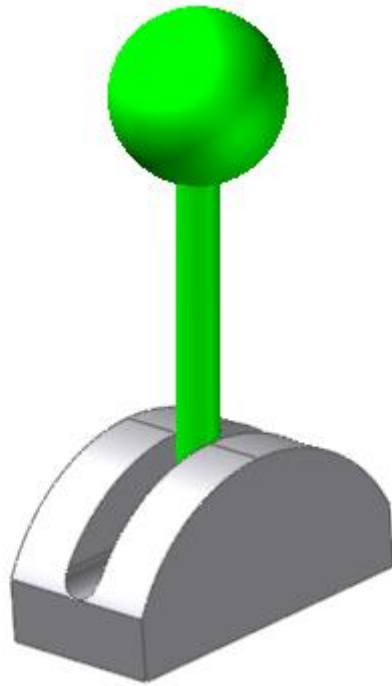




倒立摆教程（**CoLink**）



Copyright © 2017 FunctionBay, Inc. All rights reserved

User and training documentation from FunctionBay, Inc. is subjected to the copyright laws of the Republic of Korea and other countries and is provided under a license agreement that restricts copying, disclosure, and use of such documentation. FunctionBay, Inc. hereby grants to the licensed user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the FunctionBay, Inc. copyright notice and any other proprietary notice provided by FunctionBay, Inc. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of FunctionBay, Inc. and no authorization is granted to make copies for such purpose.

Information described herein is furnished for general information only, is subjected to change without notice, and should not be construed as a warranty or commitment by FunctionBay, Inc. FunctionBay, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the Republic of Korea and other countries. UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

Registered Trademarks of FunctionBay, Inc. or Subsidiary

RecurDyn[™] is a registered trademark of FunctionBay, Inc.

RecurDyn[™]/SOLVER, *RecurDyn*[™]/MODELER, *RecurDyn*[™]/PROCESSNET, *RecurDyn*[™]/AUTODESIGN, *RecurDyn*[™]/COLINK, *RecurDyn*[™]/DURABILITY, *RecurDyn*[™]/FFLEX, *RecurDyn*[™]/RFLEX, *RecurDyn*[™]/RFLEXGEN, *RecurDyn*[™]/LINEAR, *RecurDyn*[™]/EHD(Styer), *RecurDyn*[™]/ECFD_EHD, *RecurDyn*[™]/CONTROL, *RecurDyn*[™]/MESHINTERFACE, *RecurDyn*[™]/PARTICLES, *RecurDyn*[™]/PARTICLEWORKS, *RecurDyn*[™]/ETEMPLATE, *RecurDyn*[™]/BEARING, *RecurDyn*[™]/SPRING, *RecurDyn*[™]/TIRE, *RecurDyn*[™]/TRACK_HM, *RecurDyn*[™]/TRACK_LM, *RecurDyn*[™]/CHAIN, *RecurDyn*[™]/MIT2D, *RecurDyn*[™]/MIT3D, *RecurDyn*[™]/BELT, *RecurDyn*[™]/R2R2D, *RecurDyn*[™]/HAT, *RecurDyn*[™]/曲柄, *RecurDyn*[™]/PISTON, *RecurDyn*[™]/VALVE, *RecurDyn*[™]/TIMINGCHAIN, *RecurDyn*[™]/ENGINE, *RecurDyn*[™]/GEAR are trademarks of FunctionBay, Inc.

Third-Party Trademarks

Windows and Windows NT are registered trademarks of Microsoft Corporation.

ProENGINEER and ProMECHANICA are registered trademarks of PTC Corp. Unigraphics and I-DEAS are registered trademark of UGS Corp. SolidWorks is a registered trademark of SolidWorks Corp. AutoCAD is a registered trademark of Autodesk, Inc.

CADAM and CATIA are registered trademark of Dassault Systems. FLEX/m is a registered trademark of GLOBEtrotter Software, Inc. All other brand or product names are trademarks or registered trademarks of their respective holders.

Edition Note

These documents describe the release information of *RecurDyn*[™] V9R1.

目录

预备工作	5
目标	5
读者	5
预备知识	5
步骤	6
预计完成时间	6
创建初始模型	7
任务目标	7
预计完成时间	7
了解模型	8
启动 RecurDyn 并导入几何	9
增加运动副和力	10
运行仿真	12
查看结果	13
集成 CoLink	14
任务目标	14
预计完成时间	14
创建 PlantInput	15
创建 PlantOutputs	17
创建 CoLink 模型	18
仿真 CoLink 模型	21
增加微分控制	23
任务目标	23
预计完成时间	23
修改 RecurDyn 模型	24
增加微分控制	25
仿真微分控制模型	27
增加积分控制	29
任务目标	29

预计完成时间	29
加入积分控制	30
保存 PID 控制模型	32

Chapter

1

预备工作

目标

本教程使用 **CoLink** 提供的设计、仿真和测试时变系统的交互环境，仿真和控制一个动态系统。在 **CoLink** 中定义一个控制系统，去控制在 **RecurDyn** 中创建的机械系统。

本次仿真的系统是一个简单的倒立摆系统，系统固定在一个基座上，基座可以来回移动。倒立摆系统通过对基座施加一个横向力，倒立摆可以在直立状态平衡。本教程实施不同的控制系统，控制这个横向力。

读者

本教程是为 **RecurDyn** 中级用户准备的，这些用户此前应已学习如何创建几何体、运动副和力元素。所有的新任务都会仔细说明。

预备知识

首先学习三维曲柄滑块教程和螺旋桨发动机教程。假设已经具备基本的物理知识。

步骤

本教程包括以下步骤。每个步骤预计完成的时间如下表所示。

Procedures	Time (minutes)
创建初始模型	10
集成 CoLink	15
增加微分控制	10
增加积分控制	5
Total:	40



预计完成时间

本教程大约需要 40 分钟完成。

Chapter

2

创建初始模型

任务目标

学习如何创建机械系统模型，以便修改后用于带控制系统的模型。

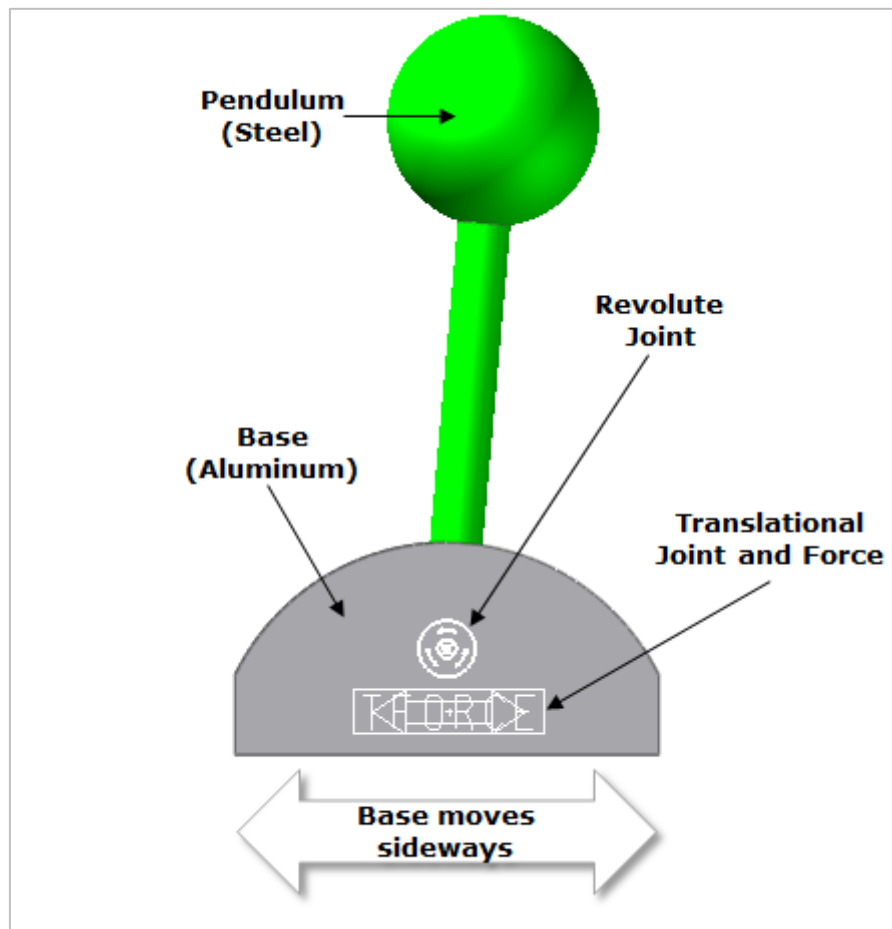


预计完成时间

10 分钟

了解模型

完整模型如下图所示：



关于模型需要注意的是：

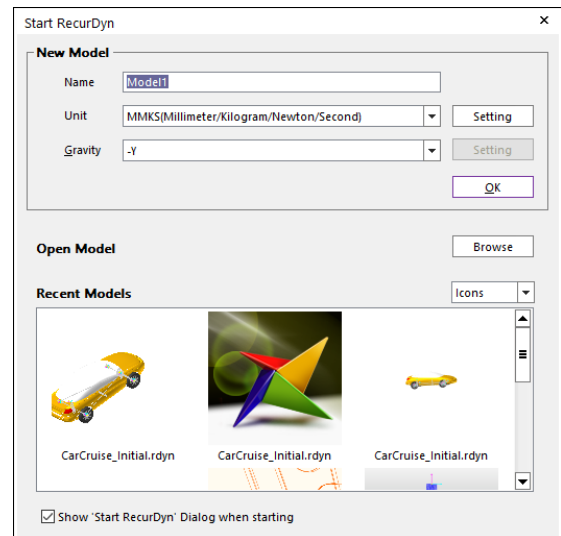
- 摆锤在偏移垂直方向 5 度角位置，用于系统摄动。
- 摆锤材质为铁，基座材质为铝。在 **Parasolid** 文件中定义合适的质量密度。导入几何后，会在 **Body Properties** 窗口的 **Body** 标签下看到不同的密度。
- 模型仿真设置为 5 秒钟。

启动 RecurDyn 并导入几何体

启动 RecurDyn

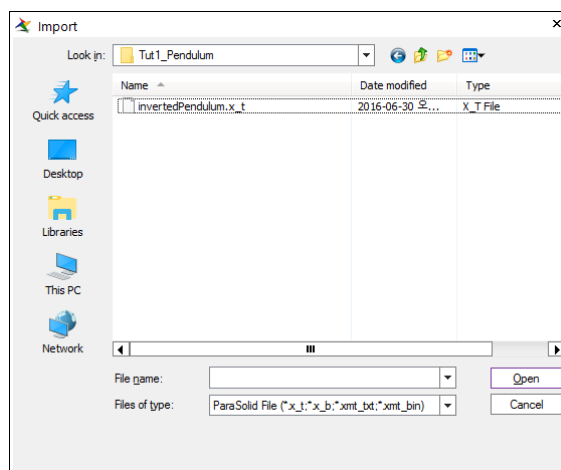


1. 双击桌面上的 **RecurDyn** 图标。
2. 在 **StartRecurDyn** 窗口，设置 **Pendulum** 为模型名称。
3. 点击 **OK**。



导入几何体

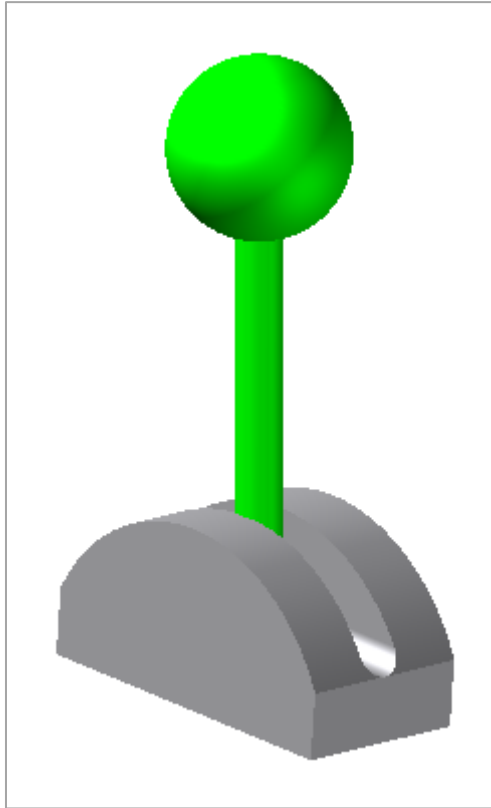
1. 在 **File** 菜单下，选择 **Import**。
2. 在下拉菜单下，选择 **ParaSolid** 文件作为导入的文件类型。
3. 转到 **CoLink** 的教程路径。
4. 选择 **invertedPendulum.x_t** 文件（文件目录：
 <InstallDir>\Help\Tutorial\Colink\Pendulum）
5. 点击 **Open**。





6. 更改渲染模式为 **Shade**。

可以看到机械模型的两部分，倒立的摆锤和基座如下图所示。



7. 选择**基座**，并右键选择 **Properties**。
8. 在 **General** 标签下，命名为 **Base**。
9. 点击 **OK**。
10. 对摆锤重复上述 7-9 步，并命名为 **Pendulum**。

增加运动副和力

增加两个运动副和一个平移力。

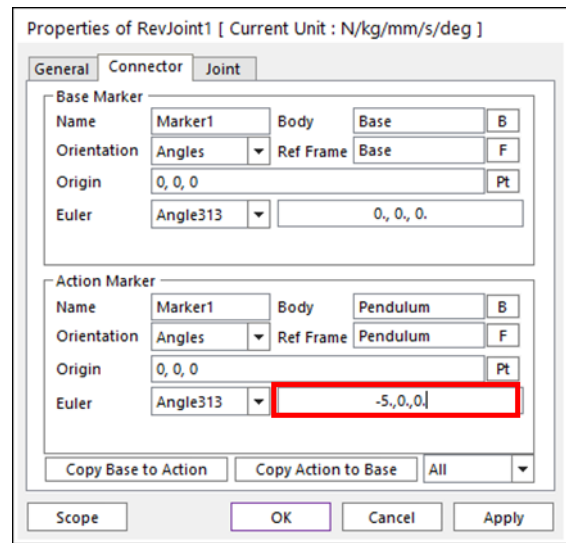
增加旋转副：



1. 在 **Professional** 标签下的 **Joint** 组中，点击 **Revolute**。
2. 设置 **Creation Method** 为 **Body, Body, Point**。
3. 选择 **Base** 作为基座体。
4. 选择 **Pendulum** 作为运动体。

5. 输入 $0, 0, 0$ 作为连接原点。
6. 在 **Database** 窗口，右键点击 **RevJoint1** 并选择 **Properties**。
7. 在 **Connector** 标签下，更改 **Action Marker** 的 Euler 角度为 $-5, 0, 0$ 。

注： 设置初始转动偏移量为 -5 度，这对于控制系统确认初始位置是必需的。

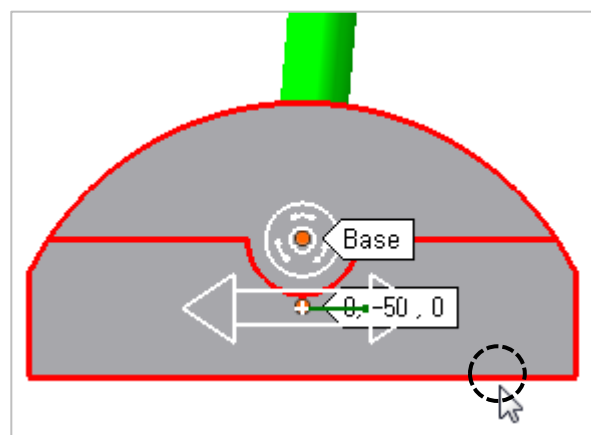


8. 点击 **OK**。

增加平动副：



1. 在 **Professional** 标签下的 **Joint** 组中，点击 **Translate**。
2. 选择 **Creation Method** 为 **Body, Body, Point, Direction**。
3. 选择 **Ground** 作为 **basebody**。
4. 选择 **Base** 作为 **actionbody**。
5. 输入坐标 $0, -50, 0$ 作为运动副原点。
6. 将鼠标移动到 **Base** 的底部边缘上方，**base** 中心的右边。运动副的方向如右图所示。
7. 点击 **Base** 的边。



增加平移力:



1. 在 **Professional** 标签下的 **Force** 组中, 点击 **Translational**。
2. 设置 **Creation Method** 为 **Body, Body, Point, Point**。
3. 选择 **Ground** 作为 **base body**。
4. 选择 **Base** 作为 **action body**。
5. 两次输入 **0, -50, 0**, 将 **base** 和 **action** 点相同设置。

保存 **RecurDyn** 模型

保存模型并命名 **Pendulum_P.rdyn**。

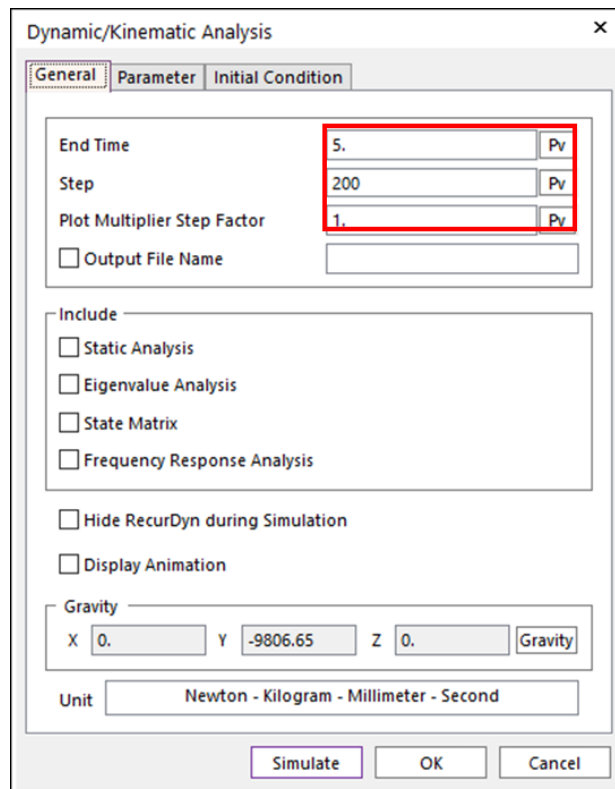
运行仿真

运行仿真并查看不受控制的摆锤系统的运动特性。

运行仿真

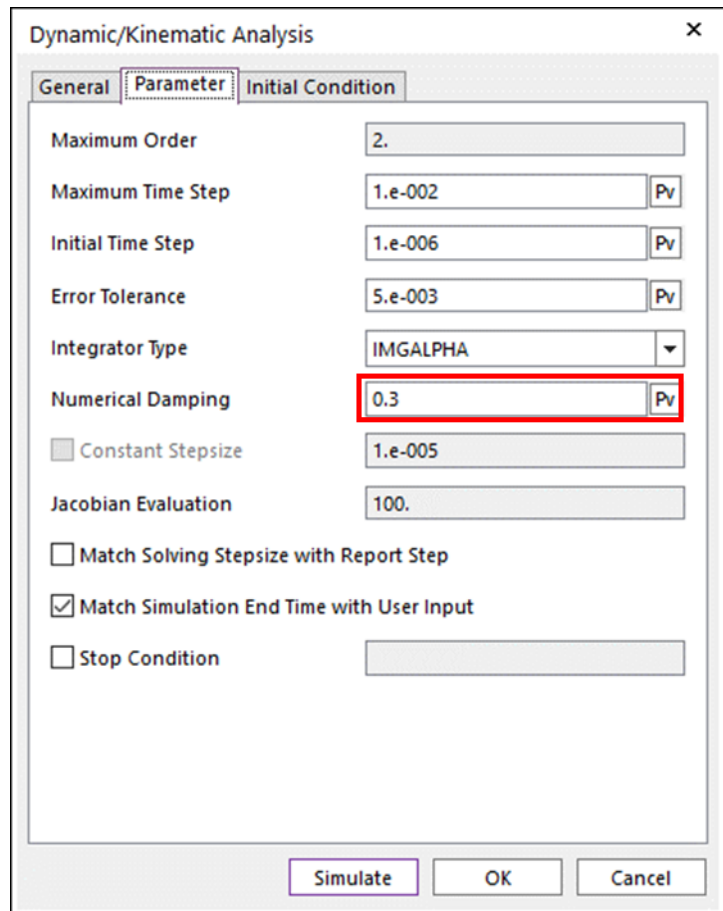


1. 点击 **Analysis** 标签下的 **Simulation Type** 组中的 **Dyn/Kin**。
2. 设置仿真的运行时间 **5** 秒钟, 步数为 **200**, 这个时间足够查看倒立摆的运动。



在 **Parameter** 标签下, 设置 **Numerical Damping** 为 **0.3**。

3. 点击 **Simulate**。

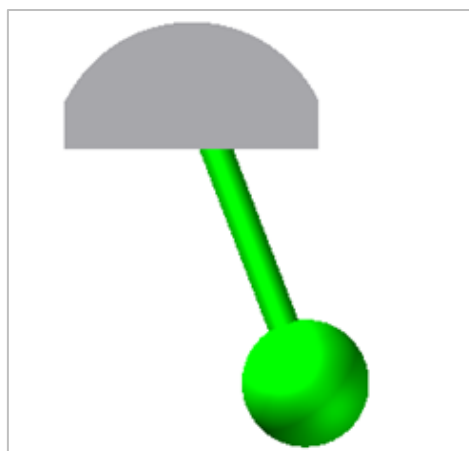


查看结果

查看结果

1. 在窗口顶端的 **Analysis** 标签的 **Animation Control** 组下，点击 **Play** 按钮，显示结果。

可以看到倒立摆自由地向下摆动，**base** 自由地两边移动。



保存 **RecurDyn** 模型

Chapter

3

集成 CoLink

任务目标

本章首先设置参数来集成 **CoLink**，并创建 **CoLink** 模型。**CoLink** 模型是个简比例控制系统。仿真控制系统，并绘制结果。



预计完成时间

15 分钟

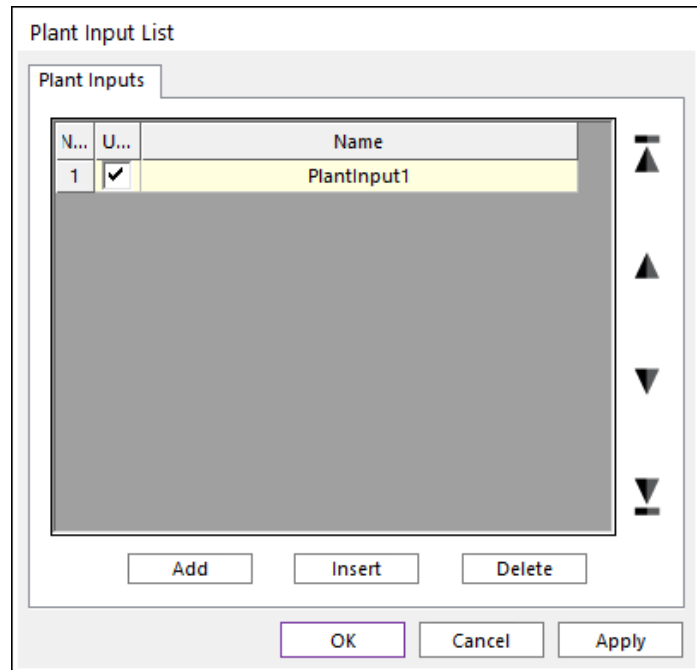
创建 PlantInput

首先创建模型的控制输入接口元素。此元素为占位符，后面将定义并返回其值。

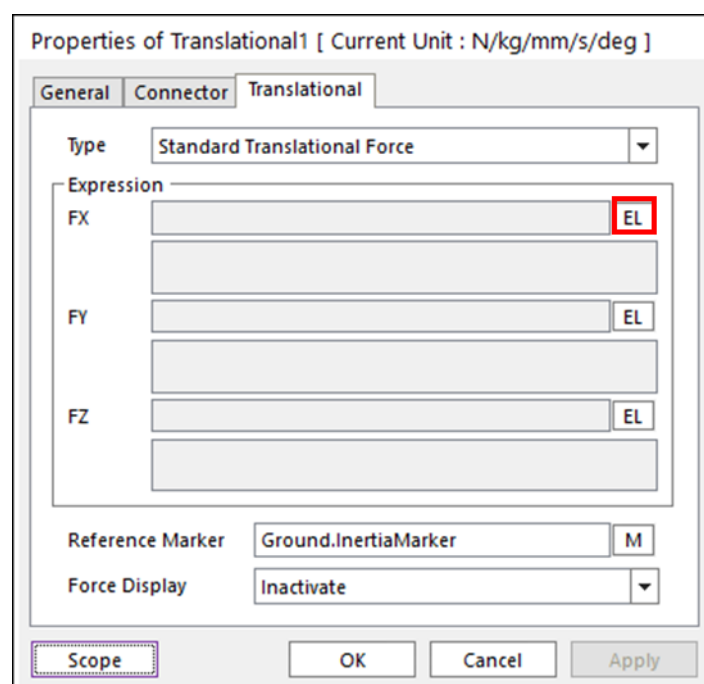
创建 PlantInput:



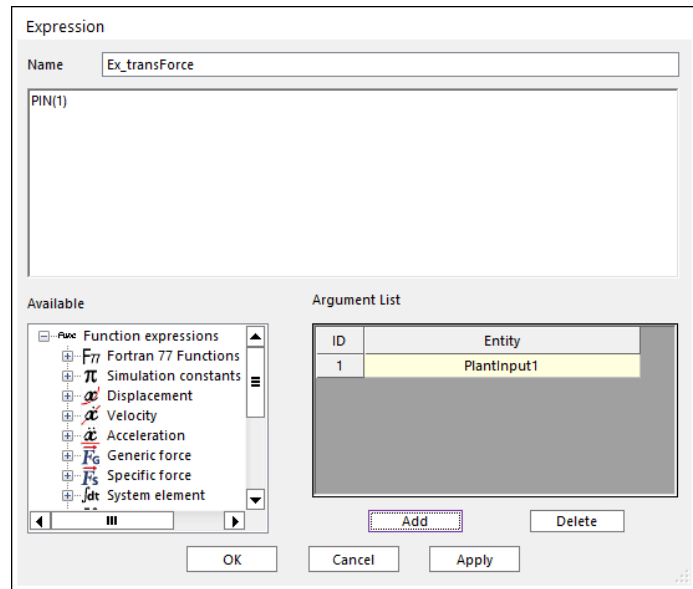
1. 在 CoLink 标签下的 Colink 组中，点击 PlantInput。
2. 点击 Add。
3. 点击 OK。



4. 在 Database 窗口，右键点击 Translational1，并选择 Properties。
5. 点击 FX 右端的 EL 按钮，定义 X 方向的驱动力。



- 在 **Expression** 列表对话框，点击 **Create** 按钮。
- 设置 **Ex_transForce** 作为表达式的名称。
- 设置 **PIN (1)** 作为表达式。
- 在 **ArgumentList** 对话框，点击 **Add**。
- 在 **Database** 窗口，将 **PlantInput1** 拖至 **Argument List** 的第一项，如右图所示。
- 点击 **OK**，完成更改。



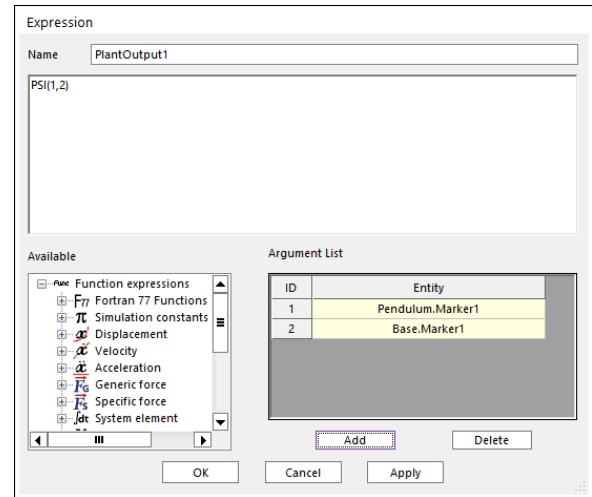
创建 Plant Outputs

创建一个从模型到控制系统的输出项。这个输出设置为倒立摆与垂直方向的夹角。

创建 plant outputs:



1. 点击 **Colink** 标签下的 **Colink** 组中的 **Plant Output**。
2. 在 **Plant Output List** 对话框，点击 **Add**。
3. 设置 **PSI(1, 2)** 作为表达式。这个表达式是指 **Eulerangle** 的 **PSI** 角度，为了定义 **Pendulum** 和 **Base** 之间的旋转角度差。
4. 增加两个元素到 **Argument List**，输入定义 **Revolute1** 的两个坐标。先输入倒立摆的坐标，如右图所示。
5. 点击 **OK**，完成更改。



创建 CoLink 模型

下面，打开 **CoLink**，创建控制系统。这涉及创建框图。框图包括 **RecurDyn** 模型，标准 **P** 控制器增益，以及示波器输出。

创建 **CoLink** 模型：

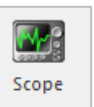


1. 在 **Colink** 标签下的 **Colink** 组中，点击 **Run CoLink**。

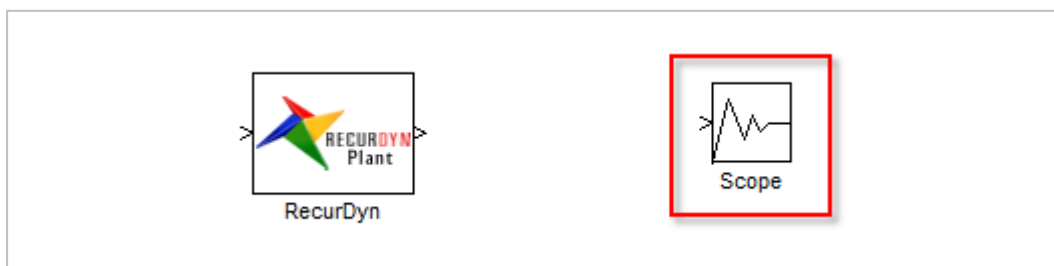
CoLink 打开一个空模型。



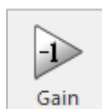
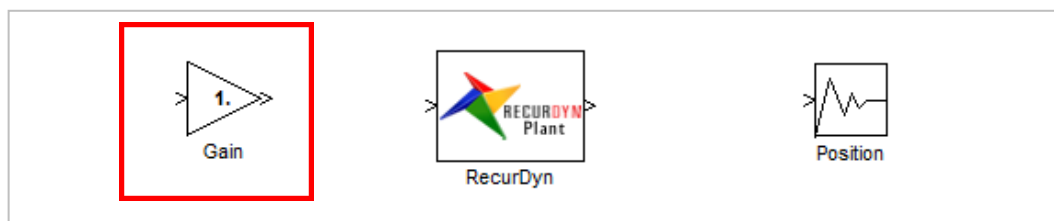
2. 在 **Connector** 标签的 **Link** 组下，点击 **RecurDyn** 块，然后点击工作窗口，把它放在 2/3 的位置处，如下图所示。



3. 在 **General** 标签的 **Output** 组下，点击 **Scope** 块，然后点击工作窗口，把它放置在 **RecurDyn** 块的右边，如下图高亮显示。



4. 双击 **Scope** 块下的文本，重命名为 **Position**。
5. 在 **Math** 标签下的 **Math** 组中，点击 **Gain** 块，然后点击工作窗口，把它放置在 **RecurDyn** 块的左边，如下图所示。

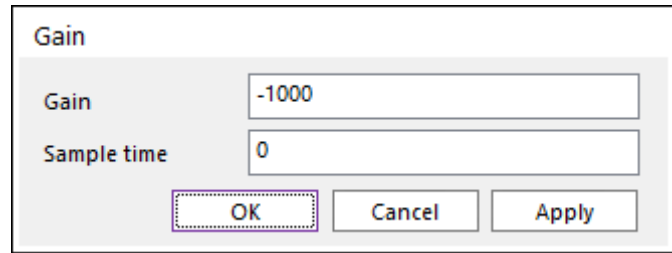


6. 通过点击命名，更改 **Gain** 块的名称为 **Kp**。

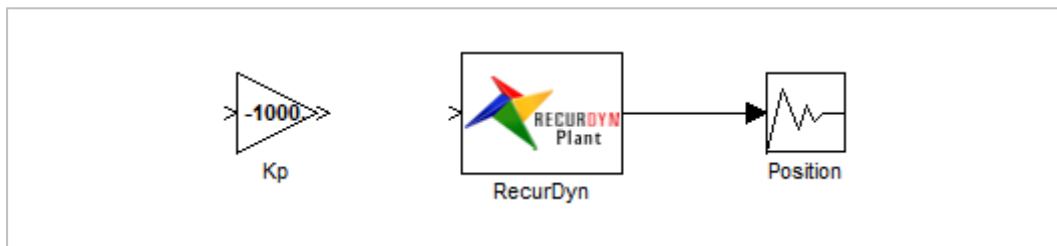
7. 双击 **Gain** 块，编辑 **gain** 值。

8. 更改 **Gain** 为-1000。

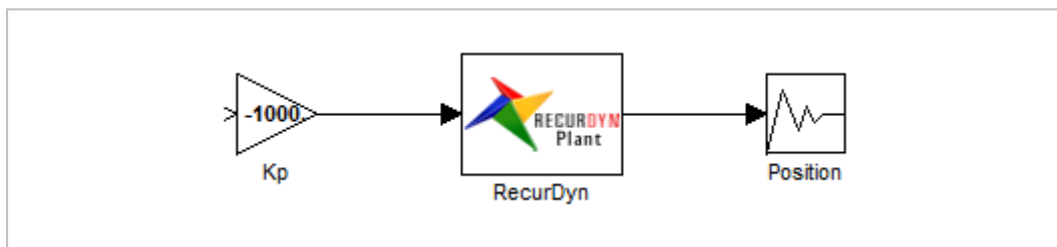
9. 点击 **OK**。



10. 将 **RecurDyn** 块与 **Position** 块连接，如下图所示，点击 **RecurDyn** 块，按住 **Ctrl** 键，并单击 **Position** 块。或者点击 **RecurDyn** 块右侧的箭头，将光标拖至 **Position** 块左侧的箭头，然后释放鼠标键。



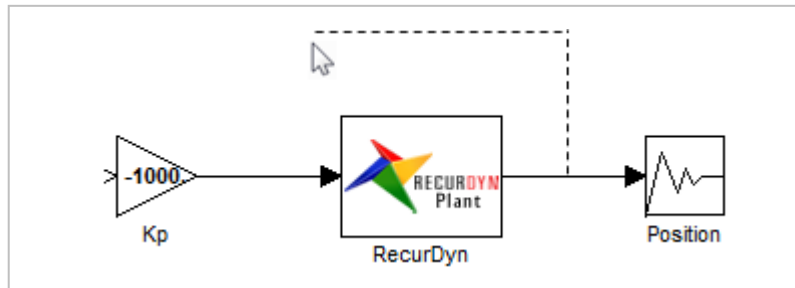
11. 将 **Kp** 块和 **RecurDyn** 块连接，如下图所示。



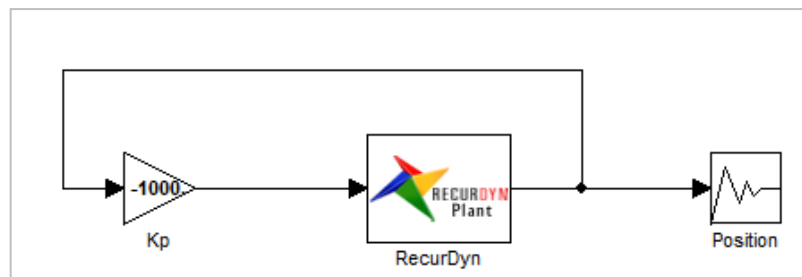
下面增加一个比例控制反馈回路。

增加一个比例控制反馈回路：

1. 右键点击连接 **RecurDyn** 块和 **Position Scope** 之间的直线，拖放到 **Kp** 块的输入。



2. 结果如下图所示。



保存控制系统

1. 在 **CoLink Quick Access** 工具栏，选择 **Save**。
2. 相同路径下的 **CoLink** 文件保存为 **RecurDyn** 模型，命名为 **Pendulum_P. clk**。

仿真 CoLink 模型

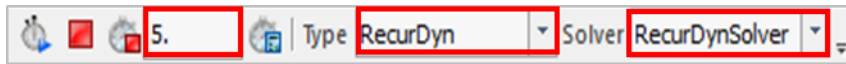
仿真在 CoLink 中创建的比例控制系统模型。

仿真控制系统的模型：

1. CoLink 没有状态窗口。因此，在运行仿真之前，调整 CoLink 和 RecurDyn 窗口的位置，以看清 RecurDyn Output 窗口。
2. 在 CoLinkQuickAccess 工具栏，选择 Show Simulation 工具栏。



3. 在 Simulation 工具栏下，更改仿真的时间为 5s，更改 Type 为 RecurDyn，更改 Solver 为 RecurDyn Solver。



4. 点击 Start 按钮，可以在 RecurDyn 的输出窗口看到运行状态。

小贴士： 如果联合仿真不运行怎么办？

如果联合仿真不运行，会弹出一个 Server Busy 对话框。如果对话框出现：

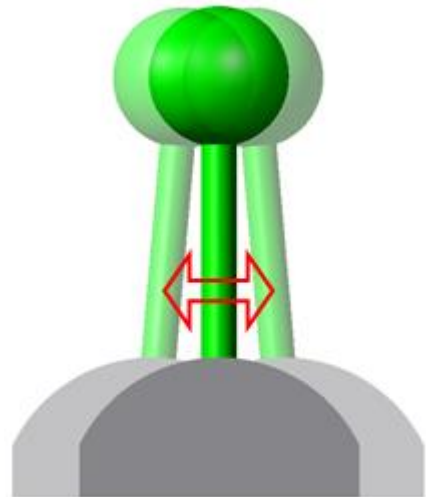
- 点击 Switchto，查看 RecurDyn 输出窗口的信息。

如果不能发现 CoLink 模型的错误信息，查看 RecurDyn 和 CoLink 模型的位置。确保两个模型在同一个路径。如果不在同一路径，调整后，并重启 RecurDyn 和 CoLink。仿真应该可以正确运行。

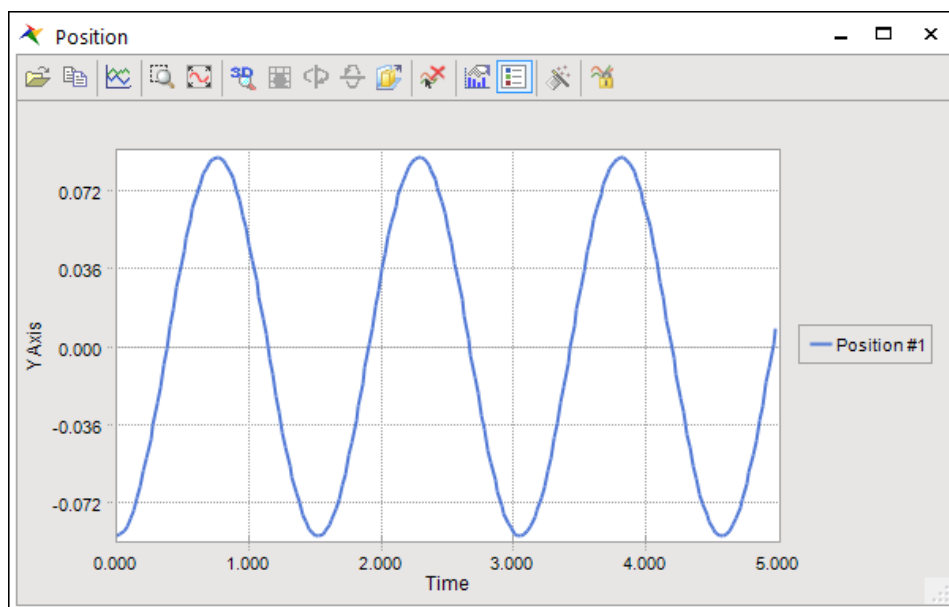
查看结果

1. 返回到 **RecurDyn** 窗口，按 **Play** 按钮。

可以看到倒立摆在控制系统作用下两边晃动，但保持竖直。



2. 返回到 **CoLink** 窗口，双击 **Position** 示波器。可以看到倒立摆还有角度位置的绘图，如图所示。



显然，倒立摆是保持直立的，绘图显示了摆动的幅度随时间缓慢衰减的变化规律。为了弥补这一点，在下一章增加微分控制。

Chapter

4

增加微分控制

任务目标

本章修改 **CoLink** 模型，包括微分控制，创建比例-微分 (**PD**) 控制系统。可以仿真这个系统并观察结果。



预计完成时间

10 分钟

修改 RecurDyn 模型

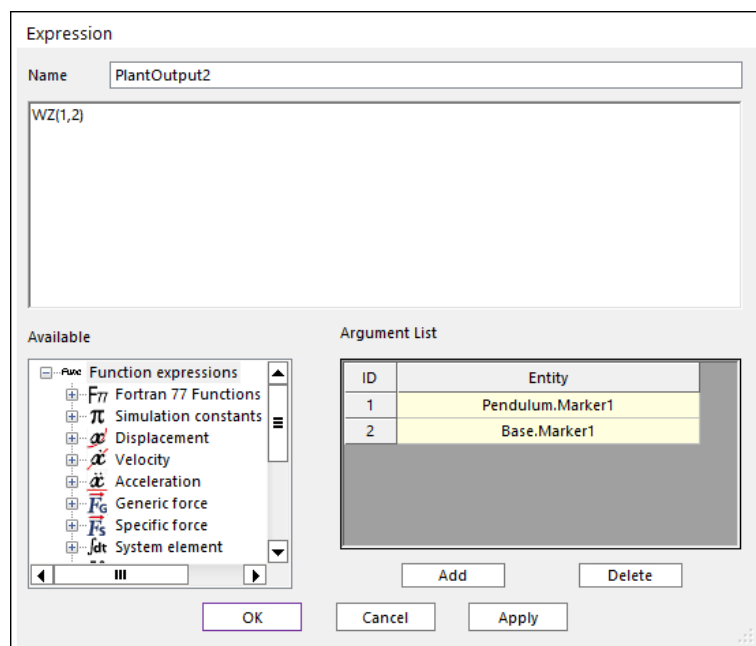
为了增加微分控制，先保存 **RecurDyn** 模型为不同的文件名，然后修改并输出控制系统的摆动位置和速度。

在不同名称下，保存模型：

1. 返回到 **RecurDyn** 窗口。
2. 在 **File** 菜单下，选择 **Save As**。
3. 保存文件为 **Pendulum_PD.rdyn**。

添加一个倒立摆的旋转速度 **plant output**。

1. 在 **Database** 窗口，双击 **PlantOutput1**，弹出 **Plant Output List** 对话框。
2. 点击 **Add** 按钮。
3. 输入 **WZ(1, 2)** 作为表达式。
4. 和之前一样，增加两个 **markers** 到 **Argument List**，如右图所示。
5. 点击 **OK**，完成更改。



增加微分控制

先保存 CoLink 模型为不同文件名，然后修改此模型来增加微分控制的反馈回路。

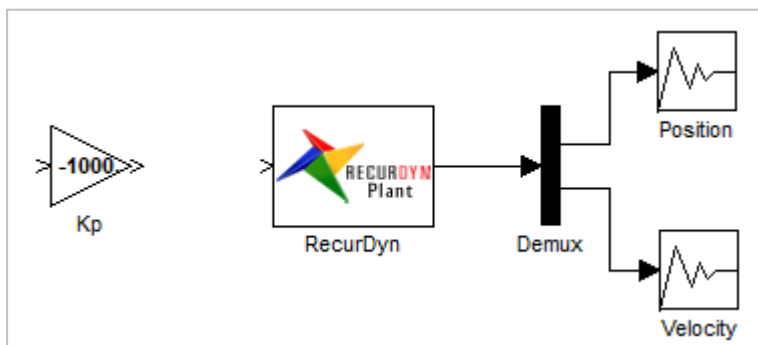
将控制系统保存在不同名称下：

1. 返回到 CoLink 窗口。
2. 在 File 菜单下，选择 SaveAs。
3. 保存文件为 Pendulum_PD. clk。

增加微分控制：

1. 通过选择 block 之间的连接，并按 Delete 键，删除所有连接线。

下图指导进入下一步：

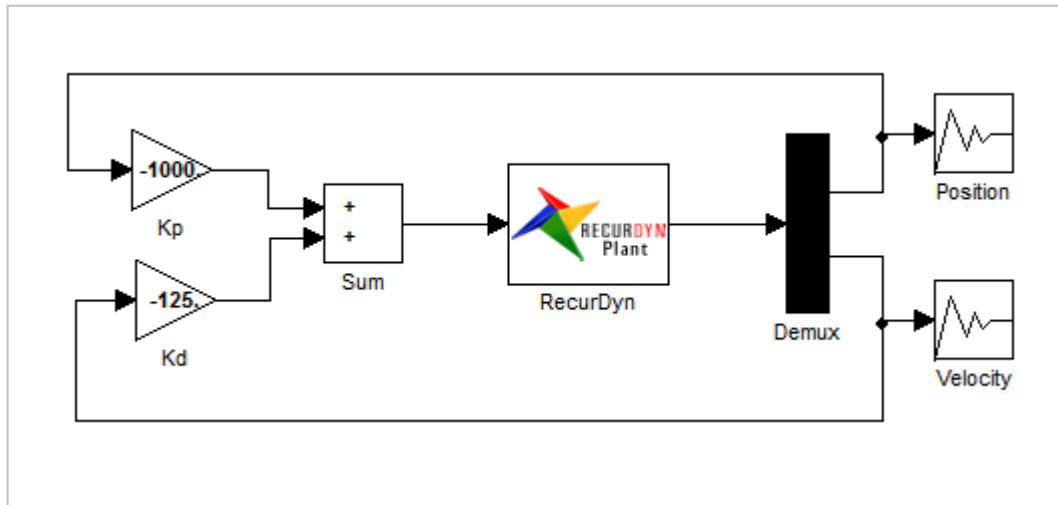


2. 在 Connector 标签下的 Connector 组中，点击 Demux 块，然后点击工作窗口。将其放置在 RecurDyn 块和 Position 示波器之间。
3. 在 Position 示波器下，增加另一个 Scope (Output)。
 - 重命名 Scope 为 Velocity。
4. 如上图所示，在 RecurDyn plant, Demux 模块, Position 和 Velocity 示波器之间建立连接。
5. 在 Kp 增益块下，增加另一个 gain block，如下所示。
 - 重命名 gain block 为 Kd。
 - 更改 gain value 为-125。



6. 在 Math 标签的 Math 组，点击 Sum block，然后点击工作窗口，将其放置在 Kp gain block 和 RecurDyn plant 之间。

7. 如下图所示, 在 gain blocks, sum block 和 RecurDyn plant 之间建立连接。



8. 保存控制系统。

仿真微分控制模型

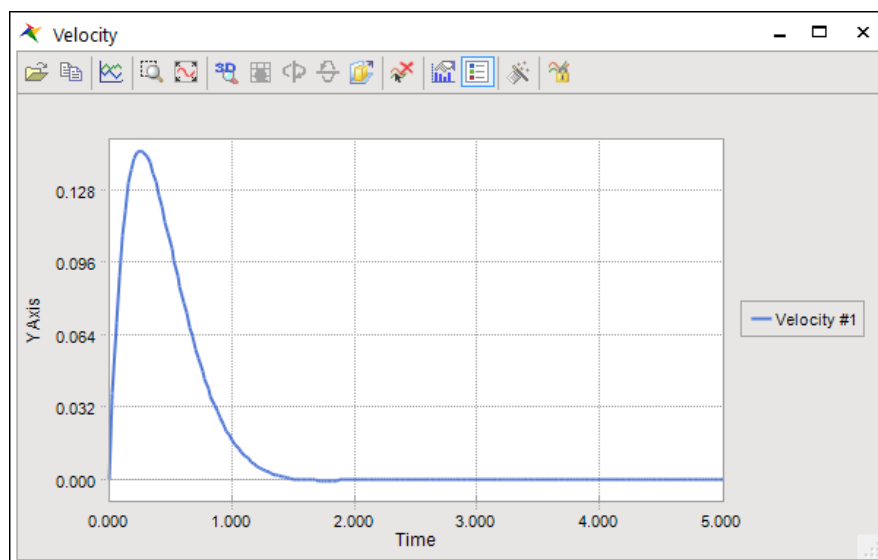
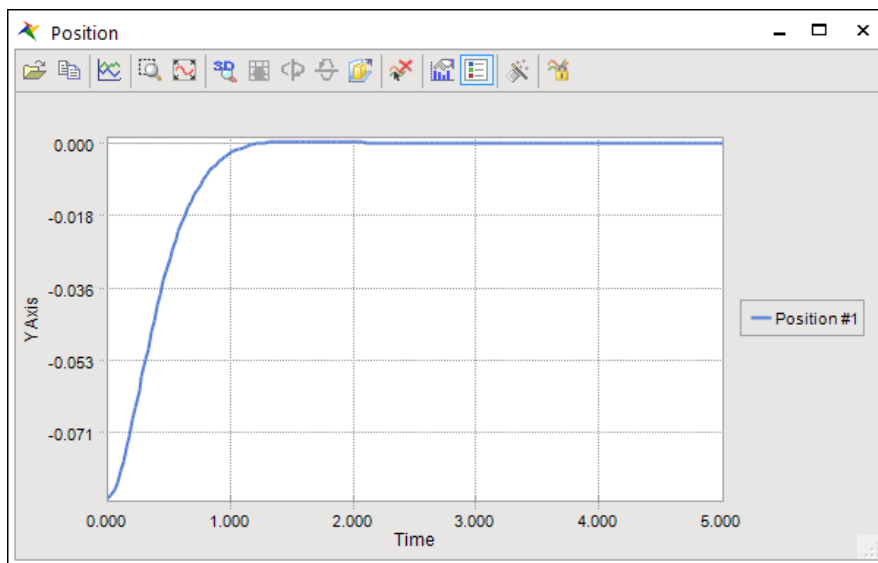
仿真模型。

小贴士： 如果运行箭头是灰色？

极少情况下，**CoLinkRun** 箭头是灰色的。如果发生这种情况，简化并保存 **CoLink** 模型，关闭 **CoLink**，重启 **CoLink** 并加载模型。

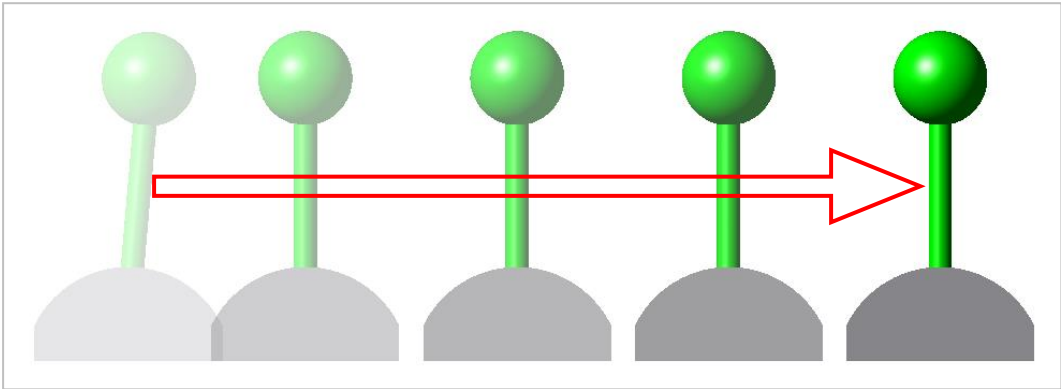
查看结果

1. 双击位置和速度示波器。可以看到以下结果：



结果显然更好，倒立摆的角位置和速度都收敛到 0，而不是振荡。

2. 回到 RecurDyn 窗口:
3. 播放动画。



动画显示：倒立摆确实保持直立，整个系统偏移到右边。为减少偏移，在下章增加积分控制。

Chapter

5

增加积分控制

任务目标

本章修改 **CoLink** 模型，包括积分控制，因此创建一个比例-积分-微分 (**PID**) 控制系统，仿真这个系统，并观察结果。



预计完成时间

5 分钟

加入积分控制

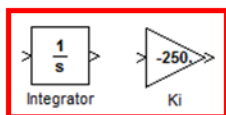
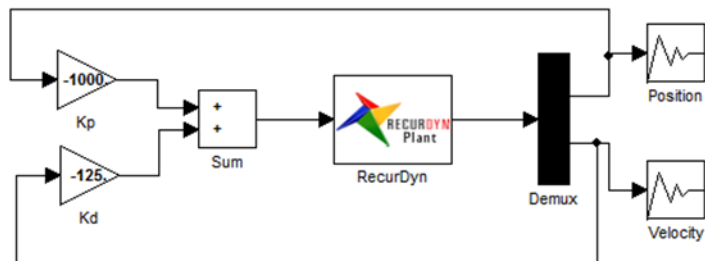
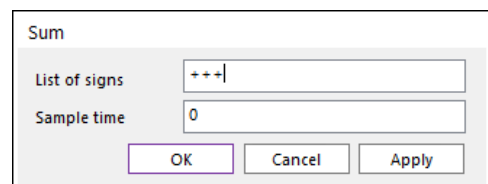
先保存 CoLink 模型为不同文件名，然后再修改，包括处理积分控制的另一个反馈回路。

在不同名称下，保存控制系统：

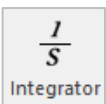
1. 在 **CoLink File** 菜单下，选择 **Save As**。
2. 保存文件为 **Pendulum_PID. clk**。

增加积分控制：

1. 双击 **Sum** 块。
2. 在 **List of Signs** 输入+++。将 **Sum** 块的输入增加为 3 个。
3. 点击 **OK**。



根据上图指导，进入下一步：



4. 在 **Continuous and Discrete** 标签的 **Continuous** 组下，点击 **Integrator** 块，然后点击工作窗口。将它放在下面，**Kd gain block** 的左边。
5. 将另一个 **gain block** 放在模型中，**Kd gain block** 的下面。
 - 命名 **gain block** 为 **Ki**。
 - 更改 **gain value** 为 **-250**。

6. 完成下面显示的 3 个连接，变为红色。

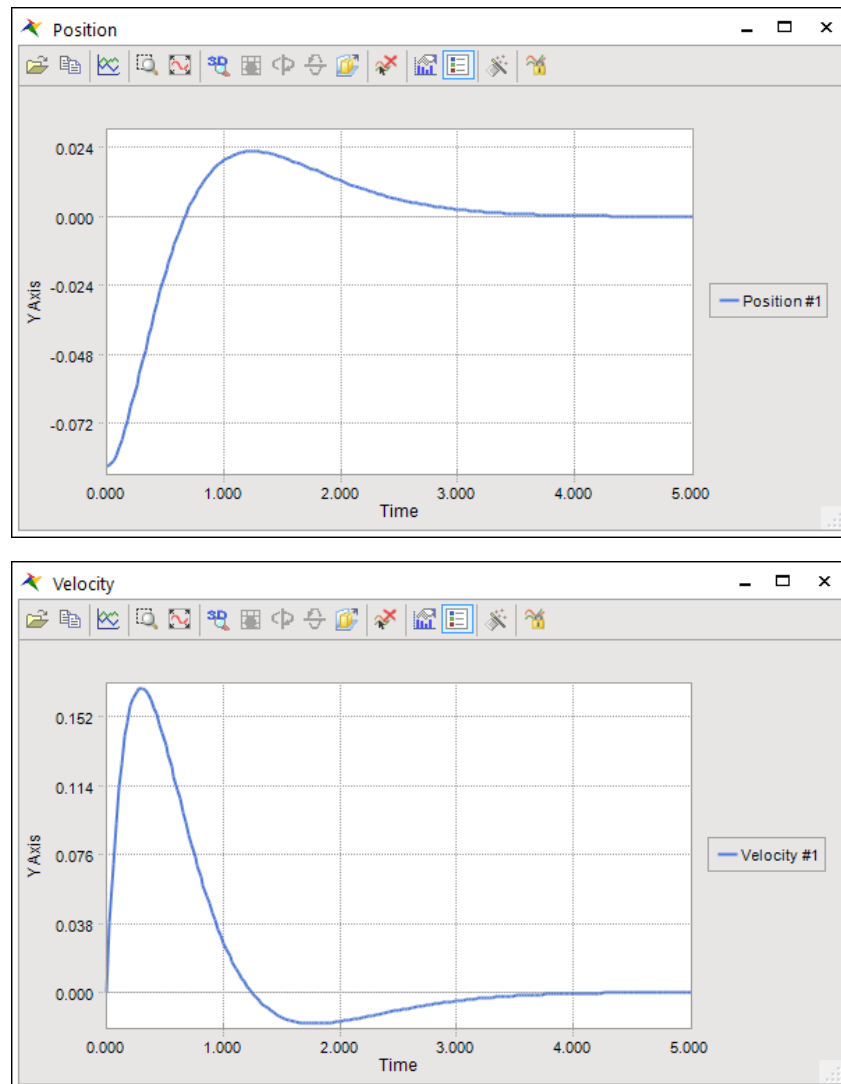
7. 保存控制系统。

仿真 PID 控制模型

仿真模型

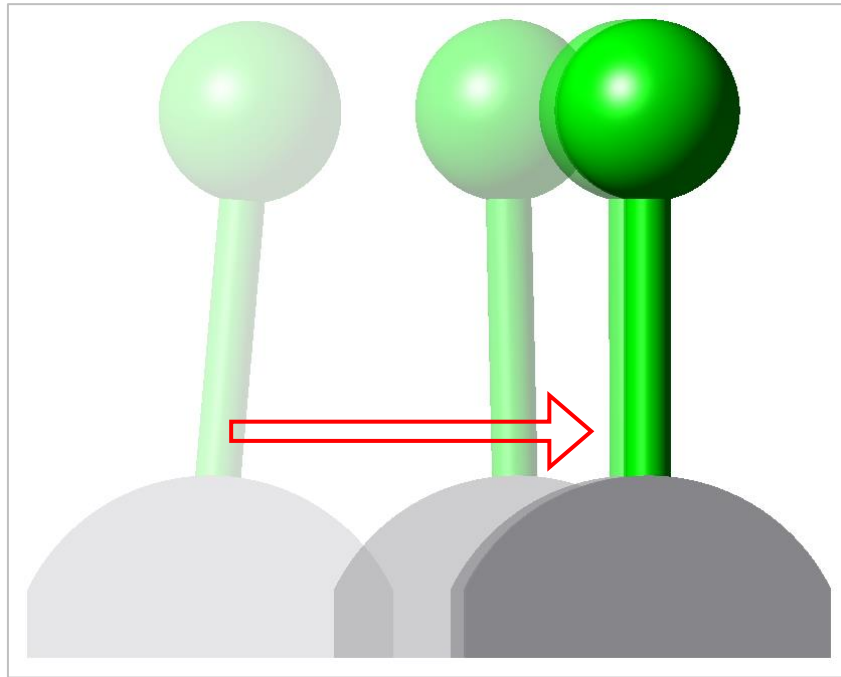
查看结果

1. 双击位置和速度范围，结果如下所示：



在位置收敛到零的过程中，出现超调。

2. 返回到 **RecurDyn** 窗口。
3. 播放动画。



动画显示：倒立摆保持直立，基座接近停止，而非和之前一样晃动。因此，更多类型的反馈回路的添加，使控制系统变得更稳定。

感谢参与本教程学习!