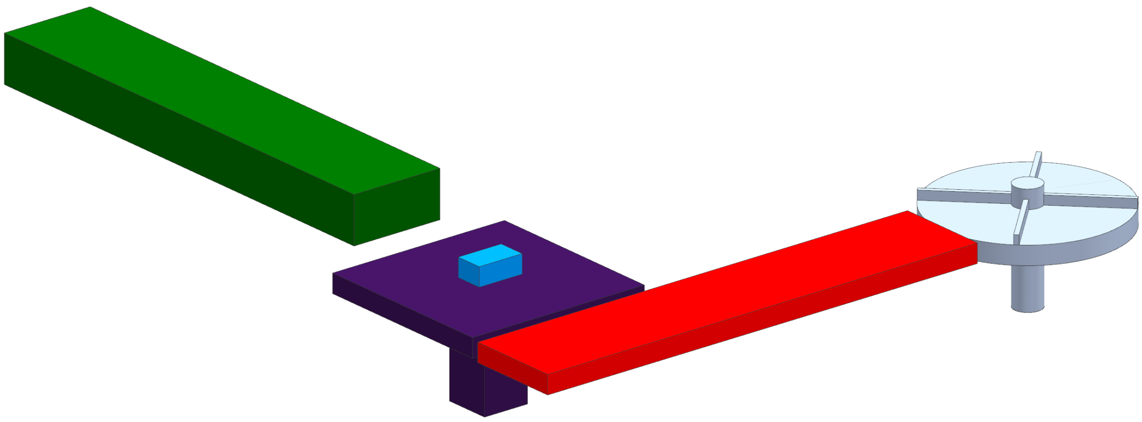
Tutorial NX MCD

Christophe JOB

August 2021



1 Introduction

NX Mechatronics Concept Designer (MCD) is a very useful tool to visualize a series of operations. It allows to create a planning of operations that follow each other while including conditions.
In this tutorial you will learn how to animate a simple series of operations with NX MCD.

2 Scenario of the production line

The production line is composed of 5 parts:

- A box (blue) : *Box_tuto.prt*
- A table (purple) : *table.prt*
- An (hydraulic or electrical) actuator (green) : *verrin.prt*
- A conveyor (red) : *Conveyor.prt*
- A turning table (light grey) : *Table_tournante.prt*
- A very thin cylinder (representing a "laser") : *collision_sensor.prt*

All assembled in the file *assembly_tuto_MCD.prt*.

The production line works in the following way:

1. The actuator pushes the box from the table to the conveyor

2. The conveyor carries the box towards the turning table
3. The turning table turns when the box is on it and brings the box to another direction

3 NX MCD: general principles

NX MCD is activated in the "Application" part of NX (figure 1).

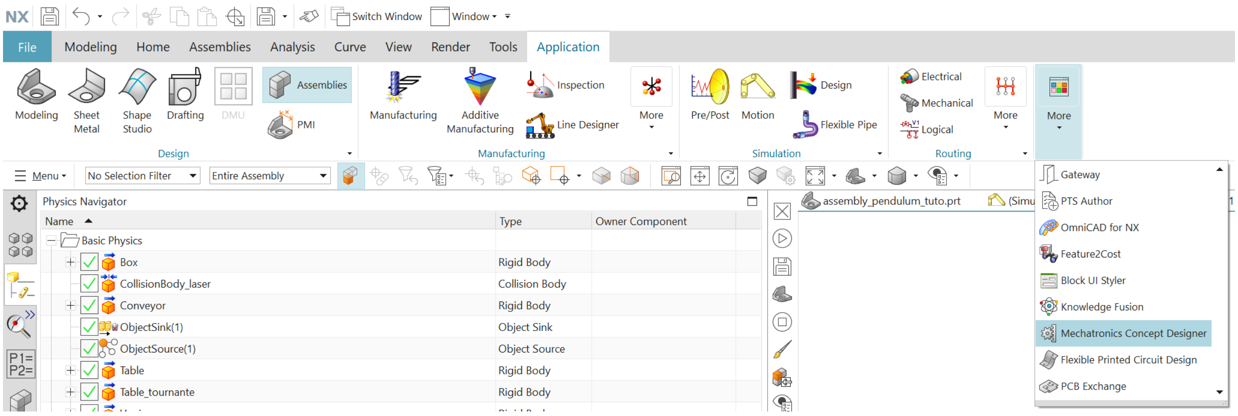


Figure 1: Activating NX MCD

3.1 Main buttons

The buttons most commonly used in NX MCD are shown on figure 2.

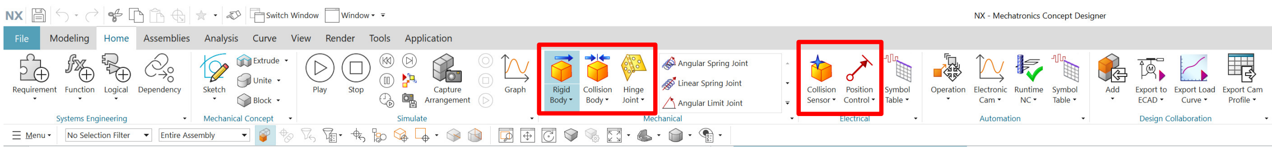


Figure 2: MCD main buttons

3.1.1 Rigid bodies and collision bodies

The first thing to do when using NX MCD is to define the **rigid bodies** and the **collision bodies**.

Rigid bodies are the parts that will move and collision bodies are the part that will be in contact with other parts.

A part can be defined only as a collision body if it is not moving. A moving part has to be defined as a rigid body **AND** a collision body if it is expected to have some contacts with other parts.

1. To create a rigid body simply click on **Rigid body** then choose the part (the steps are shown on figure 3). The mass properties are automatically computed by default. They will depend on the material you assigned to your part in the design phase. In this tutorial the default values will be kept but in your projects be sure to assign correct materials from the beginning.

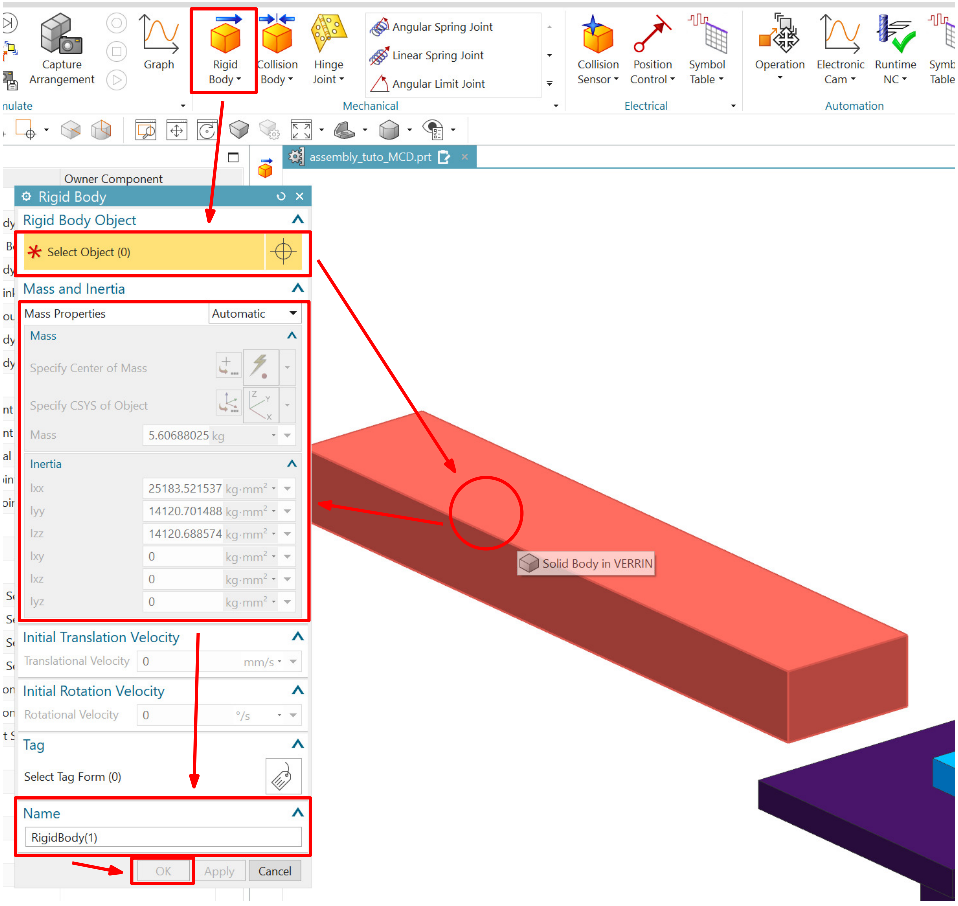


Figure 3: Creating a rigid body

Important : always give a clear name to each of your body, element, operation,... This is how you will manage to have a numerical model that is very easy to understand and to modify!

2. **Collision bodies** don't have to be a whole part, it can often be just one face. To create one (you can visualize the steps on figure 4):

   - click on **Collision body**
   - choose the face(s) that will enter contact with other parts
   - choose the appropriate collision shape (simple shapes usually work better i.e. box and cylinder. If the shape is too complex you can either choose convex, multi-convex or mesh for the most complex parts).
   - make sure the "Highlight on Collision" box is ticked, it will allow you to visualize if the contact is detected correctly
   - don't forget to give an appropriate name (ex: CollisionBody_actuator)
   - Click on OK

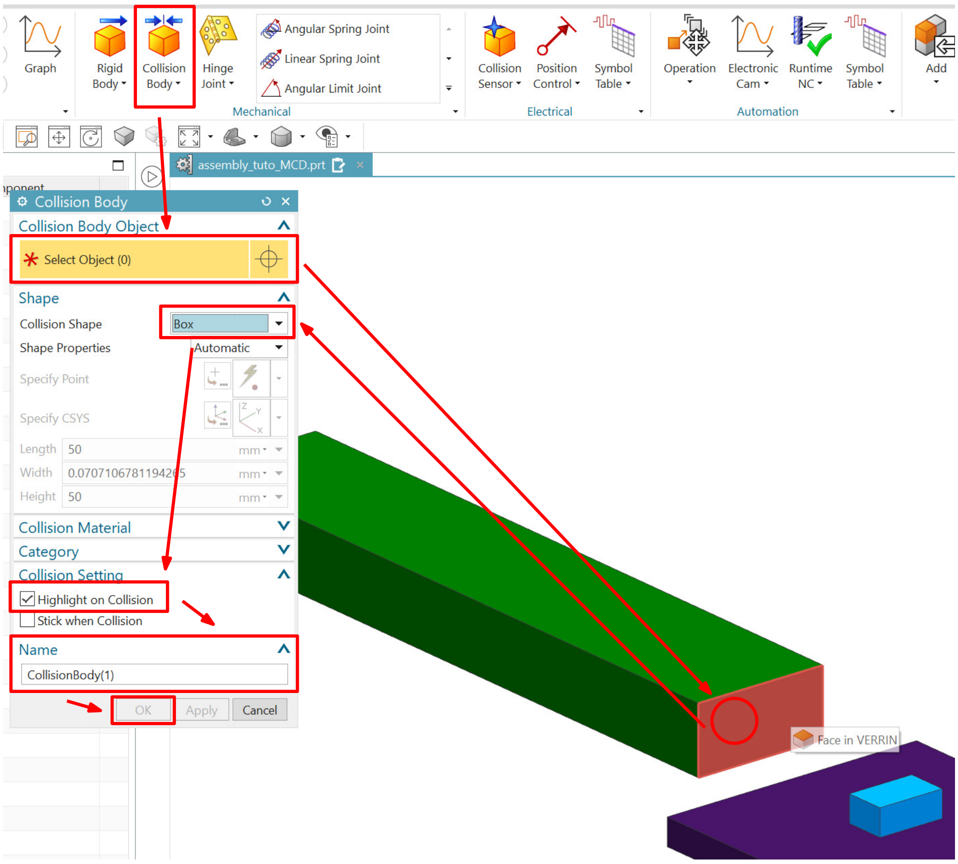


Figure 4: Creating a collision body

Note: the collision material is not discussed here but for some specific materials (rubbers,..) you can change the material properties by creating a new collision material under the **Collision material** option when creating a collision body.

3. In this assembly 5 rigid bodies and 11 collision bodies are defined, the used faces for the collision bodies are highlighted in orange in figure 5. You should be able to define them yourself with what was explained earlier.

4. The part named "Collision_sensor" is a very thin cylinder that is just used to detect a contact after the turntable has turned 180°. This part is only used as a collision body but not as a rigid body.

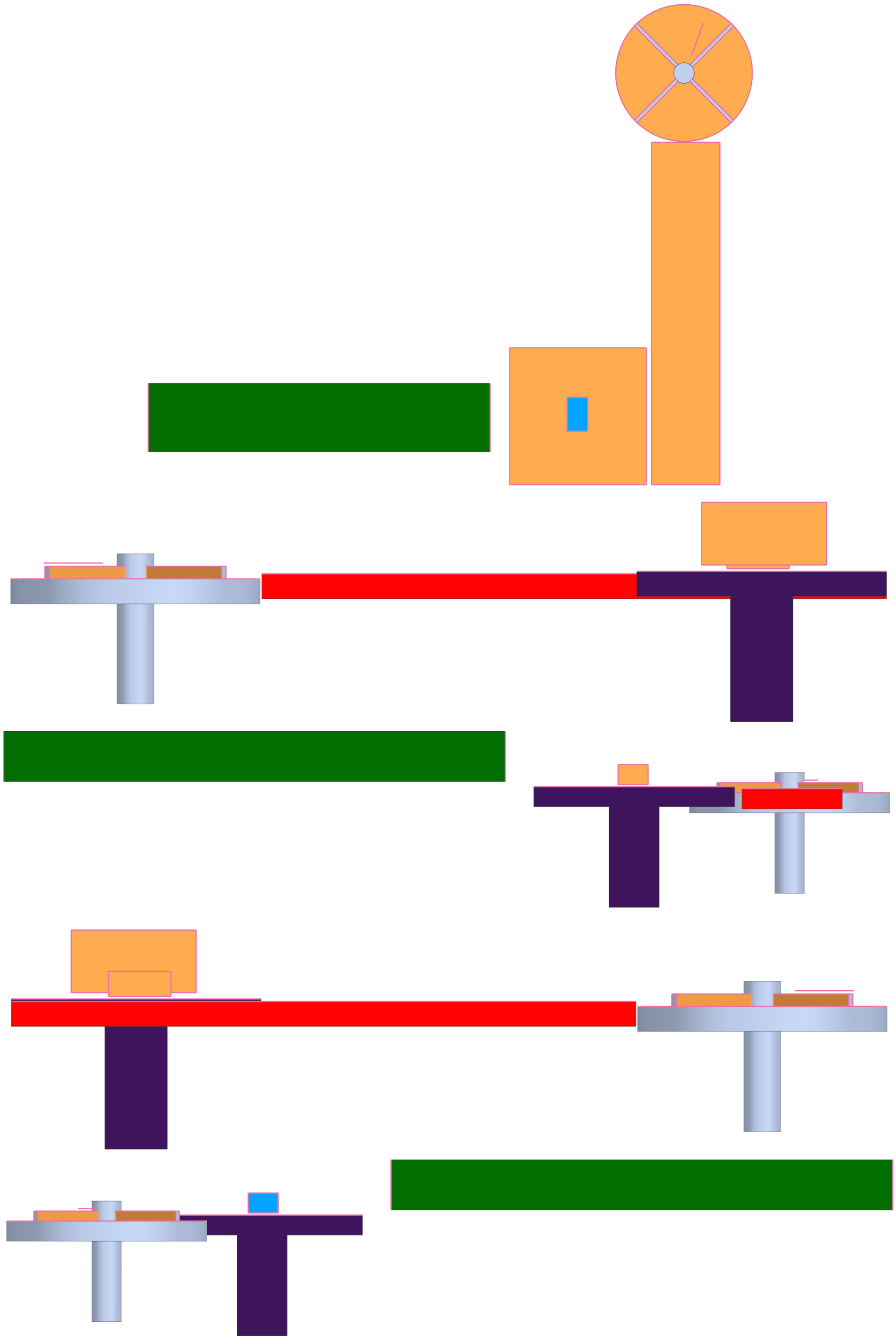


Figure 5: Faces used for the collision bodies, in order: top view, rear view, right view, front view, left view

3.1.2 Joints in NX MCD

The movement of the rigid bodies have to be described by **joints motion**. you have to identify which type of joint is suited for your application.

The steps for defining a joint are shown on figure 6 for the **sliding joint** of the actuator.

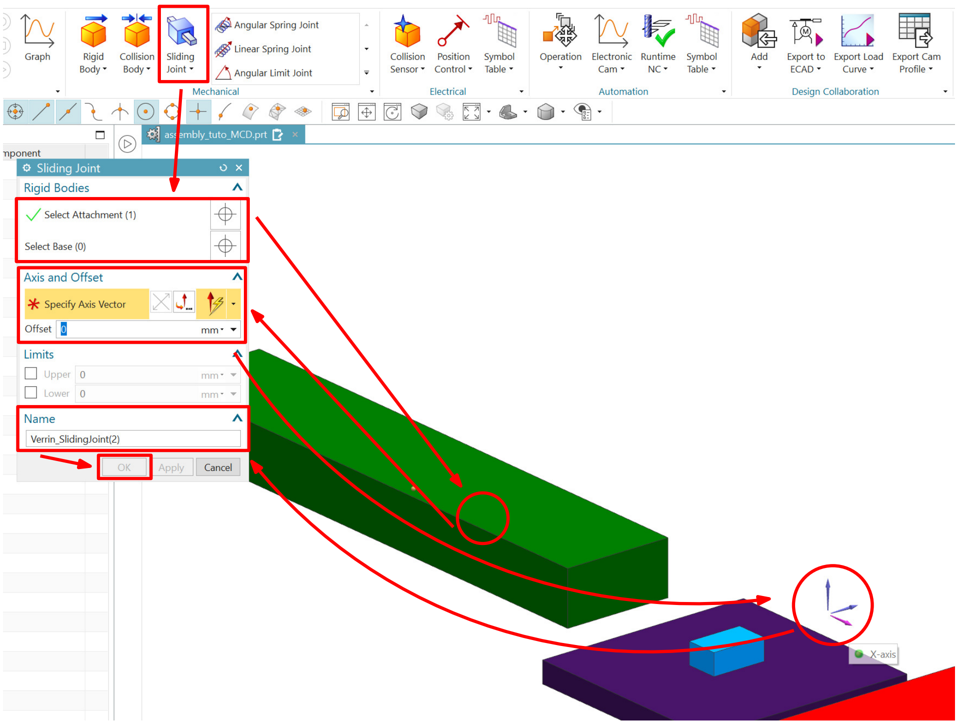


Figure 6: Creating a sliding joint

- Click on the joint button and choose the sliding joint
- Click on the body whose movement has to be restricted (if the body is alone, you just have to choose an attachment and no base, if you are linking two bodies together with a joint, then you need to choose both)
- Choose the axis along which the part can move (you can define the axis yourself through different proposed methods, like 2 points for example, or choose a predefined axis, proposed by the blue arrows). Here we choose the x-axis
- Give a clear name to your joint
- Click on OK

The table and the conveyor have to be assigned a **Fixed joint** since they don't move at all. They are not fixed to a specific part so it's not necessary to specify a base. The steps are shown on figure 7.

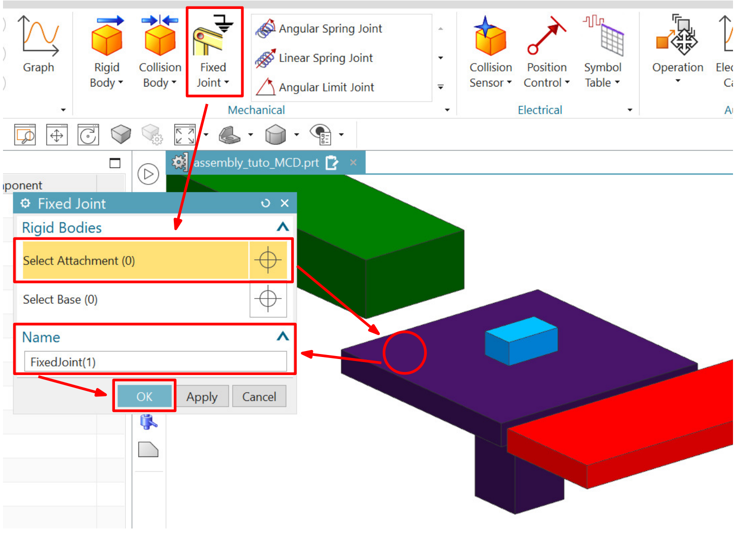


Figure 7: Creating a fixed joint

The turntable is defined by a **Cylindrical joint** (figure 8). The steps are identical to the sliding joint but an anchor point has to be defined to represent the point by which the rotation axis passes.

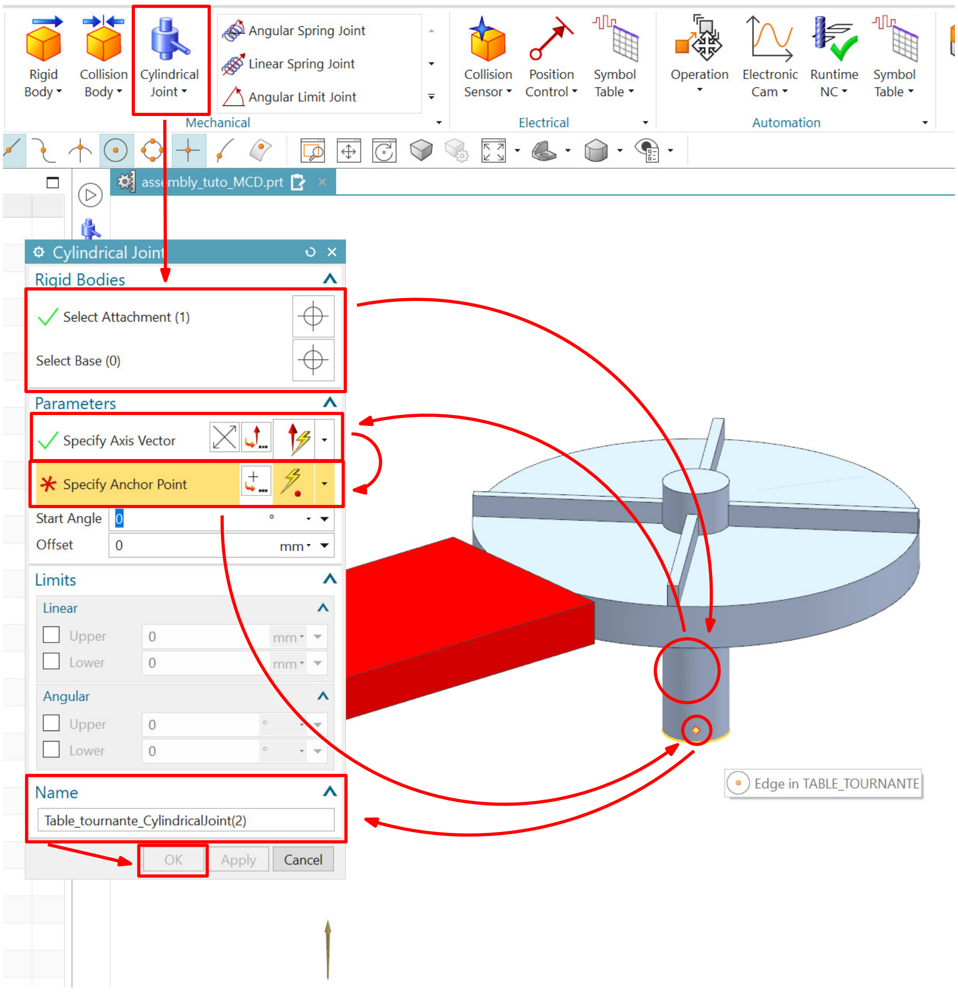


Figure 8: Creating a cylindrical joint

Important : With that cylindrical joint, the turntable can still fall vertically, we need to add a **Planar Joint** to prevent that. That planar joint only requires to choose the bottom face of the turntable and give it an upward facing vector. You should be able to do it yourself following the same steps as for the previous joints.

3.1.3 Position and speed controls

Now that the joints are defined, movements can be implemented by applying a **Position control** or **Speed control** to those joints.

The steps for creating a **speed control** are shown in figure 9.

- Click on speed control
- Choose the joint that will impose a movement
- Choose the type of movement (linear or angular)
- You can already define a speed but it is not necessary because that speed will be given afterwards when defining the series of operations, it is recommended to put a speed of 0 mm/s
- Give a clear name to the speed control indicating on which joint it is applied
- Click on OK

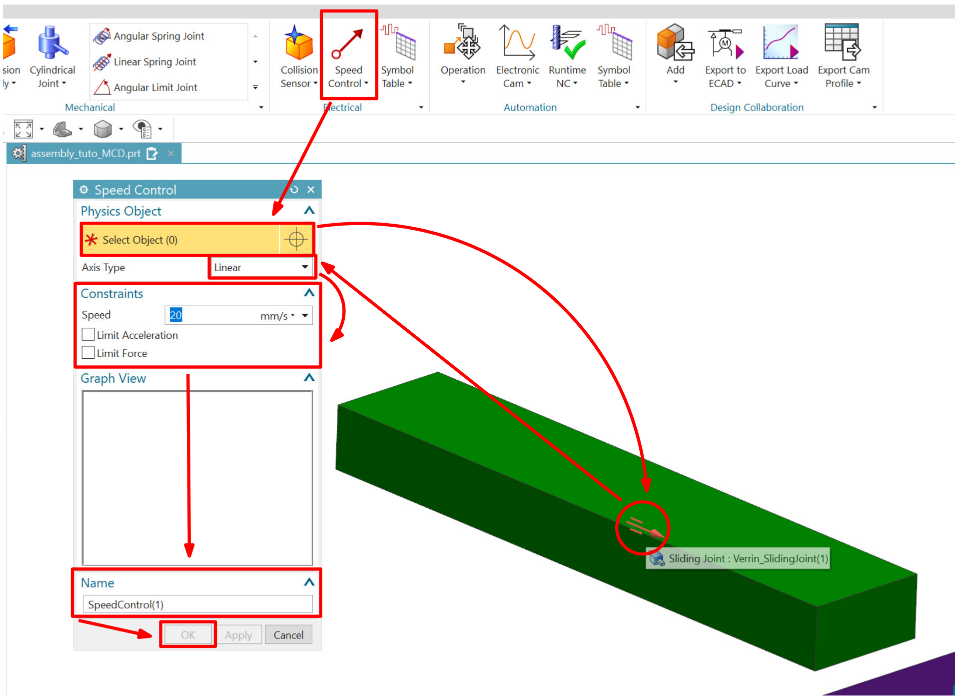


Figure 9: Creating a speed control

Create the same speed control for the **turntable**'s cylindrical joint. **Don't forget to put the axis type to "angular"!**

Information : a **Position Control** is defined in the exact same way with the addition of giving a distance along which the body will move.

3.1.4 Transport surface

It is possible in NX MCD to define a surface as if it was a real conveyor. The part touching the surface will be transported in the desired direction. This is done by creating a **transport surface**.

The steps for creating a transport surface are shown in figure 10.

- Click on transport surface
- Choose the face on which the parts will move
- Choose the motion type (here the conveyor is straight)
- Choose the direction of the movement
- You can define the parameters of the transporting movement but they can be imposed later when the operations are defined
- Choose a clear name
- Click on OK

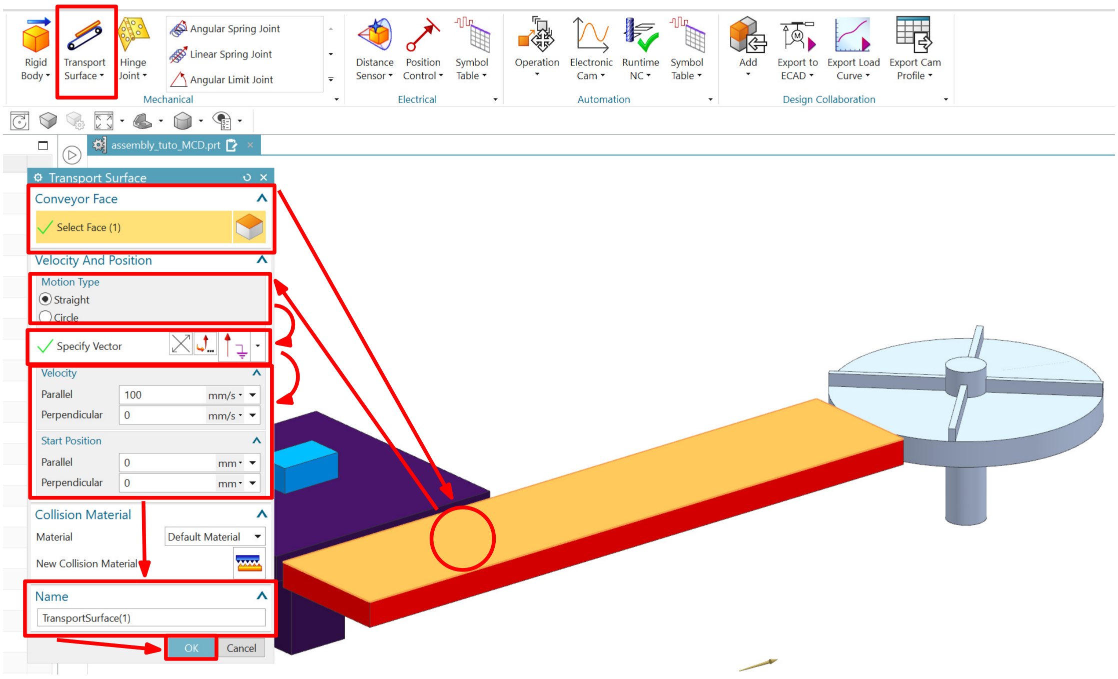


Figure 10: Creating a transport surface

3.1.5 Collision sensors and distance sensors

As said in the introduction, NX MCD allows to make different operations follow each other and apply conditions to those operations. These conditions can be for example "if a collision is detected on the turntable, activate the speed control of that turntable". These conditions are often based on whether a **sensor** is triggered or not. Two types of sensors are presented here:

1. **Collision sensor** : defined on a body, it takes the value "true" or "false" when a collision is detected.
   The steps to create the collision sensor on the turntable are shown on figure 11.

- Click on Collision sensor
- Choose the body that will serve as a collision sensor (here, the small cylinder)
- Choose the shape of the collision sensor (choose the simplest shape if the body has a simple geometry), here choose **Cylinder**.
  The shape properties can be changed if you want to displace the collision area but it is not necessary here since the body is placed specifically as a collision sensor.
- The box "highlight on collision" allows to visualize the contact detection
- Choose a clear name for your sensor (ex: CollisionSensor_sink, you will understand the meaning of "sink" in the next section).
- Click on OK

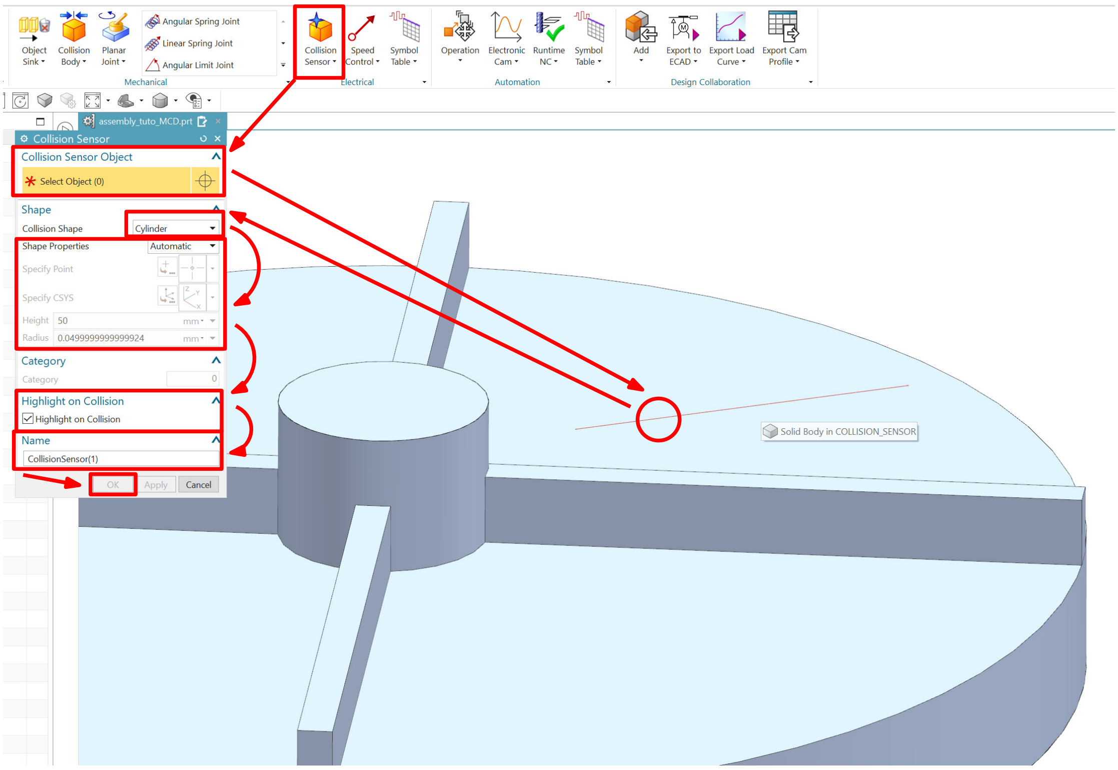


Figure 11: Creating a collision sensor

- Repeat the operation to create a collision sensor on the table plate itself

2. **Distance sensor** : defined on a body or not, it takes the value "true" or "false" when an object enters the detection area. In this scenario, we want a distance sensor to define when to stop the actuator's movement.
   The steps to create the distance sensor that stops the actuator are shown on figure 12.

   - Click on Distance Sensor
   - Choose the point you want to start from to create the sensor (here choose the point on the conveyor as shown on the figure)
   - You want the distance sensor to detect when the actuator passes through, the point we chose is too low, we need to change the coordinates. You can easily do this by clicking on the point dialog box. Choose the coordinate (-322,-155,50) for example (the important part is to make sure the sensor detects the actuator at a correct location)

- Choose the vector direction (here choose the Y-axis)
- You want the "ray" to be very thin so choose a very small opening angle, the range just has to be long enough to be in the actuator's path.
- Choose a clear name for your sensor (ex: DistanceSensor_front_actuator)
- Click on OK

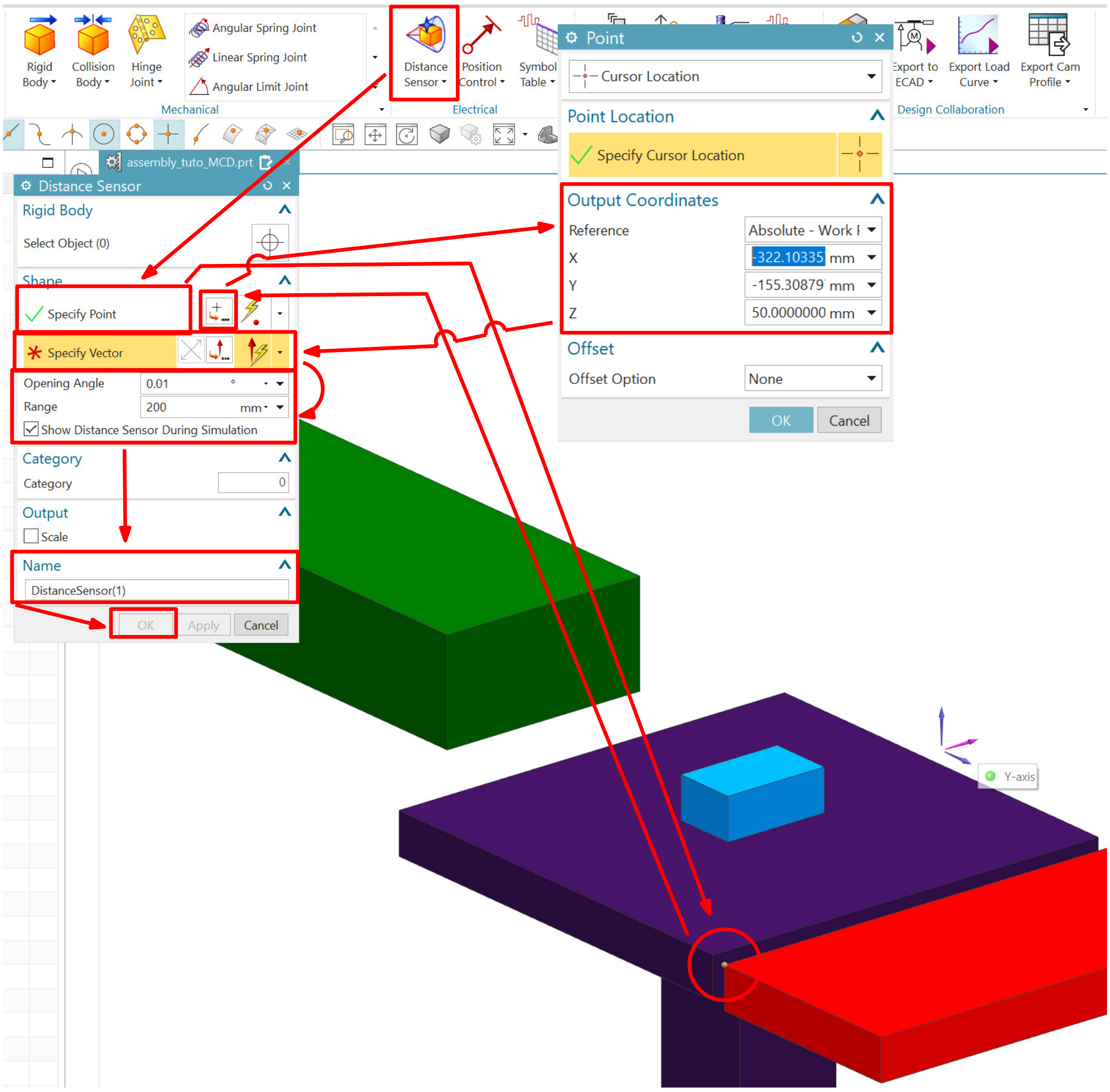


Figure 12: Creating a distance sensor

- Create another distance sensor at the rear of the actuator (you can just choose a corner of the actuator)

3.1.6 Source and Sink

NX MCD allows to make some part appear and disappear during the animation, this is done by using an **Object Source** and **Object Sink**.

1. An **object source** is created simply by clicking on the option and choosing an object that will reappear either at every activation of the source or at every bit of given time (figure 13). In this tutorial, choose the box as the source and choose "once per activation".

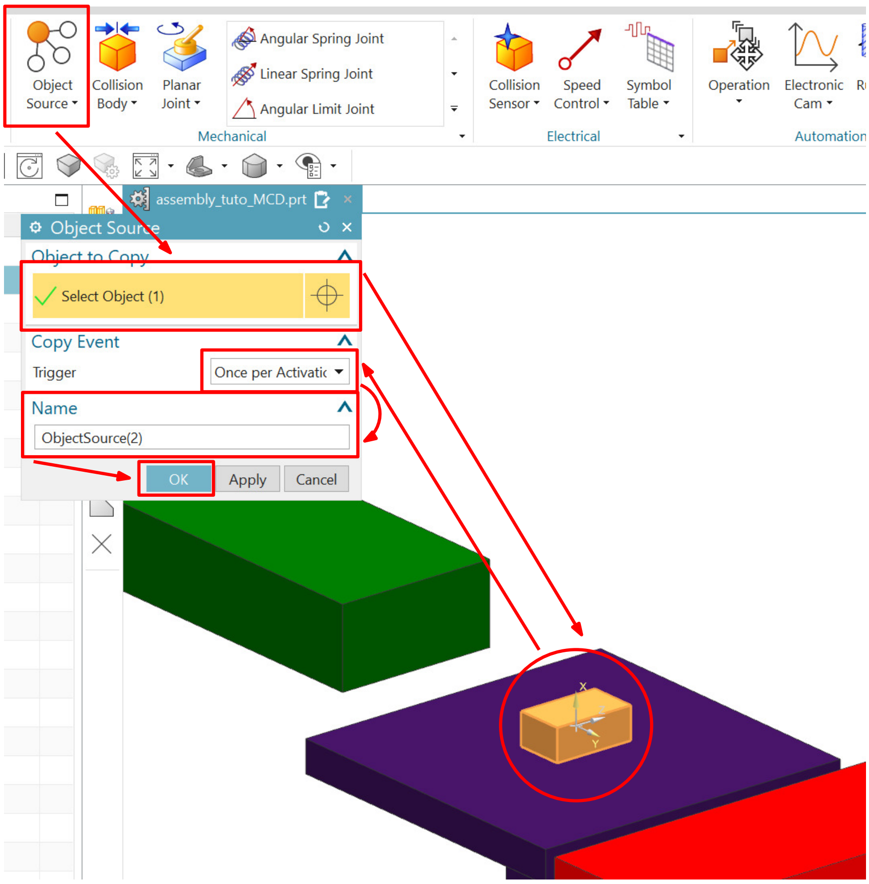


Figure 13: Creating an object source

2. An **Object sink** is created in the same manner but it has to be applied on a **collision sensor** and linked to the source you previously created. In this tutorial, choose the collision sensor your created on the thin cylinder above the turntable (*collision_sensor.prt*).

3.2 Create a series of operations

3.2.1 Create an operation with a condition

NX MCD allows to create different operations and make them follow each other. The steps to create one operation are shown in figure 14. The first operation to create is the movement of the actuator to push the box on the conveyor.

- Go to the sequence editor window
- Right click in the empty space and choose "Add Operation"
- Select the object on which you want to create an operation (movement have to be applied to position or speed controls), here choose the speed control of the actuator
- To modify a parameter, tick the box under "set", here we want the actuator to move at a speed of **100mm/s** during **5s**.
  If it was a **Position control** you would have to give a distance. If you also impose a speed, the time will be automatically computed, if you want to impose a time duration, you need to **right click on the speed number and choose "auto calculate"**, this will allow a time duration to be imposed.

- We want the actuator to move only if there is a box, we thus need to check if the *Object Source* is activated. Choose the object source as a condition object (by clicking on it in the physics navigator window), then choose that object source to be "true"

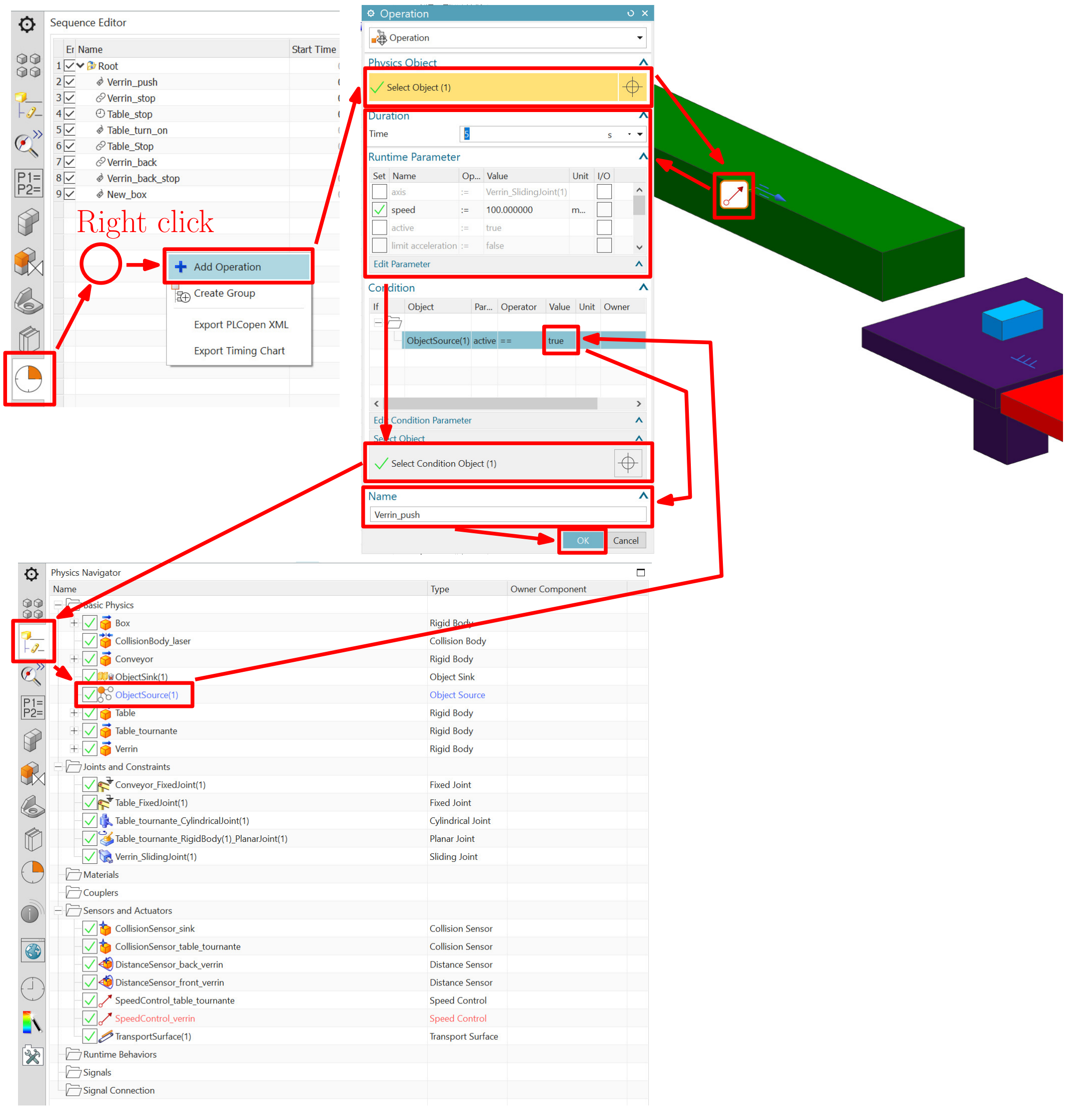


Figure 14: Creating an operation

3.2.2 Create an operation to stop a movement after a certain condition is met

Two different options are available to limit a movement:

1. Use a position control and impose the exact distances (if you know them)
2. Use a speed control and create a collision sensor or a distance sensor where you want your part to stop

The choice between the two depends on what data you possess. If you know exactly the distances then it's easy but sometime you don't have those distances or you need to measure them. Imposing a sensor at a precise location allows to be sure the movement will be correctly limited to a precise location even if the dimensions or distances change.

An example of an operation to stop the actuator when it meets the distance sensor in front of it is shown in figure 15.

- Create an operation as explained before
- Impose the speed to be 0 mm/s
- Choose the distance sensor as condition object
- Impose it to be true
- Choose a clear name for the operation
- Click on OK

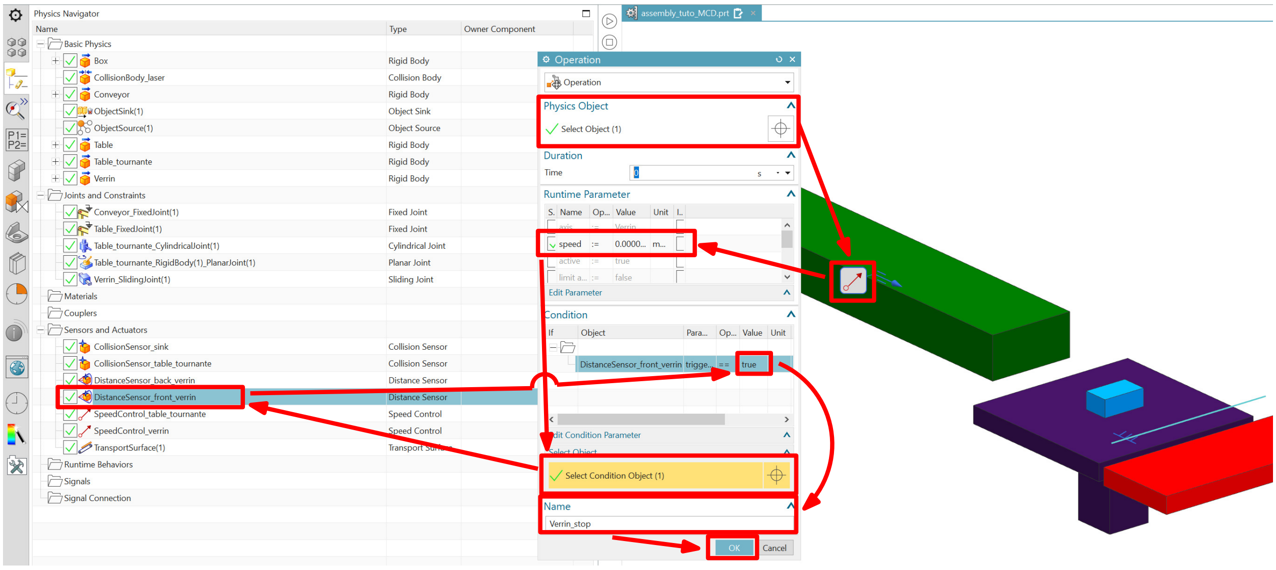


Figure 15: Creating an operation to stop a movement

3.2.3 Link two operations together

A big interest of using NX MCD is to make operations follow directly one after another. This is done by creating a **linker** by choosing both the operations, **right cliking** and choosing "Create linker" (figure 16).

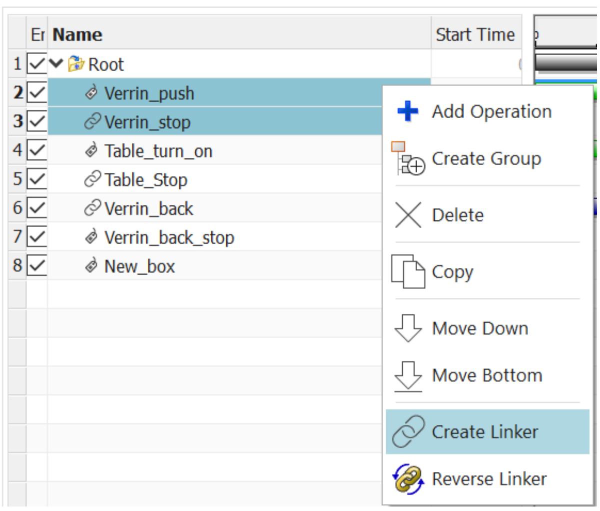


Figure 16: Creating a linker between two operations

To delete the linker, simply right click on the line in the operation window and choose "delete linker" (figure 17).

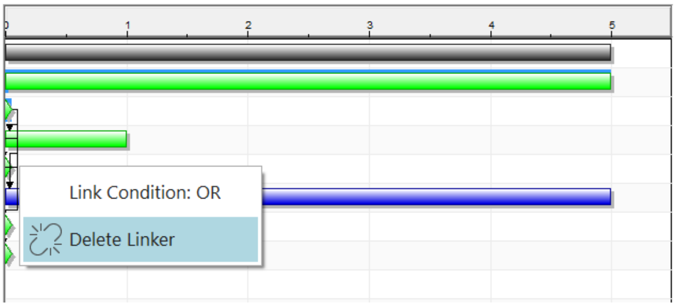


Figure 17: Creating a linker between two operations

3.3 List of operations for this tutorial

With the explanation above, you should be able to create the following list of operations:

1. Forward movement of the actuator IF the object source is active
2. In parallel, operation to stop the movement of the actuator when the front distance sensor is triggered
3. Backward movement of the actuator
4. In parallel, operation to stop the movement of the actuator when the rear distance sensor is triggered
5. Start turning the table IF the collision sensor on the turntable is triggered (speed = -90°/s during 1s)
6. Stop the table movement when the object sink is triggered
7. Activate the object source IF the object sink is triggered

For a clearer understanding of the linkers to create, refer yourself to figure 18.

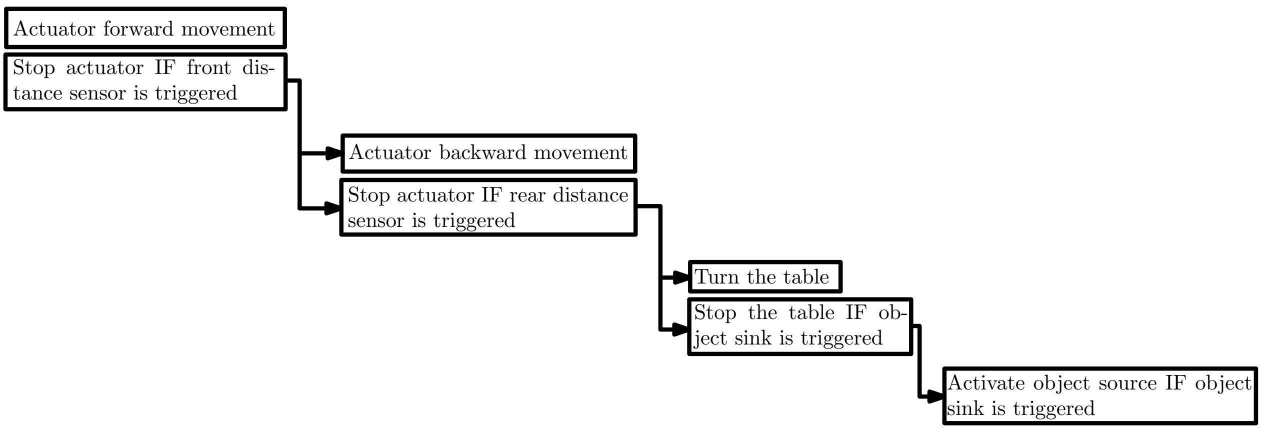


Figure 18: List of operations

You should then obtain a loop where the whole series is restarted each time the object sink/-source are activated.