

HyperStudy Altair Engineering September 4th, 2009

Agenda



Morning Session (8:30 AM - 12:30 PM)

- 1. Introduction to HyperStudy
- 2. Exercises 1.1 and 1.2
- 3. BREAK (10 Minutes)
- 4. Design of Experiments (DOE)
- 5. Exercises 2.1, 2.2 and 2.3
- 6. Approximations
- 7. Exercises 3.1 and 3.2

Afternoon Session (1:30 - 4:30 PM)

- 8. Optimization
- 9. Exercises 4.1, 4.2 and (one of these 4.3, 4.4 or 4.5)
- 10. BREAK (10 Minutes)
- 11. Stochastic
- 12. Exercises 5.1 and 5.2



Optimization is Driver in CAE Driven Design Process









Chapter 1: Introduction to HyperStudy

Altair HyperStudy



- is a software to perform Design of Experiments (DOE), approximations, optimization, and stochastic studies.
- is applicable to study the different aspects of a design under various conditions, including non-linear behaviors.
- can be applied in the multi-disciplinary optimization of a design combining different analysis types.
- is integrated with HyperWorks through direct links to the models in HyperMesh (.hm), HyperForm (.hf), and MotionView (.mdl).
- models can be easily parameterized. Aside from the typical definition of solver input data as design variables, the shape of a finite element model can also be parameterized with ease through HyperMorph.

Study Definition



	DOE	Approximation	Optimization	Stochastic
Parameters Screening System Performance Study	1 1			
Response Surface Evaluation		€}	A	
Optimum Design				Å
Robust Design	5	A	S	
Reliability Design	S	S	Ð	S



- User defined
- External Matrix

Approximations

Responses Approximation



- Least square
- Moving least square
- HyperKriging



HyperWorks



Stochastic Studies

Define input distributions





- Normal
- Uniform
- Triangular
- Exponential
- Weibull

whi when	
Conve legend	Art+8 Hangan/905/005
Orade nodels	
Could Divis	64.9
Do noninal sun	
Cestempores Yash:	
C entrely	
Oracle DOE study	01.0
Contrated Handbell	-
Coulded viewhere	
Lorenseo secoros	
Uncontrolled Interactions	and the second
Concerning and the states	- 1700000777
Charman	1112/2008/00/2012
With Support and	
Eduatorspores	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Post processing	11555558888855511
Create approximations	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
ptimization whady	
Ceale optimization study	
Osfee-design valables	#2
Coutains	
Objectives	
Pol pocesing	
ICRAWK WUDY	
Celle Porvanc INOP	•
Contract of Contra	
Contract contractory	
- Random variable (Vilore)	
Post any amount Factors	1 120 100 104 212 24 200 200 304 382
pros.	With
1 C	
	Vole Execute Exhant VolenExecuteExhant
/	- ma - ma
	- SBAS

- Simple Random
- Latin HyperCube
- Hammersley



- Call solver directly
- Using response surface



Evaluate

results

Altair[®] HyperWorks[®]

- Probability Distribution Function (PDF)
- Cumulative Distribution Function (CDF)
- Histogram Distribution
- Ant-Hill plots
- Statistical moments



HyperStudy Process Flow



Altair Proprietary and Confidential Information

Copyright © 2008 Altair Engineering, Inc. All rights reserved.

🕺 Untitled - HyperStudy v10.0					
File View Tools Utilities Help					
New Study Open Study Close Study	Ctrl+N				
Save Current Study Save Current Study As Save Current Study As (With Batch Con Save Output Files	Ctrl+S	Varname	Study note:	<u>_</u>	
Set Preference File	Set Preferenc	e File		↓	?)
Exit		nk in: 🗀 10 HST SC	DLVERS	▼ ← 🗈 💣 🖛	
Select responses Write/Execute runs Extract responses Post processing Oreate approximation Input matrix Validation matrix Build approximation Build approximation Residuals Trade-off Anova Create optimization study Create optimization study Define design variables Objectives Post processing Stochastic study Create optimization study Define design variables	Document Desktop My Docume My Comput My Networ Places	s Preference2.	mvw.		
Define correlation Select responses		File name:	preference2 mww	-	Onen

Altair Proprietary and Confidential Information

Copyright © 2008 Altair Engineering, Inc. All rights reserved.

Old Editor



Altair^{*} HyperWorks*

New Editor



Job Management



Solver Script



Altair^{*} HyperWorks

Parametrization of a FEA model



The process of selecting design variables / factors / random variables for

Optimization / DOE / Stochastic studies can be summarized as:





Direct linking to Hypermesh-MotionView-HyperForm provides HyperStudy direct access to simulation models and to the features such as thickness, concentrated masses, shape changes which are used as the design variables in DOE, optimization or stochastic studies .

When HyperStudy is launched in a standalone mode, the design variables need to be identified using a common method as HyperStudy interacts with solvers of different physics.

HyperStudy uses TEMPLEX to parameterize the solver input deck. The solver input deck is taken and "parameter" statements are added, which in turn is linked to the property such as thickness, concentrated masses. This file with "parameter" statements is called as "parameterized input deck". This deck forms the input file for HyperStudy.





HyperMesh-HyperStudy Link



Shapes in HyperMesh





Model Parameterization

Copyright © 2008 Altair Engineering, Inc. All rights reserved.



HyperMesh-HyperStudy Link

🔗 Study setup	Designation		5	in a state to the second state of	
Create studies	Design variables	;	Des	sign variable properties	
Create models					
🔀 Create design variables		😂 Model Parameters			
Do nominal run					
Create responses		Variable name: 4	F.1		
Link design variables			<u> </u>		
Sensitivity		Initial value:	1		
Σ <u>Â</u> DOE study		Laward and	a Apply to all calected items		
Create DOE study		Lower bound:	J.g I♥ Apply to all selected items		
Controlled variables		Lipper bound:	1 1 🔽 Apply to all selected items		
Controlled interactions					
Controlled allocations		HyperMesh Model Parameters	HyperStudy Parameters		
Uncontrolled variables		- ypenneen moderr arametere	rigporotady r diamoters		
Uncontrolled interactions			1 T 1		
Uncontrolled allocations					
Select responses			2.T.1		
Write/Execute runs		2 1 1	3.T.1		
Extract responses		3 1 1	4 T 1		
Post processing		4.11			
Approximation		E-ControiCards			
Create approximation					
Input matrix					
Validation matrix					
Build approximation					
Residuals					
Trade-off					
Anova					
Optimization study					
Create optimization study					
Define design variables					
Constraints					
Stochastic study					
Create stochastic study					
Define rendom verishes			UK Cancel		
MiteEvecute rups	Add Madel Davey	Add Design Marietyle	Delete Design Verieble		
	📕 Add Model Parar	neter Add Design Variable	Delete Design Variable		

Create Design Veriables (Compart Chudu - Chudu - 1)



Exercise 1.1: Create Design Variables Through HyperMesh

Parametrization Using Create Template GUI





Parametrization using Create Template GUI



A parameterized solver input deck is required when HyperStudy is launched in standalone mode.





Templex is a general-purpose text and numeric processor. Templex generates output text based on guidelines defined in a template file.

The input files can be created using any text editor or word processor that can save a file in ASCII format.

Templex solver can be used to process the template file and the output can be sent to a monitor, to another file, HyperGraph or to other output streams. **Templex** is built into HyperStudy as an expression builder for creating responses for Objectives and Constraints.

Templex can also be used as a standalone solver on mathematical expressions to perform Design of Experiments (DOE) / Optimization / Stochastic studies.

Important Templex Rules



- Statements, expressions, and variables must be placed between braces, <u>{}</u>, so that *Templex* will evaluate them. Otherwise they are treated as plain text and not processed.
- After *Templex* processes a parameterized template file, the plain text is copied to the specified output stream verbatim.
- When *Templex* encounters an expression within braces, it evaluates the expression and sends the resulting value to the output stream. If the item between the braces is a *Templex statement*, Templex carries out the statement instructions. If the Templex statement generates output, the resulting value is sent to the output stream too.
- Variable names can be up to <u>eighty characters long</u>. They consist only of letters, numbers and underscores; no other characters are allowed. Variable names must always <u>begin with a letter</u> and are <u>case sensitive</u>.
- In addition to scalars *Templex* supports <u>one- and two- dimensional arrays</u>. In *HyperStudy* to define the responses the results from the analysis runs are represented as <u>one-dimensional arrays</u> "VECTORS". The index starts with 0 as in C language.
- The *template file* must have <u>at least one parameter statement</u> to be able to read into the HyperStudy.



The parameterized input deck is a **Templex template**, *.tpl, file.

To create the *.tpl file, the following are added to the analysis input file :

- 1. Templex parameter statements,
- 2. Field variable calls, and
- 3. External-file-include statements





1. Templex parameter statements are used for individual field changes and shape vector (DESVAR) nodal perturbations. All parameter statements are placed at the top of the *.tpl file.

The templex parameter function has the following format:

{parameter(varname, label, nominal, minimum, maximum)}

Example:

```
{parameter(TH1, "Thickness 1", 1.6, 0.5, 3.5)}
```

Create Template GUI



2. Field variable statements must be inserted into the analysis deck for each field to be optimized, such as component thickness, density, etc.

The formats for field variable statements is as follows:

{variable,%field_width.decimal_places}

'Variable' is the same as in the corresponding parameter statement.

The 'field width' value depends on the solver being used so that the proper spacing is maintained in the analysis deck. Each solver will have it's own filed width: LS-DYNA is 10, OptiStruct is 8, and HyperForm is 8. The 'decimal_places' value is a user preference.

Example

{TH1, %8.5f}

Example: LS_Dyna Input Deck



Original LS-DYNA Deck

1.6	1.6	1.6	1.6	0.0		
	1	71.0		20.0	0.0	0
\$HMNAME	PROPS	1beam				

Parameterized LS-DYNA Deck (.tpl)

\$HMNAME	PROPS	1beam					
1	L	71.0	20.0	0.0	0		
{TH1,%10.5f}{TH1,%10.5f}{TH1,%10.5f}{TH1,%10.5f}0.0							

3. If the design variables in optimization are shape vectors, further editing of the deck is necessary such as external-file-include statements.

Example: Nastran Input Deck



HyperStudy offers a simple GUI to parameterize input decks.

🚳 Untitled - HyperStudy v9.0b121										<u> </u>	
Hie View Tools Utilities Help											
Job Management											
🖻 💁 Study Readers, Writers and Functions											
Register Solver Script		Create	Tomolata		ñ e i	T 1.					
🔲 cre Register Utility	Varnam	Create	remptate		Creat	e Templat	9				
Do Register Optimizer	<u> </u>	<u>F</u> ile <u>E</u> dit	<u>T</u> emplex <u>F</u>	Parameters	<u>F</u> ile <u>E</u> di	t <u>T</u> emplex	<u>P</u> arameters				
Link design variables	Г	638 3	50.500000	000000 -4	63	50.5000	Create D	esign Variable	30500	000	
Sensitivity Σ Σ ΩDE study		639 3	50.500000	000000 -3	63	9 50.5000	Shape Te	emplate	00000	000	
Create DOE study	reate Template	640 \$	50.500000	000000 -3	64(50.5000	at a la po		30500	000	
Controlled variables	Edit Templey Parameters	641 9	50.500000	000000 -2	64:	L 50.5000	steel 20	0000.0, 180000.0	, 220000.0 × 100000	000	
Controlled interactions		642 3	50.500000	000000 -2	64	2 50.50000	- 00000000 -	20.500000000	00 235.000030500	000	
Controlled allocations		643 3	50.500000	000000 -1	64	50.50000	- 00000000 -	10.500000000	00 227.500000000	000	
Uncontrolled interactions	xport xport As	645	50.500000	000000 -0	649	5 50 50000	- 00000000 -	0.50000000000	00 233.000030300	000	
Uncontrolled allocations	vi+	646 5	50.500000	000000 -0	64	5 50.50000	00000000 -	0.500000000000	00 235.000030500	000	
Select responses		647 (60.750011	440000 -0	64	7 60.7500	11440000 -	0.5000000000	00 227.50000000	000	
Virite/Execute runs		648 (60.750011	440000 -0	64	60.7500	11440000 -	0.5000000000	00 235.000030500	000	
Post processing	Open	649	71.000000	000000 -0	64	9 71.0000	0000000 -	0.5000000000	00 227.500000000	000	
🖃 📚 Approximation		650	71.000000	000000 -0	65	71.0000	00000000 -	0.5000000000	Design Va	riable Pr	
Create approximation	Look in: 🗁 Dyna	651 5	50.500000	000000 50	65:	L 50.5000	00000000 5	0.5000000000	Sesign ra		
Velidetion metrix	*1	MAT_PLAS	TIC_KINEM	ATIC	*MAT_PL	ASTIC_KINE	EMATIC	-	Manag	at a st	
Build approximation	boxbeam.bdf	HMNAME MA	ATS	1Materia.	ŞHMNAME	MATS	1Materi	al	iname:	Isteel	
Residuals			1 /.0502-	06 210.000	0.0000	1 7.850	2-06 210.0	0000 0.30000			
Trade-off	*1	MAT PIFCE	FWISE LIN	FAR PLAST	*MAT DT	CENTSE I	ייייייייייייייייייייייייייייייייייייי	TICITY	Label:	stee	
Anova	si	HMNAME M	ATS	4120ksi.	SHMNAME	MATS	4120kai	steel			
Create optimization study			47.860E-0	9 200000.	******	47.860E-	-09 200000	.0 0.28	Initial value:	200000.0	
Define design variables		0.0	0.0		0.0	0.0		3 0			
Constraints	Ş				Ş				Lower bound:	180000.0	
Objectives	*]	PART			*PART						
Stochastic study	\$]	HMNAME CO	OMPS	11	\$HMNAME	COMPS	11		Upper bound:	220000.0	
Create stochastic study	File <u>n</u> ame: box ^{\$1}	HMCOLOR (COMPS	1	\$HMCOLO	R COMPS	1	7			
Define random variables									Format:	%8.5f	
Define correlation	Files of type: All I	DADT	L	1	+DADT	1	1	1		,	
Write/Execute runs	ś	HMNAME CO	OMPS	42	CHMNAME	COMPS	42			01	Canaal
Post processing	si	HMCOLOR (COMPS	4	SHMCOLO	COMPS	42	10		UK	
					+						
		4	4	2		4	2	1	0 0	0	0
	*]	PART			*PART						
Launches input deck parameterization script.	\$]	HMNAME CO	OMPS	53	\$HMNAME	COMPS	53				
	\$1	HMCOLOR (COMPS	5	\$HMCOLO	R COMPS	5	13			
Altair Proprietary and Confidential In	nformation	PART	5	3	*PART	5	3	1	0 0	0	0

HyperWorks

Example: Nastran Input Deck

Oreate T	emplate				🖡 boxbeam.bdf.tpl - Notepad	
<u>F</u> ile <u>E</u> dit <u>T</u>	<u>T</u> emplex <u>P</u> arameters				File Edit Format View Help	
638 50	0.500000000000 -	4 <mark>0 50000000000 235 000</mark>	3050000		{parameter(steel,"steel", 200000.0, 180000.0, 220000.0)}	^
639 50	0.500000000000 -	3 🕙 Create Template				
640 50	0.50000000000000 -	2 2 File Edit Templex Para	meters		*TITLE	
642 50	0.50000000000 -				Boxbeam-Test	
643 50	0.500000000000 -	1 D000000	000 -0.50000000	000 2	<pre>\$HM_OUTPUT_DECK created 16:38:54 04-15-98 by HyperMesh Version 2.1b</pre>	
645 50	0.5000000000000000 -	Export 0000000	Save As		2.00000000 00.00000000.00000000	
646 50	0.500000000000 -	Export As INEMATI			*CONTROL_TIMESTEP	
647 60	0.750011440000 -	0 Exit 50E-06	Save in: 🗁	Dyna	0.00000000 0.90 00.00000000.00000000 0 0	0
648 60 649 71	0.750011440000 -	0.0000000 0.0000000	IT N	-	DATABASE_OPTION *DATABASE_NODOLIT	
650 7:	1.000000000000 -	<pre>o *MAT_PIECEWISE_LINEAF</pre>	boxbeam.b	df	0.1000000	
651 50	0.50000000000 5	0 \$HMNAME MATS 41			*DATABASE_GLSTAT	
*MAT_PLAST: HMNAME_MAT	IC_KINEMATIC	47.860E-09 8			0.10000000 *DATARASE SPCEAPC	
1 IIIIIII	7.850E-06 210.0	0 \$			0.1000000	
0.0000000	0.0000000 0.000	0 * PART			*DATABASE_BINARY_D3PLOT	
MAT_PIECE	WISE_LINEAR_PLAS	SHMNAME COMPS 1	•			
nminame ma: 4'	7.860E-09 steel				1 0.00000000000 50.500000000 0.000000000	
0.0 0	0.0	1 1			2 0.00000000000 50.5000000000 10.000000000	
;		*PART			3 0.00000000000 50.50000000000 20.0000000000	
*PART	MDS 11	SHMCOLOR COMPS			4 0.000000000000 50.5000000000 30.0000000000	
SHMCOLOR CO	OMPS 1		File <u>n</u> ame:	boxb	6 0.000000000000 50.5000000000 50.00000000	
		4 2	_		7 0.00000000000 50.5000000000 60.0000000000	
1	. 1	*PART SHMNAME COMPS	Save as type:	All Fi	8 0.000000000000 50.50000000000 70.0000000000	
SHMNAME CON	MPS 42	\$HMCOLOR COMPS			10 10.100000380000 50.5000000000 0.00000000000	
SHMCOLOR CO	COMPS 4				11 10.100000380000 50.50000000000 10.00000000000	
		5 3	1	0	12 10.100000380000 50.50000000000 20.0000000000	
*PART	2	\$HMNAME COMPS 6	54		13 10.10000380000 50.5000000000 30.0000000000	
SHMNAME CON	MPS 53	\$HMCOLOR COMPS	6 14		15 10.100000380000 50.50000000000 50.0000000000	
\$HMCOLOR CO	COMPS 5		1		16 10.100000380000 50.50000000000 60.00000000000	
-		*SECTION SHELL	T	U	17 10.100000380000 50.50000000000 70.0000000000 18 10 100000380000 50 50000000000 80 0000000000	
*PART	· 3	\$HMNAME PROPS 1	.1		19 20.20000760000 50.50000000000 0.0000000000	
CHMNIAME COM	MDS EI	1 2	1.0	20.	20 20.200000760000 50.50000000000 10.0000000000	
<		1.000000001.000000001		0000.	21 20.200000760000 50.50000000000 20.0000000000	
		2 2	1.0	20.	22 20.200000/60000 50.500000000000 30.00000000000	
		1.00000001.00000000	.000000001.00000	0000.	23 20.200007,00000 30.3000000000 40.000000000	~
		CHMMIAME DDODG C	13			
					In 1 Col 1	



Exercise 1.2: Create Design Variables using the Create Template GUI

Direct Solvers & Results Access



- Supported Solvers
 - ABAQUS
 - ADAMS
 - ANSYS
 - DADS
 - LS-DYNA
 - MADYMO
 - MARC
 - NASTRAN
 - PAM-CRASH
 - RADIOSS
 - SIMPACK

- HyperWorks Solvers
 - OptiStruct
 - MotionSolve
 - HyperForm
- General Result Formats
 - MS Excel (.csv file)
 - Any XY data
- File Parser
- Additional interfaces
 - Import Reader Language
 - External Readers Programmed in C

Altair.

Interfacing with Solvers



By default, HyperStudy has 4 solvers: OptiStruct, Templex, HyperForm and HyperXtrude. Additional, solvers may be added through the

Register Solver Script GUI or

Preference file located in <install_dir>/hw/prefinc directory.


Interfacing with Solvers through the GUI

Altair[®] HyperWorks[®]

🙆 Study_1 - HyperStudy v8.0											
Eile View Tools Utilities Help											
Create Template											
Job Management											
Readers, Writers and Fun	ctions										
Begister Solver Script											
Register Utility		V	arname	Study note	e.						
Register Optimizer		15		Crosted o				1000			
	Deste	Ann Calum Carlas									
	Regis	ter Solver Script									
Link design variables											
Sensitivity		ahel	Varname		Script Path						
F Σ <u>A DOE study</u>		20001	v annamo	~ ~	Competenden						
Create DOE study		OptiStruct	OS	🗁	C:/Altair/hw8.0/optistrug						
Controlled variables		Templey	templex	a	C:/Altair/bw8.0/bw/bin/V	Kegi	ister Solver Script				
Controlled interactions			templex			-					_
Controlled allocations		Hyper⊢orm	hf	🔁	C:/Altair/hw8.0/hm/bin/V		L		 1		
Uncontrolled variables		HyperXtrude	hx	E C	C:/Altair/hw8 0/hx/bin/V		Label	Varname		Script Path	
Uncontrolled interaction		, ypon arddo					OntiOterat		. 2	Or / Altacia/Issue O/anti-tasset/Isia (A/INI2O/anti-tasset Is at	_
Uncontrolled allocation						1	OptiStruct	os		C./Aitaii/nwo.u/optistruct/bin/wiw52/optistruct.bat	
Select responses							Templex	templex	Ē,	C:/Altair/hw8.0/hw/hin/WIN32/templex.exe	
Write/Execute runs				HyperStud	v - Register Scrip		Templex	tompiox			
Extract responses					,		HyperForm	ht	- E	C:/Altair/hw8.0/hm/bin/WIN32/hfsol.bat	
Post processing							HuporYtrudo	hv	2	C:/Altair/bw8.0/by/bin/M/NI32/by.ava	
Approximation				Label: [,	Ansvs		пуреілицие	IIX		G./Aitali/IIwo.o/IIX/bill/Wilwo2/IIX.exe	_
locut matrix				l.			Ansvs	Ansvs	- E	C:/User/HyperStudy/AnsysDir/ansys.bat	
Validation matrix				Westerley E							
Build approximation				Variable:	Ansys						
Residuals											
Trade-off				Script	Cillion /HumarChu						
Anova				oonpt.							
□ ₩ Optimization study				-							
Create optimization stu					OK						
Define design variable:					<u></u>						
Constraints											
Objectives											
Post processing											
E 🐪 Stochastic study											
Create stochastic stud ⊻											
											<u> </u>
Displays the Readers Solver Script dialog b	oox.									(Add Delete	
									 		_
										Save As Append Close	

Interfacing with Solvers through the preference file





🙆 final.xml - HyperStudy v9.0b121									
File View Tools Utilities Help									
Current Study : Study 1 🗵									
🖃 👲 Study setup	Model	Solver input file	Solver execution script	Solver input arguments					
Create studies	INIOUEI								
Create models	Model_1	link.inp	ABAQUS	✓ job=link interactive					
			Radioss OptiStruct Templex HyperXtrude TCL ABAQUS ANSYS						

Integration with Excel

🙆 CostCalc.xml - HyperStudy	v9.0					
<u>File View T</u> ools <u>U</u> tilities <u>H</u> elp						
🔚 🗄 💶 🎖						
Current Study : Study_1 ×	⊠ Microsoft Excel - IBeamCostCalc.xls [Rea	d-Only]				
Create models	🐏 Eile Edit View Insert Format Iools	<u>D</u> ata <u>W</u> indow <u>H</u> elp				
Do nominal run Create responses Link design variables		") - 😫 Σ - :	2↓ 🏭 100%	• • 🕐 📮	in i	
Sensitivity	A	B	C	D	F	
Create DOE study	12 Draduction Quantity	100			L	
Controlled variables	10 - Production Quantity	100				
Controlled interactions	20 Cotum Labor Rate (hourly)	100				
Uncontrolled variables	20 - Setup Labor Rate (nourly)	IZU Structural Steel				
Uncontrolled interactions	25 - Material 26 Material Cost Par Lh					
Select responses	20 - Material Cost Per LD.	1.00	0 020	L Channel		
Write/Execute runs	50 - Raw Dimensions (iii) 55 Taal Daging Labor Date (haugh)	102	0.530	L - Channer		
Post processing	55 - Tool Design Labor Rate (nourly)	130				
Approximation Create approximation	56 - Tool Fabrication Labor Rate (nouny)	130	0.51			
Input matrix	79 - Quantity Per Next Higher Assembly	2	2 Flanges			
Validation matrix	80 - Production Quantity	100				
Residuals	81 - Direct Labor Rate (hourly)	100				
Trade-off	82 - Setup Labor Rate (hourly)	120				
Anova	213 - Parts to Join	7		2 Flanges		
Create optimization study	214 - Rivets/Stakes	300				
Define design variables	215 - Rivet Type	Standard Rivet				
Objectives	457 - Weld Type	Arc, Gas Metal Arc				
Post processing	458 - Material	Aluminum Alloy				
Create stochastic study	459 - Material Thickness (in)	0.25				
Define random variables	460 - Seam Length (in)	800				<u>N</u> ext >
Define correlation Select responses	493 - QA Inspection	7.99%				∞ ×
Write/Execute runs	583 Cost	17635				<u>~</u>
Post processing	584					
	585					
						~
						>

Ready

Altair Proprietary and Confidential Information

Copyright © 2008 Altair Engineering, Inc. All rights reserved.



HyperStudy/HyperMorph Coupling



- Morphing: automatic mesh parameterization
- Morphing shapes used as design variables in HyperStudy and OptiStruct.
- Mesh based, no CAD data needed





Shape Variable Generation using HyperMorph



Shape design variables necessary for DOE/Optimization/Stochastic studies can be easily handled using HyperMorph module within HyperMesh.

The finite element model is morphed to a given shape and then saved as "shape". Multiple "shapes" can be saved in HyperMorph. These saved "shapes" are then used by HyperStudy as design variables.



• Biasing allows for C1 continuity

Shape Variable Generation using HyperMorph

HyperWorks



Shape Variable Generation using HyperMorph

HyperWorks



Copyright © 2008 Altair Engineering, Inc. All rights reserved.

Example

Design Variable: Boss Radius





Exporting Shape Variables



Shape Optimization: Ex	(port Options		comp: 🔲 ^morphface	e loadcol: 🗖 bearing
C desvar	file: E:\temp\HI	/17.shp		export as
 export 				
				animate
	analysis code:	sub-code:		
	▲ HyperStudy	OptiStruct		

📮 bezel.fem - Notepad		📮 bezel.fem - Notepad	
Ele Edit Format View Help		Eile Edit Format View Help	
<pre>\$\$ \$\$ Optistruct Input Deck Generated by HyperMesh Version : 8.0 \$\$ Generated using HyperMesh-Optistruct Template Version : 8.0 \$\$ 5\$ Template: optistruct</pre>	<	<pre>\$\$ \$\$ Optistruct Input Deck Generated by HyperMesh Version : 8.0 \$\$ Generated using HyperMesh-Optistruct Template Version : 8.0 \$\$ \$\$ Template: optistruct</pre>	<u>^</u>
SS SS STILE = BRACKET MODEL (SOL 111)		SS SS STITLE = BRACKET MODEL (SOL 111)	=
SUBITILE = RESPONSE AT NATURAL FREQUENCIES \$\$	\$	SUBTITLE = RESPONSE AT NATURAL FREQUENCIES \$\$	\$
SS Case Control Cards SS	\$	SS Case Control Cards	\$
\$\$ \$\$ HYPERMESH TAGS		\$\$ \$\$ HYPERMESH TAGS	
SSEEGIN TAGS SSEND TAGS		SS	
BEGIN BULK		\$ BEGIN BULK	
PARAM, AUTOSPC, YES PARAM, POST, -1 PARAM, PRGPST, NO S		PARAM, AUTOSPC, YES PARAM, POST, -1 PARAM, PRGPST, NO ¢	
§ GRID Data		\$ GRID Data	
GRID 5 4.2 0.0 0.0		include "C:\Altair\hw8.0\demos\hm\bezel.OptiStruct.node.tpl"}	
GRID 7 5.5 0.0 0.0 GRID 8 6 0 0 0 0		\$\$ SPOINT Data	
GRID 13 4.2 0.0 0.6 GRID 14 4.6 0.0 0.6		\$ \$ CONM2 Elements	
GRID 15 5.5 0.0 0.6 GRID 16 6.0 0.0 0.6		\$ CONM2 999 999 00.0906 0.0 0.0 0.0	
GRID 21 4.2 0.0 1.2 GRID 22 4.6 0.0 1.2		+ 0.35 0.0 0.56 0.0 0.0 0.07 \$	
GRID 23 5.5 0.0 1.2 GRID 24 6.0 0.0 1.2		\$HMMOVE 2 \$ 999	
GRID 28 2.4 0.0 1.8 GRID 29 3.0 0.0 1.8		\$ \$ RBE2 Elements – Multiple dependent nodes	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		\$ RBE2 999 999 123456 126 127 91 81 80	
GRID 32 5.5 0.0 1.8 GRID 33 6.0 0.0 1.8		+ 90 95 129 128 96 86 85 228 + 229 199 189 190 200 195 227 226	
GRID 34 0.0 0.0 2.4 Nodes		+ 194 184 185 \$	
GRID 36 1.12 0.0 2.4 GRID 37 1.8 0.0 2.2		SHMMOVE 2 S 999	
GRID 38 2.4 0.0 2.4 GRID 39 3.0 0.0 2.4		S CQUAD4 Elements	
GRID 40 3.6 0.0 2.4 GRID 41 4.2 0.0 2.2		S CQUAD4 4 1 5 6 14 13	
GRID 42 4.88 0.0 2.4 GRID 43 5.44 0.0 2.4	~	CQUAD4 6 1 7 8 16 15 CQUAD4 10 1 13 14 22 21	~
	>		2
Ln 28, Col 1		Ln 42, Col 2	

Altair Proprietary and Confidential Information



Exporting Shape Variables

🖟 bezel.optistruct.node.tpl - Notepad	
<u>Eile Edit Format View H</u> elp	
{coeff0 = read ("C:/Altair/hw8.0/demos/hm/bezel.shp",0,0,0)} {coeff1 = read ("C:/Altair/hw8.0/demos/hm/bezel.shp",0,0,1)} {II = array(1428)}	Ele Edit Format View Help 4.2000000e+000 0.0000000e+000 5
{II = COETTU} {II = II + DVAR1 * coeff1 }	0.0000000e+000 0.000000e+000 5 0.0000000e+000 -1.222222e+000 5
<pre>GKID* 5 { getvalueatindex(II , 0),%16.8e}{ getvalueatindex("I1", 1),%16.8e} *G { getvalueatindex("I1", 2),%16.8e}</pre>	} 4.6000000e+000 0.0000000e+000 6 0.0000000e+000 0.0000000e+000 6
GRID* 6 { getvalueatindex("I1", 3),%16.8e}{ getvalueatindex("I1", 4),%16.8e}	<pre>} 0.0000000e+000 -1.0185186e+000 6 5.5000000e+000 0.0000000e+000 7</pre>
GRID* 7 { getvalueatindex("I1", 6),%16.8e}{ getvalueatindex("I1", 7),%16.8e}	<pre>} 0.0000000e+000 0.000000e+000 7 0.0000000e+000 -5.6018519e-001 7</pre>
GRID* { getvalueatindex("I1", 0),%10.00 } { getvalueatindex("I1", 9),%16.8e}{ getvalueatindex("I1", 10),%16.8e	e} 6.0000000e+000 0.000000e+000 8 0.0000000e+000 0.0000000e+000 8
GRID* 13 [getvalueatindex("I1", 11),%10.8e} [getvalueatindex("I1", 12),%16.8e}{ getvalueatindex("I1", 12),%16.8e}{ getvalueatindex("I1", 13),%16.8e}	8e} 0.0000000e+000 -3.0555555e-001 8 4.2000000e+000 0.0000000e+000 13
<pre>"G { getValueatIndex("11, 14),%16.8e} GRID* 14 { getValueatindex("11", 15),%16.8e}{ getValueatindex("11", 16),%16.8e}</pre>	8e} 0.0000000e+000 0.000000e+000 13 6.0000000e-001 -1.0634921e+000 13
GRID* 15 {getvalueatindex("I1", 14),%16.80} [GRID* 15] {getvalueatindex("I1", 18),%16.8e}{getvalueatindex("I1", 19),%16.8	8e} 4.6000000e+000 0.0000000e+000 14 0.0000000e+000 0.0000000e+000 14
G { getvalueatindex("I1", 20),%16.8e} GRID 16 { getvalueatindex("I1", 21),%16.8e}{ getvalueatindex("I1", 22),%16.8	6.0000000e-001 -8.8624340e-001 14 5.5000000e+000 0.0000000e+000 15
G { getvalueatindex("I1", 23),%16.8e} GRID 21 { getvalueatindex("I1", 24),%16.8e}{ getvalueatindex("I1", 25),%16.8	0.0000000e+000 0.000000e+000 15 6.0000000e-001 -4.8743385e-001 15
G { getvalueatindex("T1", 26),%16.8e} GRID 22 { getvalueatindex("11", 27),%16.8e}{ getvalueatindex("11", 28),%16.8	6.0000000e+000 0.000000e+000 16 0.0000000e+000 0.000000e+000 16
G { getvalueatindex("I1", 29),%16.8e} GRID 23 { getvalueatindex("I1", 30).%16.8e}{ getvalueatindex("I1", 31).%16.8	6.0000000e-001 -2.6587301e-001 16 4.200000e+000 0.0000000e+000 21
*G { getvalueatindex("I1", 32),%16.8e}	0.0000000e+000 0.000000e+000 21 1 2000000e+000 -9.0476191e-001 21
*G { getvalueatindex("I1", 35),%16.8e}	perties 00e+000 0.000000e+000 22 00e+000 0.0000000e+000 22
*G { getvalueatindex("I1", 38),%16.8e}	00e+000 -7.5396824e-001 22 00e+000 0.0000000e+000 23
*G { getvalueatindex("I1", 41),%16.8e}	1 00e+000 0.0000000e+000 23 00e+000 -4.1468254e-001 23
GRID* 30 { getvalueatindex("II", 42),%16.8e} *G { getvalueatindex("II", 44),%16.8e} Base name:	DVAR 000000000000000000000000000000000000
GRID* 31 { getvalueatindex("I1", 45),%16.8e}{ getval *G { getvalueatindex("I1", 47),%16.8e}	D0e+000 -2.2619048e-001 24 D0e+000 0.0000000e+000 28
GRID* 32 { getvalueatindex("I1", 48),%16.8e}{ getval Base label: *G { getvalueatindex("I1", 50),%16.8e}	Shape D0e+000 0.0000000e+000 28 D0e+000 -1.3055556e+000 28
GRID* 33 { getvalueatindex("I1", 51),%16.8e}{ getval	0 0 0000000000000000000000000000000000
GRID* 34 { getvalueatindex("I1", 54),%16.8e}{ getval	0.0 00e+000 -1.1190476e+000 29 D0e+000 0.0000000e+000 30
GRID* 35 { getvalueatindex("I1", 57),%16.8e}{ getval Lower bound:	-1.0 00e+000 -0.0000000e+000 30 00e+000 -9.3253970e-001 30
GRID* 36 { getvalueatindex("I1", 60),%16.8e}{ getvalueatindex("I1", 60),%16.8e}{ getval	00e+000 0.000000e+000 31
GRID* 37 GRID* Gritural and and and a set of the set o	1.0 00e+000 -6.3412696e-001 31 00e+000 0.0000000e+000 32
GRID* 38 { getvalueatindex("I1", 66),%16.8e}{ getvalueatindex("I1", 66),%16.8e}{ getval	%8.5f
"G { getvalueatindex("II", 68),%16.8e} GRID* 39 { getvalueatindex("II", 69),%16.8e}{ getval	00e+000 0.000000e+000 33
G { getvalueatindex("II", 71),%16.8e} GRID 40 { getvalueatindex("II", 72),%16.8e}{ getval	Ok Cancel 000+000 0.0000000e+000 34
*G { getvalueatindex("I1", 74),%16.8e}	
	Ln 1, Col 1
Ln I, Col I	



Chapter 2: Design of Experiments (DOE)



Design of Experiments (DOE)



What is DOE?

Design of Experiments (DOE) can be defined as a series of tests in which purposeful changes are made to the input variables of a process or system so that the reasons for changes in the output responses can be identified and observed.

Objectives of DOE Study

- To determine which factors are most influential on the responses.
- To determine where to set the influential controlled input variables so that: The response is close to the desired nominal value. Variability in output response is small. The effects of the uncontrolled variables are minimized.
- To construct an approximate model that can be used as a surrogate model for the actual computationally intensive solver.

DOE Definitions



- A *Factor* is a input parameter (or design variable) of the system.
 Factors can be controlled or uncontrolled
- A *Level* is a discrete (or continuous) value of the factor.
- A factor can be either *Discrete* i.e., slow (-) or fast (+) (Ex. a variety of seed, type of paint etc.) or *Continuous* (Ex. temperature, a sinusoidal input for frequency response analysis etc.)
- Controlled factors are design variables that can be realistically controlled in the production (real world) environment. Examples include gauge thickness of sheet steel, shape of a support bracket, and mold temperature.
- Uncontrolled factors (Noise) are variables that cannot be realistically controlled in the production (real world) environment, but can be controlled in the lab. Examples include ambient temperature and occupant seating positioning.

DOE Definitions



- *Main Effects* are the influence of the factors on the responses
- Interaction Effect is the interdependence among the factors. Due to this interdependence a difference in the response occurs when the factors are changed simultaneously as against what would have occurred when the factors were changed individually

Main Effects



DOE Example



Key DOE concepts are illustrated by considering an airbag design. Here we want to study how the **Speed of Bag Inflation** (S), Size of Vent (V) and the Size of Bag (B), influence the HIC of an air bag and also find out how these factors interact with each other.

FACTOR	LABEL	LE\	/EL
(Design Variable)		(-)	(+)
Speed of Bag Inflation	S	Slow	Fast
Size of Vent	V	Small	Large
Size of Bag	В	Small	Large

DOE Example



Initially, consider the simplified problem where the vent size (V) is constant and consequently the are only two factors, S and B. For this, a simple full factorial DOE matrix is given as:

ROW	S	В	SXB
1	-	-	+
2	+	-	-
3	-	+	-
4	+	+	+
Divisor	2	2	2

The following DOE study will result in a polynomial expression that relates HIC to the two factors: Speed of the air bag and Size of the Bag.

Example: HIC = a0 + a1 * S + a2 * B + a3 * S * B

DOE Study Types - Applications



• DOE Types Available in HyperStudy

- Full Factorial
- Fractional Factorial
- Central Composite
- BoxBehnken
- Plackett Burman
- Latin Hypercube
- Hammersley
- Run Matrix
- User Defined
- DOE for Screening
- DOE for Factorial Studies
- DOE for Response Surface (RSM) evaluation

DOE for Screening



Objective

- A simple DOE study (ex. two level design with no interactions) will provide a global understanding of the complete system i.e give the magnitude and direction of effects
- This initial screening exercise will allow parameters which do not influence the system to be discarded thus reduce the number of factors and runs.
- Lower precision

Types

- Fractional Factorial
- Plackett-Burman
- D-Optimal

DOE for Factorial Study



Objective

- Fewer factors
- Main effects and some factor interaction effects
- Linear model

Types

- Full Factorial
- Fractional factorial
- D-Optimal

DOE for Response Surface



Objective

- Fewer factors
- Model of relationships
- Accurate prediction
- Optimization

Types

- Box-Behnken
- Central composites
- D-Optimal

DOE Process



- 1. Assign factors (dv's) from .tpl, .hm,.mdl, .hf file.
- 2. Perform nominal run to create responses for DOE study.
- 3. Select the **DOE type** for controlled and/or uncontrolled factors.
- 4. Divide the factors into controlled and uncontrolled if needed.
- 5. Export the solver input files for the specified runs as required by the DOE matrix.
- 6. Solve the above exported files.
- 7. Extract the responses for the above solved files.
- 8. Study the main effects, interaction effects, sensitivity index.



Exercise 2.1: DOE Study of a Rail Joint using OptiStruct



Exercise 2.2: DOE Study of a Cantilever Beam using Templex

Design Variables Width: 20 < b < 40Height: 30 < h < 90Lenght: 50 < L < 100Design Space: All beam elements



Responses:

$$U_{\max}(L,b,h) = \frac{PL^2}{3E}$$

$$Vol(L,b,h) = Lbh$$





Exercise 2.3: DOE with MotionSolve Data Mining



Chapter 3: Approximations

Approximation





Approximation



• Approximations are surrogate models that represent the actual responses.

• Why the need for approximations?

- Some simulations are computationally expensive which makes it impractical to rely on them exclusively for design studies. Use of approximations in such cases lead to substantial savings of computational resources.
- Optimization can fall into local minimum or maximum when the responses are nonlinear. Using approximate responses, the user can avoid this issue.

• The challenge:

- When using approximations, the issue of a trade off between accuracy and efficiency is ever present.
- The question is how approximate the representation of the design space can be while remaining accurate enough.

Approximations Definitions



• *Regression* is the polynomial expression that relates the response of interest to the factors that were varied. It is only as good as the levels used when performing the study. For example, a two-level parameter only has a linear relationship in the regression. Higher order polynomials can be introduced by using more levels. Note that using more levels results in more runs.

• Linear Regression model

- F(X) = a0 + a1X1 + a2X2 + (error)
- Interaction Regression Model
 - F(X) = a0 + a1X1 + a2X2 + a3X1X2 + (error)
- Quadratic Regression Model (2nd order)
 - F(X) = a0 + a1X1 + a2X2 + a3X1X2 + a4X12 + a5X22 + (error)



Anova (Analysis of Variance) is a representation of the contributing percent of each design variable for the selected response.



Approximation Types



- Least Squares Regression
- Moving Least Squares Method
- HyperKriging

Approximation Types



Least Squares Regression (LSR)

LSR attempts to minimize the sum of the squares of the differences (*residuals*) between responses generated by the approximation and the corresponding simulation results.

HyperStudy allows the creation of least squares regressions for any polynomial order.

Moving Least Squares Method (MLSM)

MLSM is a generalization of a conventional weighted least squares model building. The main difference is that the weights, associated with the individual DOE sampling points, do not remain constant but are functions of the normalized distance from a DOE sampling point to a point x where the approximation model is evaluated.

HyperStudy provides the choice of first, second, and third order functions for the Moving Least Square fit.

Approximation Types



HyperKriging

An approximation function, built using Kriging, has the property that it agrees with the original response function at the DOE points, i.e. this approach produces an interpolation model. This makes the technique suitable for modeling highly nonlinear response data that does not contain numerical noise. HyperKriging is computationally more expensive that the LSR and MLSM method.



Least squares quadratic regression

Kriging model

Approximations Process



- 2. Import the run matrix
- 3. Build approximation models for each response
- 4. Check the residuals
- 5. Do trade-off studies and view ANOVA results



Exercise 3.1: Approximation Study of a Rail Joint using OptiStruct



Exercise 3.2: Approximation Study of a Cantilever Beam using Templex


Chapter 4: Optimization

Optimization



Altair^{*} HyperWorks^{*}

Optimization Problem Formulation



Objective:	min f(x)	
	min cost (\$)	
Constraints:	$g(\mathbf{x}) \leq 0.0$	
	$\sigma < \sigma_{\text{allowable}}$	
Design Space:	lower x _{i <} x _{i <} upper x _i	
	2.5 mm < thickness < 5.0 mm	
	number of bolts ∈ (20, 22, 24, 26, 28, 30)	



Design Variables: System parameters that can be changed to improve the system performance.

beam dimensions, material properties, diameter, number of bolts

Objective Function: System responses that are required to be minimized (maximized). These responses are functions of the design variables.

mass, stress, displacement, frequency, pressure drop

Constraint Functions: System requirements that need to be satisfied for the design to be acceptable. These functions are also functions of the design variables.

displacement, frequency, pressure drop, cost

Optimization Definitions



Feasible Design: Design that satisfies all the constraints.

Infeasible Design: Design that violates one or more constraints.

Optimum Design: Set of design variables along with the minimized (maximized) objective function that satisfy all the constraints.





Optimization methods can be classified in three categories:

1. Gradient Based: are effective when the sensitivities (derivatives) of the system responses w.r.t design variables can be computed easily and inexpensively.

2. Response Surface Based: are very general in that they can be used with any analysis code including non-linear analysis codes. Global optimization methods use higher order polynomials to approximate the original structural optimization problem over a wide range of design variables.

3. Exploratory Methods: are suitable for discrete problems such as finding the optimum number of cars to manufacture. These methods do not show the typical convergence of other optimization algorithms. Users typically select a maximum number of simulations to be evaluated. These algorithms are good in search on nonlinear domains however they are computationally expensive.



Gradient Based Methods



- Design sensitivity analysis (DSA) must be available
- Applied for linear static and dynamic problems
- Mostly integrated with FEA Solvers
- Not feasible for Non-Linear solvers

Surface Approximation Methods





- Sequential Response Surface update
 - Linear step
 - Quadratic response surface
- Non-linear physics
- Experimental Analysis
- Wrap-Around Software
- HyperOpt is solver neutral

Exploratory Methods





- Gradients are not needed
- Suitable to nonlinear simulations
- Computationally expensive as it may require large number of analysis
- Convergence not guaranteed

Optimization Methods in HyperStudy



HyperStudy offers four optimization engines: Adaptive Response Surface, Method of Feasible Directions, Sequential Quadratic Programming, Genetic Algorithm. It also allows for User-defined Method.

With user-defined method, user's may interface their optimization code with HyperStudy to perform optimization.

Also, the approximations can be used as surrogate model to perform optimization.

Optimization:			
Label	Varname	Optimization Engine:	Adaptive Response Surface Method (HyperOpt)
Optimization_1	opt_1		Adaptive Response Surface Method (HyperOpt) Sequential Quadratic Programming Method of Feasible Directions Genetic Algorithm User Defined - Xopt

Adaptive Response Surface Method (HyperOpt)



- In this approach, the objective and constraint functions are approximated in terms of design variables x using a second order polynomial.
- The polynomial coefficients are determined using a least squares fit of the functions on to the previous design points (actual nonlinear analysis results).
- In general, more designs are available than are required for an exact least squares fit, making the system over-determined. HyperOpt uses a very efficient algorithm to estimate a response surface to be closer to certain designs of interest. HyperOpt also uses move limits to make the optimization algorithm robust.





The method of feasible directions is one of the earliest methods for solving constrained optimization problems.

The fundamental principle behind this method is to move from one feasible design to an improved feasible design. Hence, the objective function must be reduced and the constraints at the new design point should not be violated.



Sequential quadratic programming (SQP) is a method for solving constrained optimization problems.

The fundamental principle behind this method is to create a quadratic approximation of the Lagrangian and to solve that quadratic problem to define the search direction s.

The constraints are linearized during the search.

This quadratic problem can be solved by a variety of methods. The solution of the problem yields the search direction along which the next design that improves the objective function and does not violate the constraint can be found.

Genetic Algorithm



- A genetic algorithm is a machine learning technique modeled after the evolutionary process theory.
- Genetic algorithms differ from conventional optimization techniques in that they work on a population of designs. These designs are then evaluated for their fitness, which is a measure of how good a particular design is. Following Darwin's principle of survival of the fittest, designs with higher fitness values have a higher probability of being selected for mating purposes to produce the next generation of candidate solutions.
- In addition to reproduction, selected individual designs go through crossover and mutation. The designs that result from this process (the children) become members of the next generation of candidate solutions.
- This process is repeated for many generations in order to artificially force the evolution of a population of designs that yield a solution to a given problem.
- Genetic algorithms do not show the typical convergence of other optimization algorithms. Users typically select a maximum number of iterations (generations) to be evaluated. A number of solver runs is executed in each generation, with each run representing a member of the population.

Using Approximations as a Surrogate Model Ordinary Response Surface Method



Ordinary response surface methods are based on DOE studies

Response surface is fitted (quadratic polynomial) with a fixed number of designs analyzed, consequently it is not sufficient to approximate and optimize highly non-linear functions



Optimization Process



- 1. Assign factors (dv's) from .tpl, .hm,.mdl, .hf file
- 2. Perform nominal run to create responses.
- 3. Select the Optimization Algorithm
- 4. Select the Design Variables for optimization study
- 5. Define Objective function and optimization constraints
- 6. Launch Optimization
- 7. Postprocess the optimization results

Optimization Problem Example



 A cantilever beam is modeled with 1D beam elements and loaded with force F=2400 N. Width and height of cross-section are optimized to minimize weight such that stresses do not exceed yield. Further the height h should not be larger than twice the width b.



Optimization Problem Example



- Objective
 - Weight: min m(b,h)
- Design Variables
 - Width: $b^L < b < b^U$, 20 < b < 40
 - Height: $h^{L} < h < h^{U}$, 30 < h < 90
- Design Region: All beam elements
- Design Constraints:

 $\sigma(b,h) \le \sigma_{\max}$, with $\sigma_{\max} = 160$ MPa $\tau(b,h) \le \tau_{\max}$, with $\tau_{\max} = 60$ MPa $h \le 2*b$

Optimization Problem Example



Mathematical Design Space





HyperStudy Case Studies and Examples



- Optimization studies are increasing in popularity. It provides added value to the baseline study, it helps the engineer in understanding the physics of the problem, it allows exploration in the design space.
- The technology has been applied to full body crash. An example consists of an objective to minimize mass using footwell intrusion displacements as constraints and several material thickness values as design variables
- The technique has been applied to headform assessments. Initially to identify the worst attitude of impact. Once this is established the rib thickness of the crashbox is determined
- Four case studies are presented. Two relating to structural crashworthiness, two relating to occupant safety .
- In addition fours examples are given. Two of these examples relate to computational fluid dynamics and two of them relate to fatigue.

Case Study 1: Rail Optimization



- A box rail of length lx=800mm is clamped at the right side and is impacted by a moving stone wall of mass 1,000kg and initial velocity 2m/s from the left side
- The rail is required to absorb the maximum internal energy within an analysis time of ts=30ms. The displacement of the stone wall (xdirection) should be less than 50mm. The problem is typical of the designs encountered in car bumpers and crash boxes
- A size optimization is performed with six design variables consisting of variable thickness patches. The initial thickness of all design variables was 2mm and could vary between 0.2mm and 3mm



- The HyperOpt optimization (Sequential Optimization) is compared against local approximation and DOE approaches. Although structural behaviour is similar, HyperOpt maximizes the internal energy
- Design iteration history is presented showing the variation of the objective with the constraint

Case Study 1: Rail Optimization



Problem Setup

max Internal Energy $\Delta l_x = 50$ mm





Case Study 1: Rail Optimization

Size Optimization







- The US standard FMVSS216 requires a certification which simulates the roll over of a vehicle
- The certification test consists of moving an undeformable plate into an A-pillar at a constant velocity and the load / displacement response monitored
- The objective is to minimize the mass of the design and still ensure that the applied load meets the certification requirement. The design variables consist of five material thicknesses in the A-pillar and the roof frame



Case Study 2: Roof Crush



Case Study 2: Roof Crush



Variation of the Normalized Constraint with Time





- Airbag impactor test requires the HIC not be exceed 1000. The impactor should not displace more than 260mm. The design variables are vent orifice area, fabric permeability, ignition time and mass flow rate of the gas
- The test set-up consists of allocating an initial velocity to the impactor
- After twenty-two iterations the HIC has been considerably reduced





Objective : minimize head injury criterion (HIC)

$$HIC = \max_{t_2 - t_1 \le 36ms} \left(\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \ddot{d}_{impactor} dt \right)^{2.5} (t_2 - t_1)$$

Constraint : Deflection $d_{max} < 260 mm$

Design Variables: Ignition time Fabric permeability Mass flow rate Vent orifice area







Objective with Time





Variation of the Objective / Constraint with Design Iteration History





- This study considers the occupant behavior to determine the initial package space for a knee bolster. The model consists of the lower extremities of the dummy.
- The dummy is loaded by an acceleration pulse $a_x(t)$ resulting from the crash test in accordance with FMVSS208
- The objective is to minimize the femur load with a limit on the knee penetration into the knee bolster. The design variables consist of geometric and stiffness parameters

Case Study 4: Knee Bolster Design



Substitute Model



Case Study 4: Knee Bolster Design



Optimization Problem

Objective : Minimize Femur Load: $F_{femur}(\Delta \theta, \Delta k_x, \Delta l_x, \Delta l_z, \Delta \alpha) \Rightarrow \min$ **Constraint** : Knee Penetration: $\delta(\Delta \theta, \Delta k_x, \Delta l_x, \Delta l_z, \Delta \alpha) \le 0.07 m$ **Design Variables** : Geometric and Stiffness Parameters $(\Delta \theta, \Delta k_x, \Delta l_x, \Delta l_x, \Delta l_z, \Delta \alpha)$


Case Study 4: Knee Bolster Design



Results



Case Study 4: Knee Bolster Design



Design Variable Results

DESIGN VARIABLE	LOWER BOUND	UPPER BOUND	OPTIMAL VALUE
$\Delta artheta^{\circ}$	-10.00	10.00	-10.00
Δk_x	-0.10	0.07	-0.10
Δl_x	-0.05	0.03	0.03
Δl_z	-0.05	0.05	0.05
$\Delta lpha^\circ$	-5.00	5.00	-4.65

Case Study 5: Exhaust System Design





- **Objective:** Uniform flow velocity in section S
- Average velocity
- Local deviation:
- Uniformity-Index:
- **Constraint:** Min. pressure drop between inflow and outflow







Case Study 6 : Notch Design



Design Variables for Shape Optimization



Courtesy of MAGNA STEYR

HyperWorks

Case Study 6 : Notch Design

Design Variable **Optimization Problem** History Minimize maximum stress at notch edge Both load cases considered **Stress** Shapes scaled +/- 1 **Based** Damage deteriorated! ۲ **Optimization** Damage 314,6 N/mm² -33,2% 278 N/mm² +18,8% D_{max} = 0.72 **Stresses** Courtesy of MAGNA STEYR

Altair Proprietary and Confidential Information

Copyright © 2008 Altair Engineering, Inc. All rights reserved.

Case Study 6 : Notch Design



- HyperStudy identified dynamic load case to be sensitive to damage
- New contour different to stress optimized contour
- Damage significantly reduced

Damage Based Optimization







Altair Proprietary and Confidential Information

Copyright © 2008 Altair Engineering, Inc. All rights reserved.

GENERAL DISCUSSION



- Optimization commences once a clean baseline analysis has been performed (eg. negligible hourglassing and sliding energies)
- Care with shape variables: large perturbations can result in a reduced time step size
- Mass scaling used with shape optimization should be used with caution. Particularly if the objective is to minimize mass
- For size optimization, numerical contact can be effected by changing the thickness. Workaround by keeping the contact thickness constant
- Technology can be simultaneously applied to multiple loadcases (eg. central / oblique impact attitudes)





- Changes in design variables can result in a completely different system response (eg. components not in contact may now contact)
- Original analysis duration may extend to capture the systems response optimization
- Computational time can be a barrier with each iteration requiring a complete explicit analysis
- Convergence is quickly achieved
- Optimized design can be conveniently restarted with new design variables or further optimization starting at a new position in the response surface



Exercise 4.1: Shape and Size Optimization of a Rail Joint



Exercise 4.2: Size Optimization Study using RADIOSS



Exercise 4.3: Optimization Study Using an Excel Spreadsheet



Exercise 4.4: Shape Optimization Study using HyperForm



Exercise 4.5: Shape Optimization Study using ABAQUS



Chapter 5: Stochastic

Stochastic





Uncertainty in Design

Physical uncertainty

- Loads
- Boundary and Initial condition
- Material properties
- Geometry
- Numerical simulation uncertainty
 - Conceptual modeling
 - Mathematical modeling

Manufacturing

- Sheet metal thickness
- Welds
- Random design (controlled) variables

Loads

- Direction
- Magnitude
- Random noise (uncontrolled) variables

Material data

- Elastic properties
- Failure
- Random noise or design variables



Altair HyperV

Vorks[°]



- The design should be invariant against small changes of the parameters which are inevitable in the normal design process .
- Scattering of design parameters due to real world conditions in manufacturing and boundary conditions should not reduce the performance of the product.
- Robustness is the degree to which a system is insensitive to effects that are not considered in the design.
- Robust statistical procedures has to be designed to reduce the sensitivity of the parameter estimates to unreliability in the assumptions of the model.

Robust Design



Type I Robust Design (Flexible Specifications)

Given

Model f(x) Find

A range of control factors x,

 Δx (top-level specifications)

Satisfy

System constraints

Goals

Bring the mean μ_y on target

Minimize the variance σ_y

Minimize

Deviation Functions



Deterministic Optimization Approach



- Find acceptable solution to the design problem
- Design variables are continuous or discrete quantities
- Environmental conditions are given as deterministic load cases

 $f(\mathbf{x}) \Rightarrow \min$

□ Objective

 $g_i(\mathbf{x}) \leq 0$

Constraints

 $\mathbf{x}^{L} \leq \mathbf{x} \leq \mathbf{x}^{U}$



Design space

Deterministic Optimization Approach



Deterministic optimization problem formulation does not incorporate uncertainty of design variables

Optimal solutions for systems exhibiting highly non-linear responses can be misleading

□ Surrogate models (response surfaces) built using certain approximation techniques tend to smooth the noisy behavior

Most optimization algorithms push the constraints to the bounds in search of optimal solution

Robustness Optimization Approach



Since the objective of any robust design problem is to bring the mean on target and minimize the variance, robust design problems should be modeled as bi-objective problem

□ The two objectives are :

□ Minimize the variance of performance

□ Bring the mean on target or optimizing the "mean of performance"



Robust Design Optimization Formulation



□ Objective

$$\sigma(f(\mathbf{x},\mathbf{r})) \Rightarrow \min$$

Constraints

$$\mu(\mathbf{x},\mathbf{r}) - f_{\text{Target}} \leq 0$$

 $\mu_j(\mathbf{x},\mathbf{r}) - g_{j\text{Target}} \leq 0$

 $\mathbf{X}^{L} \leq \mathbf{X} \leq \mathbf{X}^{U}$

Design variables

Random variables



?



Exercise 5.1: Stochastic Study of a Rail Joint



Exercise 5.2: Stochastic Study of a Plate Model

REFERENCES



'Structural Optimization in Occupant Safety and Crash Analysis', U Schramm, D Schneider and H Thomas, OptiCON'98, California, Oct. 1998.

'HyperOpt User's Manual', Version 6.0, Altair Engineering, Jan. 1999

'AutoDv User's Manual', Version 3.0, Altair Engineering, June 1998

'Validation of the Hybrid III Head and Neck Models for the Pole Test of FMVSS 201 Using Optimization Techniques', L Fredriksson and A Hirth, CAD-FEM User's Meeting, Oct. 1998

'Crashworthiness optimization Study of a Bumper System', K Simpson and G Harte, HyperWorks Conf. Proceedings, Jan 1999

'Structural optimization - An Efficient Tool in Automotive Design', U Schramm, ATZ WorldWide, pg 17-20, 100 (1998) 6.

'Shape Optimization using Star-CD and HyperStudy', M Ratzel and C E Baumann, STAR American Conference, Detroit, May 2006.