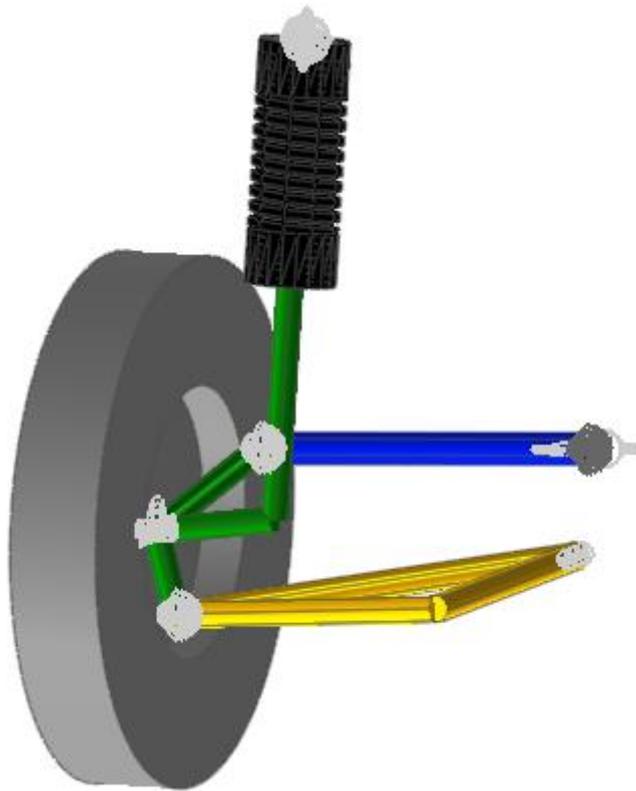




悬挂系统教程 (AutoDesign)



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(Styret), *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[™] is a registered trademark of GLOBEtrrotter 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

教程SampleD概要.....	4
悬挂系统设计问题.....	5
加载模型并查看Yaw & Roll范围.....	6
定义设计变量.....	7
定义performance index.....	9
运行设计优化.....	11
筛选变量的设计优化.....	15

教程 SampleD 概要

模型	描述
Sample_D	<p>悬挂系统设计问题： 悬挂系统设计有两个设计目标，同时，有 27 个设计变量。设计目标是减小轮胎运动的 Yaw-range 和 Roll-Range。这个问题有些复杂，因为这是个多目标问题，并且有很多设计变量。其它的设计工具，通常使用 D-Optimal 设计或 Latin Hypercube，来构建 meta 模型。假如构建二次响应表面模型，D-Optimal 需要 406 个采样点，通过使用 $1+2*27+27*26/2$ 来评价。AutoDesign 没有进行变量筛选，只使用 44 个评价来解决问题。</p> <p>关键点：学习多目标优化的概念和筛选设计变量的步骤。</p>

注意：

在 **RecurDyn V8** 中，求解器只编译 **IF (Intel Fortran)** 11.0。因此，优化结果可能与老版本有差别。

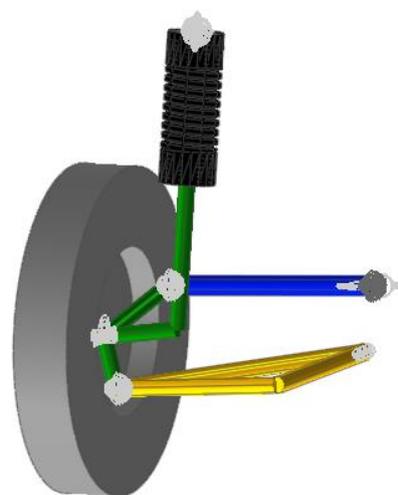


悬挂系统设计问题

考虑汽车悬挂系统的简化模型。系统由 5 个部分组成：摆臂、拉杆、转向节、减震器阻尼和轮胎。当轮胎沿垂直方向移动时，仿真结果给出了阻尼过程和运动特性。

所有的设计变量均为运动副的几何坐标。设计目标是轮胎垂直方向行为变化的 **Yaw** 与 **Roll** 同时最小化。

这个问题包含 9 个运动副的 27 个设计变量。不管怎样，我们将尝试直接减小 **Yaw-range** 和 **Roll-range**。换句话说，不进行设计变量的筛选。下一步，在筛选设计变量后，重新解决同样的问题。然后，比较优化的结果。



在 Sample-D 中，打开相关文件

Sample		<InstallDir>\Help\Tutorial\AutoDesign\SuspensionSystem\Examples\SAMPLE_D0.rdyn
Solution	1	<InstallDir>\Help\Tutorial\AutoDesign\SuspensionSystem\Solutions\SAMPLE_D0.rdyn
	2	<InstallDir>\Help\Tutorial\AutoDesign\SuspensionSystem\Solutions\SAMPLE_D1.rdyn
	3	<InstallDir>\Help\Tutorial\AutoDesign\SuspensionSystem\Solutions\SAMPLE_D2.rdyn

注意：可以更改文件的安装目录，文件可以放置在设置的任意文件夹。

Chapter

1

加载模型并查看 Yaw & Roll 范围

加载 base 模型并查看动画:

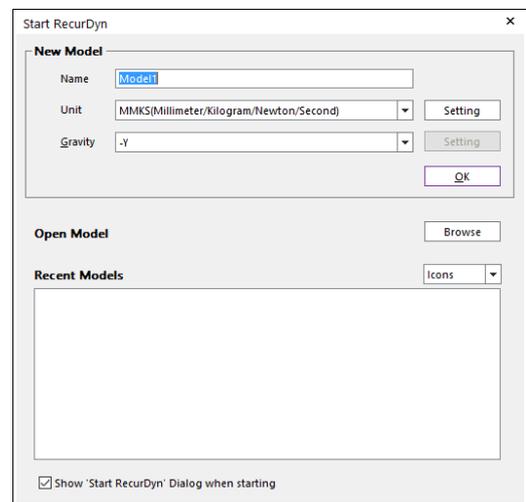


1. 双击桌面上的 RecurDyn。

RecurDyn 启动，并弹出 Start RecurDyn 对话框。



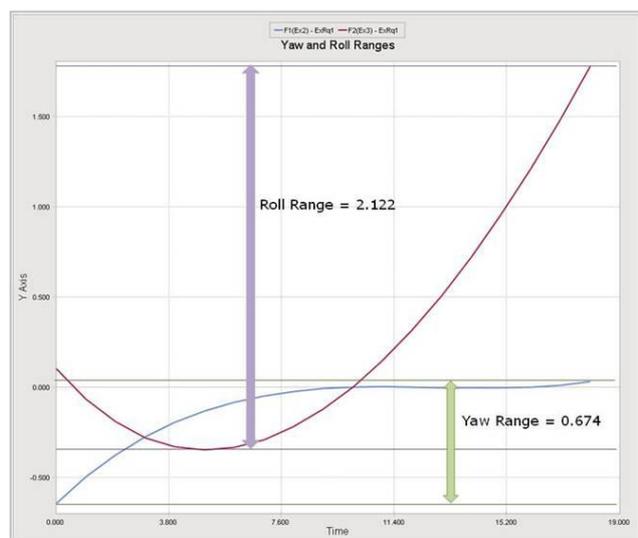
2. 关闭 Start RecurDyn 对话框，使用现有的模型。
3. 在工具栏，点击 Open 工具，并在 tutorial 相同目录下选择 'Sample_D0.rdyn'。悬挂系统显示在模型窗口。



4. 点击 Dynamic/Kinematic 按钮。



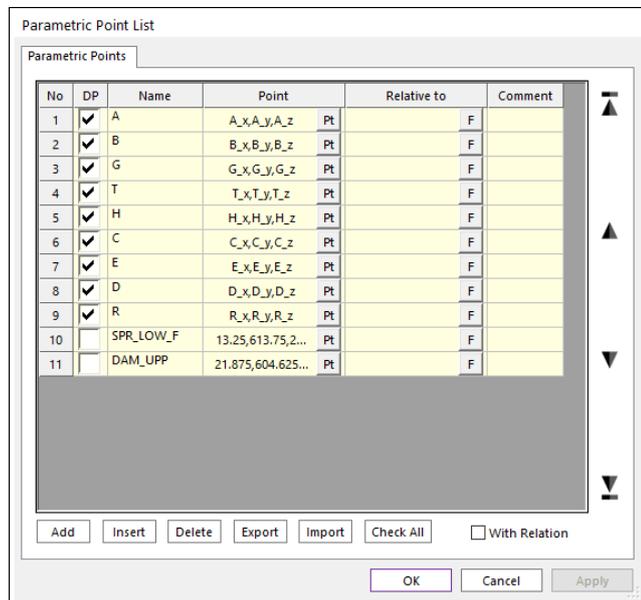
5. 点击 Plot Result 按钮。分别点击 Request-expression-F1 和 F2, 可以看到如右图所示的绘图数据。Expression 2 & 3 分别是 Yaw and Roll angles。



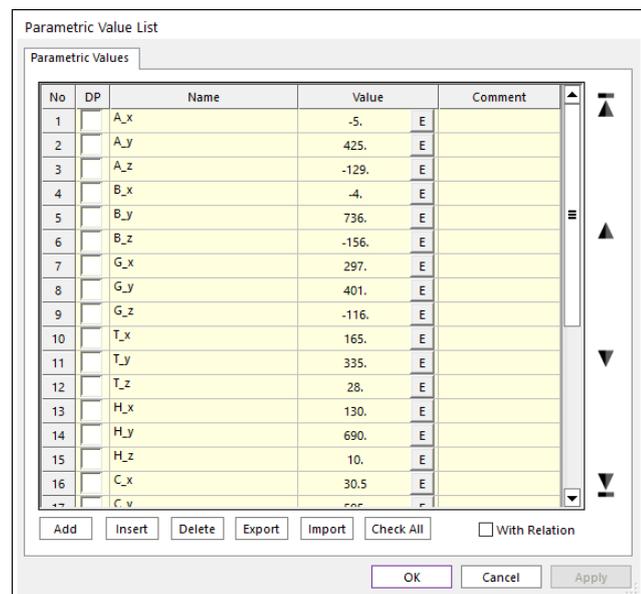
Chapter 2

定义设计变量

采用 **Parametric Points**，定义运动副的所有几何坐标，如右图，表示为 **A_x**, **A_y**, ..., **R_y** 和 **R_z**。



下一步，在 **Subentity** 的 **Parametric Value** 菜单下，定义参数值。

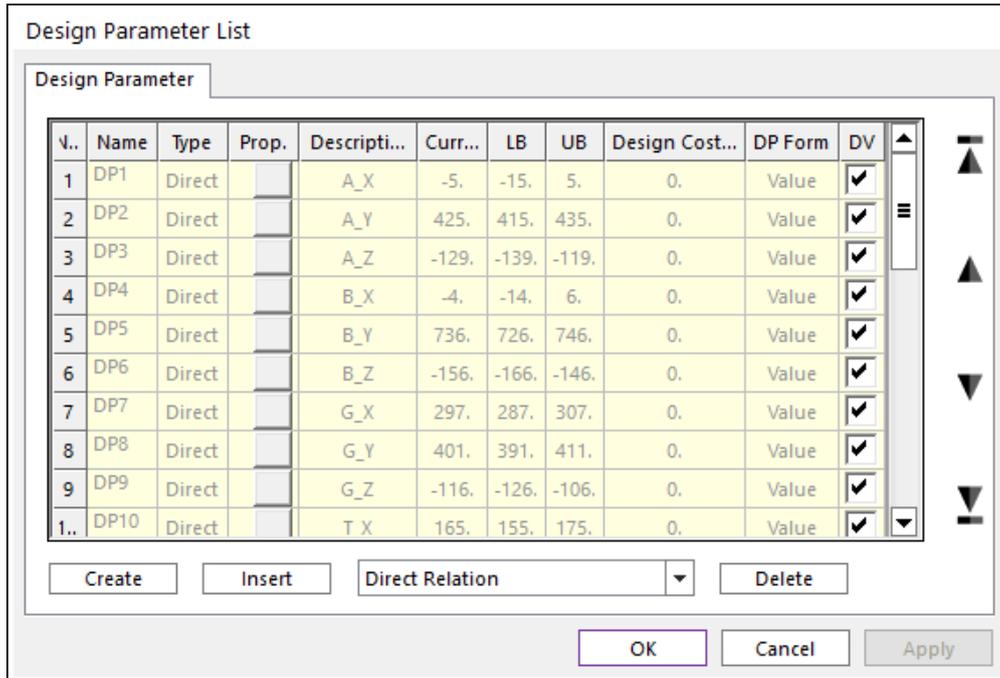


所有上述运动关系，在模型 SAMPLE_D0.rdyn 文件中都已经定义。



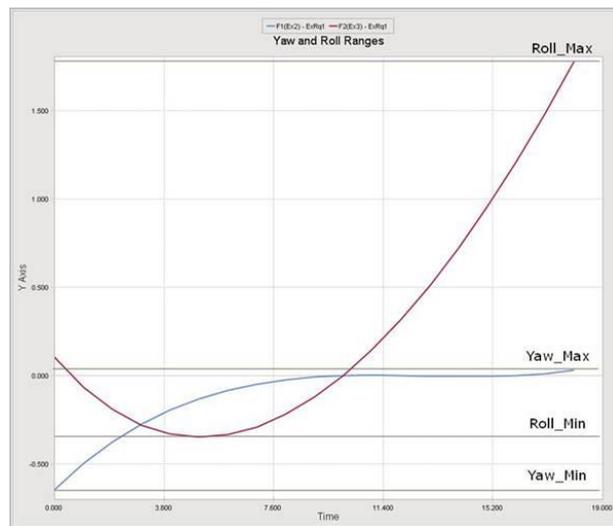
下一步，选择 **Design Parameter** 菜单。

然后，可以看到与设计参数相关的 27 个参数值。



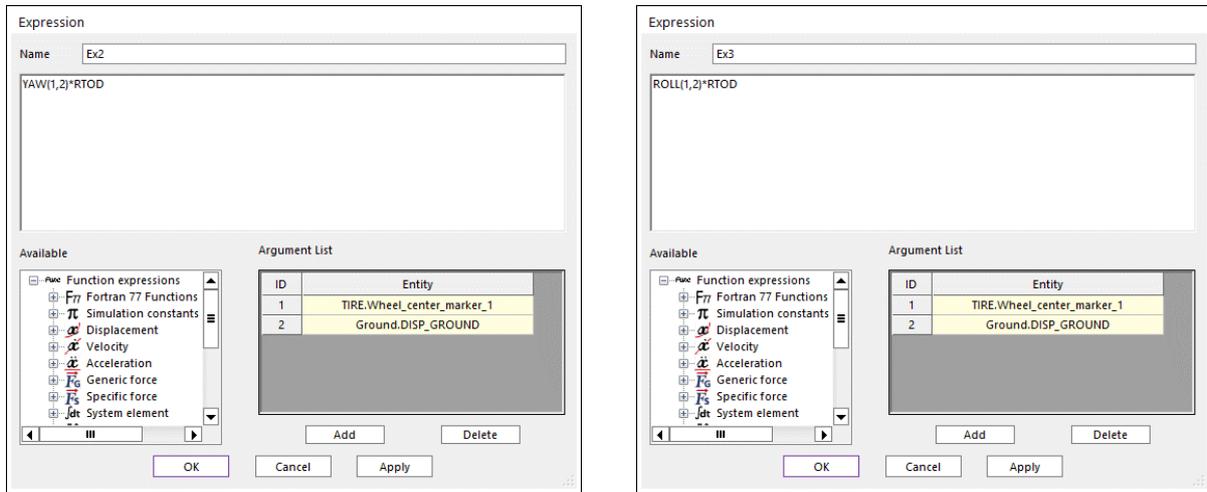
定义 performance index

接下来考虑 **performance indexes**。我们将在轮胎的垂直方向上减小 **Yaw** 和 **Roll** 范围。**RecurDyn** 不能直接提供这些值，因此通过使用 **Expression** 和 **Variable Equation** 来进行计算。

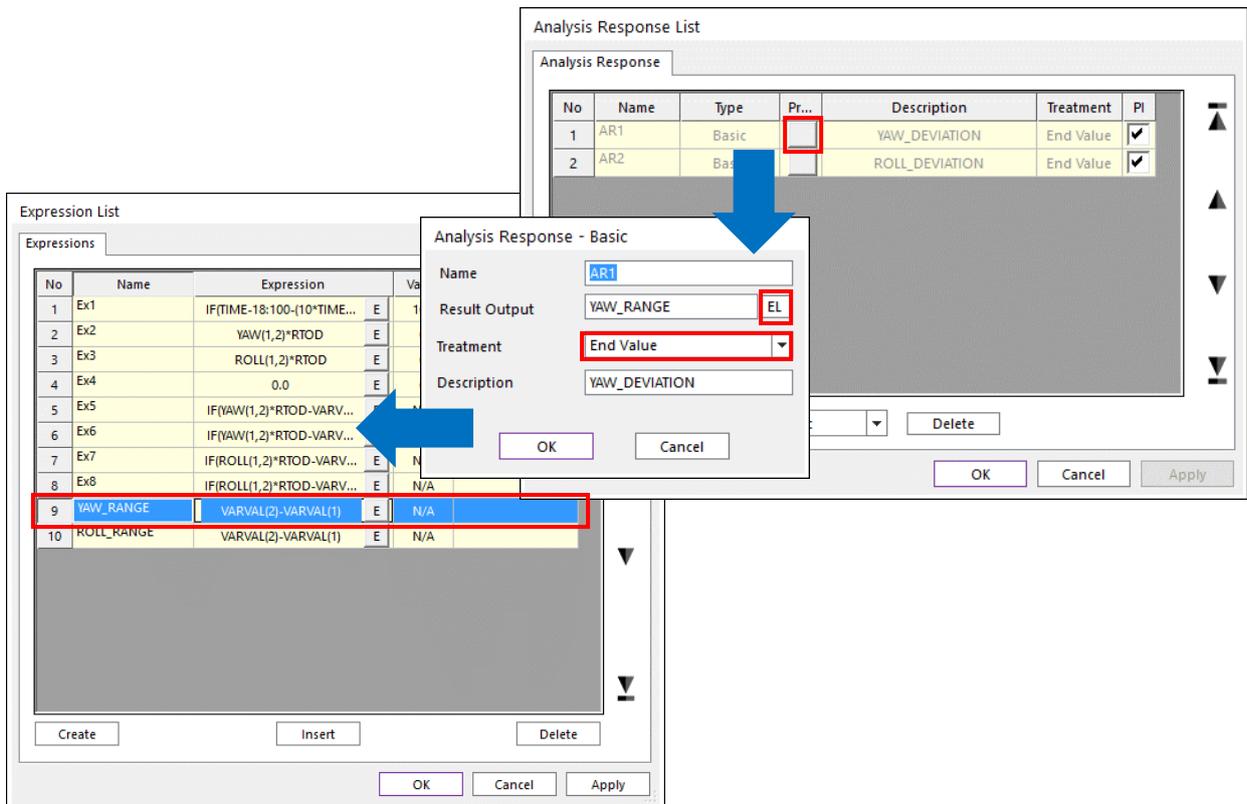


为了评估他们，我们可以计算 **Yaw** 和 **Roll** 角的最小值和最大值。这样，最大值和最小值之间的差就是它们的变化范围。

- **Step 1:** 创建 Yaw 和 Roll angles 的表达式。下图显示了保存 Yaw 和 Roll angles 的表达式。



- **Step 2:** 通过使用表达式，定义 ‘YAW_DEVIATION’ 和 ‘ROLL_DEVIATION’。然后，在 AutoDesign 中，可以定义它们作为响应进行分析。下图显示了这个过程。



运行设计优化

选择 **Design Optimization**，在 **DV** 中勾选的所有设计变量就会包含在设计中。**performance index** 是空的。然后，增加 ARs 到 **performance index**。

优化问题是要减小 **Yaw Range** 和 **Roll Range**，同时要满足：这两个角度的变化范围小于当前设计中的值。同时，我们认为 **Yaw Range** 比 **Roll Range** 更重要。因此，**Yaw** 的比重设置为 **Roll** 的两倍。

Minimize Yaw_Deviation*2 &Roll_Deviation

满足

$$\text{Yaw_Deviation} \leq 0.67$$

$$\text{Roll_Deviation} \leq 2.12,$$

0.67 和 2.12 是当前设计的范围值。这两个不等式约束似乎是多余的，因为两个目标本来就应该减小，但又无法保证。因此，解决多目标优化问题是有困难的。在 **Part II** 手册：**AutoDesign** 的指导手册，第 5 章解释了多目标优化。

从多目标优化过程的角度来考虑上述设计问题。如果有需要，我们指定 $f_1 = \text{Yaw_Deviation}$ 和 $f_2 = \text{Roll_Deviation}$ 。若初始样本构造了近似函数 $f_1(\mathbf{x})$ 和 $f_2(\mathbf{x})$ ，则称之为基本模型。 f_1^* 和 f_2^* 是初始样本的最小值。

$$\text{Min max} \left\{ \left(\frac{f_1(\mathbf{x}) - f_1^*}{f_1^*} \right), \left(\frac{f_2(\mathbf{x}) - f_2^*}{f_2^*} \right) \right\}$$

因此，多目标优化的质量取决于初始样本的数量。不幸的是，我们的问题有 27 个设计变量，这需要非常多的采样点。即使我们减小初始样本的数量，这两个不等式约束仍旧可以避免不成熟收敛。

解决多目标优化问题



1. 点击 **Performance Index** 标签。然后，定义上面的优化公式，如下所示：

Design Optimization									
Design Variable		Performance Index		Optimization Control		Result Sheet		Summary Sheet	
PI	Use	AR	Description	Definition	Goal	Weight/Limit Value			
1	<input checked="" type="checkbox"/>	AR1	YAW_DEVIATION	Objective	MIN	2.			
2	<input checked="" type="checkbox"/>	AR2	ROLL_DEVIATION	Objective	MIN	1.			
3	<input checked="" type="checkbox"/>	AR1	YAW_DEVIATION	Constraint	LE	0.67			
4	<input checked="" type="checkbox"/>	AR2	ROLL_DEVIATION	Constraint	LE	2.12			

2. 点击 **Optimization Control** 标签。直接用默认值。然后点击 **Execution** 按钮。

Design Optimization									
Design Variable		Performance Index		Optimization Control		Result Sheet		Summary Sheet	
DOE Meta Modeling Methods				Methods					
Convergence Tolerance									
Objective Change Rate in Consecutive Iterations				5.e-02					
Equality Constraints				1.e-03					
Inequality Constraints				1.e-03					
Maximum Iteration of SAO				30.					
Convergence Relaxation Control				OFF					
Simulation Type				Dynamic/Kinematic					
<input checked="" type="checkbox"/> Save Results				All_Variables\		Number of Trials		33	
Analysis Setting					Execution				
				OK		Cancel		Apply	

然后，可以看到设计公式的概览。查看 **Design Variables**，**Performance Index** 和 **Meta-Model information**。如果所有信息都正确，点击 **OK** 按钮。开始优化过程。

Execution

Summary for Execution

Design Variables

No	DV	Description	Current	LB	UB	Type	Value
1	DP1	A_X	-5.	-15.	5.	Variable	0.
2	DP2	A_Y	425.	415.	435.	Variable	0.
3	DP3	A_Z	-129.	-139.	-119.	Variable	0.
4	DP4	B_X	-4.	-14.	6.	Variable	0.
5	DP5	B_Y	736.	726.	746.	Variable	0.

Performance Indexes

No	AR	Description	Definition	Goal	Weight/Limit Value
1	AR1	YAW_DEVIATION	Objective	MIN	2.
2	AR2	ROLL_DEVIATION	Objective	MIN	1.
3	AR1	YAW_DEVIATION	Constraint	LE	0.67
4	AR2	ROLL_DEVIATION	Constraint	LE	2.12

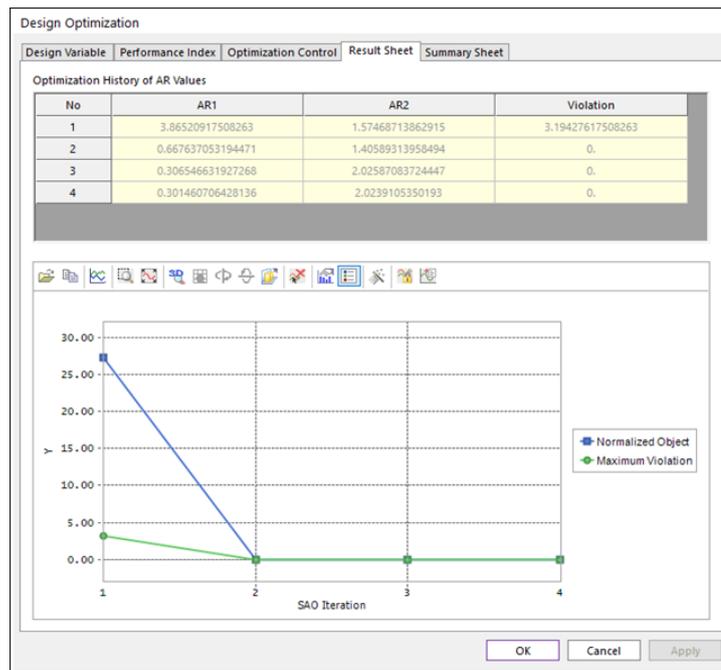
Meta - Model

Initial DOE Method	Incomplete Small Composite Design -2
Meta-Model Method	Radial Basis Functions Model(Multi-Quadratic)
Polynomial Type	Auto
Trial No	33

OK Cancel

3. 优化过程结束后，**Result Sheet Tab** 会自动显示。优化过程仅在 4 个迭代步便收敛。因此，**AutoDesign** 解决了有 27 个设计变量的悬挂系统设计，共 37 次分析，包括 33 次初始采样点的分析。最后设计给出了 **AR1=0.667** 和 **AR2=1.405**，并可以减小 0.3 % 的 **Yaw deviation**，33.7 % 的 **Roll deviation**。

收敛历程



■ 汇总表

Design Optimization

Design Variable | Performance Index | Optimization Control | Result Sheet | Summary Sheet

Design Variables

No	Name	Description	Optimum	Current	LB	UB
1	DP1	A_X	5.	-5.	-15.	5.
2	DP2	A_Y	415.	425.	415.	435.
3	DP3	A_Z	-139.	-129.	-139.	-119.

Analysis Responses

No	Name	Description	Optimum
1	AR1	YAW_DEVIATION	0.667637053194471
2	AR2	ROLL_DEVIATION	1.40589313958494

Performance Indexes

No	AR	Description	Definition	Goal	Weight/Limit Value
1	AR1	YAW_DEVIATION	Objective	MIN	2.
2	AR2	ROLL_DEVIATION	Objective	MIN	1.
3	AR1	YAW_DEVIATION	Constraint	LE	0.67
4	AR2	ROLL DEVIATION	Constraint	LE	2.12

SAO

Initial DOE Method: Incomplete Small Composite Design -2

Meta - Model: Radial Basis Functions Model(Multi-Quadratic) | Polynomial Type: Auto

Initial Sample Runs: 33 | SAO: 4(0) | Total Evaluations: 37

Optimal Design: E:\SVN\GT\Trunk\AddFile\Tutorial\AutoDesign\SuspensionSystem\All_Variables_DO_004

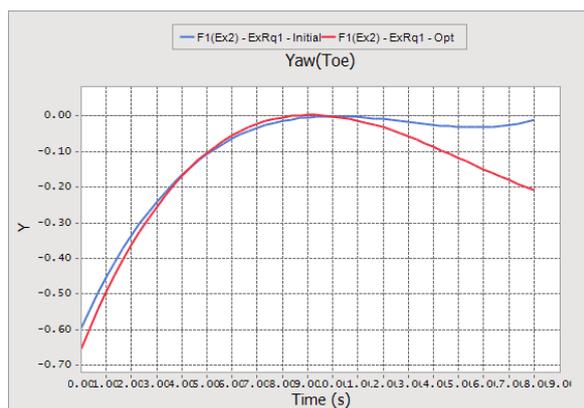
Create New Optimum Model | Apply to Current Model | Open Summary file

OK | Cancel | Apply

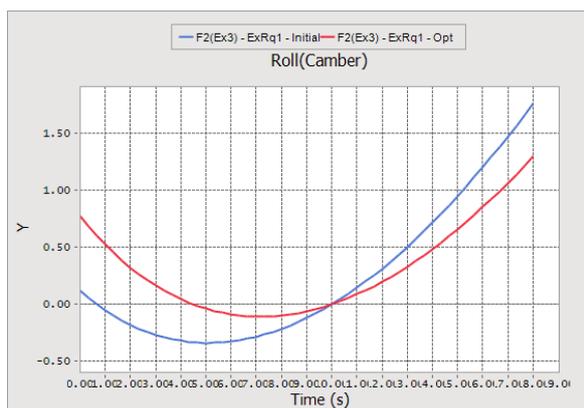
在上面最新提供的汇总表中，在设计变量和分析响应表中，有概述的优化结果。同时，概述的 SAO 信息显示 SAO 需要 4 个迭代步。因此，总的计算次数是 37。最优化设计是 DO_002。

- 最后，比较原始设计和最后设计的 yaw and roll ranges。DO_002 是最后设计。同时，DOE33 是原始设计。下图显示了比较后的结果。当 ISCD-1 和 ISCD-2 组成了初始样本，初始样本的最后一个是当前设计。下图中，红线是优化结果。

Yaw (Toe)



Roll (Camber)



Chapter

5

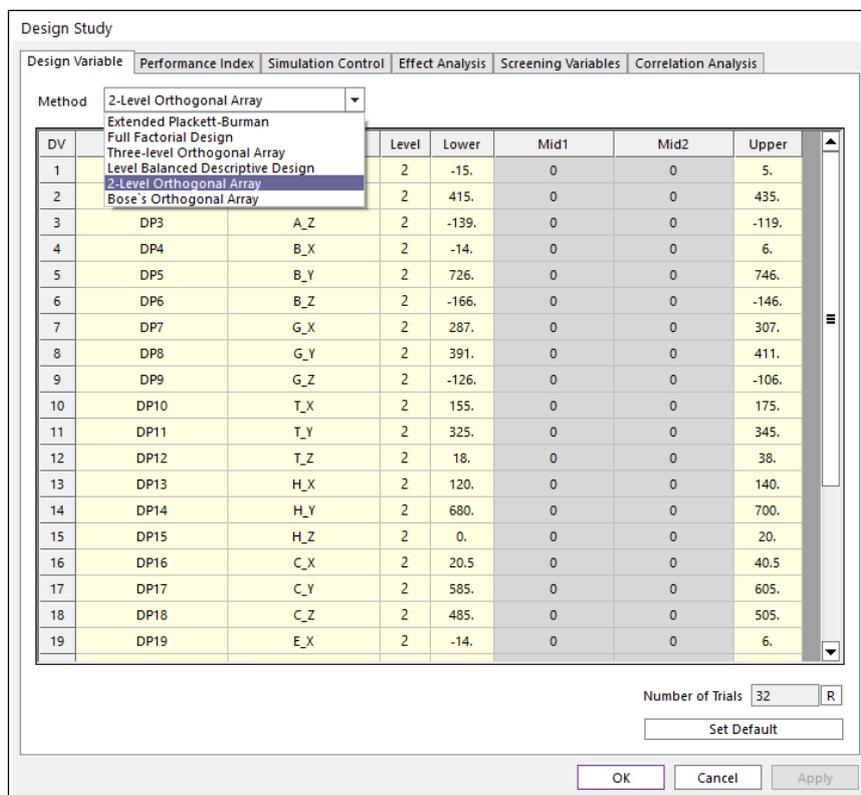
筛选变量的设计优化

现在，对所有的设计变量和筛选设计变量，比较优化的结果。因此，保存模型文件 **Sample_D0.rdyn** 为 **Sample_D1.rdyn**。删除之前所有的仿真结果。

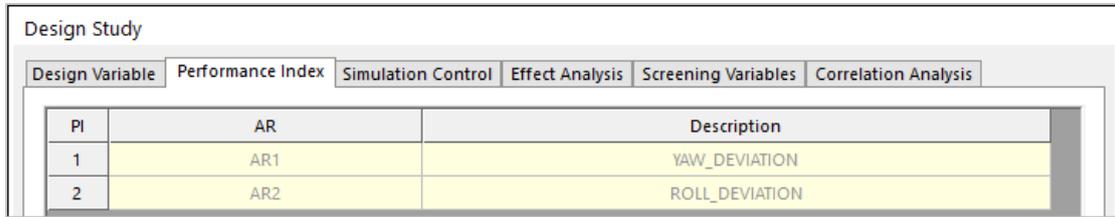
首先，我们筛选设计变量。筛选是基于有效的分析结果，选择 **AutoDesign**。下述步骤说明筛选过程：



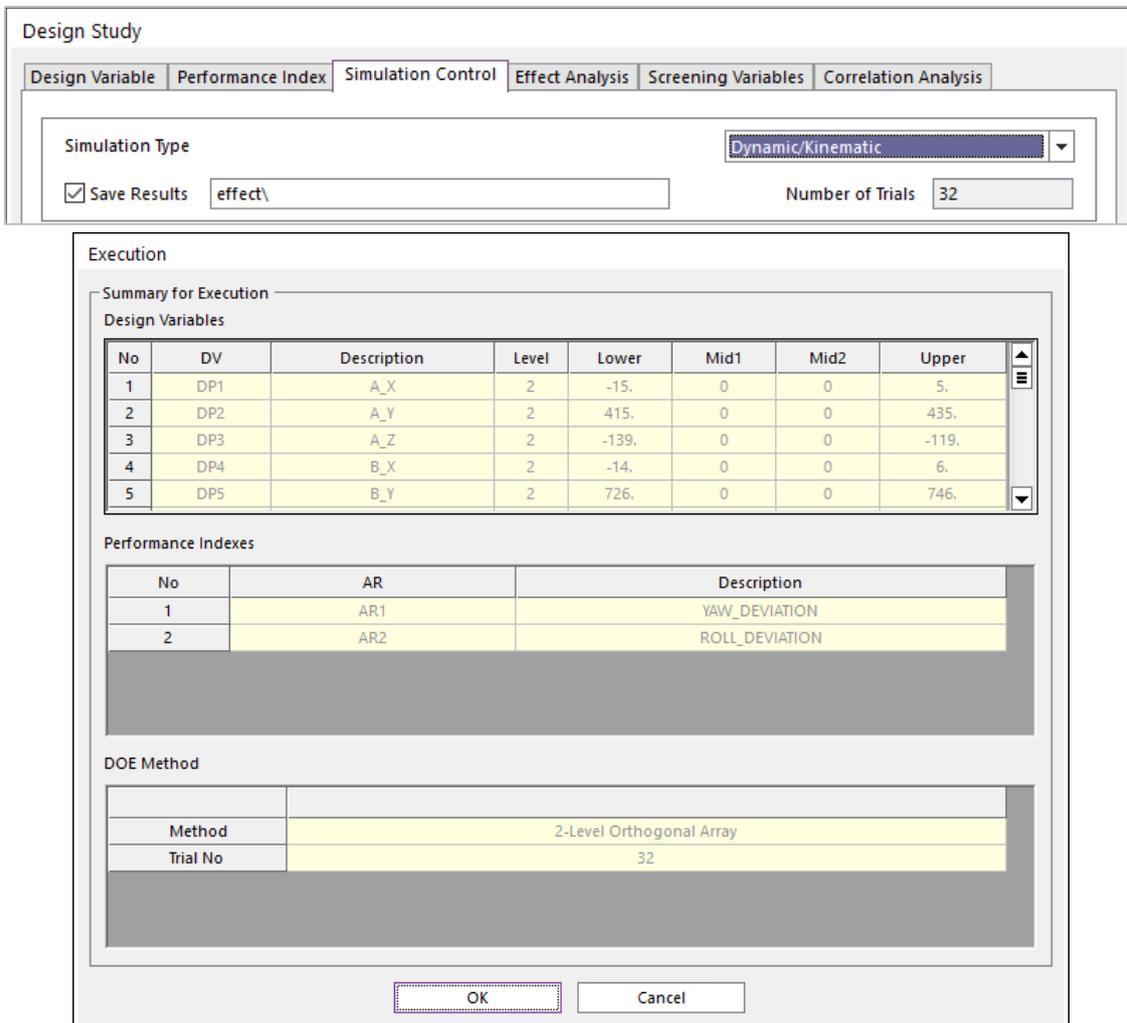
1. 输入 **Design Study** 菜单。
2. 然后，可以看到 **Design Variable** 标签中的设计变量列表。由于设计变量的个数是 27，选择 **2-Level Orthogonal Array**，减少试验次数。如果选择 **3-Level Orthogonal Array**，试验次数为 81。



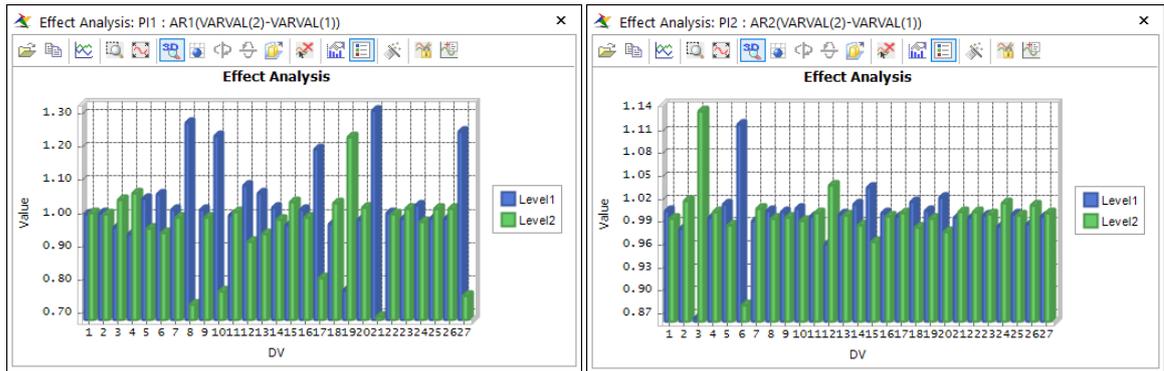
- 下一步，在 **Performance Index** 标签下，有 **AR1** 和 **AR2** 两个分析响应出现。如果没有，可以从 **Sample_D0.rdyn** 中，重新创建 **Sample_D1.rdyn**。



- 下一步，点击 **Simulation Control** 标签。如果保存每次试验的分析结果，勾选 **Save Results**，输入 **Folder Name**。然后，点击 **Execution** 按钮。

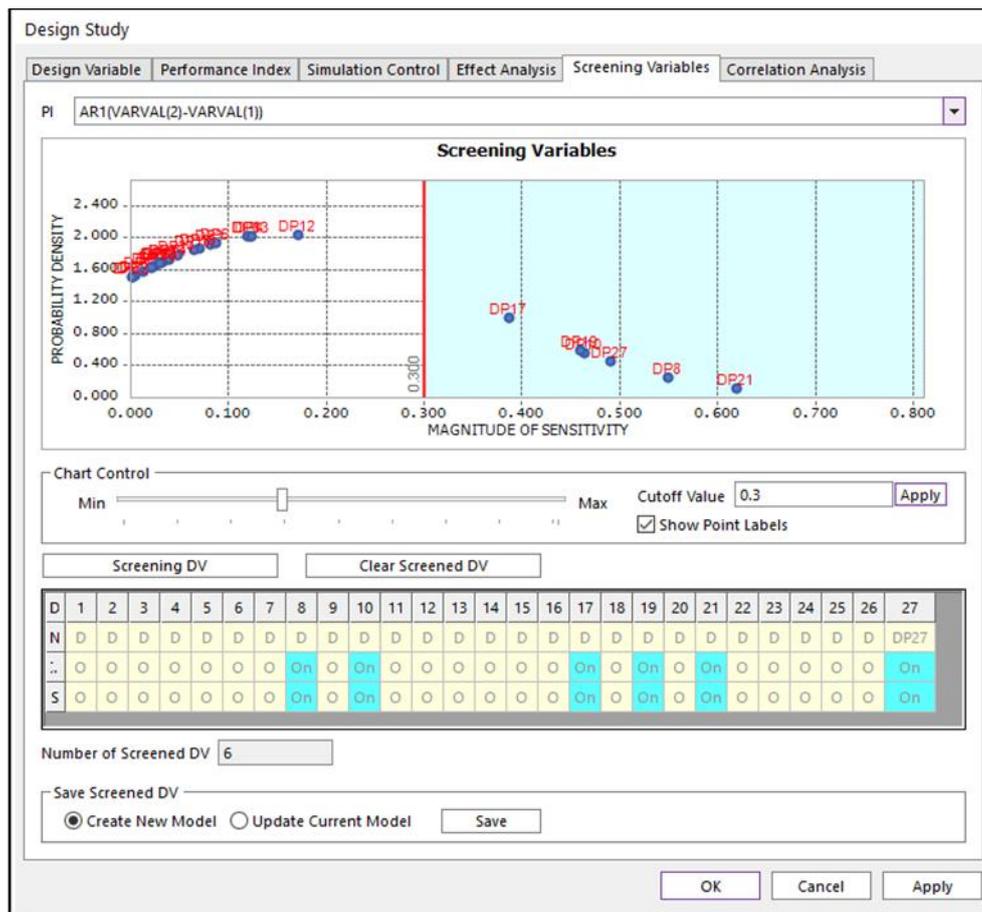


- 完成所有分析后，会激活 **Design Study** 窗口。点击 **Effect Analysis** 标签。然后，可以通过勾选 **Effect Values** 列中的复选框，点击 **Draw** 按钮，查看分析图表。有效的分析结果如下所示：



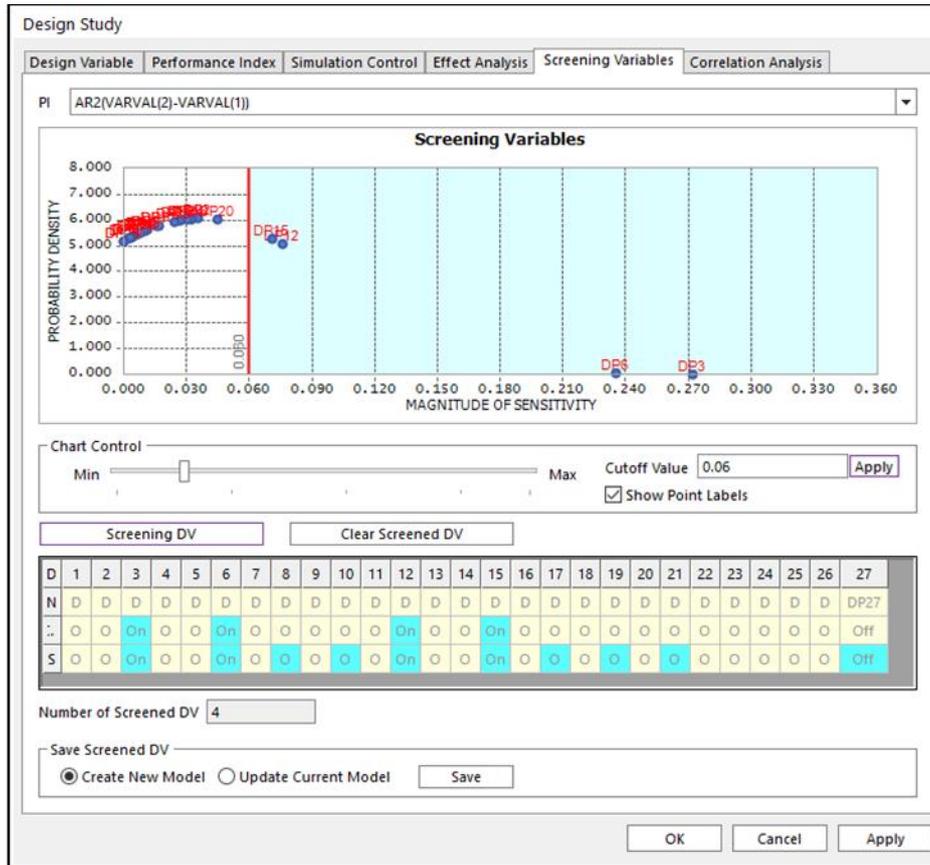
两张图显示：不同变量对 **PI₁** 和 **PI₂** 是敏感的。一般，用户从上面的图表中选择敏感的参数。但是这并不简单。因此，**AutoDesign** 提供了筛选变量的统计学指导。

6. 点击 **Screening Variables** 标签。选择 **PI** 的 **AR₁**。然后，可以看到散布点。右边的位置点比左边的点更敏感。移动控制器，划分设计变量组。然后 **Cutoff Value** 是 0.3，并点击 **Screening DV** 按钮。



然后，红线将散布点分为两组。右边的组是剩余变量，筛选后的变量标记为 **On**，其它标记为 **Off**。

下一步，选择 **PI** 的 **AR2**，并进行类似的同样处理，除了定义 **Cutoff Value** 是 0.6。

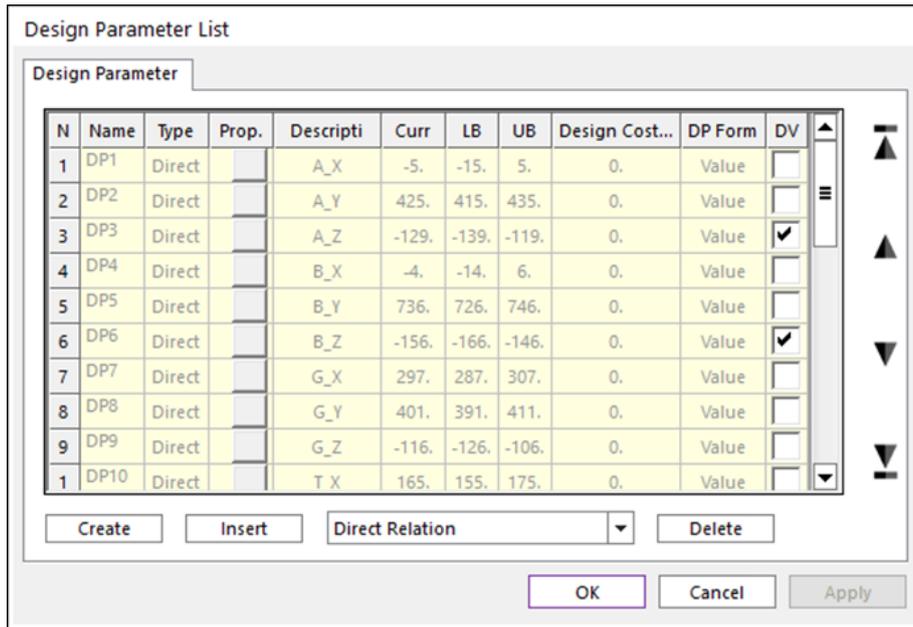


筛选 **DV**，给出：有效的设计变量 (**DV**) 是 **DP3**, **DP6**, **DP12** 和 **DP15**。

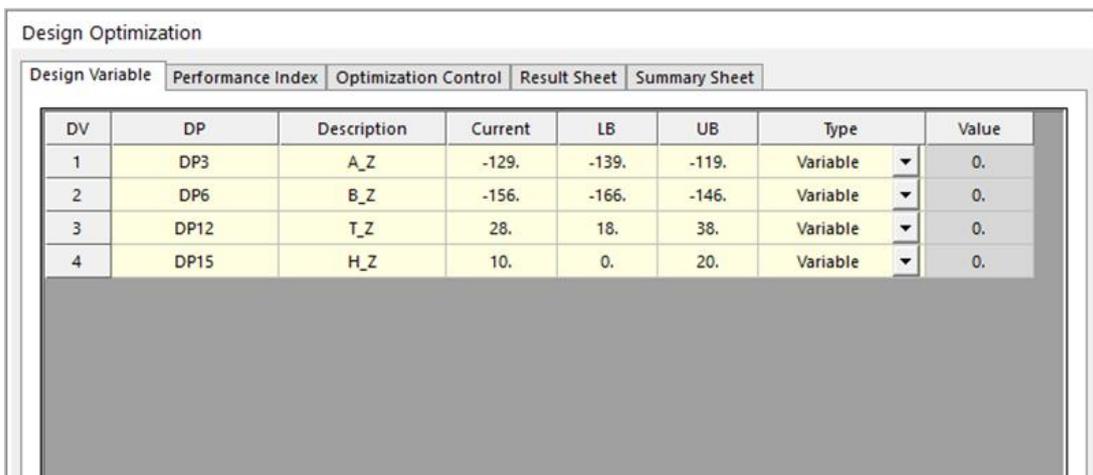
现在，通过勾选 **Create New Model** 复选框，点击 **Save** 按钮，来创建新模型。新模型命名为 **Sample_D2.rdyn**。



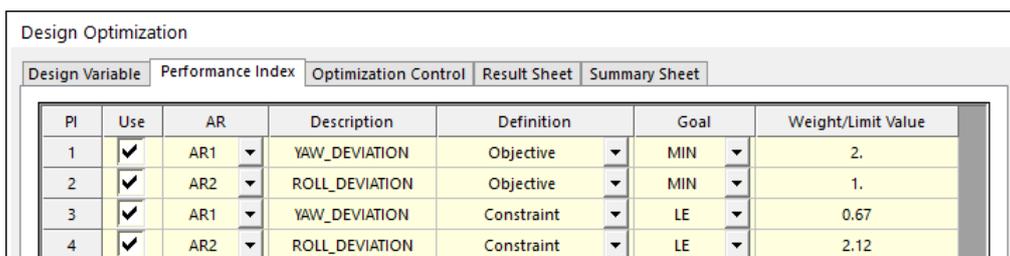
- 现在，可以看到模型名称改为 **Sample_D2.rdyn**。点击 **AutoDesign** 的 **Design Parameter** 图标。不同于 **Sample_D1.rdyn** 模型，在 **DV** 列，只选中筛选的变量。这表示 **DV1=DP3**, **DV2=DP6**, **DV3=DP12** 和 **DV4=DP15**。大多数的有效设计参数是 **Z** 轴的值。



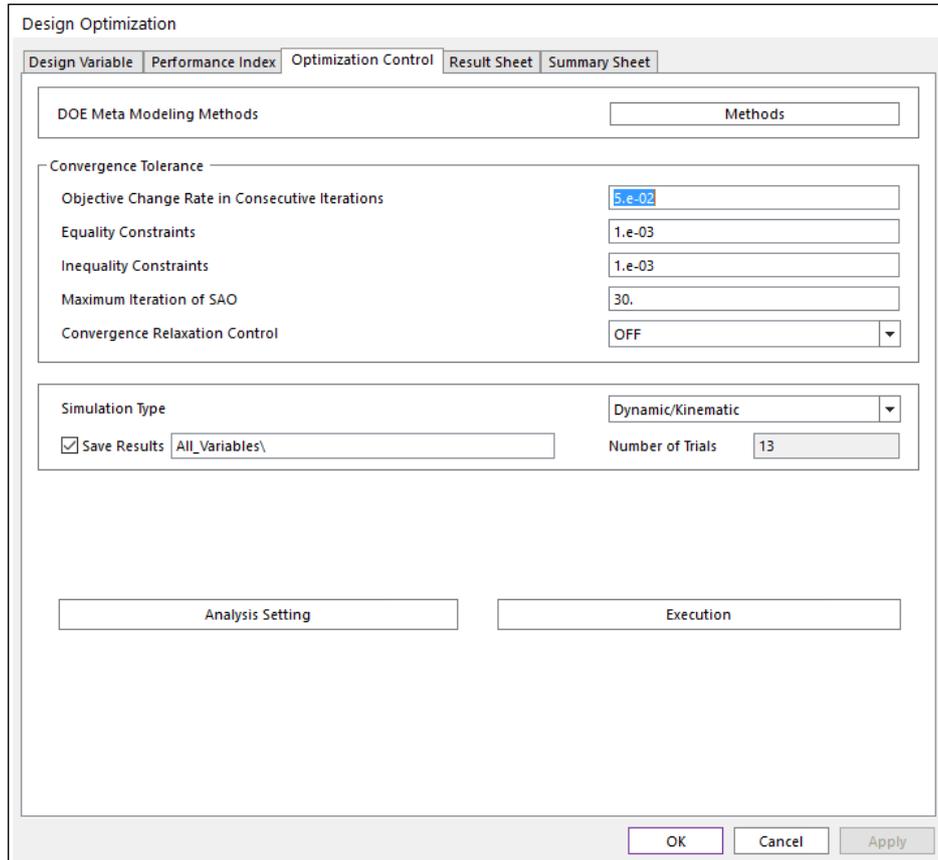
8. 下一步，点击 **Design Optimization** 图标。然后，筛选后的变量如下图所示：



9. 点击 **Performance Index** 标签。优化公式和 **Sample_D1.rdyn** 的一样。这里采用相同的公式。

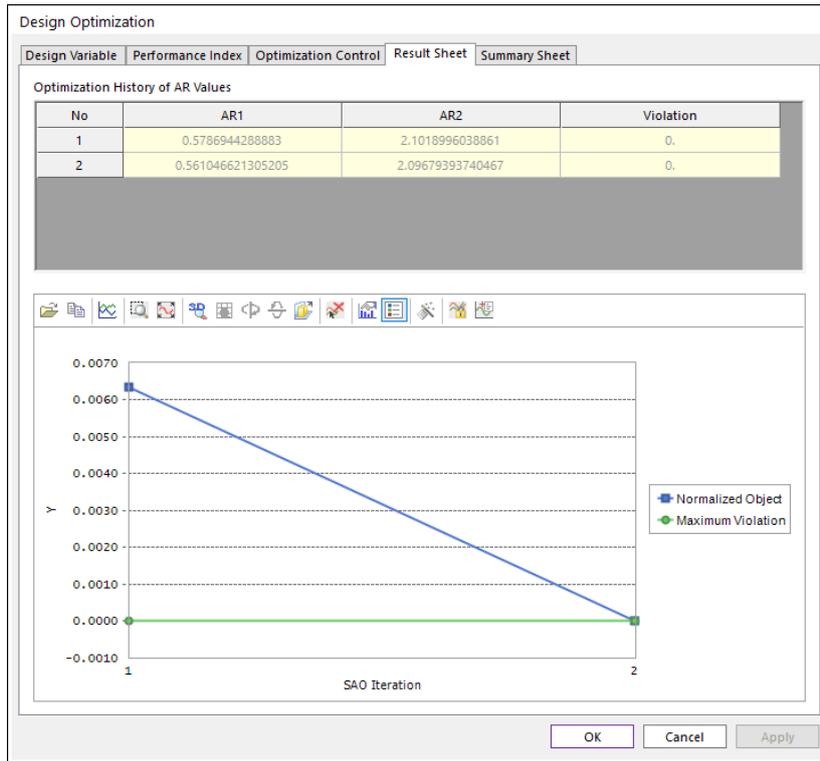


10. 在 **Optimization Control** 标签，所有的收敛容差用默认值。结果文件保存在 **Screening_Variables** 文件夹。



11. 优化完成后，我们看下结果表。**AutoDesign** 在 2 个迭代步内收敛。优化后的 **AR1** 和 **AR2** 分别是 0.609 和 2.084。**AR1** 和 **AR2**，与未进行筛选的设计相似。

■ 收敛历程



不同于没有筛选的设计的收敛性。上面的收敛性有所下降。比较两次优化问题的结果。

	筛选	不筛选
变量数	4	27
初始 samples	9	33
运行的 SAO 数量	2	4
筛选的 Samples	32	—
优化响应	0.609, 2.084	0.656, 1.405

感谢参与本教程学习!