Practical Course Matlab/Simulink

PHYSICAL MODELING

Table of Contents

1 Basics: Building a model using SimMechanics

In this exercise, you will learn the basics of *SimMechanics* and how to build a simple *SimMechanics* model. First, you will model a simple pendulum:

Exercise

(1) Open an empty Simulink diagram and add all required blocks for a SimMechanics model (*Solver Configuration*, *World Frame*, *Mechanism Configuration*) and connect them to each other.

(2) Add a Solid block to the diagram. This block represents the rod of the pendulum.
Dimensions of the rod: Length 1m; Height: 0.05m; Width: 0.05m;
Density: 1500 kg/m³

(3) A revolute joint allows the rod to swing. Add a *Revolute Joint* block to the diagram.

(4) Use a *Rigid Transform* block to copy a coordinate frame 5cm ahead of the rod's end. *Pay attention to the correct rotation.*

(5) Connect the *World Frame* to the mechanism. Review the motion using the *Mechanics Explorer* to watch the visualization.

2 Sensing & Actuating Joints

Sensing forces and motion in joints as well as their actuation is important for further analysis or using controllers for example.

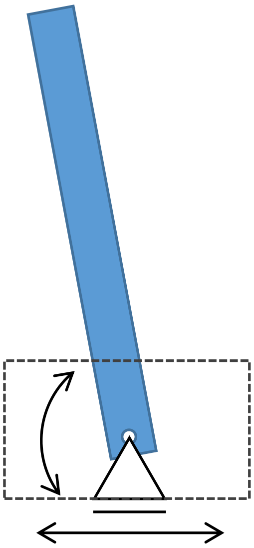
Exercise

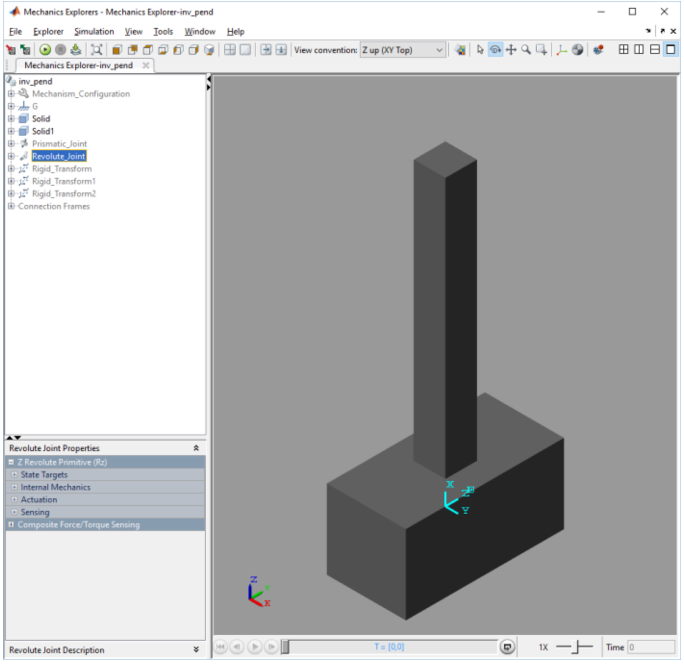
(1) Sense the angular velocity of the pendulum and use a scope to visualize the measurement.

(2) Actuate the *Revolute Joint* block in order to rotate the rod with an angular velocity of 45 deg/s.

3 Inverse Pendulum

In this exercise, you will build a model of an inverse pendulum mounted on a slider. You will implement a controller that keeps the rod balanced at a given slider position.





Exercise

(1) Create a SimMechanics model that represents this mechanical system.
Rod:
Dimensions: Length 1m; Height: 0.05m; Width: 0.05m;
Density: 1000 kg/m³
Distance axes of rotation – end of rod: 5cm
Slider:
Dimensions: Length 0.5m; Height: 0.25m; Width: 0.25m;

Axis of prismatic and rotary motion is located at the center of the slider.

(2) Create sensing ports for position as well as velocity at both joints. Create a force actuation port for the prismatic joint. Add one Simulink In (force) and four Simulink Out (positions, velocities) blocks. Connect them to the corresponding ports.

(3) Controller gain calculation:
Save the model as *inv_pend.slx*.
Now use the following command to linearize the SimMechanics model:

```
linsys = linearize('inv_pend')
```

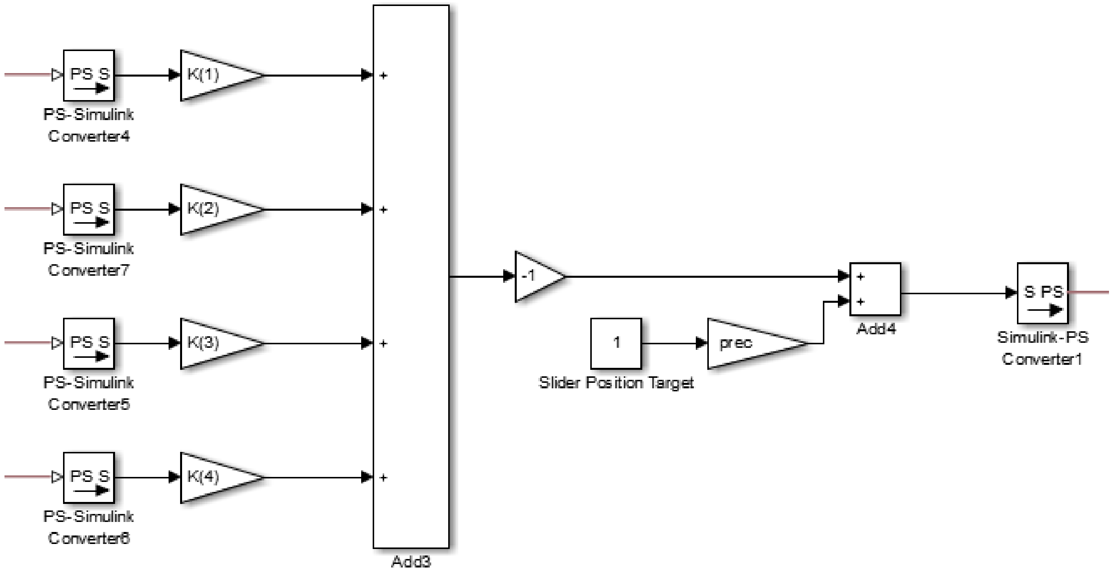
To calculate the gains of the controller, the following command has to be used:

```
K = place(linsys.a, linsys.b, [-6, -6.01, -6.02,-6.03])
```

Finally calculate the feed forward control:

```
c=[1 0 0 0]; prec=((c*(linsys.b*K-linsys.a)^-1)*linsys.b)^-1;
```

(4) Delete all the *In* and *Out* blocks first. Now implement the controller logics and connect the joint's inputs and outputs with the controller logics. Have a look at `linsys(1).StateName` to know what controller input belongs to which joint output.



(5) *Slider Position Target* specifies the target position of the slider at which the rod should be kept balanced. Try several positions to check the controller. It works only within a specific scope.