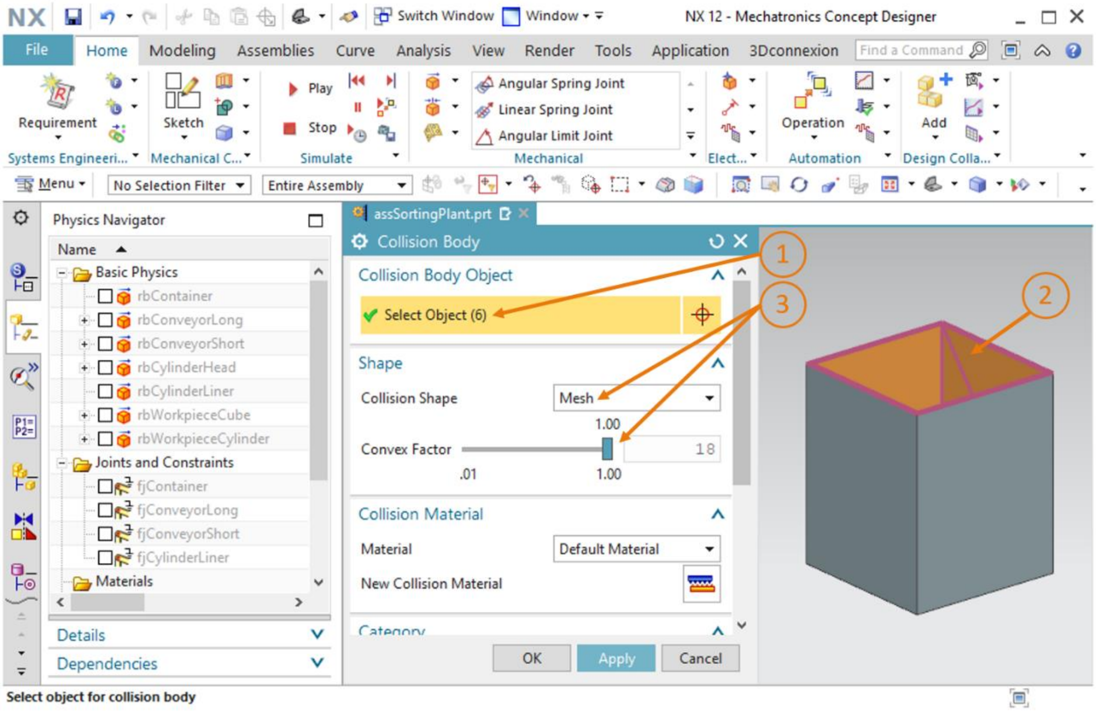
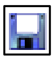


Figure 32: Creating a collision body for a container

→ Follow the same procedure to define the collision body of the second container as for the first container. Use "**cbContainerCube**" as the name for this collision body and exit the settings by clicking the "**OK**" button.

→ After finishing this collision body, you can display the entire assembly again as described in Chapter 7.4.1, "**Section: Displaying/hiding components and assemblies**". Switch to the trimetric view and use  to save your project.

→ You have now created all the collision bodies required for the sorting plant. Check the behavior of your assembly by starting a simulation. Follow the instructions in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**" for this. You should see that both workpieces stay on the short conveyor belt (see Figure 33). By defining the collision surfaces, the surfaces interact with one another and repel each other. However, the ejector head still falls out of the window.

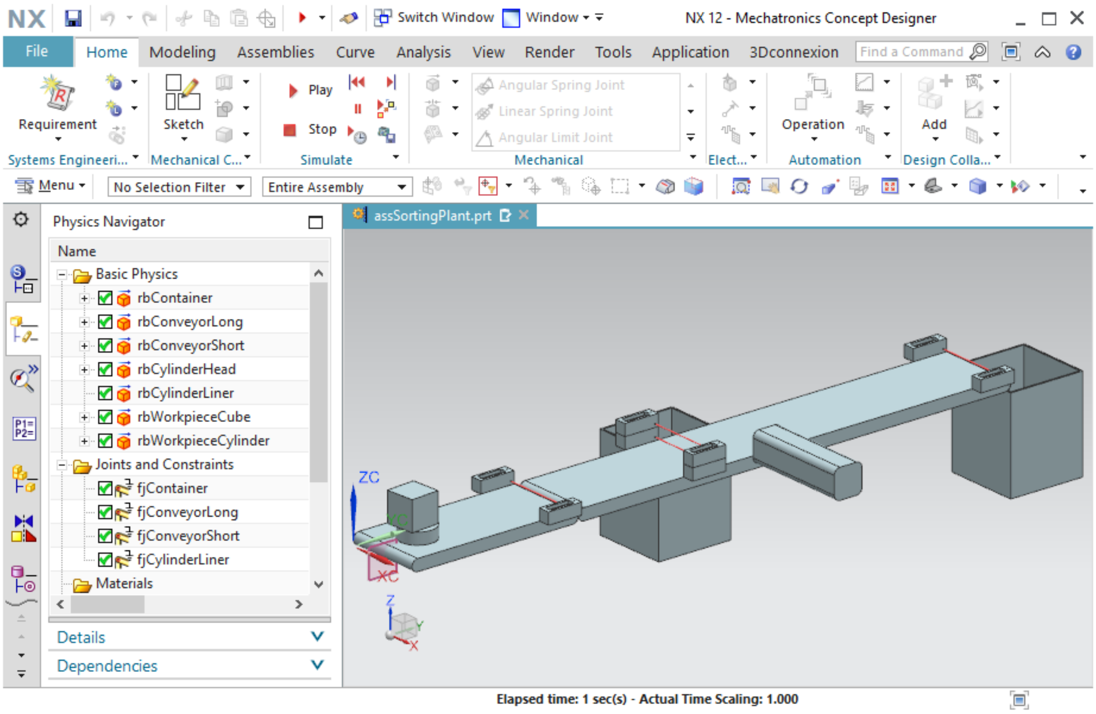
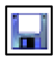


Figure 33: Simulation of collision bodies in MCD

Stop the simulation and save your entire project by clicking the "Save" button  in the menu bar.

7.5 Definition of a sliding joint for the ejector

To prevent the ejector head from falling down and to use it in accordance with its function for ejecting workpieces, you should define the head of the ejector as a sliding joint. This allows the movement of a rigid body along a vector.

Perform the following steps for creating a sliding joint:

→ Search for the "**Sliding Joint**" command in the "Mechanical" menu group or via the Command Finder. Click the corresponding button to open the "Sliding Joint" command window (see Figure 34, step 1). Here, you first must select two rigid bodies in the Rigid Bodies group.

- The attachment selects the rigid body that is to move along a defined vector.
- The base represents the rigid body to which the attachment is linked.

Consequently, the rigid body of the head of the ejector "**rbCylinderHead**" must be selected as the **attachment** (see Figure 34, steps 2 + 3). The rigid body of the guide cylinder of the ejector "**rbCylinderLiner**" acts as the **base** (see Figure 34, steps 4 + 5). You can select both rigid bodies on the "Physics Navigator" tab  of the Resource bar. Then select the "**Specify Vector**" button in the "Axis and Offset" group of the command window in order to define the vector along which the joint should slide (see Figure 34, step 6). For this, select the **X-axis** in the three-dimensional graphics window (see Figure 34, step 7).

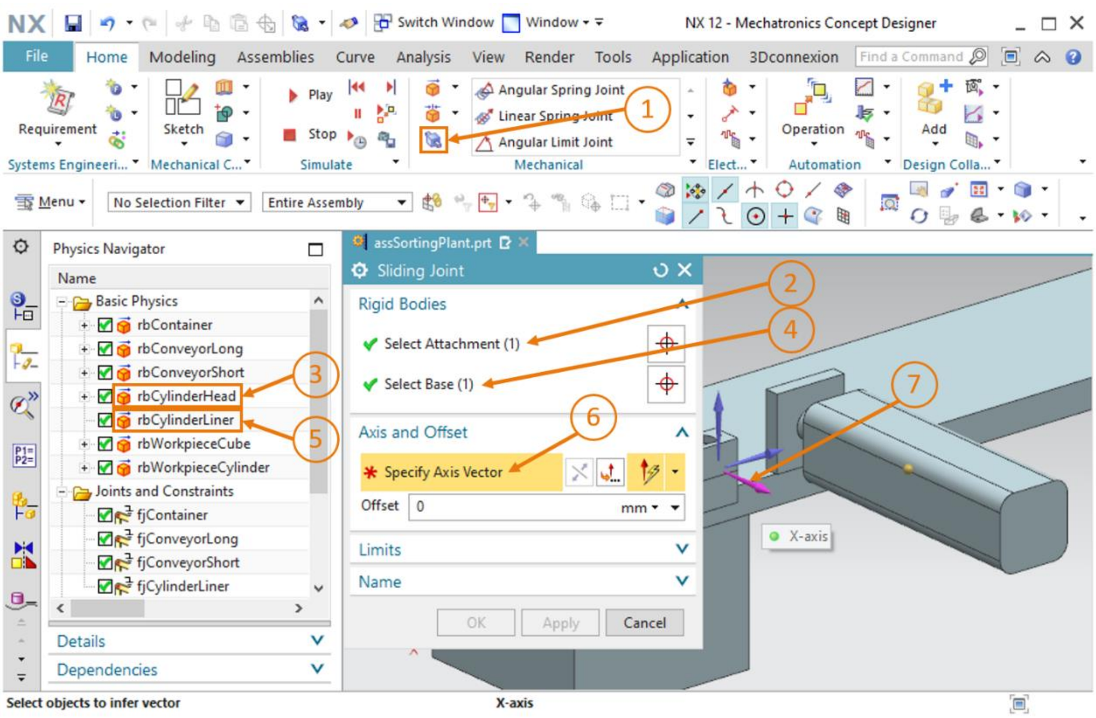


Figure 34: Creating a sliding joint for the ejector – Selection of rigid bodies and the axis vector

→ You can tell which direction the ejector would move in by looking at the orange arrow along the ejector. Mirror the axis vector by clicking the "**Reverse Direction**" button  because the ejector is to be extended from its initial position (see Figure 35, step 1).

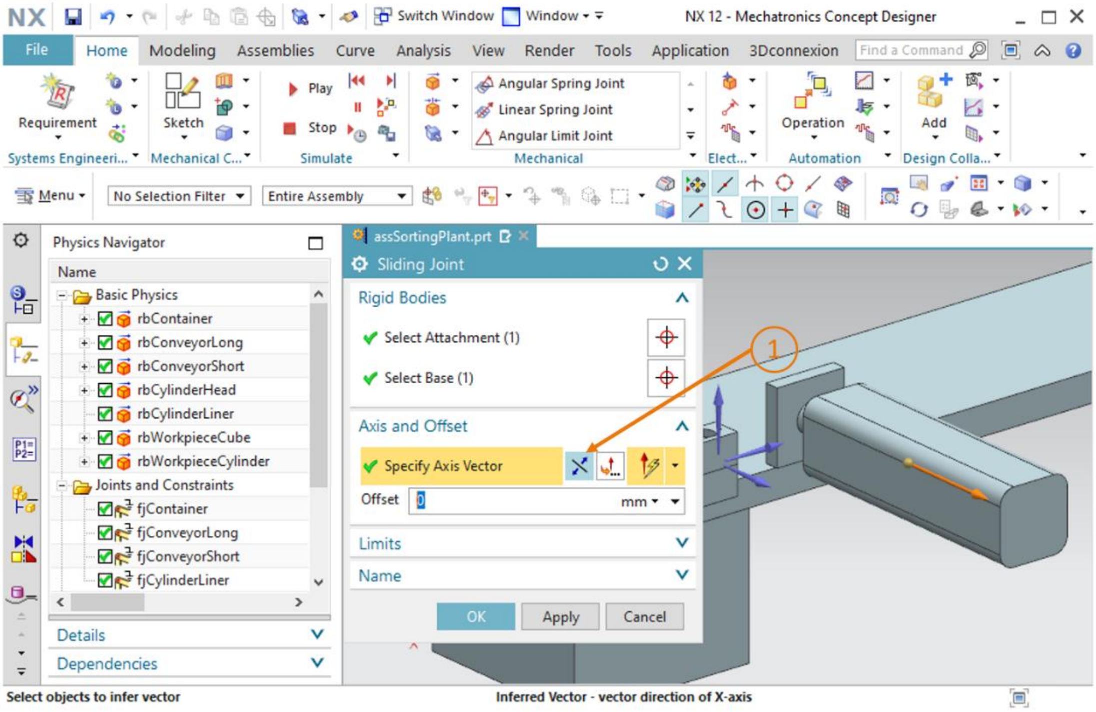


Figure 35: Creating a sliding joint for the ejector – Mirroring the axis vector

→ You can specify the maximum extension and retraction positions in the "Limits" group. The **Upper Limit** is to be **79 mm** and the **Lower Limit** is to be **0 mm** (see Figure 36, step 1). Enter "**sjCylinderHead_CylinderLiner**" as the name (see Figure 36, step 2) and finish the creation of the sliding joint by clicking the "OK" button (see Figure 36, step 3). The prefix "sj" stands for "sliding joint".

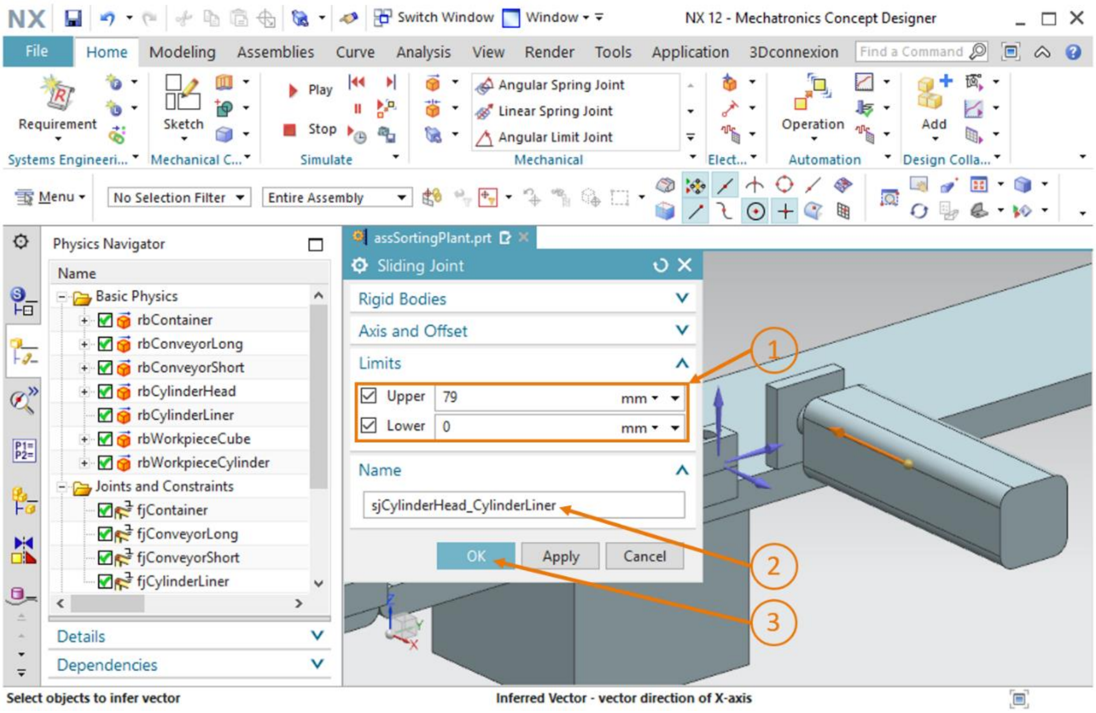


Figure 36: Creating a sliding joint for the ejector – Introducing sliding limits

→ Start a simulation again as described in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**". You can see that the head of the ejector does not leave its position and remains in the base of the ejector (see Figure 37). A controlled movement of the ejector head is to be implemented in the next step.

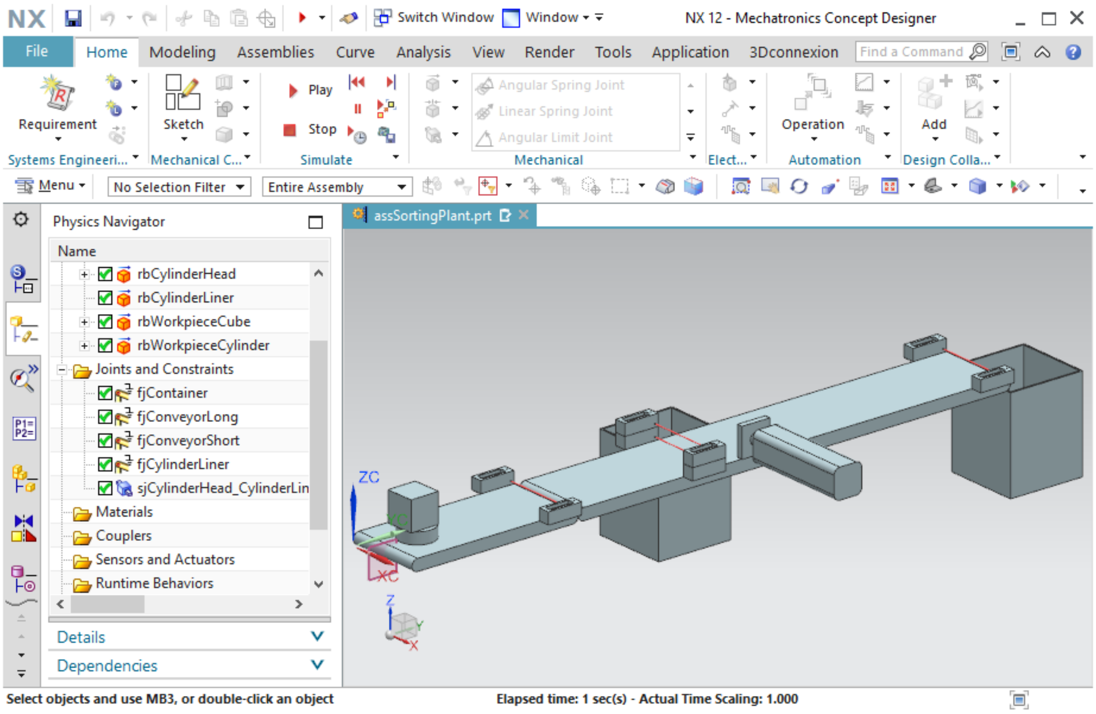
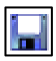


Figure 37: Simulation of the sliding joint in MCD

Stop the simulation and save your project by clicking the Save button .

7.6 Position control for ejector

For a controlled movement of the ejector head, you should use another dynamic property: position control. With a specified position together with a preselected speed, it is possible to move a movable element, such as a sliding joint, in a coordinated manner. The ejector has two travel processes: extending of the ejector head and retracting of the ejector head. A separate position control must be implemented for each of the two travel processes. Proceed as follows to create the two position controls:

Creating the position control for extending the ejector:

→ Navigate to the "Electrical" menu group and select the "**Position Control**" command from the drop-down menu for actuators (see Figure 38, step 1). The "Position Control" dialog window opens. In the "**Physics Object**" group, select the **sliding joint** you created in Chapter 7.5 (see Figure 38, steps 2 + 3). Specify the following values for the parameters in the "Constraints" group:

- A **Destination** of **80 mm** and a **Speed** of **80 mm/s** (see Figure 38, step 4)
- Activate "**Limit Acceleration**" with a value of **10000 mm/s²** for the maximum acceleration and maximum deceleration (see Figure 38, step 5)
- Activate "**Limit Force**" with a value of **100 N** for the forward force and reverse force (see Figure 38, step 6)

These values enable the head of the ejector to travel to its maximum extension position without a significant loss of time. Assign the name "**pcCylinderHeadExtend**" for this property and complete the creation of the position control by clicking "OK". The prefix "pc" stands for "position control".

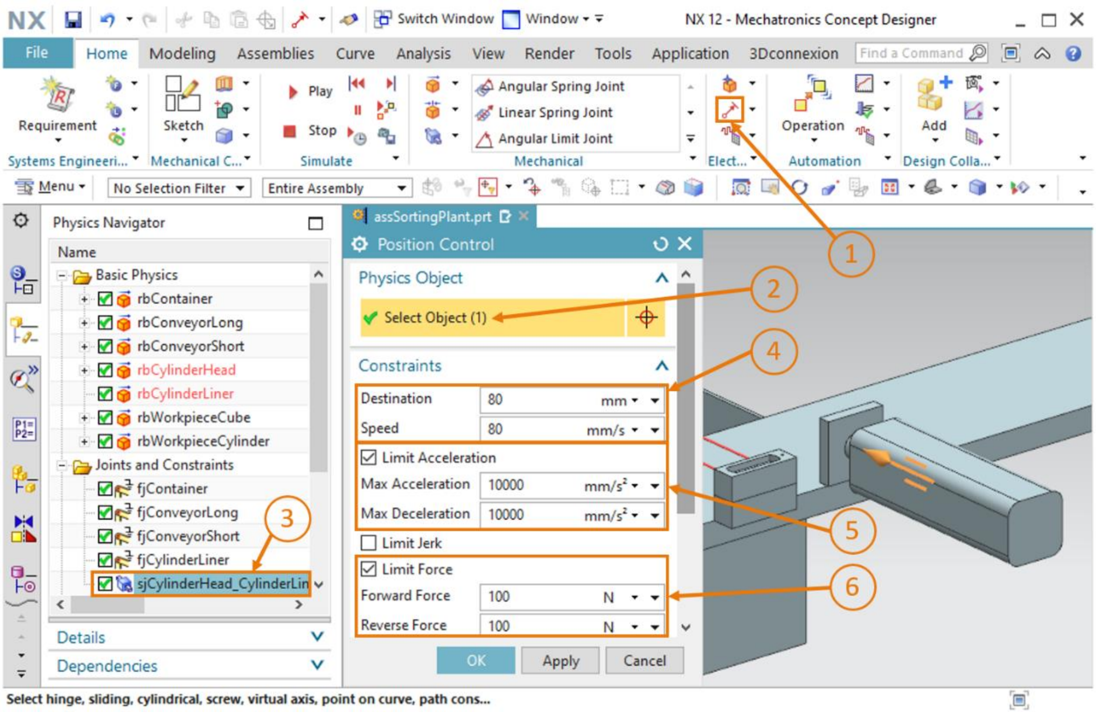


Figure 38: Creating a position control for extending of the ejector

→ Start a simulation. You will find a description for this in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**" for this. You can see here that the head of the ejector extends fully (see Figure 39). You must create an additional position control for retracting.

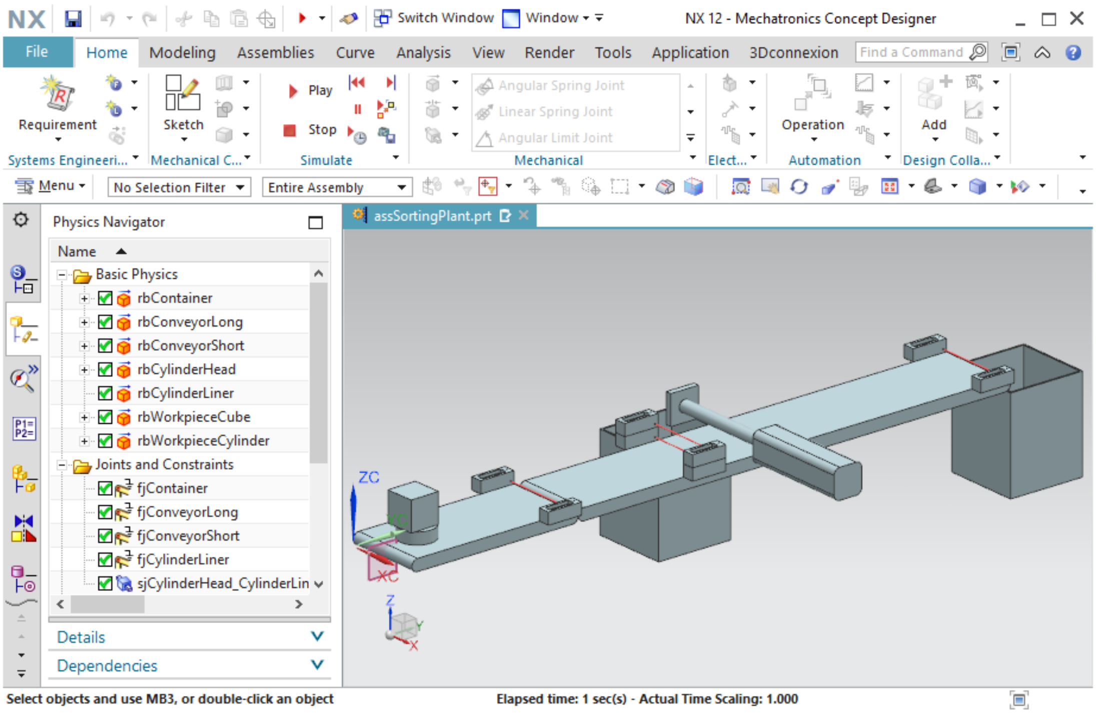
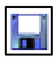


Figure 39: Simulation of the first position control of the ejector

Stop the simulation and save your project by clicking the "Save" button .

Creating the position control for retracting the ejector:

→ To create the second position control, proceed similarly as for creating the first position control of the ejector. However, be sure to specify a value of **0 mm** as the **Destination**. The other values are identical to the values for the preceding position control. Specify "**pcCylinderHeadRetract**" as the name and confirm your settings by clicking the "**OK**" button.

→ Now, run another simulation. Before you do this, however, make the two position controls "pcCylinderHeadExtend" and "pcCylinderHeadRetract" available for the Runtime Inspector. Follow the instructions in Chapter 4.3, "**Section: Adding and controlling a property in the simulation**" for this.

→ As explained in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**", when you start the simulation you should notice that the head of the ejector does not move at first. Using the Runtime Inspector, you can see that both the position controls for extending and retracting the ejector head are active. As a result, the two commands compete against each other so that no change is visible. However, as soon as you set the "**active**" signal of "**pcCylinderHeadRetract**" to "**false**", the ejector head extends fully (see Figure 40, step 1).

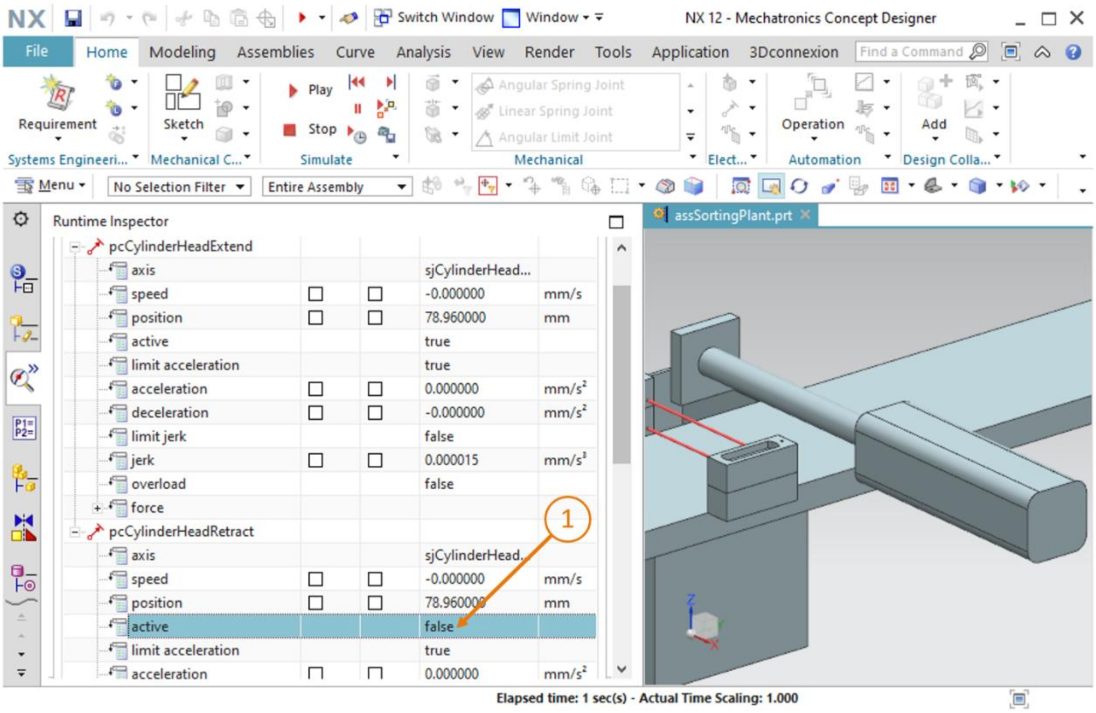


Figure 40: Simulation of the position controls of the ejector – Extending is active

→ On the other hand, if you deactivate extending again and instead set the **active** signal of "**pcCylinderHeadRetract**" to "**true**", the ejector head is retracted again (see Figure 41, step 1).

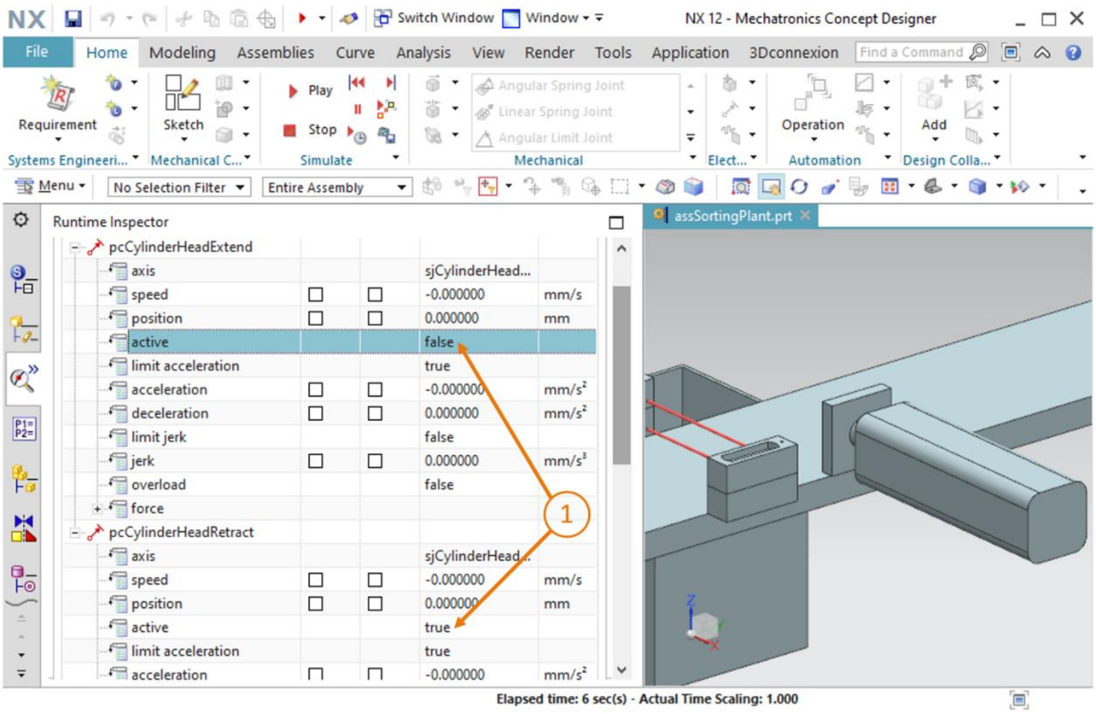
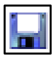


Figure 41: Simulation of the position controls of the ejector – Retracting is active

This allows the ejector to be controlled. Later on in Chapter 7.9, you will provide limit switches for the ejector unit with a sensor signal. Stop the simulation and save your project by clicking the "Save" button .

7.7 Specifying transport surfaces for the conveyor belts

Although the current intermediate status of your assembly allows you to hold all bodies in space and have them interact with one another, no controlled movements are currently possible except for the position controls of the ejector head. In this chapter, you are to insert transport surfaces for the two conveyor belts that will enable workpieces to be guided along the sorting process. Use the following procedure for this:

Creating a transport surface for conveyorShort:

→ Open the "**Transport Surface**" command via the "Mechanical" menu group or the Command Finder. You start by selecting the conveyor faces of a body in the "Conveyor Face" group. Use the **planar surface of the conveyorShort conveyor belt** for this, as shown in Figure 42, step 2.

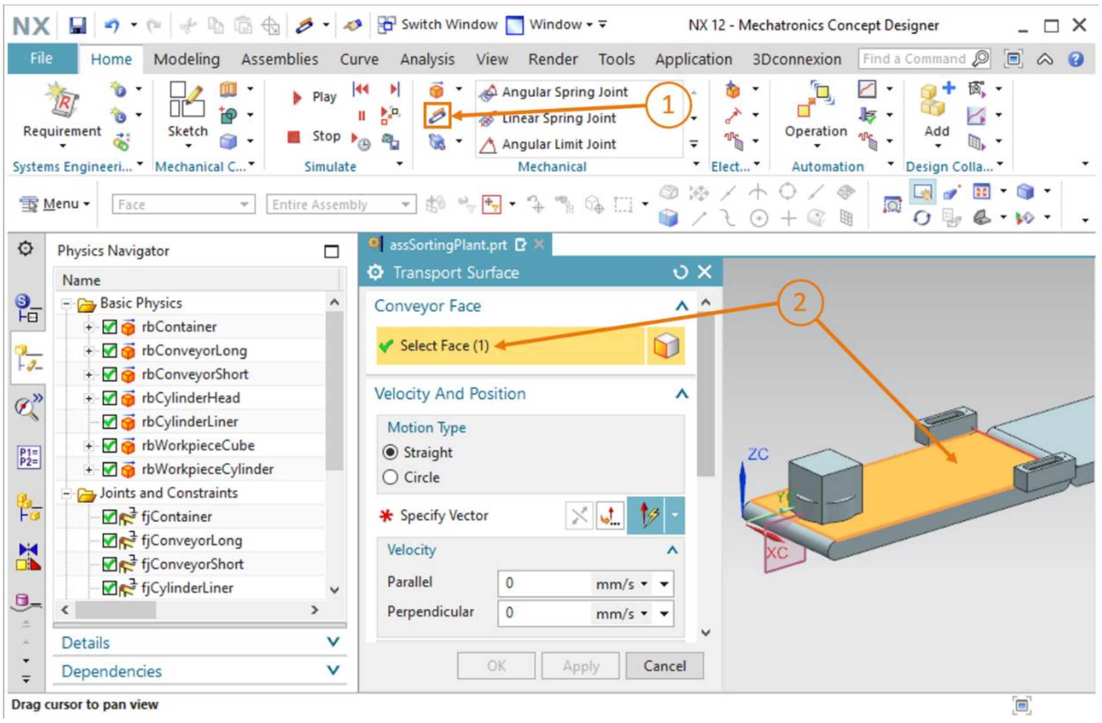


Figure 42: Creating a transport surface for the conveyorShort conveyor belt – Selection of conveyor face

→ Next, you need to specify the vector indicating the travel direction. This is along the Y-axis in this model. For this, you select the "**Specify Vector**" button in the "**Velocity and Position**" group and then click on the "**Y-axis**" vector displayed in the three-dimensional graphics window (see Figure 43, step 1). Leave the default values for velocity and start position. Specify "**tsConveyorShort**" as the name and finish the creation of this transport surface by clicking the "OK" button. The prefix "ts" stands for "transport surface".

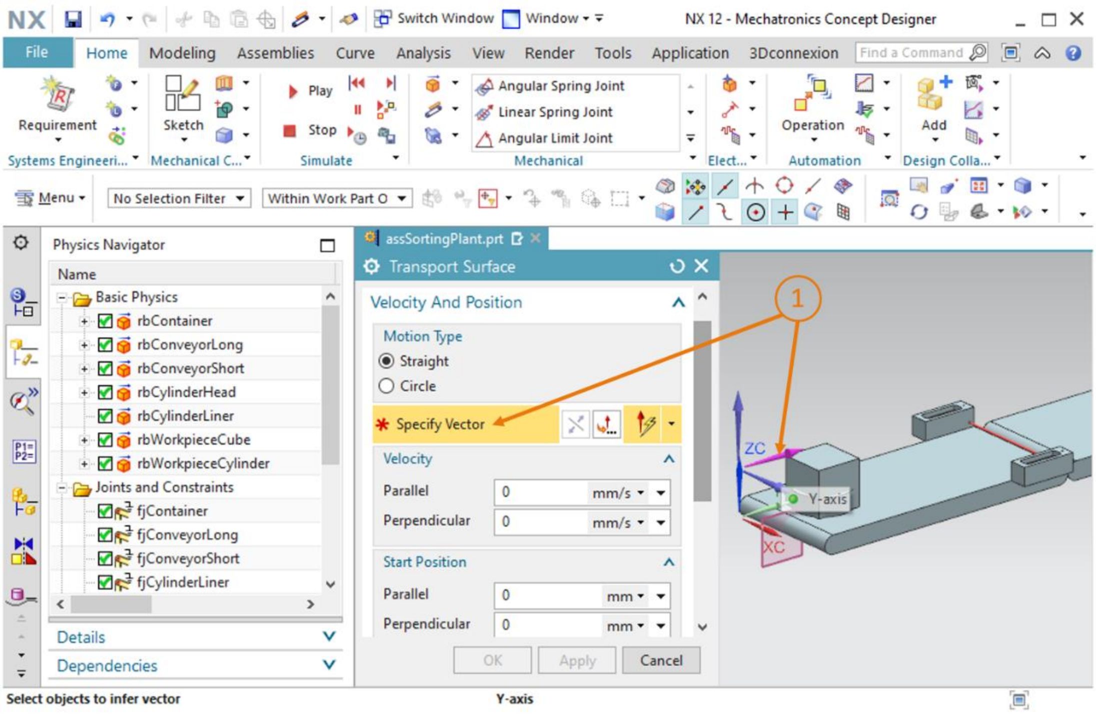


Figure 43: Creating a transport surface for the conveyorShort conveyor belt – Specifying the travel vector

Creating a transport surface for conveyorLong:

→ Follow the same procedure to create a transport surface for conveyorLong as you used to create conveyorShort. Here, select the **planar surface of the conveyorLong body** as the conveyor surface instead.

→ Start a simulation as explained in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**". You will not notice any change in this simulation compared with the preceding simulation in Chapter 7.6 (see Figure 44). The reason is that the speed of the conveyor belts is not yet controlled. This will be covered in the next chapter.

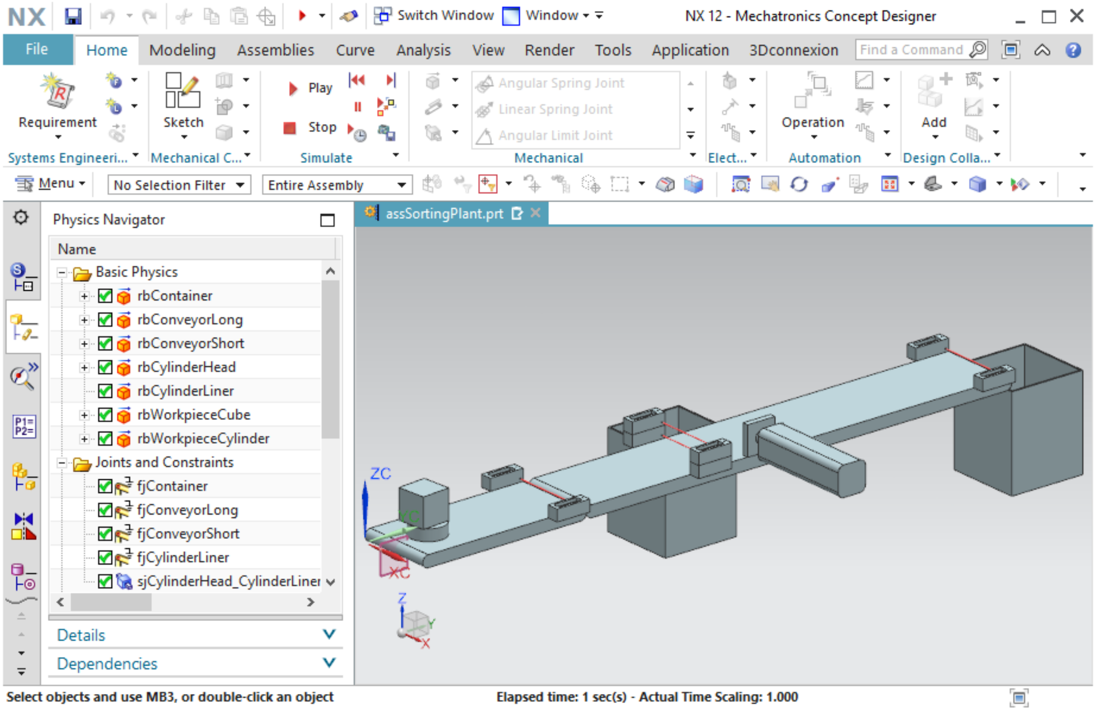
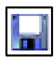


Figure 44: Simulation of the transport surfaces in MCD

Stop the simulation again and save your project by clicking the "Save" button .

7.8 Speed control for conveyor belts

You will use the dynamic property "Speed Control" to control the conveyor belts. You are to create two speed controls for each conveyor belt. One speed control is to provide for moving of the conveyor belt at constant speed, while the other enables the conveyor belt to be controlled with a variable speed. Use the following guidelines for creating these dynamic properties:

Speed controls for the conveyorShort conveyor belt:

→ Start by creating the speed control for the specification of a constant speed. To do this, start the "**Speed Control**" command in the "Electrical" menu group or via the Command Finder (see Figure 45, step 1). This opens the "Speed Control" command window. Same as for the position control in Chapter 7.6, a movable element of your assembly to which the speed is to apply is required as the physics object. In this case, this is your transport surface "**tsConveyorShort**", which you are to select as shown in Figure 45, step 2. The direction is to run "**Parallel**" with the vector of the transport surface (see Figure 45, step 3) Specify a constant speed of **50 mm/s** as a constraint (see Figure 45, step 4). Assign the name "**scConveyorShortConstSpeed**" (see Figure 45, step 5) and complete the creation of the speed control by clicking the "OK" button. The prefix "sc" stands for "speed control".

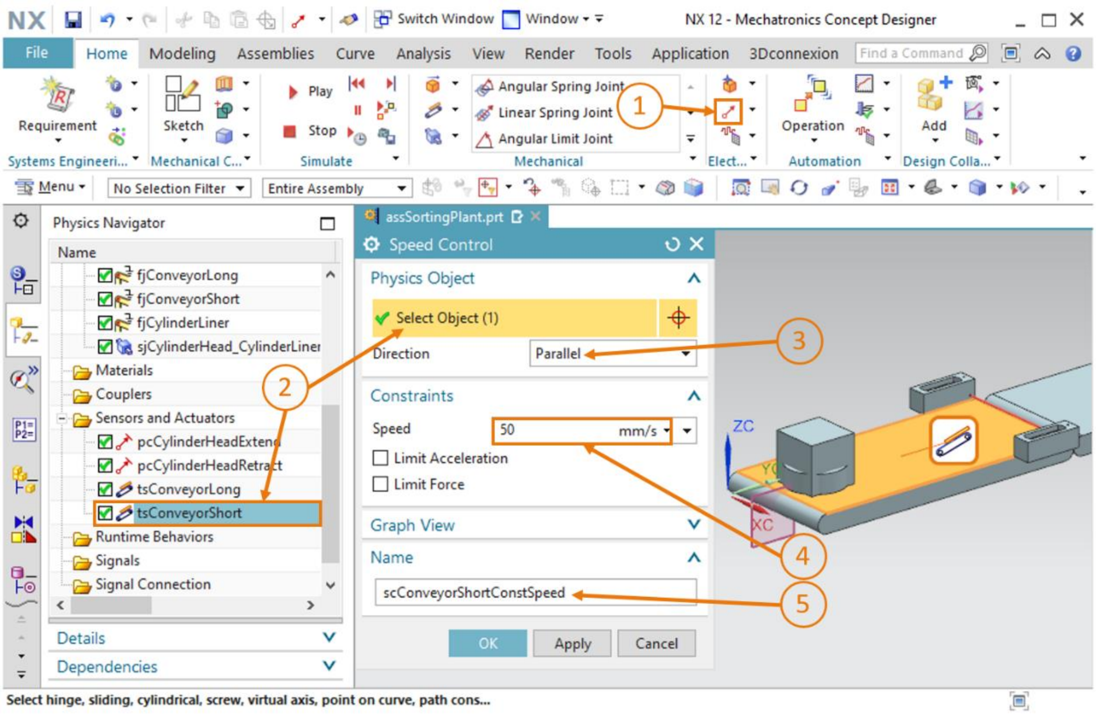


Figure 45: Creating a speed control for a conveyor belt

→ Continue by creating the second speed control for conveyorShort for moving the conveyor belt with a variable speed. Proceed exactly the same as for creation of the first control. Here, as well, select the "tsConveyorShort" transport surface and the "Parallel" direction. So that the conveyor belt does not move when the control is activated, do not specify any speed as a constraint, i.e. value = **0 mm/s**. The speed can then be variably set by the user during the course of a simulation. Select "**scConveyorShortVarSpeed**" as the name.

Speed controls for the conveyorLong conveyor belt:

→ Follow the same principle for the two speed controls of conveyorLong as for the conveyorShort conveyor belt. However, in this case, use the "**tsConveyorLong**" transport surface as the physics object.

→ All speed controls for the conveyor belts have now been defined. Simulate the result. Beforehand however, add the speed controls you created in this chapter to the Runtime Inspector. Proceed as described in Chapter 4.3, "**Section: Adding and controlling a property in the simulation**" for this. Make the following changes in the Runtime Inspector tab of the Resource bar before starting the simulation (partially shown in Figure 46):

- For **scConveyorShortConstSpeed**, set the **active** signal to the "**false**" value
- For **scConveyorShortVarSpeed**, set the **active** signal to the "**false**" value and **speed to 5 mm/s**
- For **scConveyorLongConstSpeed**, set the **active** signal to the "**false**" value
- For **scConveyorLongVarSpeed**, set the **active** signal to the "**false**" value and **speed to 10 mm/s**

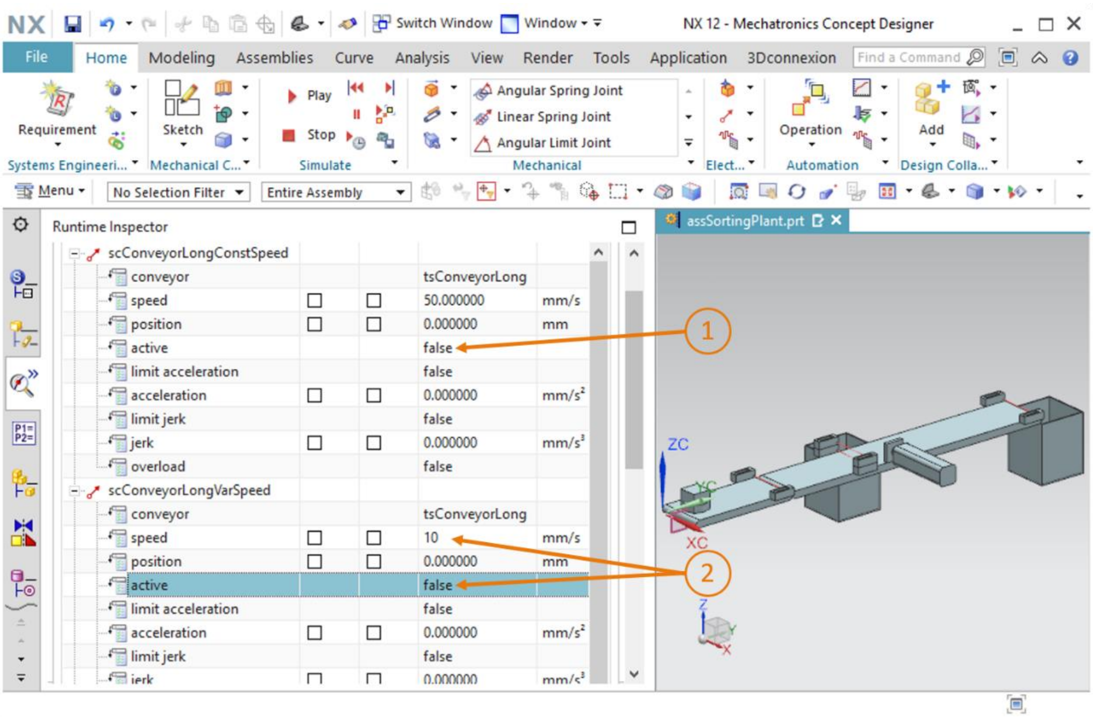


Figure 46: Preparing simulation of the speed control via Runtime Inspector

→ Start the simulation as explained in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**". You should see in the model that no conveyor belt is moving yet.

→ Modify the **active** signal of the "**scConveyorShortConstSpeed**" control to the "**true**" value. The conveyor should now move at a speed of 50 mm/s. Monitor the value in the "position" field for this.

→ Change the **active** signal of the "**scConveyorShortConstSpeed**" control back to the "**false**" value. Despite of that, the conveyor still moves along at the constant speed of 50 mm/s. That is because the speed setting is not reset when the active signal is removed.

→ Specify the "**true**" value for the **active** signal of the "**scConveyorShortVarSpeed**" control. The belt now moves at a speed of 5 mm/s, as you have specified for the system. You can also check this in the "position" field, which is highlighted in Figure 47, step 1.

→ Test the same behavior with the speed controls of the "**conveyorLong**" conveyor belt. You can expect a similar result here. Also monitor the position changes in each case, as presented in Figure 47, step 2 by way of example.

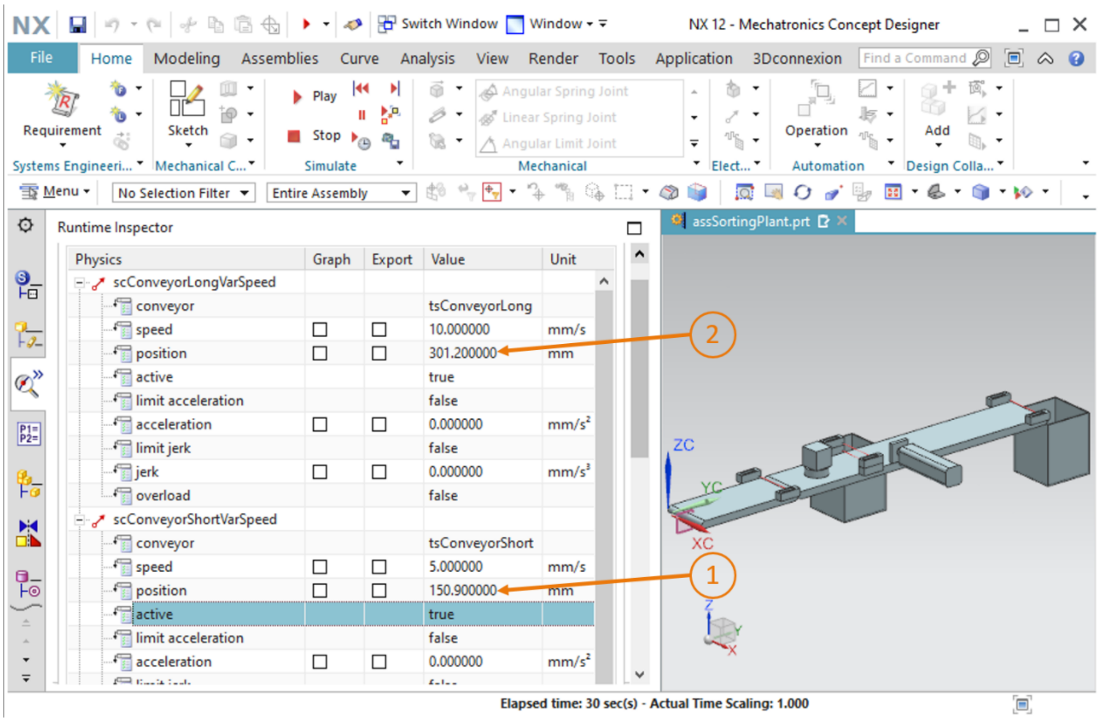
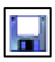


Figure 47: Simulation of the speed controls in MCD

You have now checked the basic functionality of the speed controls. Stop the simulation and save your project by clicking the "Save" button .

7.9 Collision sensors for the light sensors and limit switches

With the current intermediate status of your dynamic model, it is possible to transport the two workpieces on the conveyor belts and to actuate the ejector. For an orderly sorting process, however, it is not yet possible to differentiate between the various workpieces. In addition, an external indication of the ejector position is also not yet possible. For these tasks, you are to define the light sensors along the conveyor belts as well as the limit switches in the ejector as collision sensors. These sensors will enable you to detect when a collision with another collision body occurs. Use the following procedure to create the collision sensors:

→ To create individual collision sensors, you need to be able to access individual components in the assembly. Open the "**Assembly Navigator**" tab in the Resource bar for this (see Figure 48, step 1). Select the packed components "**lightRay x4**" and "**limitSwitchSensor x2**" one after the other and right-click on them (see Figure 48, step 2). Click the "**Unpack**" command in the shortcut menu (see Figure 48, step 3). You can now access the individual components in a direct way.

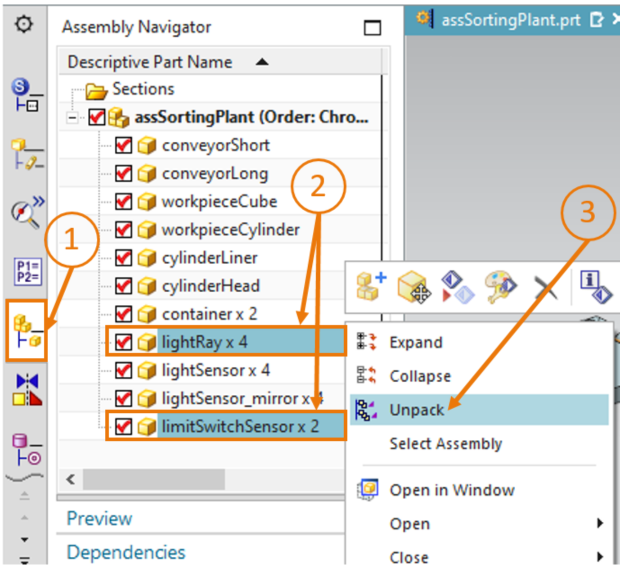


Figure 48: Unpacking models of the same type in an assembly

NOTE

Before starting the following steps, it is recommended that you individually select the individual "lightRay" and "limitSwitchSensor" models in the Assembly Navigator and make a note of which bodies are located at specific locations in your assembly.

→ Find the "**Collision Sensor**" command in the "Electrical" menu group or via the Command Finder (see Figure 49, step 1). Click on the corresponding icon opens the "Collision Sensor" command window. Open the "Collision Sensor Object" group and click the "**Select Object**" button (see Figure 49, step 2). In the Assembly Navigator, select **the light sensor at the end of the first conveyor belt** "**conveyorShort**" (see Figure 49, Step 3). Specify "**Line**" as the collision shape in the "Shape" group.

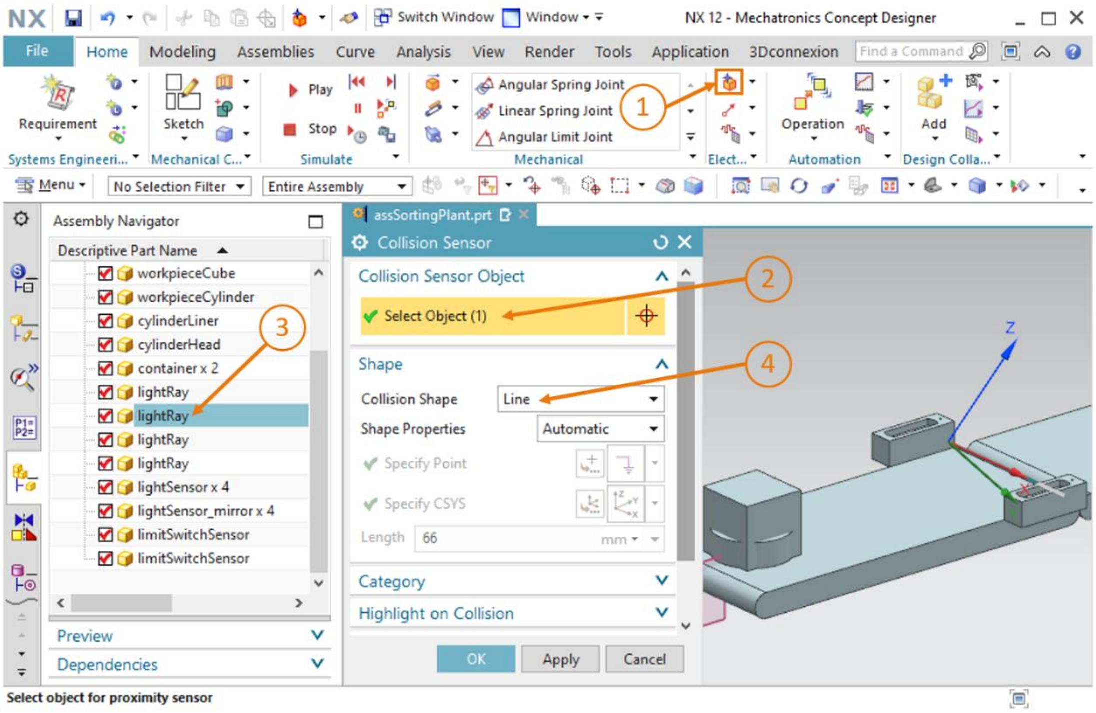


Figure 49: Creating the collision sensor for counting all workpieces – Selection of the collision object and collision shape

→ Leave the category set to the value "**0**" (see Figure 50, step 1). Clear the "**Highlight on Collision**" check box (see Figure 50, step 2). Finish by assigning the name "**csLightSensorWorkpiece**" (see Figure 50, step 3) and confirm your inputs by clicking the "OK" button (see Figure 50, step 4). The prefix "cs" stands for "collision sensor".

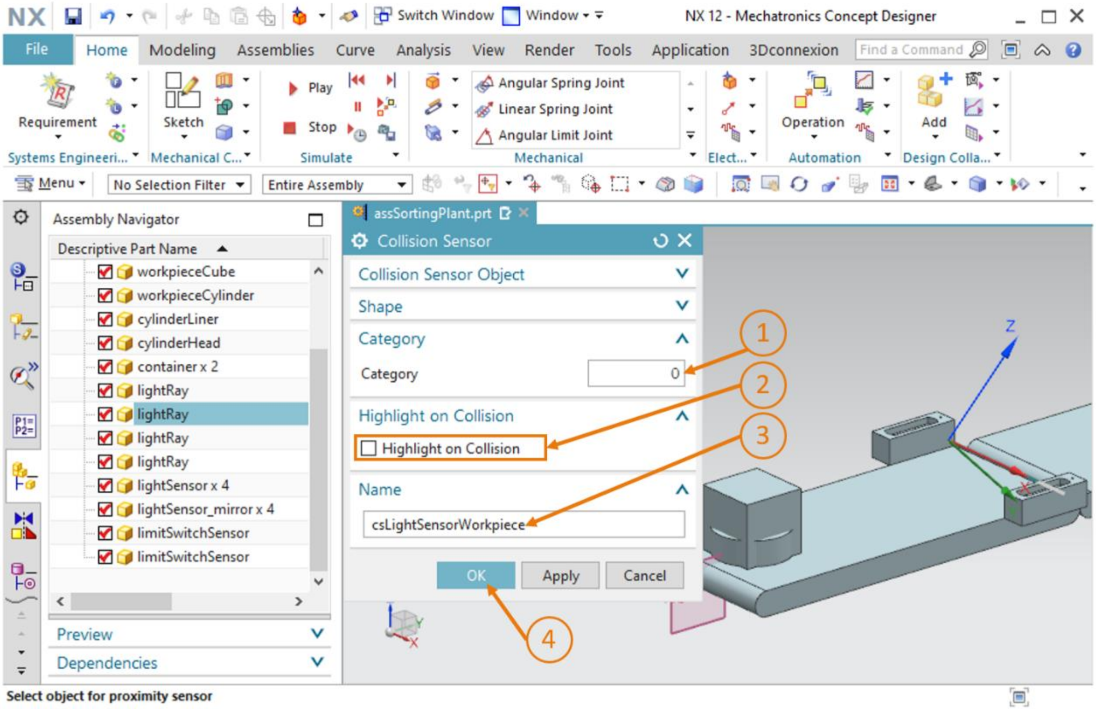


Figure 50: Creating the collision sensor for counting all workpieces – Specifying other settings and the name

→ You have now created the first collision sensor (highlighted in Figure 51, step 1). Define the remaining collision sensors as follows:

- The **lower light sensor in the middle of the rear conveyor belt "conveyorLong"** (see Figure 51, step 2) is to be created as a collision sensor named "**csLightSensor Cylinder**" for the purpose of detecting cylindrical workpieces.
- The **upper light sensor in the middle of the rear conveyor belt "conveyorLong"** (see Figure 51, step 3) is to be created with the name "**csLightSensorCylinderTop**" for the unequivocal differentiation of cylindrical and cuboid workpieces. This is possible because the cylindrical and cuboid workpieces have different heights so that only the smaller cylindrical workpieces are detected by the lower light sensor and the larger cuboid workpieces are detected by both light sensors.
- The remaining workpieces on the conveyor belt are counted with the **light sensor at the end of the long conveyor belt "conveyorLong"** (see Figure 51, step 4). With correct sorting, only cuboid bodies will be remaining. The name of the associated collision sensor is to be "**csLightSensorCube**".
- The **limit sensor at the end of the ejector** (see Figure 51, step 5) will detect if the ejector is not yet fully extended. In this case, select **the limitSwitchSensor at the end of the ejector** as the collision sensor object in the Assembly Navigator. Name the collision sensor "**csLimitSwitchCylinderNotExtended**".
- The **limit sensor at the beginning of the ejector** (see Figure 51, step 6) signals that the ejector is fully retracted. Assign this collision sensor the name "**csLimitSwitchCylinderRetracted**".

Proceed exactly the same as for creation of the first collision sensor. Note only the new name and the selection of the right components as collision sensor objects. For differentiating the two limit switches, it is recommended that you hide the ejector (cylinderHead and cylinderLiner). Proceed as explained in Chapter 7.4.1, "**Section: Displaying/hiding components and assemblies**" for this.

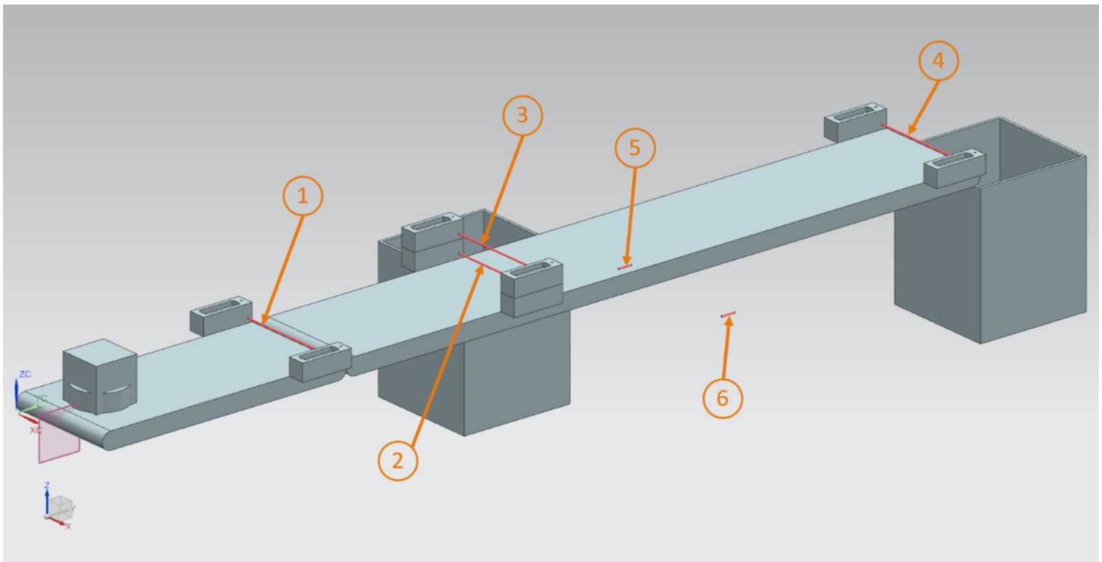


Figure 51: Overview of all collision sensors in the sorting plant

→ Now start a simulation again. Make sure beforehand that you have displayed all components in the assembly again. If components are missing in the assembly, reactivate them as described in Chapter 7.4.1, "**Section: Displaying/hiding components and assemblies**". Add all collision sensors to the Runtime Inspector as explained in Chapter 4.3, "**Section: Adding and controlling a property in the simulation**". Now, also add the speed controls **"scConveyorShortConstSpeed", "scConveyorShortVarSpeed"**, **"scConveyorLong ConstSpeed"** and **"scConveyorLongVarSpeed"** for controlling the conveyor belts. To test the limit switches, you should also load the two position controls **"pcCylinderHeadExtend"** and **"pcCylinderHeadRetract"** in the Runtime Inspector.

→ Start the simulation as explained in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**". Start by testing only the behavior of the light sensors of the sorting plant. To do this, set the **active** signal of the "**scConveyorShortConstSpeed**" and "**scConveyorLongConstSpeed**" speed controls to "**true**", and the **active** signal of the two other speed controls to "**false**". You can see the conveying of both workpieces in the simulation. When the collision sensors of the light sensors are passed through (see Figure 51, steps 1 – 4), the "**triggered**" fields of the respective sensor are set to "**true**". Otherwise, they remain set to "**false**". This is illustrated by way of example for the first light sensor "**csLightSensorWorkpiece**" in Figure 52, step 1.

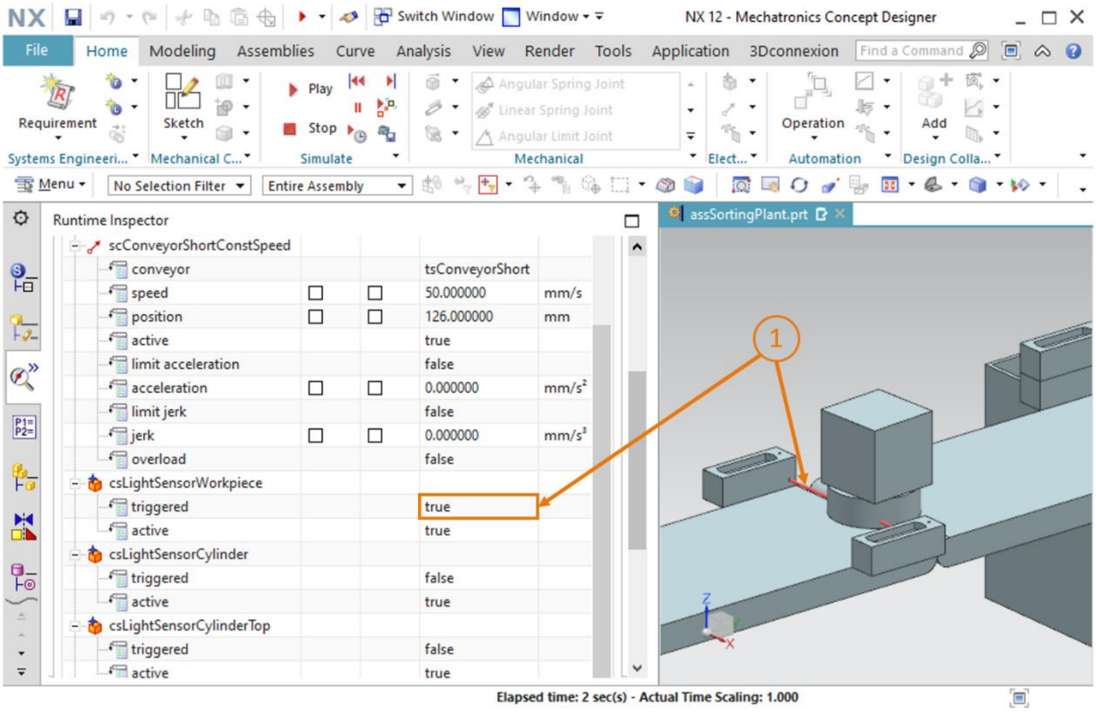


Figure 52: Behavior of the collision sensors of the light sensors during the simulation

→ For the second part of the simulation, only examine the collision sensors of the limit switches and the position controls of the ejector. When the simulation starts, the ejector remains retracted and **both limit switches** are set to "**true**". Now, specify the **active** signal "**false**" for "**pcCylinderHeadRetract**" in the Runtime Inspector. At the same time, the active signal for the "**pcCylinderHeadExtend**" position control is to continue to remain set to "**true**". The ejector now extends. During the extension movement, "**csLimitSwitchCylinderRetracted**" is set to "**false**" and "**csLimitSwitchCylinderNotExtended**" remains set to "**true**" (see Figure 53, step 1). Only when the ejector head is fully extended is the "**csLimitSwitch CylinderNotExtended**" collision sensor also set to "**false**".

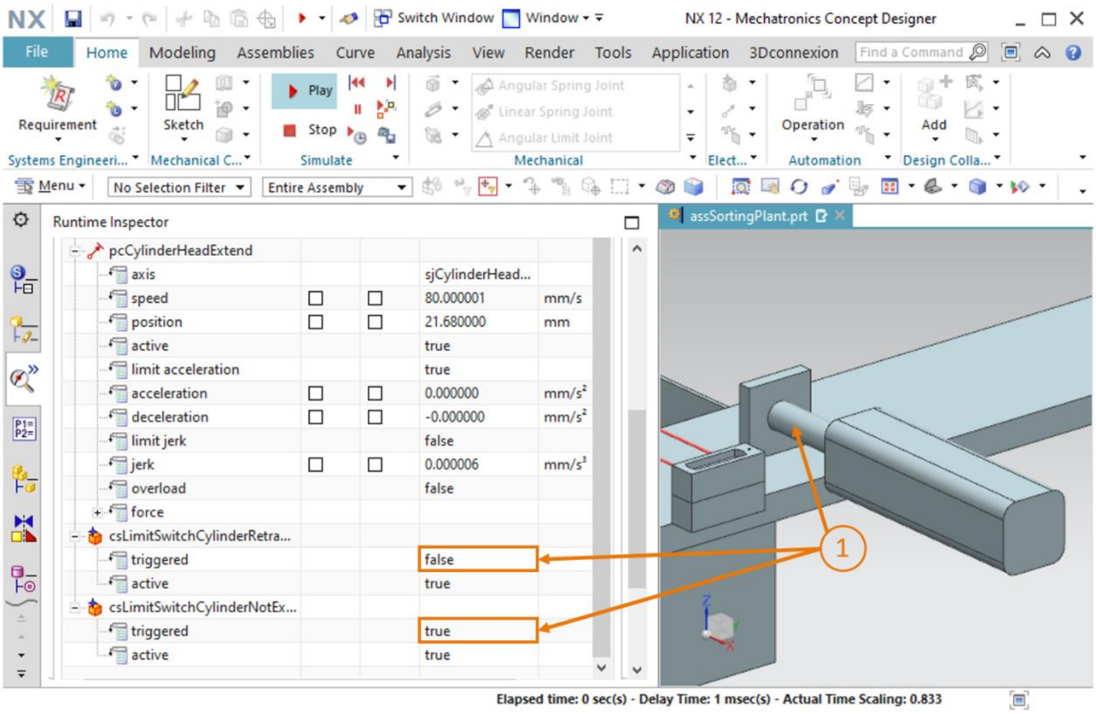
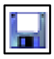


Figure 53: Behavior of the collision sensors of the limit switches during the simulation

All collision sensors now react as expected. Stop the simulation and save your entire project by clicking the "Save" button .

7.10 Object sources for the workpieces

Now that the conveying of the two workpieces by the conveyor belts and their detection by the collision sensors is functioning, various workpieces are to be generated at regular intervals. For this, you use the dynamic property "Object Source", whereby a rigid body will be generated as a new replica in a simulation, either triggered by an event or after expiration of a time interval. Follow the steps below for this:

→ Navigate in the "Mechanical" menu group or via the Command Finder to the "**Object Source**" command and click on it (see Figure 54, step 1). This opens the corresponding command window. Select the "**Select Object**" button in the "Object to Copy" group (see Figure 54, step 2). In the Physics Navigator tab of the Resource bar, select the "**rbWorkpieceCube**" rigid body as the object, so that the object source generates a cuboid workpiece (see Figure 54, step 3). In the "Copy Event" group, specify "**Time Based**" as the trigger, whereby the workpiece will be generated at regular time intervals. The **time interval** is to be **10 s** and the **start offset** is to be **0 s** (see Figure 54, step 4). Finally, assign the name "**osWorkpiece Cube**" (see Figure 54, step 5) and confirm your settings by clicking the "OK" button (see Figure 54, step 6). The prefix "os" stands for "object source".

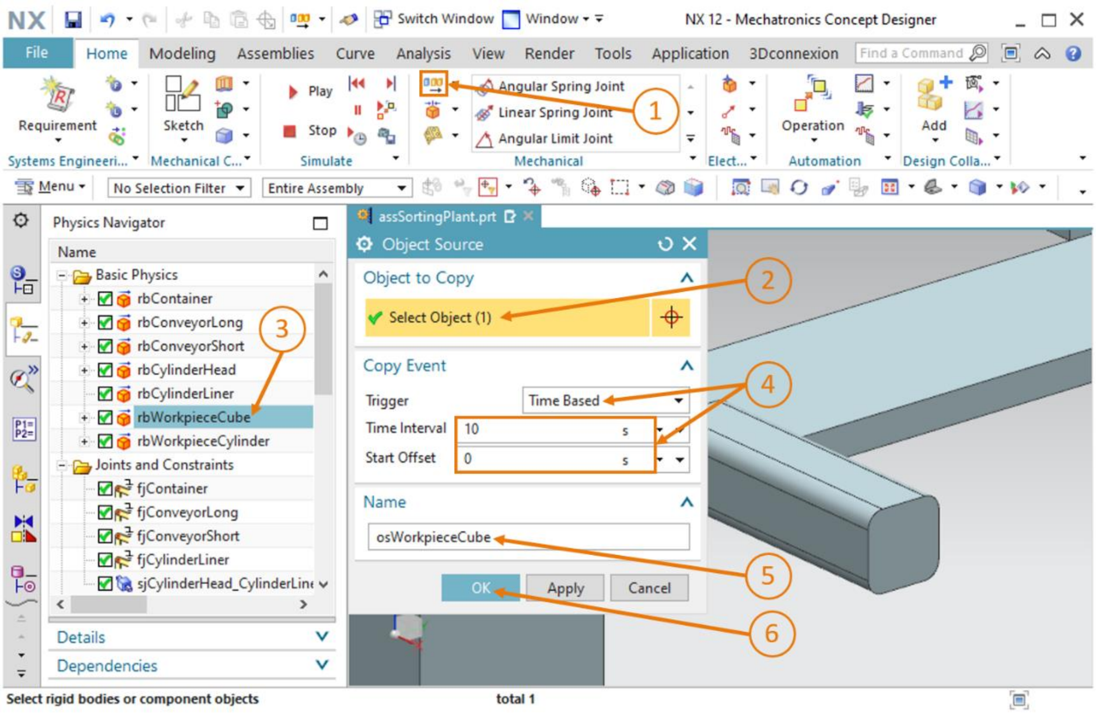


Figure 54: Creating an object source for a workpiece

NOTE

For the object source, the time is counted internally in MCD. As a result, an external reset of the counter during a simulation (for example, via PLCSim Advanced) is not readily possible. However, the generation of new workpieces can be prevented externally using the "active" signal, which has been put to use previously in Modules 1 - 3 of this workshop series.

→ Now, also add the object source for the cylindrical workpiece to the project. You can proceed in basically the same way as for the first object source. However, this time select the "**rbWorkpieceCylinder**" rigid body as the object to copy and specify a **start offset** of **5 s**. The first cylindrical workpiece will then be generated 5 s after the start of the simulation. Additional cylindrical workpieces will be generated every 10 s thereafter.

→ Test the behavior by starting a simulation. Beforehand, however, you should add the speed controls of the conveyor belts to the Runtime Inspector and ensure that only the two controls "**scConveyorShortConstSpeed**" and "**scConveyorLongConstSpeed**" are active. After adding both object sources, also ensure that they are active via the Runtime Inspector. Proceed as described in Chapter 4.3, "**Section: Adding and controlling a property in the simulation**". By starting a simulation as explained in Chapter 7.1, "**Section: Starting and stopping a simulation in MCD**", you can observe how another workpiece is added to the simulation at 5 s intervals (see Figure 55).

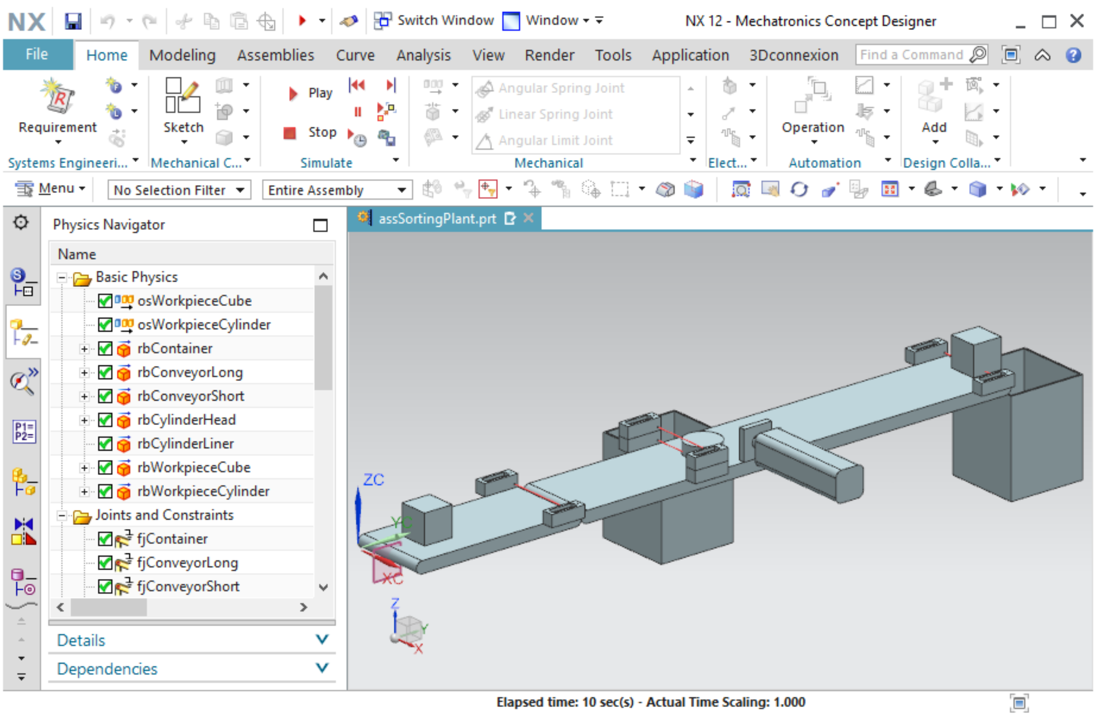
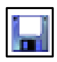


Figure 55: Simulation of the object sources in MCD

Stop the simulation and save your project completely by clicking the "Save" button .

For removing objects in a simulation, you can also define collision sensors as an object sink, as explained in Chapter 4.2.1. However, this is not part of this workshop series.

You have now transformed a static 3D model into a functional dynamic 3D model involving a variety of dynamic properties. To externally control these properties, you need to establish a connection between your PLC program and the digital twin, which completes the virtual commissioning. You will learn how to do this in Module 6 of this workshop series.

8 Checklist – Step-by-step instructions

The following checklist helps students/trainees to independently check whether all tasks in the step-by-step instructions have been carefully completed and enables them to successfully finish the module on their own.

No.	Description	Checked
1	The "assSortingPlant" assembly with the complete static 3D model was successfully opened in MCD.	
2	All rigid bodies of the sorting plant were created, and the behavior was checked by a simulation.	
3	The fixed joints of the individual components were successfully defined and tested in the simulation.	
4	The required collision bodies were successfully assigned to the rigid bodies and their behavior was checked in the simulation.	
5	A sliding joint was successfully defined for the ejector and was then tested in a simulation.	
6	The required position controls were successfully specified for the sliding joint and their functionality was tested in the simulation.	
7	Transport surfaces for the conveyor belts were successfully defined and simulated in the sorting plant.	
8	The speed controls for the transport surfaces were successfully created and tested in the simulation.	
9	The collision sensors for the light sensors and for the limit switches of the ejector were implemented and their functionality was successfully tested in a simulation.	
10	Object sources were successfully defined for the workpieces and checked in a simulation.	

Table 1: Checklist for "Creation of a Dynamic 3D Model Using the Mechatronics Concept Designer CAE System"

9 Additional information

You can find additional information as an orientation aid for initial and advanced training, for example: Getting Started, videos, tutorials, apps, manuals, programming guidelines and trial software/firmware, at the following link:

Preview "Additional information" – In preparation

Here are some interesting links:

[1] support.industry.siemens.com/cs/document/90885040/programming-guideline-for-s7-1200-s7-1500?dti=0&lc=en-US

[2] support.industry.siemens.com/cs/document/109756737/guide-to-standardization?dti=0&lc=en-US

[3] omg.org/spec/UML/2.5.1/PDF

[4] geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/

[5] geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/

Further Information

Siemens Automation Cooperates with Education
siemens.com/sce

SCE Learn-/Training Documents
siemens.com/sce/documents

SCE Trainer Packages
siemens.com/sce/tp

SCE Contact Partners
siemens.com/sce/contact

Digital Enterprise
siemens.com/digital-enterprise

Totally Integrated Automation (TIA)
siemens.com/tia

TIA Portal
siemens.com/tia-portal

TIA Selection Tool
siemens.com/tia/tia-selection-tool

SIMATIC Controller
siemens.com/controller

SIMATIC Technical Documentation
siemens.com/simatic-docu

Industry Online Support
support.industry.siemens.com

Product catalogue and online ordering system Industry Mall
mall.industry.siemens.com

Siemens
Digital Industries, FA
P.O. Box 4848
90026 Nuremberg
Germany

siemens.com/sce