RECURDYN

Car Tutorial (CoLink)

FunctionBay

Edition Note

This document describes the release information of **RecurDyn V9R4**.

목차

Chapter 1

개요

목적

이 튜토리얼에서는 설계, 시뮬레이션, 시간 변이 시스템의 테스트를 위한 상호 환경인 CoLink 를 사용하여 동역학 시스템의 시뮬레이션 수행 및 제어에 대해 다루게 될 것입니다. 또한, RecurDyn 에서 설계한 기계 시스템을 제어하기 위해서 제어 시스템을 정의하게 될 것입니다.

시뮬레이션을 실행하게 되는 시스템은 단순히 1 차원적인 문제이며, 그 시나리오에는 2 개의 자동차를 포함시켜서, 앞에는 느린 속도의 자동차, 그 뒤에는 더 빠른 속도의 자동차가 뒤따라오도록 할 것입니다. 이 상황은 크루즈 컨트롤 또는 자동차를 운전하는 중에 갑자기 장애물이 나타나 예상치 못한 제어를 가정한 것이며, 컨트롤러는 자동차 사이의 거리를 설정하기 위해 주행 중인 바퀴에 적용되는 토크의 양을 원하는 값으로 자동 조절합니다.

그래서, 간단한 PID 컨트롤러는 이러한 목적으로 사용될 것이며, 이 튜토리얼의 마지막 부분에서는 조정 가능한 크루즈 제어 시스템을 모델링 하기 위해 복잡한 컨트롤러에 대해 배우게 될 것입니다.

독자

이 튜토리얼은 RecuDyn 을 이전에 배워 본 경험이 있고, 지오메트리(Geometry), 조인트(Joint), 포스(Force)를 생성할 수 있는 중급 수준의 사용자를 대상으로 하며, 모든 작업은 자세히 설명되어 있습니다.

필요 요건

3D Crank-Slider 와 Engine with Propeller 튜토리얼을 익혔거나 이에 상응하는 작업을 해 본 것을 전제로 하며, 물리학에 대한 기본 지식이 있어야 합니다.

과정

이 튜토리얼은 다음의 과정으로 구성되며, 각 과정을 완성하기까지 걸리는 시간은 다음의 표와 같습니다.

과정	시간(분)
초기 모델 열기	5
조인트와 커플러의 생성	15
모델의 수정	15
CoLink 와 통합	15
CoLink 모델 시스템의 확장	15
총합	65

예상 소요 시간

약 65 분

Chapter 2

초기 모델 열기

목적

제어 시스템을 사용하기 위한 초기 모델을 열어서 모델을 수정하기 위한 준비를 할 것입니다.

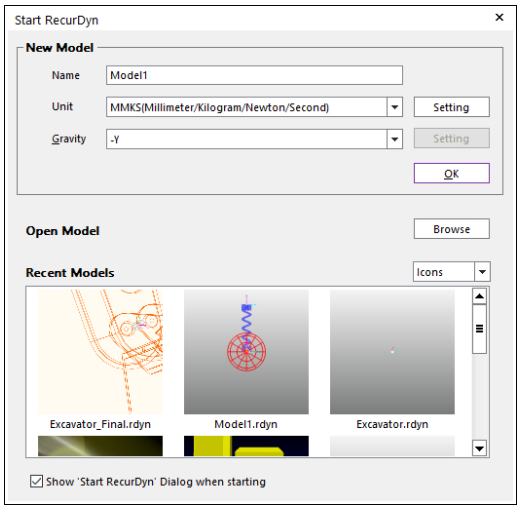
예상 소요 시간

5 분

RecurDyn 의 실행

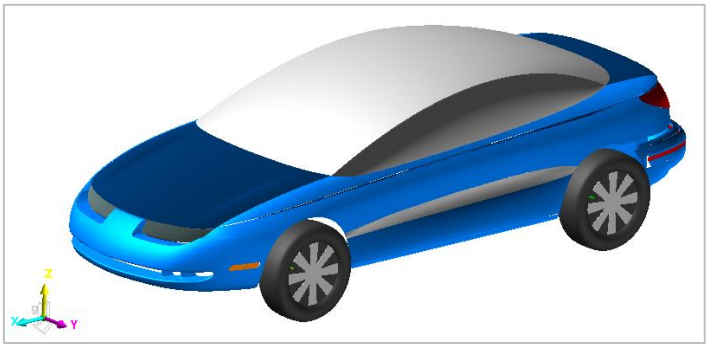
RecurDyn 을 실행하여 초기 모델 열기**:**

1. 바탕화면에서 **RecurDyn** 아이콘을 더블 클릭합니다.
2. 새로운 모델을 생성하지 않고 기존에 있는 모델을 불러와서 활용할 것이기에 Start RecurDyn 윈도우를 닫아 줍니다.



3. **File** 메뉴에서, **Open** 을 클릭합니다.
4. CoLink 튜토리얼 관련 디렉토리 (<Install Dir> \Help \Tutorial \Colink \Car)에서, **CarCruise_Initial.rdyn** 파일을 선택합니다.
5. **Open** 을 클릭합니다.

   다음과 같이 모델이 나타납니다.



모델 저장하기**:**

1. **File** 메뉴에서, **Save As** 를 클릭합니다.
2. Tutorial 디렉토리에서는 시뮬레이션을 할 수 없기 때문에 다른 디렉토리에 다시 저장합니다.

Chapter

3

조인트와 커플러의 생성

목적

이 장에서는 자동차의 주행 속도를 설정하기 위해 자동차 바퀴에 조인트와 커플러를 생성할 것입니다. 또한, 자동차의 초기 속도를 정의하고, 자동차의 구름 저항을 보여주는 Translational 조인트의 마찰 저항도 정의할 것입니다.

예상 소요 시간

15 분

Revolute 조인트의 생성

자동차의 바디에 바퀴를 달기 위해서 Revolute 조인트를 생성해보겠습니다.

조인트 생성하기**:**

1. 작업 평면의 방향을 **XZ** 으로 변경합니다. 단축키는 (Shift + A) 입니다.
   방향을 XZ 평면으로 변경하면 Revolute 조인트를 정확한 위치에 정의할 수 있습니다.

Tip: 자동차의 중심을 선택하려면 키보드에서 C 를 누르면 편리하게 선택할 수 있습니다.

2. 동일한 작업을 계속 할 경우, **Auto Operation** 모드로 전환합니다. 이 모드는 키보드에서 A 를 누르면 편리하게 전환할 수 있습니다.

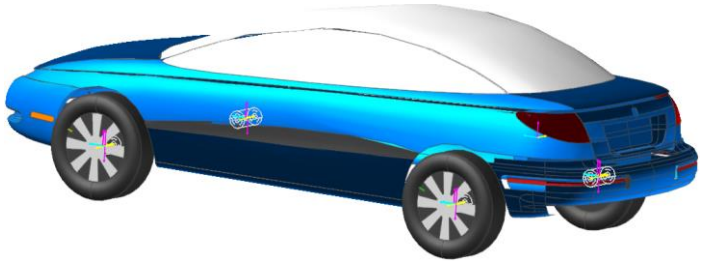
Tip: **Auto Operation** 모드는 동일한 작업을 반복적으로 하게 될 때 편리하게 작업할 수 있도록 도와주는 기능으로, 이 기능을 사용하면 리본메뉴에서 여러 번 아이콘을 선택하거나 메뉴를 선택해야 하는 번거로움을 없애줍니다. Auto Operation 으로 모드를 전환하고 리본메뉴에서 하고자 하는 작업을 선택하면 RecurDyn 은 Auto Operation 모드를 종료할 때까지 반복하여 작업할 수 있게 해줍니다.

3. **Professional** 탭의 **Joint** 그룹에서 **Revolute** 를 클릭합니다.
4. 모델링 생성 방법을 **Body, Body, Point, Direction** 설정합니다.
5. 4 개로 구성된 각각의 바퀴에 대해서 차체(Car_Body) 와 바퀴 사이 마다 바퀴의 무게 중심이 되는 마커에 회전 조인트를 생성합니다.
   - **Car_Body** 를 클릭합니다.
   - 바퀴를 클릭합니다.
   - 바퀴의 무게 중심이 되는 마커를 클릭합니다.

   4 개의 바퀴마다 위 과정을 반복합니다.
6. 4 개의 조인트를 모두 생성했다면, 키보드에서 A 를 다시 눌러서 **Auto Operation** 모드를 종료하고 또 다른 회전 조인트의 생성을 취소하기 위해서 키보드에서 ESC 를 누릅니다.
7. 새로 생성한 4 개의 조인트에 **Rev_Front_Right, Rev_Front_Left, Rev_Rear_Right, Rev_Rear_Left** 로 이름을 다시 정의해줍니다.

Tip: Database 윈도우의 **Revolute Joint** 위에서 오른쪽 마우스 버튼을 클릭한 후 **Property** 를 선택합니다. 그리고 이름을 변경합니다.

이제 다음과 같이 모델이 보일 것입니다.

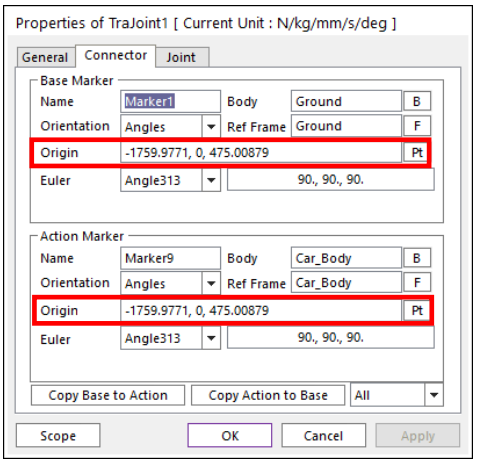


Translational 조인트의 생성

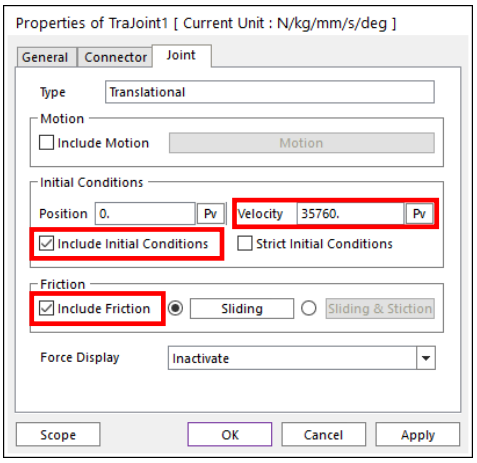
자동차의 모션을 정의하기 위한 Translational 조인트를 생성해보겠습니다.

조인트 생성하기**:**

1. **Professional** 탭의 **Joint** 그룹에서 **Translate** 를 클릭합니다.
2. **Body, Body, Point, Direction** 생성 방법을 설정한 후 **Ground, Car_Body** 를 클릭하고 다음의 좌표를 입력합니다.:
   - **-1759.9771, 0, 475.00879**
3. **Ground InertiaMarker** 의 **+X** 축을 클릭합니다.

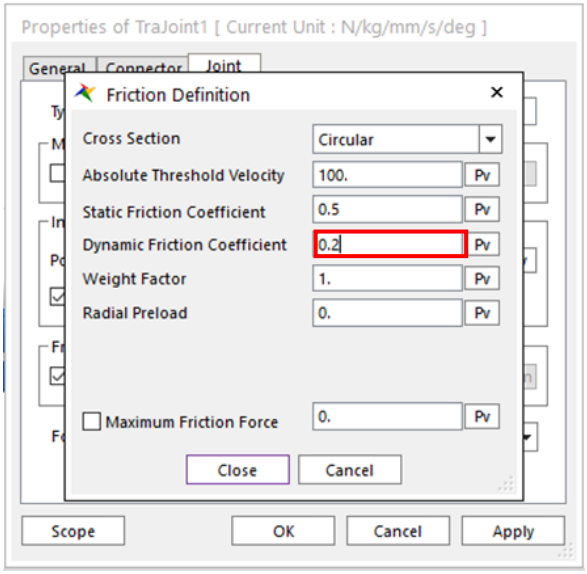


4. **Translational** 조인트의 속성을 변경합니다. (조인트 위에서 오른쪽 마우스 버튼을 클릭한 후 **Property** 를 클릭합니다.)
   - Properties 다이얼로그 윈도우의 **General** 페이지에서 병진 조인트의 이름을 **Tra_Car_Ground** 으로 다시 지정합니다.
   - **Joint** 페이지에서 초기 속도를 설정합니다.
     - **Include Initial Conditions** 체크 박스를 클릭합니다.
     - 속도를 **35760 mm/s** (35.76 m/s or 80 mph)로 설정합니다.



이제, 일반적으로 자동차의 속도를 줄여주는 저항력과 마찰력에 대한 시뮬레이션을 실행하기 위해 모델에 마찰력을 설정해보겠습니다.

5. **Include Friction** 체크 박스를 클릭합니다.
6. **Sliding** 을 클릭한 후 **Dynamic Friction Coefficient** 를 다음과 같이 **0.2** 로 설정합니다.



7. **Close** 를 클릭한 후 변경된 것이 적용되도록 **OK** 버튼을 클릭하고 해당 윈도우를 닫습니다.

이제, 자동차의 바퀴와 주행 모션 사이에 Coupler 를 생성할 준비가 되었습니다.

Coupler 의 생성

자동차의 주행 속도에 영향을 주는 타이어의 각속도를 설정하기 위해서 커플러(Coupler)를 생성해보겠습니다.

커플러 생성하기**:**

1. **Professional** 탭의 **Joint** 그룹에서 **Coupler** 를 클릭합니다.
2. **Joint, Joint, Joint** 생성 방법을 이용하여 병진 조인트인 **Tra_Car_Ground** 를 클릭하고 **Rev_Front_Right** 과 **Rev_Front_Left** 를 클릭합니다.
3. 새롭게 생성된 커플러에 대한 **Properties** 다이얼로그 엽니다.

Tip: 커플러에 대한 척도 인자의 정의

커플러는 다음과 같은 방정식으로 정의됩니다. 이 방정식에서 d' 는 척도 인자이며, u 는 해당 조인트의 속도입니다.

- $d_1 \times u_1 + d_2 \times u_2 + d_3 \times u_3 = 0$

자동차의 주행 속도는 기본적인 관계식 $v = w \times r$ 에 의해서 타이어의 회전과 관련이 있으며, 이 식에서 v 는 차량의 속도이며, w 는 바퀴의 각속도, r 는 바퀴의 반지름입니다. 물론 이것은 바퀴가 같은 속도로 회전하여, 자동차가 오직 직선으로 앞을 향해 주행할 경우를 가정한 것입니다.
다음 단계에서는 커플러를 추가로 더 생성하여 이 부분을 더 강화해보겠습니다. 제공된 지오메트리에 대해 바퀴의 반지름은 336mm 로 하면 위의 방정식은 다음과 같이 됩니다.

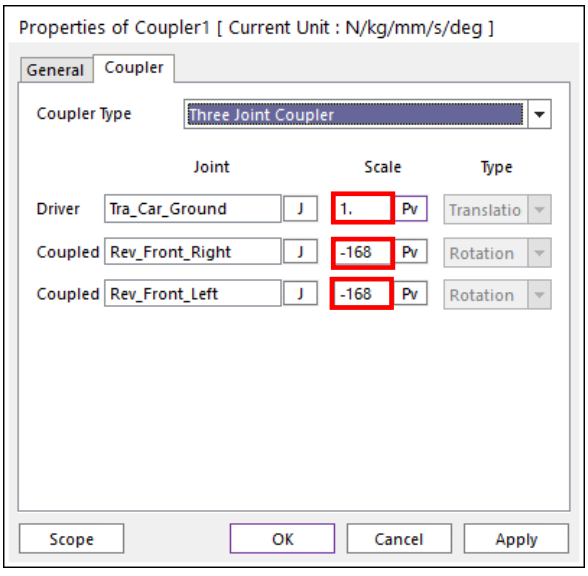
$$d_1 \cdot r \times w + d_2 \times w + d_3 \times w = 0$$
$$d_1 \cdot r + d_2 + d_3 = 0$$

모두 *d2=d3* 라는 동일한 반지름을 갖는 바퀴이기 때문에 척도 인자는 상대적이며 그 중에 하나는 선택적으로 필요함으로 *d1=1* 이라고 가정합니다. 이것은 다음과 같은 식을 이끌어 냅니다.

$$r + d_3 + d_3 = 0$$
$$d_3 = -r/2$$
$$d_3 = -336/2$$
$$d_3 = -168$$
$$d_2 = -168$$

즉, Translational 조인트에 대한 척도 인자는 두 Revolute 조인트 모두에 대해 1 과 -168 로 볼 수 있습니다.

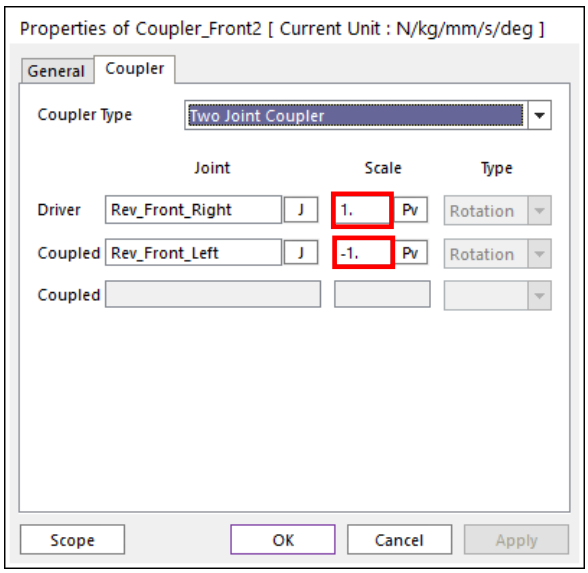
4. Coupler Properties 다이얼로그 윈도우의 **Coupler** 페이지에서 Scale 을 오른쪽 그림과 같이 **1, -168, -168** 로 입력합니다.
5. **General** 페이지에서 **Coupler_Front** 로 이름을 지정하고 **OK** 를 클릭합니다.
6. 뒷바퀴에 대해서 1 번부터 5 번까지의 과정을 반복한 후, 커플러의 이름을 **Coupler_Rear** 으로 지정하고, **Scale** 에 동일한 값을 설정합니다.



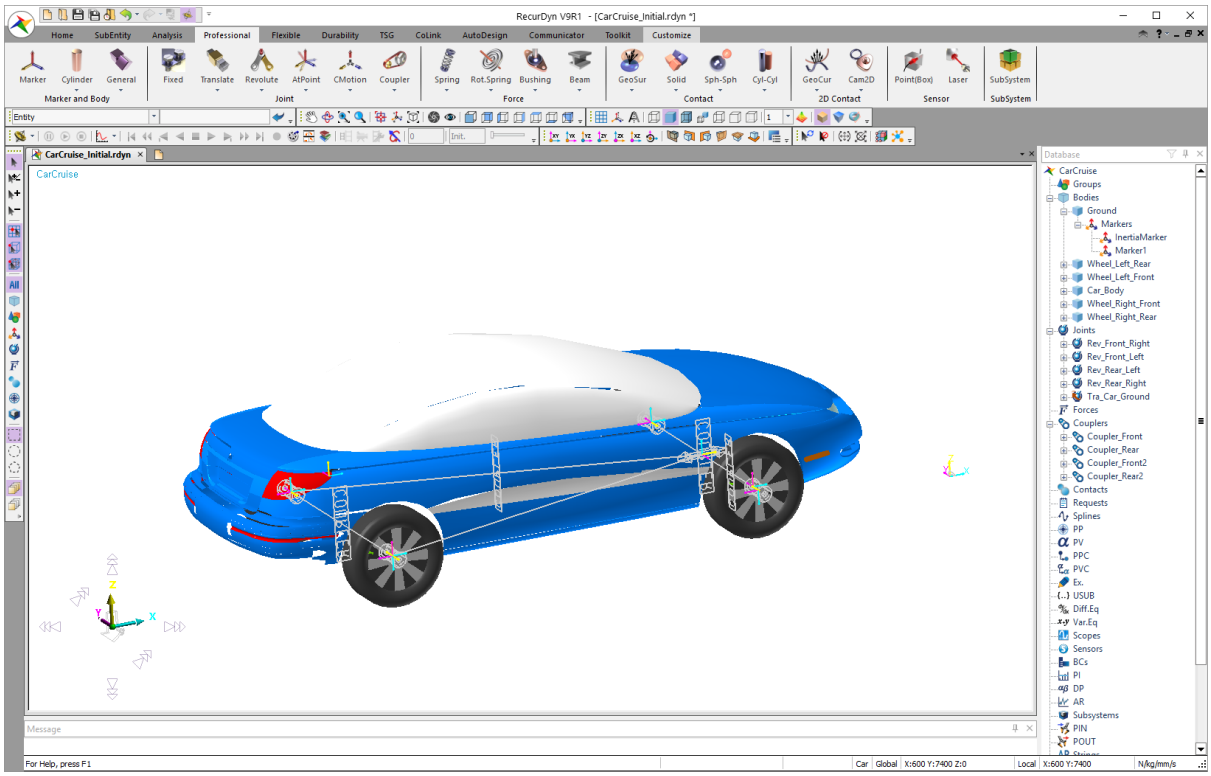
이제 동일한 속도로 회전하는 두 개의 앞 바퀴 사이에 커플러를 생성해보겠습니다.

커플러 생성하기**:**

1. **Professional** 탭의 **Joint** 그룹에서 **Coupler** 를 클릭합니다.
2. Joint, Joint 생성 방법을 이용하여 **Rev_Front_Right** 과 **Rev_Front_Left** 를 클릭합니다.
3. 새롭게 생성된 커플러에 대한 속성을 다음과 같이 수정합니다.
   - 이름을 **Coupler_Front2** 로 지정합니다.
   - 두 개의 회전 속도가 동일해야 하므로 **Scale** 을 **1** 과 **-1** 로 설정합니다.
   - **OK** 를 클릭합니다.
4. 나머지에 대해서도 1 번에서 3 번 과정을 반복합니다. **Coupler_Rear2** 로 이름을 지정하고, **1** 과 **-1** 로 scale 값을 동일하게 설정합니다.



모든 과정이 완료되면 다음과 같이 모델이 보여집니다.

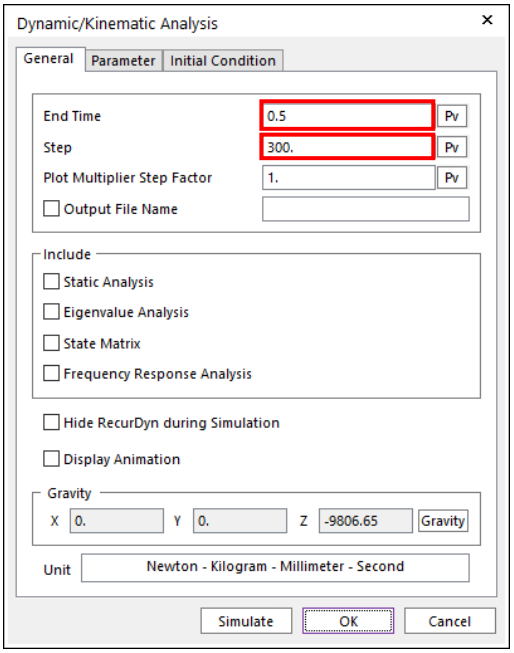


시뮬레이션의 실행

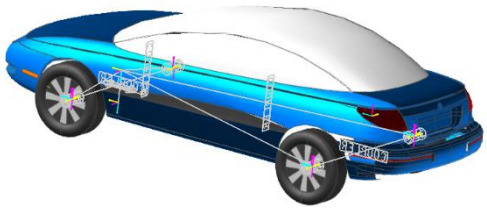
이제 시뮬레이션을 실행하기 위한 설정을 해보겠습니다.

시뮬레이션 실행하기**:**

1. **Analysis** 탭의 **Simulation Type** 그룹에서 **Dyn/Kin** 를 클릭합니다.
2. 오른쪽 그림과 같이 **End Time** 을 **0.5**, **Step** 을 **300** 으로 설정합니다. 이것은 자동차와 바퀴의 모션을 적절한 시간대에서 볼 수 있도록 합니다.
3. **Parameter** 페이지로 가서 **Maximum Time Step** 을 **1.e-003** 로 설정합니다.
4. **Simulate** 를 클릭합니다.



시뮬레이션의 결과 보기



시뮬레이션의 결과 보기**:**

- Analysis 탭의 Animation Control 그룹에 있는 Play 를 클릭합니다.

Tip: 자동차에 움직임이 없다면?

커플러에 설정한 모든 계수에 대한 크기와 표시를 주의 깊게 확인해보십시오. 예를 들어, 두 계수 모두 +1 로 설정되어 있다면 자동차는 움직이지 않을 것입니다.

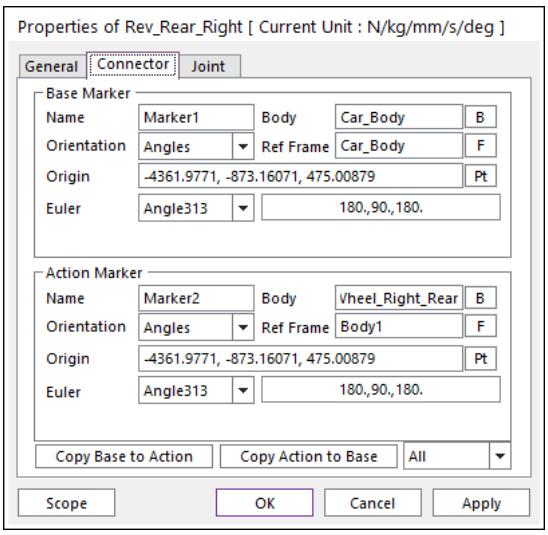
시뮬레이션의 결과를 살펴보면, 이 설정에 따르고 있음을 알 수 있습니다. 회전 조인트를 어떻게 설정하였는지에 따라 자동차 바퀴는 자동차가 주행할 때 잘못된 방향으로 회전할 수 있습니다.

이에 대한 해결은 모든 회전 조인트의 Z 축을 같은 방향으로 지정하는 것이며, 이것은 다음 부분에서 배워보겠습니다.

Revolute 조인트의 수정

회전 조인트 수정하기**:**

1. 바퀴의 Revolute 조인트가 잘못된 방향으로 회전되고 있다면 Database 윈도우의 해당 조인트의 이름 위에서 오른쪽 마우스 버튼을 클릭한 후, **Properties** 다이얼로그 윈도우를 엽니다. 예를 들어, **Rev_Rear_Right** 조인트가 잘못되었다면 다음과 같이 설정합니다.
2. Base 와 Action 마커 모두에 대해서 '0, 90, 0'를 '**0, -90, 0**'로 313 Euler Angle 의 값을 변경합니다.



Note: RecurDyn 은 Euler Angle 의 값이 '**0, -90, 0**'이면 '**180, 90, 180**'로 변경합니다. Euler Angle 의 자세가 같기 때문에 이것은 문제가 되지 않습니다. (Euler angle 설정에 대한 추가적인 정보는 아래의 Tip 에서 다시 설명하겠습니다.)
이것으로 인해 마커의 방향이 변경되어 다이얼로그 윈도우가 위 그림처럼 보여집니다.

---

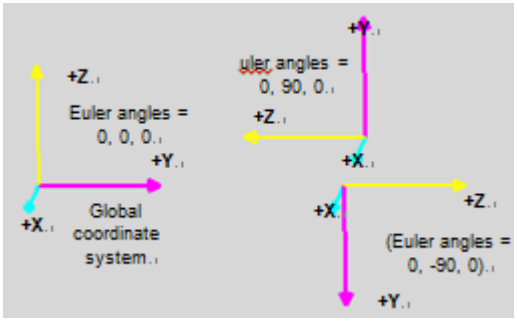
3. **OK** 클릭합니다.
4. 필요 시 다른 조인트들에 대해서도 이 과정을 반복합니다.
5. 문제가 되는 부분을 수정하였으면 모델에 대한 시뮬레이션을 다시 실행합니다.

이제, 자동차에 대한 작업이 완료되었습니다. 다음 장에서는 이 자동차 모델을 복제하고 CoLink 와 연동하여 모델에 대한 설정을 해보도록 하겠습니다.

Tip: Euler Angle 에 대한 설정

RecurDyn 에서 표준 Euler Angle 은 3-1-3 Euler Angle 입니다. 이 각 변환은 다음 과정에 의해서 이루어집니다.

- 먼저, 3 번(또는 Z) 축에 대해서 회전
- 새로운 축인 1 번(또는 X)축에 대해서 회전
- 마지막으로, 새로운 축인 3 번(또는 +Z)축에 대해서 회전



오른쪽 그림에 있는 좌표 시스템은 표준 좌표 시스템이며, 위의 그림의 오른쪽 상단에 보여지는 것처럼 좌표 시스템에서 +X 축으로 90 도만큼 +방향으로 회전한 것을 볼 수 있습니다.
그리고, 위의 그림의 오른쪽 하단에 보여지는 것처럼 좌표 시스템에서 +X 축으로 90 도만큼 -방향으로 회전한 것을 볼 수 있습니다. 0, 90, 0 에서 0, -90, 0 까지 Euler Angle 을 변경하는 것은 Z 축 방향을 변경한다는 것에 유의하십시오.

Chapter 4

모델의 수정

목적

이 장에서는 두 번째 자동차 모델을 만들기 위해서 첫 번째 자동차 모델을 복사하여 그 모델을 수정하여 작업할 것입니다. 그리고 나서, 두 번째 모델의 위치를 이동시킨 후, 다른 색으로 색상을 변경하고, 첫 번째 자동차 모델 보다 느린 속도를 지속적으로 주어 주행시킬 것입니다.

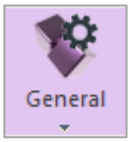
예상 소요 시간

15 분

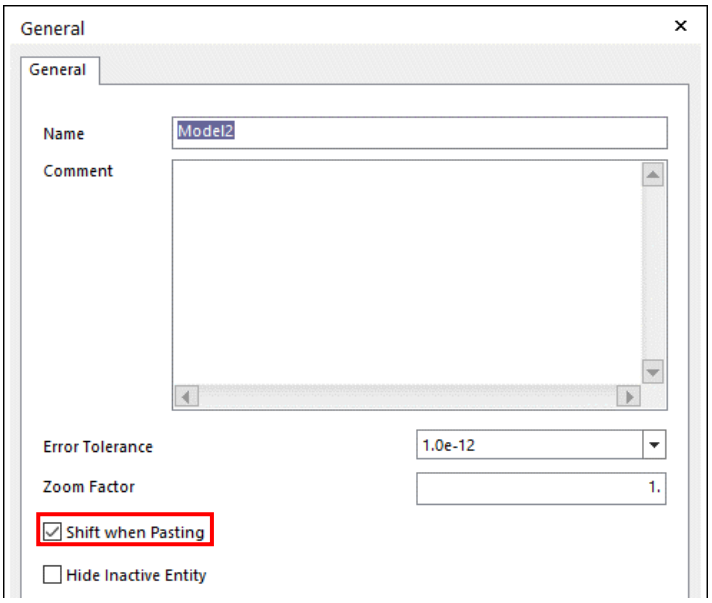
두 번째 자동차 모델의 생성

두 번째 자동차 모델을 생성하기 위해서, 첫 번째 자동차 모델을 복사하여 그 위치를 이동시킨 후, 수정해보겠습니다.

자동차 모델 복사하기

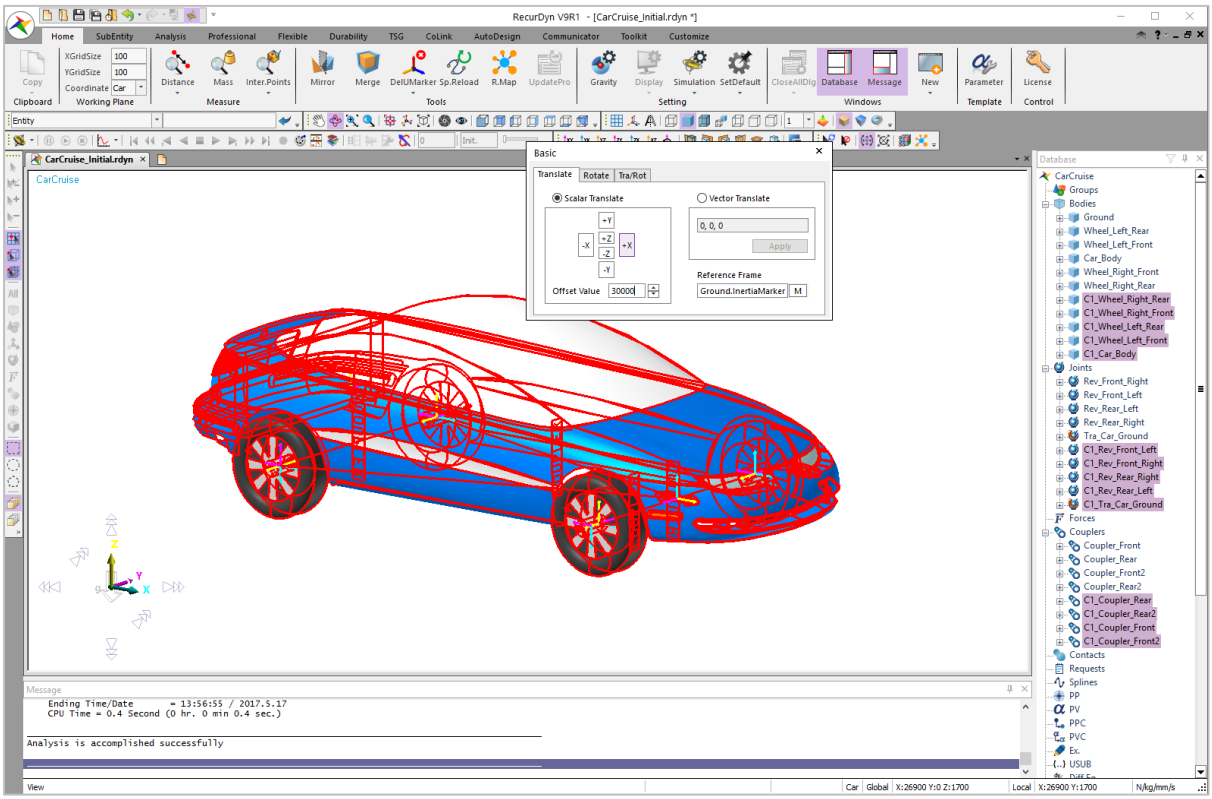


1. **Home** 탭의 **Settings** 그룹에서 **General** 를 클릭하고 오른쪽 그림에서처럼 **Shift When Pasting** 의 체크를 해제합니다.
2. **OK** 를 클릭합니다.
3. 차량 전체를 선택하기 위해서 선택 박스를 차량 전체 주위에 그립니다.
4. **Ctrl** 키와 **C** 를 동시에 눌러서 모델을 복사한 후, **Ctrl** 키와 **V** 를 동시에 눌러서 모델을 붙여 넣습니다.



Tip: 모델을 복사하기 위해 Home 탭의 Clipboard 그룹에서 **Copy** 를 선택한 후 다시 Home 탭의 Clipboard 그룹에서 **Paste** 를 선택하는 방법을 이용해도 됩니다.

5. 모델 전체 선택을 유지한 상태에서 툴바에서 **Object Control** 를 클릭합니다.
6. 다음 그림처럼 복제한 자동차 모델을 **+X** 방향으로 **30,000mm** 이동시킵니다.



두 번째 자동차 모델의 수정

복사한 자동차 모델 수정하기**:**

1. 두 번째 자동차의 Translational 조인트에 해당하는 Properties 다이얼로그 윈도우를 보기 위해, **Database** 윈도우에 있는 **C1_Tra_Car_Ground** 위에서 마우스 오른쪽 버튼을 클릭한 후에 **Property** 를 선택합니다. 이 두 번째 자동차 모델은 일정한 속도로 주행할 것이므로, **Include Friction** 을 선택한 것을 해제합니다.
2. **OK** 를 클릭합니다.
3. **C1_Car_Body** 의 초기 속도 (X)를 50 mph 인 22,350 mm/로 변경하기 위해 주행 속도를 감소시킵니다.

Tip: C1_Car_Body 의 초기 속도 변경하기

1. **Database** 윈도우에 있는 **C1_Car_ Body** 위에서 마우스 오른쪽 버튼을 클릭하여 **Property** 를 선택합니다.
2. **Body** 페이지를 선택합니다.
3. **Initial Velocity** 버튼을 클릭합니다.
4. **Translational Velocity** 의 **X** 를 클릭합니다.
5. 값을 **22350** 으로 입력합니다.

---

6. **Translational Velocity** 의 **Reference Marker** 를 **Ground.InertiaMarker** 로 정의합니다.
7. **Rotational Velocity** 의 **Reference Marker** 를 **C1_Car_Body.CM** 로 정의합니다.
8. **Close** 를 클릭합니다.
9. **OK** 를 클릭합니다.

---

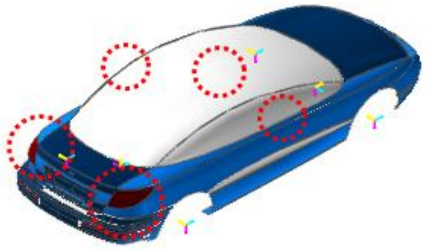
4. 이제 도로 위를 천천히 주행하는 것을 더 잘 보이게 하기 위해서 자동차 모델의 색을 변경할 것입니다.
   - **Body Edit** 모드로 들어가서 복사한 자동차의 바디인 **C1_Car_Body** 를 더블 클릭합니다.

Tip: **Body Edit** 모드로 들어가려면 **Database** 윈도우에 있는 **C1_Car_Body** 위에서 오른쪽 마우스 버튼을 클릭한 후 **Edit** 를 클릭합니다.

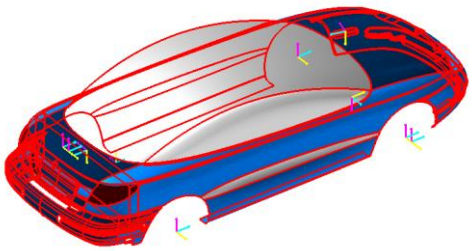
- 자동차 모델의 바디 전체를 드래그로 선택합니다. 그리고 나서, 차량 창문을 해제하기 위해 Ctrl 키를 누른 채로 하얀색 창문 세 부분과 빨간색 미등 두 부분을 클릭합니다

Tip: 자동차의 뒤쪽을 보기 위해서 자동차 모델을 회전시키거나 Database 윈도우에서 다음 아이템의 선택을 해제합니다.

- C1_C1_C2_ImportSurface1
- C1_C1_C3_ImportSurface1
- C1_C3_ImportService1
- C1_C2_C1_ImportSurface1
- C1_C2_C2_ImportSurface1

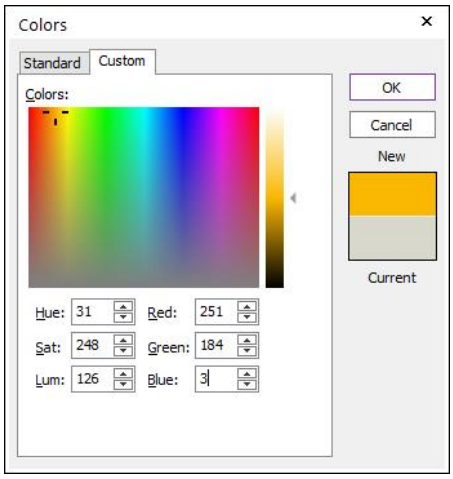


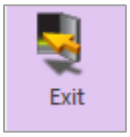
이제, 다음과 같이 모델이 보일 것입니다.



- 선택된 바디 위에서 오른 쪽 마우스 버튼을 클릭한 후, **Property** 를 클릭합니다.

- **Graphic Property** 페이지를 클릭한 후 **Color** 를 다른 색으로 설정합니다.
- Colors 다이얼로그 윈도우에서 **Custom** 페이지로 가서, **Red, Green, Blue** 값을 **251, 184, 3** 으로 변경합니다.
- **OK** 버튼을 눌러서 변경한 색을 적용합니다.





5. **Body Edit** 모드를 종료하기 위해 **Exit** 를 클릭합니다.

이제, CoLink 와 모델을 통합시키기 위한 준비가 끝났습니다

Chapter

5

CoLink 와의 통합

목적

이 장에서는 CoLink 와의 통합을 위해 모델을 수정하고 CoLink 모델을 생성할 것입니다. 그리고 그 시스템에 대해 시뮬레이션을 실행하여 그 결과를 Plot 을 통해 확인해보겠습니다.

예상 소요 시간

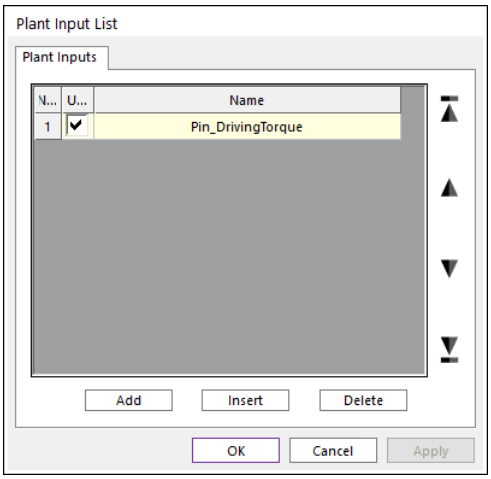
15 분

Plant Input 의 생성

제어 시스템에서 모델에 대한 첫 Input 을 생성해보겠습니다.

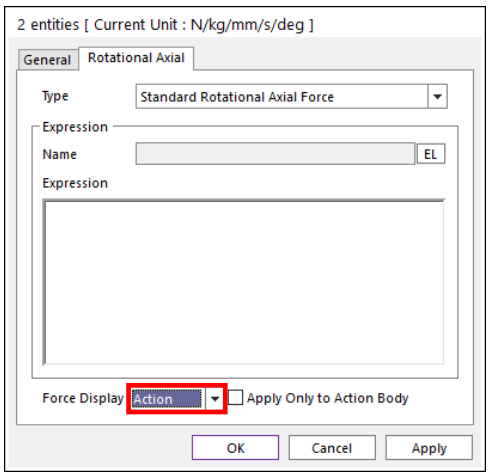
Plant Input 생성하기**:**

1. **CoLink** 탭의 **CoLink** 그룹에서 **Plant Input** 을 클릭합니다.
2. **Add** 를 클릭한 후 Plant 의 이름을 **Pin_DrivingTorque** 로 변경합니다.
3. **OK** 를 클릭합니다.



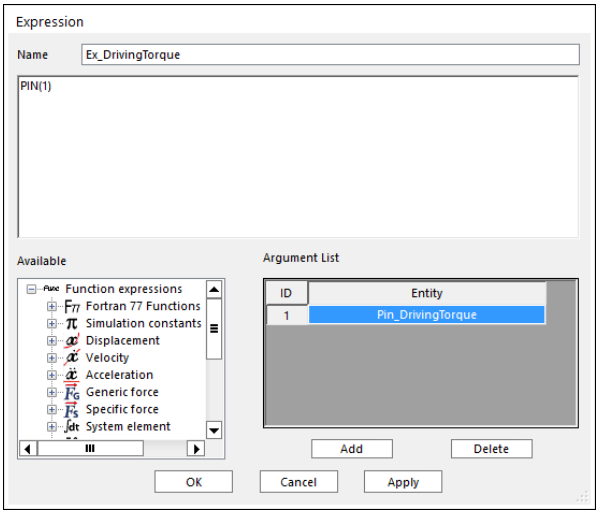
토크 생성하기**:**

1. **Professional** 탭의 **Force** 그룹에서 **Rotational Axial** 을 클릭합니다.
2. Force 생성 방법을 **Joint** 로 설정한 후, **Rev_Front_Right** 를 클릭합니다.
3. **Rev_Front_Left** 도 위와 동일한 방법을 적용합니다.
4. **RotationalAxial1** 을 클릭한 후, Ctrl 키를 누른 채로 **RotationalAxial2** 를 클릭합니다. 두 가지를 모두 선택한 상태에서, 오른 쪽 마우스 버튼을 클릭한 후, **Property** 를 클릭합니다.



애니메이션에서 계산된 토크를 보기 위해 **Force Display** 를 **Action** 으로 설정합니다.

5. 주행 토크에 대한 식을 입력하기 위해서 **EL** 버튼을 클릭합니다.
6. **Expression List** 다이얼로그 윈도우가 나타나면, **Create** 를 클릭합니다.
7. Name 을 **Ex_DrivingTorque** 로 입력하고 수식을 **PIN(1)**으로 입력합니다.
8. **Argument List** 에서 **Add** 클릭한 후 Database 윈도우에서 **Pin_DrivingTorque** 를 첫 번째 Entity 로 드래그합니다.



Tip: 노란색 박스를 더블 클릭하여 **Pin_DrivingTorque** 를 직접 입력해도 됩니다.

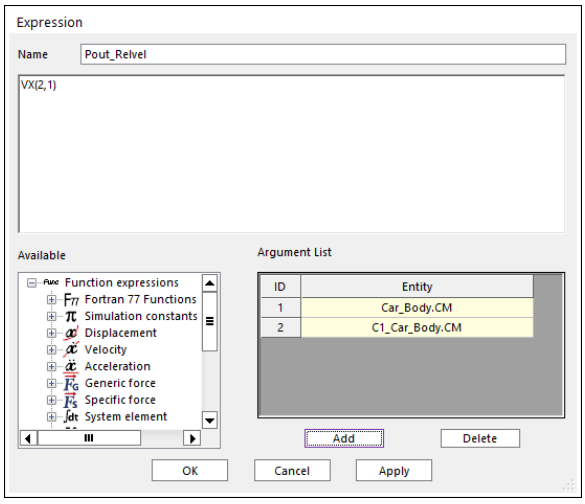
9. **OK** 버튼을 눌러서 모든 변경 사항을 적용합니다.

Plant Output 의 생성

제어 시스템에서 두 자동차 모델 사이의 거리를 고려하여 상대 속도를 출력해보겠습니다.

Plant Output 생성하기**:**

1. **CoLink** 탭의 **CoLink** 그룹에서 **Plant Output** 을 클릭합니다.
2. **Plant Output List** 다이얼로그 윈도우에서 **Add** 를 클릭합니다.
3. Expression 다이얼로그 윈도우가 나타나면, **Name** 을 **Pout_RelVel** 로 입력하고 수식을 **VX(2,1)**로 입력합니다.
4. **Argument List** 에 두 개의 Entity 를 추가하고 **Car_Body** 와 **C1_Car_Body** 에 대한 **CM** 마커를 노란색 박스 안에 오른쪽 그림과 같이 입력합니다.
5. **OK** 를 클릭하여 Plant Output List 로 돌아옵니다.



6. **Plant Output List** 에서 다시 **Add** 를 클릭하고 **VX** 대신에 **DX** 로 함수 종류를 바꾸어 3 번~5 번 과정을 반복하고 **Name** 을 **Pout_RelVel** 대신 **Pout_Distance** 로 입력합니다.
7. **OK** 를 클릭합니다.

CoLink 모델 생성하기

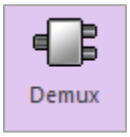
이제, CoLink 를 실행하여 제어 시스템을 생성할 것입니다. 이것은 블록 다이어그램을 생성하는 것을 포함하며, 블록은 표준 PID 컨트롤러, Sum 블록을 통해 일정한 기준 신호를 얻는 RecurDyn 모델을 나타내기 위해 생성될 것입니다.

CoLink 모델 생성하기**:**

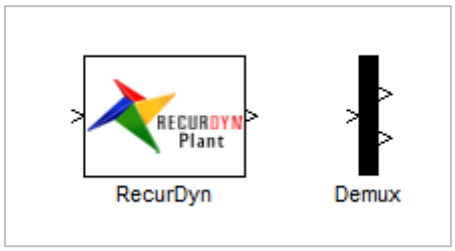
1. **CoLink** 탭의 **CoLink** 그룹에서 **Run CoLink** 를 클릭합니다.

   모델이 없는 CoLink 빈 화면이 나타날 것입니다.

2. **Connector** 탭에서 **Link** 그룹의 **RecurDyn** 블록을 클릭한 후, 아래의 그림과 같이 모델링 윈도우의 3/2 지점에 놓습니다



3. 아래의 그림처럼 **Connector** 탭에서 **Connector** 그룹의 **Demux** 블록을 클릭한 후, 아래

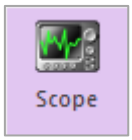


그림과 같이 **RecurDyn** 블록의 오른쪽에 놓습니다.

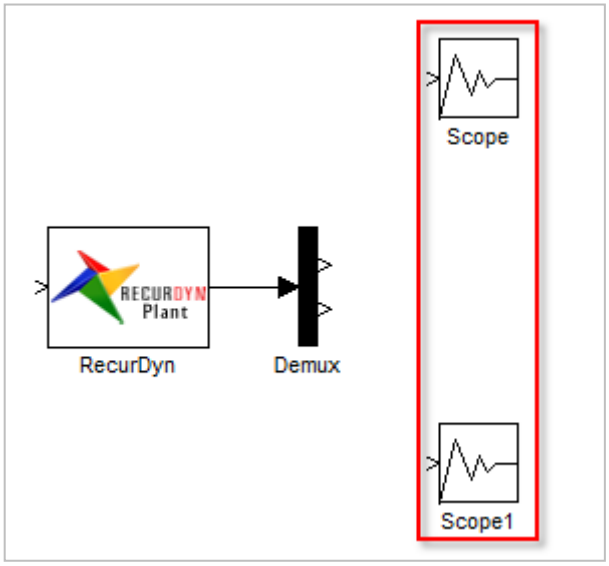
이것은 RecurDyn 모델에 생성한 두 개의 출력에 대하여 **RecurDyn** 블록을 **RelVel** 과 **Distance** 신호 출력으로 분리합니다.

4. 두 블록을 연결하기 위해서 **RecurDyn** 블록 클릭하고 **Ctrl** key 를 누른 상태에서 **Demux** 블록을 클릭합니다. CoLink 모델 생성시 나머지 블록들도 이와 같은 방법을 사용하여 완성할 것입니다.

시뮬레이션의 결과를 보기 원한다면 상대 속도 신호화 거리 신호에 대한 Scope 를 생성합니다. Scope 는 x-y Plot 으로 보이며, X 축은 시뮬레이션 시간, Y 축은 입력한 데이터를 나타냅니다.



5. **General** 탭에서 **Output** 그룹의 **Scope** 블록을 클릭한 후, 아래 그림에서처럼 오른쪽 상단과 하단에 **Scope** 블록을 놓습니다.

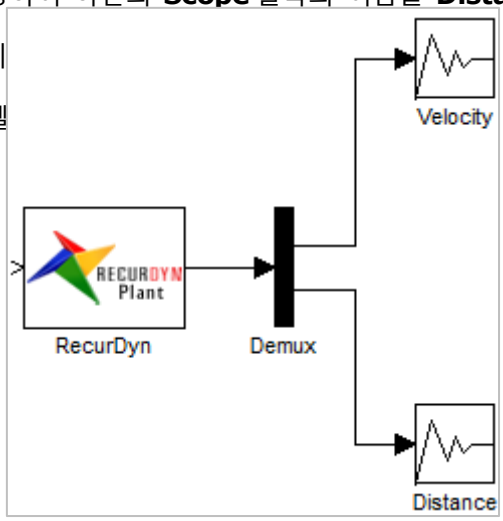


6. 5 번에서 했던 방법을 이용하여 두 개의 **Scope** 블록을 **Demux** 블록과 연결시킵니다.

7. 상단에 있는 **Scope** 블록 아래의 글씨를 더블 클릭하여 이름을 **Velocity** 로 변경합니다. 그리고 같은 방법을 이용하여 하단의 **Scope** 블록의 이름을 **Distance** 로 변경합니다.

8. **CarCruise_Control** 이

   그러면 다음과 같이 모델

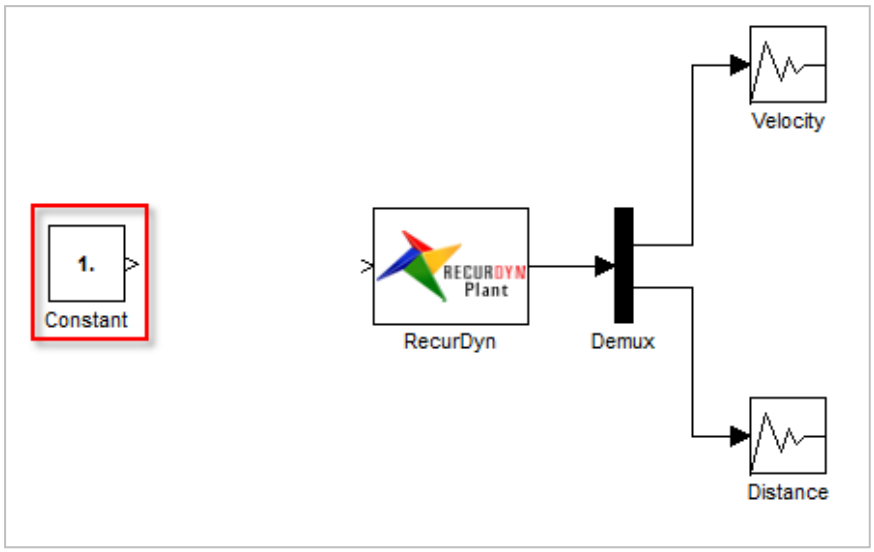


비례 피드백 제어 생성하기

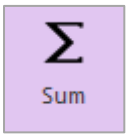
이제, 기준 입력 신호와 피드백 루프를 생성하여 비례 제어 컨트롤러를 추가할 것입니다. 이것은 PID 컨트롤러를 만들기 위한 첫 과정입니다.

비례 제어 생성하기**:**

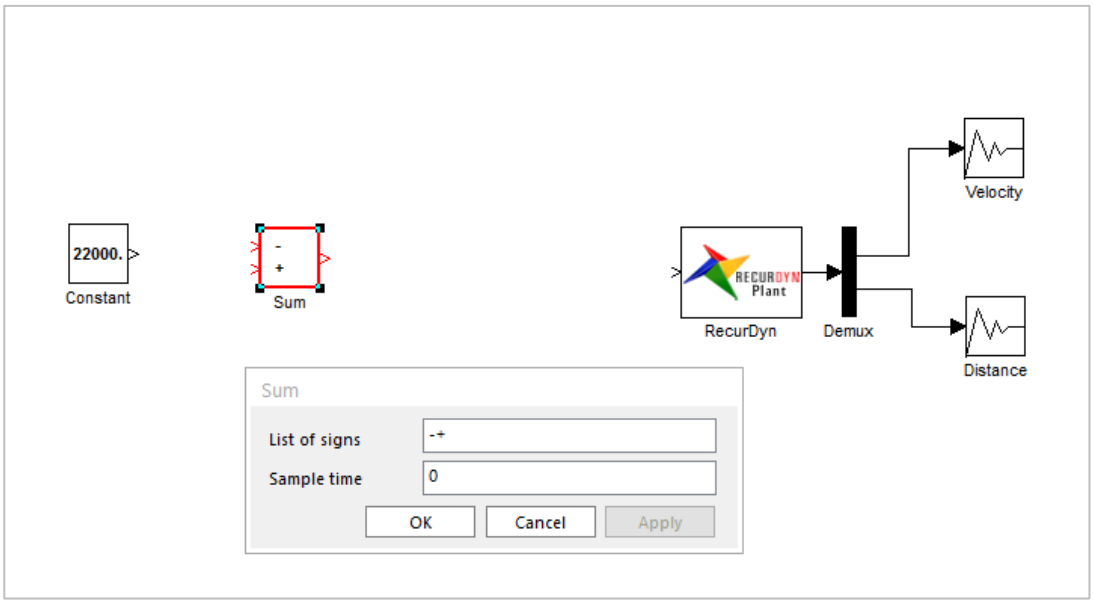
1. **General** 탭에서 **Source** 그룹의 **Constant** 블록을 클릭한 후, 아래의 그림처럼 모델링 윈도우의 왼쪽 중앙에서 떨어진 곳에 놓습니다.



2. 이 **Constant** 블록은 자동차 사이에서 원하는 거리를 나타내는 것입니다. 모델에서 **Constant** 블록을 더블 클릭하여 그 값을 **22,000** 으로 설정합니다.



3. **Math** 탭에서 **Math** 그룹의 **Sum** 블록을 클릭한 후, **Constant** 블록 오른 편에 놓습니다.
4. **Sum** 블록의 상단 입력 자리에 **Constant** 블록을 연결시키고 **Sum** 블록을 더블 클릭하여 아래의 그림과 같이 **List of sign** 를 **- +**로 설정합니다.

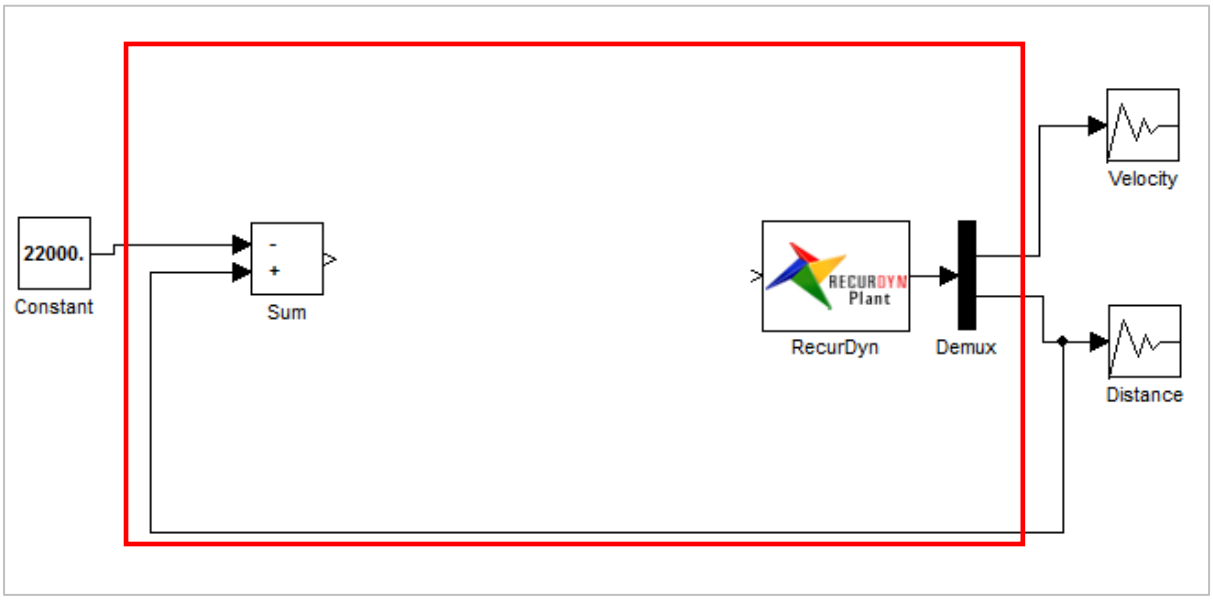


5. **OK** 를 클릭합니다.

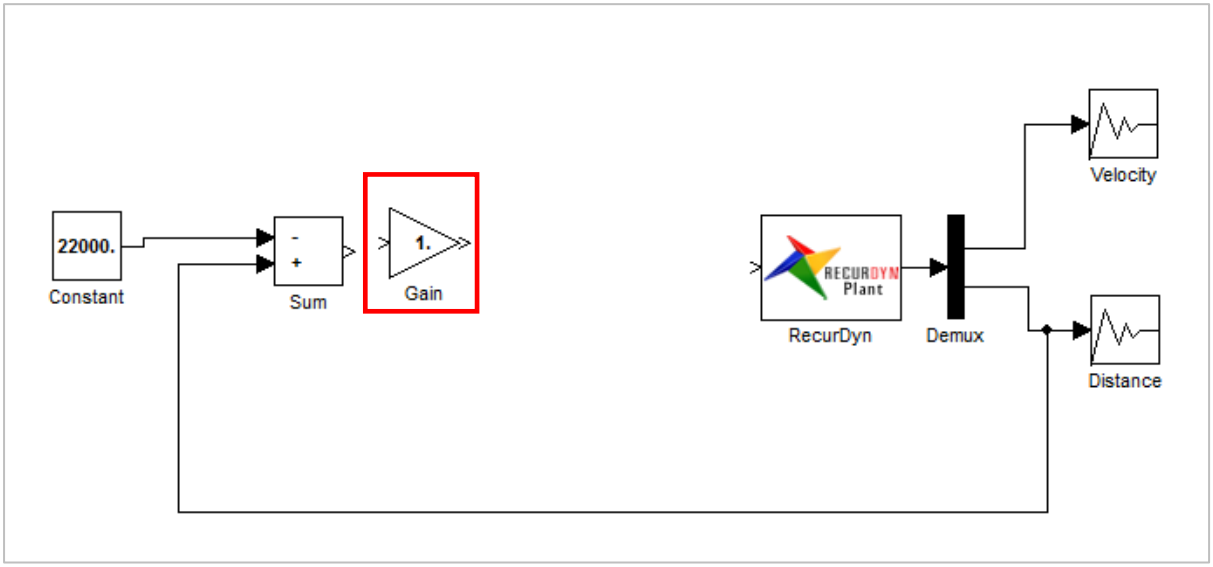
   이제, Distance 신호와 Sum 블록의 하단 입력 자리를 연결시켜보겠습니다. 이는 자동차 모델 사이의 실제 거리에 대하여 오차 범위의 신호를 생성합니다.

6. **Distance** 출력과 **Sum** 블록을 다음과 같이 연결합니다.

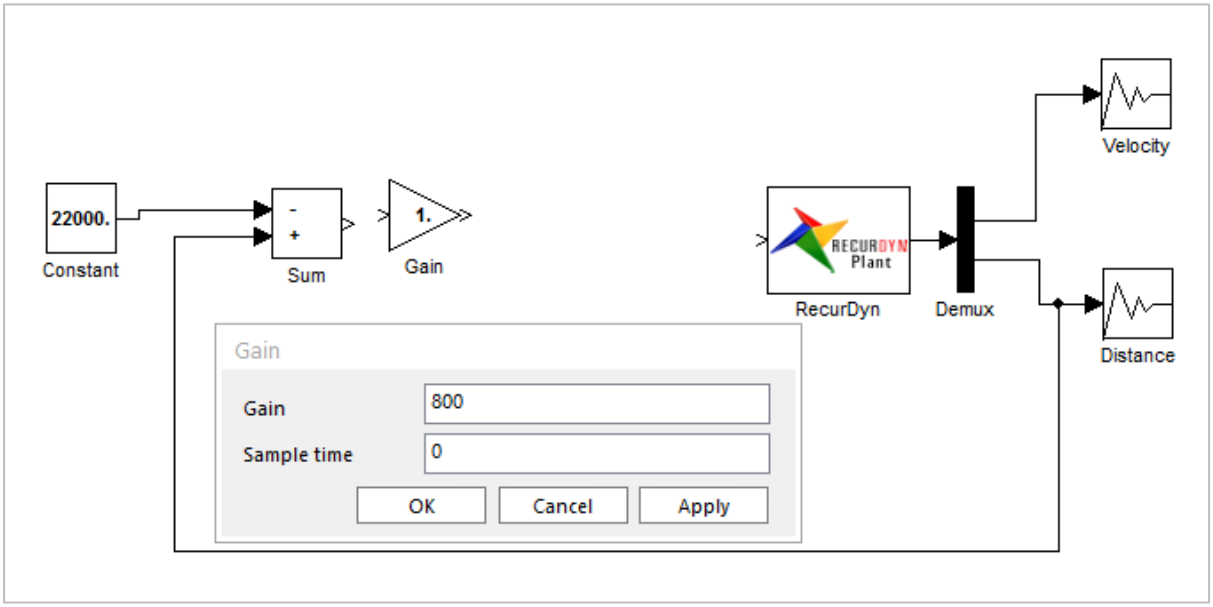
   - **Distance Scope** 에 대해서 **Demux** 블록에 연결된 선을 클릭합니다.
   - **Distance Scope** 에 대해서 **Demux** 블록에 연결된 선의 구부러진 부분에서 오른쪽 마우스 버튼을 클릭하여 Sum 블록의 맨 아래 입력 자리로 선을 드래그한 후 오른쪽 마우스 버튼을 누릅니다. 이로 인해 처음 오른 쪽 마우스 버튼을 클릭한 지점에서 연결점이 만들어 졌습니다. 이제 다음과 같이 모델이 보일 것입니다.



7. 리본 메뉴에서 **Math** 를 클릭하여 **Gain** 을 모델링 윈도우로 드래그하고 아래의 그림처럼 **Sum** 블록의 오른 편에 놓습니다.



8. **800** 으로 **Gain** 을 설정하고 아래의 그림처럼 두 블록을 연결합니다.

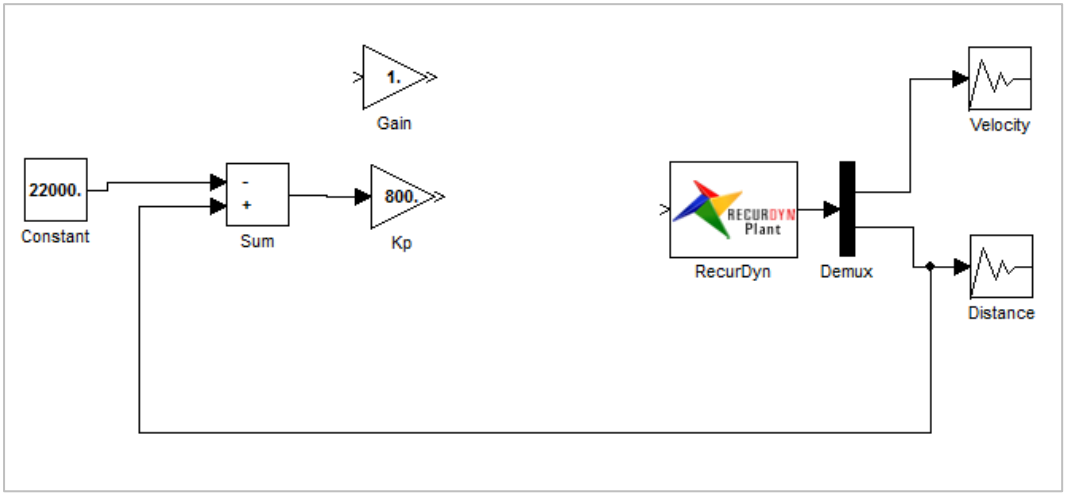


9. **PID** 컨트롤러의 비례 부분에 대해 값을 준 것이므로 **Gain** 블록의 이름을 **Kp** 로 재설정합니다.

미분 및 적분 제어의 추가

이제, 제어 컨트롤러에 대해서 미분과 적분 피드백을 추가해보겠습니다. Velocity 신호는 Distance 의 미분 신호이기 때문에 미분 이득 블록으로 입력한 것처럼 그 신호를 직접적으로 이용할

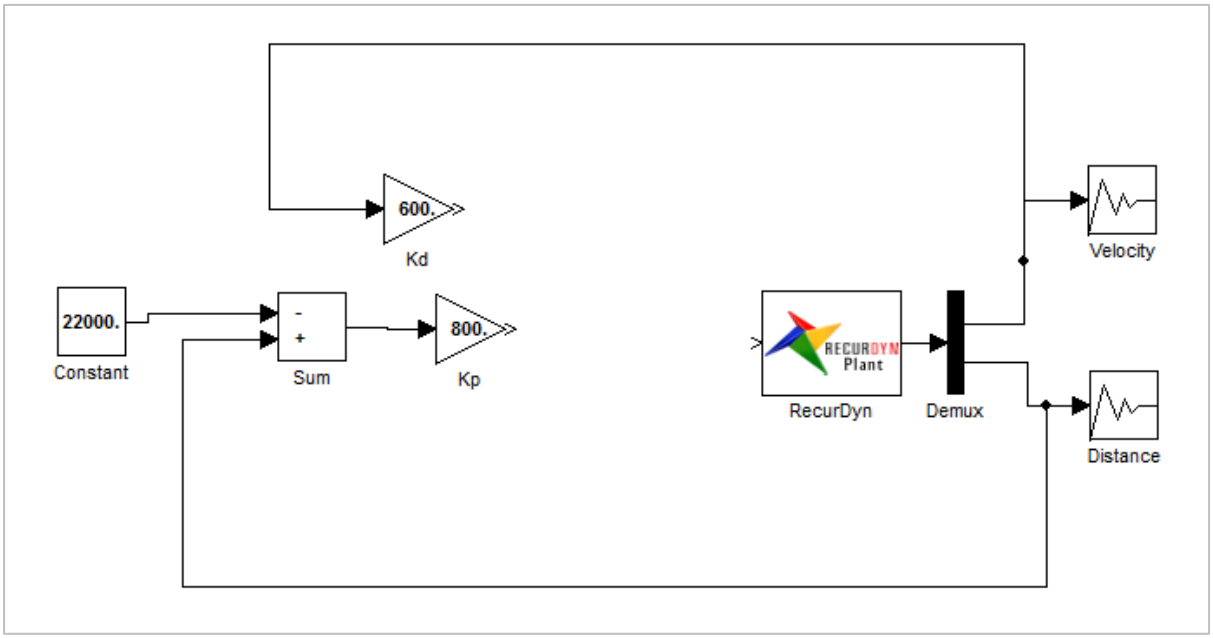
수 있습니다. 적분 제어에 대해서는 Integrator 블록을 추가하고 Integral Gain 블록을 입력한 것처럼 Integrator 블록을 이용해 보겠습니다.



미분 제어 추가하기**:**

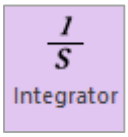
1. **Math** 탭에서 **Math** 그룹의 **Gain** 블록을 클릭한 후, **Kp** 로 지정된 첫 번째 **Gain** 블록 위에 놓습니다.
2. 미분 제어를 위한 Gain 으로서 **Kd** 라고 그 블록의 이름을 재설정합니다.
3. **Velocity** 출력과 **Gain** 블록을 아래의 그림처럼 연결합니다.
   - **Velocity Scope** 에 대해서 **Demux** 블록에 연결된 선의 구부러진 부분에서 오른쪽 마우스 버튼을 클릭하여 **Gain** 블록의 맨 아래 입력 자리로 선을 드래그한 후 오른쪽 마우스 버튼을 누릅니다.
   - 

4. **Gain** 값을 **600** 으로 설정합니다. 이제 모델이 다음과 같이 보일 것입니다.



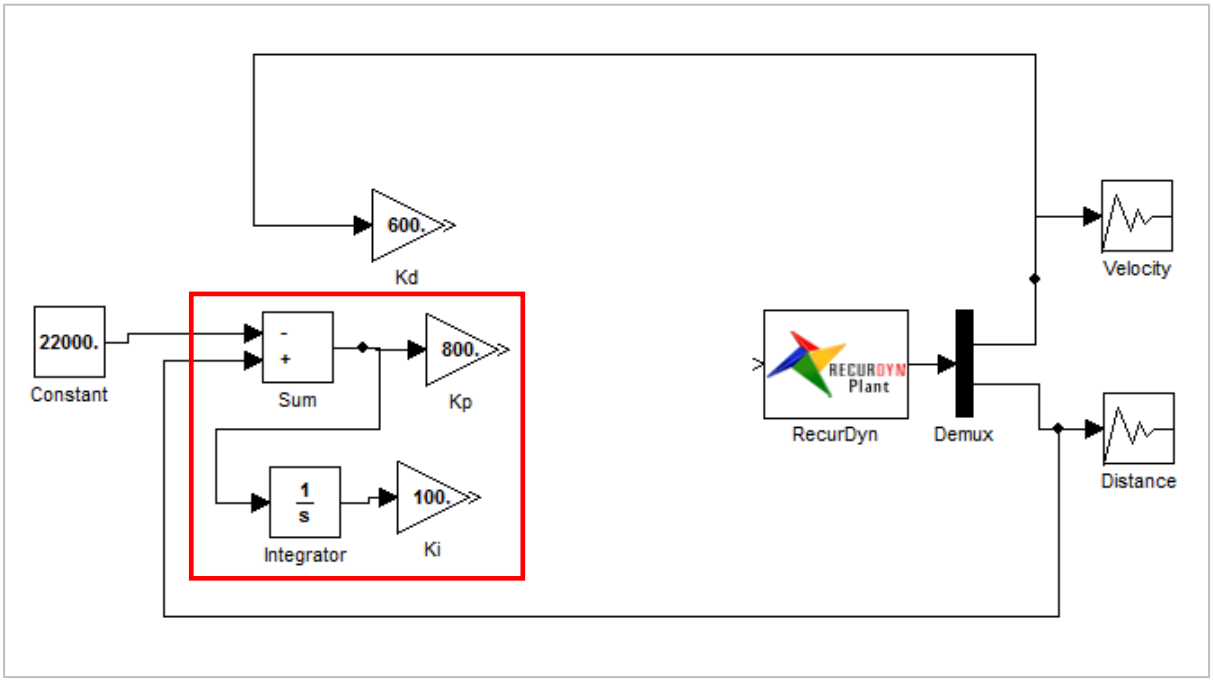
적분 제어 추가하기**:**

1. **Math** 탭에서 **Math** 그룹의 **Gain** 블록을 클릭한 후, 다음 그림과 같이 **Kp** (Gain) 블록 아래에 놓습니다.
2. **Ki** 로 블록의 이름을 재지정하고 값을 **100** 으로 설정합니다.



3. **Continuous and Discrete** 탭에서 **Continuous** 그룹의 **Integrator** 블록을 클릭한 후, **Ki** (Gain) 블록 왼 편에 놓습니다.
4. **Sum** 블록과 **Kp** 블록을 연결한 선에서 오른 쪽 마우스 버튼을 클릭하여 **Integrator** 블록으로 연결합니다.
5. **Ki** 블록을 **Integrator** 블록과 연결합니다.

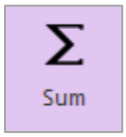
이제 다음과 같이 모델이 보일 것입니다.



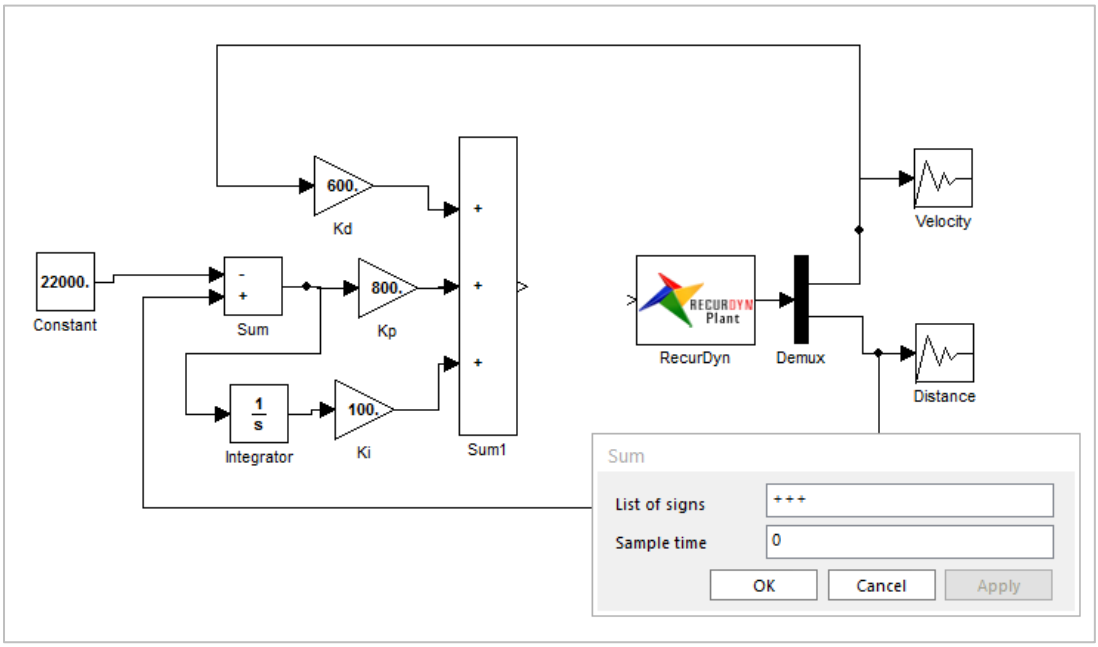
루프의 폐쇄

이제, 3 개의 모든 피드백 신호를 추가하고 RecurDyn Plant 블록에 Sum 블록을 연결하여 루프를 폐쇄해보겠습니다.

루프 폐쇄하기**:**



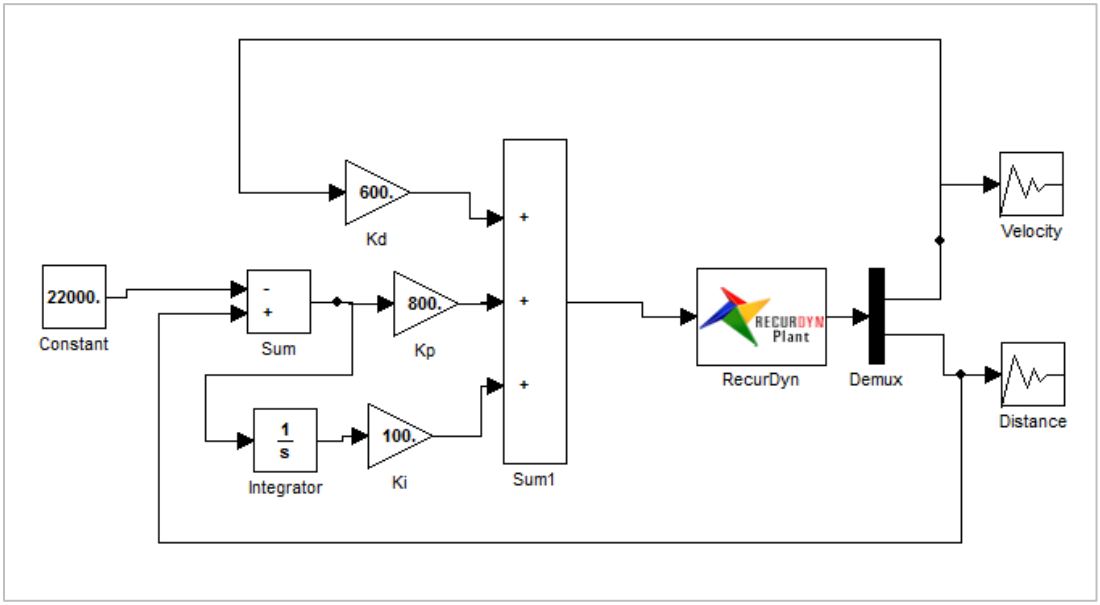
1. **Math** 탭에서 **Math** 그룹의 **Sum** 블록을 클릭한 후, **Gain** 블록과 **RecurDyn** 블록 사이에 아래의 그림과 같이 놓습니다.
2. 새롭게 추가된 **Sum** 블록을 더블 클릭하여 **List of signs** 를 +++로 변경합니다.



이것은 피드백 제어기의 모든 세 요소가 Sum 블록에 대한 입력으로 정의되도록 하기 위해 3 개의 입력을 추가한 것입니다.

3. Summer 에 대한 **Gain** 블록들을 연결하고 **RecurDyn Plant** 블록과 Summer 를 연결합니다.

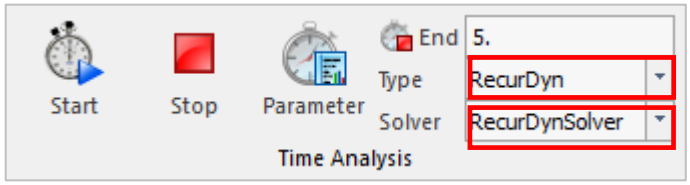
이제, 제어기 모델을 만들기 위한 모든 작업을 완료하였습니다. 작업이 완료되었다면, 다음과 같이 보여야 합니다.



모델의 시뮬레이션 실행

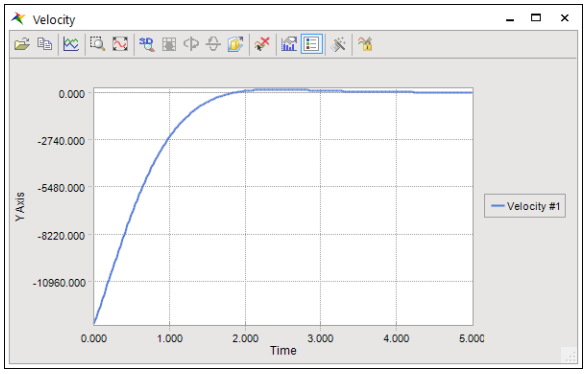
이제 모델에 대한 시뮬레이션을 실행하여 그 결과를 살펴보겠습니다.

모델의 시뮬레이션 실행하기**:**



1. **Type** 을 **RecurDyn** 변경하고 **Solver** 를 **RecurDynSolver** 으로 변경합니다.

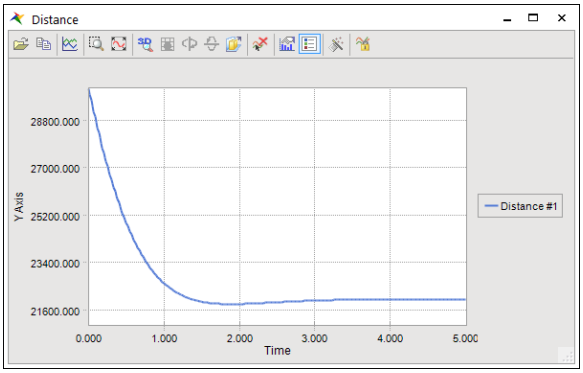
2. **Simulation** 탭에서 **Time Analysis** 그룹에 있는 **Start** 아이콘을 클릭합니다.
3. 시뮬레이션이 완료되면 **Velocity** 와 **Distance** 에 대한 **Scope** 를 더블 클릭하여 그 결과를 봅니다.



4. Plot 결과는 오른 쪽 그림처럼 보여집니다. 두 자동차 모델의 Velocity 는 과도 구간 이후에 고정된 거리를 유지함에 따라 0 에서 수렴합니다. Distance Plot 은 마지막 값이 22,000 mm (22 m)이라는 것을 보여줍니다.

   반응 시간은 다소 느리지만 그 오버슈트는 매우 큽니다.

   이제, 이러한 결과를 바탕으로 시뮬레이션을 다시 실행해 보겠습니다.



Tip: Co-Simulation 이 실행되지 않는다면?
Co-Simulation 이 실행되지 않는다면 아마도 Server Busy 다이얼로그 윈도우가 나타날 것입니다. 그 다이얼로그 윈도우가 나타나면 다음과 같은 작업을 실행합니다.

- **Switch** 를 클릭하여 **RecurDyn Output** 윈도우에서 메시지를 검토합니다.
- CoLink 모델을 찾을 수 없다는 에러 메시지가 나온다면 RecurDyn 과 CoLink 모델의 경로를 확인합니다. 두 모델이 같은 경로에 있는지 확인합니다.
- 그렇지 않다면, 두 모델을 같은 경로에 두고 RecurDyn 과 CoLink 를 다시 실행합니다. 그러면, 시뮬레이션이 제대로 실행될 것입니다.

제어 **Gain** 조절 하기**:**

1. **Kp** 블록을 더블 클릭하여 **Gain** 을 **1000** 으로 변경합니다. 추가적으로 **Kd** 의 값을 **800** 으로 변경합니다.
2. **Start** 아이콘을 클릭하여 시뮬레이션을 실행시킵니다.
3. **Scope** 에서 보인 **Plot** 이 변경되지 않았음을 확인합니다.

   이 Plot 들은 현재 시뮬레이션의 결과를 보기 위하여 시뮬레이션의 초기 값만을 업데이트하므로 시뮬레이션을 다시 실행하겠습니다. 그러면 Plot 이 다시 업데이트 될 것입니다.

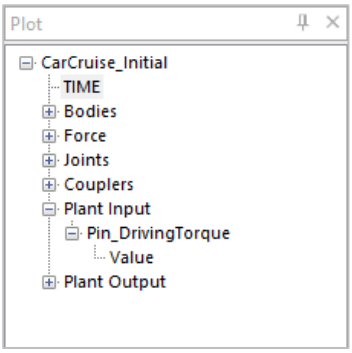
   이제 시스템에 설정 값들이 적용된 것 같습니다. 이제, RecurDyn Plot 윈도우로 돌아가서 Plot 의 결과를 살펴보겠습니다.

결과 보기

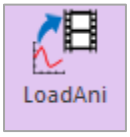
이제, 두 자동차 모델 사이의 주행 토크와 상대 속도, 거리를 Plot 으로 그려보고 이에 대한 애니메이션을 살펴보겠습니다.

결과 보기**:**

1. CoLink 프로그램을 열어둔 채로 RecurDyn 프로그램으로 돌아옵니다.
2. **Plot** 윈도우를 엽니다.

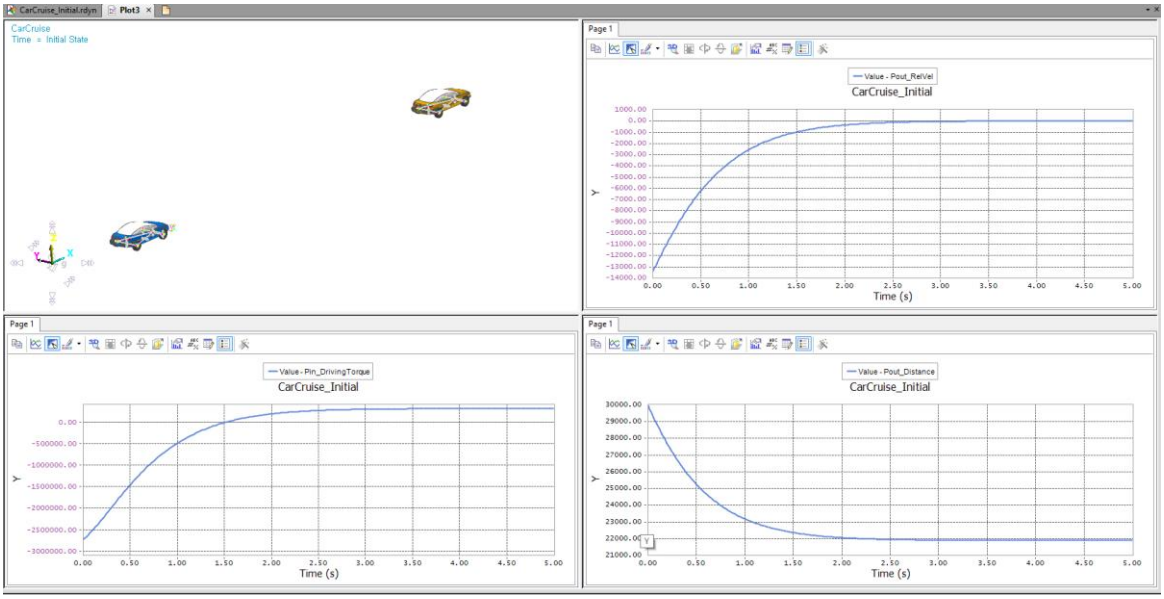


3. 윈도우를 4 개의 Pane 으로 나누기 위해서 **Home** 탭의 **View** 그룹에서 **Show all Windows** 를 클릭합니다.
4. 왼쪽 상단의 Pane 을 활성화시키기 위해 클릭합니다.



5. **Tool** 탭의 **Animation** 그룹에서 **Load Animation** 아이콘을 클릭하여 애니메이션을 로드시킵니다.
6. 해당 Pane 에 대한 보기를 삭제하라는 경고가 나타납니다. Plot 에 어떠한 데이터가 없다면, **Yes** 를 클릭하고 작업을 계속 진행합니다.
7. 애니메이션의 **View** 와 **Rendering** 모드를 수정합니다.
8. 왼쪽 하단 Pane 을 클릭하여 활성화시킨 후 아래 그림의 오른쪽에 보이는 것처럼 **Driving Torque** 에 대한 Plot 을 그립니다.
9. 오른 쪽 상단 Pane 을 클릭하여 활성화시킨 후 첫 번째 Plant Output 인 **Relative Velocity** 에 대한 Plot 을 그립니다.
10. 오른쪽 하단 Pane 을 클릭하여 활성화시킨 후 두 번째 Plant Output 인 두 자동차 사이의 **Distance** 에 대한 Plot 을 그립니다. 그러면 다음 그림처럼 Plot 결과가 보여집니다.

이제, 이 튜토리얼의 주요 과정을 완료하였습니다. 모델의 성능을 개선하기 위한 Gain 제어 또는 자동차 모델들 사이의 실제 거리를 기초로 하여 그 거리가 조정 가능하도록 제어



컨트롤러에서 복잡성을 증가시키거나 충격에 대한 시간의 복잡성을 증가시키는 등의 작업들은 추가 과정에서 다루도록 하겠습니다.

다음 장에서는 자동차가 주행할 때 원하는 속도로 주행 속도의 조정이 가능하도록 시스템을 개선해보겠습니다.

Chapter

6

CoLink 모델 시스템의 확장

목적

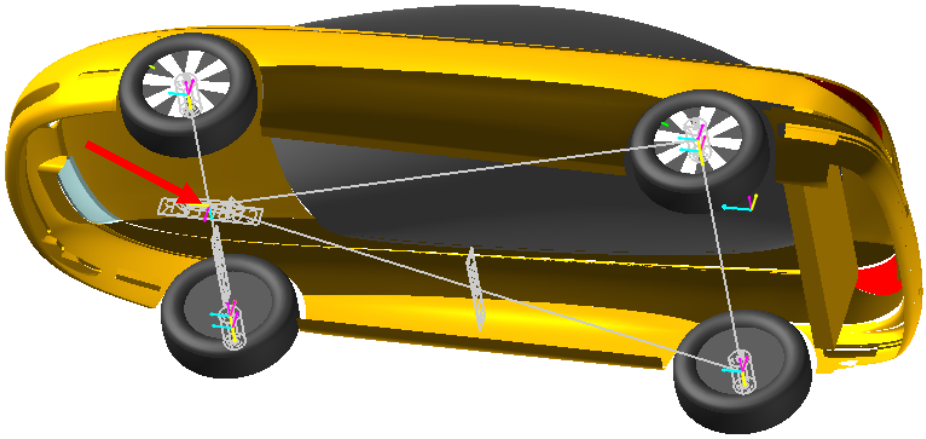
이 장에서는 조정 가능한 크루즈 제어를 위해 CoLink 모델을 수정해보겠습니다. 그리고 나서, 시스템에 대한 시뮬레이션을 실행하고 그에 대한 결과를 살펴보도록 하겠습니다.

예상 소요 시간

15 분

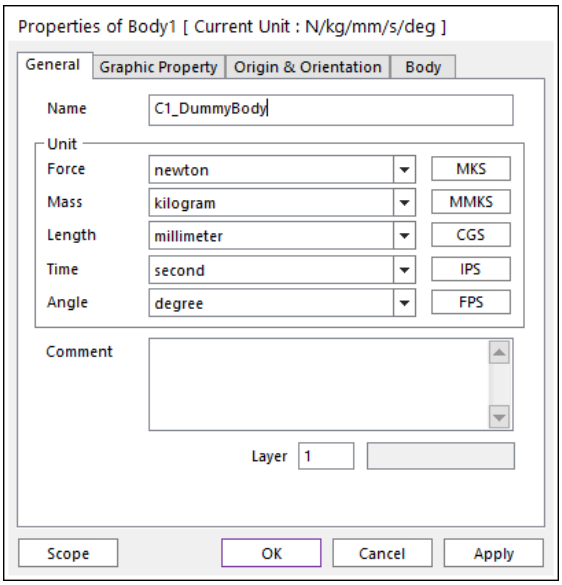
RecurDyn 모델의 수정

이제, 조정 가능한 크루즈 제어 시스템의 수정을 위해 RecurDyn 모델을 먼저 수정할 것입니다. 이 작업은 앞 쪽 자동차 모델의 측면 모션에 Translational 조인트를 추가하고, 모델의 제어 시스템에 Plant Output 의 추가하는 과정을 필요로 합니다. 먼저, 앞 쪽 자동차의 앞 차축 중앙에 더미 바디를 생성해보겠습니다.

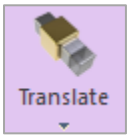


더미 부분 생성하기:

1. **Professional** 탭의 **Body** 그룹에서 **Ellipsoid** 를 클릭합니다.
2. **Point, Distance** 생성 방법을 이용하여, 다음 그림과 같이 앞 쪽 자동차의 기존의 **Translational** 조인트와 커플러에 있는 마커들을 선택합니다.
3. 구의 반지름을 **35** 로 입력합니다.
4. 구에서 오른쪽 마우스 버튼을 클릭한 후 **Properties** 를 클릭합니다.
5. **General** 페이지에서 **Name** 을 오른쪽 그림과 같이 **C1_DummyBody** 로 입력합니다.
6. **OK** 를 클릭합니다.



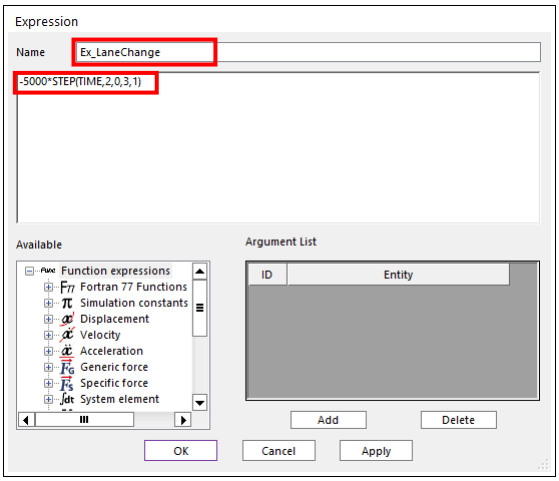
Translational 조인트 생성하기**:**



1. **Professional** 탭의 **Joint** 그룹에서 **Translate** 를 클릭합니다.
2. **Body, Body, Point, Direction** 생성 방법을 이용하여 **Ground** 와 **C1_DummyBody**, 구의 중심을 클릭하고, direction 으로 '0, 1, 0'을 입력합니다.
   (또는 **Ground InertiaMarker** 의 **+Y-axis** 를 클릭합니다.)

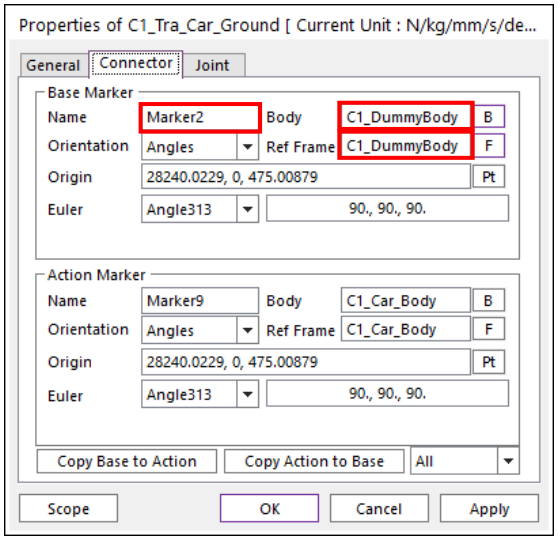
새로 생성한 조인트의 속성 변경하기**:**

1. 새로 생성한 조인트에서 오른쪽 마우스 버튼을 클릭한 후 **Property** 를 클릭합니다.
2. **General** 페이지에서 그 조인트의 이름을 **C1_TraLaneChange** 로 입력합니다.
3. **Joint** 페이지에서 차선 변경에 대한 수식을 생성합니다.
   - **Include Motion** 옆에 있는 체크 박스를 체크한 후, **Motion** 을 클릭합니다.
   - **Expression** 다이얼로그 윈도우를 열기 위해 **EL** 버튼을 클릭한 후 새로운 수식을 생성하기 위해 **Create** 를 클릭합니다.
   - **Name** 을 **Ex_LaneChange** 로 입력하고 다음의 수식을 입력합니다.
   - -5000*STEP(TIME,2,0,3,1)
   - 이 수식은 시뮬레이션이 2 초에 시작해서 3 초에 끝날 때 자동차가 오른쪽으로(-Y 축 방향으로) 5m 이동하게 할 것입니다.
   - Expression 다이얼로그 윈도우가 오른쪽과 같이 보여질 것입니다.
   - 모든 변경 사항을 적용하기 위해 **OK** 를 클릭하고 다이얼로그 윈도우를 모두 닫습니다.



이제, 자동차 바디(**C1_Car_Body**)와 Ground 대신에 구(**C1_DummyBody**)를 연결하기 위해서 기존의 Translational 조인트를 수정할 것입니다.

4. **Database** 윈도우의 **C1_Tra_Car_Ground** 에서 오른 쪽 마우스 버튼을 클릭한 후 **Property** 를 선택합니다.
5. **Connector** 페이지에서 다음 그림과 같이 **Database** 윈도우에서 보여지는 더미 바디와 관련된 마커의 리스트를 이용하여 **Base Marker Body** 와 **Ref Frame** 을 **C1_DummyBody** 로 변경하고 **Name** 을 **Marker2** 로 변경합니다.



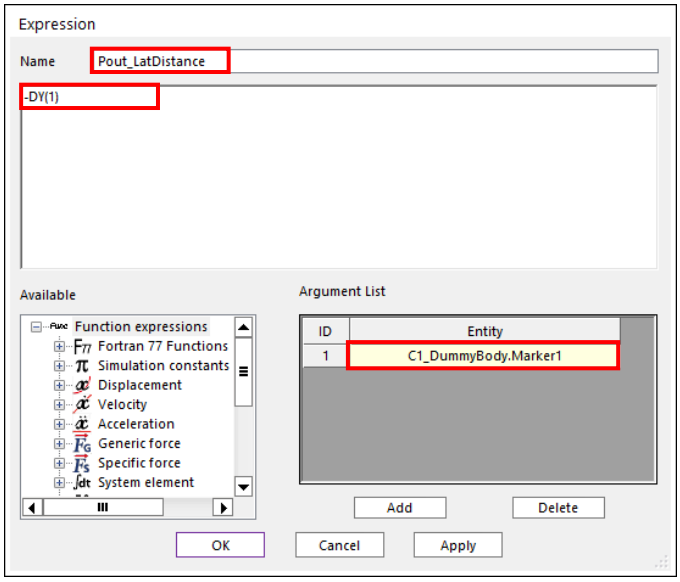
6. 작업이 원활 하도록 하기 위해 시뮬레이션을 실행합니다.

   5 초대의 시뮬레이션은 앞 쪽 자동차가 차선을 변경하게 되는 예외 상황의 이전 상황으로 보여야 합니다.

이제, CoLink 제어시스템을 수정하기 위해 Plant Output 을 추가적으로 정의해보겠습니다.

Plant Output 정의하기**:**

1. **Plant Output List** 다이얼로그 윈도우를 열기 위해 기존의 **Plant Output** 을 더블 클릭합니다.
2. 새로운 **Output** 을 생성하기 위해 **Add** 를 클릭합니다.
3. **Expression** 다이얼로그 윈도우에서 **Add** 버튼을 클릭하여 **Argument List** 에 Entity 를 추가하고 노란색 박스에 **C1_DummyBody.Marker1** 를 입력하거나 **Database** 윈도우에서 **C1_DummyBody.Marker1** 를 노란색 박스로 드래그합니다.
4. **Name** 을 **Pout_LatDistance** 로 변경하고 **–DY(1)**을 입력합니다. 이 수식에서 1 은 **Argument List** 의 목록을 기준으로 합니다.
5. **OK** 를 클릭합니다.



6. **Plant Output List** 다이얼로그 윈도우에서, **Add** 를 다시 클릭하여 **Plant Output** 을 하나 더 입력합니다.

7. Name 을 **Pout_AbsVel** 로 입력하고 Argument List 에 **Car_Body.Marker5** 를 입력한 후, 수식 **VX(1)**를 입력합니다.

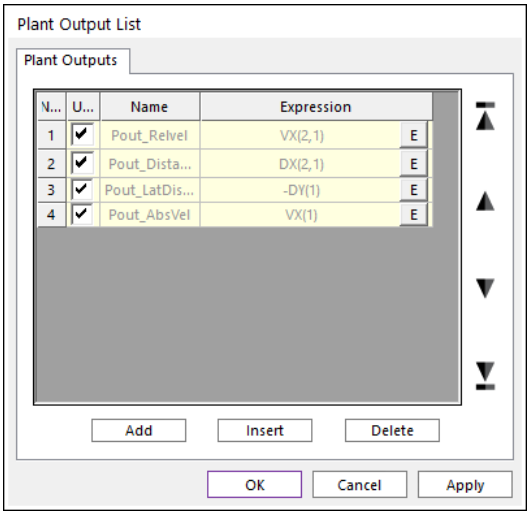
   Car_Body 의 Marker5 는 병진 조인트 Tra_Car_Ground 와 연관된 마커입니다.

Tip: 잘못 생성한 조인트를 지우거나 다시 생성하는 과정을 실행하려면 튜토리얼에서 언급한 번호가 아닌 다른 번호의 마커를 입력해야 될 수도 있으며, 마커의 번호가 이 튜토리얼에서 보여지는 것과 다를 수 있습니다.
Database 윈도우에서 **Tra_Car_Ground** 의 목록을 펼쳐서 정확한 마커를 찾은 후, **Car_Body** 와 연관된 마커를 찾아야 합니다.

---

8. 변경 사항을 적용하기 위해 **OK** 버튼을 클릭하고 다이얼로그 윈도우를 닫습니다.

9. **Plant Output List** 가 오른쪽 그림처럼 보이면 **OK** 버튼을 다시 누릅니다.



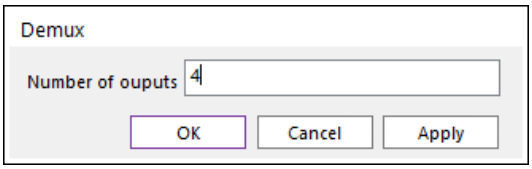
CoLink 모델의 수정

이제, 조정 가능한 크루즈 제어 시스템을 확장하기 위해 CoLink 모델을 수정해보겠습니다. 이것은 앞 쪽 자동차의 측면 모션을 기초로 하며, 앞 쪽 자동차가 도로 밖으로 움직일 때 원하는 속도를 유지하기 위해서 추가한 피드백 제어 시스템을 전환시키게 됩니다.

CoLink 모델의 수정**:**

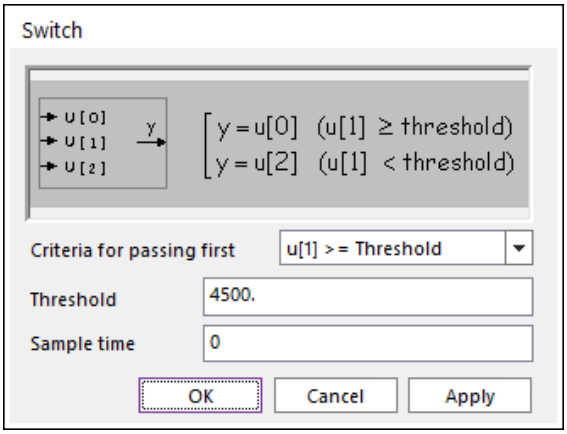
1. 이전에 모델을 생성했었던 CoLink 모델을 엽니다.

2. **File** 메뉴에서 **Save As** 를 클릭하여 새로운 파일로 CoLink 모델을 저장합니다.
3. **Demux** 블록을 더블 클릭하여 **Number of outputs** 을 **4** 로 지정합니다.



4. **Math** 탭에서 **Nonlinear** 그룹의 **Switch** 블록을 클릭한 다음, 블록 다이어그램을 클릭합니다.
5. **Switch** 블록을 더블 클릭하고 **Threshold** 값을 **4500** 으로 변경합니다

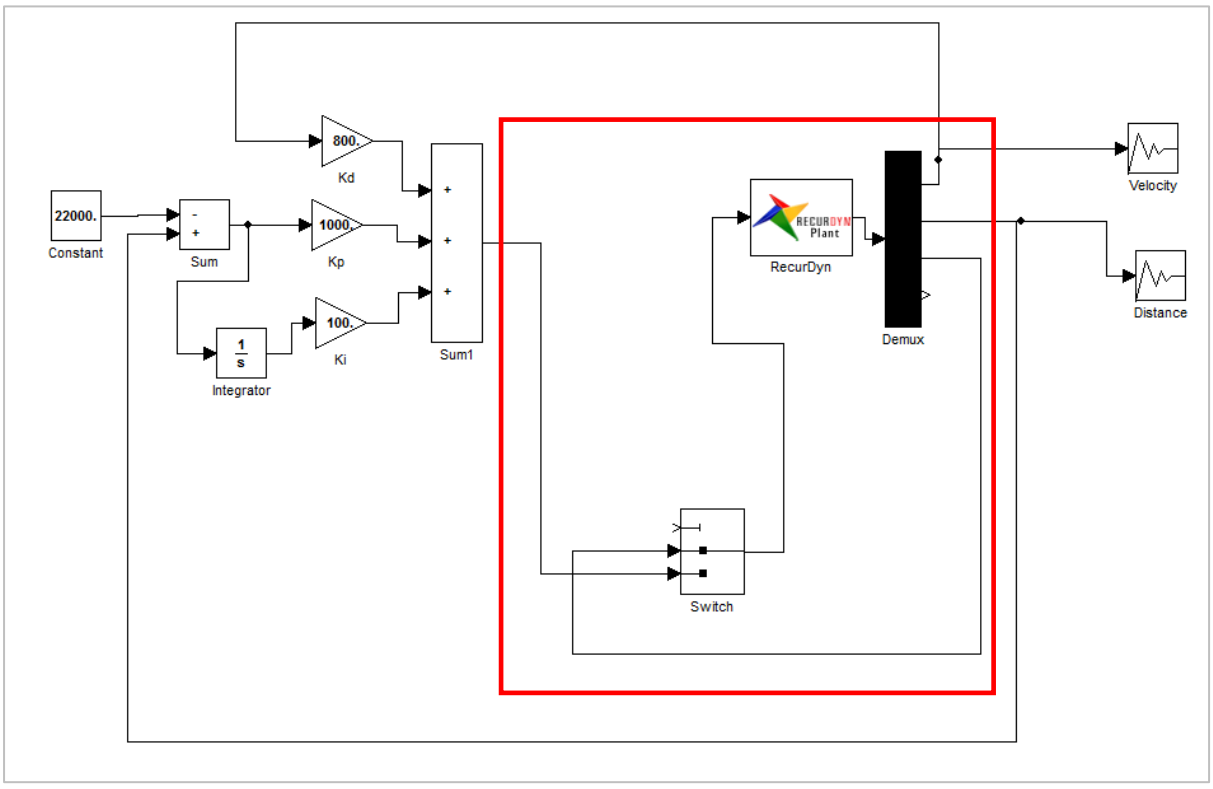
6. **Threshold** 를 **4500** 으로 설정한 것은 두 번째 Input 인 u[1] 이 4.5 m 보다 크거나 같다는 것을 의미하며, 첫 번째 Input 인 u[0]은 y 블록의 Output 이 지나갈 것을 의미합니다.



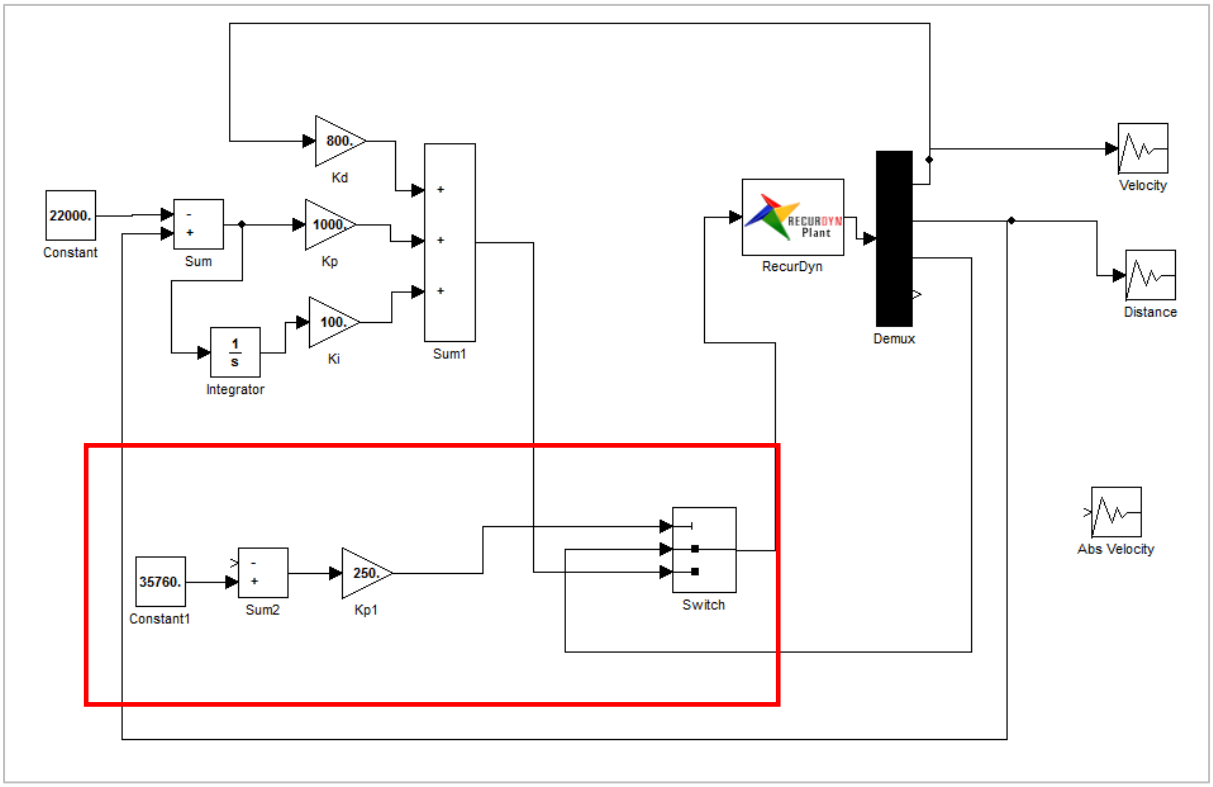
여기서, u[1]은 시뮬레이션에서 두 자동차 사이의 측면 거리로 설정된 것이며, u[2]는 두 자동차가 같은 차선에 있을 때의 기존 시스템에 대한 제어 신호입니다. 그리고, u[0]은 원하는 주행 속도를 설정한 것으로 새로운 제어 시스템에 대한 제어 신호입니다.

7. 아래 그림을 참고하여 다음과 같이 작업을 합니다.

   - **Sum1** 블록과 **RecurDyn** 블록 사이에 연결을 지우고, **Sum1** 블록과 **Switch** 블록의 맨 아래 포트를 연결합니다.
   - **RecurDyn** 블록에 **Switch** 블록의 **Output** 을 연결하고 **Demux** 블록의 세 번째 포트와 **Switch** 블록의 두 번째 포트를 연결합니다.



8. 이제, 아래의 그림에 표시한 것처럼 새로운 **Constant/Sum** 블록을 생성합니다.



a. **Sum** 블록을 클릭하고 Ctrl +C 키를 눌러서 그 블록을 복사한 후, Ctrl +V 키를 눌러 복사한 것을 붙여 넣습니다. 그리고 나서 그 복사한 블록을 **Switch** 블록 왼쪽에 있는 블록 다이어그램의 하단 쪽으로 드래그합니다.

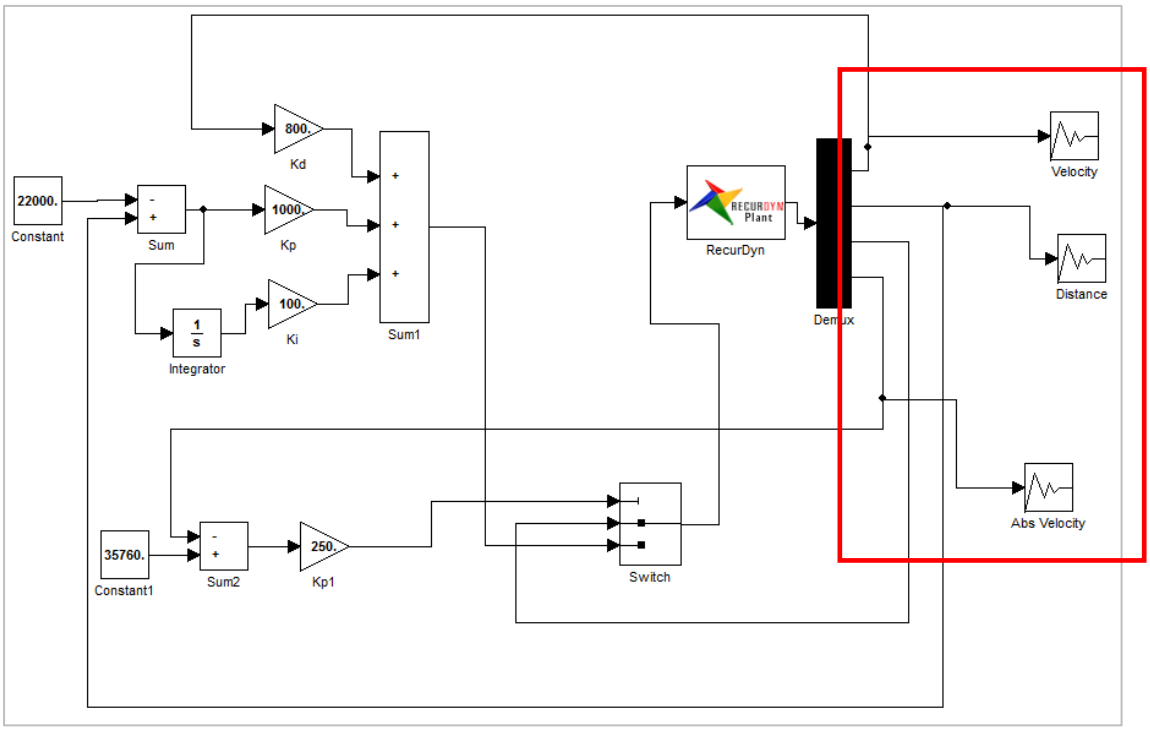
b. 위에서 보여지는 것처럼 새로운 **Sum2** 블록의 신호를 -+로 변경합니다.

c. 새로운 Sum 블록인 **Sum2** 에서 **Demux** 블록에서부터 – 터미널까지 4 개의 Output 포트를 연결합니다.

d. 유사한 방법으로, **Constant** 블록과 **Kp** 블록을 복사하여 **Sum2** 블록의 왼쪽과 오른쪽에 놓은 후 다음의 작업을 합니다.

- **35760** (mm/s, or 80 mph)로 Constant 의 값을 변경합니다.
- **Constant1** 블록을 **Sum2** 의 + 포트와 연결합니다.
- **Sum2** 의 Output 과 **Kp1** 의 Input 을 연결합니다.
- **Kp1** 의 **Gain** 값을 **250** 으로 변경하고 **Kp1** 의 Output 과 Switch 블록의 첫 번째 Input 포트를 연결합니다.

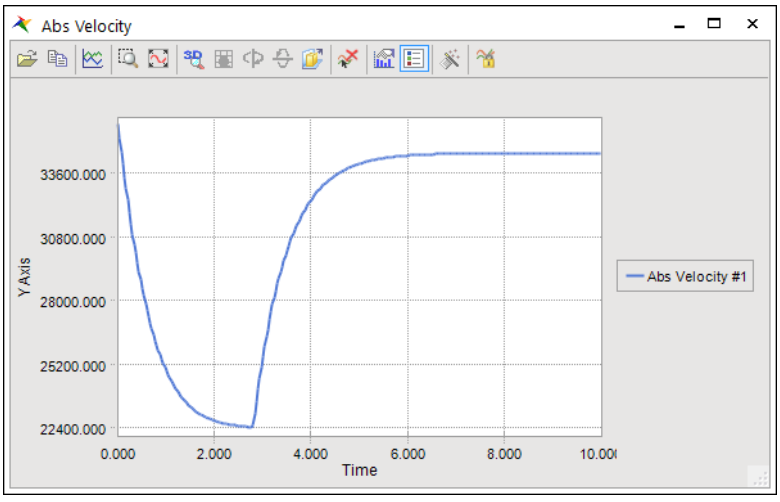
9. 아래의 그림에 표시된 것처럼, **Demux** 블록의 4 개의 Output 에 대하여 **Scope** 를 추가하고 이 Scope 를 통해 파란색 자동차의 실제 주행 속도를 확인할 것이므로, 그 이름을 **AbsVelocity** 로 지정합니다.



10. 작업을 원활하게 하기 위해서 시뮬레이션을 **10** 초로 하여 실행합니다.

Tip: 시뮬레이션 시간을 변경하려면,
툴바의 **Animation Control** 옆 입력 칸에 **10** 을 입력하거나 **Simulation →Parameter** 를 클릭하여 **End Time** 을 **10** 으로 설정합니다.

두 번째 자동차의 속도는 다음 Plot 과 같이 보여집니다.

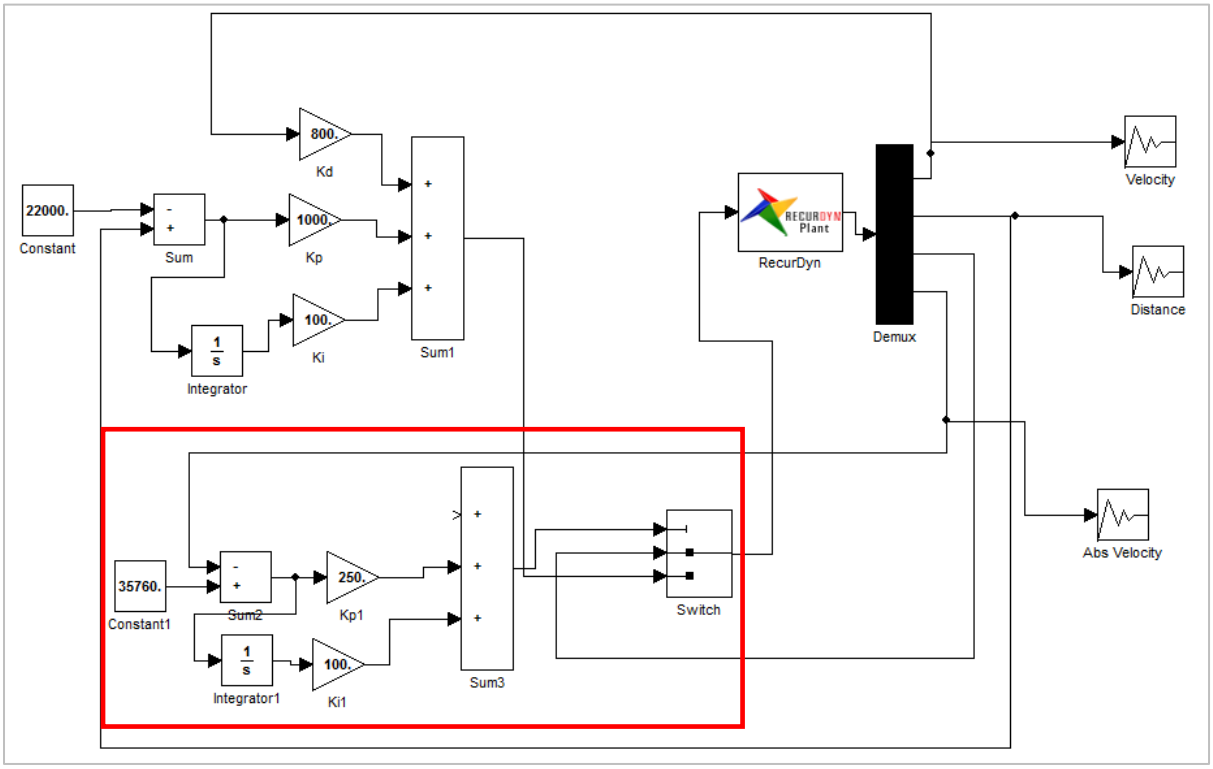


여기서, 주행 속도는 시뮬레이션 초기 값에 도달하지 않고 있는데 이 부분을 올바르게 수정하기 위해, 제어 시스템에 적분 제어를 추가해보겠습니다. 또한, 시뮬레이션의 3 초대에서 나타나는 빠른 가속도를 제어하기 위해 미분 제어와 속도 파라미터를 추가해보겠습니다.

적분 제어와 미분 제어 추가하기**:**

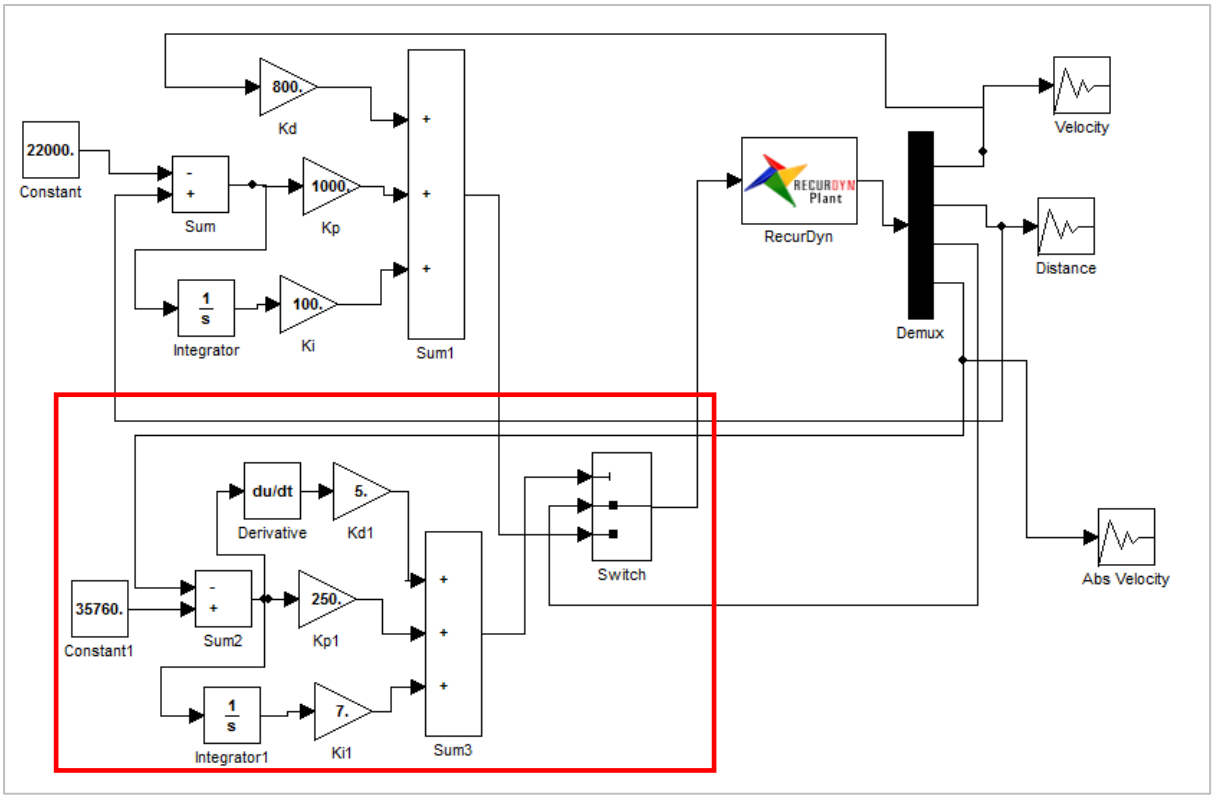
1. **Kp1** 블록과 **Switch** 블록 사이의 연결 블록을 지웁니다.
2. **Sum1** 블록을 복사하여 **Kp1** 블록의 오른 편에 붙여 넣습니다.
3. 새로 만든 **Sum3** 블록의 신호를 아래의 그림처럼 +++로 변경합니다.
4. **Kp1** 을 새로 만든 **Sum3** 의 두 번째 포트와 연결합니다.
5. **Sum3** 의 Output 을 **Switch** 블록의 첫 번째 포트에 연결합니다.
6. 기존의 제어 시스템에서 **Integrator** 와 **Ki** 블록을 복사하여 확장된 부분의 동일한 위치에 붙여 넣습니다. 이전에 실행하였던 방법과 동일한 방법으로 그 블록들을 연결합니다.

   이 과정이 완료되면 모델은 다음과 같이 보여집니다.

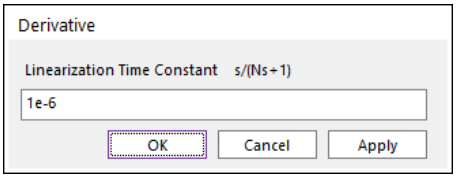


7. Continuous and Discrete 탭에서 Continuous 그룹의 **Derivative** 블록을 클릭한 후, **Sum2** 블록 위에 놓습니다.
8. **Kd** 블록을 복사하여 **Kp1** 블록 위에 붙여 넣습니다.
9. 다음의 것들을 연결합니다.
   - Sum2 와 Kp1 사이 실행 신호와 Derivative 블록
   - Derivative 블록과 Kd1 블록
   - **Kd1** 블록과 **Sum3** 블록의 첫 번째 Input 포트

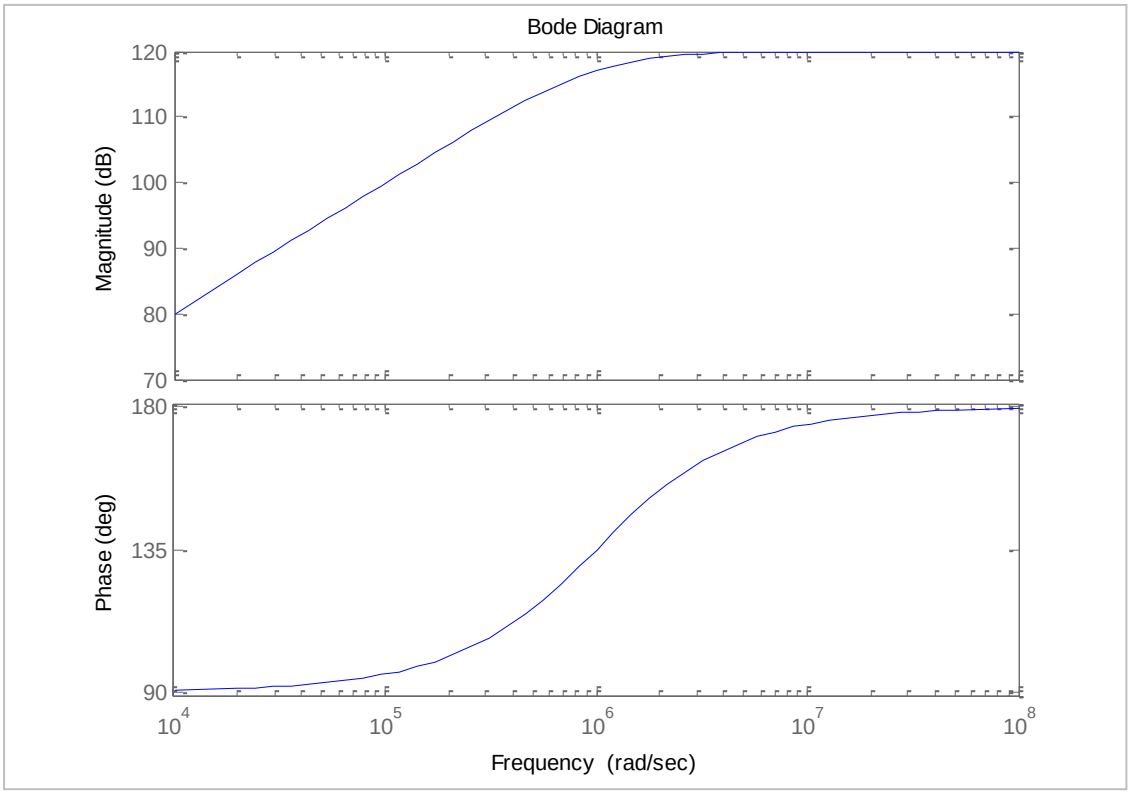
10. **Kd1** 과 **Ki1** 의 **Gain** 값을 **5** 와 **7** 로 각각 변경합니다. 이제 모델은 다음과 같이 보여집니다



11. **Derivative** 블록을 더블 클릭하여 **Linearization Time Constant** 를 **1e-6** 로 변경합니다.



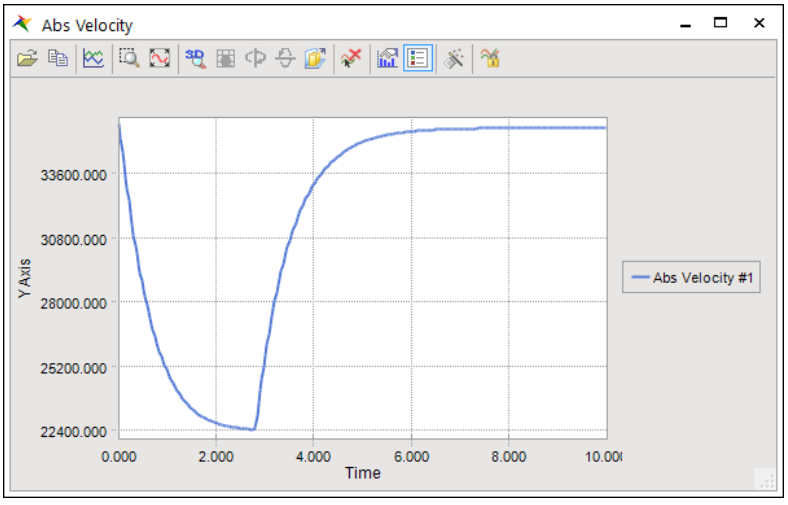
Derivative 다이얼로그 윈도우에서 보여지는 것처럼, 이에 대한 전달 함수는 현재 s/(-1e-6s + 1)입니다. 완전 미분은 단순히 s 또는 s/1 의 전달 함수를 가지는데, 이것은 고주파에서는 Gain 이 무한함에도 불구하고, 물리적으로 실현이 가능하지 않습니다. 그 대신에, 아래의 Bode 선도에서 보여지는 것처럼 1e6 rad/s 보다 작은 주파일 경우 블록이 미분기와 비슷하므로 Linearization Time Constant 을 선택합니다.



12. 시스템이 원활하게 작업되고 마지막 속도 값을 원하는 값이 되도록 하기 위해 시뮬레이션을 다시 실행합니다. 자동차의 속도는 아래 그림과 같이 보여집니다.

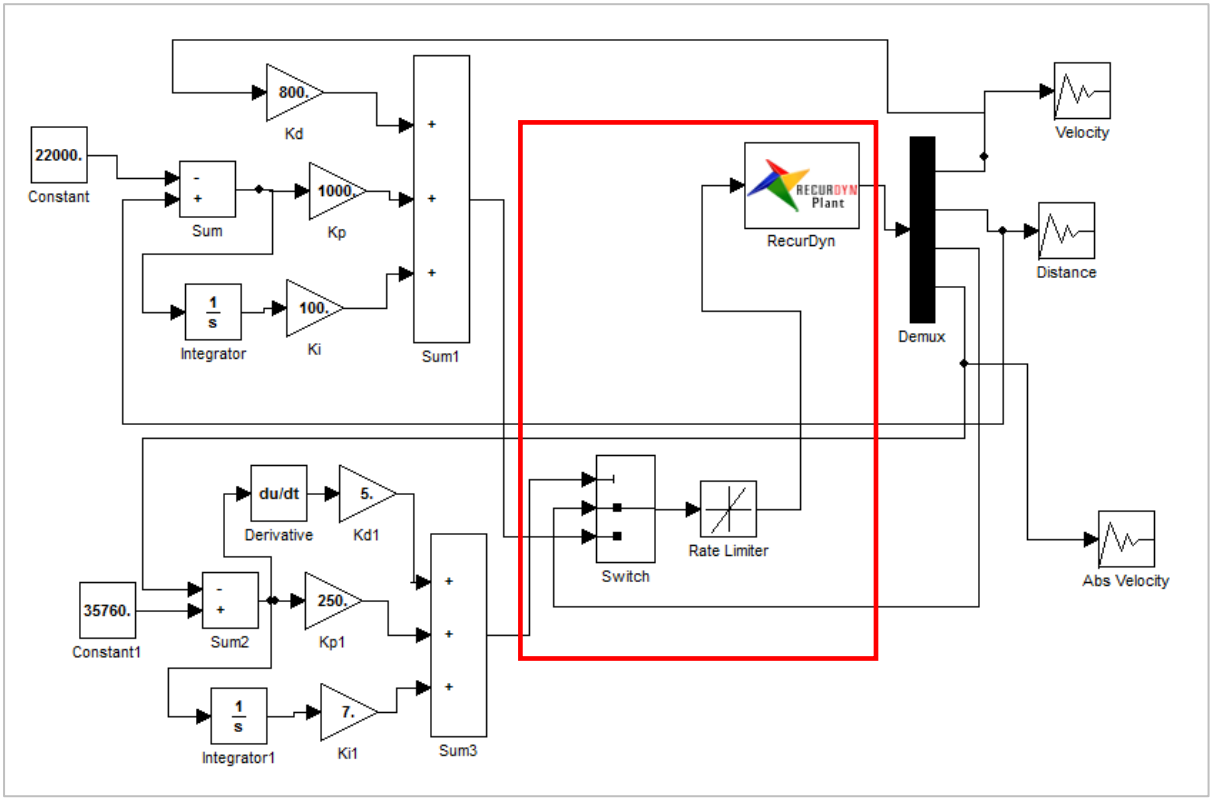
앞 쪽의 자동차가 주행하자마자 뒤 쪽의 자동차의 움직임이 변화가 없는 속도였다가 빠르게 증가된 속도로 변경되었습니다. 이것은 자동차의 가스 페달을 밟는 것처럼 느껴져서 탑승자에게는 불편하게 느껴집니다.

또한, 자동차는 어떠한 대상이 도로에 있을 경우 빠르게 속도를 감소시킬 수 있기 때문에, 여기에서, 속도의 빠른 감소는 문제가 되지 않습니다. 그럼 이제, 가속도가 너무 높지 않도록 하기 위해서 제어 시스템에 Rate Limiter 를 추가해 보겠습니다.

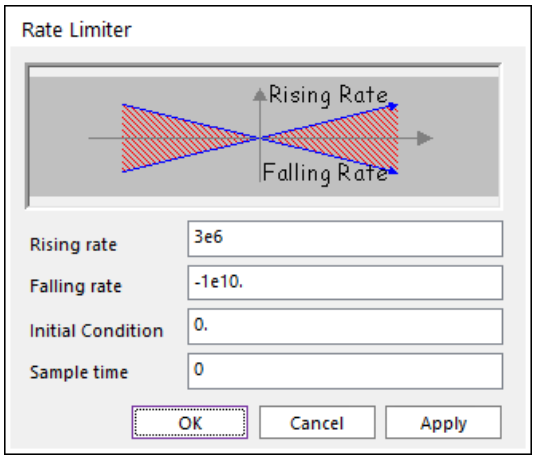


Rate Limiter 추가하기**:**

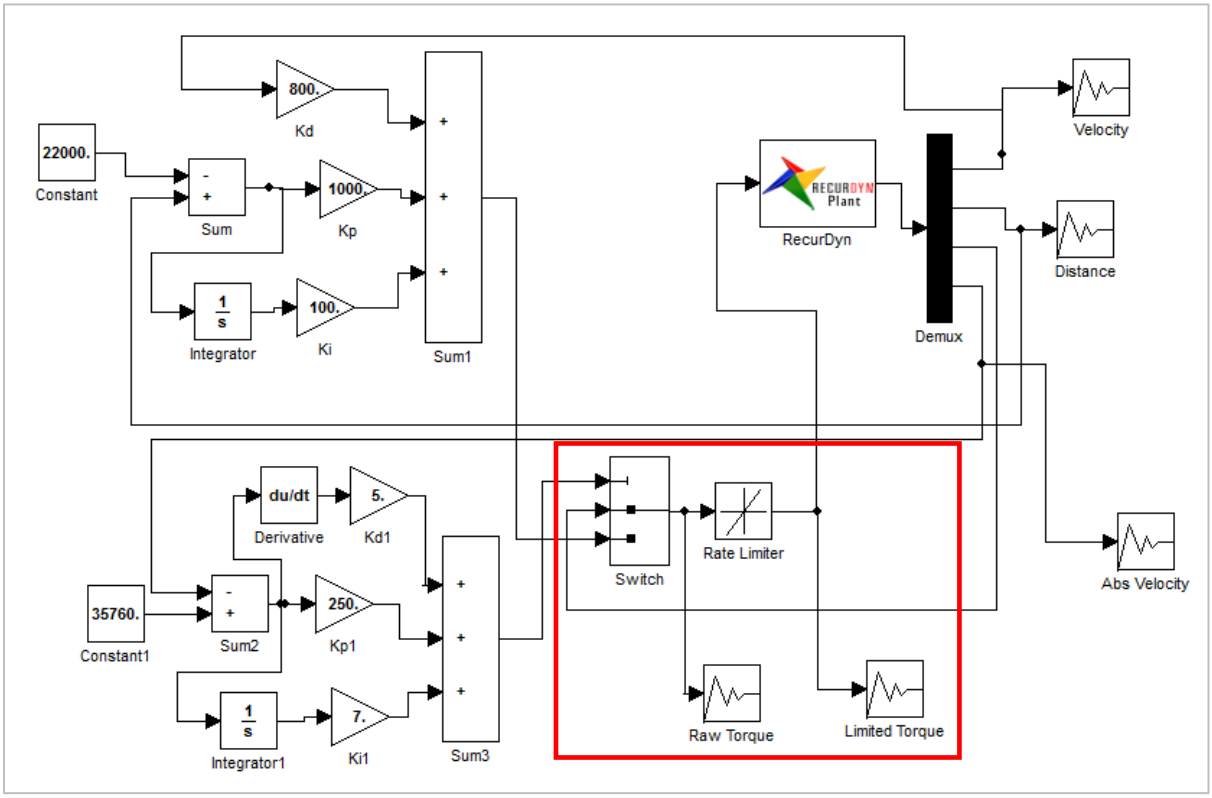
1. **Switch** 와 **RecurDyn** 블록 사이에 연결 블록을 지웁니다.
2. **Math** 탭에서 **Nonlinear** 그룹의 **Rate Limiter** 블록을 클릭한 후, **Switch** 블록의 오른 쪽에 놓습니다.
3. 아래의 그림처럼 **Switch** 와 **Limiter** 블록을 연결하고 **Rate Limiter** 블록과 **RecurDyn** 블록을 연결합니다.



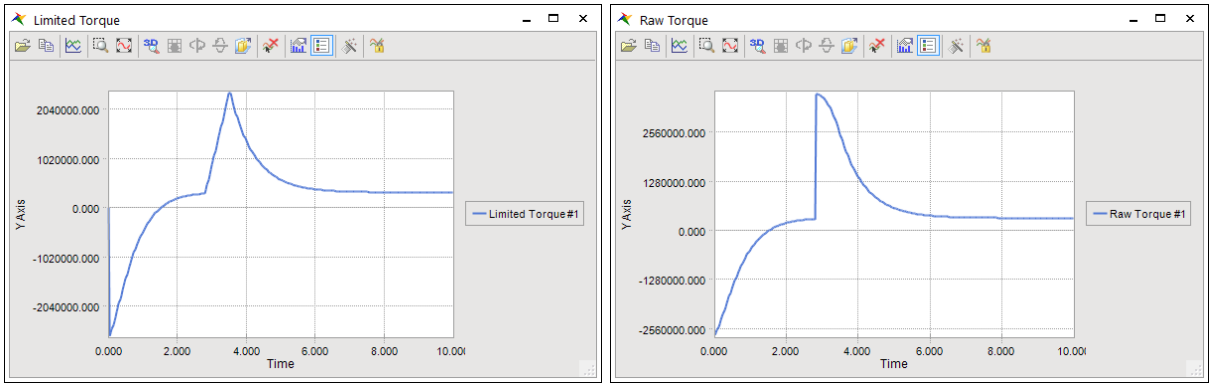
4. **Rate Limiter** 를 더블 클릭하여 **Rising rate** 를 **3e6** 으로 변경하고 **Falling rate** 를 **-1e10** 로 변경합니다. 나머지 항목들은 초기 값으로 그대로 둡니다.



5. **Rate Limiter** 에 대한 효과를 살펴보기 위해 모델에 대해서 두 개의 **Scope** 를 추가합니다. 두 개의 Scope 중 하나는 **Rate Limiter** 블록 앞에, 나머지 하나는 **Rate Limiter** 블록 뒤에 연결합니다. Scope 의 이름을 **RawTorque** 로, Scope1 의 이름을 **LimitedTorque** 로 변경합니다.



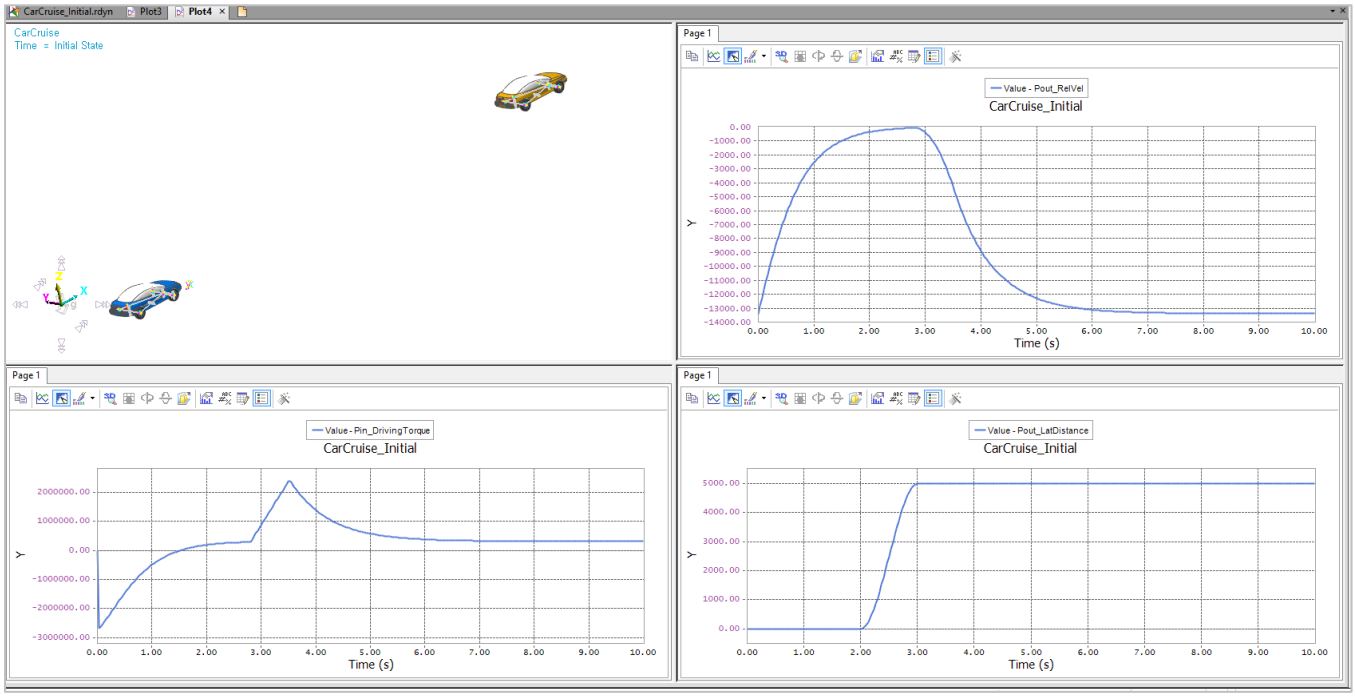
6. 모델에 대해 다시 시뮬레이션을 실행한 후, 새로 생성한 두 개의 **Scope** 를 더블 클릭하여 그



결과를 확인합니다. 그 결과는 다음과 같이 보여집니다.

7. RecurDyn 모델로 다시 돌아가서 **Analysis** 탭의 **Animation Control** 그룹에서 **Play** 버튼을 누른 후, 시뮬레이션을 관찰합니다.

모델이 원활하게 작업되었다면, Plot 윈도우를 열어서 5 장의 끝 부분에서 했던 과정대로 Plot 을 동일하게 그리되, 오른쪽 하단 Pane 에는 자동차 사이의 거리가 아닌 측면 거리인 **Pout_LatDistance** 을 Plot 으로 그립니다. 그러면, 다음과 같이 Plot 이 보여집니다.



모델에 대해 더 작업해보고 싶다면, 아래의 과정을 추가적으로 작업해보기 바랍니다.

- 성능을 높이기 위해 제어 Gain 을 조정하는 과정
- Kd 와 Ki 를 0 으로 설정한 후 제약 없이 제어 컨트롤러를 조정하는 과정

Thanks for participating in this tutorial!