Mechanics

Simple pendulum

Introduction

What is the value of the oscillation period of a pendulum? Does the value of the mass have an impact on this period? What is the impact of the length? Let's use system simulation in order to predict the behavior of a simple pendulum.

Then, let's consider a physical application with the example of a bridge crane. We can first represent it as a simple pendulum. We can also improve the accuracy of the model, representing the chain as a set of bodies and pivot junction. It results in a multiple pendulum.

Using system simulation, we can also represent the translation of the bridge crane and evaluate the dynamic behavior of the system.

Theory

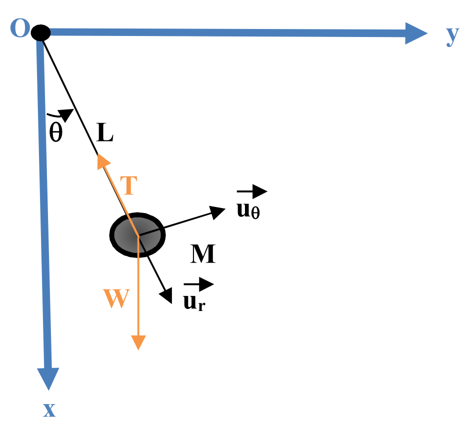


Figure 1: Simple pendulum system schematic

Let's consider a pendulum of mass M rotating around a fixed point O. The distance between O and the mass is L.

We can define a mobile frame with 2 vectors:

- $\overrightarrow{u_r}$: the radial vector
- $\overrightarrow{u_\theta}$: the ortho radial vector

The position of the mass can be then defined by:

- The distance L
- The angle θ

2 forces are applied on the mass:

- The tension of the wire: T
- The weight: W

We can write:

$$\sum \overrightarrow{F_{ext}} = \vec{W} + \vec{T} = m * \vec{a}$$

$$\vec{W} \begin{vmatrix} m * g * cos(\theta) \\ -m * g * sin(\theta) \end{vmatrix}$$

$$\vec{T} \begin{vmatrix} -T \\ 0 \end{vmatrix}$$

The position of the mass can be defines as: $\overrightarrow{OM} = L * \overrightarrow{u_r}$

The mass velocity can then be written: $\vec{v} = \frac{d\overrightarrow{OM}}{dt} = \frac{dL}{dt} * \overrightarrow{u_r} + L * \frac{d\overrightarrow{u_r}}{dt}$

As L is constant, $\vec{v} = \ L * \frac{d\overrightarrow{u_r}}{dt}$

With $\frac{d\overrightarrow{u_r}}{dt} = \frac{d\overrightarrow{u_r}}{d\theta} * \frac{d\theta}{dt}$

As $\overrightarrow{u_r} = \cos(\theta)\,\overrightarrow{u_x} + \sin(\theta)\overrightarrow{u_y}$, then $\frac{d\overrightarrow{u_r}}{d\theta} = -\sin(\theta)\,\overrightarrow{u_x} + \cos(\theta)\,\overrightarrow{u_y} = \overrightarrow{u_\theta}$

Then, $\vec{v} = L * \frac{d\theta}{dt} * \overrightarrow{u_\theta}$

The mass acceleration is then: $\vec{a} = \frac{d\vec{v}}{dt} \ = \ L * \frac{d^2\theta}{dt^2} * \overrightarrow{u_\theta} + L * \frac{d\theta}{dt} * \frac{d\overrightarrow{u_\theta}}{dt}$

With $\frac{d\overrightarrow{u_\theta}}{dt} = \frac{d\overrightarrow{u_\theta}}{d\theta} * \frac{d\theta}{dt}$

As $\overrightarrow{u_\theta} = -\sin(\theta)\,\overrightarrow{u_x} + \cos(\theta)\,\overrightarrow{u_y}$, then $\frac{d\overrightarrow{u_\theta}}{d\theta} = -cos\overrightarrow{u_x} - \sin(\theta)\,\overrightarrow{u_y} = -\overrightarrow{u_r}$

Then, $\vec{a} = L * \frac{d^2\theta}{dt^2} * \overrightarrow{u_\theta} - L * \left(\frac{d\theta}{dt}\right)^2 * \overrightarrow{u_r}$

$$\vec{a} \begin{vmatrix} -L * \left(\frac{d\theta}{dt}\right)^2 \\ L * \frac{d^2\theta}{dt^2} \end{vmatrix}$$

Finally: $\begin{cases} m * g * cos(\theta) - T = -m * -L * \left(\frac{d\theta}{dt}\right)^2 \\ -\cancel{m *} g * sin(\theta) = \cancel{m *} L * \frac{d^2\theta}{dt^2} \end{cases}$

For small oscillations (small values for θ), we can approximate: sin(θ)=θ.
Then, the second equation becomes: $\frac{g}{L} * \theta + \frac{d^2\theta}{dt^2} = 0$

A solution of this equation is $\theta(t) = \theta_0 * cos(\omega_0 * t)$ with $\omega_0 = \frac{g}{L}$
With a period: $\boxed{T_0 = \frac{2*\pi}{\omega_0} = 2 * \pi * \sqrt{\frac{L}{g}}}$

We notice that this period does not depend on the mass value. It only depends on the value of the length L.

For instance, if L = 1m, the period will be close to 2 seconds, the Frequency close to 0.5 Hz.

Simulation, validation and practice with LMS Amesim

Building the sketch

In sketch mode, the model of the pendulum system (Figure 3) can be built easily and fast, selecting 4 components from the mechanical library (Figure 2):

- The Planar Mechanical assembly tool used to generate the 2D animation and the visualization of the system.
- A reference fixed body for the fixed point of the system
- A pivot junction
- A rigid body with 1 port.

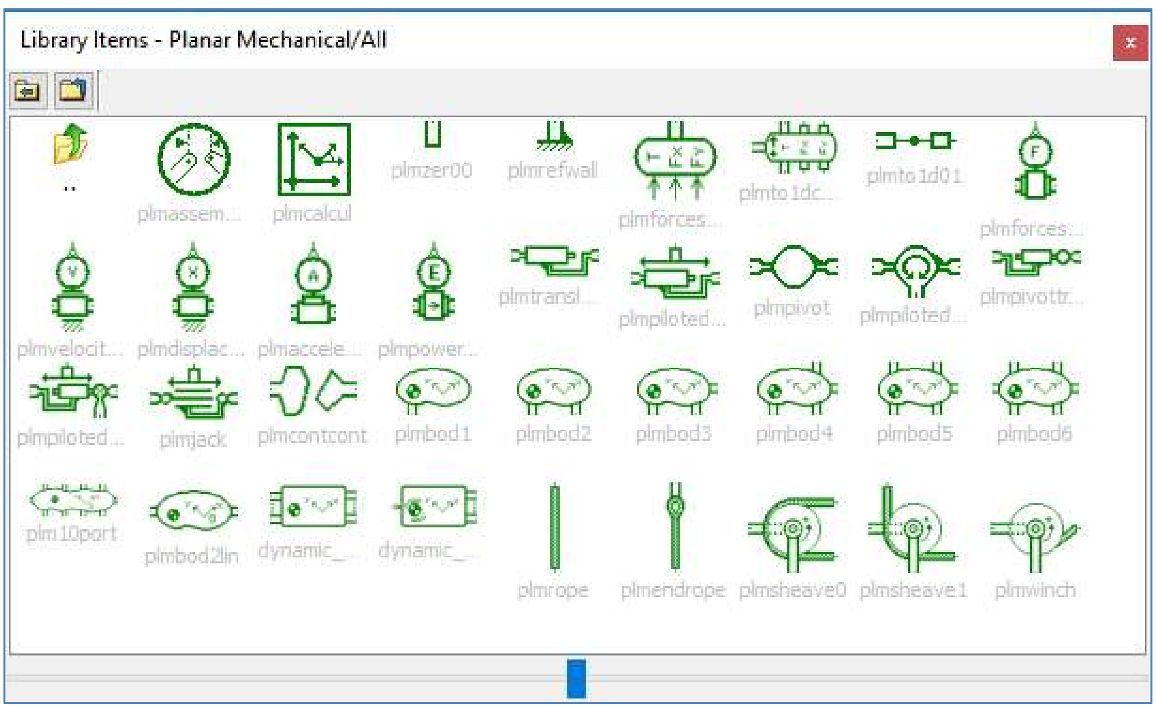


Figure 2: Planar Mechanical Library components

Selecting submodels

In this example, in submodel mode, submodels can be quickly selected using the "premier submodel".

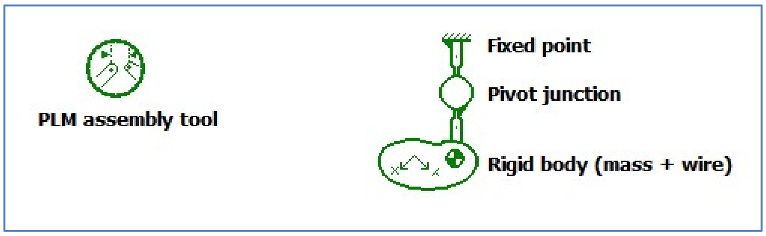


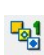
Figure 3: model of the system

Setting parameters
The model is representing a bridge crane (Figure 4) carrying a load. In this 1st step, we assume that the chain remains rigid.

Figure 4: bridge crane

We will consider that we do not have any friction in the system and that:

- M = 1000 kg
- L = 1 m

So, in parameter mode , we will define the following values (default values are kept for other parameters):

- Pivot junction:
  - *Damping coefficient*: 0 Nm/(rev/min)

- Rigid body:
  - *initial absolute angular position*: 20°
  - *G: y position*: -1m
  - *Mass*: 1000 kg

Once parameters are defined, you can check that you have correctly defined your system, thanks to the PLM assembly tool. Just double click on the component in order to open the visualization window (Figure 5).

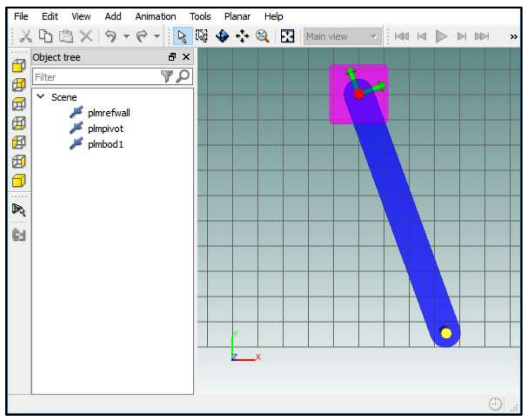


Figure 5: PLM assembly tool window

Running the simulation and analyzing the results

Run the simulation.

When the simulation is performed, you can visualize the dynamic behavior of the pendulum running the animation of the PLM assembly tool.

You can also plot the rigid body angular position. Then, in the plot window, select *Tools* and *FFT* in order to get quickly the oscillations frequency. As calculated from the equations, the frequency is close to 0.5 Hz (Figure 6).

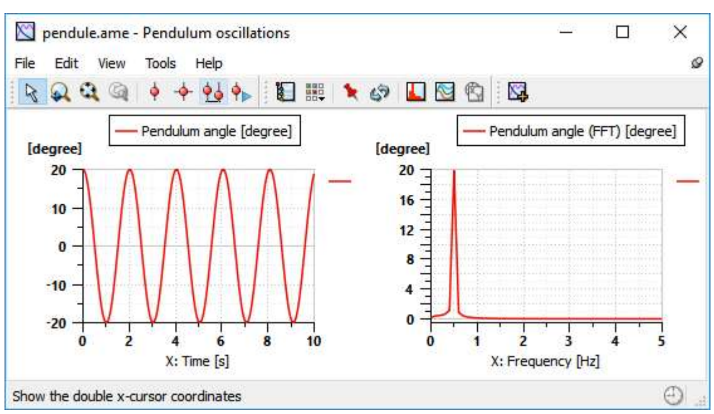


Figure 6: Simulation results

Now, set the rigid body mass as a batch parameter and define 2 values: 1kg and 1000 kg (Figure 7).

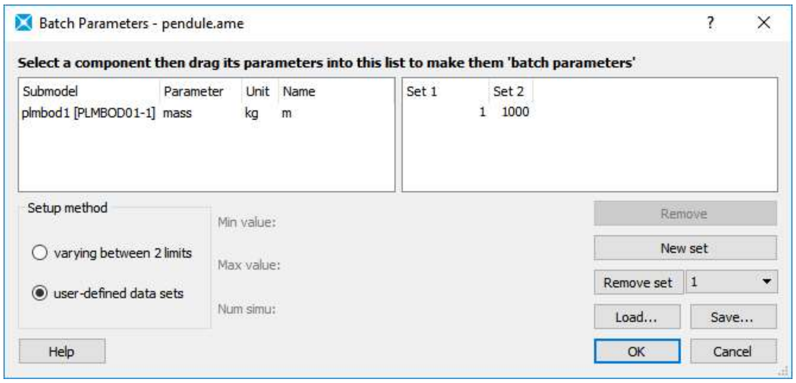


Figure 7: Batch parameters

Perform the batch simulation and compare, on the plot, the behavior of the pendulum with the 2 values (Figure 8).

As expected, the behavior (at least for small oscillations) does not depend on the value of the mass.

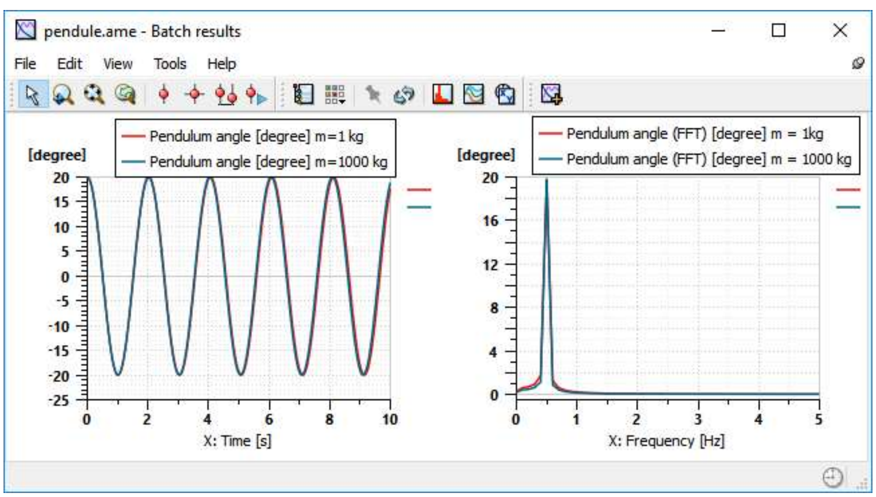


Figure 8: Batch simulation results

Improving the level of accuracy of the model: modeling of the chain

In the 1st step, we assumed that the chain was rigid. Let's now improve the level of accuracy of the model, representing the chain as an assembly of bodies and pivot junctions representing the chain links. We can, for instance discretize the chain into 10 elements, building the following LMS Amesim model ().

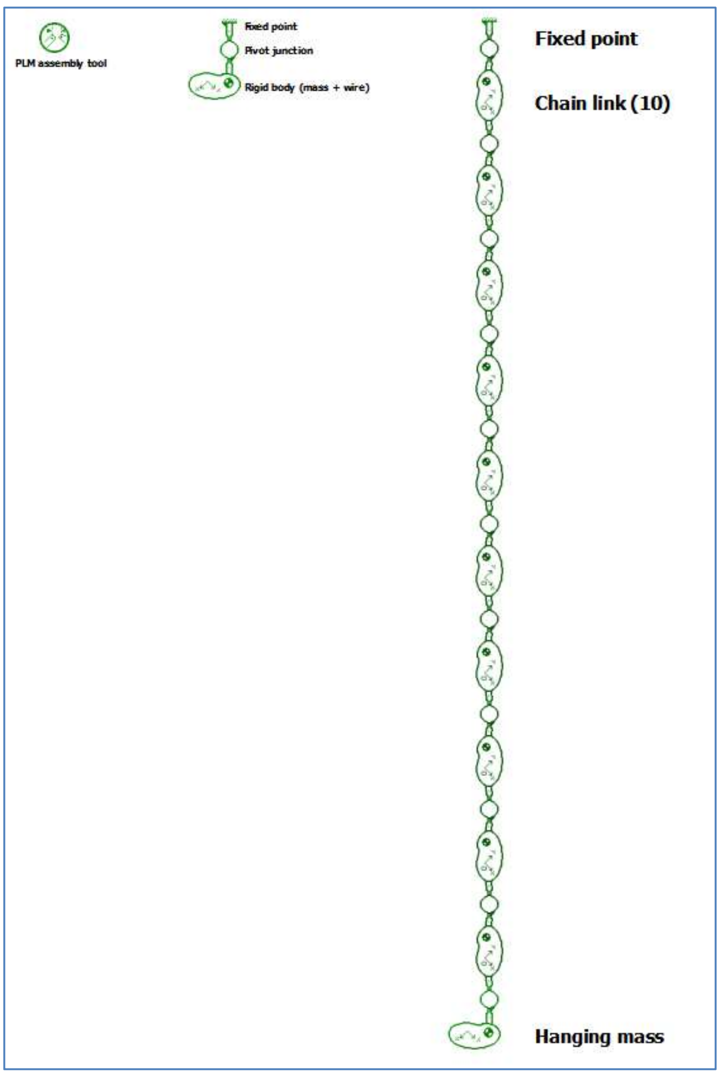


Figure 9: Model including the chain

In parameter mode, define global parameters (Figure 10)in order to define:

- An offset applied between the 2 system models: 20 cm in the x direction
- The length of the chain: 1 m
- The number of chain links: 10
- The chain mass: 1 kg

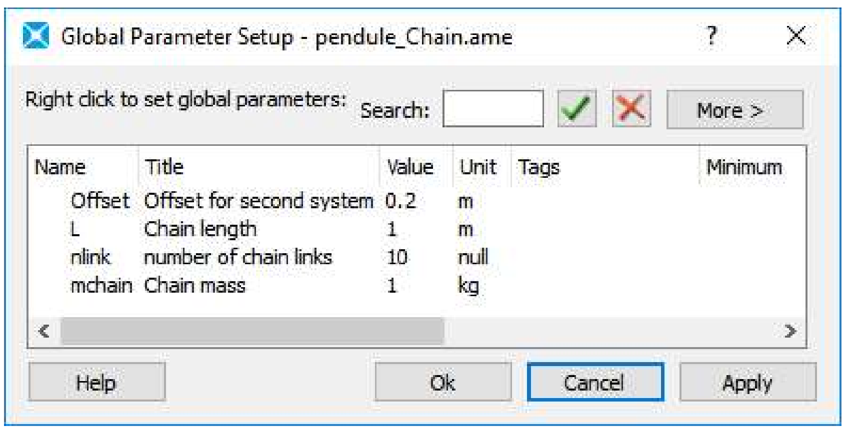


Figure 10: Model including the chain

Select the chain bodies and, using global parameters, set the values of the following parameters:

- *G: y position*: L/nlink/2 m
- *y position at port 2* : L/nlink m
- *mass* : mchain/nlink kg

Other parameters are kept at their default values.

For the hanging mass, set the values of the following parameters:

- *O: initial absolute x position*: Offset+L*sin(20*PI/180)
- *O: initial absolute y position*: -L*cos(20*PI/180)
- *Mass*: 1000 kg

For the fixed point, set the value of the following parameter:

- *absolute x position at port 1*: Offset

Right click on the hanging mass body and select *View lock states.* Lock the state *G: absolute x position*.in order to define, for the PLM assembly tool the constraint to be locked for the system initialization (Figure 11).

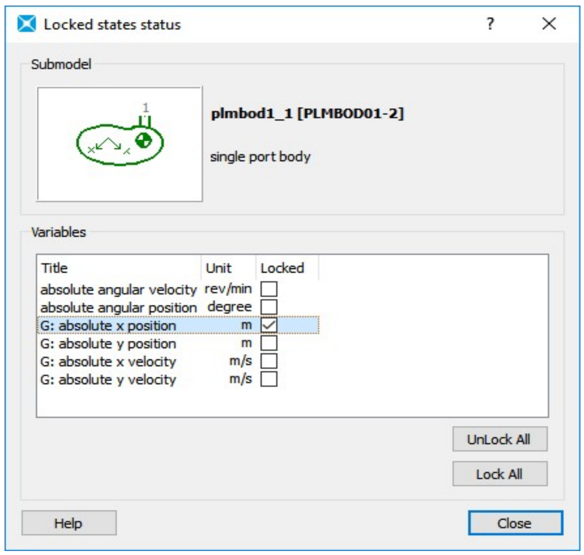


Figure 11: *Locked states status* window

Switch to simulation mode. Launch the run and check the initial position of the system in the the PLM assembly tool animation widow (Figure 12). Run the animation in order to visualize the dynamic behavior of the system

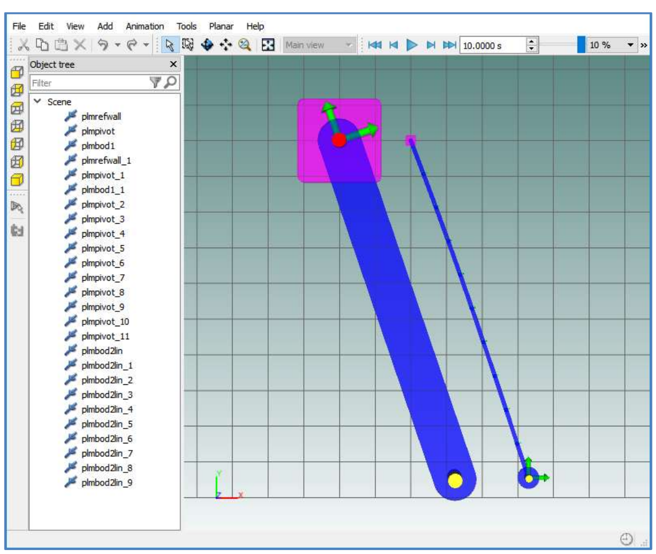


Figure 12: Representation of the chain on the visualization tool

Improving the level of accuracy of the model: adding the translation of the trolley motion

Up to now, we assumed that the chain was attached to a fixed body. Let's now consider the translation of the trolley in the model, adding an additional body for the trolley and an additional junction: a piloted translation junction (Figure 13).

The mechanical port, at the top of this junction component I used to apply a force on the trolley and generate a motion.

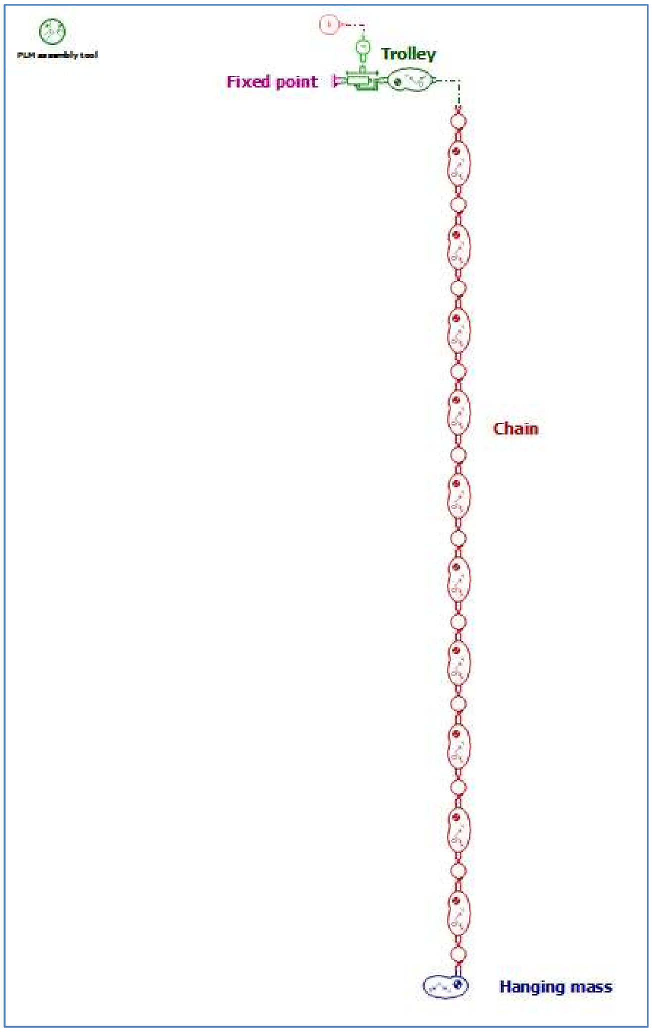


Figure 13: System model adding the trolley translation

Let's assume that the Trolley has a mass of 10 kg and set the body parameter to *O: initial absolute x position* to 0.1 m.
Let's apply a force of 500 N on the trolley.
All the other parameters are kept to their initial values.

We can now run the simulation and visualize the results on the animation.

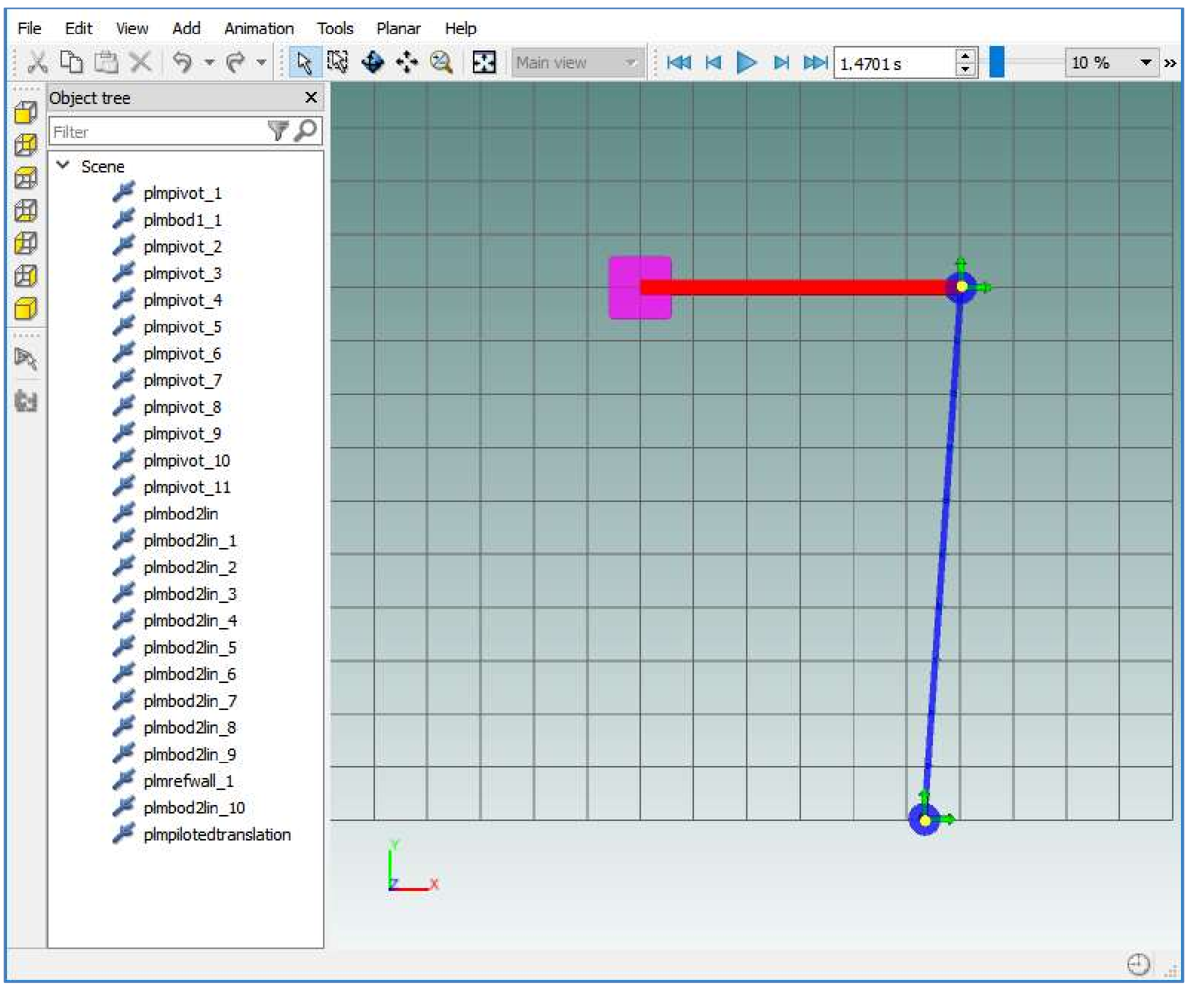


Figure 14: Animation of the model including the translation of the trolley

Improving the level of accuracy of the model: adding the actuation system
In the previous step, we added the translation of the trolley, using a simple piloted translation junction. Let's now model more into detail the actuation of the trolley, using cables and pulleys (Figure 15).
In order to reach this target, we need to select another body model for the trolley in order to connect it to the 2 cables. So we need a component with 4 ports instead of 2.
Regarding the translation junction, we can also change it in order to select a translation junction without the piloting port.
Finally, we need to add the components used to model the pulleys and the cables:

- 2 pulleys
    - Port 1 is connected to a fixed point
    - Ports 2 and 3 are connected to rope components (used to model the cables)
    - Port 4 can be connected to an inertia
- 3 rope components
- 2 rope end components used to connect ropes components with the trolley
- 2 inertias
    - The 1st one connected to a zero torque source
    - The second one to a torque source used to actuate the trolley

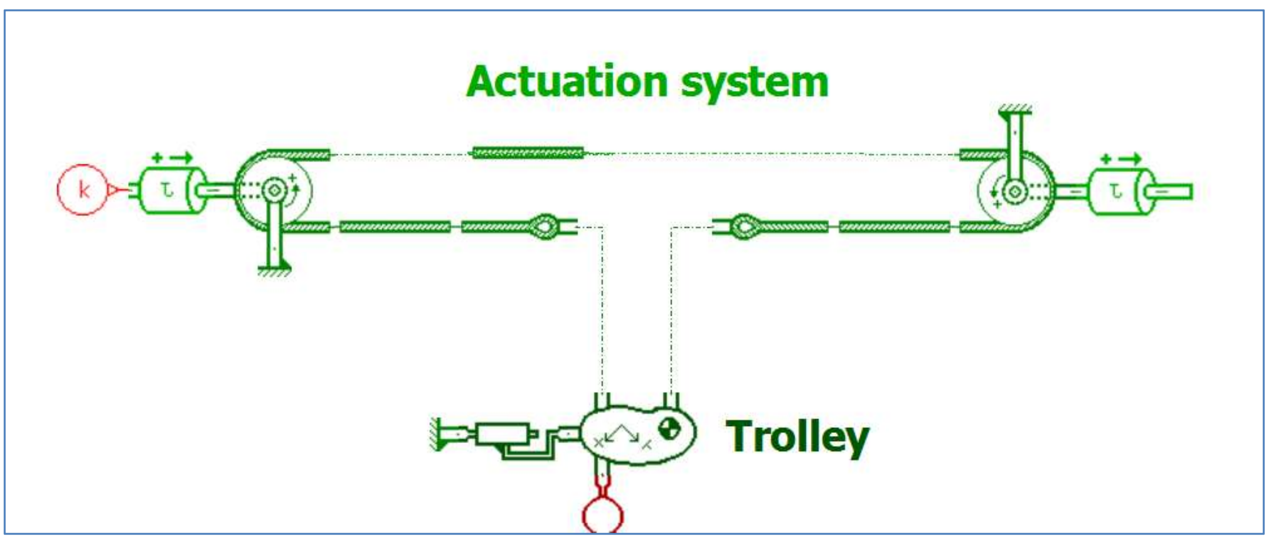


Figure 15: Modeling of the actuation system

We can define a new Global Parameter: Rpul = 0.5/2 m.

We can then set the components parameters as follows:

- Translation junction, ropes, inertias, rope ends: default values
- Pulleys:
    - *diameter*: 2*Rpul m
- Translation junction fixed point:
    - *absolute x position at port 1* : -1 m
    - *absolute y position at port 1* : -Rpul
- Left-side pulley fixed point :
    - *absolute x position at port 1* : -1 m
- Right-side pulley fixed point :
    - *absolute x position at port 1* : 5 m
- Trolley body :
    - *O: initial absolute y position* : -Rpul
- Torque source :
    - *constant value* : 20 N.m

We can now run the simulation and visualize the results on the animation.

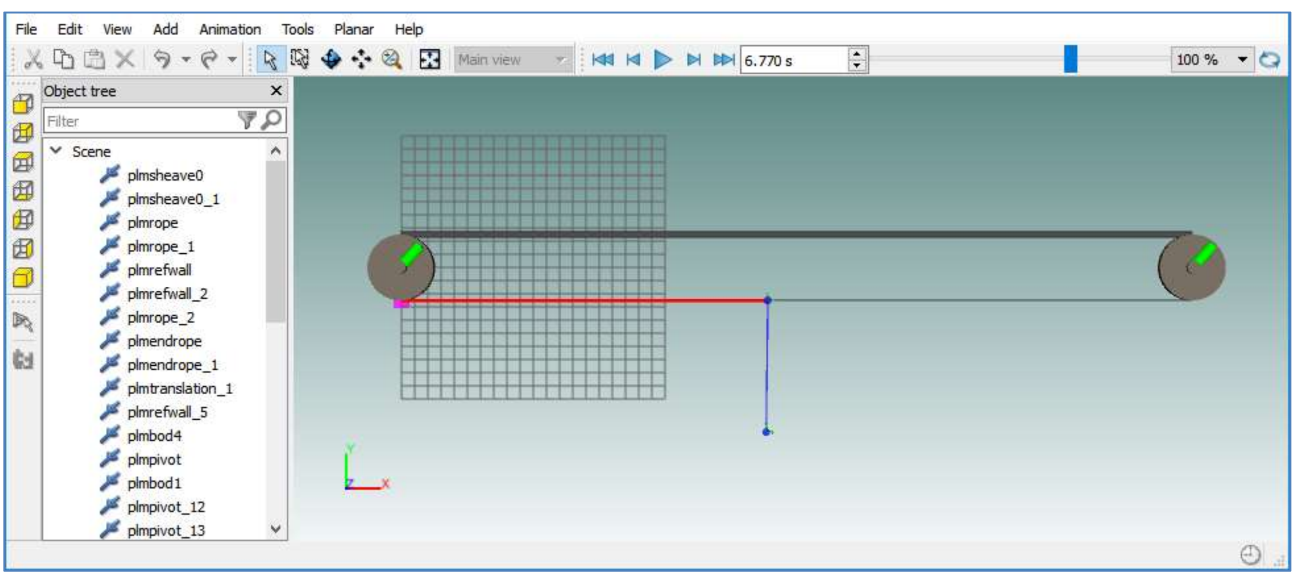


Figure 16: Animation of the model including the actuation of the trolley

Improving the level of accuracy of the model: adding controls
Let's now add, in the model, the control used to manage the position of the trolley (Figure 17). We will use a simple control based on a PI (proportional, integrator).
The position request is set thanks to the *piecewise linear signal source* from the signal library.
A position sensor is also needed.

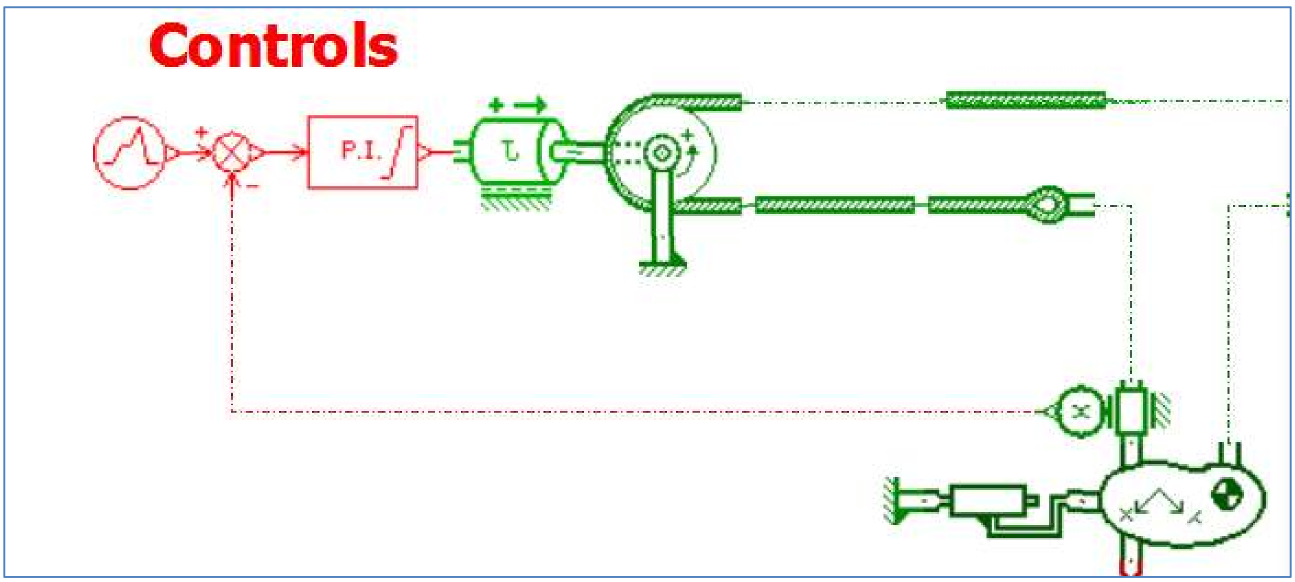


Figure 17: Control model

The *piecewise linear signal source* parameters can be defined as follows:

Title	Value	Unit	Tags
number of stages	**4**		
cyclic	no		
time at which duty cycle starts	0	s	
output at start of stage 1	0	null	
output at end of stage 1	0	null	
duration of stage 1	**1**	s	
output at start of stage 2	0	null	
output at end of stage 2	**4**	null	
duration of stage 2	**15**	s	
output at start of stage 3	**4**	null	
output at end of stage 3	**4**	null	
duration of stage 3	**30**	s	
output at start of stage 4	**4**	null	
output at end of stage 4	**1**	null	
duration of stage 4	**30**	s	

Tune the PI parameters in order to get the best control as possible, balancing the responsiveness and the stability.

If the system is not stable enough, you can change the inertia component, selecting a new one introducing some friction. You can, for instance, set for this inertia the value of *coefficient of viscous friction* to 5 N.m/(rev/min).

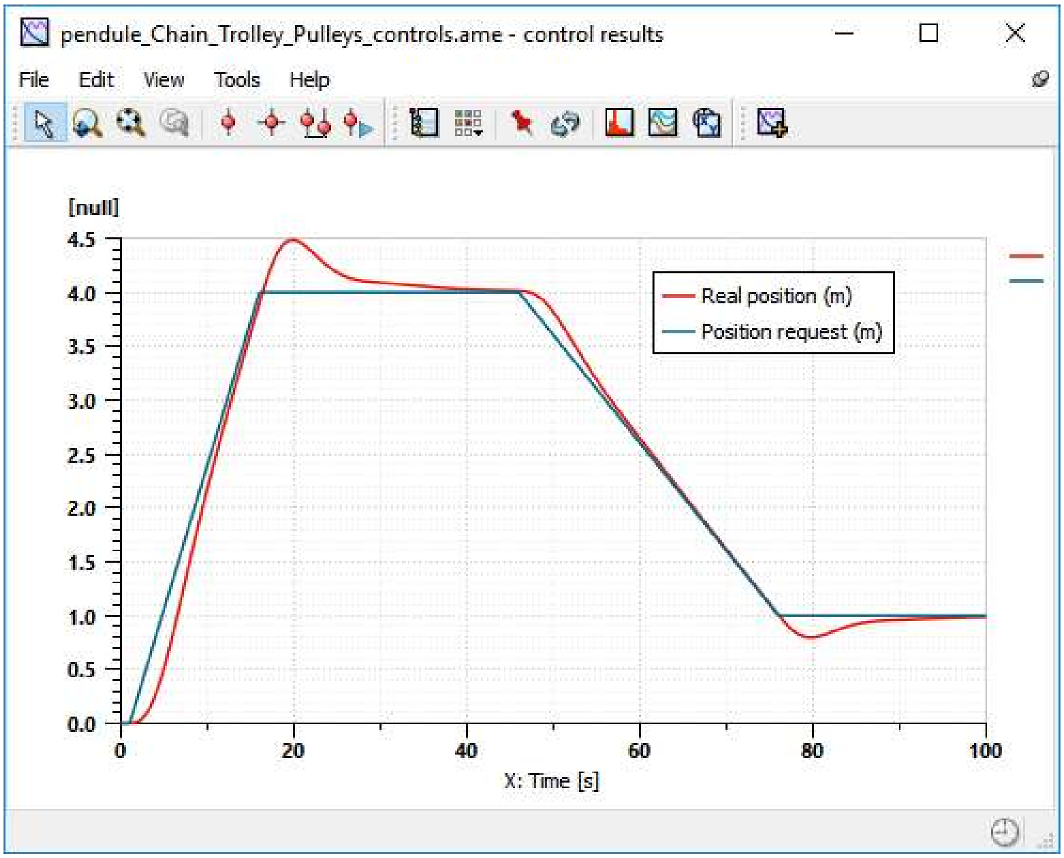


Figure 18: Control results

Summary

With this tutorial, we started from a simple pendulum. We used this model in order to represent a bridge crane.

In order to improve our model, we step by step added details:

- Modeling of the non-rigid chain of the crane.
- Modeling of the trolley translation motion
- Modeling of the trolley actuation with pulleys and cables
- Modeling of the trolley control