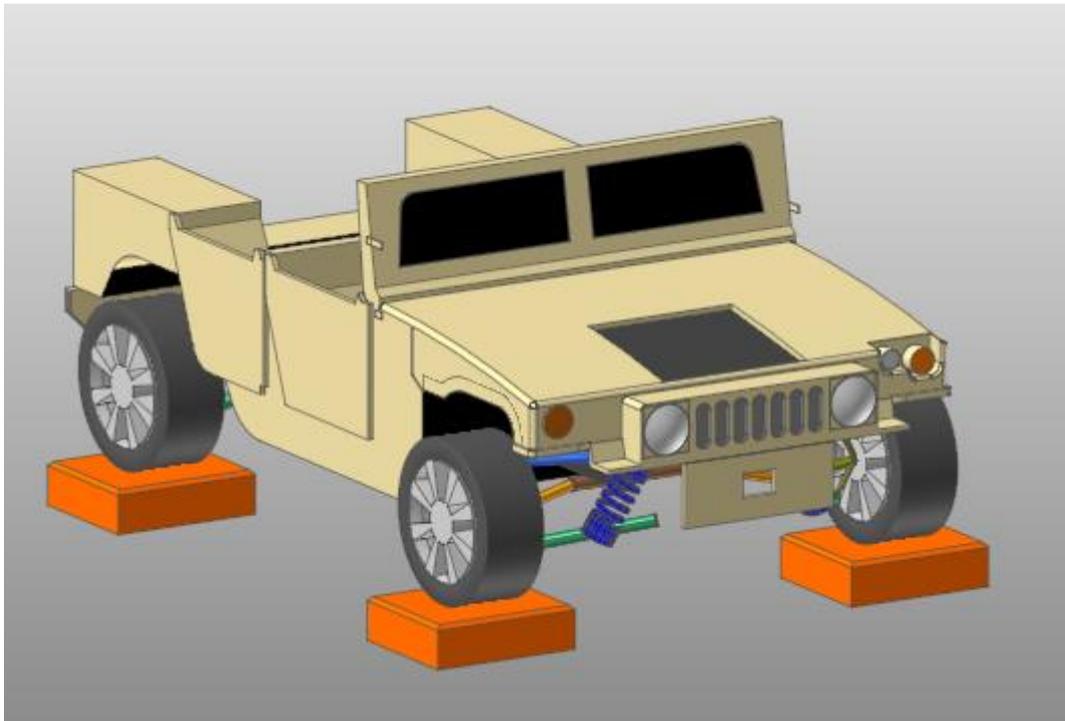




Automotive Road Testing (TSG)



Copyright © 2020 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/Professional, RecurDyn/ProcessNet, RecurDyn/Acoustics, RecurDyn/AutoDesign, RecurDyn/Bearing, RecurDyn/Belt, RecurDyn/Chain, RecurDyn/CoLink, RecurDyn/Control, RecurDyn/Crank, RecurDyn/Durability, RecurDyn/EHD, RecurDyn/Engine, RecurDyn/eTemplate, RecurDyn/FFlex, RecurDyn/Gear, RecurDyn/DriveTrain, RecurDyn/HAT, RecurDyn/Linear, RecurDyn/Mesher, RecurDyn/MTT2D, RecurDyn/MTT3D, RecurDyn/Particleworks I/F, RecurDyn/Piston, RecurDyn/R2R2D, RecurDyn/RFlex, RecurDyn/RFlexGen, RecurDyn/SPI, RecurDyn/Spring, RecurDyn/TimingChain, RecurDyn/Tire, RecurDyn/Track_HM, RecurDyn/Track_LM, RecurDyn/TSG, RecurDyn/Valve are trademarks of FunctionBay, Inc.

Edition Note

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

목차

개요	4
목적	4
필요 요건	6
과정	6
예상 소요 시간	6
초기 모델 불러오기	7
목적	7
예상 소요 시간	7
RecurDyn 모델 불러오기	8
신호 정의하기	10
목적	10
예상 소요 시간	10
Actuator 정의하기	11
Sensor 정의하기	13
Target Signal 정의하기	14
FRF 수행하기	18
목적	18
예상 소요 시간	18
FRF 수행하기	19
FRF 결과보기	21
Iteration 수행하기	23
목적	23
예상 소요 시간	23
Iteration 수행하기	24
Iteration 결과 보기	25

Chapter

1

개요

목적

일반적으로, 자동차 부품 혹은 시스템 내구성 테스트를 위해 실제 차량 주행시험을 통해 센서로부터 가속도, 속도, 위치 등과 같은 계측데이터를 획득하게 됩니다. 이때, 계측데이터는 차량의 외부 가진으로 인해 특정 위치에서 도출되는 출력신호(Output Signal)이므로, 실제 차량 시험장치(Test-rig) 또는 리커다인을 통한 모의 차량 시험장치(Virtual Test-rig)에서 가진기(Actuator)의 입력신호(Input Signal)로 적용하기 어렵습니다.

또한, 리커다인으로 모델링 된 시스템(Virtual System)은 실제시스템(Physical System)의 모든 비선형 요소를 반영하기 어렵기 때문에 두 시스템을 동일하다고 가정하기 어렵습니다. 그러나 획득된 계측데이터가 리커다인 모델링에서 도출될 수 있도록 가진기의 입력신호를 생성할 수 있다면, 계측데이터 활용성 및 리커다인 모델링의 신뢰성에 대한 문제점을 해결할 수 있습니다. 그러므로, TSG Toolkit 은 리커다인 모델링에 포함된 가진기의 입력신호를 계측데이터를 활용하여 실제시스템의 가진 상황과 최대한 유사하게 모사할 수 있도록 만들어 줍니다.

본 튜토리얼에서는 TSG Toolkit 을 이용하여 차량 주행시험 후 얻은 계측데이터를 바탕으로 가진기의 입력신호를 생성합니다.

- 계측데이터 활용 및 시뮬레이션 결과 확인을 위한 센서(Sensor) 설정
- 시뮬레이션을 통해 센서 결과가 도출될 수 있도록 가진기(Actuator)를 설정
- 반복 시뮬레이션(Iteration)을 통해 센서에서 계측데이터가 도출될 수 있도록 가진기의 입력신호 생성
 - 센서에서 도출되는 출력신호: Response Signal
 - 센서에서 도출되어야 하는 출력신호(계측데이터 활용): Target Signal
 - 가진기의 입력신호: Drive Signal

- 센서에서 도출된 결과(Response Signal)와 계측데이터(Target Signal)가 유사하게 도출되었는지 결과(Result) 확인

필요 요건

본 튜토리얼은 RecurDyn 에서 제공되고 있는 Basic Tutorial 을 숙지한 사용자를 위한 것입니다. 따라서 본 튜토리얼을 사용하기 위해서는 앞서 언급된 교재를 선행해야 본 교재의 이해를 높일 수 있습니다.

과정

본 튜토리얼은 다음의 과정들로 구성되어 있으며, 소요되는 시간은 다음의 표와 같습니다.

Procedures	Time (minutes)
초기 모델 불러오기	10
신호 정의하기	10
FRF 수행하기	10
Iteration 수행하기	10
Total	40



예상 소요 시간

40 분

Chapter
2

초기 모델 불러오기

목적

초기 모델을 열어 모델을 관찰해봅시다.



예상 소요 시간

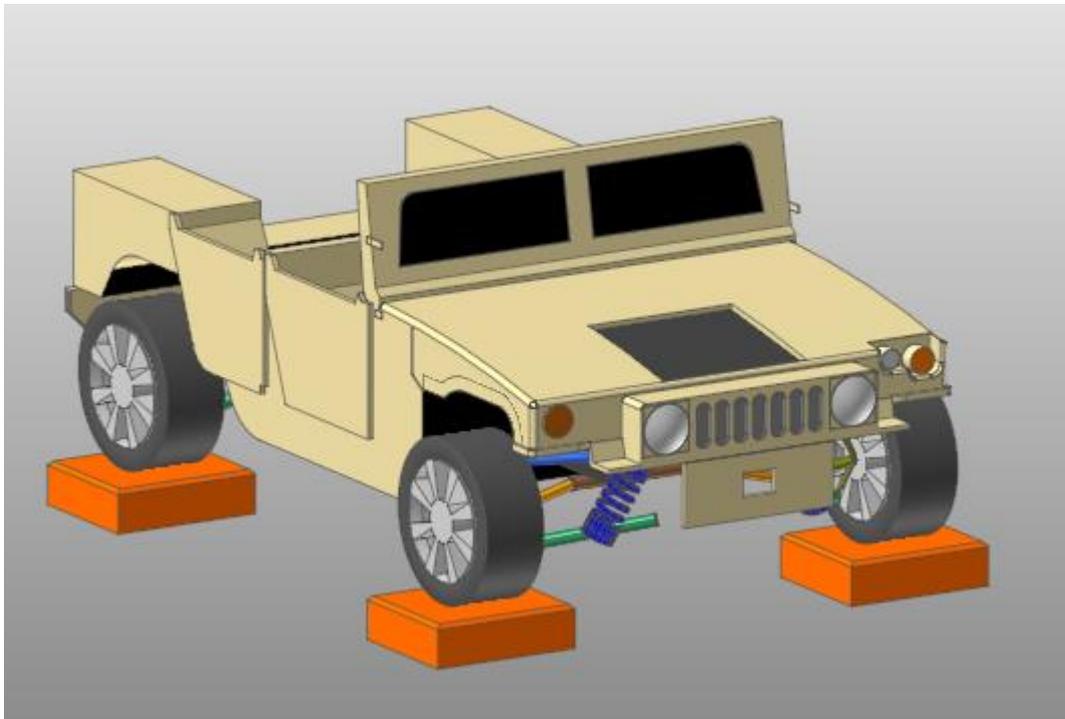
10 분

RecurDyn 모델 불러오기

RecurDyn 실행 및 초기 모델 불러오기

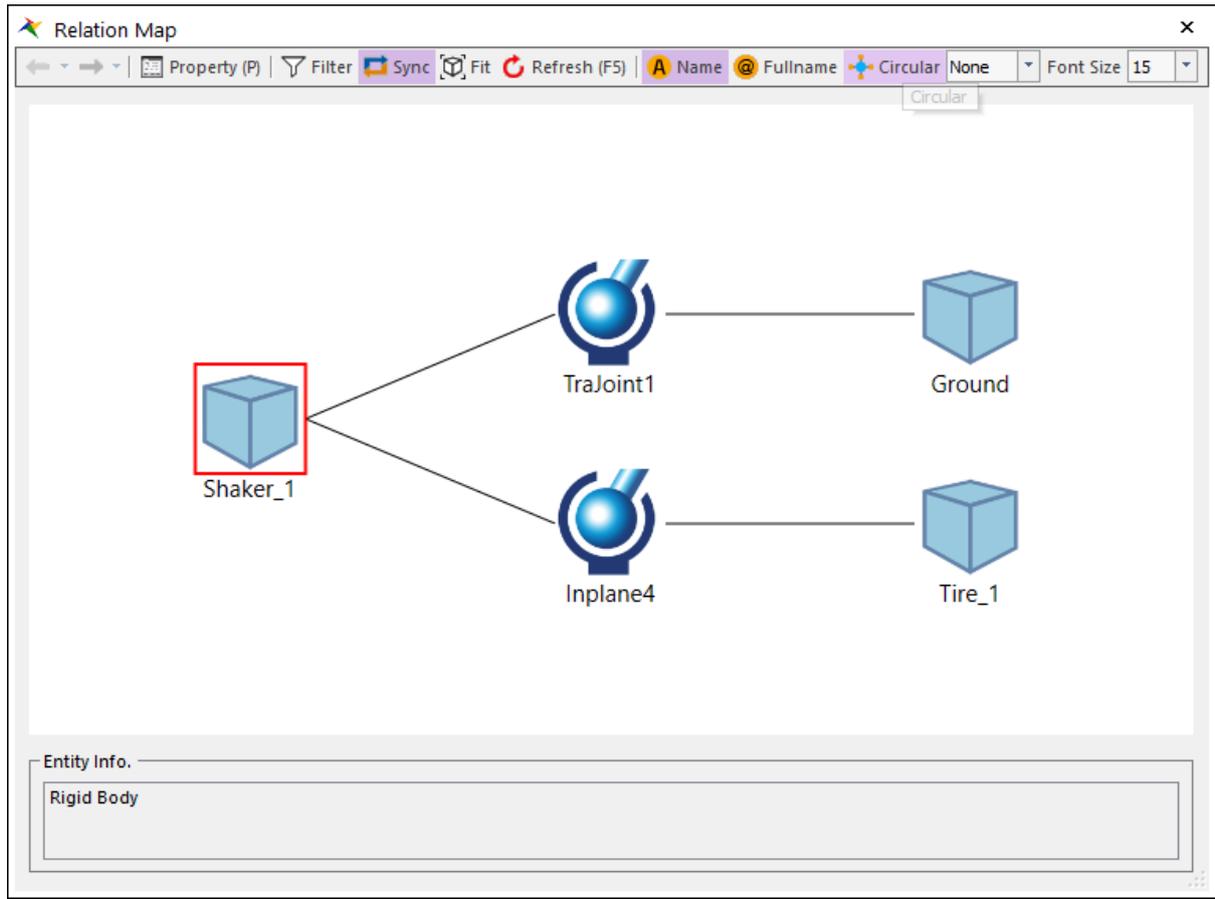


1. 바탕화면에서 **RecurDyn** 아이콘을 더블클릭하면, RecurDyn 이 실행되면서 **Start RecurDyn** 다이얼로그 박스가 나타납니다.
2. **Start RecurDyn** 다이얼로그 박스가 나타나면 닫아줍니다.
3. **File** 메뉴에서 **Open** 을 클릭합니다.
4. 튜토리얼 경로에서 **TSG_Tutorial_Car_Start.rdyn** 을 선택합니다. (파일 경로: <Install Dir> \Help\Tutorial\TSG\AutomotiveRoadTesting).
5. **Open** 을 클릭합니다. 아래의 그림처럼 모델이 보여집니다.



모델의 구성은 다음과 같습니다.

차량의 바퀴마다 shaker 가 하나씩 있습니다. Shaker 는 위/아래로만 움직일 수 있도록 Ground 와 Translational Joint 로 되어 있으며, Tire 와는 한 면으로만 움직일 수 있도록 Inplane Joint 가 정의되어 있습니다. 이는 **RecurDyn** 에서 제공하는 **Relation Map** 을 통해서 쉽게 확인이 가능합니다.



모델 저장하기

1. **File** 메뉴에서, **Save As** 를 클릭합니다.

(튜토리얼 경로에서는 직접 시뮬레이션 실행이 불가하므로 다른 경로에 본 모델을 다시 저장해야 합니다.)

Chapter

3

신호 정의하기

목적

TSG 을 수행하기 위해서는 이를 통해 생성될 신호를 정의하고, 시뮬레이션 후 응답확인을 위해 Sensor 을 정의한 다음 측정 데이터를 가공합니다.



예상 소요 시간

10 분

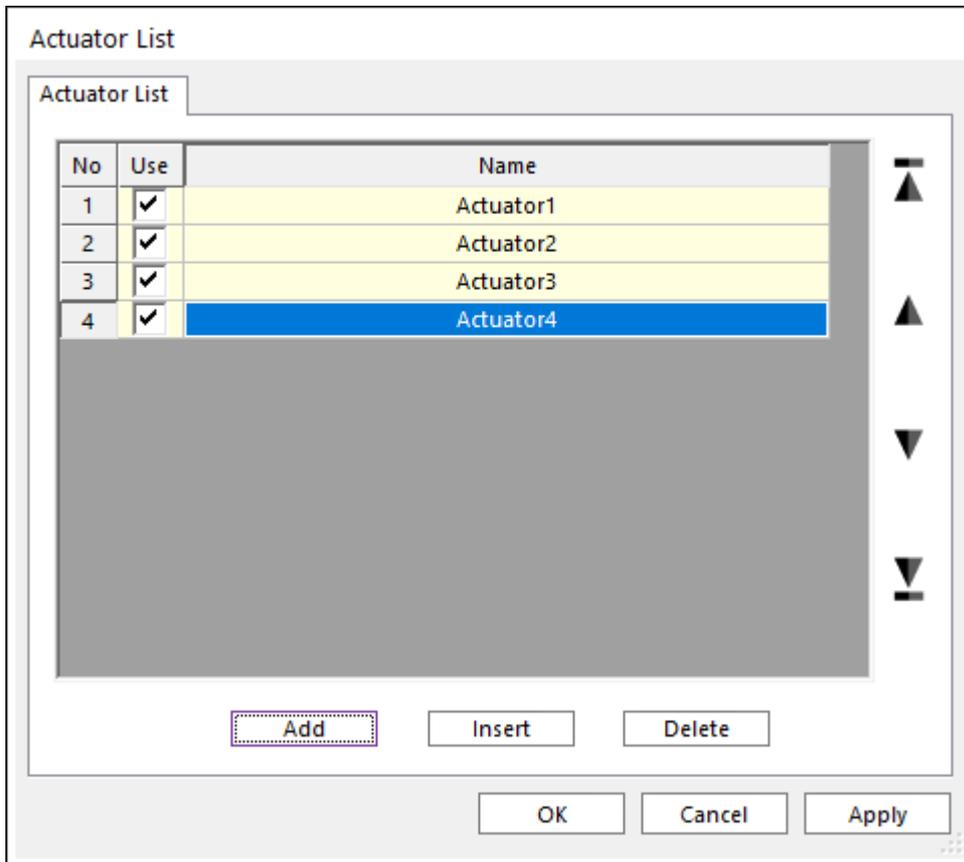
Actuator 정의하기

생성될 신호의 개수대로 Actuator 을 정의합니다. 정의된 Actuator 는 실질적으로 Joint 엔티티의 Motion 혹은 Force 엔티티의 성분을 Expression 으로 정의할 때 적용하게 됩니다. 본 튜토리얼에서는 각각 Tire 아래에 Shaker 4 개를 Translational Joint 로 정의한 후, Displacement Motion 을 적용하여 각각의 Tire 를 상하로 구동시킵니다.

Actuator 생성하기



1. TSG 탭의 **Signal** 그룹에서 **Actuator** 아이콘을 선택합니다.
2. 총 4 개의 **Actuator** 을 정의합니다.



Actuator 적용하기

1. **Subentity** 탭의 **Expression** 그룹에서 **Expression** 아이콘을 클릭합니다.
2. **Expression List** 대화상자에서 **Create** 을 클릭합니다.

3. **Expression** 대화상자에서 **Name** 을 **EX_Actuator1** 로 정의하고 **TACT(Actuator1)** 로 내용을 입력합니다.
4. **Expression** 대화상자를 **OK** 로 닫고 나머지 3 개의 Actuator 에 대해서도 **EX_Actuator2**,

Expression List

Expressions

No	Name	Expression	Value	Comment
1	Ex1	0	E	0
2	EX_Actuator1	TACT(Actuator1)	E	N/A
3	EX_Actuator2	TACT(Actuator2)	E	N/A
4	EX_Actuator3	TACT(Actuator3)	E	N/A
5	EX_Actuator4	TACT(Actuator4)	E	N/A

EX_Actuator3, EX_Actuator4 로 만듭니다.

5. **TraJoint1** 의 **Property** 대화상자를 열고 **Include Motion** 을 체크합니다.
6. **Motion** 을 클릭하여 **Motion** 대화상자를 엽니다.

Properties of TraJoint1 [Current Unit : N/kg/m/s/deg]

General Connector Joint

Type Translational

Motion

Include Motion Motion

7. **Type** 을 **Displacement(Time)**로 하고 **EL** 을 클릭합니다.
8. 앞서 만든 **EX_Actuator1** 을 **Expression List** 대화상자에서 가져옵니다.
9. **TraJoint2, TraJoint3, TraJoint4** 에 대해서도 **EX_Actuator2, EX_Actuator3, EX_Actuator4** 로 Motion 을 입력합니다.

Motion

Motion

Type Standard Motion

Displacement (time)

Expression

Name EX_Actuator1 EL

Expression

TACT(Actuator1)

OK Cancel Apply

Sensor 정의하기

시뮬레이션 수행 후 응답을 확인하기 위한 성능 지표로 Sensor에서 도출된 결과는 Target Signal과 비교하게 됩니다. 본 튜토리얼에서는 Chassis에 대한 가속도 값을 측정할 것입니다.

Sensor 생성하기



1. TSG 그룹의 **Signal** 탭에서 **Sensor** 아이콘을 클릭합니다.
2. **Expression** 대화상자를 열기 위해 **Add** 를 클릭합니다.
3. **Expression** 대화상자가 나타나면 아래와 같이 정의합니다.

Expression

Name:

Available

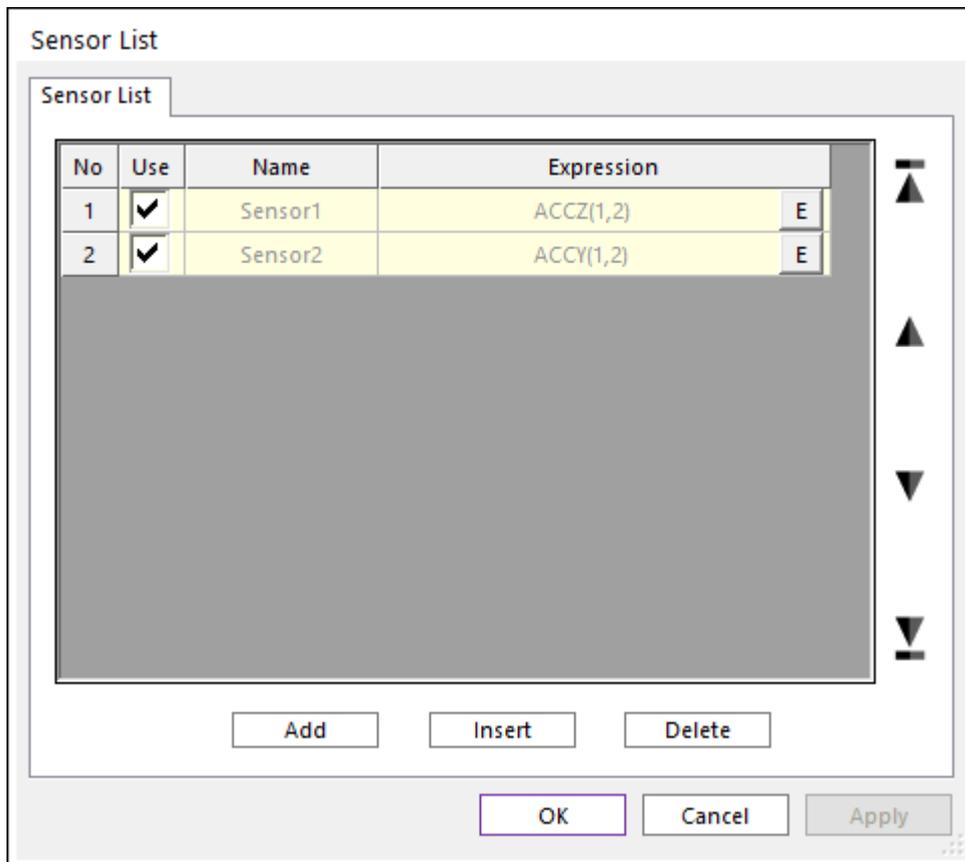
- [-] Func Function expressions
 - [+] F77 Fortran 77 Functions
 - [+] π Simulation constants
 - [+] a Displacement
 - [+] v Velocity
 - [+] a Acceleration
 - [+] F_G Generic force
 - [+] F_S Specific force
 - [+] ∫dt System element

Argument List

ID	Entity
1	CHASSIS.CM
2	Ground.InertiaMarker

4. **Expression** 대화상자에서 **OK** 를 클릭합니다.
5. **Sensor List** 대화상자에 추가되었음을 확인합니다.

6. 위와 동일하게 **ACCY(1,2)**에 대한 Sensor 를 하나 더 만들어 줍니다. 총 2 개의 Sensor Signal 이 생성되어 있어야 합니다.



Target Signal 정의하기

사용자가 정의해야 하는 입력 데이터로써, 시험 또는 시뮬레이션을 통해 획득한 시간에 대한 연속적인 데이터 모음으로 정의할 수 있습니다. 이는 TSG의 성능 평가의 지표가 됩니다.

선택한 Sensor 의 수만큼 **Target Signal** 정의가 필요하며 Target Signal 은 *.csv file 타입으로

0	0	0	0
0.001	-0.0004	0.001	0.000186
0.002	-0.00303	0.002	0.001564
0.003	-0.01027	0.003	0.009046
0.004	-0.02232	0.004	0.033633
0.005	-0.03801	0.005	0.080458
0.006	-0.05629	0.006	0.131874

Sensor1 의 Target

Sensor2 의 Target

시작에 대한 적용 데이터가 시간 1, 데이터 1, 시간 2, 데이터 2, ... 순으로 정의되어 있어야 합니다.

시험 또는 시뮬레이션을 통해 획득한 시간에 대한 연속적인 데이터는 저주파수 성분뿐만 아니라 고주파 성분이 다수 포함하고 있으며, 이는 TSG Toolkit 적용 과정에서 Noise 발생 및 Error 유발의 요인이 됩니다. 따라서, 초기 획득한 신호를 Low Pass Filter 를 사용하여 50Hz 혹은 100Hz 이하의 Signal 만 포함될 수 있도록 *.csv file 을 생성하기 이전에 조치하는 것을 권장합니다.

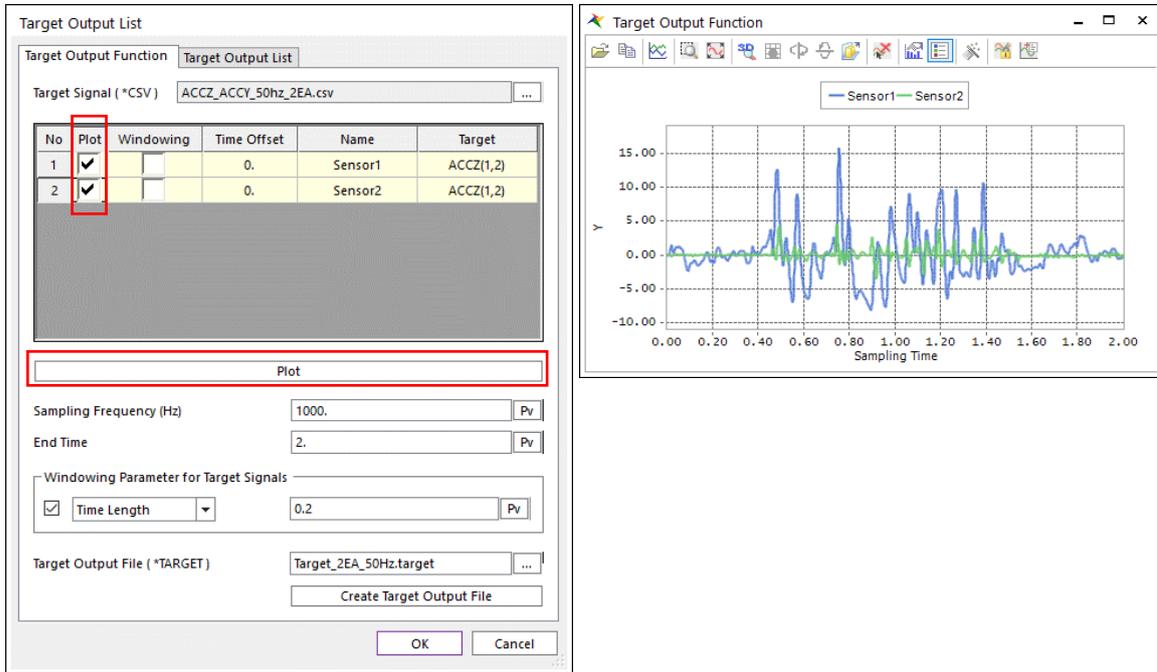
본 튜토리얼 에서는 2 개의 Sensor 가 정의되었으므로, csv file 은 4 개의 데이터가 시간 순서대로 기록되어 있어야 합니다.

Target Signal 생성하기



1. TSG 탭의 **Signal** 그룹에서 **Target** 아이콘을 클릭합니다.
2. **Target Output List** 대화상자에서 **Target Output Function** 탭으로 이동 후 다음과 같이 작업합니다.
 - **Target Signal** 영역의 "..." 버튼을 누릅니다.
 - 초기 모델이 존재하는 경로에 위치한 **ACCZ_ACCY_50hz_2EA.csv** 파일을 선택하도록 합니다.

- **Plot** 을 체크하여 **Plot** 을 클릭하면 아래와 같이 **Target Signal** 을 **Scope** 로 확인 가능합니다.



- **Sampling Frequency** 는 **1000** 을 입력합니다.
 - **End Time** 은 **2** 을 입력합니다. 입력한 csv file 에는 1 초당 1000 개의 data 을 가졌기 때문에 이와 일치하도록 End Time 을 입력하여야 합니다.
 - **Windowing Parameter for Target Signal** 로 **Time Length** 을 선택하여 **0.2** 을 입력합니다. 이는 Time Signal 을 Fourier Transform 을 통해 Frequency Signal 로 변환함에 있어 Error 을 최소화 하기 위해 처음과 끝신호를 강제로 Zero 로 만드는 기능을 수행합니다.
 - **Target Output File** 에 **Target_2EA_50Hz.target** 이라는 이름으로 모델과 같은 폴더에 생성되도록 정의합니다.
 - **Create Target Output File** 을 클릭합니다.
3. **Target Output List** 대화상자에서 **Target Output List** 탭으로 이동 후 생성된 Target Signal 을 확인합니다.

- **Plot** 을 체크하여 **Plot** 을 클릭하면 아래와 같이 가공된 **Target Signal** 을 **Scope** 로 확인 가능합니다.

The image shows two windows from a software interface. The left window is titled "Target Output List" and contains a table with the following data:

No	Plot	Name	Target
1	<input checked="" type="checkbox"/>	Sensor1	ACCZ(1,2)
2	<input checked="" type="checkbox"/>	Sensor2	ACCZ(1,2)

Below the table is a "Plot" checkbox, which is also checked and highlighted with a red box. At the bottom of the dialog are "OK" and "Cancel" buttons.

The right window is titled "Target Output" and displays a scope plot. The plot shows two signals, "Sensor1" (blue line) and "Sensor2" (green line), over a "Sampling Time" of 0.00 to 2.00. The y-axis ranges from -10.00 to 15.00. Both signals show high-frequency oscillations, with Sensor1 having a higher amplitude than Sensor2.

Chapter

4

FRF 수행하기

목적

FRF(Frequency Response Function)은 Time Signal 을 Frequency Signal 로 변환하는 신호처리 과정을 통해 시스템의 특성을 나타내는 전달함수($H(f)$)를 도출하게 됩니다.



예상 소요 시간

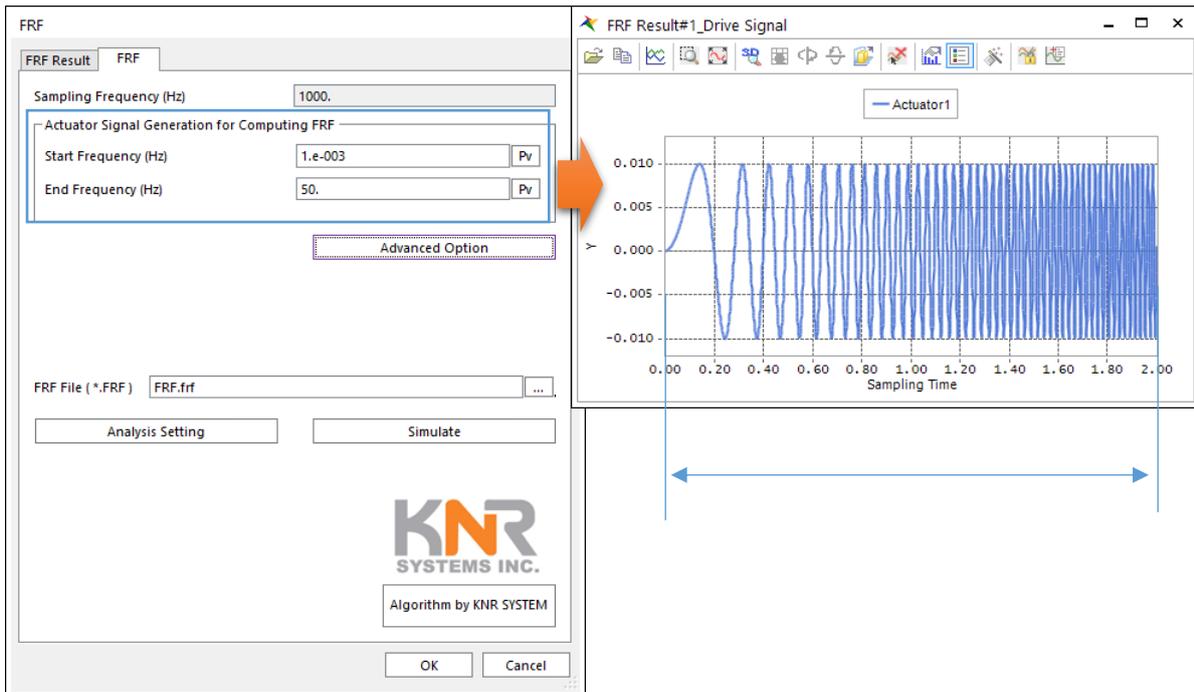
10 분

FRF 수행하기

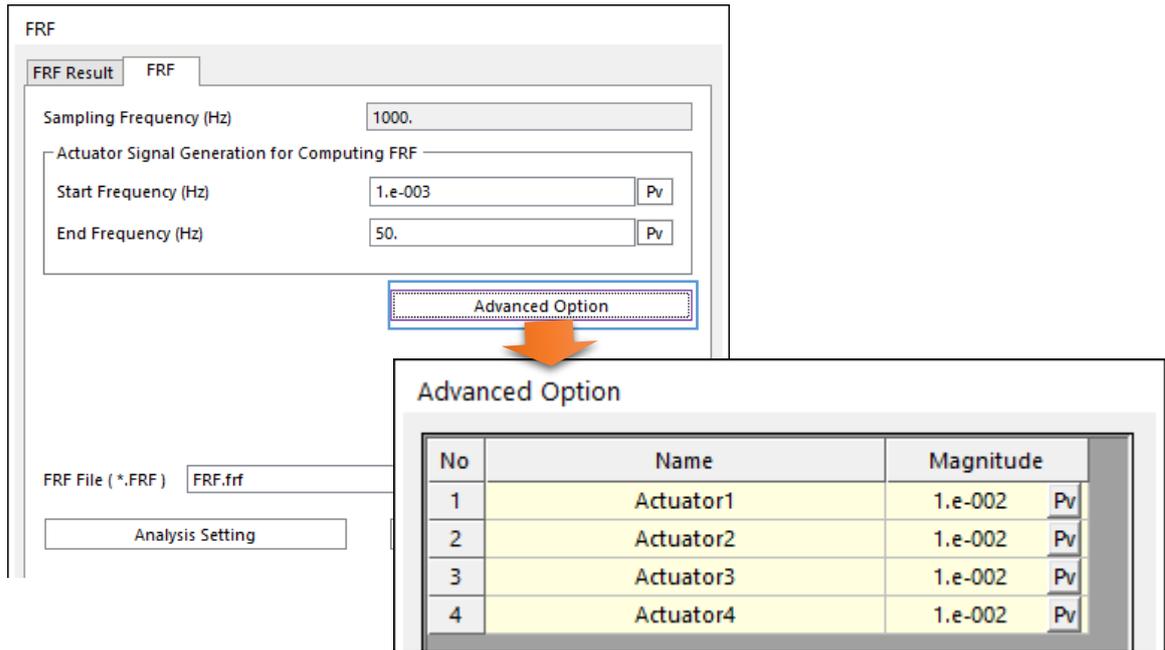
FRF 을 수행하기 위해서는 TACT(Actuator1), TACT(Actuator2) 등의 Expression 이 정의된 부분에 입력신호 주파수가 순차적으로 변경되는 Sweep Sine Function 이 적용됩니다. 이때 필요한 주파수 구간을 Start/End Frequency(Hz)에서 정의하게 됩니다.



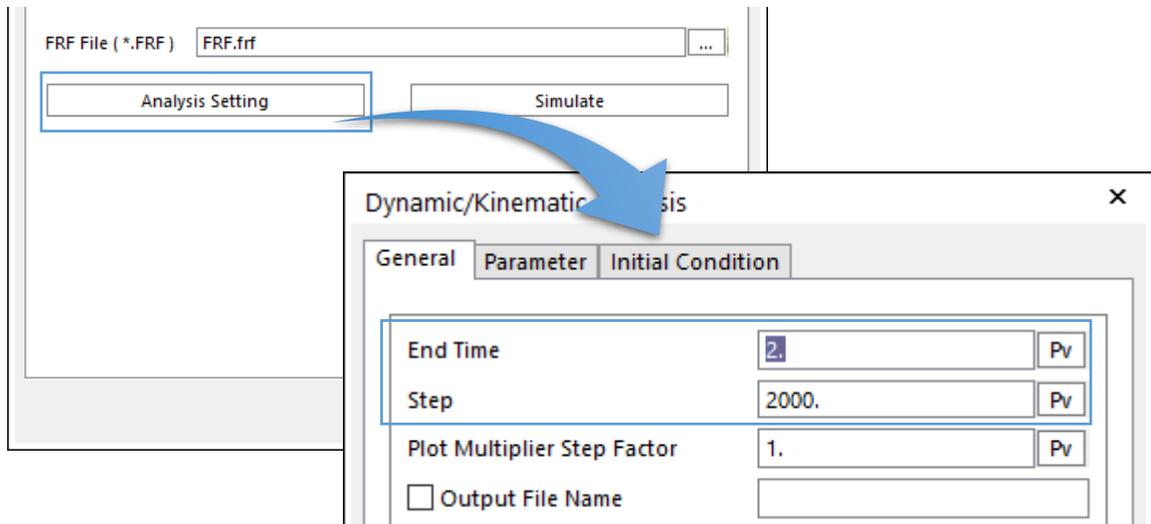
1. TSG 탭의 **Simulation** 탭에서 **FRF** 아이콘을 클릭합니다.
2. **FRF** 대화상자에서 다음과 같이 작업합니다.
 - **Start Frequency** 을 **0.001** 로 입력합니다.
 - **End Frequency** 을 **50** 으로 입력합니다. Target Signal 은 앞서 50Hz 구간까지 필터링 된 결과이기 때문에 50Hz 을 End Frequency 로 적용합니다.



- **Advanced Option** 을 클릭하여 **Sweep Sine Function** 의 **Magnitude** 을 **0.01** 로 입력합니다. 본 튜토리얼 모델은 MKS 단위를 사용하고 있으므로, 기본 설정 값인 1 을 적용하게 되면 Tire 의 변위가 1.0m 만큼 이동하게 되어 과도한 조건이 됩니다.



- **FRF File** 에 **FRF.frf** 라는 이름으로 모델과 같은 폴더에 생성되도록 정의합니다.
- **Analysis Setting** 을 클릭하여 **End Time** 을 **2** 로, **Step** 을 **2000** 으로 입력합니다. 여기서 중요한 사항은, **Target Signal** 을 생성함에 있어 **Sampling Frequency** 를 고려하여 **End Time** 과 **Step** 을 설정해야 합니다. 본 튜토리얼의 **Sampling**



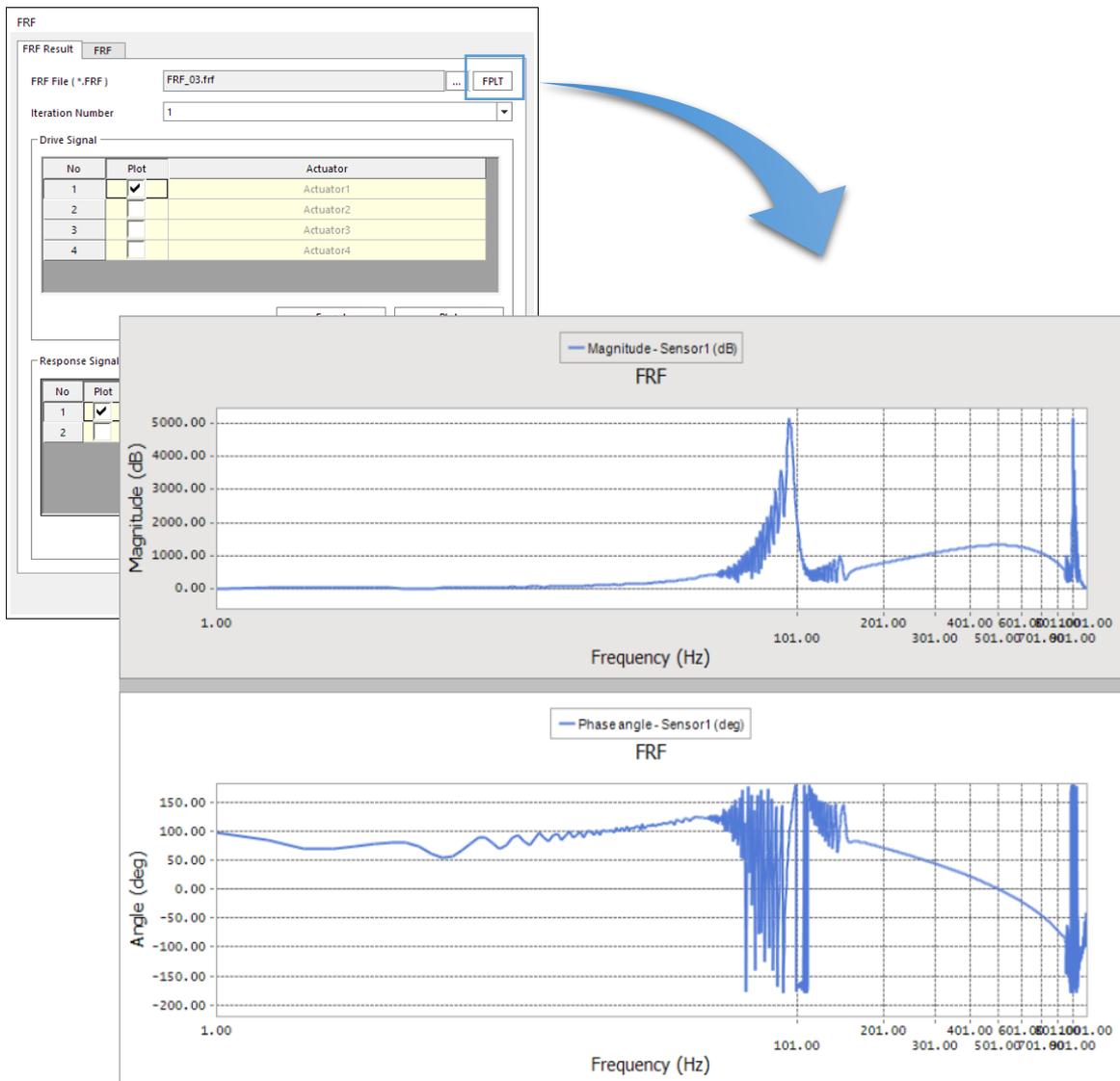
Frequency 는 **1000Hz** 이므로 **End Time** 은 **2**, **Step** 은 **2000** 으로 설정합니다.

- **Simulate** 을 클릭하여 해석을 진행합니다. 설정한 **Actuator** 수만큼 해석을 수행하게 됩니다.

FRF 결과보기

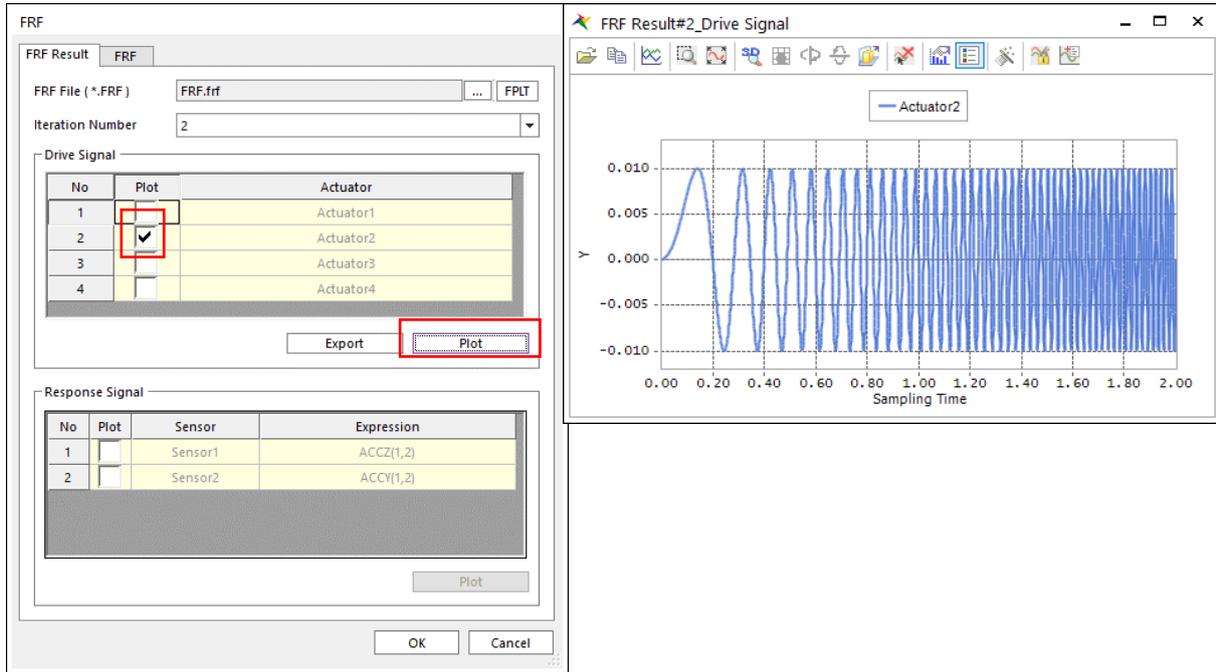
FRF 수행 후 생성된 frf file 의 결과를 Scope 을 통해서 확인합니다.

1. FRF 해석이 종료되면 자동으로 **FRF** 대화 상자에서 **FRF Result** 탭으로 이동합니다.
2. **FPLT** 을 클릭하여 **Plot Window** 를 연 뒤, 필요한 응답 값을 더블 클릭하여 FRF 결과를 직접 확인합니다.

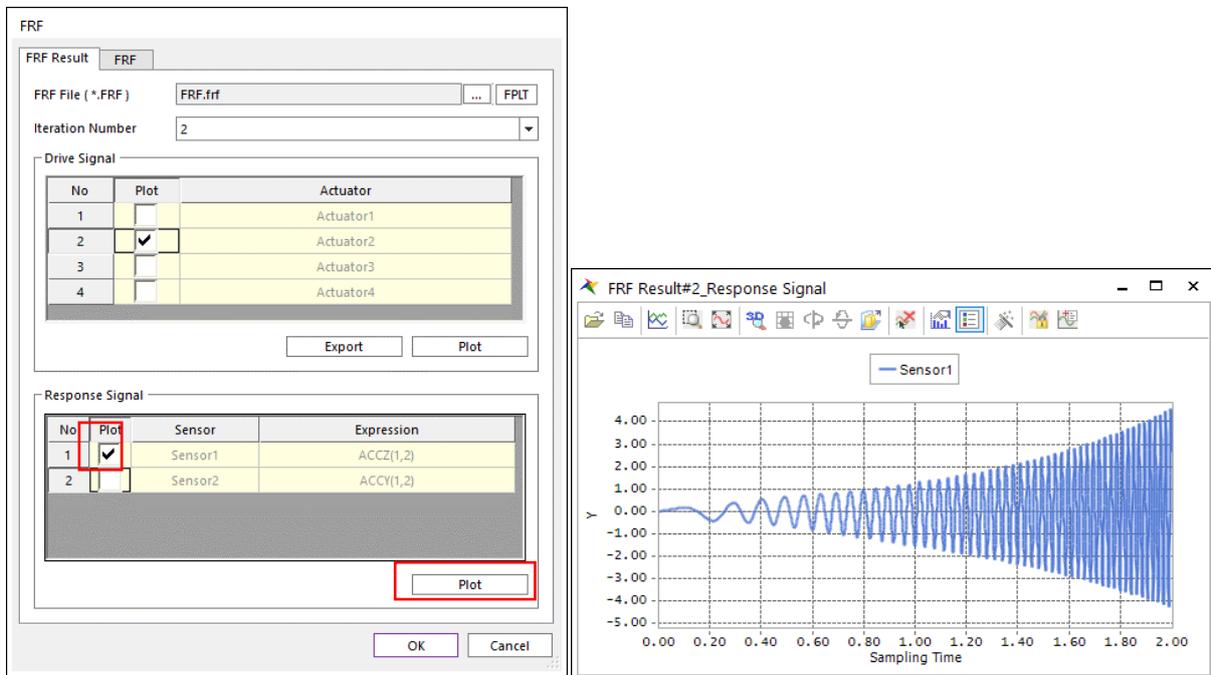


3. **FRF** 아이콘을 다시 클릭하여 **FRF** 대화상자를 엽니다.
4. **Iteration Number** 을 2 로 바꾸고 **Drive Signal** 에서 **Actoator2** 의 **Plot** 을 체크하고 **Plot** 을 클릭하여 Actuator2 에 대한 Drive Signal 을 Scope 로 그려봅니다. FRF 과정에서

Actuator 에 적용되는 Sweep Sine Function 은 각각 개별적으로 적용하게 되는데, 하나의 Actuator 가 가진을 하게 되면 나머지 Actuator 에는 zero 값이 입력되게 됩니다.



5. **Sensor1** 에 대한 Response Signal 을 **Plot** 에서 체크하고 **Plot** 을 클릭하여 확인합니다. **Drive Signal** 로 **Sweep Sine Function** 이 입력되었을 때 **Sensor** 에서 도출된 결과를 보여줍니다.



Chapter

5

Iteration 수행하기

목적

Iteration 과정은 계산된 FRF 결과를 바탕으로 Sensor 에서 측정되는 Response Signal 을 사용자가 정의한 Target Signal 과 최대한 일치시키기 위해서 Actuator 에 가해지는 Drive Signal 을 반복적인 시뮬레이션을 통해 도출하는 과정입니다

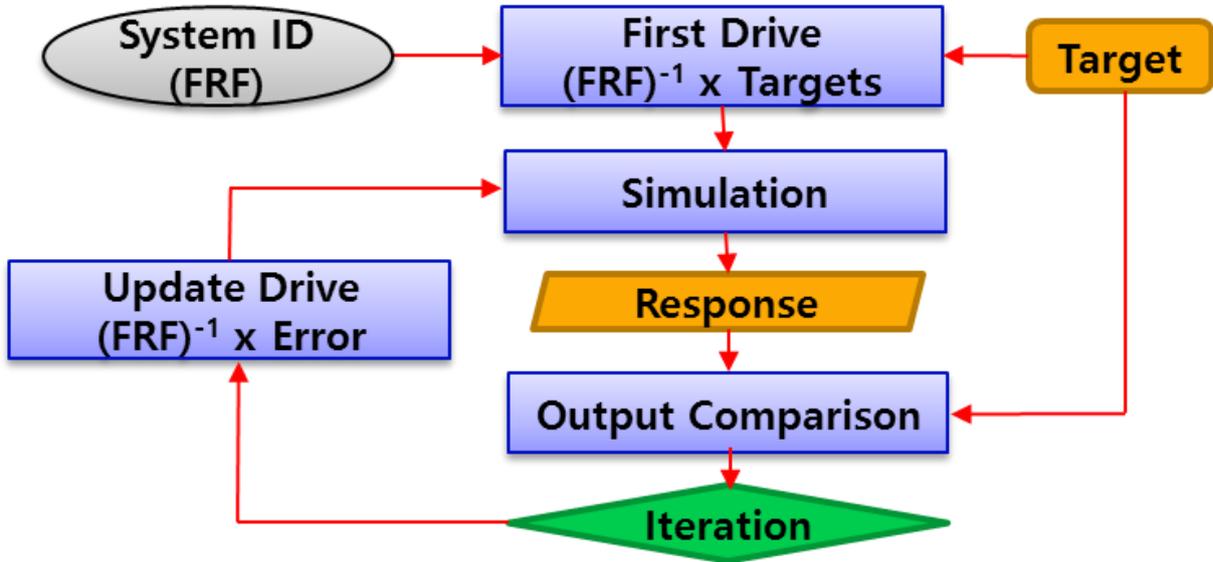


예상 소요 시간

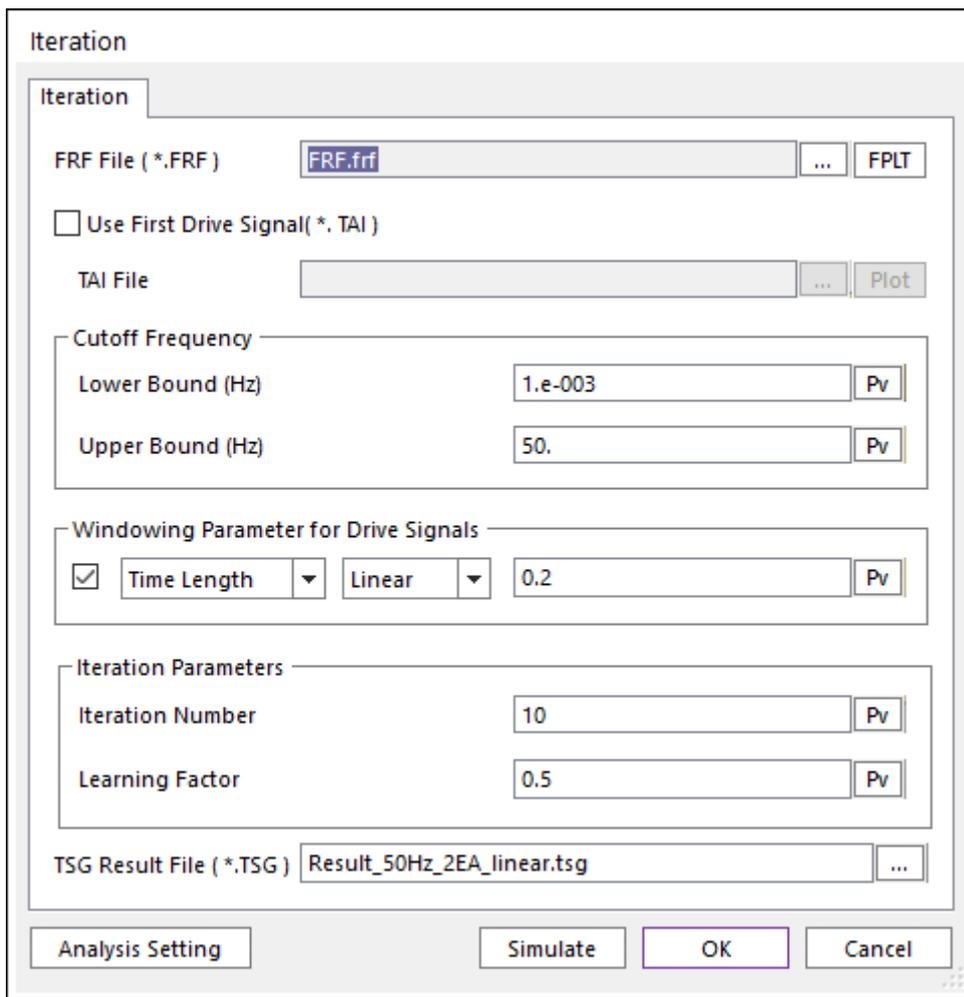
10 분

Iteration 수행하기

다음과 같은 순서로 해석을 수행하게 됩니다.



1. TSG 탭의 **Simulation** 그룹에서 **Iteration** 아이콘을 클릭합니다.
2. **Iteration** 대화상자에서 다음과 같이 입력합니다.
 - **FRF File** 에 이전에 생성한 FRF 파일을 가져옵니다. 일반적으로는 자동으로 불러와 있습니다.
 - **Cutoff Frequency, Windowing Parameter** 는 **Target Output List** 대화상자와 **FRF** 대화상자에서 적용된 설정을 그대로 사용합니다.
 - **Iteration Number** 을 **10** 으로 입력하여 시뮬레이션 반복횟수를 정의합니다.
 - **Learning Factor** 에 **0.5** 를 입력하여 **Target Signal** 과 **Response Signal** 과의 오차를 보정할 계수 값을 정의합니다.
 - **TSG Result File** 에 **Result_50Hz_2EA.tsg** 라는 이름으로 모델과 같은 폴더에 생성되도록 정의합니다.

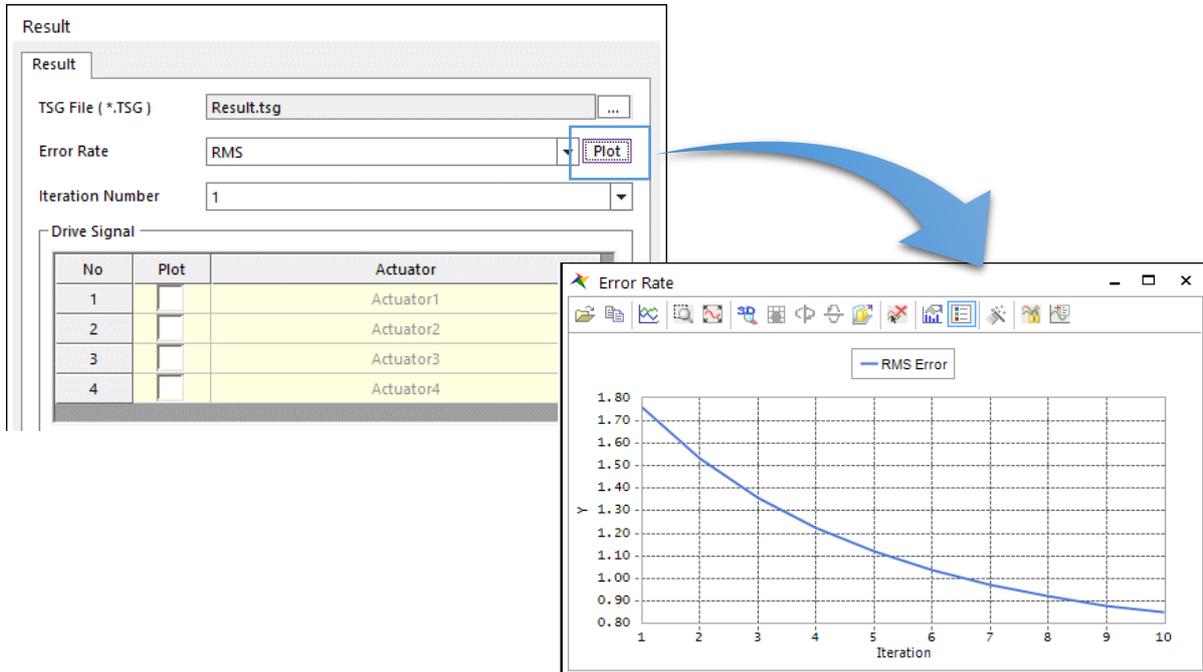


- **Simulate** 을 클릭하여 해석을 수행합니다.

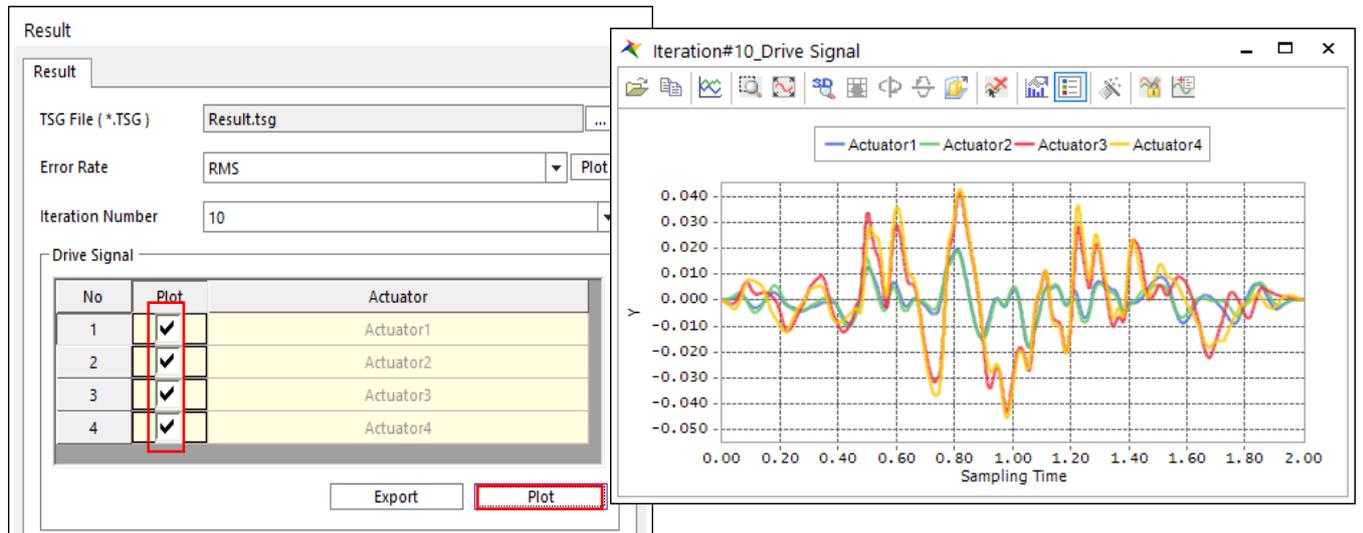
Iteration 결과 보기

Iteration 과정을 거친 후 생성된 tsg file 의 결과를 Scope 을 통해서 확인합니다.

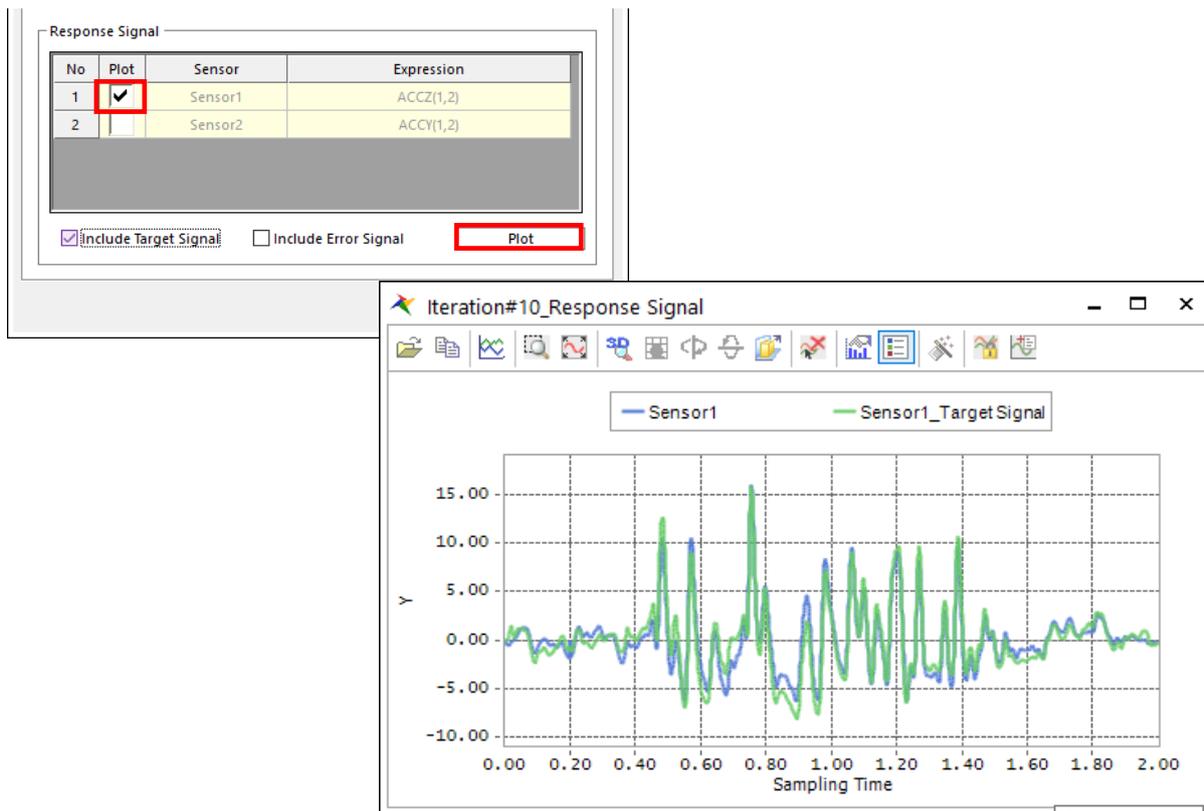
1. Iteration 이 종료되면 자동으로 **Result** 대화상자가 나타납니다. (만약, 별도로 실행하고 싶다면, **TSG** 탭의 **Result** 그룹에서 **Result** 아이콘을 클릭합니다.)
2. **Result** 대화상자에서 다음과 같이 작업합니다.
 - **Error Rate** 을 **RMS** 로 선택하고 **Plot** 을 클릭합니다. Iteration 순서에 따른 Response Signal 에 대한 **Target Signal** 의 매 순간 오차를 **RMS** 기법으로 계산하여 도출된 결과를 보여줍니다.



- 마지막 해석에서는 가장 작은 0.85 정도의 오차율을 가집니다.
- **Iteration Number** 을 **10** 으로 선택하고 전체 Actuator 에 대한 **Drive Signal** 을 보기 위해 **Plot** 을 체크하고 **Plot** 을 클릭합니다.



- **Iteration Number** 을 **10** 으로 선택하고 **Sensor1** 에 대한 **Response Signal** 을 **Target Signal** 과 함께 보기 위해 **Plot** 을 체크하고 **Include Target Signal** 을 체크한 다음 **Plot** 을 클릭합니다.



- **Response Signal** 이 **Target Signal** 과 거의 유사한 경향을 가진 것으로 확인되었습니다. 이로써 생성된 **Drive Signal** 이 문제없음을 확인하였습니다.
3. **Plot Window** 에서 **TSG/TimeSignalGeneration1/TSG Actuator** 값들을 file 로 export 하여 같은 가진을 다른 system 에도 적용할 수 있습니다.

Thanks for participating in this tutorial!