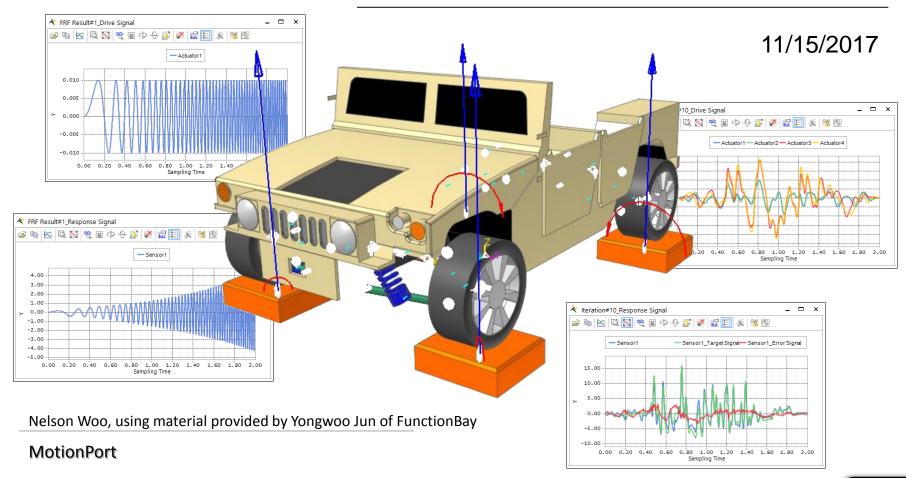


#### **RecurDyn/TSG (Time Signal Generator)**

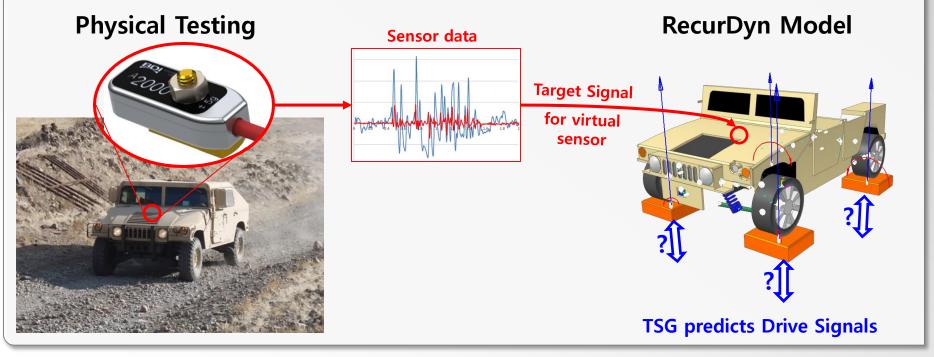




#### RECURDYN

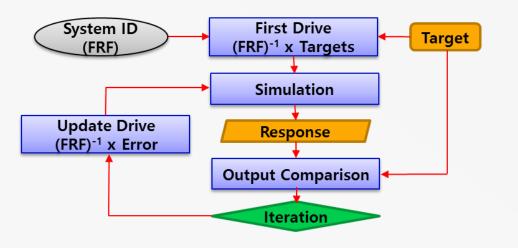
### **Overview**

- RecurDyn/TSG (Time Signal Generator) allows analysts to use data from physical testing to replicate test conditions in an analogous CAE (Computer Aided Engineering) model.
- Data from physical testing is limited by the type and number of sensors that can be instrumented, leading to incomplete picture of what happens during the test.
- RecurDyn/TSG can take available test outputs (i.e. vehicle chassis acceleration from accelerometers) and backsolve to find the test inputs (i.e. displacement to be enforced at the tires), which can be difficult to obtain due to physical limitations of test instrumentation.
- For automotive applications, removes the requirement for virtual terrain, tire, and driver modeling, which can be difficult to characterize correctly.
- Complex nonlinear behavior of the model is automatically taken into account as RecurDyn/TSG solves for the input signals.



### Procedure

- 1. Create actuators to apply the displacement/load input to the model.
- 2. Create virtual sensors in the RecurDyn model where the actual sensors are located in the physical model.
- 3. Import the physical test data to become the desired output for the virtual sensors. This is the **Target Signal**.
- 4. RecurDyn/TSG runs a frequency response analysis to determine the system's Frequency Response Function (FRF).
- Using the FRF and the Target Signal, RecurDyn/TSG then iteratively backsolves to find the proper **Drive Signal** for the actuators to reproduce the Target Signal from the virtual sensors.
- 6. If the responses of the sensors are similar to the measured Target Signals, then RecurDyn model can be regarded that it is actuated similarly to the real system.



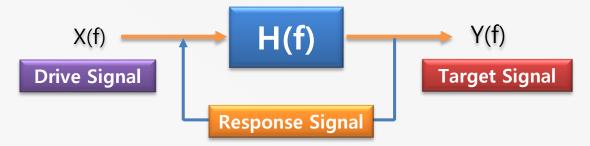
### **TSG Terminology**

#### Actuator:

- Drives the system at a various location of the system.
- It is defined using a Joint Motion or a Force in an MBD model
- This is done using the function **TACT()** in an expression.

#### Sensor:

- The response of a system is calculated using sensors.
- Eventually, the responses of the sensors should match the user-defined Target Signals closely.



**Drive Signals:** Inputs to MBD model at the locations of the virtual **Actuators**. **Response Signals:** Outputs of RecurDyn at the location of the virtual **Sensors**.

**Target Signals: User-Defined Signals** that were obtained from physical testing. RecurDyn varies the Drive Signals iteratively until the Response Signals match the Target Signals.

### **TSG Function Layout**



The RecurDyn/TSG functions are arranged in the order of the user workflow:

- 1. Define the **Actuator**s as forces or motion assigned to joints.
- 2. Define the **Sensor**s as measurements of displacement, velocity, acceleration, force, or torque at a specific location.
- 3. Generate the **Target Signal** from imported data.
- 4. Obtain the **FRF** (Frequency Response Function) by running a frequency response analysis.
- 5. Perform an **Iteration** of simulations to solve for the Drive Signals that make the Response Signals match the Target Signal.
- 6. Review the **Results**, observing how closely the Response Signals match the Target Signals.

# **RecurDyn/TSG Tutorial**

# Tutorial (1)

#### 1. Actuator

#### **①** Actuator: define the number of actuators.

- a. As shown in the figure below, select the **Actuator** icon to open Actuator List dialog.
- b. Create 4 actuators using Add button.

#### Apply the actuators to the joints.

- a. The actuators are assigned to Joint Motion or Force using the function expression TACT().
  - Apply the below functions to TraJoint1-TraJoint4 as joint motions (Displacement type).
  - Use TACT(Actuator1), TACT(Actuator2), TACT(Actuator3), and TACT(Actuator4).
- b. In this tutorial, the actuators will move 4 shakers below each tire up and down.

Actuator       Image: Sensor Target       Image: Sens	Properties of TraJoint1 [ Current Unit : N/kg/m/s/deg ]         General Connector Joint         Type Translational         Motion         Include Motion
No     Use     Name       1     ✓     Actuator1       2     ✓     Actuator2       3     ✓     Actuator3       4     ✓     Actuator4	Motion TraJoint1 Ground.Marker5 Shaker_01.Marker1 Motion TraJoint2 Motion TraJoint3 Shaker_02.Marker1 Shaker_03.Marker8 Shaker_03.Marker1 Motion TraJoint4 Shaker_04.Marker9 Shaker_04.Marker1
RECURDYN	© 2017 MotionPo

# Tutorial (2)

- 1. Sensor
  - Sensor: The response of the simulation which will be compared with the Target Signal.

Result Result

- a. As shown in the below figure, select the **Sensor** icon to open Sensor List Dialog.
- **b.** Add 2 sensors in Sensor List dialog.

Sensor

100

Actuator

**②** Define the function expressions for the sensors

Target

- a. Any function expression can be used for sensors.
  - Acceleration(ACCX, ACCY, ACCZ), Velocity(VX, VY, VZ), Disp.(DX, DY, DZ)
  - Force(FX, FY, FZ, TX, TY, TZ), Stress(SX, SY, SZ), Strain(EX, EY, EZ), Etc.
- b. In this tutorial, Z-Acceleration and Y-Acceleration of CM of Chassis will be used.

 sor Li nsor Li					Expression Name Sensor1		Sensor1 ACCZ(1, 2)
No 1 2	Use	Name Sensor1 Sensor2	Expression ACCZ(1,2) ACCY(1,2)	E	ACCZ(1,2)		Sensor2 ACCY(1, 2)
							<b>Argument List</b> 1: CHASSIS.CM 2: Ground.InertiaMarker
				v	Available → Two Function expressions → Type From Fortran 77 Functions → Type Simulation constants → Type Displacement → Type Acceleration	Argument List	Entity CHASSIS.CM Ground.InertiaMarker

# Tutorial (3)

- 1. Target (1)
- Actuator Sensor Target Signal
- 1 Target: User-defined input data.
  - a. Time-dependent continuous data set measured from experiment or simulation. Performance index of RecurDyn/TSG.
  - b. After importing measured data, Target data needs to be regenerated (into a .target file).

#### 2 Import csv file

- a. .csv file (text file) is used.
- b. The number of Target Data in .csv file is dependent on the number of Sensors.
- c. The sequence of the data in. csv file must be, time1, data1, time2, data2, ...
- d. In this tutorial, there are 2 sensors, so 4 data columns must be written in .csv file as shown in the below figure.
  - Even if the time data is duplicated, it should be written respectively.
  - The Target Data must be written according to the sequence of the Sensors

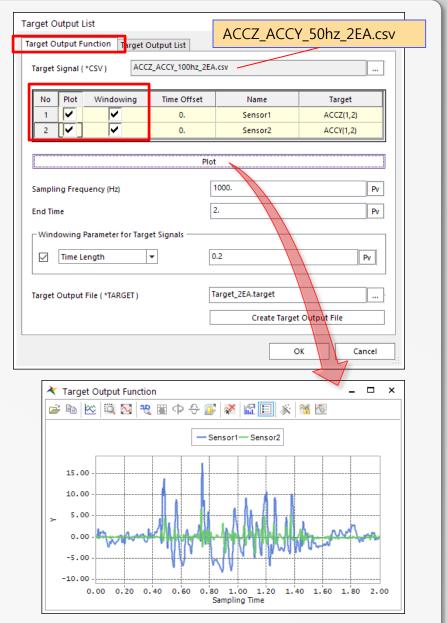
	time1	data1	time2	data2
	0	0	0	0
	0.001	-0.0004	0.001	0.000186
	0.002	-0.00303	0.002	0.001564
	0.003	-0.01027	0.003	0.009046
	0.004	-0.02232	0.004	0.033633
	0.005	-0.03801	0.005	0.080458
	0.006	-0.05629	0.006	0.131874
	Υ	]	L	
Т	arget of S	Sensor1	Target	of Sensor2

# **Tutorial (4)**

### 1. Target (2)

#### 1 Tips to generate Target Data

- a. The data measured from experiment usually includes high-frequency data as well as low-frequency data.
- b. The high-frequency data can cause noise and error during simulation using TSG.
- c. So it is recommended to filter the data using Low Pass Filter so that the filtered data can include the signal below 50~100Hz when you generate \*.csv file.
  - You can use Low Pass Filter in RecurDyn/Plot.
  - The sample file of this tutorial, ACCZ\_ACCY\_5 Ohz\_2EA.csv includes the signal below 50Hz.
- ② Import .csv file
  - a. Import .csv file in **Target Output Function** tab of Target Output List dialog.
  - b. You can **plot** the Target Data for Sensor1 and Sensor2.



### **Tutorial (5)**

#### 1. Target (3)

#### **1** Sampling Frequency

- a. The number of data per 1 second. 1000 is used in this tutorial.
  - If Simulation End Time is 2 sec, the number of data must be 2000.
- b. Since the number of data in .csv file doesn't match the required number, you must regenerate the data file for the given sampling frequency and end time.
- c. You will create \*.target data in the next page.

#### **②** Windowing Parameter for Target Signals

- a. When the Time Signal is converted to frequency signal using Fourier Transform, the initial signal and the final signal is set to zero to minimize the error.
  - Windowing is applied about 10% of the entire time.
  - In this tutorial, 0.2 with Time Length type is used.

Sampling Frequency (Hz)	1000. Pv
End Time	2. Pv
Windowing Parameter for Target Signals	0.2 Py
Target Output File ( *TARGET )	Target_2EA.target Create Target Output File
	OK Cancel

# **Tutorial (6)**

2. Pv	1. Target (4)
0.2 Pv	<ol> <li>Create Target Data         <ol> <li>Create .target file from .csv file based</li> </ol> </li> </ol>
Target_2EA.target ) Create Target Output File	Sampling Frequency, End Time, Wind Parameter.
OK Cancel	② Create Target Output File
List	atarget file is a binary format for bett performance.
Iarget_2EA.target Iame Target Insor1 ACCZ(1,2) Dear2 ACCV(1,2)	b. After specifying the file name and th path, click <b>Create Target Output File</b> button to create .target.
	c. Click <b>Plot</b> button to plot the data in .target.
	A Target Output     _ □ ×
	Image: Image
	0.2 Pv Target_2EA.target Create Target Output File OK Cancel List target_2EA.target

#### © 2017 MotionPort

#### RECURDYN

# Tutorial (7)

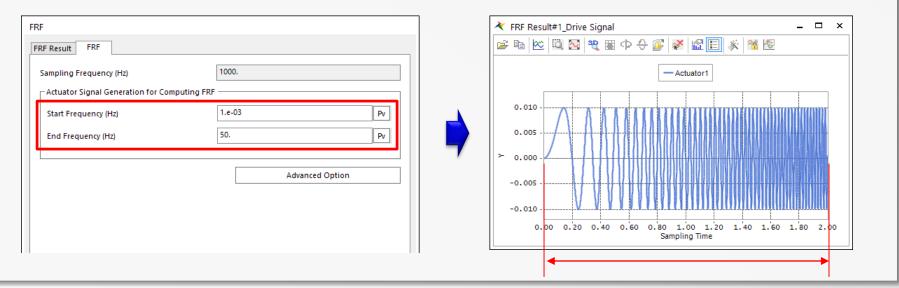
### 1. FRF (1)

#### **①** FRF (Frequency Response Function)

a. Computes the linearized model for System Identification (Transfer Function, H(f)).



- 2 Procedure (1)
  - a. Start/End Frequency(Hz)
    - To perform FRF, the frequency of the signal for the actuators ('TACT(Actuator1)') is gradually increased using a Sweep Sine Function. Start/End Frequency define the sweep sine function.
    - Since 0Hz is not valid, Set Start Frequency 0.001Hz.
    - Since the Target Signal is the data below 50Hz, set End Frequency 50Hz.



#### RECUR<mark>dyn</mark>

### **Tutorial (8)**

### 1. FRF (2)

- ① Procedure (2)
  - a. Set Magnitude of Sweep Sine Function in Advanced Option.
    - Since the model in this tutorial uses MKS unit, Magnitude = 1 means that the displacement of the tire is 1m, which is too large for this model.
    - Therefore set all the Magnitudes 0.01 to displace the tires by 10 mm.

FRF			Advar	nced Option	
FRF Result FRF			No	Name	Magnitude
Sampling Frequency (Hz)	1000.		1	Actuator1	1.e-02 Pv
Sampling requercy (nz)		<b>x</b>	2	Actuator2	1.e-02 Pv
Actuator Signal Generation for Computing FRF			3	Actuator3	1.e-02 Pv
Start Frequency (Hz)	1.e-02 Pv :		4	Actuator4	1.e-02 Pv
End Frequency (Hz)	50. Pv	'			
	Advanced Option				

- b. Specify the file name and path for FRF result.
- c. Adjust Analysis Setting for Dynamic Analysis, then click Simulation button.
  - End Time and Step must be consistent with the Sampling Frequency.
  - Since the Sampling Frequency in this tutorial is 1000Hz,
  - Set End Time = 2sec, and Step = 2000.

			D	ynamic/Kinematic Analysis		>
FRF File (*.FRF)	FRF_03.frf			General Parameter Initial Conditio	ı	
Analy	vsis Setting	Simulate		End Time	2.	Pv
				Step	2000.	Pv
				Plot Multiplier Step Factor	1.	Pv

## **Tutorial (9)**

### 1. FRF (3)

- ① Procedure (3)
  - a. After you click **Simulation** button, the simulation is performed once for each actuator.
    - In this tutorial 4 simulations are performed.
    - During the FRF, the Sweep Sine Function is applied to one Actuator while all the other actuators are held at 0.
  - b. After simulation, under the FRF Result tab, you can click **FPLT** to plot the frequency responses of the system.
  - c. You can also **Plot** the **Drive Signal** (Sweep Sine Function) of the actuators and **Response Signal** of the sensors (next page).

FRF			Magnitude - Sensori (dB) FRF
FRF Result FRF			5000.00
FRF File (*.FRF)		FRF_03.frf FPLT	g 2000, 0
Iteration Number		1	E 2000.00
Drive Signal —			0.00 1.00 201.00 401.00 601.00 100 101.00 301.00 501.00 701.00051.0
No	Plot	Actuator	Frequency (Hz)
1	<ul> <li>Image: A state</li> </ul>	Actuator1	- Phase angle - Sensor1 (deg)
2		Actuator2	FRF
3		Actuator3	150.00 100.00
4		Actuator4	100.00 G 50.00 V 6.00 V 50.00 -150.00 -150.00
		Export Plot	-200.00

### **Tutorial (10)**

F	🛎 🖻 🗠 🖾 🤫 🖩 🗘 🕂 🎑 🔛 🖄 🖉
RF Result FRF	- Actuator1
FRF File (*.FRF)     FRF_03.frf      FPLT       Iteration Number     1        Drive Signal	0.010 -
No     Plot     Actuator       1     ✓     Actuator1       2      Actuator2       3      Actuator3       4      Actuator4	> 0.000
Export Plot	✓ FRF Result#1_Response Signal
No Plot Sensor Expression	
No         Plot         Sensor         Expression           1         ✓         Sensor1         ACCZ(1,2)           2         ✓         Sensor2         ACCY(1,2)	

#### RECUR<mark>dyn</mark> |

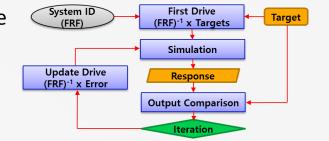
# Tutorial (11)

- 1. Iteration
  - Iteration: performs iterative simulations to find the Drive Signals applied to Actuators to match the Response Signal of sensor to the Target Signal as closely as possible using FRF results.



#### Procedure

- **FRF file:** FRF result calculated in the previous step is set automatically. Or you can specify .frf file.
- **Cutoff Frequency and Windowing Parameters:** Use the same values used during FRF.
- Iteration Parameters: Set Iteration Number = 10 and use the default value for Learning Factor (0.5).
- **TSG Result File:** Specify the file name and path to save the TSG Results.
- Click **Simulate** button.



Iteration				
Iteration				
FRF File (*.FRF)	FRF_50HZ_04.frf			FPLT
Use First Drive Signal( *.	TAI)			
TAI File				Plot
Cutoff Frequency ——				
Lower Bound (Hz)		1.e-03		Pv
Upper Bound (Hz)		50.		Pv
└── Windowing Parameter fo	r Drive Signals			
Time Length	•	0.2		Pv
Iteration Parameters				
Iteration Number		10		Pv
Learning Factor		0.5		Pv
TSG Result File (*.TSG )	Result_50Hz_2EA.ts	g		
Analysis Setting		Simulate	ОК	Cancel

# Tutorial (12)



① Post-processor of TSG to review the result in .tsg after iterative simulations.

Iteration

FRF

FRF

Simulation

 $\sqrt{3}$ 

No.

Result

Result

- a. Error Rate (RMS): For each iteration, the RMS of the difference between Response Signal (Sensor) and Target Signal at every instant.
- **b.** Error Rate (RMS (Error Rate)): The relative difference between the RMS of Target Signal for entire time and the RMS of Response Signal (Sensor) for entire time.
- 2 Procedure (1)

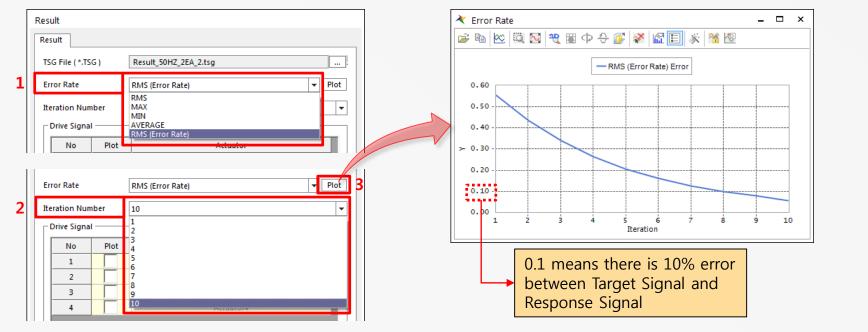
Actuator

Sensor

Signal

Target

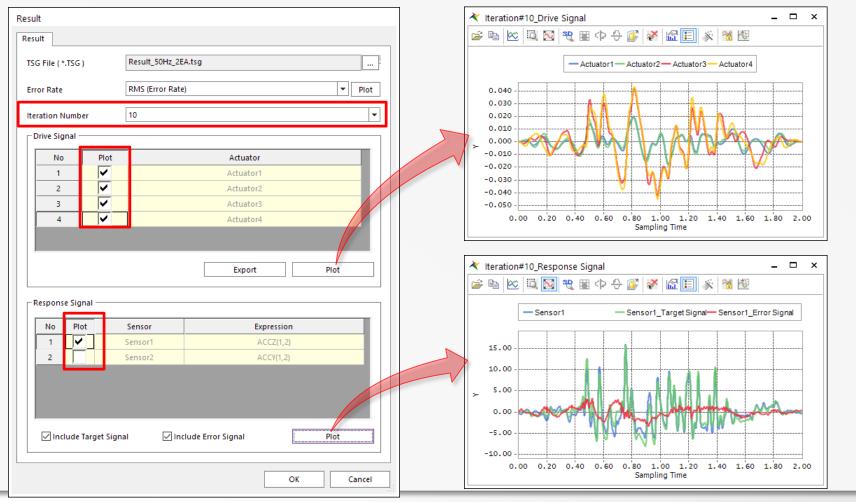
a. Specify the type of **Error Rate**, **Iteration Number**, and click **Plot** button to review the error rate of that iteration.



### **Tutorial (13)**

#### 1. Result (2)

- ① Procedure (2)
  - a. Select the desired Iteration Number
  - **b.** Plot the Drive Signal of the selected actuators and Response Signal of the selected Sensors.



### Tutorial (14)

### 1. Result (3)

- ① Procedure (3)
  - a. To perform additional iterations, drive signal results from the last iteration can be transferred to the next iterations using a .tai file.
    - Select Iteration Number and click Export, storing Drive Signals of selected Iteration Number in .tai file.
    - In Iteration dialog, check Use First Drive Signal and specify .tai file you just generated.

Result		Iteration	
Result		Iteration	
TSG File (*.TSG ) Result_50HZ_2EA.tsg		FRF File (*.FRF)	FRF_50HZ_04.frf FPLT
Error Rate RMS (Error Rate) 💌 Plot		🗹 Use First Drive Signa	I(*. TAI)
Iteration Number 10 🔹		TAI File	Actuator10.tai Plot
Drive Signal	.tai file	Cutoff Frequency —	
No Plot Actuator	containing	Lower Bound (Hz)	1.e-03 Pv
1 Actuator1	drive	Upper Bound (Hz)	50. Pv
2     ✓     Actuator2       3     ✓     Actuator3	signal	Mindauine Deservation	- for Drive Gineral
3     Actuator3       4     Actuator4		Windowing Parameter	▼ 0.2 Pv
Export	<b>_</b>	- Iteration Parameters -	
		Iteration Number	10 Pv
Response Signal		Learning Factor	0.25 Pv
No         Plot         Sensor         Expression           1         V         Sensor         ACC7(1/2)			
1         ✓         Sensor1         ACCZ(1, 2)           2         Sensor2         ACCY(1, 2)		TSG Result File (*.TSG)	Result_50HZ_2EA.tsg
		Analysis Setting	Simulate OK Cancel

### **Tutorial (15)**

### 1. Result (4)

- ① Procedure (4)
  - a. When you plot Sensor data in 'Response Signal'
    - You can plot Target Signal or Error Signal as well as the output of Sensor.
    - You can use the options Include Target Signal and Include Error Signal.

sult				
esult				
SG File (*.TS	G)	Result_50HZ_2EA.tsg		
rror Rate		RMS (Error Rate)	▼ Plot	
teration Num	ber	10		
- Drive Signal				
No	Plot	Actuator		
1	~	Actuator1		
2		Actuator2		
3		Actuator3		
4	~	Actuator4		★ Iteration#10_Response Signal
				🖙 🖻 🗠 🖾 🔂 🤫 🖩 🗘 🕀 🌮 🌌 🔚 🚿 🆄 🙋
		Export	Plot	
-Response Si	gnal —			- Sensor1 - Sensor1_Target Signal - Sensor1_Error Signal
No Plo	_	Sensor Expres	ssion	15.00
1		Sensor1 ACCZ		15.00
2		Sensor2 ACCY		10.00
				> 5.00
🗹 Include	Target S	iignal 🛛 🗹 Include Error Signal	Plot	-5.00
				-10.00
				0.00 0.20 0.40 0.60 0.80 1.00 1.20 1.40 1.60 1.80 2.00
			OK Cancel	Sampling Time

### **Tutorial (16)**

#### 1. Result (5)

① Procedure (5)

Once the user is satisfied with the drive signal results, they can transfer the data into a spreadsheet or text file to be used in other models.

a. From the plot of the drive signals, click the **Export** button.

esult			★ Iteration#10_Drive Signal     _ □ ×
Result			📴 🖻 🖄 🧠 🤁 🖩 🗘 🗘 🌠 🌠 🔛 🚿
TSG File ( *.TSG )	Result_50HZ_2EA.tsg		Import     Actuator1—Actuator2—Actuator3—Actuator4
Error Rate	RMS (Error Rate)		0.000
Iteration Number Drive Signal	10 💌		
No Plot	Actuator	$\square$	
1	Actuator1		-0.000
2	Actuator2		-0.000
3 🗸	Actuator3 Actuator4		-0.000
Response Signal	Export Plot		0.00 0.20 0.40 0.60 0.30 1.00 1.20 1.40 1.60 1.30 2.00 Sampling Time
No Plot	Sensor Expression		
1	Sensor1 ACCZ(1, 2)		
2	Sensor2 ACCY(1, 2)		
Include Target :	Signal 🛛 Include Error Signal 🛛 Plot		Export Curves, shown on next pa
	OK Cancel		

# Tutorial (17)

#### 1. Result (6)

① Procedure (6)

From the Export Curve Data window that appears, the user can then export the data for each curve to a file, which can be imported into another model to be used as a driving signal.

- a. Load the Export List with data from the Source X List and Source Y List.
- b. Specify an Output Filename.
- c. Click **Export**.

Source			Target
Source X List	Source Y list		Export List
X : Actuator1	Y: Actuator1		X : Actuator1
X : Actuator2	Y: Actuator2		Y : Actuator1
X : Actuator3	Y: Actuator3		
X : Actuator4	Y : Actuator4		
Add X Data		Add Y Data	Remove
Export Options			
Include Title Header		Yes	
Use Scientific Notation		No	
Number of Significant Digit		8	
Output Filename		/\TimeSignalGe	nerator\Actuator1Export.txt

### Conclusions

- RecurDyn/TSG (Time Signal Generator) allows analysts to utilize available physical test data to replicate loading conditions in an analogous MBD model.
- Provides a consistent, automated method for doing this.
- Removes the requirement for virtual terrain, tire, and driver modeling, which can be difficult to characterize correctly.
- Makes it easy for analysts to test new designs based on physical test data of existing designs.

### Questions

Email <u>support@motionport.com</u> to obtain more information.