DESIGN-ORIENTED TRANSLATORS FOR AUTOMOTIVE JOINTS

by
Luohui Long

DISSERTATION SUBMITTED TO THE FACULTY OF
VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
AEROSPACE ENGINEERING

APPROVED

E. Nikolaidis, Chairman	O. F. Hughes
E. R. Johnson	R. Kapania
C. E. Knight	C. P. Agrawal

September 1998
Blacksburg, Virginia

DESIGN-ORIENTED TRANSLATORS FOR AUTOMOTIVE JOINTS

by
Luohui Long

Committee Chairman: Efstratios Nikolaidis
Aerospace Engineering

(ABSTRACT)

A *hierarchical* approach is typically followed in design of consumer products. First, a manufacturer sets performance targets for the whole system according to customer surveys and benchmarking of competitors' products. Then, designers cascade these targets to the subsystems or the components using a very simplified model of the overall system. Then, they try to design the components so that they meet these targets. It is important to have efficient tools that check if a set of performance targets for a component corresponds to a feasible design and determine the dimensions and mass of this design.

This dissertation presents a methodology for developing two tools that link performance targets for a design to design variables that specify the geometry of the design. The first tool (called translator A) predicts the stiffness and mass of an automotive joint, whose geometry is specified, almost instantaneously. The second tool (called translator B) finds the most efficient, feasible design whose performance characteristics are close to given performance targets.

The development of the two translators involves the following steps. First, an automotive joint is parameterized. A set of physical parameters are identified that can

completely describe the geometry of the joint. These parameters should be easily understood by designers. Then, a parametric model is created using a CAD program, such as Pro/Engineer or I-Deas. The parametric model can account for different types of construction, and includes relations for styling, packaging, and manufacturing constraints. A database is created for each joint using the results from finite element analysis of hundreds or thousands of joint designs. The elements of the database serve as examples for developing Translator A. Response surface polynomials and neural networks are used to develop translator A. Stepwise regression is used in this study to rank the design variables in terms of importance and to obtain the best regression model.

Translator B uses optimization to find the most efficient design. It analyzes a large number of designs efficiently using Translator A. The modified feasible direction method and sequential linear programming are used in developing translator B. The objective of translator B is to minimize the mass of the joint and the difference of the stiffness from a given target while satisfying styling, manufacturing and packaging constraints.

The methodologies for Translators A and B are applied to the B-pillar to rocker and A-pillar to roof rail joints. Translator B is demonstrated by redesigning two joints of actual cars. Translator B is validated by checking the performance and mass of the optimum designs using finite element analysis.

This study also compares neural networks and response surface polynomials. It shows that they are almost equally accurate when they are used in both analysis and design of joints.

Acknowledgements

During the last four years I was fortunate to work with the research group of professors, and graduate students at the Aerospace and Ocean Engineering Department of Virginia Polytechnic Institute and State University, and engineers from the CAE department of Ford Motor Company. The unique environment of cooperation and support both in studies and in research encouraged and greatly contributed to my academic and professional growth.

In the very first lines of this work I would like to express my deep gratitude and thankfulness to my advisor and committee chairman, Dr. Efstratios Nikolaidis. From my very first days at Virginia Tech he was a source of guidance and support to me. No matter what kind of problem was discussed, his advice and encouragement always provided new perspectives. The expertise that he shared with me remains a tremendous source for the professional growth to me. I am very thankful for everything he has done for me.

I am most thankful to all the members of my committee: Dr. E. Johnson, Dr. O. Hughes, Dr. R. Kapania, Dr. C. Knight, Dr. C. Agrawal and Dr. Z. Gürdal. Their technical expertise and assistance played the key role in the progress of my research. All their extraordinary help is greatly appreciated.

The great contribution to everything I have done was made by my many friends - both undergraduate and graduate students from Virginia Tech. It is impossible to list all of them over here, but particular thanks are addressed to Q. Ling, S. Murphy, N. Guyot, and M. Beeken.

Financial support for this research project was provided by the Department of Aerospace and Ocean Engineering and Ford Motor Company. I am thankful to our contract monitors, Dr. C. Agrawal, Dr. S. Patill, Dr. K. Sohn, Mr. B. Eagan, Mr. P. Geck

and S. Jasuja who provided technical and financial support. The author would also like to acknowledge the support of Dr. S. Kelkar of Ford Motor Company, who understood the potential of this project in its early stages and helped secure support.

Foremost, I would like to thank my family who has been unfailingly supportive throughout my educational career. I dedicate this work to them.

Contents

List of Tables

List of Figures

Nomenclature

AIC	Akaike's Information Criterion
b_i^k	bias of the i^{th} neuron in k^{th} layer in a neural network
c_i	coefficient in a polynomial
cov	coefficient of variation
CAD	computer aided design
CAM	computer aided manufacturing
C_N^i	combination of i number from a total of N numbers
C_P	Mallows criterion
DOE	design of experiment
DOF	degree of freedom
E^d	d-dimension space
E^n	n-dimension space
$f_i^k(\cdot)$	transfer function of a neuron
$F(\cdot)$	objective function
F/A	forward/afterward
FEA	finite element analysis
FEM	finite element method
$g_j(\cdot)$	inequality constraint
$h(\cdot)$	equality constraint
I/O	inboard/outboard
IMSE	Integrated Mean Squared Error
K_D	prediction from a double regression model, can be I/O, F/A, torsion stiffness or mass
$K_{F/A}$	predicted F/A stiffness
$\hat{K}_{F/A}$	target F/A stiffness
$\overline{K}_{F/A}$	value used to normalize F/A stiffness
$K_{I/O}$	predicted I/O stiffness
$\hat{K}_{I/O}$	target I/O stiffness

$\overline{K}_{I/O}$	value used to normalize I/O stiffness
K_L	prediction from a linear polynomial model, can be I/O, F/A, torsion stiffness or mass
K_Q	prediction from a quadratic polynomial model, can be I/O, F/A, torsion stiffness or mass
K_{Tor}	predicted torsion stiffness
$\hat{K}_{Tor}$	target torsion stiffness
$\overline{K}_{Tor}$	value used to normalize torsion stiffness
M	mass
MFD	modified feasible direction method
MPP	massively parallel processing
n	number of designs
n_C	number of designs used to validate the translator B
n_D	degree of double regression model
n_V	number of design variables in regression
n_f	number of designs used in fitting/training
n_m	total number of weights and bias of a neural network
$n_{n,i}^k$	number of inputs for the i^{th} neuron in the k^{th} layer of a neural network
NCON	number of constraints in optimization
NDV	number of design variables in optimization
NN	neural networks
NURBS	non-uniform rational B-splines
OED	optimal experimental design
p	number of terms in a polynomial
$\bar{r}$	average of r_i
r_i	ratio of predicted value over measured value
R^2	coefficient of determination (R-square)
RSP	response surface polynomials
s	square root of estimate of variance of the error in the measurements
$\boldsymbol{S}^q$	search direction in optimization
SA	simulated annealing
SLP	sequential linear programming

SQP	sequential quadratic programming
SSE	sum of square errors
SSR	regression sum of squares
SSTot	total sum of squares of the variations of measurements from the mean
$w_{i,j}^{k}$	weight of i^{th} neuron in the k^{th} layer of a neural network on the input from the j^{th} neuron of $(k-1)^{th}$ layer
x_i	i^{th} variable
x_i^L	lower bound of the i^{th} design variable
x_i^U	upper bound of the i^{th} design variable
$\boldsymbol{X}$	vector of variables
$y_{FEA,j}$	FEA result of j^{th} design
y_i	measured response
$y_{O,j}$	optimization result of j^{th} design
$\overline{y}_{FEA}$	average of FEA results
$\overline{y}_i$	average response
$\overline{y}_O$	average of optimization results
$\hat{y}_D$	prediction of response from double regression model
$\hat{y}_i$	prediction of response
$\hat{y}_L$	prediction of response from linear polynomial
$\boldsymbol{Y}_{FEA}$	vector of the FEA results
$\boldsymbol{Y}_O$	vector of optimization results
z_i^k	total input for the i^{th} neuron in the k^{th} layer
α	coefficient in objective function of translator B
λ	step used in optimization
β	coefficient used in optimization
$\rho_{X,Y}$	correlation coefficient
σ	standard deviation
θ	coefficient used in sigmoid function
ϑ	multiplier in sequential linear programming

Chapter 1
Introduction

This chapter defines the problem of developing approximate design tools for joints in Section 1.1. Section 1.2 reviews previous studies on analysis and design of joints in automotive structures and approximate tools for rapid analysis of structures. Section 1.3 explains the objectives and benefits of the project presented in this dissertation.

1.1 Definition and Significance of the Problem

This section defines the problem of developing approximate tools for rapid analysis and design of automotive joints, and explains why these tools are important. Then it presents the goals of the study presented in this dissertation.

Typically, design of a complex system starts by setting targets for its performance characteristics. Then, design engineers cascade these targets to the components and design the components to meet these targets.

A *hierarchical* approach is followed in design of many real life complex systems, such as cars: first, the overall system is optimized using a concept model, which represents the constituent components using engineering parameters. Then, the components are optimized. For example, in design of a car body, first, we optimize the values of the engineering parameters describing the performance characteristics of its components (e.g., cross sectional properties of beams, and stiffness of joints). Then, we optimize the design variables of the components, such as plate gages and blending curve radii, using the optimum values of the engineering parameters. This approach allows design engineers to work with relatively simple models, which involve a manageable number of parameters, at each level.

Joints affect significantly the static and dynamic behavior of a car (Chang, 1974). When using the hierarchical approach we need tools that, for a given set of values of the engineering parameters of a joint, rapidly determine if there is a feasible design that corresponds to these values and estimate the mass and dimensions of this design.

Traditionally, a joint is defined to be any region of a structure containing the intersection or junction of two of more load-carrying members. A joint is formed by metal sheets fastened by spot welds. In the automotive industry, seven regions of the car body are generally considered to be joints, namely, *A-pillar to roof rail*, *A-pillar to hinge pillar to shotgun to cowl, B-pillar to roof rail, C-pillar to roof rail, front hinge pillar to rocker, B-pillar to rocker,* and *rocker to rear quarter* (Figure 1.1).

Joints are flexible; they allow relative rotation between the adjacent branches. The flexibility of joints affects significantly the static and dynamic response of car bodies. In concept models of car bodies, a joint is modeled by allowing its most flexible branch to rotate relative to the remaining branches, which are rigidly connected. The performance of a joint is characterized by its stiffness and mass. To define properly the stiffness, one should use the complete stiffness matrix corresponding to the ends of the joint branches. For example, the performance of the joint in Fig. 1.2 should be characterized by an 18×18 stiffness matrix corresponding to the points A, B, C at the ends of the three branches. However, this would require use of a large number of elements to characterize the performance of a joint. For example, we would need 78 elements to completely specify the stiffness matrix of the B-pillar to rocker joint in Fig. 1.2. Therefore, automotive manufacturers use simpler models involving fewer parameters to characterize the performance of a joint. A joint is thought as a set of beams connected by hinges at the joint center. Rotational springs constraint the rotations of the members relative to each other. The stiffnesses of these springs are used to characterize the performance of a joint. Usually, all branches are assumed rigidly connected except for the most flexible one. The stiffness of a spring in a particular direction is estimated by applying a moment to the most flexible branch and dividing this by the resulting rotation of the branch in the direction of the moment. Note that since there is coupling between the stiffnesses in

different directions, the stiffness obtained using above method may be considerably different than the corresponding element in the stiffness matrix of the joint.

Figure 1.2 shows the definition of three stiffnesses, namely, *inboard/outboard* (I/O), *forward/afterward* (F/A) and *torsion* stiffnesses, for the B-pillar to rocker joint. Figure 1.3 explains the definition of these stiffnesses for A-pillar to roof rail joint. Here I/O, F/A and torsion stiffnesses are defined by

$$\text{Stiffness} = \frac{\text{Moment in a direction}}{\text{Resulting rotation in same direction}}$$

In the preliminary design stage of a car, it is important to have simple, design-oriented tools that can predict the stiffness and mass of a joint rapidly once its dimensions and type of construction are specified. Another type of tool is needed that finds a feasible design, that is a design that satisfies packaging, manufacturing and styling constraints if the stiffness and mass requirements are specified. Unfortunately, in many industries, such as the automotive, there are no such tools. Instead, crude empirical relations are used to link engineering parameters to physical design variables and mass, which can lead to setting unrealizable targets for the engineering parameters of the components. It is not efficient to use *Finite Element Analysis* (FEA) to predict the performance of a joint because we do not have enough information to model joints in the concept design stage, and FEA could be too slow. One way to overcome this problem is to develop approximate tools that link engineering parameters to physical design variables. These tools should be efficient so that they can be used to perform many design iterations at low cost. We call these tools *translators*. The first tool (*Translator A*) can translate information about the parameters that define the geometry of a joint into information about the stiffness and mass. The second tool (*Translator B*) can translate information about the requirements of the stiffness and/or the mass of a joint into information about design variables. *Response surface polynomials* (RSP) and *neural networks* (NN) can be used to develop such approximate tools because they can greatly reduce the time required to analyze a design.

The goals of the study presented in this dissertation are to develop: a) a methodology for developing translators A and B for a joint using neural networks and response surface

polynomials. b) demonstrate and validate the methodology on some actual automotive joints. c) compare neural networks and response surface polynomials. Translator A rapidly calculates the stiffness and mass of a joint whose dimensions are specified. Translator B solves the inverse problem; given some requirements about the stiffness, it finds an optimum design of the joint that satisfies these requirements as well as given packaging, styling and manufacturing constraints.

1.2 Previous Work

This section reviews studies in the following areas that are important to this dissertation:

a) Joints in automotive structures.
b) Response surface polynomials and neural networks in structural analysis and design.
c) Optimization of automotive structures.
d) Software for *computer aided design* (CAD).

1.2.1 Structural Analysis of Joints in Automotive Structures

When engineers started using FEA for analysis and design of car bodies, they recognized that it is incorrect to consider a joint as rigid, that is, to assume that one joint branch does not rotate relative to the other branches. Chang (1974) demonstrated that a car body frame model that assumes that joints are rigid underestimates seriously static deflections. Therefore, the flexibility of joints must be taken into account in order to predict accurately the static and dynamic response of a car.

Since 1970's researchers have tried to understand the deformation mechanisms of joints and to develop simplified models for joints. Chang (1974) studied the influence of body connection flexibility on the in-plane response of an automobile body structure. He used an element consisting of springs and hinges to account for the joint flexibility in the analysis of the overall car body stiffness. Nikolaidis and Lee (1992) developed a procedure to estimate the parameters of simplified joint models that consist of springs using FEA results of very detailed models of joints or from measurements. Both studies

demonstrated that a frame model of the car body will be more accurate if it accounts for the joint flexibility.

Sunami and Yoshida (1987,1990) studied the stiffness of T- and L- shaped joints made of two box beams under in-plane and out-of-plane bending, respectively. They used plates and stringers to model the joints and obtained analytical expressions of joint deflections using shear flow theory. They also studied the effects of bulkhead on the rigidity of joints. Their results agreed well with both experimental and FEA results. They found that inserting bulkheads is an effective means of preventing cross-sectional deformation, and thus increases the joint rigidity. However, these approaches have limited application because they apply to rectangular cross-sections only and they can not account for details in the construction of a joint, such as welding access holes or spot welds.

Zhu (1994) developed a generic CAD model for the B-pillar to rocker joint. In creating the model, he used flat plates to model the blending areas of the joint. This model predicted accurately the *inboard/outboard* (I/O) stiffness. However, this model did not predict the *forward/afterward* (F/A) stiffness and the *torsion* stiffness accurately (see Figure 1.2 for the definition of I/O, F/A, and torsion stiffness). The relative error for F/A and torsion stiffness exceeded 30%. Zhu's model was relatively simple and it applied to one type of construction and one shape only. Specifically, his model could not account for different types of reinforcements, different types of cross sections, and details in the geometry of a joint.

1.2.2 Response Surface Polynomials in Structural Analysis and Design

Using *response surface polynomials* (RSP) to model the performance, mass and cost of a system is a mature technique. Usually, linear or quadratic polynomials are used. RSP become prohibitively expensive when cubic and higher degree polynomials are chosen for design problems involving several variables. In addition, cubic and higher degree polynomial models may contain one or more inflection points. In gradient-based numerical optimization schemes the optimizer may converge to an inflection point rather than to a local or global optimum (Giunta, 1997).

RSP have been widely used. For example, Giunta (1997) applied RSP to the analysis and design of aircraft. He used stepwise regression to obtain the best model.

Balabanov (1997) used quadratic polynomials to construct a response surface polynomial for the weight of the wing bending material. In his study, he used backward stepwise regression to obtain the best regression model.

Venter *et al*. (1996) discussed a procedure for constructing response surface approximations for design analysis and optimization. In this paper, Venter *et al.* used a response surface to approximate the response of a homogeneous isotropic plate, which is a function of the thickness. The fitting and testing data were obtained using FEA. *Mixed stepwise regression* (a procedure employing both forward and backward stepwise regressions) was used in this paper to obtain the best regression model. The developed response surfaces were very accurate (the error was about 3%).

Roux *et al.* (1996) reviewed and investigated techniques for finding more accurate approximations, including, experimental design techniques, selection of the "best" regression equation, intermediate response functions and the location and size of the region of interest. An intermediate response function uses a function of design variables as its independent variable. In this paper, they considered three examples, including a two-bar truss, a three-bar truss, and a ten-bar truss. They found that the use of engineering knowledge about the problem could yield large benefits in terms of accuracy of the approximation and size of the region over which it is applicable. If higher accuracy is desired for a given approximation, it is best achieved by using smaller subregions – thereby reducing the bias error. The use of more experimental points could make the response surface more accurate over the region of interest, but the ultimate accuracy is limited by the choice of the approximating function. The bias error (error due to lack of fit) cannot be eliminated through the use of more experimental points. If a low order function (e.g., first or second degree polynomial) is used on data having a very small amount of noise, the selection of the "best" regression equation does not significantly contribute to the optimization process.

Zhu (1994) built a translator for an automotive joint using RSP. In constructing the RSP translator, he used a complete second degree polynomial. When increasing the number of design variables defining the geometry of a joint in the polynomial, the *generalization performance* of the polynomial translator first improved, but after a certain number, it gradually deteriorated. *Generalization performance* is the capability of an approximation tool to predict the stiffness and mass of designs that it did not see during fitting. The reason was that Zhu used a complete second degree polynomial that included some unimportant terms. The coefficients of these terms were not estimated accurately, which reduced the generalization performance. Although Zhu's approach yields reasonably accurate approximating polynomials in most cases, the approach requires that the user identifies and excludes unimportant terms.

1.2.3 Neural Networks in Structural Analysis and Design

1.2.3.1 Neural Networks

The area of *neural networks* (NN) is relatively new. Recently, it has attracted a lot of attention from both researchers and practitioners. Figure 1.4 shows a typical multi-layer feed forward NN. A NN is composed of computational elements called *neurons*. The neurons are generally arranged in several layers. The function that transfers the input information of a neuron into the output information is called *transfer function*. There are several different transfer functions, such as the *sigmoid function*, and *linear function*. These functions will be discussed in detail in the second chapter.

We generally present some examples (exemplars) to a NN, from which the NN generates general information through *a training process*. There are different *learning algorithms* to train a NN. NN learning is performed mainly through the readjustment of the weights through the application of certain learning algorithms. NN can be classified as either supervised or unsupervised according to the way by which weights are adjusted. A supervised NN will require information to be provided both from the NN itself (local information), and from the environment (external information) to guide its learning task. On the other hand, for an unsupervised NN, the learning process is accomplished based

on the local information only, implying that the NN involved must be self-adaptive without the help provided by external guidance. *Back-propagation* is a learning algorithm for supervised NN. *Competitive learning* is a learning algorithm for unsupervised NN. The competitive learning scheme provides a typical example of unsupervised NN that modify their parameters during training without external guidance (Wu, 1992).

A number of different NN have been proposed. Different training algorithms have been studied by Lippmann (1987), and Alon *et al.* (1991).

Huang *et al.* (1991) investigated some fundamental issues concerning the capability of multi-layer *perceptrons* with one hidden layer. The *perceptron* was first conceived by Rosenblatt in 1958. The input of a perceptron is a n-dimensional vector. The perceptron forms a weighed sum of the n components of the input vector and adds a bias value. A perceptron is the same as a neuron in this study. Huang *et al.* (1991) focused on realizations of functions that map from a finite subset of a *n*-dimension space E^n into a *d*-dimension space E^d. Both real-valued and binary-valued functions were considered. They derived a least upper bound for the number of hidden neurons needed to realize an arbitrary function that maps from a finite subset of E^n into E^d. They also obtained a nontrivial lower bound for realizations of *injective* functions (In this paper, the injective function mapped the inputs in a m-polytope into the outputs into the same m-polytope. Therefore, it is a one-to-one mapping function), which are useful in studying pattern recognition and database retrieval.

1.2.3.2 Application of Neural Network in Structural Analysis

Many engineers in several industries have tried to apply NN to practical problems (Jagota, 1994, Hajela, and Berke, 1990).

Wu (1992) examined the architecture and functionality of NN, and provided some examples of applications in manufacturing. In this paper, Wu explained the basic concepts in NN, including neurons, transfer functions, classification of neural networks, and learning algorithms.

Hajela and Berke (1990) used NN to represent the force-displacement relationship in static structural analysis. The trained NN provided computationally efficient capabilities for reanalysis, and were well suited for optimum design. They used underdetermined NN, that is, networks that had more unknown parameters than the number of training pairs.

Keeler (1992) used NN and fuzzy logic to predict and optimize manufacturing processes. He found that NN require little human expertise; the same neural network algorithm will work for many different systems. The predictions of NN are nonlinear functions of the parameters, which means that NN can be more realistic models than RSP in same problems. NN typically work much better than traditional rule-based expert systems on problems in which the important relationships and rules are difficult to discern, or the number of rules can be overwhelming. NN are powerful nonlinear regression algorithms, and have proven to be the ideal tool for leveraging inexpensive computer time to build models directly from data in which the input vector and/or the output vector have many components. NN achieves this by "learning" the underlying relationships between the input data and the output response.

Cook and Shannon (1992) used NN to predict the occurrence of out-of-control process conditions in a composite board manufacturing facility. The manufacturing operations considered were complex, involved many process parameters, and the relationships and interactions of these variables were difficult to understand. When applying a multi-variable regression model to these operations, they found that it could explain only 25% of the variation in the response variable. However, the values of many process variables were typically collected by the operators, allowing data sets of examples of operating conditions to be constructed. Cook and Shannon used back-propagation to train NN to represent the values of the process parameters along with historical data on relevant parameters, including temperature, moisture content, and bulk density. The trained NN were able to successfully predict the state of control of the specific manufacturing process parameters with 70% accuracy, thus, demonstrating the potential of NN in manufacturing process analysis. In this paper, they also observed that

some NN were overtrained. However, they did not propose any method to avoid this problem.

Since it generally takes too long to train a NN, some researchers tried to train the NN in a *massively parallel processing* (MPP) environment (Shieh, 1994). In his study, Shieh used a 4-layer NN to simulate the structural analysis process of trusses with 10 to 1008 bars. Shieh performed a large number of structural response analyses that is required for training a set of relatively large size NN very efficiently in a single run in parallel in a MPP environment. The trained NN provided a nearly instant output solution from a given set of input variables.

A lot of work has been done on the application of NN in optimization (Berke and Hajela, 1992). There are mainly two approaches of using NN in the optimization process. The first approach creates a NN model of the structural response and then attaches it to any conventional optimization algorithm. In the second approach, a NN model is trained using optimum designs that correspond to different requirements and conditions. The NN is trained to estimate the optimum design directly for given design conditions and bypass all the analyses and optimization iterations of the conventional approach. In many practical problems, successful similar designs could be collected within some domain of design conditions. These designs can be used as input-output pairs to train a NN that can serve as "intelligent corporate memory". This network can provide a new design for different design requirements.

Swift and Batill (1991) demonstrated the use of NN for preliminary structural design. In their study, the authors used NN to simulate the structural analysis process of a 5-bar truss, a 10-bar truss, and a wing box. Using NN as a fast structural analysis tool, the authors optimized the structures considered. A back-propagation training algorithm was used. The number of unknown parameters in the NN was larger than the total number of training designs (i.e. the NN used was *underdetermined*). They used a pre-specified percentage error as a target and stopped training process when this target was met. The NN worked well in the examples considered in this paper. However, an underdetermined NN may have poor generalization performance in other applications.

Botkin and Lust (1995) applied NN to the shape optimization of some automotive components, including an upper control arm and a steering knuckle. They used a multi-layer feed forward NN as a design tool during preliminary design to approximate the mapping relation of the optimal design space. They investigated the effect of various combinations of key part dimensions on the optimal part mass using parametric studies. Since the training data was relative expensive to generate, the NN was trained using a relatively small amount of training data (36 input-output pairs) evenly distributed over the optimal design space of interest. Near optimal values of the design variables, performance measures and the objective function (typical mass) were obtained as outputs from the networks. The trained NN could predict the optimal part mass, design variable values and performance measures with very small error (3.5%).

Zhu (1994) built a translator using both RSP and NN. In obtaining the NN results, he used the Box-Behnken method to create the training pairs. He found that the NN translator gives better predictions than the RSP translator.

Cohn (1994) used *optimal experimental design* (OED) to create the input-output mappings for training. OED is concerned with the design of experiments that are expected to minimize variability in a parameterized model. The intent of OED is usually to maximize confidence in a given model, minimize parameter variances for system identification, or minimize the model's output variance. An experimental design method allow us to efficiently explore the input space, which gives us the most amount of information using the smallest number of training sets.

In structural design of aerospace or automotive vehicles, the difficulty of choosing among a set of discrete design variables must be faced. Conventional optimization methodologies, which are typically based on sensitivity analysis and gradient based search techniques, are not suitable for the discrete problem. Optimization problems dealing with discrete design variables, while often mathematically well defined, are typically difficult to solve in practice. Swift and Batill (1992) applied NN to the optimization of structures involving discrete design variables using a *simulated annealing*

(SA) algorithm. Simulated annealing, which is based on an analogy to the annealing process in metallurgy, has been used effectively for combinatorial optimization problems. In this paper, they used two trusses and a wing as examples to test the proposed method. Objective function was the mass for the three examples. Design variables for the first two examples were the types of materials of the rods in the truss, and the lamina orientations for the wing example. NN were used to approximate the relation between the design variables and the response. They found that the application of SA and NN appears promising for the design problems considered.

1.2.3.3 Effects of Disturbance and Cross-validation on NN

Since there is generally some noise in the training data, some work has been done on the effects of the corrupted data and the ways to improve the robustness of NN. Liu (1994) studied the robustness of a conventional back-propagation algorithm for NN regression. He found that this algorithm is robust to leverages (data with *x* corrupted), but not to outliers (data with *y* corrupted).

Adding a *regularizer* to the error function being minimized during training is a technique for preventing a NN from overfitting noisy data. *Regularizer* is a function, such as a spline that can typically smooth the fit to noisy data. Wu and Moody (1995) derived a smoothing regularizer for general dynamic models, which was derived by considering perturbations of the training data. They also tested the regularizer in several case studies, and found that it performed well.

Tresp *et al.* (1994) studied how to incorporate data with uncertain or missing information into the training of NN. In their work, the general solutions required a weighed integration over the range of unknown or uncertain inputs.

Many researchers noticed that the *generalization error* decreases in an early period of training, reaches the minimum and then increases as training continues, whereas the error of training results monotonically decreases. *Generalization error* is the error of a polynomial or a neural network when using this tool to predict the performance of designs that were not in the training or fitting set. Therefore, it is advantageous to stop

training at a certain point in time. Amari *et al.* (1995) studied the effect of *cross-validation* on the generalization error of the trained NN. To avoid overtraining, the following stopping rule has been proposed, which is based on cross-validation. Divide all the available examples into two sets. One set is used for training. The other set is used for testing in a way that the behavior of the trained network is evaluated by using the test examples and training is stopped at the point that minimizes the testing error. They found that when the number n_m of network parameters is large, the best strategy is to use almost all n_f examples in the training set and to use only $1/\sqrt{2n_m}$ percent examples in the testing set. For example, when n_m=100, only 7% of the training patterns are to be used in the set determining when to stop training. When $n_f > 30\, n_m$, the results show that cross-validation does not improve the generalization error, so no overtraining is observed. Because in most real actual problems, the number of examples is limited, early stopping does improve significantly the *generalization ability* of a network. *Generalization ability* is the capability of a polynomial or a neural network to predict designs that it had not seen before.

Plutowski *et al.* (1994) studied the effects of cross-validation on the estimates of NN. They found that the cross-validation average squared error is an unbiased estimate of the *Integrated Mean Squared Error* (IMSE). IMSE is a version of the usual mean squared error criterion, averaged over all possible training sets of a given size.

Dodier (1995) presented a theory about early stopping for linear networks. He found that given a training set and a validation set, all weight vectors found by early stopping must lie on a quadric surface, which is usually an ellipsoid.

1.2.4 Optimization of Automotive Structures

Optimization of automotive structures typically needs to evaluate the performance characteristics of a design (stress, strain, or stiffness) many times. This repeated analysis process can be done using FEA or approximate tools such as RSP and NN. A lot of work has been done on the application of optimization techniques to automotive structural design.

Botkin (1992) proposed to work with design-oriented geometric features associated to part recognizable features. He integrated a fully automatic mesh generator into the three-dimensional shape optimization of a steering knuckle of a car. Unlike other papers on shape optimization, in which stress and strain constraints are imposed on every finite element and the number of elements remains constant throughout the iteration history, he used an automatic mesh generator that allowed the number of elements to vary throughout the convergence history. The stress and strain constraints were assigned to the design model rather than to the mesh.

Lindby and Santos (1994) proposed an approach that uses the capabilities offered by CAD modelers to provide shape design parameterization and design velocity computations, and uses a commercial finite element code for performance analysis. They used the shape optimization of a torque arm to illustrate the proposed approach.

Schramm *et al.* (1993,1994) used *non-uniform rational B-splines* (NURBS) in shape optimization of a thin walled beam. They used NURBS curves to describe the cross sectional shape of the beam. Design variables were the control parameters that define NURBS curves and the wall thickness of the beam. An advantage of Schramm's study is that one does not need to remesh the structure as the dimensions change during optimization. However, the method was demonstrated on structures that were practically two-dimensional. One needs to investigate how practical the method is for complex, three-dimensional structures.

Belegundu (1993) proposed a way of using *basis shapes* to optimize the shape of mechanical components. Basis shapes are the different shapes of the components that a designer wants to optimize and their FEA results. Basis shapes can be generated in a number of ways. For example, the shapes from past designs can be used as basis shapes. By using the basis shapes, Belegundu reduced the optimization problem to finding the right combination of the basis shapes. In this paper, Belegundu applied the proposed method to a torque arm component and a connecting rod. The objective function was the weight of the component. The optimum design reduced the weight of the component by 33-40%.

Puttre (1993) discussed about combining CAD and FEA to automate the repetitive tasks associated with design optimization. In this paper, Puttre summarized the work on shape optimization using existing CAD and FEA packages. Puttre recommended that the number of design variables for shape-optimization problems not exceed 20. Insignificant design variables should be removed to make the optimization problems more manageable.

Fenyes (1992) optimized thin-wall beam type member by taking into account both structural and manufacturing requirements. His approach can be applied in the early design stages. His results indicated that it is possible to simultaneously consider structural and manufacturing constraints for structures composed of simple closed channel sections. However, his results could not be extended directly to complex multi-piece sections, which are typically used for the frame members in a vehicle because the simple formulae he used to evaluate the forming constraints of sheet metals were not applicable to other cross sections or forming conditions.

1.2.5 CAD Software

Today, there are many powerful commercial CAD software packages that can help automate the design process. Among all the CAD/CAM programs, I-Deas, Pro/Engineer, CATIA, and PATRAN are among the most popular. CATIA is a CAD/CAM software that was first developed by the French aerospace industry. CATIA has many specific modules and many capabilities. Although PATRAN has many capabilities, it only has limited parametric modeling capabilities. Most people use PATRAN as a pro- or post-processor of FEA. The author only had access to I-Deas and Pro/Engineer. According to the author's experience, it is easier to change or redefine the shape or properties of a structure in Pro/Engineer than in I-Deas. I-Deas also lacks some assembly capabilities of Pro/Engineer. It is also easier to develop complex relations between the dimensions of one or more parts in Pro/Engineer than in I-Deas. On the other hand, I-Deas has more function modules, such as more finite element types. Both I-Deas and Pro/Engineer have structural analysis modules. I-Deas also has a structural optimization module. To

use the optimization module of I-Deas, the user only needs to create a model in I-Deas, and specify the objective function and constraints. Then, I-Deas can optimize the design.

The structural analysis and structural optimization capabilities of I-Deas and Pro/Engineer have greatly improved in the past few years. However, their structural analysis and structural optimization modules are still not superior in speed, accuracy, reliability, and versatility to other special structural analysis codes such as MSC/NASTRAN and ANSYS, and structural optimization codes such as DOT and GENESIS. Moreover, CAD/CAM software such as Pro/Engineer can not create some types of elements, such as rigid elements that are frequently used to simulate the spot welds.

1.2.6 Neural Networks or Polynomials versus Finite Element Analysis in Optimization

Structural optimization needs to analyze a large number of alternate designs (from a few dozens to many hundreds). Therefore, the computation effort required for analysis is an important issue. Before performing structural optimization, we need to know whether it is better to use a NN or a polynomial translator to simulate the input-output relations in the optimization process instead of directly using FEA. The following paragraphs compare the cost of using NN and RSP in optimization with the cost of performing FEA in each step of optimization.

Neural Networks

Developing and verifying a NN requires one to develop a parametric CAD model of the structure, analyze the structure for many combinations of values of the dimensions (from few hundred to few thousand), train and validate the network, and perform optimization. The computational time for the first three tasks is much larger than the last task. Indeed, it takes only a very small fraction of a second to predict the performance of a structure using a NN that has already been trained.

Carpenter *et al*. (1992) compared the efficiency and accuracy of RSP and NN for approximation of functions. In this paper, they used *Fox's banana function,* a five bar

truss, a 35 bar truss with four design variables, and a 35 bar truss with 15 design variables to compare the performance of NN and RSP. The Fox's banana function can be expressed as: $y = 10x_1^4 - 20x_2x_1^2 + 10x_2^2 + x_1^2 - 2x_1 + 5$. They concluded that the two methods were comparable in terms of accuracy and efficiency. They also suggested that the number of nodes on the hidden layer be somewhere between the average and the sum of the input and output nodes.

Shi *et al.* (1998) compared Holographic NN with RSP. They concluded that Holographic NN are more accurate than RSP.

Rogers (1994) compared the time needed by using NN to simulate analysis and the time needed by directly performing structural analysis using FEA in the optimization process. He used a 3000 *degrees-of-freedom* (DOF) beam as a test problem. The objective was to optimize the shape of the beam to minimize the weight while satisfying stress constraints. He found that there is a break-even point, beyond which, a neural network can reduce the total amount of time required for analysis (Fig. 1.5). Guyot (1996) did a similar comparison between the two approaches. He found that the initial cost to perform optimization using a NN is relatively high. Consequently, if the optimization is performed only once, it may not be advantageous to use a NN to predict the performance of the system because of the high initial cost of developing such a tool. However, the cost of using a NN is negligible once the NN is built. Thus, if we need to run an optimization program many times to perform a parametric study or to try different initial points to find the global optimum, it is advantageous to use a NN as an alternative to FEA in optimization.

Polynomials

Like NN, RSP can predict the performance characteristics of a design in a fraction of a section, thus, it can greatly reduce the time of optimization. In general, it takes less time to construct and fit a RSP than to construct and train a NN. Therefore, it is better to use a polynomial instead of a NN, if a RSP can be found that can approximate the data with acceptable accuracy.

NN and RSP can also be better than FEA if there is noise in the results of structural analysis because they avoid problems with discontinuities in the gradients. A RSP reduces the effects of corrupted data by providing a smooth approximation of the response. Liu (1994) showed that optimization using NN is not sensitive to a few corrupted data. This property of NN will be explained in chapter 2.

1.3 Objectives and Benefits of the Project Presented in this Dissertation

The objectives of this dissertation are:

a) Develop a methodology for developing translators A and B for design of automotive joints. This methodology consists of the following steps:
   - Parameterize a joint using a set of few physical design variables that can fully determine the shape of the joint. These design variables can be controlled in the early design stage of a car.
   - Develop a parametric model. The parametric model should account for packaging, styling and manufacturing constraints, and determine if a design whose design variables and type of construction are specified is feasible. This model should be able to create a FEA data file, which can be used by a finite element analysis code to give reasonably accurate predictions for I/O, F/A, torsion stiffness and mass.
   - Develop a database of designs that will be used for developing and validating translator A.
   - Understand and rank the most important design variables using stepwise regression, and identify the most important design variables that affect the performance of the joint most. It is better to use stepwise regression because it introduces only the important parameters to the regression model.
   - Develop a translator A for design guidance using both RSP and NN, which can predict the performance and mass of a given joint at a fraction of a second. This will allow the use of translator A to optimize the joint, or even the entire car body very efficiently. Unlike Zhu, who used a complete second degree polynomial to construct the polynomial translator, we will identify and

only consider the important terms when we construct the polynomial translator. By doing so, we can minimize the effects of unimportant terms on the generalization performance of the translator.

- Develop a translator B using both RSP and NN translator A. Translator B can find the values of design variables of a joint that satisfies given performance targets and packaging and manufacturing constraints. This is achieved using optimization, in which we minimize the mass of a joint subject to packaging, manufacturing and styling constraints.

b) Apply the methodology to the B-pillar to rocker and A-pillar to roof rail joints of an actual car.

For this purpose, we will complete the following tasks:

- Parameterize both the B-pillar to rocker joint (Fig. 1.2, and 1.6) and A-pillar to roof rail joint (Fig. 1.3 and 1.7). The reasons that we select B-pillar to rocker and A-pillar to roof rail joints are the following. The B-pillar to rocker joint is the simplest joint in a car, and most cars have similar types of construction. The A-pillar to roof rail joint is more complicated than the B-pillar to rocker joint. We can apply the experience gained from the parameterization of B-pillar to rocker joint and A-pillar to roof rail joint to parameterization of other joints.
- Develop two parametric joint models. One is for the B-pillar to rocker joint, and the other is for the A-pillar to roof rail joint.

Zhu's study was a very good step toward developing parametric models of joints. The differences between our B-pillar to rocker joint model and that of Zhu's (1994) are:

– Our models are created using Pro/Engineer and I-Deas. They can account for more design details, such as different reinforcements, spot welds, and access holes. Zhu wrote a FORTRAN program that generated a Finite Element Model given the dimensions of a joint. Straight lines were used in his model to simulate the blending areas.

– Our models will account for about seventy packaging, manufacturing and styling constraints. Following are some new constraints that are included in our model. Our

model will be able to determine if a joint can be manufactured in terms of stamping. The B-pillar to rocker model will also be able to determine whether the shape of the cross section of the rocker allows the water to run off. The A-pillar to roof rail model will be able to determine whether the A-pillar is too wide so that it obstructs vision, and whether the reinforcement can fit into the two outside parts of A-pillar.

- Our model will accurately predict not only I/O stiffness, but also F/A stiffness, torsion stiffness and mass.
- The user will only need to change the value of control parameters in order to change the type of construction of a joint. Zhu's model accounted only for one type of construction.

c) Develop a database of example designs for each joint that will be used to fit a response surface polynomial or train a neural network that can predict the performance and mass of a given joint.

d) Identify and rank the important design variables for each joint using stepwise regression.

e) Develop a translator A for each joint using RSP and NN. Validate and compare the RSP and NN based translators.

f) Develop a translator B for B-pillar to rocker joint and A-pillar to roof rail joint. Validate the results of the translator B by comparing the optimization and FEA results, and study the effect of changing the constraints on the most important design variables on the mass of the optimum design.

As mentioned in the introduction, this project will establish a methodology for developing two much needed tools for design of joints. The tools will help a designer analyze joints, minimize the weight of the joints of a new car model, and reduce the time it takes to design them. By using this methodology, the designer will find the most important physical design variables (dimensions) that affect the performance of a joint and establish relations between these design variables and packaging and manufacturing

constraints, which is very important information in design. The methodology developed in this project will be useful in developing similar design tools for other components of a car such as beams, and local structures at the positions where the suspension is attached to the car body

1.4 Outline of the Dissertation

Chapter 2 describes the general methodology for developing translator A. This chapter covers the parameterization of joints, selection of design variables, creation of a database, and development of translators A using RSP and NN.

Chapter 3 presents the general methodology for developing translator B, including, selection and classification of design variables, definition of objective function and constraints used in translator B, and validation of translator B.

In chapter 4, the general methodology explained in chapter 2 is applied to a B-pillar to rocker joint of an actual car. First, we create a parametric model for the B-pillar to rocker joint. The parametric model can account for packaging, styling, and manufacturing constraints, and determine if a given design whose design variables and type of construction are specified is feasible. These models can create FEA data files, which can be used with a finite element analysis code, such as MSC/NASTRAN, to predict I/O, F/A, torsion stiffness and mass. Using this generic parametric model, translator A is built and validated for the B-pillar to rocker joint. Finally, this chapter compares the RSP and NN based translators.

Chapter 5 demonstrates the work on developing translator A on the A-pillar to roof rail joint of an actual car. First, a parametric model is developed. Then, both a polynomial and a neural network translator A are built and validated. The two translators are compared.

In Chapter 6, we apply the general methodology for developing translator B, which is explained in chapter 3, to the B-pillar to rocker joint of an actual car to find designs that

meet given performance requirements. The objective function and the constraints used in this translator are explained in detail. Then, the translator B is validated. Specifically, we first check if the translator B converges to a global minimum. Then, we compare the performance of the optimum design predicted using translator B and FEA. Finally, the effects of changing constraints associated with the most important design variables on the objective function of the final optimum designs are studied.

Chapter 7 demonstrates the methodology for developing translator B on the A-pillar joint. The objective function and constraints are explained. The translator B is validated and parametric studies are performed to determine the effect of constraints on the optimum designs.

Chapter 8 summarizes the work and the conclusions. It also presents recommendations for future work.

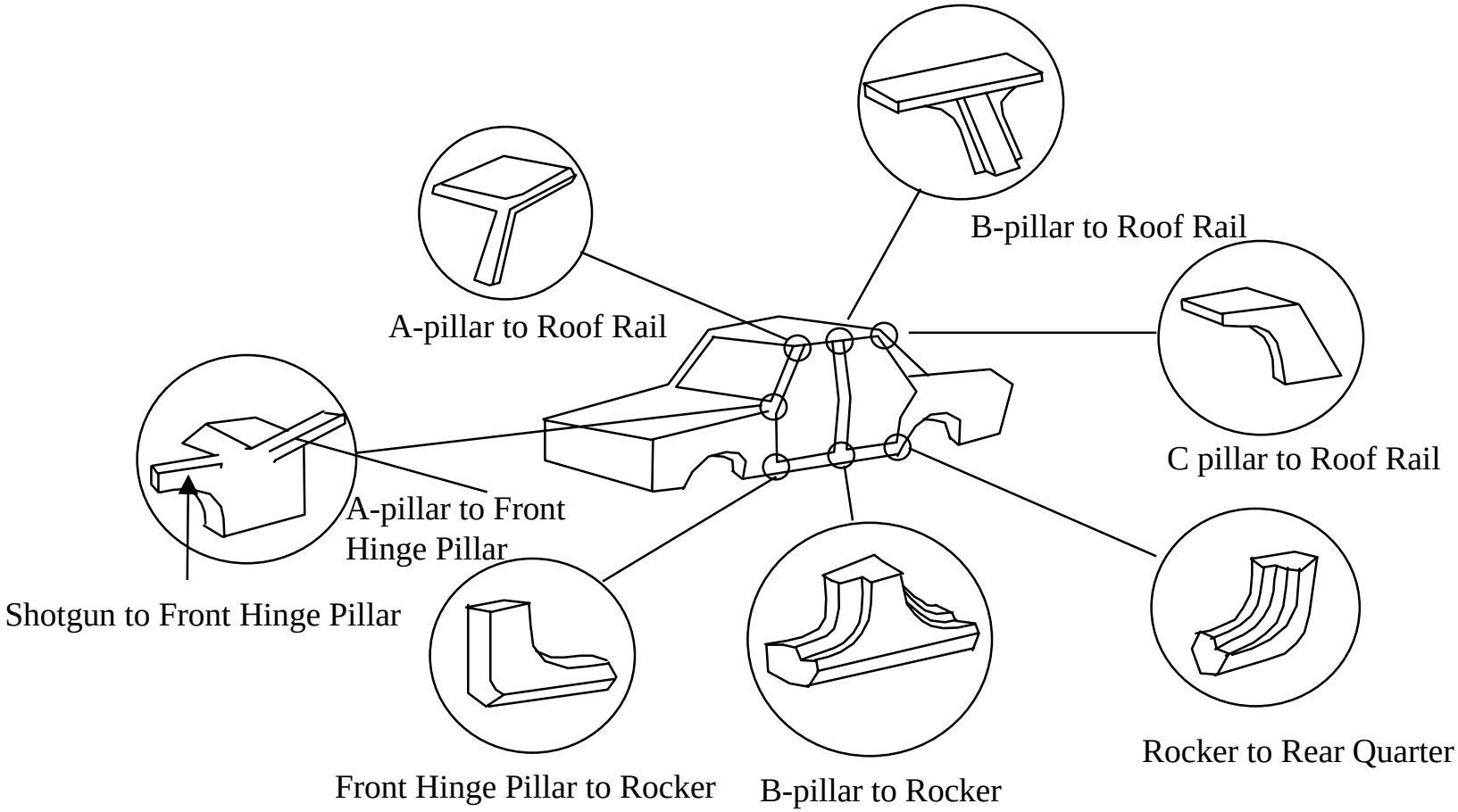


Figure. 1.1: Body Structural Joints

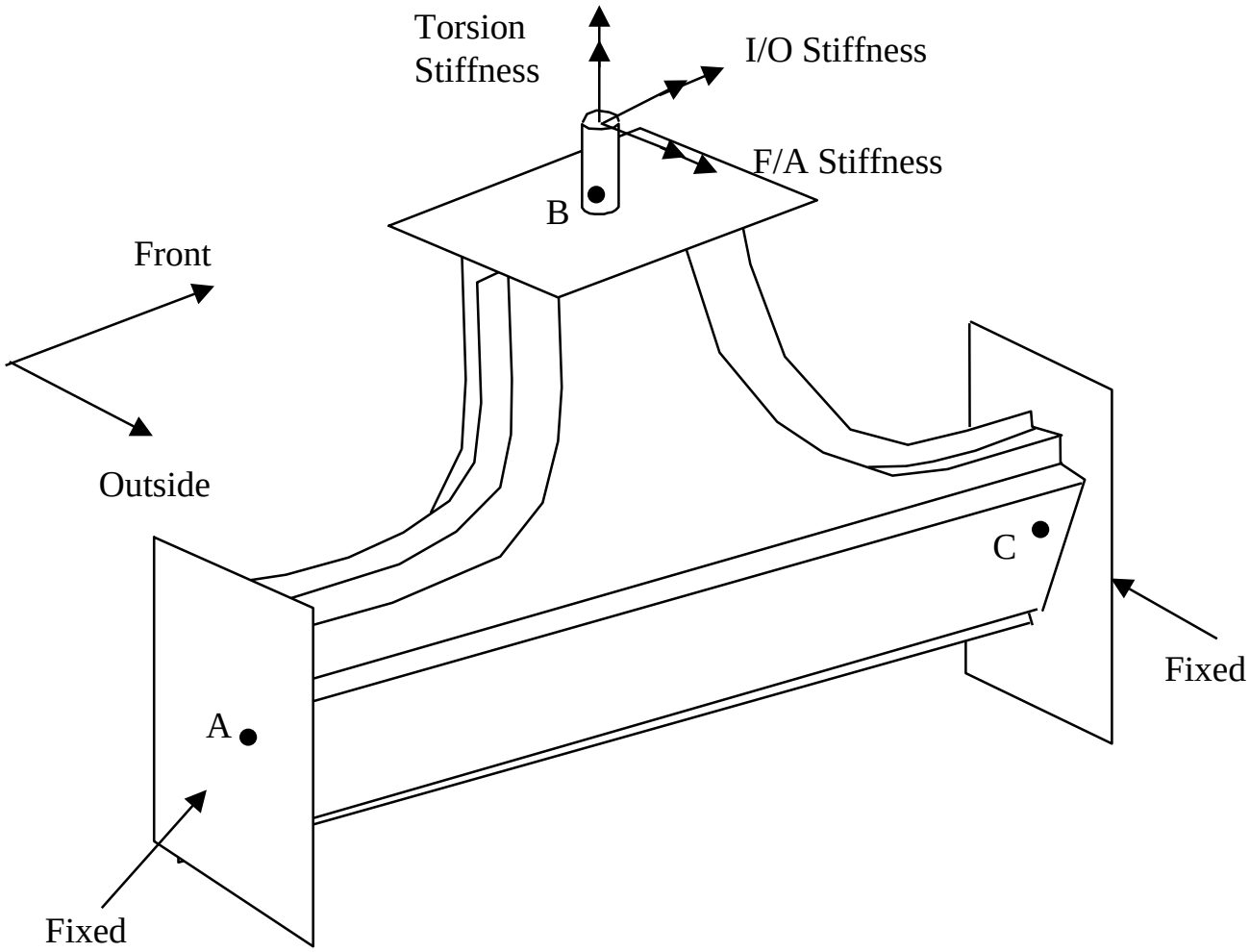


Figure 1.2: Definition of Stiffness for B-pillar to Rocker Joint

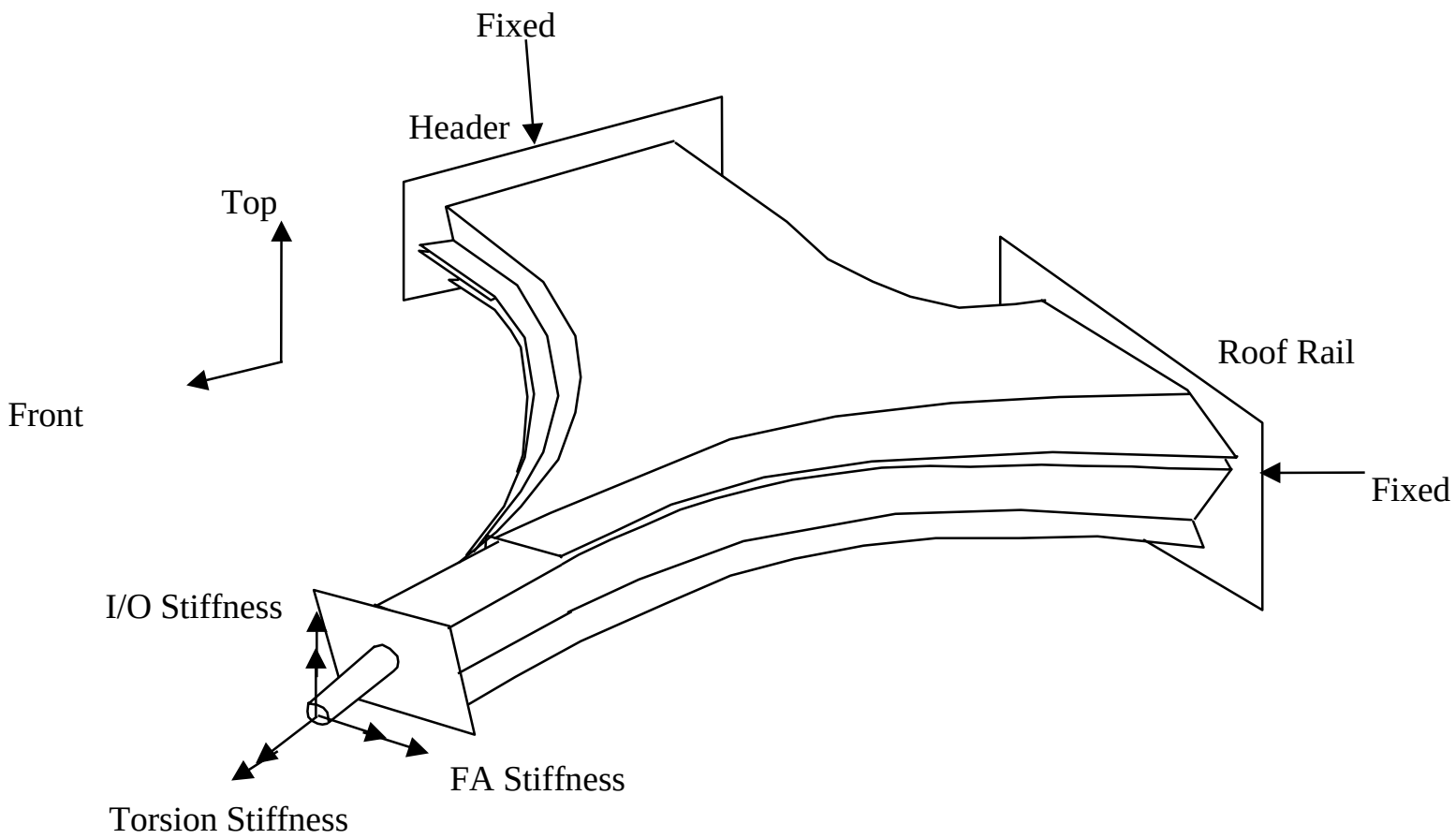


Figure 1.3: Definition of Stiffness for A-pillar to Roof Rail Joint

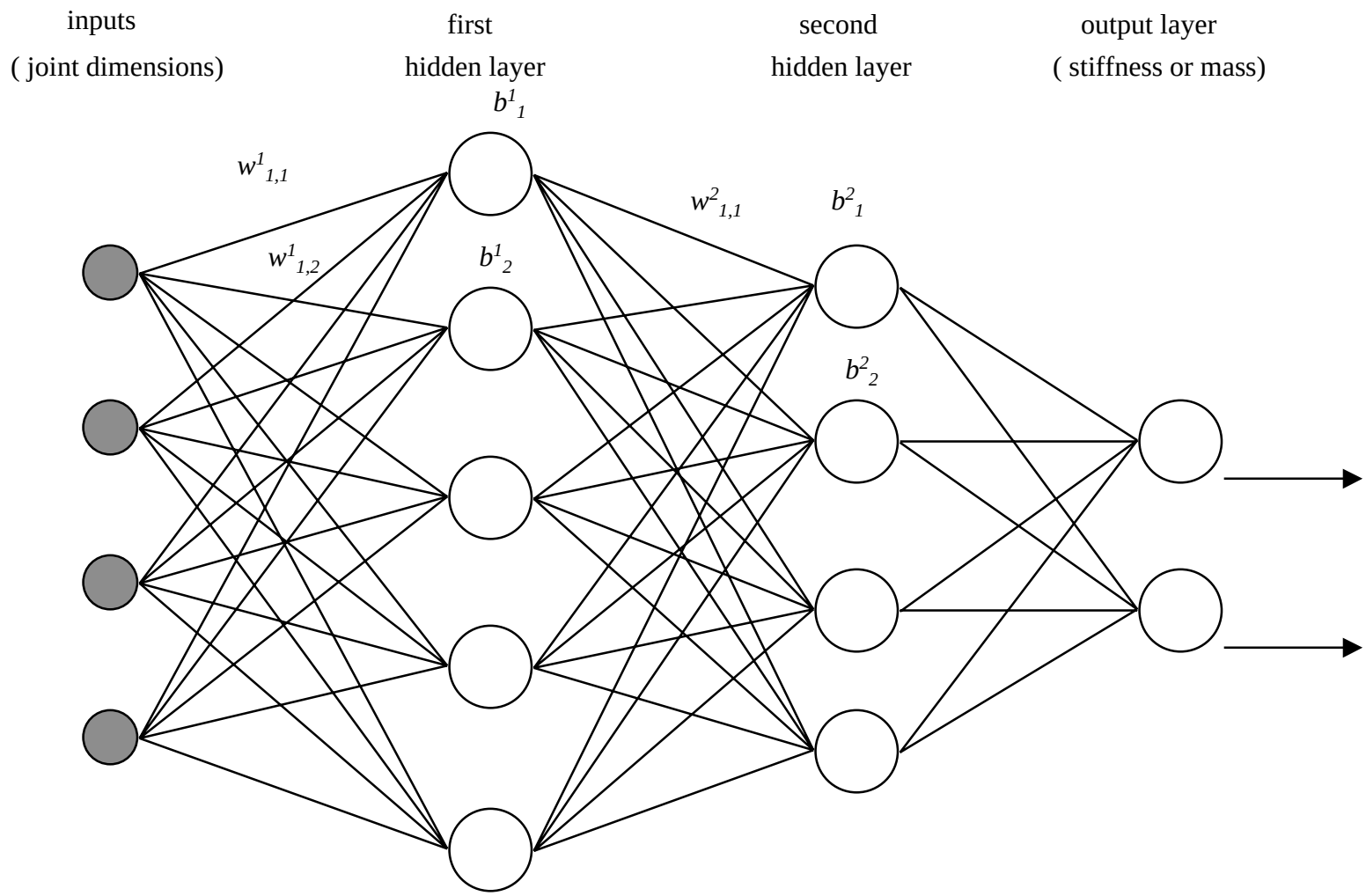


Figure 1.4 :Architecture of a Typical Multi-layer Neural Network

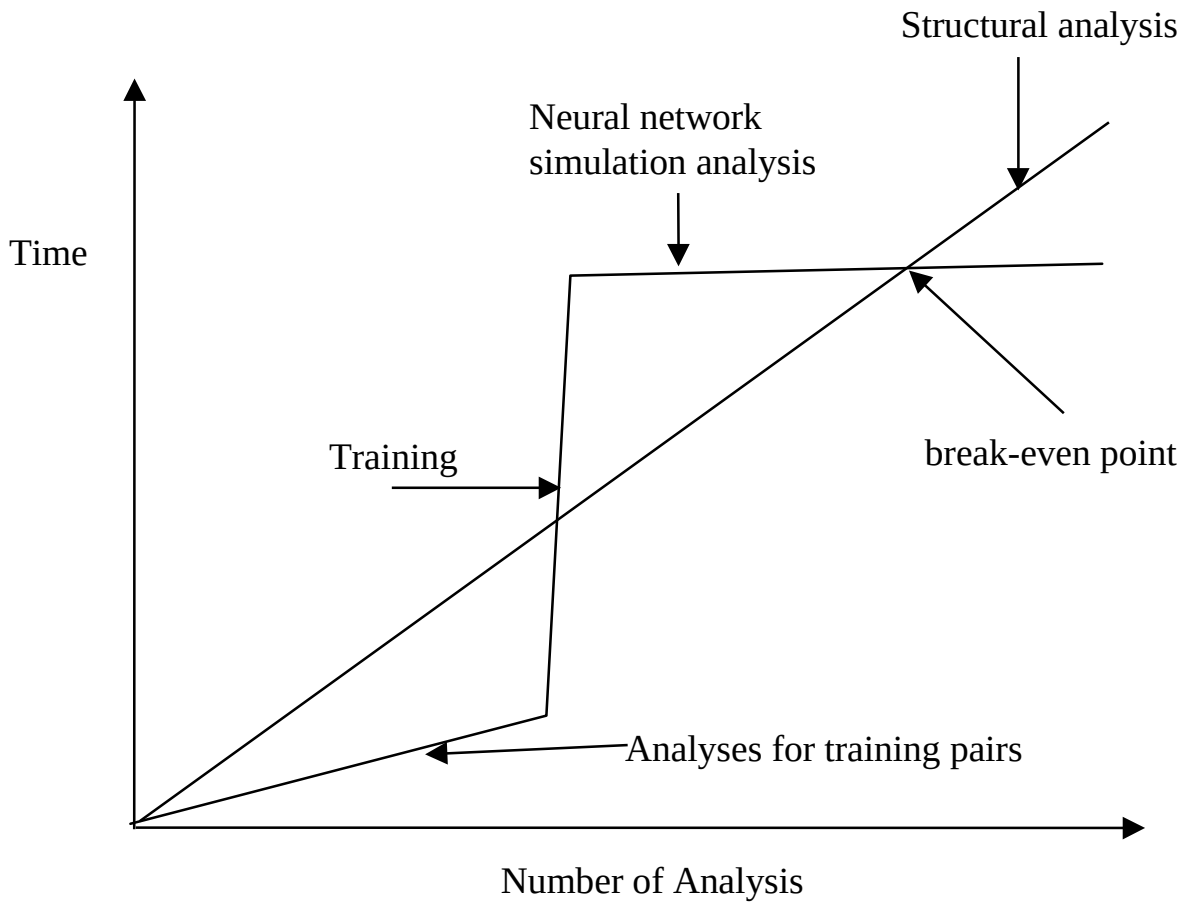


Figure. 1.5: Comparison Between Structural Analysis and Neural Network Simulation

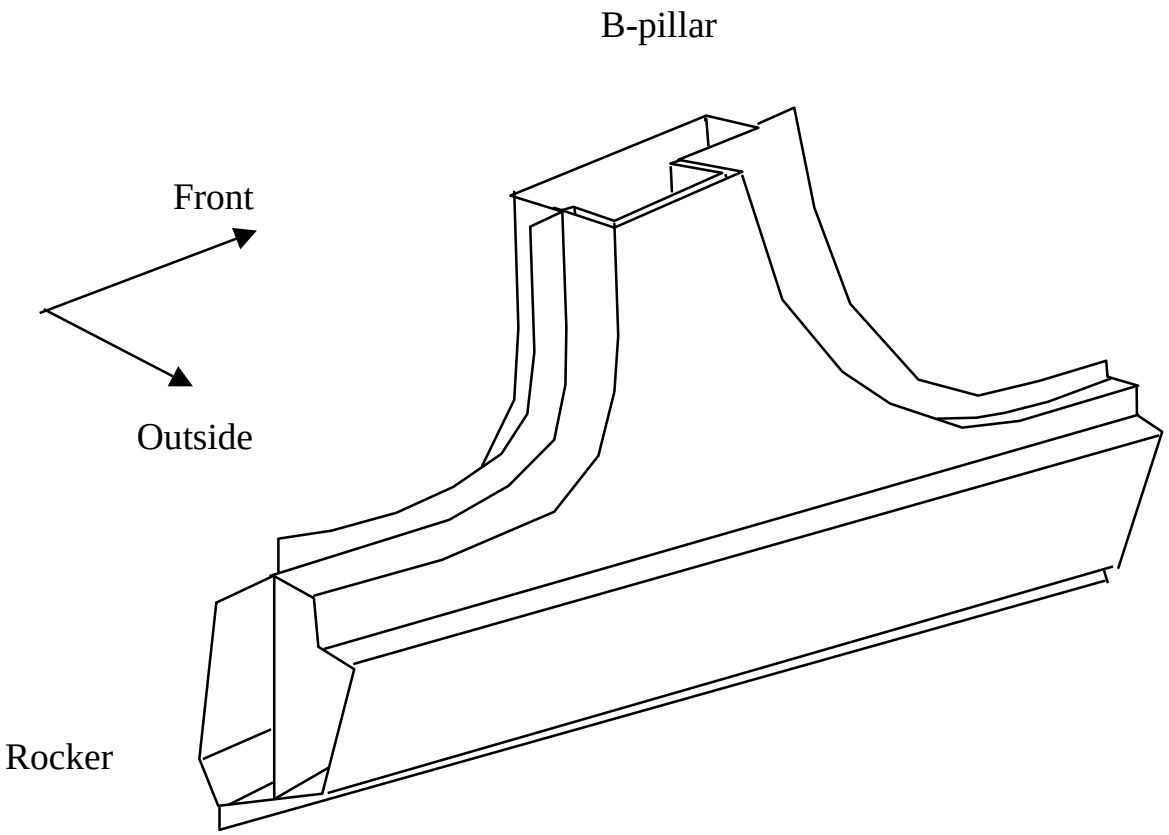


Figure 1.6: B-pillar to Rocker Joint

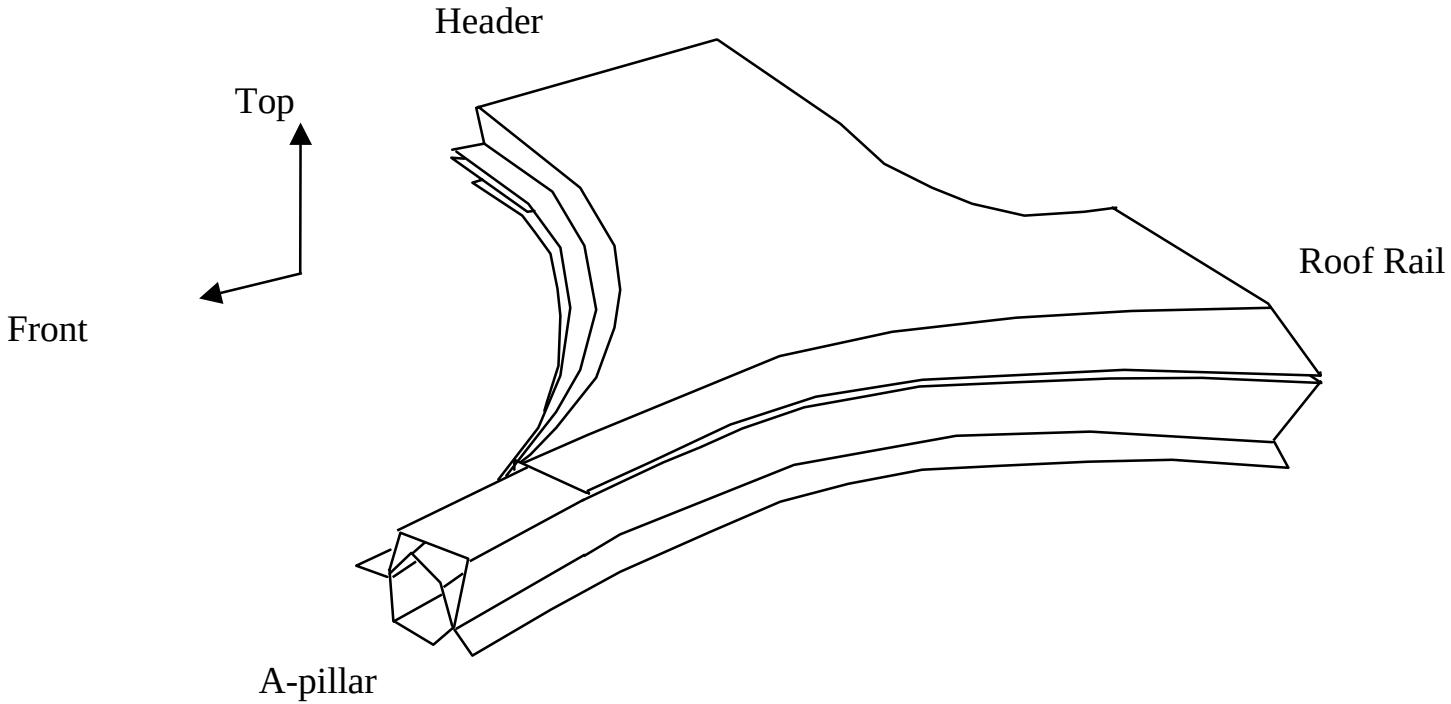


Figure 1.7: A-pillar to Roof Rail Joint

Chapter 2
Methodology for Developing Translator A

2.1 Introduction

Structural analysis programs used to solve design problems are often computationally expensive. Obtaining optimal solutions typically requires numerous iterations involving analysis and optimization programs. This process is often prohibitive due to the amount of computer time required for converging to an optimum design. One promising technique is to simulate a slow, expensive structural analysis program with a fast inexpensive polynomial or neural network. Such a polynomial or neural network is called *Translator A* in this study.

Using polynomials to create regression models is a mature technique. There are a lot of books covering this topic. The methods used in this study are briefly explained in the following sections.

The area of neural network is relatively new. A lot of work has been done on the application of neural networks on structural mechanics problems (see Chapter 1 for a review of such applications).

A neural network provides a rapid mapping of a given input into the desired output quantities, thereby enhancing the efficiency of the design process. Besides, degeneration of a few computing elements due to hardware problems or due to a few corrupt data, does not seriously affect such networks.

Even though significant progress has been made on the theory and application of neural networks in the past few years, there are still many unresolved issues. Such issues

include determination of optimal training size to meet the required error tolerance, and determination of the number of neurons in each layer of a multi-layer network.

This study develops tools (*translators A*) using both response surface polynomials and artificial neural networks that can estimate the performance characteristics (stiffness and mass) of a joint once the values of the design variables (dimensions) of the joint are specified. The translator can predict the stiffness and the mass of a given joint almost instantaneously, which will allow for an efficient optimization of the joint or the entire car body.

The development of translator A can be divided into three steps. Figure 2.1 shows these steps. First, a parametric model of a structure needs to be created. This model can be easily modified by changing the parameters that specify the dimensions and the topology of the model. Next, a database is created, which includes many feasible designs and their corresponding *finite element analysis* (FEA) results (stiffness and mass). Polynomial and neural network translators can be developed using the database.

Section 2.2 briefly describes the parametric model that is used to create translator A. Section 2.3 explains a procedure for using the parametric model to develop a database for fitting and testing. Section 2.4 explains how to develop and validate polynomial translators using the database. Section 2.5 describes the procedure of developing and validating neural network translators.

2.2 Parametric Models

To develop translator A, a parametric model is first created for each joint. CAD packages such as CATIA, I-Deas, and Pro/Engineer can be used for this purpose. The parametric model is generic, and can be easily modified by changing values of the parameters that determine the dimensions and the type of construction.

The input variables of the parametric model can be divided into several types. The first type are binary variables, which can take the values of "Yes" or "No". These variables are called control parameters, and they specify the type of construction of a joint. For example, if a branch has a bulkhead then we use a binary parameter to specify that there is a bulkhead in a particular design. The second type can be further divided into two sub types: design variables controlling the shape of the joint, such as the length and height of a branch; and design variables storing the FEA information, such as the thickness of each part.

The development of a parametric model consists of four steps. Murphy (1995) explained these steps in detail. Here each step is briefly described.

- Identify construction types of the joints by examining actual joint hardware, simplifying the geometry, and classifying joints into different types on the basis of their type of construction.
- Parameterize a joint using six level of parameters. The higher levels define the overall joint shape while the lower levels define the details of joint geometry. The design variables of each level are used to define the following:

  Level 1: Number and position of branches

  Level 2: Cross sections of branches

  Level 3: Blending areas

  Level 4: Internal reinforcements

  Level 5: Connections

  Level 6: Openings
- Establish constraints and relations between design variables to ensure that a joint design is feasible. The constraints include manufacturing, packaging, styling, and mathematical constraints.
- Create a parametric model that can account for different construction types. This model should include all the constraints mentioned in the previous step. It should warn the user when a design violates a constraint.

To develop translator A, the type of construction of the joint is fixed. This means that the control parameters are fixed. Only the second type of design variables, which control the shape and FEA characteristics of the joint, can vary.

2.3 Database for Developing Translator A

To create and validate a response surface polynomial or a neural network, a database is needed that contains many designs (between fifty to few thousands) and their performance characteristics. These designs, called *examples,* are used to train a neural network or develop a polynomial. These designs should be selected in a way that they provide the maximum amount of information about the relation between the design variables and the performance characteristics of a joint. This allows us to determine the polynomial or the neural network with good accuracy.

Many methods can be used to select the values of the design variables of the joints in the database. A lot of work has been done to compare the advantages and disadvantages of different methods. *Design of experiments* (DOE) (Box *et al.* 1960, 1978) is a branch of statistics that provides the researcher with methods for selecting the independent variable values for which a limited number of experiments will be conducted. Experimental design methods create certain combinations of experiments in which the independent variables are prescribed at specific values or levels. The results of these planned experiments are used to investigate the sensitivity of some dependent quantity (response) to the independent variables.

Rogers (1994) compared the results of four different methods, namely, *hypercube, linear, PROSSS,* and *random*. The hypercube method builds a hypercube around the midpoint of the design space, which corresponds to the initial or nominal values of the design variables. The best choice of points for the hypercube is points at each of the corners (2^{n_V} points for n_V design variables), the midpoint for each face ($2 \cdot n_V$ points), and the midpoint of the design space (the initial design). However, if the corner points are chosen, the number of required fitting pairs quickly becomes very large as n_V

increases. Therefore, it might be better to try to represent the design space by choosing only the midpoints of the faces. The linear method creates the training pairs starting at the lower bound of each design variable and stepping through the design space until reaching the upper bound. PROSSS (*Programming System for Structural Synthesis*) is a software that is designed to integrate the *Engineering Analysis Language* (EAL) structural analysis program and CONMIN optimization program. In PROSSS method, the user first executes the optimization program CONMIN for a few cycles to optimize the structure in which the user is interested (CONMIN does not reach the optimum). A new design is analyzed in each iteration using EAL. PROSSS uses those intermediate designs and their responses obtained from FEA to train the neural network. In random method, designs are created using a random number generator with no values allowed outside the lower and upper bounds of the design variables.

Taguchi method is a design of experiments method that is very popular in the industry. In the Taguchi method, a set of orthogonal matrices has already been prepared. The different values in those matrices correspond to the different levels of the design variables. For example, in a two-level matrix, the numbers 1 and -1 in each position of the matrix can denote the two levels of each design variable such as high or low, loose or tight, and good or poor.

Taguchi method uses simple charts to describe how to design an experiment, and how to extract the information needed from the experimental results. The user just needs to follow a procedure in a cookbook without knowing statistics theory and details behind this method.

For a model with n_V variables, there are 2^{n_V} different combinations if each variable only has two levels, and 3^{n_V} different combinations if each variable has three levels. In Taguchi method, only a fraction of all possible combinations are used in designing the experiment. For example, consider that there are 3 two-level variables, and we only want to get the effect of each variable on the output, and that we are not interested in the effects of the interactions of different variables on the output. Only four experiments are needed instead of nine if one wants to consider all combinations of values of the variables. The

total number of experiments required will increase very quickly if the user needs the effects of the interactions of different variables on the output. However, the method is still more efficient than using all possible combinations of variables.

The hypercube, the Taguchi method or other design of experiments methods require the stiffness and mass of joints for many predetermined combinations of values of the joint dimensions that correspond to infeasible designs. Many of these designs violate manufacturing, styling, and packaging constraints, while others have unreasonable geometry and cannot be modeled using a parametric model. Based on these considerations, the random method is used to create designs. This method uses a random number generator to create values of design variables corresponding to joint designs. Each joint design is checked against the constraints and those designs violating the constraints are rejected. The remaining designs are analyzed using a finite element program, such as MSC/NASTRAN to estimate their performance characteristics. These estimates are considered as actual values of the performance characteristics when developing translator A.

Several thousand designs are typically checked. The dimensions of all the feasible designs and their performance characteristics are stored in the database that will be used later to create the polynomial and neural network translators.

Creating the database is one of the most time-consuming parts of developing translator A. Generally, it takes 2-4 minutes to check if a design is feasible. If a design is feasible, it takes 2-3 more minutes to create the FEA mesh and a NASTRAN bulk data file. If a design causes the parametric model to crash, it takes 1-2 minutes to recover the model. It takes 4-5 minutes to analyze a design using a finite element program like MSC/NASTRAN to get its performance characteristics. For a typical joint with 40-50 design variables, the database contains 500-1000 designs. The author estimates that it takes two to three months to develop such a database.

2.4 Developing and Validating a Response Surface Polynomial

The degree of a polynomial depends on the relation between the design variables and the performance characteristics. This study *uses linear polynomials, second degree polynomials* and *double regression models,* to create the translators. Double regression, first approximates the performance characteristics as a function of the design variables using a polynomial and then uses another polynomial of the estimated values of the performance characteristics to obtain better estimates of these quantities. *Stepwise regression* is used throughout this study to obtain the best regression model. The best model is the model that fits the data well and contains only important terms, that is terms that affect significantly the response. This section describes how to develop a response surface polynomial. Specifically, it will describe the following tasks:

- Identification of important design variables.
- Construction of regression models.
- Comparison of different regression models, and validation of regression models.

2.4.1 Determining the Most Important Design Variables

Generally, many design variables are needed to completely describe the geometry of a joint. For example, the B-pillar to rocker joint considered in this study has 52 design variables, while the A-pillar to roof rail joint has 48 design variables. Among all these design variables, only a few significantly affect the performance characteristics (stiffness and mass) of the joint. Therefore, to develop an efficient polynomial and neural network that simulates the relations between the design variables (input) and the stiffness or mass (output), it is important to determine those design variables that affect significantly the stiffness and mass.

One can rank design variables in terms of importance by comparing all possible polynomials with all possible combinations of design variables. This approach is obviously impractical in most real life design problems with many design variables (more than 30). It is also impractical to use stepwise regression on a complete second degree polynomial. For example, a complete second degree polynomial with 50 design variables

has 1326 terms. Since stepwise regression considers many combinations of these terms in each step, it is impractical for such polynomials. To overcome these difficulties, we use stepwise regression on a linear polynomial to identify and rank the most important design variables as described below. First, we find the design variable that has the biggest effect on error between the predicted results and FEA results and include this design variable in the model. For this purpose we compare single-variable models, each containing one design variable, where n_V (= *p-1* for a linear polynomial) is the number of design variables. Then we find the next most important variable among the remaining design variables that when included in the model would result in the biggest reduction in the error and add it in the regression. This process is repeated until all important variables enter in the model. The improvement in the overall fit resulting from adding a design variable is measured by the corresponding significance probability. This is the probability that the actual effect of a term on the fitting error is zero and consequently the observed improvement in the overall fit is due to luck. In every step, a hypothesis test is performed to decide when to stop. Specifically, we test the hypothesis that the actual effect of a parameter on the overall fit is zero. If the reduction in the overall error is statistically significant, the parameter is added to the model. Otherwise, we conclude that there are no important parameters left and the procedure stops. Theoretically, the obtained regression model is the best linear model, and it contains the most important design variables.

In every step of linear stepwise regression, the *F*-ratio of each design variable is calculated. Section 2.4.3 gives the definition of *F*-ratio. *F*-ratio can be used as a measure of the importance of design variables: the variable with bigger *F*-ratio has bigger effect on the output. According to the *F*-ratios, the design variables are listed from the one with the biggest *F*-ratio to the one with the smallest *F*-ratio. This list is used to compare the importance of the design variables, and will be used throughout the following sections.

2.4.2 Choosing a Polynomial Regression Model

Before creating a response surface polynomial, we select its degree. Linear polynomials can only simulate linear relations between the independent and dependent variables. Higher degree polynomials are more flexible in simulating complicated input-

output relations. On the other hand, higher degree polynomials can have a large number of parameters. If we do not have enough designs for fitting, then the generalization performance will be poor. The reason is that some coefficients cannot be predicted with high confidence using a reasonable number of examples. The total number of parameters in the regression model will increase exponentially as we increase the degree of the polynomial. Moreover, a higher degree polynomial may become ill-conditioned. Another drawback of using a high degree polynomial is that we will have less confidence in the parameters of the polynomial when more parameters are used. Because of these reasons, generally, we use polynomials with degree higher than three in a few cases only.

In this study, linear, second degree polynomial models, and double regression models are considered.

Linear Polynomials

In general, the most parsimonious regression polynomial that fits the data well is the best. Therefore, a linear polynomial should be used if the relation between the independent and the dependent variables is almost linear. This is the case in many real life design problems because there are many constraints that do not allow the design variables to vary outside a narrow zone (Hughes, 1983).

Second Degree Polynomials

Linear polynomial models can only simulate linear relations between the independent and dependent variables. The fitting and testing results from the linear regression models are not always accurate. Therefore, we consider second degree polynomials for developing translators for some performance characteristics.

The most straightforward way of using second degree polynomials is to consider all design variables, all the interactions, and the square terms of all design variables in stepwise regression. Such a polynomial includes $C_{n_V}^0 + 2C_{n_V}^1 + C_{n_V}^2$ (here $C_{n_V}^j = \frac{n_V(n_V-1)\cdots(n_V-j+1)}{j(j-1)\cdots 1}$) terms, which means that we need at least $C_{n_V}^0 + 2C_{n_V}^1 + C_{n_V}^2$ designs to determine the coefficients of the polynomial. For the B-pillar

to rocker joint, the number of design variables, n_V, is 50. It is impractical to use a complete second degree polynomial because such a polynomial requires at least 1326 designs to determine the coefficients.

To address this problem, we rely on the ranking of the design variables obtained using a linear polynomial. Figure 2.2 explains this process. First, we consider only a small group of the most important design variables found using a linear polynomial. This small group can have about five design variables. We create a complete second degree polynomial using these few most important design variables, and use stepwise regression to determine which terms should be included in the regression model. The stepwise regression yields an incomplete second degree polynomial in which only the important terms are left. Then, we add the next most important design variable among the remaining design variables and its square to the model according to the ranking determined by the linear polynomial. The new regression model considers the new design variable, the square of the new design variable, the remaining terms from the former regression, and the interactions between the new design variable and former design variables. Using stepwise regression, the best regression model can be found again. This process is repeated by adding one variable at a time to the model. At each step, the accuracy of the predictions of the polynomial increases, but the rate of improvement reduces as we add more parameters. We stop when we reach the point of diminishing returns where the improvement in the accuracy becomes insignificant as we add more variables.

Double Regression Models

When creating the polynomial translators for some joints, it is found that both linear and second degree polynomials are inaccurate. The performance characteristics can be more accurately predicted if we use a second regression to predict the performance characteristics as a function of the predictions of a linear regression model. We can identify this case by observing a scatter plot of the exact values of the dependent variables (performance characteristics of the joint) versus the predictions from the linear regression model. For example, as Figure 2.3 shows, the predictions from a linear regression model are not evenly distributed along the two sides of a straight line inclined at 45 degrees

relative to the horizontal axis. Instead, the predictions are almost evenly distributed along the two sides of a cubic/quadratic polynomial of which the predicted value of the performance characteristics are independent variables. Based on this observation, a second regression is performed that uses a cubic/quadratic polynomial to simulate the relation between the predicted results from the linear regression model and the actual performance characteristics. The following is the formulation for the second regression (Fig.2.3).

$$\hat{y}_D = \sum_{i=0}^{n_D} c_i \, (\hat{y}_L)^i \tag{2.1}$$

where $\hat{y}_D$ and $\hat{y}_L$ are the values of the dependent variables predicted from the double regression and linear regression models, respectively. n_D is the degree of the double regression model. Coefficients c_i in above formulation can be determined using regression. Note that, in some cases, double regression significantly improves the prediction when compared with the linear regression model and the second degree models.

2.4.3 Criteria for Selecting Polynomial Translators

When creating the polynomial translators, only a subset of all design variables is considered in the regression model. It is important to know which design variables and what interactions between design variables should be considered in the regression model. This study uses four criteria to determine whether a regression model is appropriate in terms of its generalization performance.

- *Significance probability*
- *Determination factor,* R^2
- C_p *criterion*
- *AIC* (*Akaike*'s Information Criterion)

We describe each criterion in the following.

Significance Probability

Significance probability is the probability that the actual effect of a term on the fitting error (error between the predicted values of some performance characteristic and the calculated values of this performance characteristic from FEA) is zero. Consider that we want to decide if we should add a particular term to a given polynomial model that contains *p-1* terms plus the constant term. The significance probability associated with that term is calculated by finding the value of the *F*-ratio:

$$F = \frac{SSR_{im} - SSR_r}{SSE_{im}/(n_f - p)} \tag{2.2}$$

where n_f is the number of designs used in fitting the polynomial. Subscript *im* denotes the model with the additional term (improved model), and *r* denotes the original model without the term (reduced model). The quantity, n_f *-p*, is called *number of degrees of freedom* of the improved model. *SSR* is the *regression sum of squares*:

$$SSR = \sum_{i=1}^{n_f} (\hat{y}_i - \bar{y})^2 \tag{2.3}$$

where, $\hat{y}_i$ is the response of the i^{th} design predicted by the polynomial and $\bar{y}$ is the average of the responses of the n_f designs. *SSE* is the *sum of square errors* of a polynomial:

$$SSE = \sum_{i=1}^{n_f} (y_i - \hat{y}_i)^2 \tag{2.4}$$

where y_i is the measured response of the i^{th} design (in this study, y_i is the response calculated using FEA).

The *F*-ratio can be used to test the hypothesis that the additional term is actually zero (this is called *null hypothesis*). If this were the case, then *F* would follow the F_{1,n_f-p} probability distribution, which has been tabulated in many books (e.g., Draper and Smith,

1981). A large observed value of F (say 10) indicates that the model will be significantly improved when the p^{th} term is added to the model. This is evidence that this hypothesis is wrong and consequently that the additional term is non zero. Indeed, if the null hypothesis were right, then it would be very unlikely for F to assume such a high value. To determine if the additional term is important we calculate the significance probability, which is the probability of a variable, which follows the F_{1,n_f-p} distribution, exceeding the observed value of the F-ratio (Eq. 2.2). A low probability (say 5%) shows that the additional parameter should be non zero.

The significance probability is used to determine how important a term should be in order to be included in the regression model. A significance probability that is too high will result in including terms that are not statistically important, and will reduce the generalization performance of the polynomial. On the other hand, using a significance probability that is too low will not allow us to include all the important terms in the regression model. In most applications, a significance probability between 1% and 10% is used. In our work, we use a 5% significance probability.

The determination factor, R^2

R^2 expresses the percent of the variation in the response of the designs in the database from the mean response that is explained by the model:

$$R^2 = \frac{SSR}{SSTot} = \frac{SSTot - SSE}{SSTot} \tag{2.5}$$

where *SSTot* is the total sum of squares of the variations of the measurements from the mean. *SSE* is the sum of square errors defined in Eq. 2.4.

A model with a value of R^2 close to one is accurate because it explains the biggest portion of the variation in the response of all the designs used for fitting.

When using R^2, we only include in the model those terms that increase significantly R^2. Those terms that result in small increases in R^2 are not included in the model because

they will corrupt the generalization performance even though they can slightly improve the fitting results.

C_p criterion

C_p criterion is an alternative statistic to determine the predictive capability of a polynomial. It is calculated using the following equations:

$$C_p = \frac{SSE}{s^2} - (n_f - 2p) \tag{2.6a}$$

where

$$s^2 = \frac{\sum_{j=0}^{n_f} (\hat{y}_j - \bar{y})^2}{n_f - p} \tag{2.6b}$$

s^2 is an estimate of the variance of the error in the measurements, which is also called surface-fitting sample variance. If we use data from FEA, obviously, there is no measurement error. However, we can still use Eq. 2.6b and calculate s^2 from Eq. 2.6b. C_p is compared to the number of terms, p, of a polynomial to find if the polynomial includes all the important terms and only these terms. If C_p is considerably larger than p, this indicates either that the polynomial does not include all important terms or that it includes redundant terms. If we start with a polynomial with very few terms, then C_p decreases as we add important terms. The reason is that this increases the model accuracy and reduces *SSE* significantly. However, as we continue adding terms we reach a point where redundant terms enter the model. This means that the function representing C_p in terms of p will flatten out or will start increasing (Eq. 2.6a). We should stop adding terms when the value of C_p first approaches the number of terms in the polynomial.

Akaike's Information Criterion, *AIC*

AIC is defined by the equation below:

$$AIC = n_f Log(SSE / n_f) + 2p \tag{2.7}$$

where n_f is the total number of fitting pairs, and *p* is the number of parameters in the polynomial, including the constant.

AIC is similar to the definition of *SSE* (Sum of Squared Errors) except that *AIC* also takes into account the effect of *p*. The model with the smallest *AIC* value is the best.

After creating a polynomial translator by using stepwise regression, the standard deviation is used to test the generalization performance of the obtained translator.

Standard Deviation

Standard Deviation is used to compare the accuracy of different models. It is defined as

$$\sigma = \sqrt{\left(\sum_{i=1}^{n_f} (r_i - \bar{r})^2\right) \Big/ (n_f - 1)} \tag{2.8a}$$

where

$$\begin{aligned} r_i &= \hat{y}_i / y_i \\ \bar{r} &= \frac{1}{n_f} \sum_{i=1}^{n_f} r_i \end{aligned} \tag{2.8b}$$

y_i and $\hat{y}_i$ are measured and predicted values of the response, respectively. r_i is the ratio of the predicted value over the measured value. An unbiased model has $\bar{r} \approx 1$. A precise model has a standard deviation close to zero because $r_i \approx \bar{r}$ for every design in the training or testing set.

2.4.4 Validation of Polynomials

After obtaining a polynomial, it is important to assess its accuracy when using it to predict the performance characteristics of new designs. As explained in the former sections, a model may fit the data accurately but have poor generalization performance mainly because it contains unimportant terms. Using stepwise regression and the above

criteria usually ensures that a polynomial has good generalization performance. This allows using the entire database of examples for fitting. To be on the safe side, one may want to use a portion of the database for validation of a polynomial model.

To test the generalization performance of a polynomial. This study splits the database into two groups. One group is used for fitting. Once a polynomial is obtained, we use the other group to test its performance. The standard deviation of the predicted over the actual values is calculated. The polynomial with the best generalization performance should have the smallest standard deviation for the testing results.

2.5 Developing and Validating a Neural Network

Neural networks have been studied for many years in the hope of achieving human-like performance (Lippmann, 1987). Neural networks consist of many linear and nonlinear computational elements called *neurons*. Neurons are connected via *weights* and *bias* that are adapted during the training process to improve the generalization performance of the neural network (Fig. 2.4). Neural networks can be used to simulate complex mapping relations (Hajela and Berke, 1990).

In selecting a neural network, we have to select the architecture of the neural network (for example, the number of hidden layers and the number of neurons in each hidden layer), the transfer functions of the neurons, and the training algorithm for determining the parameters of the neural network so that it can predict the response of a given structure (Lippmann, 1987, Alon *et al.*, 1991). The multi-layer perceptron is the most widely used static network model. The back-propagation training algorithm is selected because it is the most popular one and has been implemented in many commercial software packages.

Figure 2.4 shows a typical multi-layer neural network. Most applications use neural networks with one or two hidden layers to approximate input-output relations. Cybenko (1989) showed that, a neural network with one hidden layer can approximate any continuous function. However, a neural network with two hidden layers may require far

less neurons than a single hidden layer neural network in some cases. On the other hand, a neural network with two hidden layers may need much more training time. In this study, we found that a neural network with one hidden layer can predict the performance characteristics of a joint with acceptable accuracy.

2.5.1 Transfer Function

The transfer functions of the hidden layer neurons and the output layer neurons are *sigmoid* and *linear functions*, respectively. One advantage of the sigmoid function is that it is differentiable. This property is important because it makes possible to derive a gradient search learning algorithm for networks with multiple layers. Another reason that the sigmoid function is popular is that, it varies monotonically from 0 to 1 as y varies from -∞ to +∞. It is particularly suitable for pattern recognition problems. Using linear function in the output layer makes it easier to change the output in a big range.

The *sigmoid function* can be expressed as:

$$f_i^k(x) = \frac{1}{1 + e^{-z_i^k / \theta}} \tag{2.9a}$$

where z_i^k is the input to the i^{th} neuron:

$$z_i^k = \sum_{j=1}^{n_{n,i}^k} w_{i,j}^k x_j^{k-1} + b_i^k \tag{2.9b}$$

Coefficient θ is used to control the steepness of the sigmoid function. The smaller the θ, the steeper the sigmoid function. θ is set to be one in this study. k denotes k^{th} layer. Subscript i stands for i^{th} neuron in a layer. $n_{n,i}^k$ is the number of inputs for the i^{th} neuron in the k^{th} layer. j denotes the j^{th} neuron of the former layer. $w_{i,j}^k$ is the weight between the i^{th} neuron of k^{th} layer and the j^{th} neuron of former layer. b_i^k is the bias of the i^{th} neuron in the k^{th} layer. x_j^{k-1} is the output from the j^{th} neuron of the $(k\text{-}1)^{\text{th}}$ layer (Fig. 2.4).

The *linear transfer function* has the following form:

$$f_i^k(x) = \sum_{j=1}^{n_{n,i}^k} w_{i,j}^k x_j^{k-1} + b_i^k \tag{2.10}$$

2.5.2 Choosing Inputs and Outputs of a Neural Network

To use a neural network to simulate the mapping between the design variables and the performance characteristics of a design, such as an automotive joint, first, we need to decide how many design variables to consider in the neural network. As explained in Section 2.4.1, among all the design variables that are used to describe the geometry of a joint, only a few affect significantly the performance characteristics. Therefore, it is important to consider only those design variables that affect significantly the performance characteristics.

First, the design variables are ranked in terms of importance using the results of stepwise regression with linear polynomials (see Section 2.4.1). According to the importance of design variables, first, we only consider a small group of few most important design variables in a neural network. This group can have five or six design variables. Then, we gradually increase the number of design variables in the neural network by adding one variable at a time. We terminate this process when the error of prediction of the testing results falls below a given value.

We can use one neural network to predict all performance characteristics of a joint and mass, or one neural network for each quantity. Since the important design variables for one performance characteristic are different from those for another characteristic, the first approach would result in one network with too many design variables. Some design variables of this network may affect only one output, but not the other outputs. On the other hand, the second approach only considers the important design variables for each performance characteristic and the mass. These networks can be more robust than a big network obtained by the first approach because they only contain the important design

variables. The drawback of the second approach is that it works with more neural networks than the first approach. This study follows the second approach because translator A must be robust.

2.5.3 Normalization of the Input-Output Sets

Some design variables, such as thickness, can only change less than 1 *mm*, while some other design variables, such as the height of a beam can change in a big range. The magnitude of most input design variables varies from *1* to 10^2, while the magnitude of the output varies from 10^7 to 10^8 for stiffness. The magnitude of mass is 10.

It is useful to normalize both the input and the output variables so that they can all vary in the same range [-1,1]. The reason is that by normalizing all variables into the same range we can avoid the problem that the output of the neural network is over sensitive to some variables and insensitive to others. Moreover, if the ranges of design variables and the performance characteristics are too big, this can cause numerical difficulties. Li and Parsons (1995) also observed that selecting an appropriate range in the magnitude could reduce the training time by an order of about three.

There are many methods to normalize the variables. Two methods are considered and tested in this study. In the first method, the parameters are normalized by dividing each variable by its mean value. The second method normalizes the parameters so that they can vary in the range [-1,1]. It was observed that the normalization method can have an appreciable effect on the final results. Generally, when using the first normalization method, the testing results converge much slower than when using the second normalization method (Fig.2.5). Figure 2.5 shows that the neural network reaches the minimum at about 1300 *epoches* when using the second normalization method. In neural networks, "*epoch*" refers to one iteration of the training process. On the other hand, it takes more than 5000 epoches to reach the minimum if the first normalization method is used. The reason is that the transfer function is the sigmoid function (Eq. 2.9), which varies between 0 and 1 when z_i^k changes from -∞ to +∞. [-1,1] is the activating range for the sigmoid function. The convergence speed is related to the derivative of the function $f_i^k(x)$ with respect to the inputs of the neurons, z_i^k. This derivative is bigger for the

same change of z_i^k when z_i^k is within the range [-1,1] than that when z_i^k is outside this range. Since in the second method, all input variables (design parameters of a joint) are normalized so that they can vary in the range [-1,1], the second method generally converges faster than the first.

2.5.4 Determining the Number of Neurons in the Hidden Layer

As mentioned earlier, this study uses a neural network with one input layer, one output layer, and one hidden layer. For our problem, the number of neurons in the input layer is equal to the number of design variables considered in the neural network. The number of neurons in the output layer is one for each neural network used to predict each performance characteristic. Only the number of neurons in the hidden can be changed.

The number of neurons in the hidden layer affects the generalization performance of the neural network. If there are too few neurons, the neural network will neither learn nor be able to generalize relationships from the training data. On the other hand, if there are too many neurons relative to the number of examples, then the neural network will memorize the training data but will not be able to generalize relationships.

The number of unknowns in a neural network is a key factor in choosing the number of neurons in a neural network. For example, a neural network with three layers, including the input and output layers, has $(n_1 + n_3 + 1)n_2 + n_3$ unknowns, including the weights and bias, where n_1, n_2, and n_3 are the number of neurons in each layer.

There are several methods to decide how many neurons to consider in the hidden layer. Some publications use a neural network that has more unknowns than the input-output pairs. Most papers select a neural network in a way that the total number of unknowns is less than the total number of the input-output pairs used in training (Carpenter and Barthelemy, 1992). The networks in these papers are *determined*. Carpenter and Barthelemy (1992) suggested that the number of neurons in the hidden layer should be between the sum and the average of the number of neurons in input and output layers. Hush *et al.* (1993) suggested starting with the smallest possible network

and gradually increasing the number of neurons until the performance of the network begins to level off if one has little or no prior knowledge.

Only *determined* neural networks are used in this study. That is, the number of hidden layer neurons is such that the total number of unknowns is less than the total number of designs for training. For relatively simply mapping relations, the number of design variables is changed from one to the maximum value for which the neural network is determined. The number of neurons in the hidden layer is chosen so that the total number of unknown neural network parameters is approximately equal to 80-90% of the total number of designs in the training set. For complex mapping relations, both the number of design variables and the number of neurons are changed from one to a maximum value to determine the best combination. The neural network with smallest testing error is selected as the translator.

2.5.5 Method of Determining When to Stop Training

First, the network is trained to implement a mapping that matches the examples in the training set as closely as possible. Back-propagation, which is an iterative process , is used. There are several stopping criteria. The first is based on the magnitude of the gradient of the total error in training with respect to the parameters (weights and biases). The iteration process is terminated when the magnitude of the gradient is sufficiently small, since by definition, the gradient will be zero at the minimum.

Second, one might stop the training when the sum of square errors, *SSE*, falls below a fixed value. However this requires to know the minimal value of *SSE*, which is not always available.

Third, one might stop the training process when a fixed number of iterations has been performed. However, there is little guarantee that this stopping condition will terminate the training when the sum of square errors of the testing results reaches its minimum.

Forth, the method of *cross-validation* can be used to monitor the generalization performance during training, and terminate training when there is no longer an

improvement. Cross-validation splits the data into two sets: a training set, which is used to train the network, and a test set which is used to measure the generalization performance of the neural network. During training, the performance of the neural network on the training data will continue to improve, but its performance on the test data will only improve to a point, beyond which it will start to degrade. The training process is terminated at this point, because the network starts to overfit the training data. Cross-validation is explained in Fig. 2.6. The solid and the dashed lines denote the standard deviations of the training and testing results versus the number of epoches, respectively. It is observed that the standard deviation of the training results consistently decreases when increasing the training time. However, the testing results only improve to a point (at about 1300 epoches). Beyond this point, the standard deviation of the testing results begins to increase.

Amari, Murata, and Muller (1995) discussed the problem of overtraining and cross-validation. They demonstrated that when $n_f > 30 n_m$, where n_m is the size of neural network (number of weights and biases), and n_f is the number of training pairs, cross-validated early stopping does not improve the generalization error. Indeed, no overtraining was observed on the average in this range. For most practical problems, this range of n_f is often inaccessible due to the limited size of the data pairs. The paper also demonstrated that early stopping does improve the generalization ability to a large extent for $n_f < 30 n_m$. This paper proposed that, for large n_m, only $1/\sqrt{2n_m}$ percent of input-output pairs should be used to determine the point of early stopping in order to obtain the best performance.

The first three criteria are sensitive to the parameters of the solution process, such as the minimum value of the total error, which are difficult to determine. Cross-validation, however, does not have this problem. This study uses cross-validation. The available designs are split into three sets: a training set, and two testing sets. The first testing set is used to determine when to terminate the training process, while the second testing set is used to test the generalization performance of the neural network after training has been completed.

An additional benefit of cross-validation is that the resulting neural networks are insensitive to the number of neurons in the hidden layer (Chapter 4).

2.5.6 Method of Testing the Generalization Performance of the Trained Neural Network

After training a neural network, we test its generalization performance by using it to predict the performance of designs that are not used in training.

The second testing set is used to test the trained neural network. After getting the testing results, the ratio of the predicted result over the FEA result can be calculated. The standard deviation of this ratio (Eqs. 2.8a and 2.8b) is used to measure the accuracy of the neural networks and decide if a neural network is acceptable. The same measure is used to compare the results from neural networks with different architectures (number of neurons in hidden layer, and number of input variables). The neural network with the smallest standard deviation for the testing results is chosen as the neural network translator. We also compare neural network with response surface polynomials according to their standard deviations.

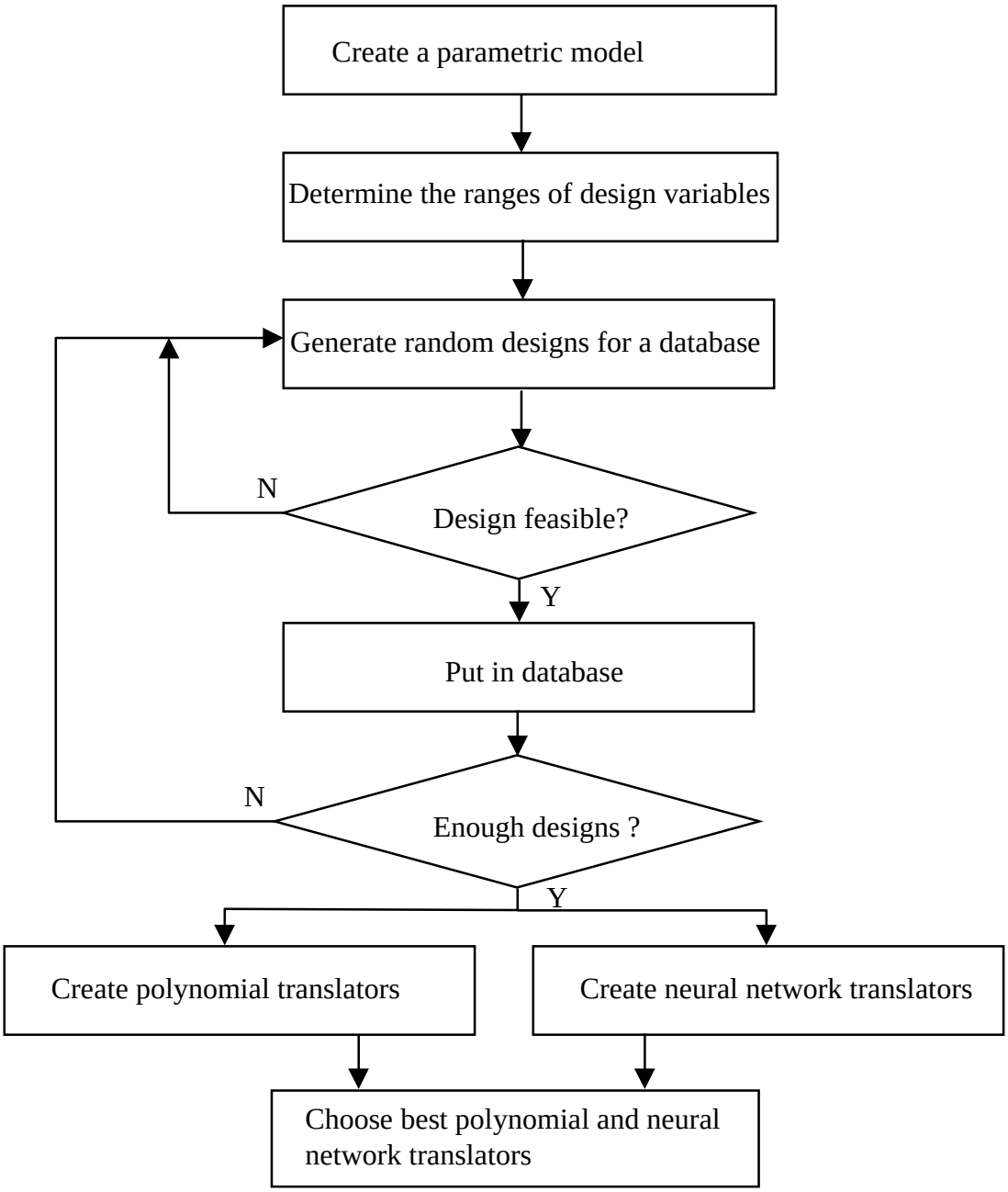


Figure 2.1: Method for Creating Translator A

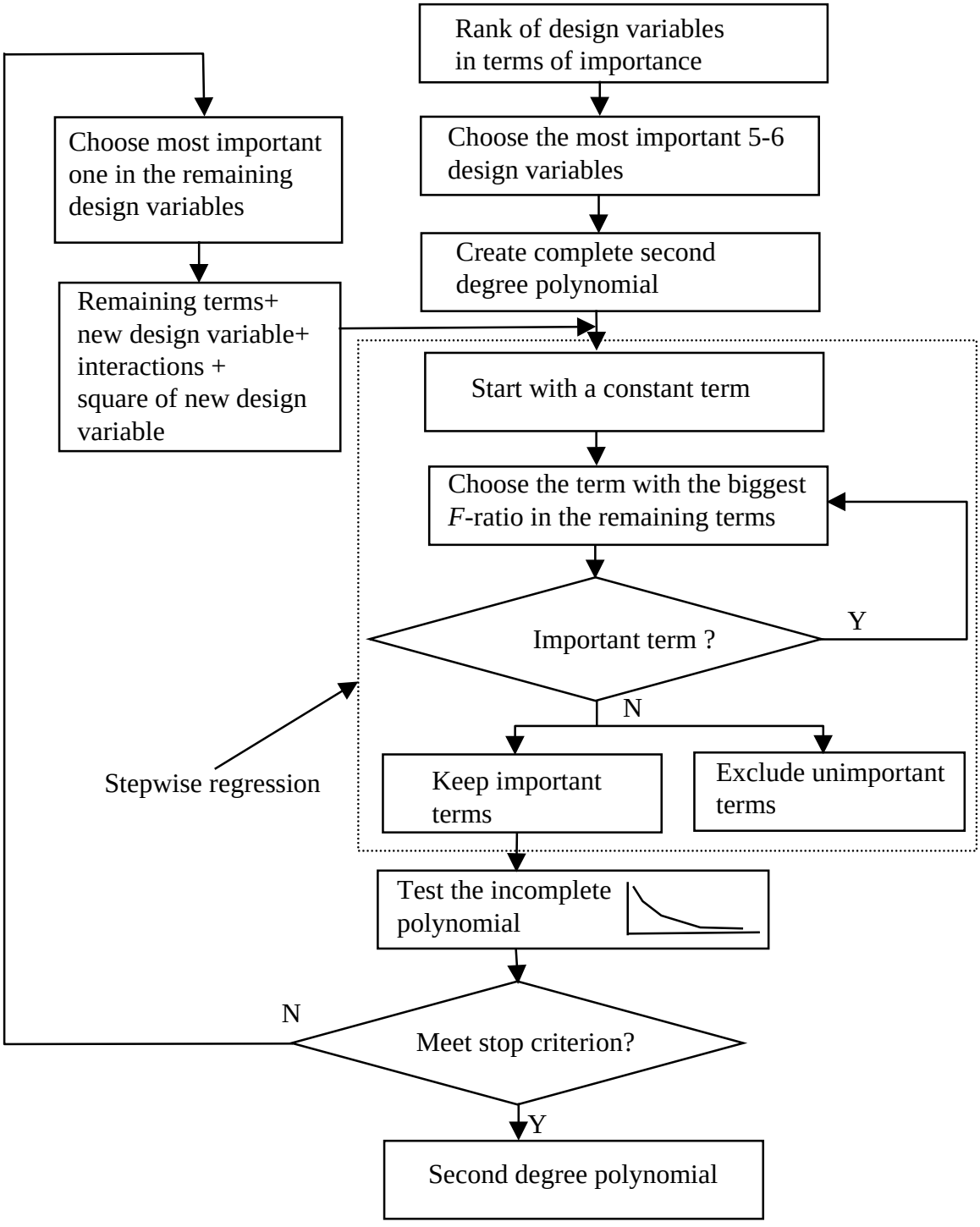


Figure 2.2: Method for Developing Translator A Using Second Degree Polynomials.

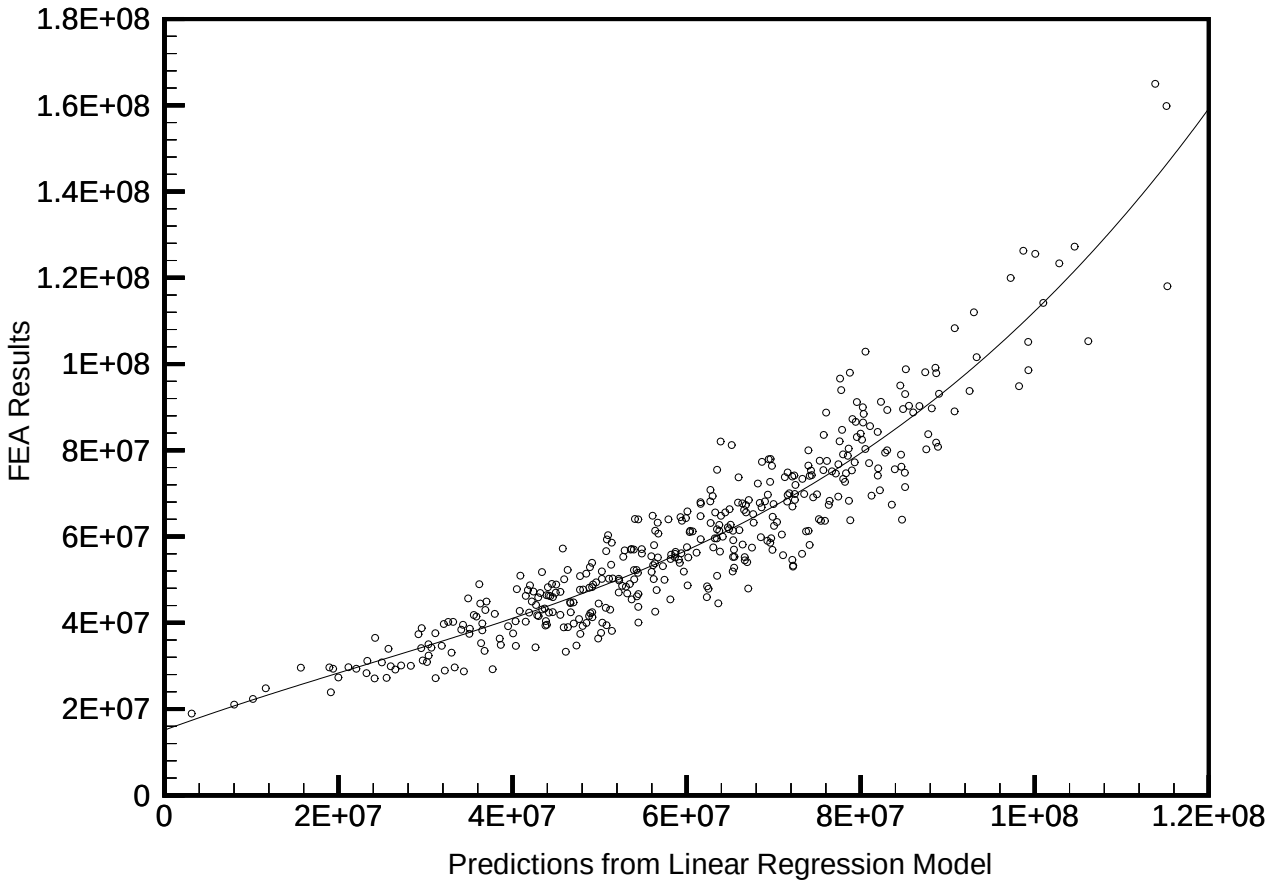


Figure 2.3: Comparison of Predictions from Linear Polynomial Model and FEA Results and Explanation of Double Regression

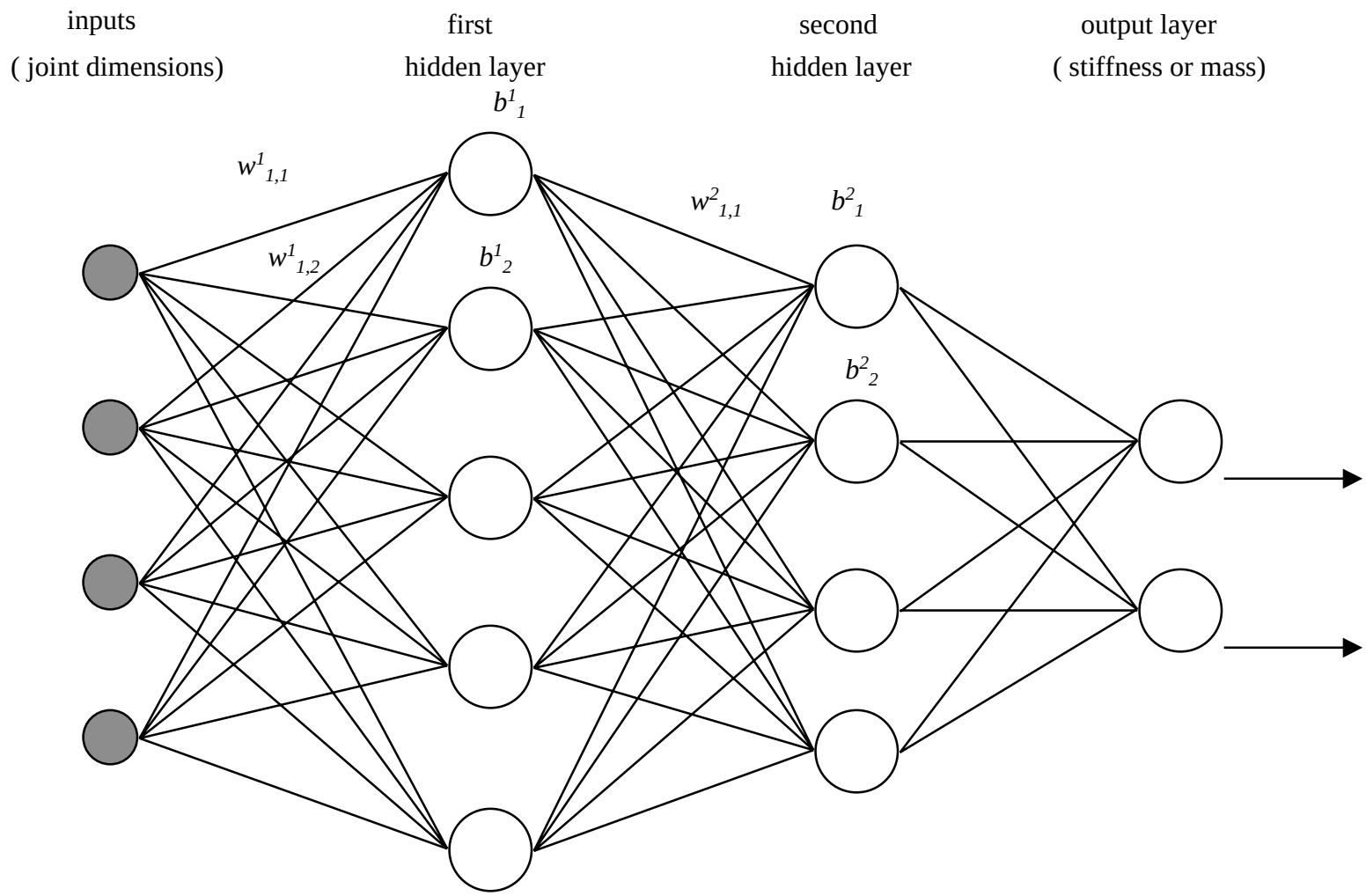


Figure 2.4: Architecture of a Typical Multi-layer Neural Network

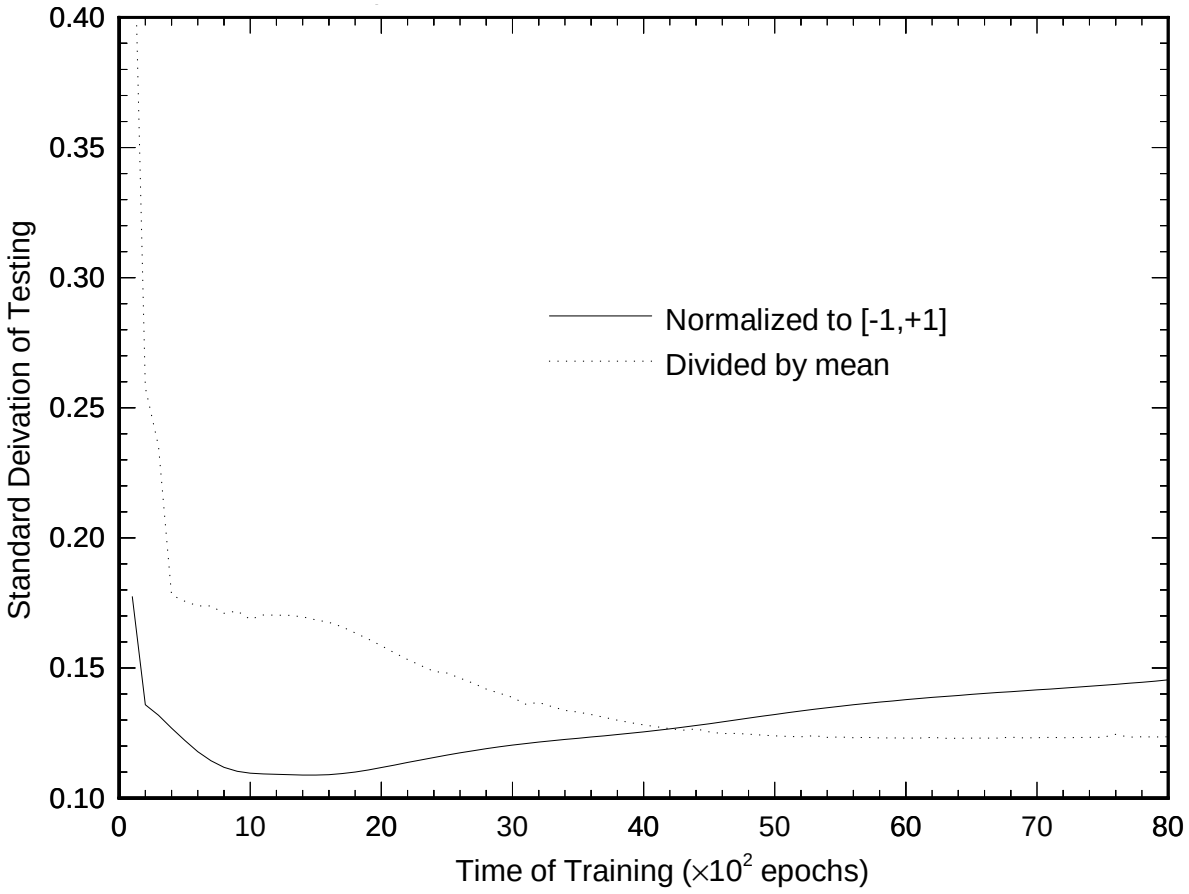


Figure 2.5: Comparison of Two Different Normalization Methods

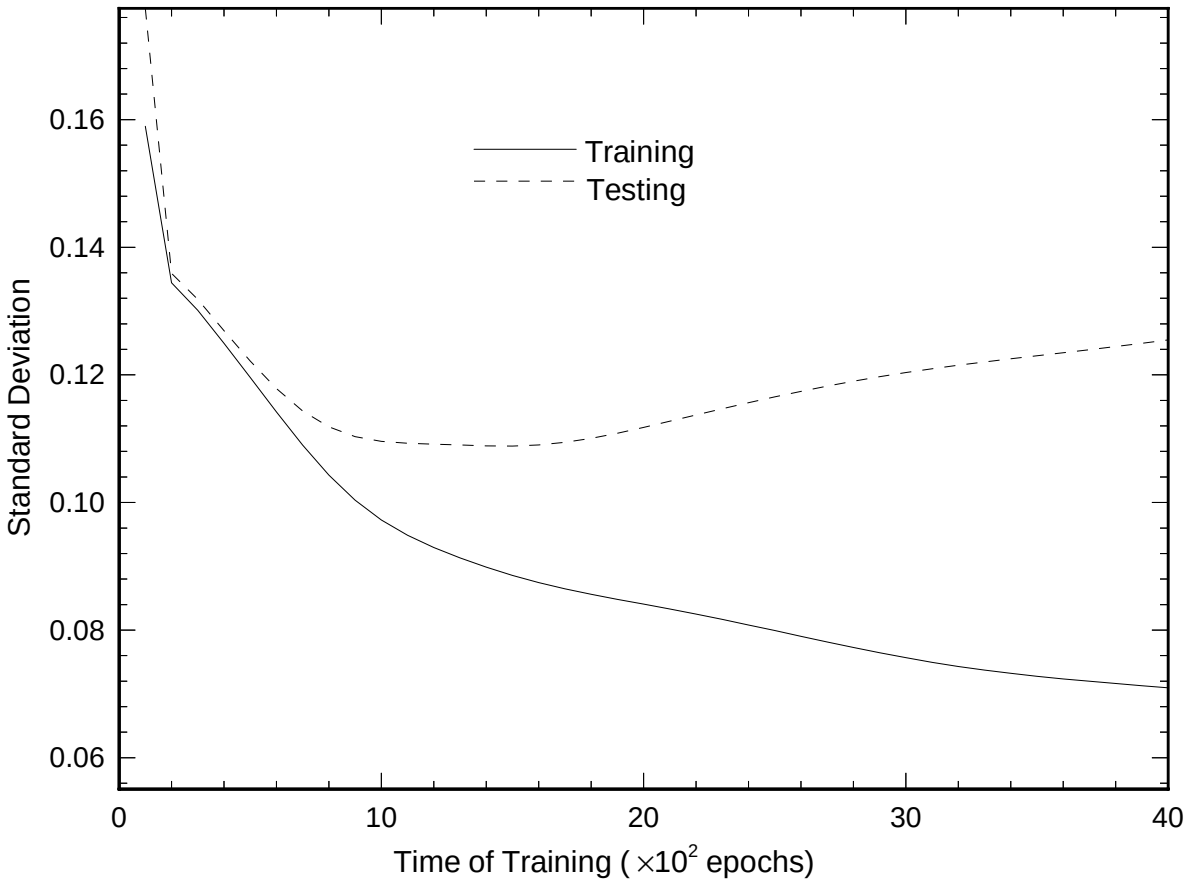


Figure 2.6: Relation Between Standard Deviation and Time of Training

Chapter 3
Methodology for Developing Translator B

3.1 Introduction

Translator B is a tool that finds the most efficient, feasible joint design that meets given performance targets. An efficient joint has the lowest mass and its stiffness is as close as possible to a given target. In this study, we refer the most efficient feasible design as the optimum joint design. This tool is important at the early design stage because it links performance requirements for a joint to the design variables and mass of a joint. As explained in Chapter 1, this tool will allow designers set up meaningful performance targets.

Translator B finds the optimum design using numerical optimization methods. The objective function to minimize can be the mass of a joint or some combination of the mass and stiffness. Design variables include both the dimensions of a joint and variables that specify FEA information (e.g., plate thicknesses). Constraints can be classified under packaging, styling, manufacturing, mathematical, and performance target constraints.

Section 3.2 describes the formulation of the optimization problem for translator B. Section 3.3 explains and compares two optimization algorithms used in this study. Section 3.4 studies the verification and validation of the optimization results.

3.2 Formulation of the Optimization Problem

A general optimization problem can be expressed in the following form.

Find $\mathbf{X} = \{x_1, x_2, \cdots, x_{NDV}\}^T$

To minimize $F(\boldsymbol{X})$ (3.1a)

Subject to

$$\begin{aligned} & g_j(\mathbf{X}) \le 0,\ j = 1, NCON \quad \text{(Inequality constraints)} \\ & x_i^L \le x_i \le x_i^U, i = 1, NDV \quad \text{(Side constraints)} \end{aligned} \tag{3.1b}$$

where $\boldsymbol{X}$ is the vector of design variables. x_i is the i^{th} design variable. x_i^L and x_i^U are the lower and upper bounds of i^{th} design variable. *NCON* and *NDV* are the number of constraints and the number of design variables, respectively.

There is no equality constraint in equation 3.1b. The reason for this is that the optimization program does not directly handle equality constraints. The program always converts an equality constraint into two inequality constraints. For example, the program converts the following equality constraint:

$$h(\boldsymbol{X}) = 0 \tag{3.1c}$$

into two inequality constraints:

$$\begin{aligned} & h(\boldsymbol{X}) \le 0 \\ & h(\boldsymbol{X}) \ge 0 \end{aligned} \tag{3.1d}$$

Another way of handling equality constraints is finding an expression of a design variable in terms of other design variables from the equality constraint. This method can reduce the number of design variables of the optimization problem. Therefore, it can increase the speed of convergence and improve the reliability of the optimization program. Unfortunately, it is not always possible to find an explicit expression of a design variable from an equality constraint.

3.2.1 Selection of Design Variables

Design variables are physical design variables that define the geometry and specify FEA information. These design variables are linked to the performance characteristics of the joint through translator A. 50-200 independent design variables are generally needed to completely describe a joint. As explained in Chapter 2, among all the design variables that are needed to completely describe a joint, only a few affect significantly its performance characteristics. Therefore, it is important to consider only these important design variables in optimization to reduce the size of the problem and avoid numerical problems.

Solving an optimization problem with even a modest number of design variables can be very difficult if the gradients of the objective function and constraints with respect to design variables are very small or zero. Moreover, only a few design variables affect significantly the objective function. Among all design variables, some are generally fixed by the designer because of manufacturing, packaging, or styling considerations. For example, the length of a branch, the width of flange and the distance between two adjacent spot welds are often fixed because of manufacturing consideration. Member lengths are fixed so that different joints can be fairly compared. Some design variables are not independent for a particular type of construction. For example, the depth of the pillar reinforcement can be determined from other design variables.

In translator B, design variables are divided into four types:

- Type 1: design variables that are *fixed by the optimization program*, such as the width of a flange.
- Type 2: design variables that are *fixed by the user*, such as the orientation of a branch. These variables are usually specified from the general shape of the car body and they cannot change in optimization of a joint.
- Type 3: *dependent design variables* whose values can be derived from other design variables. The values of dependent design variables are automatically updated when other variables change.
- Type 4: *independent design variables* that can change in optimization.

The number of variables that are fixed by the optimization program and the number of dependent variables are fixed for a particular joint. The user specifies which variables are fixed.

The user of translator B can easily change the numbers of independent and fixed variables by moving variables from one group to another. To do so, the user needs to change the input file only. This feature makes the optimization program easier to use and more effective than an optimization program that requires changing the source code to move a variable from one group to another. Consider that the user wants to optimize both plate thicknesses and shape variables. In practice, the thickness of a plate assumes discrete values because of manufacturing considerations. The user can initially optimize both the shape variables and thicknesses by treating them as independent variables. Then, the user can fix each plate thickness at the allowable value that is closest to the optimum. An improved design can be found by only optimizing the shape variables in this step.

This feature of translator B has some additional benefits:

- The user may not be interested in changing all design variables to get the optimum. Instead, the user can change only these design variables that should affect significantly the mass and stiffness.
- For a particular joint, the values of some design variables are derived from styling or other requirements, and are usually fixed. Although the user can not change those design variables in some cases, the user may want to perform a parametric study to determine the effect of releasing these variables on the optimum design. Translator B allows to perform such parametric studies easily.

3.2.2 Objective Function

Objective function is the target that the user wants to minimize. In our study, the mass of the joint can be chosen as the objective function. The best design should have the minimum mass among all designs that satisfy stiffness requirements (i.e. larger than the required stiffness) as well as manufacturing, packaging, styling and mathematical constraints.

However, in practice, a designer may want to design a joint that not only has low mass but also has performance characteristics (stiffness) close to some performance targets. These targets are determined from the optimization of the overall car body using a concept model or they are derived from experience. To account for this situation, the objective function is more generally defined as the weighed sum of mass and some measure of the difference of between performance targets and the corresponding performance characteristics of a design.

The general form of objective function can be expressed as follows:

$$F = \alpha M + (1-\alpha)\sqrt{((K_{I/O} - \hat{K}_{I/O})/\overline{K}_{I/O})^2 + ((K_{F/A} - \hat{K}_{F/A})/\overline{K}_{F/A})^2 + ((K_{Tor} - \hat{K}_{Tor})/\overline{K}_{Tor})^2} \quad (3.2)$$

where α is a weight that varies form zero to one. M is the mass of the joint. $K_{I/O}, K_{F/A}$, and K_{Tor} are the I/O, F/A and torsion stiffnesses of the joint. $\hat{K}_{I/O}, \hat{K}_{F/A}$, and $\hat{K}_{Tor}$ are the user-specified requirements (targets) for I/O, F/A and torsion stiffness. $\overline{K}_{I/O}, \overline{K}_{F/A}$, and $\overline{K}_{Tor}$ are nominal stiffness values. Generally, the mass varies from 2 to 8 kg. Each normalized stiffness term in the radical varies from 0 to 3. We normalize the stiffness so that the mass and the stiffnesses can have compatible magnitude.

α is set to be one if we only want to consider mass in the objective function. On the other hand, if it is more important for the stiffnesses to be close to their target values, then one should use a small value of α (i.e., $\alpha \cong 0.1$).

Translator A predicts the stiffness and mass of a joint in Eq.3.2. This makes translator B very efficient and robust because translator A can give accurate prediction almost instantaneously, even for the infeasible designs whose stiffness and mass can not be obtained using a parametric CAD model.

3.2.3 Constraints

In the design of a joint, the interaction between different components of the car, such as the position and shape of doors and seats, and the joint should be taken into account. The joint should be manufacturable and should meet the styling requirements. In general, we use many constraints (a few dozens to more than one hundred) to ensure the feasibility of the optimum design.

The constraints can be classified as packaging, manufacturing, styling, mathematical, and performance target constraints. Each type is explained below:

- Packaging constraints are related to the arrangement of car components in space. They are used to ensure that the joint is compatible with the adjacent components. For example, the width and the angle of the inner part of the rocker should be constrained within a specified limit to leave enough space for the seat. The dimensions *door_edge_height* and *door_edge_width* of the B-pillar to rocker joint should be large enough to accommodate the sealant.
- Manufacturing constraints are due to manufacturing limitations. They are used to ensure the parts of a joint can be manufactured from a piece of sheet metal. Manufacturing constraints in this study are classified into stamping and welding constraints. Stamping constraints include constraints for plastic strains, die lock and sprint back. For example, the angle between two adjacent plates can not be too small to avoid large plastic strains. The angle between the upper and lower edges of the rocker section should be constrained to avoid die lock. The depth of draw should not be too large relative to its width.
- Styling constraints are due to the styling requirements. For example, in B-pillar joint, the flange at the bottom of rocker section should not be visible to a person standing on the right side of the car.
- Mathematical constraints are used to control the range of different design variables to ensure that a design has a feasible geometry.
- Performance target constraints are the minimum values of the stiffnesses that the designer wants to achieve through optimization.

3.3 Optimization Algorithms used in Translator B

Because the objective function and some constraints explained in the former section are nonlinear, the optimization problem involved is a nonlinear constrained optimization problem.

We can use several methods, including, *Modified Feasible Direction Method (MFD), Sequential Linear Programming (SLP),* and *Sequential Quadratic Programming (SQP),* to solve a nonlinear optimization problem. Two optimization algorithms, *modified feasible direction method* and *sequential linear programming,* are used in this study.

The MFD is reliable and uses the least computer memory (Vanderplaats Research & Development, Inc, 1995). SLP is often most efficient for general applications in terms of the number of function evaluation required. SLP works well for problems that are almost linear. SQP is considered theoretically best if the optimization problem is "well conditioned". SQP is good for problems in which the objective function and the constraints can be approximated by quadratic functions. In many real life problems, design variables vary in a very narrow band because there are many constraints (e.g., a few dozens to hundreds). SLP and SQP work well in these problems because in the region of the design variables, the objective function and the constraints are close to a linear or quadratic surface.

Only the MFD and SLP are considered in this study. SQP is not used in developing translator B. The reason for this is that because some constraints are linearly related to design variables, it is not efficient to use sequential quadratic programming. It was also found that, when using sequential quadratic programming, the optimization program could not find a feasible design in a few cases when the starting point is in the infeasible domain (even though a starting design satisfies all other constraints, it may not satisfy the stiffness requirements, and thus it is an infeasible design).

In the following we describe the MFD and SLP methods:

3.3.1 Modified Feasible Direction Method

The *modified feasible direction* (MFD) method is a powerful general method, and can be applied to most constrained nonlinear problems. MFD is a modification of the classical steepest descent method. In the steepest descent method, the search direction is the gradient of objective function and the active and/or violated constraints. During each iteration, the search direction is always perpendicular to the former search direction, which makes the steepest descent method very slow to converge when the design approaches the optimum.

The conjugate search direction represents a simple modification to the steepest descent algorithm, but provides a major improvement in efficiency. When using conjugate directions, each search direction uses the steepest descent direction plus some fraction of the previous search direction. This algorithm is extremely simple but it dramatically improves the rate of convergence to the optimum (Vanderplaats Research & Development, Inc, 1995).

The modified feasible direction method takes into account not only the gradients of objective function and the retained active and/or violated constraints, but also the search direction in the former iteration. Let $\boldsymbol{X}^0$ be an initial $\boldsymbol{X}$ vector. We update the design according to the following equation:

$$\boldsymbol{X}^q = \boldsymbol{X}^{q-1} + \lambda \times \boldsymbol{S}^q \tag{3.3}$$

where $\boldsymbol{S}^q$ is the search direction. λ is a scalar whose value is determined through a one-dimensional search.

Different optimization methods are characterized by different methods to determine the search direction $\boldsymbol{S}^q$. The search direction $\boldsymbol{S}^q$ in MFD is determined using the Fletcher-Reeves conjugate direction method when there is no active or violated constraint.

$$\boldsymbol{S}^q = -\nabla F(\boldsymbol{X}^{q-1}) + \beta \boldsymbol{S}^{q-1} \tag{3.4a}$$

where

$$\beta = \frac{|\nabla F(\boldsymbol{X}^{q-1})|^2}{|\nabla F(\boldsymbol{X}^{q-2})|^2} \tag{3.4b}$$

The method of finding search direction $\boldsymbol{S}^q$ becomes complicated when there are active and/or violated constraints (Vanderplaats Research & Development, Inc, 1995, Haftka *et al.*, 1992)).

3.3.2 Sequential Linear Programming

The basic concept of *sequential linear programming* (SLP) is very simple. First, a Taylor series approximation to the objective function and constraints are created. Then, this approximation is used for optimization, instead of the original nonlinear functions. This makes it easier to get the values of the objective function and constraints during the optimization process. Also, since the approximate problem is linear, the gradients of the objective and constraints are available directly from the Taylor series expansion.

The Taylor series expansion has the following form:

$$F(\mathbf{X}) \cong F(\mathbf{X}^{q-1}) + \nabla F(\mathbf{X}^{q-1})^T \delta\mathbf{X} \tag{3.5a}$$

$$g_j(\mathbf{X}) \cong g_j(\mathbf{X}^{q-1}) + \nabla g_j(\mathbf{X}^{q-1})^T \delta\mathbf{X} \qquad j \in J \tag{3.5b}$$

where

$$\delta\boldsymbol{X} = \boldsymbol{X}^q - \boldsymbol{X}^{q-1} \tag{3.5c}$$

and J is the set of active and/or violated constraints.

Note that everything in the above equations is constant except for the values of $\boldsymbol{X}^q$. These equations can be rewritten as:

$$F(\mathbf{X}) \cong F^0 + \nabla F(\mathbf{X}^{q-1})^T \mathbf{X} \tag{3.6a}$$

$$g_j(\mathbf{X}) \cong g_j^0 + \nabla g_j(\mathbf{X}^{q-1})^T \mathbf{X} \qquad j \in J \tag{3.6b}$$

where

$$F^0 = F(\mathbf{X}^{q-1}) - \nabla F(\mathbf{X}^{q-1})^T \mathbf{X}^{q-1} \tag{3.6c}$$

$$g_j^0 = g_j(\mathbf{X}^{q-1}) - \nabla g_j(\mathbf{X}^{q-1})^T \mathbf{X}^{q-1} \qquad j \in J \tag{3.6d}$$

Now the approximate optimization problem becomes:

Minimize $$F(\mathbf{X}^q) \cong F^0 + \nabla F(\mathbf{X}^{q-1})^T \mathbf{X} \tag{3.7a}$$

Subject to:

$$\begin{aligned} & g_j(\mathbf{X}^q) \cong g_j^0 + \nabla g_j(\mathbf{X}^{q-1})^T \mathbf{X} \le 0 \qquad j \in J \\ & x_i^L \le x_i \le x_i^U \quad i = 1, NDV \end{aligned} \tag{3.7b}$$

where

$$\begin{aligned} x_i^L &= x_i^q - \vartheta \,|\, x_i^q \,| \\ x_i^U &= x_i^q + \vartheta \,|\, x_i^q \,| \end{aligned} \tag{3.7c}$$

ϑ is a scalar multiplier. Its value is sequentially reduced during optimization.

The above linear approximate problem can be solved using different methods such as the simplex method. Some computer programs use MFD to solve this problem because it solves linear problems nicely and works well under most conditions (Vanderplaats Research & Development, Inc, 1995).

We recommend using both the MFD and SLP for optimization. If one method consistently provides better results than the other, it can be selected for translator B.

3.4 Validation of Results of Translator B

There are several possible problems in optimization, including:

- Optimization may not converge.
- Optimization may converge to a local minimum instead of a global minimum.
- Optimization may lead to a design that appears to satisfy performance constraints but is actually infeasible because translator A overestimates stiffness.
- The user may specify performance targets that cannot be achieved. For example, the stiffness requirements can be too high.

One reasonable approach is to start from a relatively simple version of the problem that has less design variables and constraints than the original design problem (this can be done by fixing more design variables and removing some constraints). After solving the simplified problem, we can solve more complicated versions of the problem by adding more design variables and constraints. Finally, we can solve the real problem with confidence.

Generally, nonlinear constrained optimization problems are computationally difficult to solve and suffer from the fact that the optimization program may converge to a local minimum instead of a global minimum. Several methods can be used to check the results of the optimization program.

- One way to check whether the global minimum is found is to solve the same problem from different starting points. If we always get the same solution, then we can have confidence that the global minimum is found.
- Another way is to use the optimum design as a starting point, and solve again the optimization problem to see if the optimum design changes. If the second optimization yields the same point consistently, we can conclude that the optimization procedure has converged.
- We check if we get the same optimum using different optimization algorithms to solve the same problem.

Whenever possible, we should also solve the same problem using different software package. For example, one can use a commercial Fortran optimization program and Mathematica (Maeder, 1997) to solve the same problem.

The above checks help ensure that the optimization does converge to the global optimum solution. We also need to check if the optimum design is actually feasible. This is important because in many design problems, the optimizer tends to take advantage of errors in predictive models (translator A in our study) and thus it yields a design that actually has high weight and unacceptable performance (low stiffness).

For this purpose, the optimum design is visually checked using a CAD program, such as Pro/Engineer, to see if the optimum design has a reasonable shape.

To validate the optimization results of translator B, we select a number of random performance targets (stiffness requirements). Using translator B, we obtain the optimum design corresponding to each set of stiffness requirements. Then, a generic CAD model is used to create the models, and generate FEA data files. These models are analyzed using a FEA program, such as MSC/NASTRAN, to obtain the performance characteristics (stiffnesses and mass) of the optimum designs. By comparing the optimization and FEA results, we can find if there is any general trend between the optimization and FEA results. We also calculate the correlation coefficients between the optimization and FEA results for the stiffness in different directions and the mass of the joint. In general, a correlation coefficient larger than 0.9 shows good agreement between predictions of translator B and FEA results.

We also use the stiffness of actual car joints as performance targets and find the optimum design that satisfies these targets. We compare the optimum design from translator B to the actual design. We try to identify and explain trends. We also compare the stiffness of the optimum design predicted by translator B with FEA results.

For an optimum design, the values of some design variables are close to the boundary of the region corresponding to the ranges of the design variables in the database. The user may only be interested in changing the design variables in a small range. In such case, the user can narrow the ranges of the design variables.

Finally, the relation between the objective function and some important design variables are studied by modifying the lower and upper bounds of these variables and optimizing the joint again. The results should be checked to identify trends and try to explain them. These results should give the designer useful information about improving the design.

Chapter 4
Developing Translator A for the B-pillar to Rocker Joint

4.1 Introduction

This chapter applies the general methodology for developing translator A to the B-pillar to rocker joint (Fig. 1.1).

This chapter is organized as follows: Section 4.2 describes the geometry of the B-pillar to rocker joint, and some common types of construction. Section 4.3 explains how to create a database for developing translator A. Section 4.4 explains in detail the development and validation of translators A for the B-pillar to rocker joint using both response surface polynomials and neural networks. Finally, Section 4.5 discusses the results of translator A for B-pillar to rocker joint.

4.2 Description of the B-pillar to Rocker Joint

The B-pillar to rocker joint, which lies between the front and rear doors, is a T- like joint (Figs. 4.1a, 4.1b and 4.2). The vertical branch is called B-pillar, and the horizontal branch is called rocker. The B-pillar supports the rear door. The front door latch is also attached to the B-pillar. The rear door of the car is connected to the B-pillar using hinge. The stiffness of the B-pillar to rocker joint affects significantly the overall stiffness of the passenger cabin of the car.

The performance of the B-pillar to rocker joint is characterized by its I/O, F/A and torsion stiffnesses. Figure 4.1b shows the definition of these stiffnesses.

Murphy (1995) established a sound, generic method for parameterizing joints. He created a parametric model for the B-pillar to rocker joint that accounted for most

important characteristics of the geometry of the joint. However, this model had the following limitations:

- Since the direction of the B-pillar was fixed relative to that of the rocker in the original model of B-pillar to rocker joint, the user could not change the orientation of the B-pillar.
- Murphy's model crashed frequently when changing the values of design variables because some methods he used in creating the model were not very robust. For example, he used a fixed length between spot welds to create these spot welds.
- In general, Murphy did not put enough constraints to control the ranges of dimensions.

This study improved the parametric model. The new procedure for developing the parametric model uses the relative length between spot welds to create the spot welds. The author added constraints on the parametric model of the B-pillar to rocker joint to ensure feasibility of a design, and provide guidance to the user for correcting the model. The author also changed the sequence of creating different entities in the model to make the model more reliable.

The generic model is briefly explained in the following sections. The figures with a "*" sign were extracted from Murphy's thesis.

Most B-pillar to rocker joints have at least five parts (Figs. 4.2-4.3b), namely,

- Front part of the B-pillar (*Front rocker*)
- Back part of the B-pillar (*Pillar back*)
- Center plate of the rocker (*Center plate*)
- Back part of the rocker (*Back rocker*)
- Pillar reinforcement (*Pillar reinforcement*)

The names in the parentheses are used in the rest of the dissertation and in the computer program for translator A.

4.2.1 Types of Reinforcements

To increase the stiffness of the B-pillar to rocker joint, different types of reinforcements are generally used in the joint, including,

- A pillar reinforcement that extends into the rocker and forms a box (Fig. 4.3a).
- A pillar reinforcement that does not extend into the rocker (Fig. 4.3b).
- A rocker reinforcement that is parallel to the upper plate of the rocker and covers the hole formed by the connection of the rocker and pillar (called *pillar bridge* in this study) (Fig. 4.4).
- Transverse bulkheads in the rocker (Fig.4.5).

From the above we observe that there are two types of pillar reinforcements. In the first type, the reinforcement extends into the rocker, and forms a box inside the rocker (Fig. 4.3a). In the second type, the pillar reinforcement does not extend into the rocker. All hardware joints we examined had this type of reinforcement also had a pillar bridge (Figs. 4.3b and 4.4). Since both types of pillar reinforcements are widely used in the automotive industry, our model can account for both types.

Different types of reinforcements are not always compatible. For example, an extended pillar reinforcement is incompatible with a pillar bridge. On the other hand, if the non-extended pillar reinforcement is used, the pillar bridge is generally used. Bulkheads can be used for any joint.

4.2.2 Types of Rocker Cross Section

The generic B-pillar model can account for two different types of rocker cross section (Fig.4.6). In the first type (*generic type* of rocker cross section), the front part is connected to the center plate, and the center plate is connected to the back part of the rocker. In the second type (*non generic type* of rocker cross section), the front part of the rocker is connected with the back part of the rocker, and the center plate is connected to the front part of the rocker at the bottom. The B-pillar to rocker joint model in this study can account for both types of rocker cross section. Joints with the first type of rocker cross section are easier to manufacture because the front part of the joint is manufactured

from the same piece of sheet metal as the side shell of the car body. The joint with the first type of section also has lower mass than the joint with the second type of cross section because the front plate of the rocker is usually thicker than the center plate. We compared the FEA results for three pairs of joints. The joints in each pair were identical except that one had a generic type of cross section, whereas the other had a non-generic one (Table 4.1). It is found that the stiffnesses of the two joints that have the same dimensions but different types of rocker cross section are practically the same. The joint with the generic type of rocker cross section has slightly lower mass than the joint with the non-generic type of section, but the difference is small (less than 5%).

4.2.3 Type of Construction used in Developing Translator A

The reinforcement has big effects on the stiffness and mass of the B-pillar to rocker joint. Joints with the extended pillar reinforcement have higher stiffness than joints that have the non-extended pillar reinforcement and pillar bridge (Murphy, 1995). We examined several hardware joints, and listed the dimensions and type of construction of each joint. After consulting with engineers from an automotive company, we chose the following type of construction: There is an extended pillar reinforcement but no rocker reinforcement (pillar bridge) and no bulkhead. The center plate is connected to the front part of rocker and the back part of the rocker (generic type of rocker cross section).

Figures 4.7*-4.15 show the dimensions used to define the geometry of B-pillar to rocker joint. Table 4.2 presents the dimensions of the hardware joints examined in this study.

4.3 Developing a Database for Translator A

The development of translator A involves two steps. First, we create a database that includes the values of the dimensions for many designs and their corresponding FEA results. The designs in the database are used as examples to teach translator A how to predict the performance of a design. Second, we use response surface polynomials and

neural networks to simulate the mapping relations between the dimensions and the performance characteristics of the joint (stiffness and mass).

In this study, we first measured the dimensions of the actual joints that were available (Table 4.2). We found the maximum and minimum values of each design variable. As explained in Chapter 2, some design variables are generally fixed because of manufacturing, styling, and packaging requirements, such as, the width of the flange (*length_of_flange*), the distance between two adjacent spot welds (*spot_weld_spacing*), and the position of spot welds on the flange (*spot_weld_placement*). Some variables depend on other variables. By observing the hardware joints, we found that, the values of the blending radii at the outside of the B-pillar are generally close to values of corresponding blending radii at the inner side of the B-pillar. For example, the value of *fwd_outer_ver_blending_rad* is close to the value of *fwd_inner_ver_blending_rad*, and *aft_outer_hor_blending_rad* is close to the value of the dimension *aft_inner_hor_blending_rad*. For all the joints we examined, the B-pillar branch was always vertical. In other words, the two angles defining the orientation of B-pillar, namely, *pillar_io_angle* and *pillar_angle*, were equal to 90 degrees (Fig. 4.7). According to these observations, we assumed that the dimensions related to the flange and spot welds are fixed, the blending radii at the outside of B-pillar are equal to their corresponding blending radii at the inner side of B-pillar, and the two angles defining the orientation of B-pillar are equal to 90 degrees. Table 4.3 shows the range and states (fixed, dependent or free to change) of each design variable.

We used a random number generator to create random designs. The Pro/Engineer model was then used to check if a design is feasible. Infeasible designs were screened out. For each feasible design, we updated the parametric model and created a NASTRAN bulk data file. Figure 4.16 shows a FEA model. Then, all the feasible designs were analyzed using MSC/NASTRAN to get their stiffnesses and masses. Among all random designs, only about 20-25% designs were feasible. The dimensions of all the feasible designs and their performance characteristics were stored in a database

that was used later to create the polynomial and neural network translators. There are 600 designs in the database for the B-pillar to rocker joint**.**

4.4 Developing Translator A

Two different methods, namely, response surface polynomials and neural networks, were used to develop translator A.

To develop translator A, we first ranked the design variables in terms of their effect on the stiffnesses and mass using a linear polynomial (see Chapter 2 for a detailed explanation). Linear and second degree polynomials were used to develop the polynomial translators for the B-pillar to rocker joint. Stepwise regression was used to find the most appropriate regression model (stepwise regression was explained in Chapter 2). We compared the results of the linear and second degree polynomials, from which we chose the best polynomial translators for the stiffness and mass. Neural networks with different number of input design variables and number of hidden layer neurons were studied and compared, from which we chose the best neural network translators. We describe each step in the following sections.

4.4.1 Ranking Important Design Variables

Many design variables are needed to completely describe the geometry of a joint. However, only a few affect significantly the stiffness and mass of a joint. A good model for predicting the performance of a joint should include only the important design variables. It is also important to know the relative importance of each design variable because this information can help designers find the most effective way to improve joint design. The design variables were ranked in terms of importance using a linear regression model. Tables 4.4-4.7 rank the important design variables for the I/O, F/A, torsion stiffnesses, and mass, respectively.

4.4.2 Developing Polynomial Translators

We use both linear and second degree polynomials to develop the translator A for the B-pillar to rocker joint. Each polynomial is presented in detail in the following.

Linear Polynomial Models

First, we obtained the linear polynomial model using stepwise regression. We developed a linear model that included the important design variables. Table 4.8 presents the results for the linear regression model. The linear polynomials for the I/O, F/A, and torsion stiffnesses had 23, 21, and 23 terms, respectively. The linear polynomial for mass had 31 terms. The standard deviations of the ratio of predictions over FEA results for fitting are 13.5% and 7.2% for I/O stiffness and torsion stiffness, respectively. For F/A stiffness and mass, the standard deviations of the fitting results are 5.1% and 0.6%, respectively.

Second Degree Polynomial Models

The predictions from the linear regression models were not satisfactory for some performance characteristics.

The second degree polynomials were found more accurate than the linear polynomials for the I/O and torsion stiffnesses (Table 4.8). The standard deviations of fitting results for I/O and torsion stiffnesses were reduced to 8.8%, and 6.5%, respectively. The standard deviations of the fitting results for F/A stiffness and mass were 5.2%, and 1.2%, respectively. Figures 4.17-4.20 show the relation between number of design variables in the second degree polynomial and the standard deviations of fitting and testing results for the I/O, F/A, torsion stiffnesses and mass. Section 4.5 discusses the results and compares them with results from neural networks.

4.4.3 Developing Neural Networks

Neural Networks were used as an alternative to response surface polynomials to simulate the mapping relations between the design variables and the stiffness and mass of the joint. To develop a neural network for simulating the mapping between the design variables and the performance characteristics of a joint, first, we needed to decide how many design variables to consider. We used the rank found using a linear polynomial to determine which design variables should be considered in the neural network. According to the rank of importance of design variables, first, we considered the most important design variable in the neural network. We gradually increased the number of design variables considered in developing the neural network by adding one design variable at a time. Each neural network was trained using a set of training designs (300 designs in this study). Cross-validation was used to determine when to stop training. Then, we tested the trained neural network using a set of new designs that the neural network had not seen before. The neural network that had the smallest standard deviation according to the testing results was chosen as the neural network translator.

Since the range of each design variable varies greatly, some design variables, such as thickness, can only change less than 1 *mm*, while some other variables, such as the overall height of the rocker can change in a big range. The magnitude of some design variables varies from 1 to 10^2, while the magnitude of the output varies from 10^7 to 10^8 for stiffness. The magnitude of mass is 10. It is useful to normalize both the input and the output variables so that they can all vary in a smaller range. This makes it more efficient to develop a neural network and improves the robustness of the network.

Two methods were tested in this study. The first method normalizes the design variables and the performance characteristics by dividing them by their mean values. In the second method, the parameters were normalized so that they could vary in the range [-1,1]. It was observed from Fig. 4.21 that results from the first normalization method converges much slower than those using the second normalization method. The neural network reached the minimum at about 1300 *epoches* (an *epoch* is one iteration during the training of a neural network) when using the second normalization method (solid line

in Fig. 4.21). It took more than 5000 epoches for this neural network to reach the minimum if the first normalization method was used.

If we train most neural networks over a very long period, we overfit the examples. Although the resulting networks may fit the examples well, their generalization performance may be poor. That is, the network does not predict accurately the stiffness of designs that it has not seen. Figure 4.22 shows the relation between the standard deviation and training time for training and testing results. It is found that the error of the testing result (dashed line in Fig. 4.22) first decreases as training time increases. The error corresponding to testing results reaches the minimum at about 1300 epoches, and then gradually increases when the training time increases even though the error of the training results consistently decreases. To avoid overtraining the network, we used *cross-validation*. Specifically, we split the 600 designs into three groups: 300 designs were used to do training; 100 designs were used to monitor the training process and to determine when to stop training; and the remaining 200 designs were used to test the generalization performance of the trained neural network. Cross validation can prevent the neural networks from being over trained, and thus improves the generalization performance.

The effects of the number of hidden layer neurons on the predictions of neural networks were studied. Figure 4.23 shows the effect of the number of neurons in the hidden layer on standard deviations of the testing results for I/O stiffness when 6, 12, 13, and 24 design variables were considered. Cross-validation was used in obtaining those results. It was observed that the standard deviations of the testing results were insensitive to the number of hidden layer neurons. The difference between the standard deviations corresponding to different number of neurons is probably due to noise. The lack of sensitivity of the accuracy of the neural networks to the number of neurons in the hidden layer is probably due to the use of cross-validation in training. Cross-validation prevented neural networks with large numbers of hidden layer neurons from overfitting the examples.

We only used *determined* neural networks to develop the neural network translator A. A determined neural network involves less unknown parameters than designs used for training. For relatively simple mapping relations, such as the F/A, the torsion stiffness, and mass, we changed the number of design variables from one to the maximum value for which the neural network is determined. The number of neurons in the hidden layer was chosen so that the total number of unknown neural network parameters was approximately equal to 80-90% of the total number of designs in the training set. For complex mapping relations, such as the I/O stiffness, both the number of design variables and the number of neurons were changed from one to the maximum allowable value that corresponds to a determined neural network to find the best combination. The neural network with the smallest testing error was selected as the translator. Figures 4.17 to 4.20 show the relation between standard deviations of training and testing results and the number of design variables in the neural network for the I/O, F/A, torsion stiffnesses and mass of the B-pillar to rocker joint.

4.5 Results and Discussion

Figures 4.17-4.20 compare the results obtained using different translators. The best results from the alternate translators are listed in Table 4.8. Figure 4.24 compare the predictions of translators A and FEA results. The following discuss the results corresponding to the stiffnesses in each direction and the mass.

4.5.1 Comparison of Predictions from Translators A and FEA Results

We compared the relation between the stiffness and four most important design variables (*outborad_cell_with, door_edge_width, thickness of frontrock,* and *A7*) as predicted from translators A and FEA. Figure 4.24 shows that the predictions from both RSP and NN translators have the same trends as the FEA results. Both RSP and NN results correlate well with FEA results.

4.5.2 I/O Stiffness

Table 4.8 and Figure 4.17 compare the results of polynomials and neural networks for different numbers of design variables.

It is observed that the standard deviation of fitting and testing results of the polynomial decreases with the number of design variables when this number is less than 13. For more than 13 variables, the standard deviations of the fitting and testing results are practically constant as the number of design variables increases. Since it is difficult to choose the appropriate second degree polynomial model on the basis of the standard deviation only, we use two other criteria, namely, C_p statistic and *AIC* to help determine which model has the best generalization performance (see Chapter 2.4.3 for the definitions of C_p statistic and *AIC* criterion). Based on the results of the C_p statistic and the *AIC* criterion (Section 4.5.6), we decide to use the second degree polynomial model with 24 design variables as the polynomial translator for the I/O stiffness.

It is observed that the neural network is more accurate than the polynomial according to the fitting and testing results. However, the difference between the neural network and the polynomial is very small in testing. Moreover, the accuracy of the neural network deteriorates more than the accuracy of the polynomial when we compare the training and testing results. This indicates that the polynomial is more robust than the neural network, probably because the latter has more unknown parameters than the former.

The neural network with 24 design variables has the smallest standard deviation for the testing results, and is chosen as the neural network translator for the I/O stiffness.

4.5.3 F/A Stiffness

Table 4.8 and Figure 4.18 show the standard deviations of both fitting and testing results for the F/A stiffness. The error in the F/A stiffness is almost 50% smaller than that in the I/O stiffness. It is observed that testing results are at least as accurate as fitting results for the polynomial. This indicates that sufficient designs have been used to fit the polynomial, and the second degree polynomial model does not have any unimportant (redundant) design variables. When we consider more than 5 design variables in the

regression model, the standard deviations of the testing results fall below those of the fitting results. We think that this is due to the randomness of the designs that we used for fitting and testing. Even if we only consider three variables in the regression model, the standard deviation of the testing results fall below 10%. Comparing the results from the linear polynomial model and from second degree polynomials (Table 4.8), it is found that the linear regression model is slightly more accurate. The reason is that the linear polynomial has more design variables than the second degree polynomial. The *AIC* criterion leads to the same conclusion (Figure 4.29). Therefore, the linear polynomial model with 21 design variables is chosen as the polynomial translator.

The neural networks are almost equally accurate as the second degree polynomials. It is found that the errors of fitting and testing results consistently decreases as the number of design variables increases. We did not consider the second degree polynomial and the neural network with more than 12 designs because the polynomial and neural network predictions are accurate enough compared with the prediction for the I/O stiffness. The neural network with 12 design variables gives the best prediction for the testing designs, and is chosen as the NN translator for F/A stiffness.

4.5.4 Torsion Stiffness

Table 4.8 and Figure 4.19 show the standard deviations for the fitting/training and testing results for the torsion stiffness. The error is smaller than that in I/O stiffness but larger than that in F/A stiffness. It is also observed that the polynomial and the neural network are almost equally accurate in terms of the testing results.

Figures 4.19 and 4.18 look similar. The only difference is that when we include more design variables in the second degree polynomial, the standard deviations of the testing results are larger than those of the fitting results for the torsion stiffness, while for F/A stiffness, the testing curve is below the fitting curve.

Since both the second degree polynomial and the neural network with 12 design variables gave the best predictions, they were chosen as the polynomial and the neural network translators for torsion stiffness.

4.5.5 Mass

The relation between the design variables and the mass is the simplest. From Figure 4.20 and Table 4.8, it is observed that increasing the number of design variables in the polynomial consistently decreases the standard deviations of both fitting and testing results. Since in the future, mass will be used as the objective function when we optimize the joint, we want to include as many design variables as possible. Table 4.8 shows that the linear polynomial is more accurate than the neural network and the second degree polynomial. The C_p statistic and the *AIC* criterion also showed that the linear model should be the best model (see section 4.5.6 for details). Therefore, we selected the linear polynomial with 31 design variables as the translator A for the mass. The neural network with 12 design variables gives the best prediction compared with the other neural networks. Therefore, it is chosen as the NN translator for mass.

4.5.6 Validation

The polynomial regression models obtained using stepwise regression were checked using the C_P criterion and the *AIC* criterion. Figures 4.25-4.28 show the results of validating the regression model selected in Section 4.5.2-4.5.5 using the C_P criterion. Figures 4.29-4.32 show the results when we use the *AIC* criterion to validate previous regression models.

C_P Criterion

For the I/O stiffness, the C_P criterion was used to check the linear polynomial model and the second degree polynomial model using 24 design variables. The results are shown in Figure 4.25. The linear and the second degree polynomial models selected in Section 4.5.2 have 23 and 24 design variables, respectively. It is observed that the value of C_P for the selected polynomials is very close to the value of p, which is the number of parameters in the polynomial. Therefore, according to C_P criterion, both regression models are suitable for predicting the I/O stiffness.

Figure 4.26 shows the results for the F/A stiffness of the B-pillar to rocker joint for the linear polynomial model and the second degree polynomial model with 12 design variables. The two models have 22 and 31 terms, respectively. It is observed that both models are appropriate according to the C_P criterion.

Figure 4.27 shows the results for the torsion stiffness. It is observed that both the linear regression model and the second degree polynomial model with 12 design variables are appropriate according to the C_P criterion.

Figure 4.28 shows the values of C_P versus p for the mass of the B-pillar to rocker joint. The C_P values for both the linear regression model and the nonlinear regression model with 12 design variables are very close to the number of parameters in the polynomials. However, the value of C_P is larger than the number of terms for the second degree polynomial. This indicates that the second degree polynomial does not have enough terms. However, we did not try to improve this polynomial because its error is already much smaller than the errors corresponding to the stiffnesses in the other directions.

AIC Criterion

Figure 4.29 shows the relation between the *AIC* values and number of terms in the polynomial, p, for both the linear polynomial model and the second degree polynomial model for the I/O stiffness. It is observed that the nonlinear regression model with 24 design variables (the polynomial has 54 terms) has the lowest *AIC* value. According to *AIC* criterion, the second degree polynomial with 24 design variables is the best polynomial model. This observation is consistent with the results of Table 4.8.

The relations between the *AIC* values and p of two models for F/A stiffness are compared in Fig. 4.30. The linear polynomial and the second degree polynomial have 21 and 12 design variables, respectively. It is observed that the linear regression model has

lower *AIC* value compared with the nonlinear regression model. This indicates that the linear regression model has better generalization performance than the second degree polynomial. This is also consistent with the results in Table 4.8.

Figure 4.31 shows that the linear polynomial model and the second degree polynomial with 12 design variables have almost equal *AIC* value. This shows that the two models have almost same accurate performance. Again, this observation agrees with the results in Table 4.8.

Figure 4.32 shows that, for the mass of the B-pillar to rocker joint, the linear polynomial has much small *AIC* value compared with the second degree polynomial that has 12 design variables. According to *AIC* criterion, the linear regression model is more accurate than the second degree polynomial. Table 4.8 leads to the same conclusion: the standard deviation of the linear polynomial is 0.6%, while the standard deviation of the second degree polynomial is 1.4%.

4.5.7 Conclusions

In this study, the procedure of constructing translators A for the B-pillar to rocker joint was explained. Several polynomial models were compared. The C_P criterion and the *AIC* criterion were used to validate obtained results. Neural networks were also used as an alternative to simulate the mapping relation between the design variables and the stiffness and mass of the joint. The results from different models were compared. Both polynomials and neural networks simulated FEA results quite accurately. It was observed that the neural networks and the polynomials were almost equally accurate. However, the neural networks deteriorated more than the polynomials when they predicted the stiffness of designs that they had not seen in training. Using the C_P and *AIC* criteria to compare models, we obtained results that were consistent with those from using the standard deviation of the ratio of predictions over FEA results.

Table 4.1: Comparison of Stiffness and Mass
for Different Types of Rocker Cross Section

Cars	Cross Section Type	I/O (Nmm)	F/A (Nmm)	Torsion (Nmm)	Mass (Nmm)
Car A	Generic	4.5456E7	5.3119E8	8.7844E7	5.9589
	Non-Generic	4.5709E7	5.3122E8	8.7850E7	6.1517
Car B	Generic	9.1601E6	2.1839E8	5.3031E7	5.0346
	Non-Generic	9.1488E6	2.1838E8	5.3040E7	5.1210
baseline	Generic	3.3169E7	3.4940E8	8.1886E7	5.5694
	Non-Generic	3.3168E7	3.4932E8	8.1887E7	5.6927

Cars A and B are the same as the cars 1 and 2 in Table 4.2.
Baseline is the joint whose dimensions are equal to the mean values of dimensions measured in Table 4.2.

Table 4.2: Measured Dimensions for B-pillar to Rocker Joint

Name of Design Variables	Dimensions (mm, degree)					
	Car 1	Car 2	Car 3	Car 4	Min	Max
Length_of_flange	17	18	20	20	17	20
Spot_weld_spacing	45	40	50	55	40	55
Spot_weld_placement	8	7	8	8	7	8
Pillar_base	185	157	215	180	157	215
Pillar_angle	90	90	90	90	90	90
Pillar_io_angle	90	90	90	90	90	90
Pillar_height	220	210	250	210	210	250
Pillar_location	143	186	160	163	143	186
Outer_pillar_width	70	83	67	50	50	83
Inner_pillar_width	10	6	6	25	6	25
Pillar_outer_length	90	77	107	110	77	110
Pillar_inner_length	122	130	160	155	122	160
Fwd_inner_ver_blending_rad	130	124	155	120	120	155
Fwd_inner_hor_blending_rad	125	118	80	90	80	125
Fwd_outer_ver_blending_rad	130	124	155	120	120	155
Fwd_inner_hor_blending_rad	125	118	80	90	80	125
Aft_inner_ver_blending_rad	105	120	133	95	95	133
Aft_inner_hor_blending_rad	105	135	115	100	100	135
Aft_outer_ver_blending_rad	117	120	140	95	95	140
Aft_outer_hor_blending_rad	115	135	130	100	100	135
Inner_pillar_base_width	15	12		2	2	15
Rocker_length	490	480	483	480	480	490
Pillar_reinforcement_depth	100	80		80	80	100
Pillar_reinf_extended	Yes	Yes	No	Yes		
Pillar_reinf_base_width	60	30		20	20	60

Pillar_reinf_expansion	10	15		2.	2.	15
A1	90	90	85	83.	83.	90
A2	70	65	75	78	65	78
A3	10	11	36	22	10	36
A5	75	80	87	87	75	87
A7	70	86	80	80	70	86
A8	80	85	90	90	80	90
Frontrock_generic_type	No	No	No	Yes		
Rocker_height	117	117	120	110	110	120
Inner_flange_distance	22	18	28	45	18	45
Inner_rocker_height	90	96	110	90	90	110
Rocker_width	147	130	120	115	115	147
Outboar_cell_width	65	74	95	70	65	95
Door_edge_height	6	21	38	32	6	38
Door_edge_width	19	16	7	10	7	19
Low_door_ht_minus_clearane	66	75	65	60	60	75
Fwd_bulk_head_position		10				
Aft_bulk_head_position	37	470				
Outboard_rocker_bulkheads	No	No	No	No		
Inboard_rocker_bulkheads	Yes	Yes	No	No		
Top_pillar_hole	22	35	80	39	22	80
Bottom_pillar_hole	35	28	50	45	28	50
Fwd_pillar_hole	17	18	30	17	17	30
Aft_pillar_hole	15	15	20	18	15	20
Thickness of frontrock	0.71	0.94	1.07	1.27	0.71	1.27
Thickness of pillar_reinf	0.71	0.74	1.52	1.07	0.71	1.52
Thickness of pillarback	0.89	1.27	1.07	1.27	0.89	1.27
Thickness of backrock	1.78	1.65	1.98	1.27	1.27	1.98
Thickness of centerplate	0.71	0.74	1.07	1.27	0.71	1.27
Thickness of pillar_bridge						
Thickness of bulkheads	1.78	1.40				

Table 4.3: Ranges of Design Variables Used in Creating the Database

Seq No.	Name of Design Variables	Bounds (mm, degree)		Comments
		Lower	Upper	
1	Length_of_flange #	19	19	1
2	Spot_weld_spacing #	47.5	47.5	2
3	Spot_weld_placement #	8.5	8.5	3
4	Pillar_base	186	215	
5	Pillar_angle #	90	90	4
6	Pillar_io_angle #	90	90	5
7	Pillar_height	230	250	
8	Pillar_location	164.5	186	
9	Outer_pillar_width	66.5	83	
10	Inner_pillar_width	15.5	25	
11	Pillar_outer_length	93.5	110	
12	Pillar_inner_length	141	160	
13	Fwd_inner_ver_blending_rad	137.5	155	
14	Fwd_inner_hor_blending_rad	102.5	125	
15	Fwd_outer_ver_blending_rad ##	137.5	155	6
16	Fwd_inner_hor_blending_rad ##	102.5	125	7
17	Aft_inner_ver_blending_rad	114	133	
18	Aft_inner_hor_blending_rad	117.5	135	
19	Aft_outer_ver_blending_rad ##	117.5	140	8
20	Aft_outer_hor_blending_rad ##	117.5	135	9
21	Inner_pillar_base_width	8.5	15	
22	Rocker_length #	485	485	10
23	Pillar_reinforcement_depth ##	90	100	11
24	Pillar_reinf_base_width	40	60	
25	Pillar_reinf_expansion	8.5	15	
26	A1_	86.5	90	
27	A2_	71.5	78	
28	A3_	23	36	
29	A5_	81	87	
30	A7_	78	86	
31	A8_	85	90	
32	Rocker_height	115	120	
33	Inner_flange_distance	31.5	45	
34	Inner_rocker_height	100	110	
35	Rocker_width	131	147	
36	Outboar_cell_width	80	95	
37	Door_edge_height	22	38	
38	Door_edge_width	13	19	
39	Low_door_ht_minus_clearane	67.5	75	
40	Fwd_bulk_head_position	470	470	
41	Aft_bulk_head_position	10	10	
42	Top_pillar_hole	51	80	

43	Bottom_pillar_hole	39	50	
44	Fwd_pillar_hole	23.5	30	
45	Aft_pillar_hole	17.5	20	
46	Thickness of frontrock	1	1.27	
47	Thickness of pillar_reinf	1.11	1.52	
48	Thickness of pillarback	1.08	1.27	
49	Thickness of backrock	1.63	1.98	
50	Thickness of centerplate	1	1.27	
51	Thickness of pillar_bridge	1	1	
52	Thickness of bulkheads	1.59	1.78	

fixed

dependent

1 *Length_of_flange* is fixed at 19 mm.
2 *Spot_weld_spacing* is fixed at 47.5 mm.
3 *Spot_weld_placement* is fixed at the center of flange.
4 *Pillar_angle* is fixed at 90 degrees.
5 *Pillar_io_angle* is fixed at 90 degrees.
6 *Fwd_outer_ver_blending_rad* is assumed to be equal to *Fwd_inner_ver_blending_rad*.
7 *Fwd_outer_hor_blending_rad* is assumed to be equal to *Fwd_inner_hor_blending_rad*.
8 *Aft_outer_ver_blending_rad* is assumed to be equal to *Aft_inner_ver_blending_rad.*
9 *Aft_outer_hor_blending_rad* is assumed to be equal to *Aft_inner_hor_blending_rad.*
10 *Rocker_length* is fixed at 485 mm.
11 *Pillar_reinforcement_depth* can be derived from other design variables.

Table 4.4: Ranking of Important Dimensions for I/O Stiffness

Rank	Dimensions
1	Outboard_cell_width
2	Door_edge_width
3	Thickness of frontrock2
4	A7
5	Pillar_base
6	Inner_pillar_base_width
7	Door_edge_height
8	Thickness of backrock2
9	Aft_inner_ver_blending_rad
10	Pillar_location
11	Rocker_width
12	A5
13	Thickness of pillarback2
14	Inner_rocker_height
15	Pillar_inner_length
16	A2
17	Bottom_pillar_hole
18	Fwd_inner_hor_blending_rad
19	Outer_pillar_width
20	Aft_inner_hor_blending_rad
21	Thickness of pillar_reinf
22	Low_door_ht_minus_clearance
23	Pillar_height
24	Fwd_inner_ver_blending_rad

Table 4.5: Ranking of Important Dimensions for F/A Stiffness

Rank	Dimensions
1	Thickness of frontrock2
2	Pillar_inner_length
3	Pillar_base
4	Thickness of pillarback2
5	Bottom_pillar_hole
6	Outer_pillar_width
7	Pillar_outer_length
8	Thickness of backrock2
9	Top_pillar_hole
10	Aft_inner_ver_blending_rad
11	Pillar_height
12	Fwd_pillar_hole
13	Rocker_width
14	Outboard_cell_width
15	Door_edge_height
16	Door_edge_width
17	A5
18	Thickness of pillar_reinf
19	Low_door_ht_minus_clearance
20	A8

Table 4.6: Ranking of Important Dimensions for Torsion Stiffness

Rank	Dimensions
1	Outer_pillar_width
2	Thickness of frontrock2
3	Pillar_height
4	Outboard_cell_width
5	Top_pillar_hole
6	Pillar_base
7	Inner_pillar_base_width
8	Door_edge_width
9	Fwd_pillar_hole
10	Thickness of pillarback2
11	Pillar_inner_length
12	Inner_pillar_width
13	Aft_inner_ver_blending_rad
14	Door_edge_height
15	Aft_pillar_hole
16	A7
17	Fwd_inner_hor_blending_rad
18	Bottom_pillar_hole
19	Pillar_location
20	Pillar_reinf_base_width

Table 4.7: Ranking of Important Dimensions for Mass

Rank	Dimensions
1	Thickness of frontrock2
2	Thickness of pillar_reinf
3	Thickness of backrock2
4	Thickness of centerplate
5	Rocker_width
6	Pillar_height
7	Thickness of pillarback2
8	Inner_flange_distance
9	Rocker_height
10	Outer_pillar_width
11	Pillar_base
12	A2
13	A5
14	Pillar_inner_length
15	Inner_rocker_height
16	A3
17	Pillar_outer_length
18	Pillar_reinf_base_width
19	Top_pillar_hole
20	Outboard_cell_width

Table 4.8: Comparison of Different Models for B-pillar to Rocker Joint

Stiffness/ Mass	Type	Model	No of Variables	Std deviation Fitting/Training	Std deviation Testing
I/O	Polynomial	Linear	23	0.1354	0.1182
		2nd degree	24	0.0880	0.0982
	NN		24	0.0755	0.0872
F/A	Polynomial	Linear	21	0.0508	0.0447
		2nd degree	12	0.0519	0.0518
	NN		12	0.0533	0.0485
Torsion	Polynomial	Linear	23	0.0716	0.0745
		2nd degree	12	0.0647	0.0720
	NN		12	0.0609	0.0716
Mass	Polynomial	Linear	31	0.0062	0.0063
		2nd degree	12	0.0124	0.0138
	NN		12	0.0127	0.0139

Note: Only the best second order polynomial results and the best neural network results are listed.

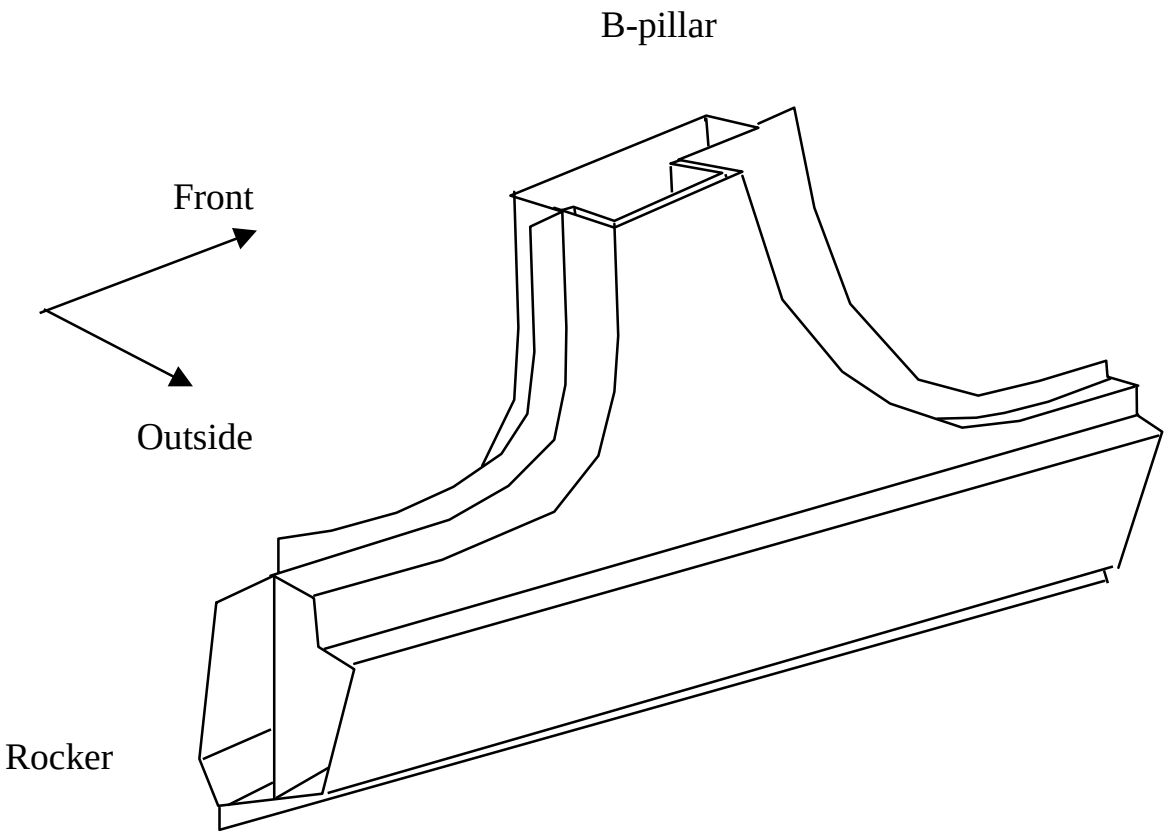


Figure 4.1a: B-pillar to Rocker Joint

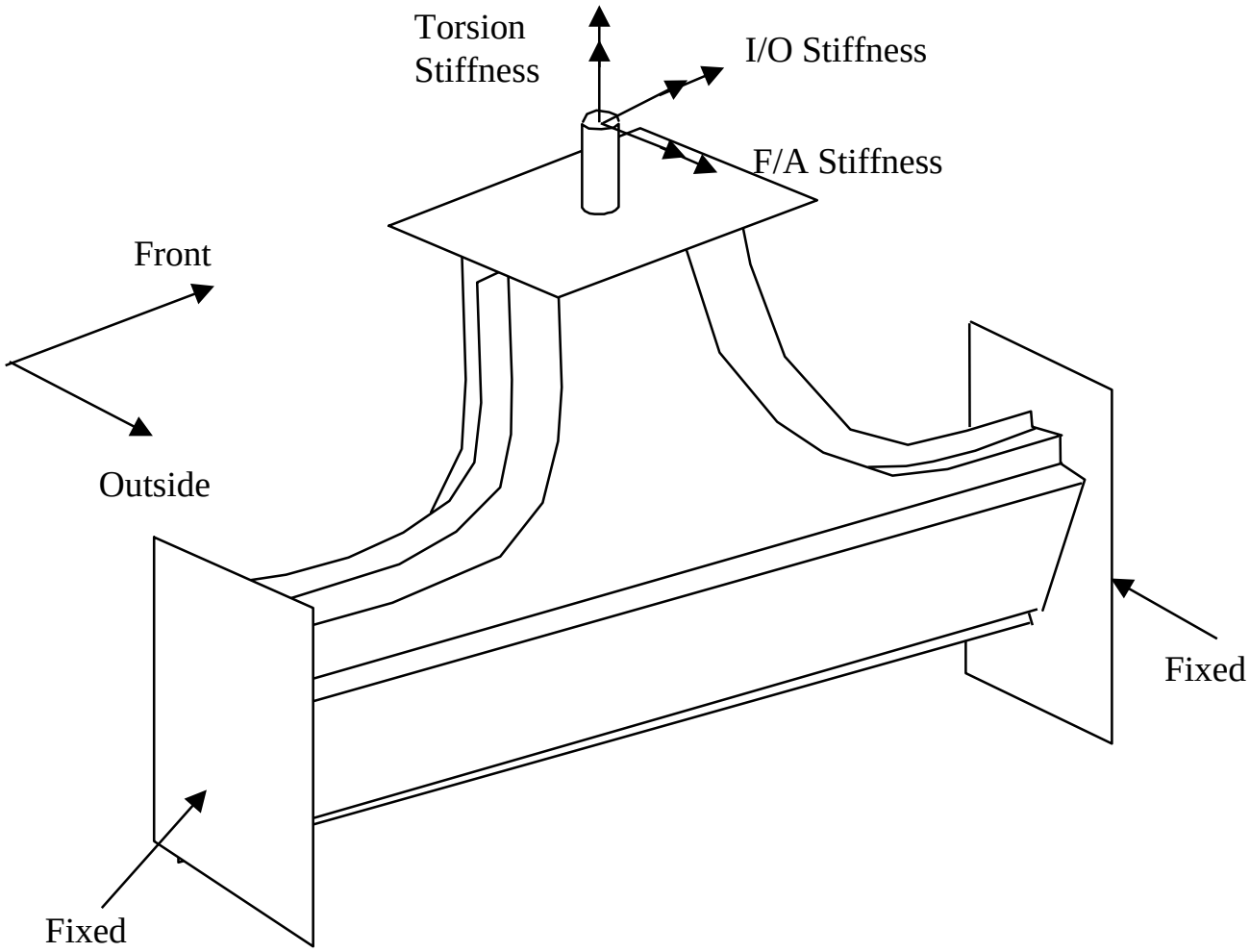


Figure 4.1b: Definition of Stiffness for B-pillar to Rocker Joint

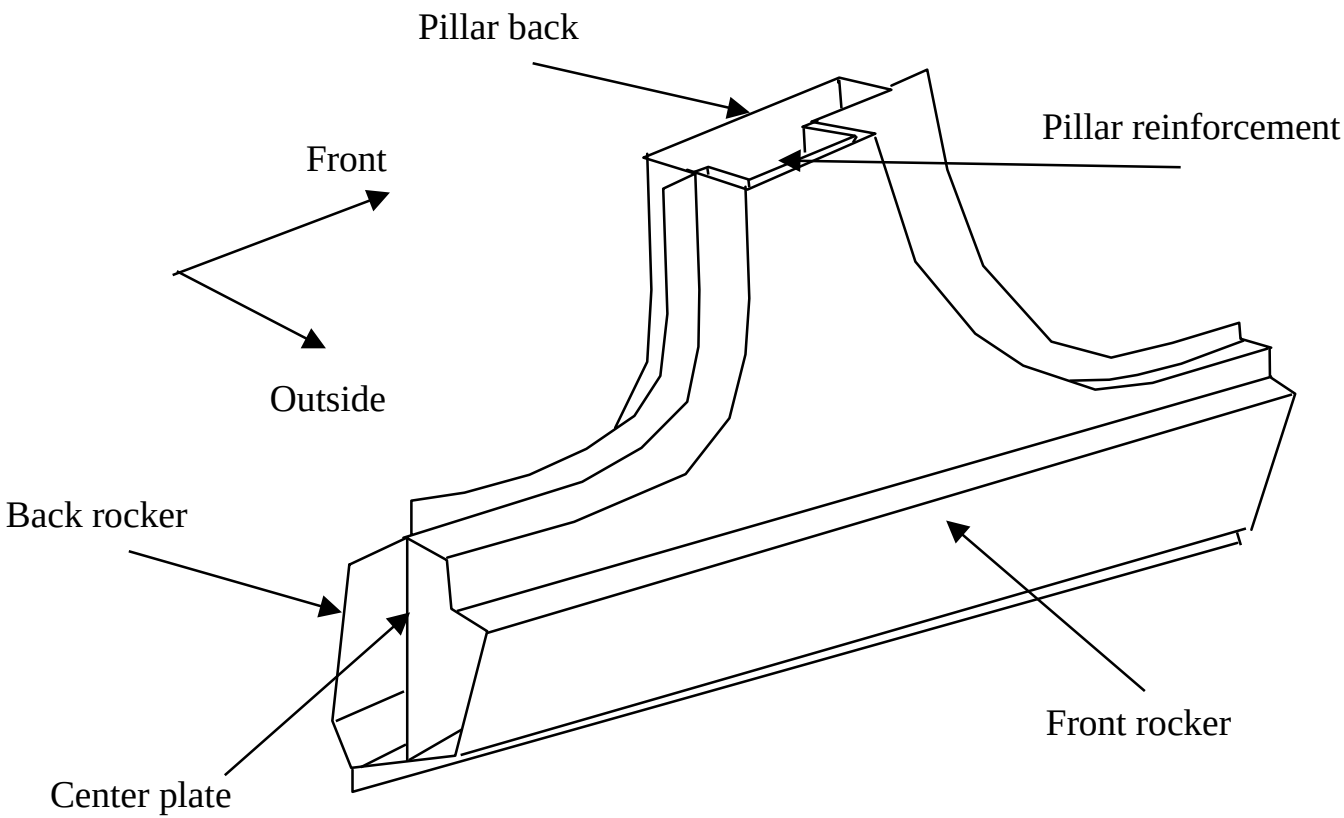


Figure 4.2: Parts of B-pillar to Rocker Joint

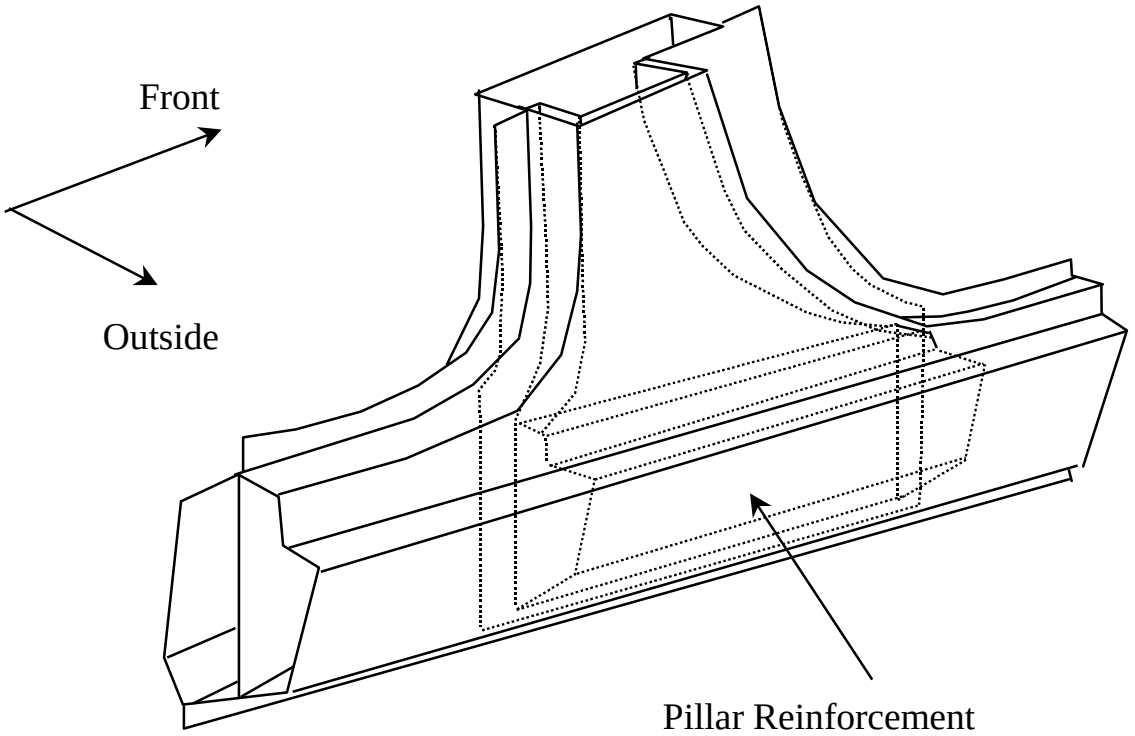


Figure 4.3a: Extended B-pillar Reinforcement:

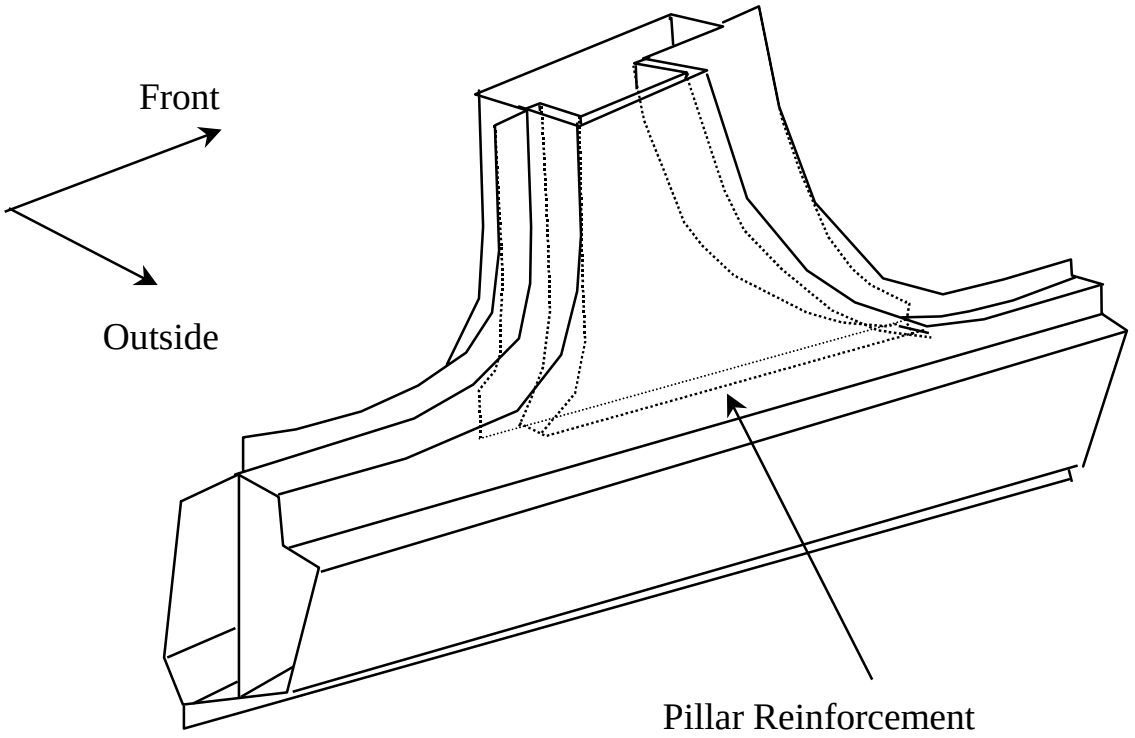


Figure 4.3b: Non-Extended B-pillar Reinforcement:

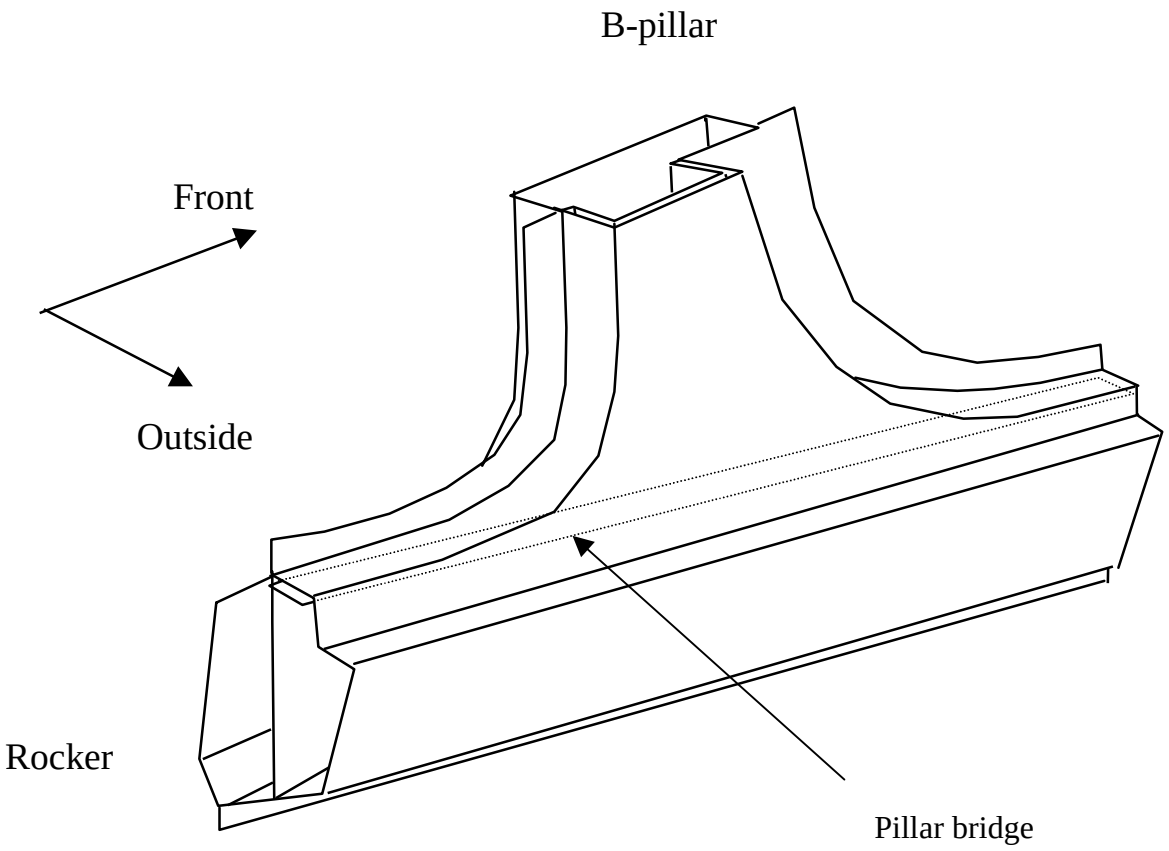


Figure 4.4: Pillar Bridge Reinforcement

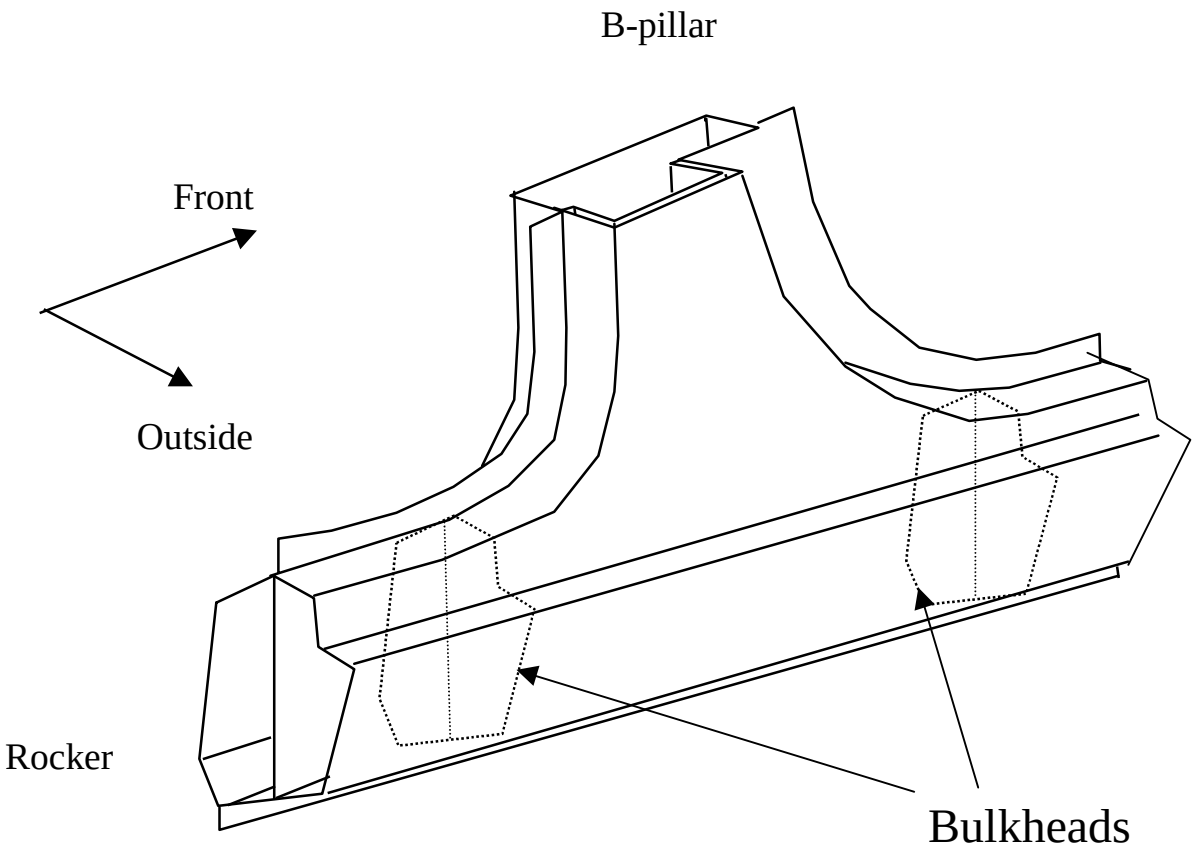
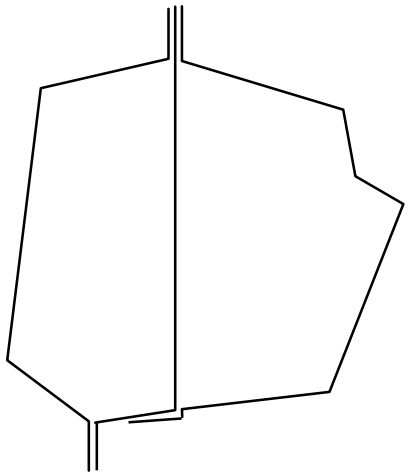
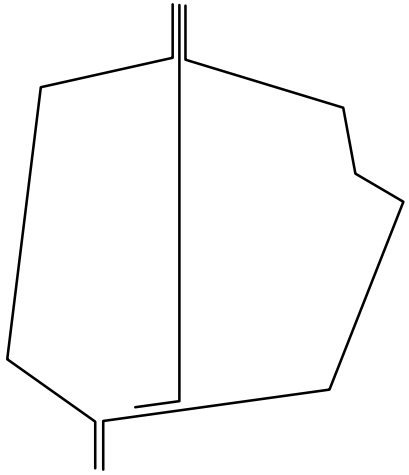


Figure 4.5: Bulkhead Reinforcement



Type 1: Generic Type



Type 2: Non Generic Type

Figure 4.6: Two Different Types of Rocker Cross Section

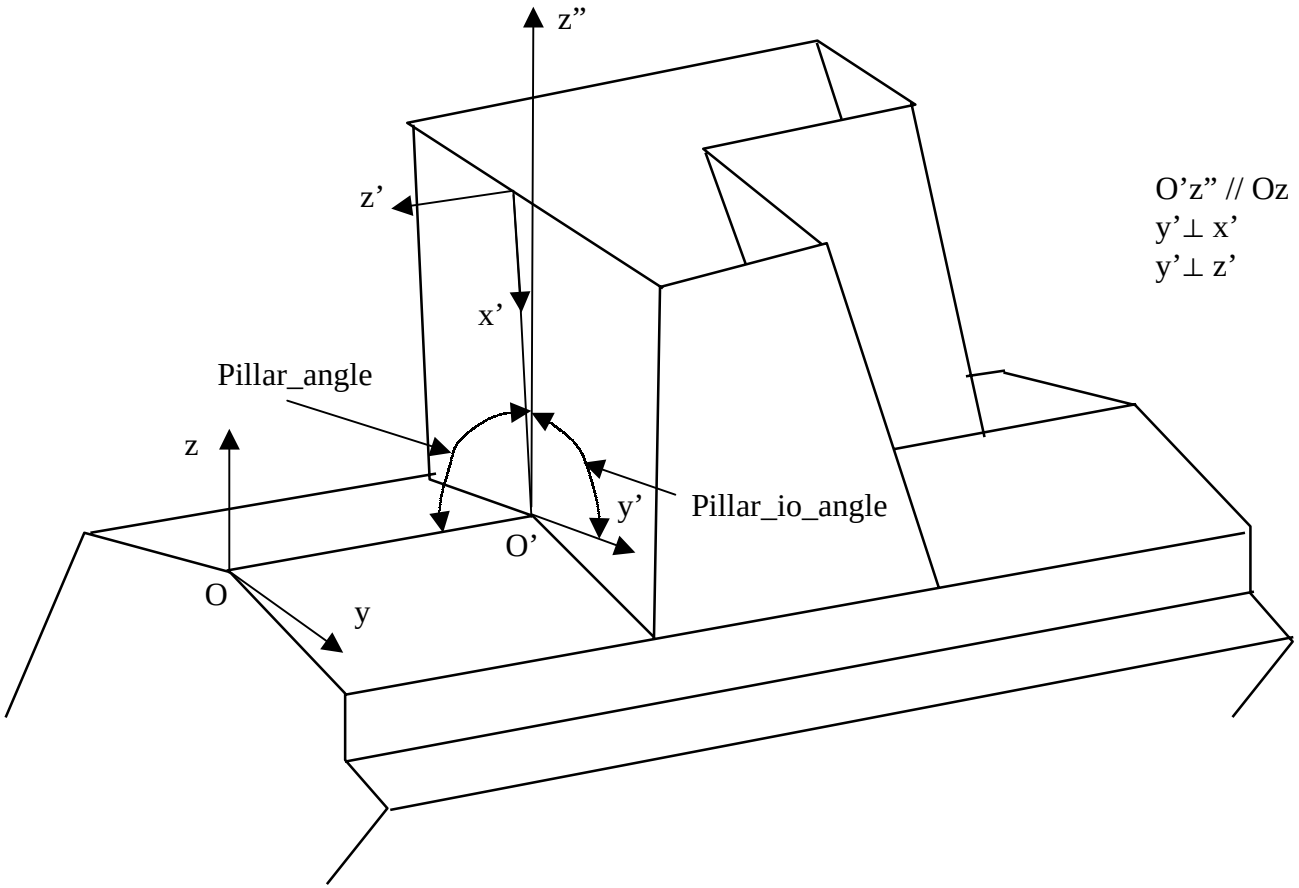


Figure 4.7*: B-pillar Orientation

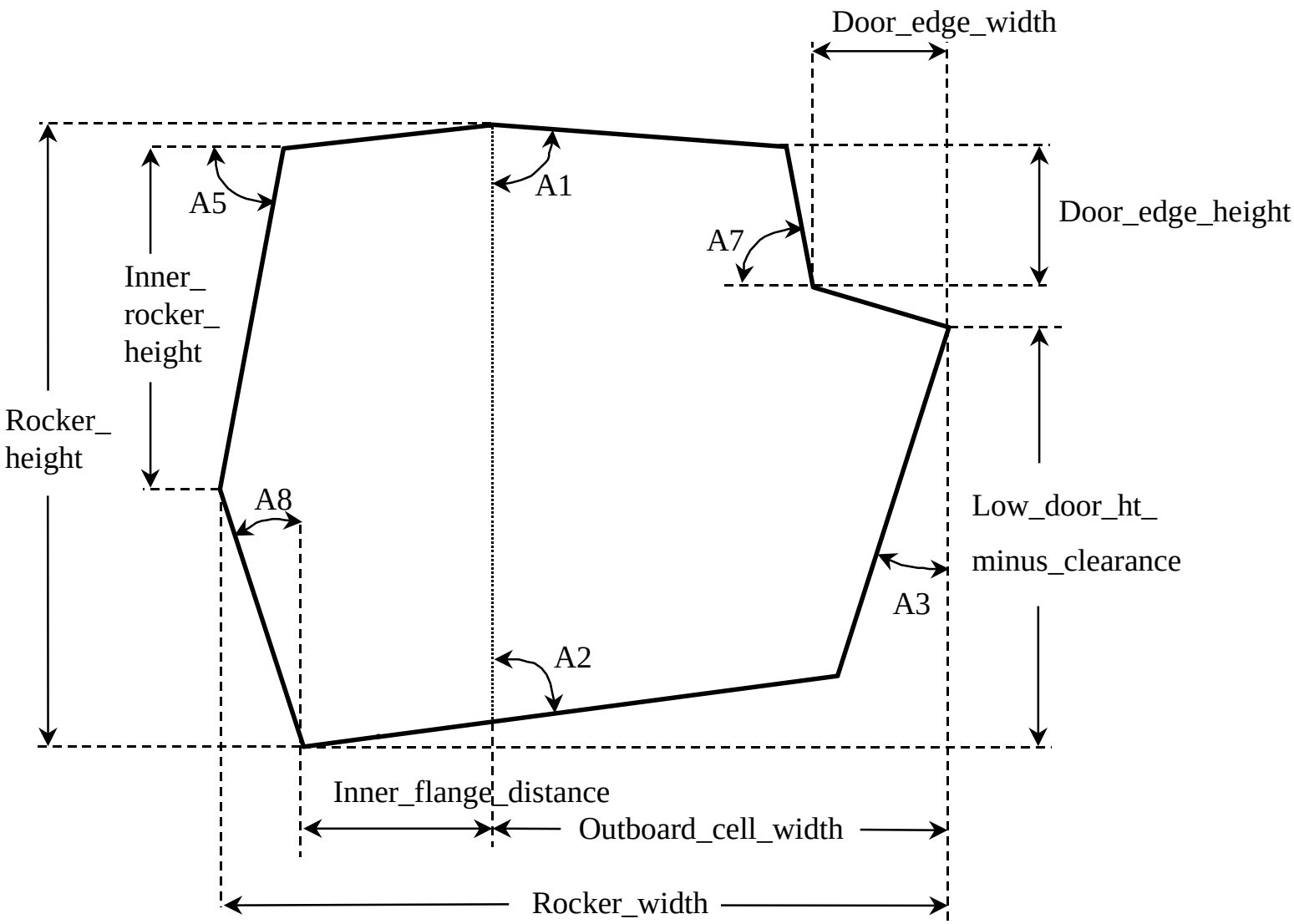


Figure 4.8: Rocker Cross Section

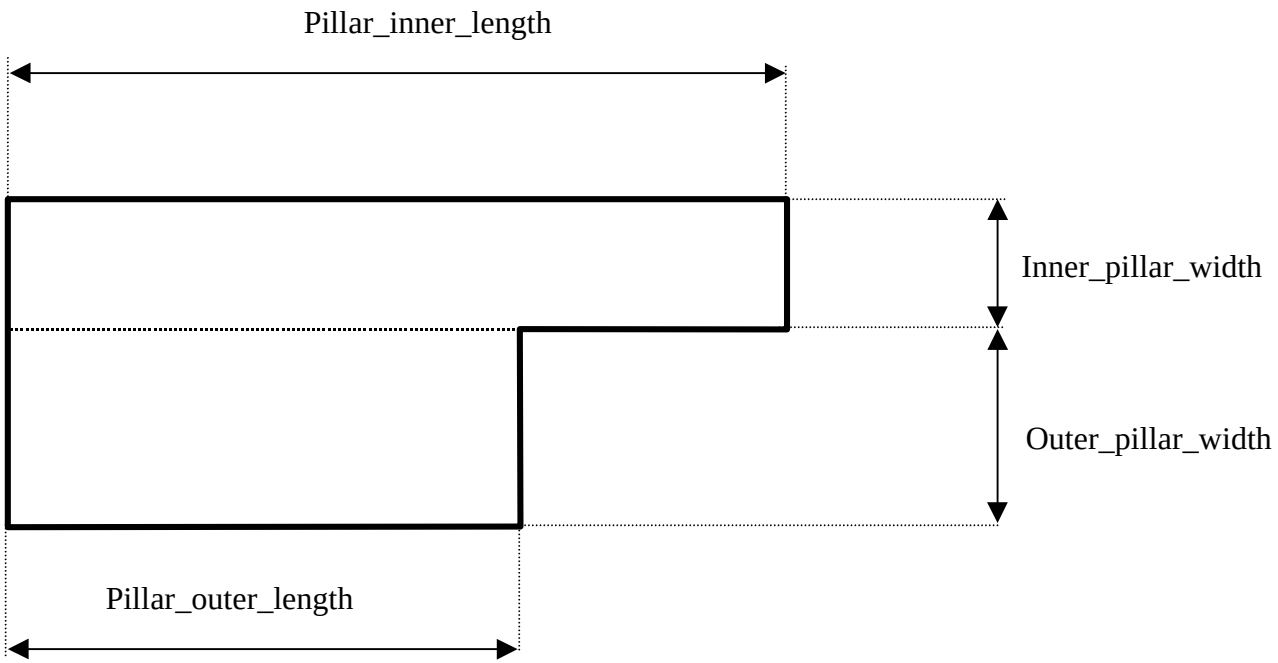


Figure 4.9*: B-pillar Dimensions

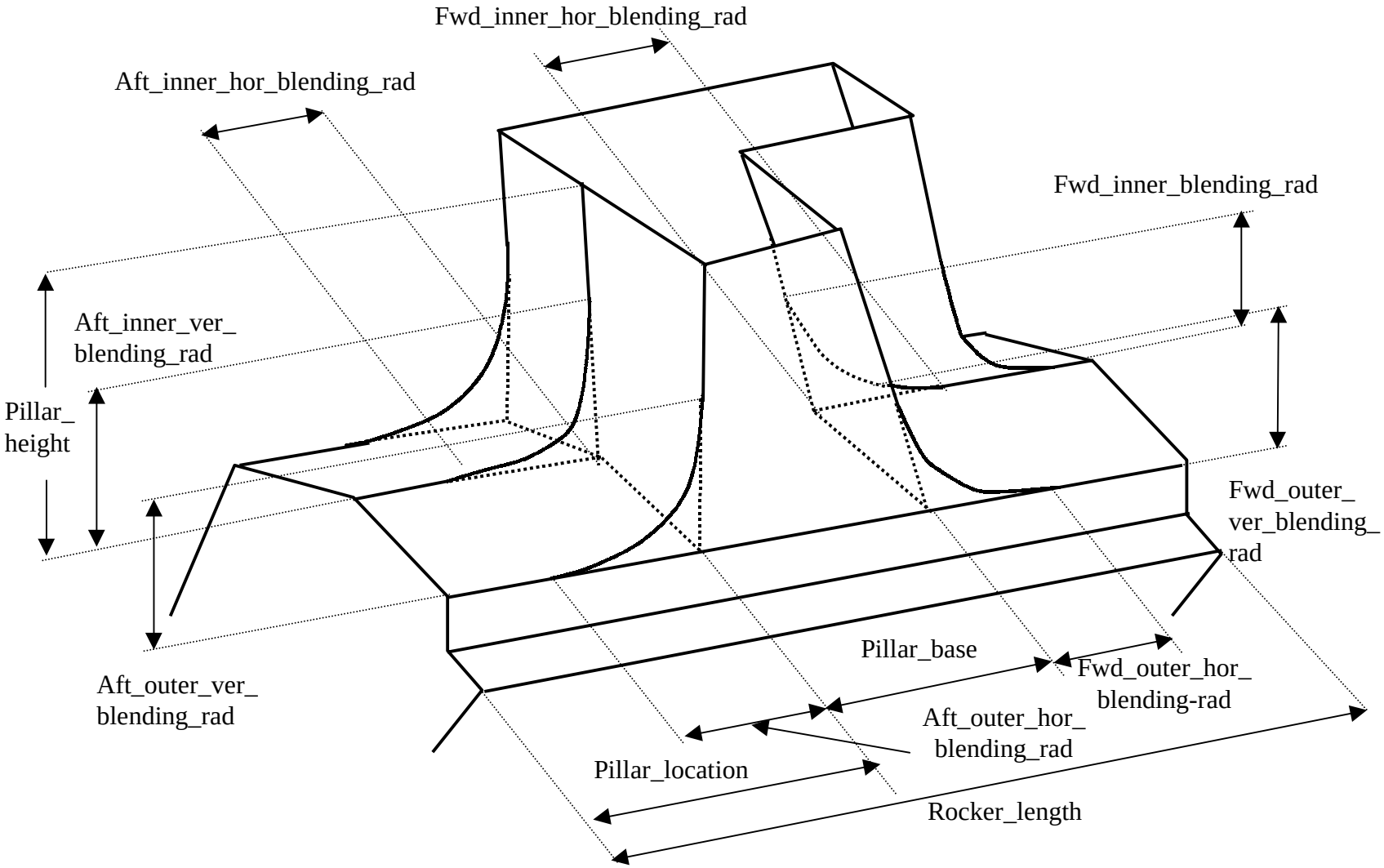


Figure 4.10*: B-pillar to Rocker Blending Radii and Rocker Dimensions

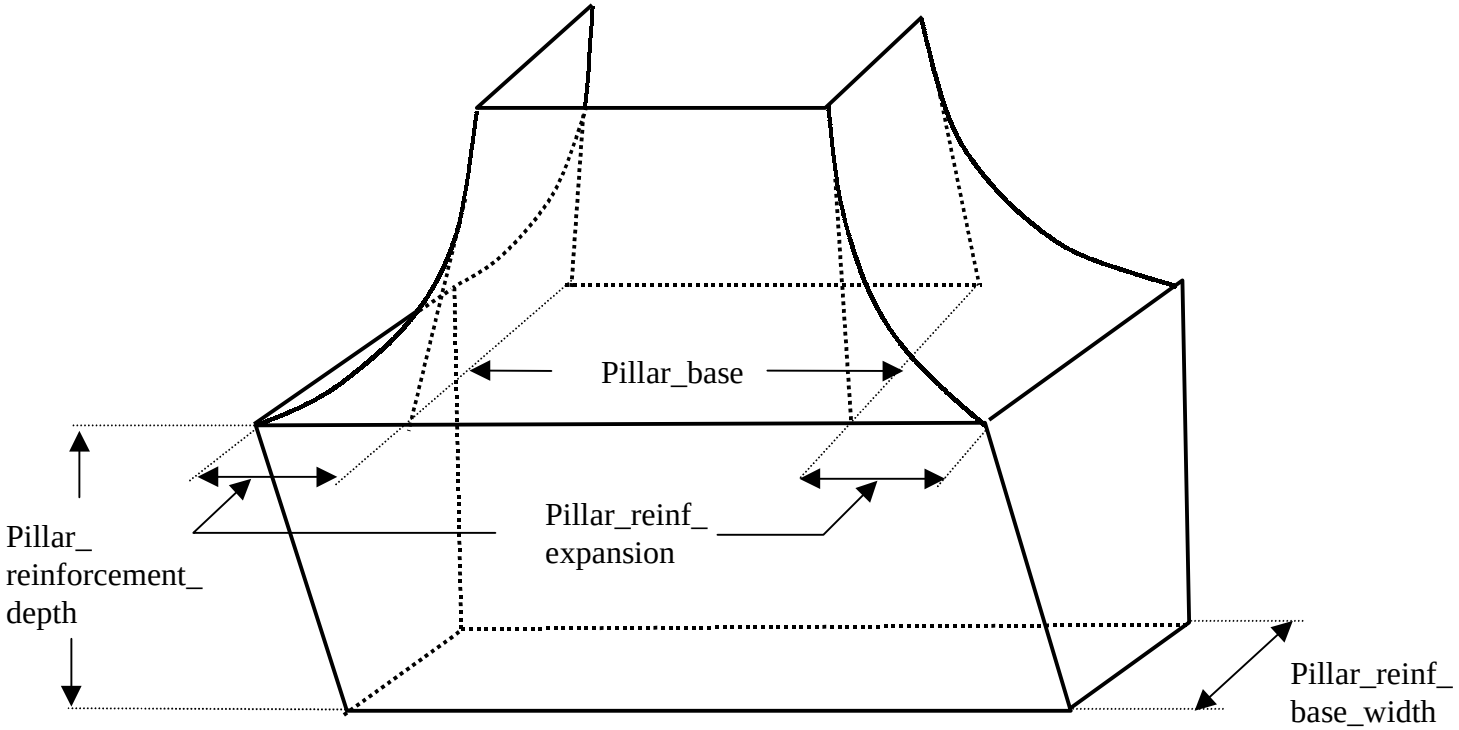


Figure 4.11*: Extended Pillar Reinforcement

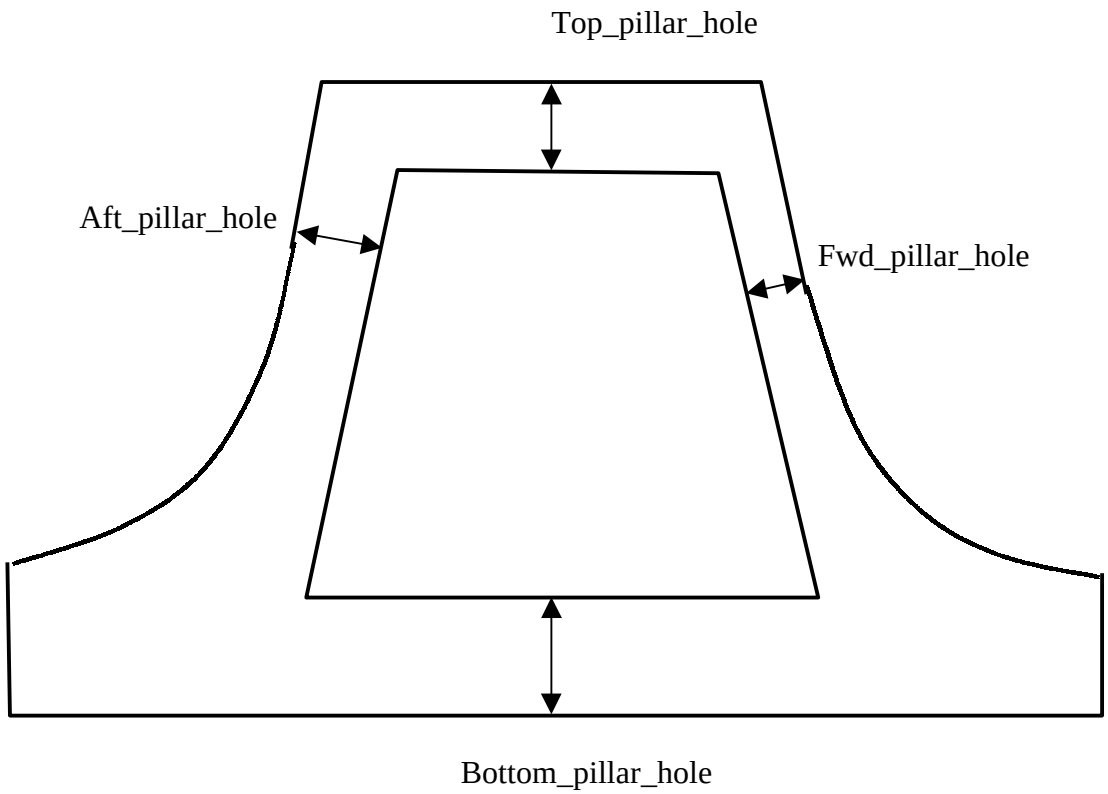


Figure 4.12*: Opening in Back of Pillar

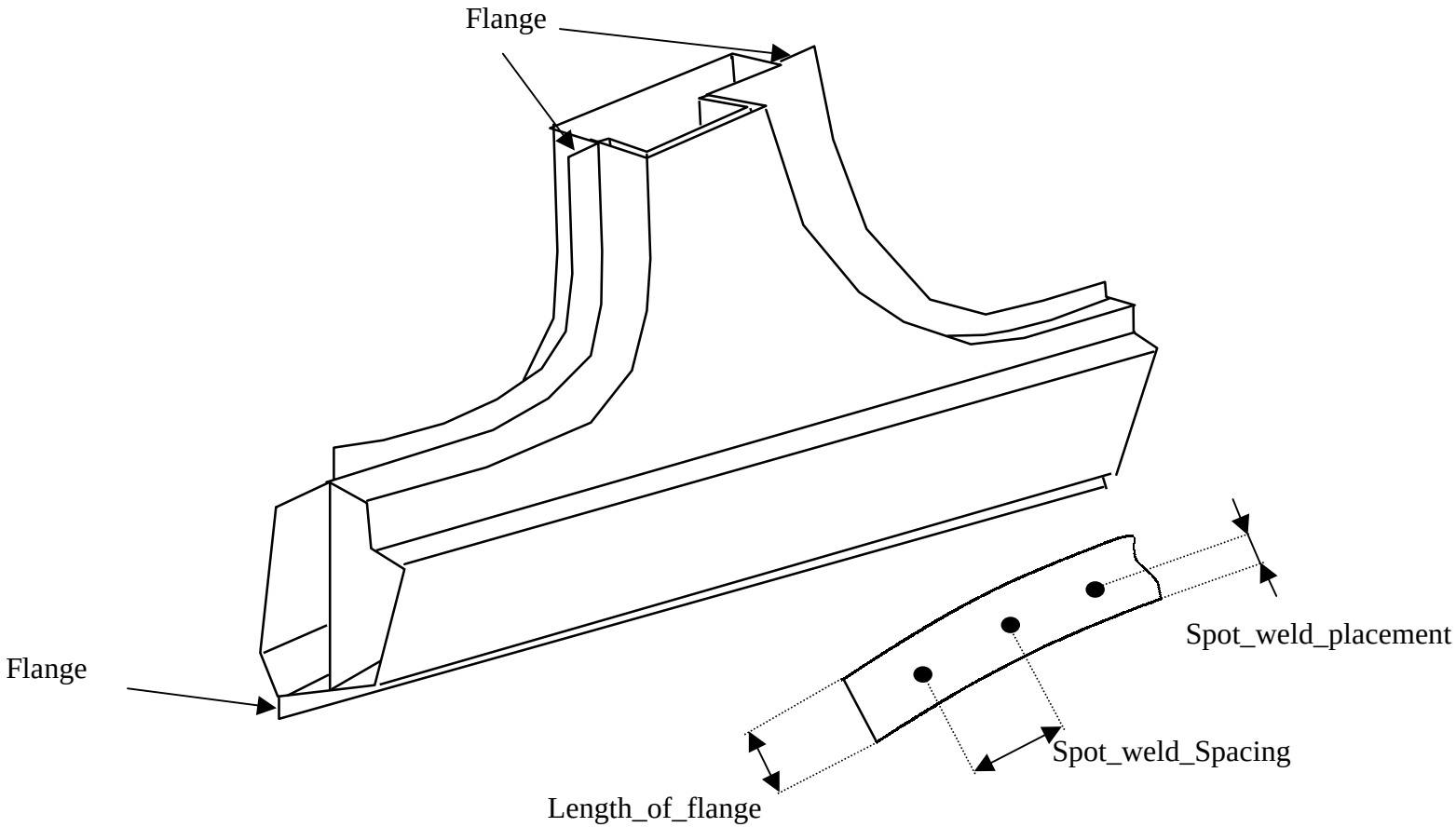


Figure 4.13: Flanges and Spot Welds

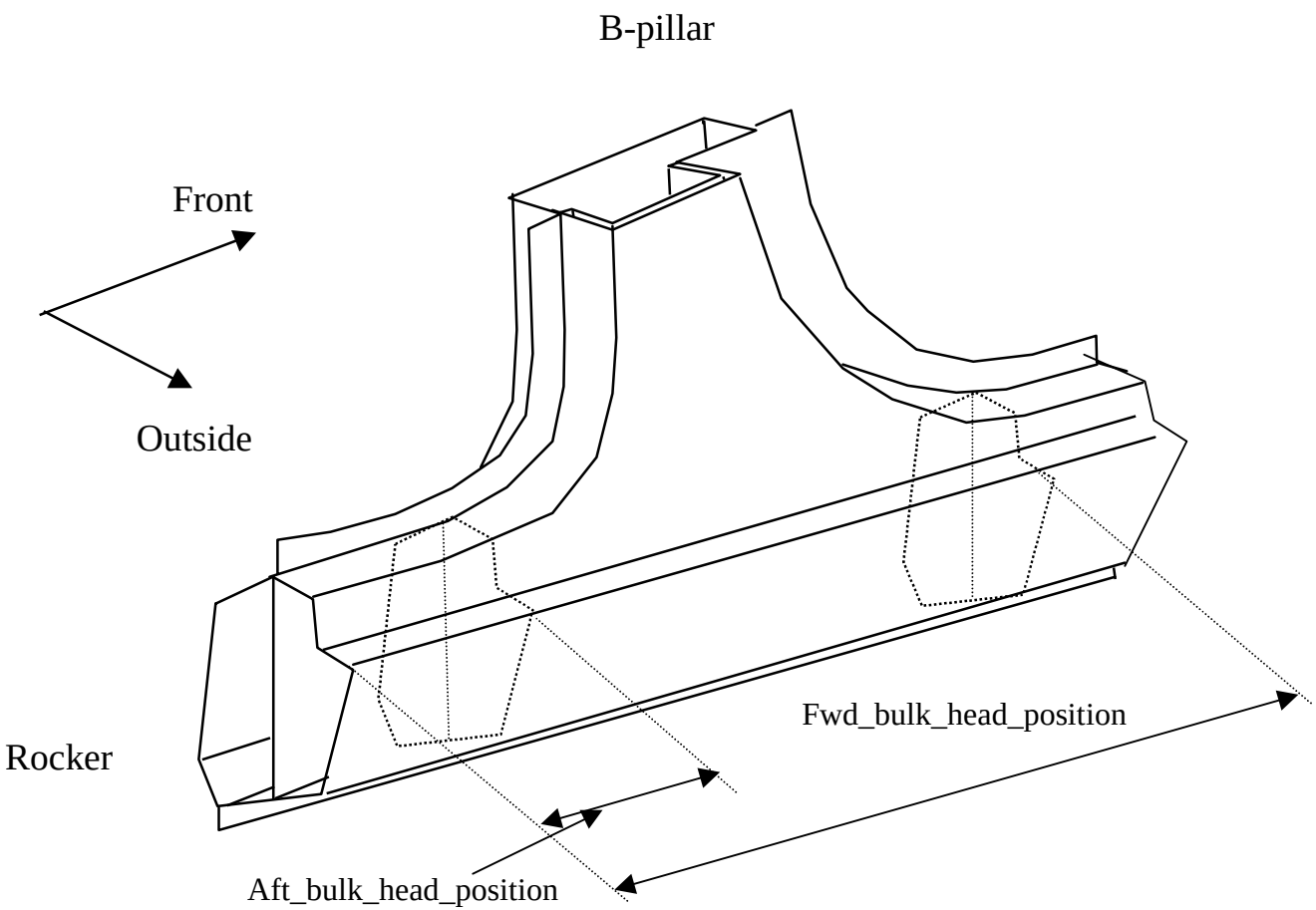


Figure 4.14: Dimensions for Bulkhead

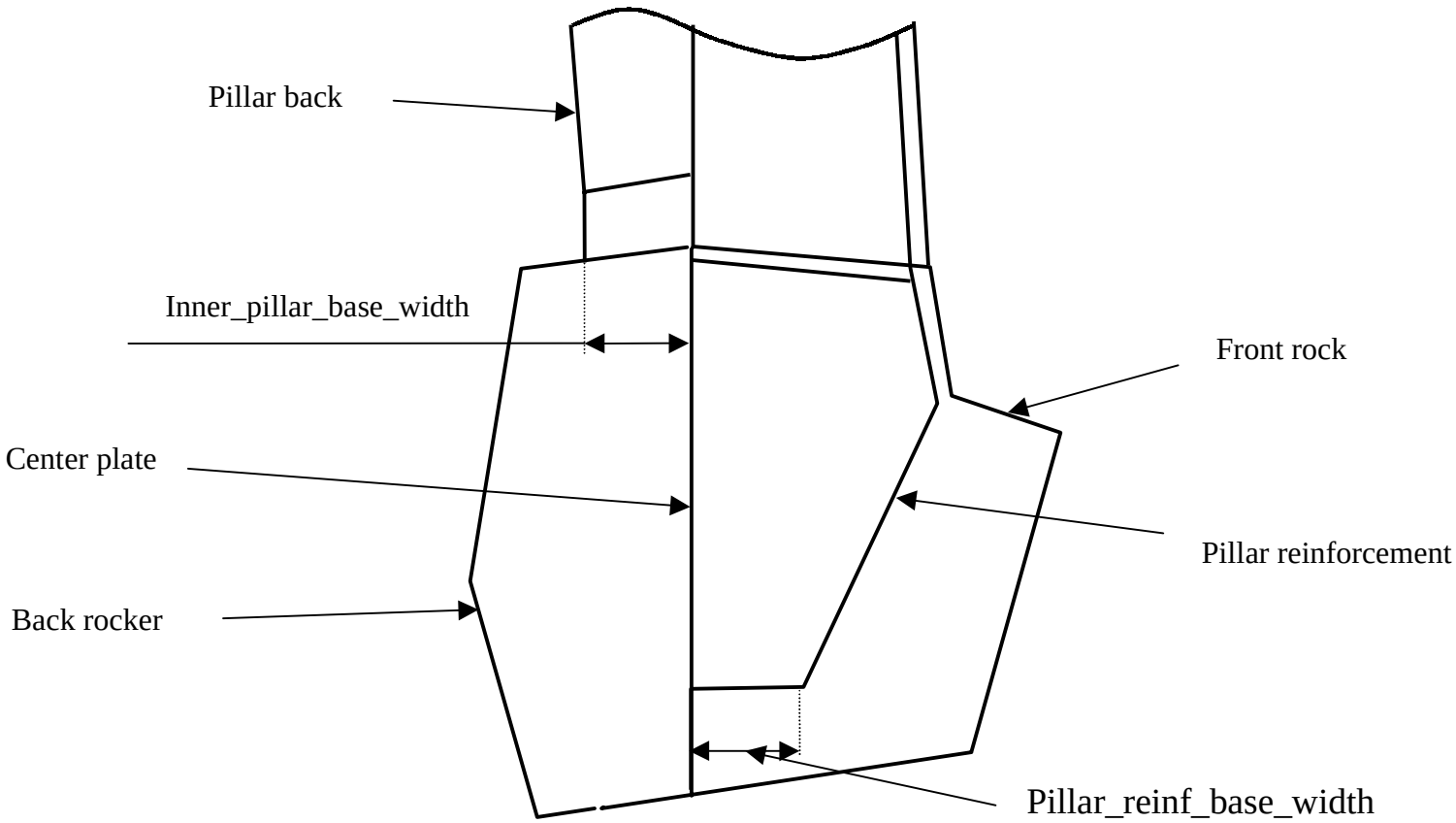


Figure 4.15: Dimensions for Back Rocker and Pillar Reinforcement

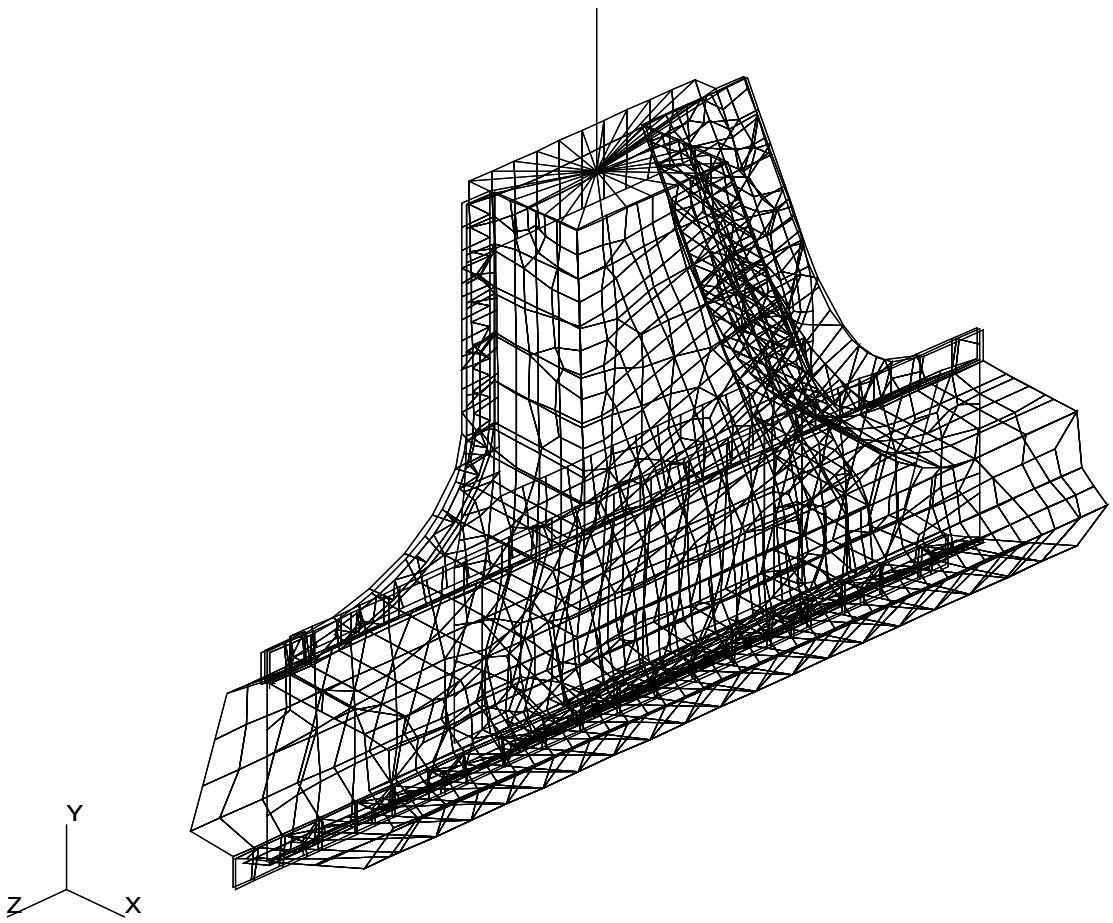


Figure 4.16: FEA Mesh of B-pillar to Rocker Joint

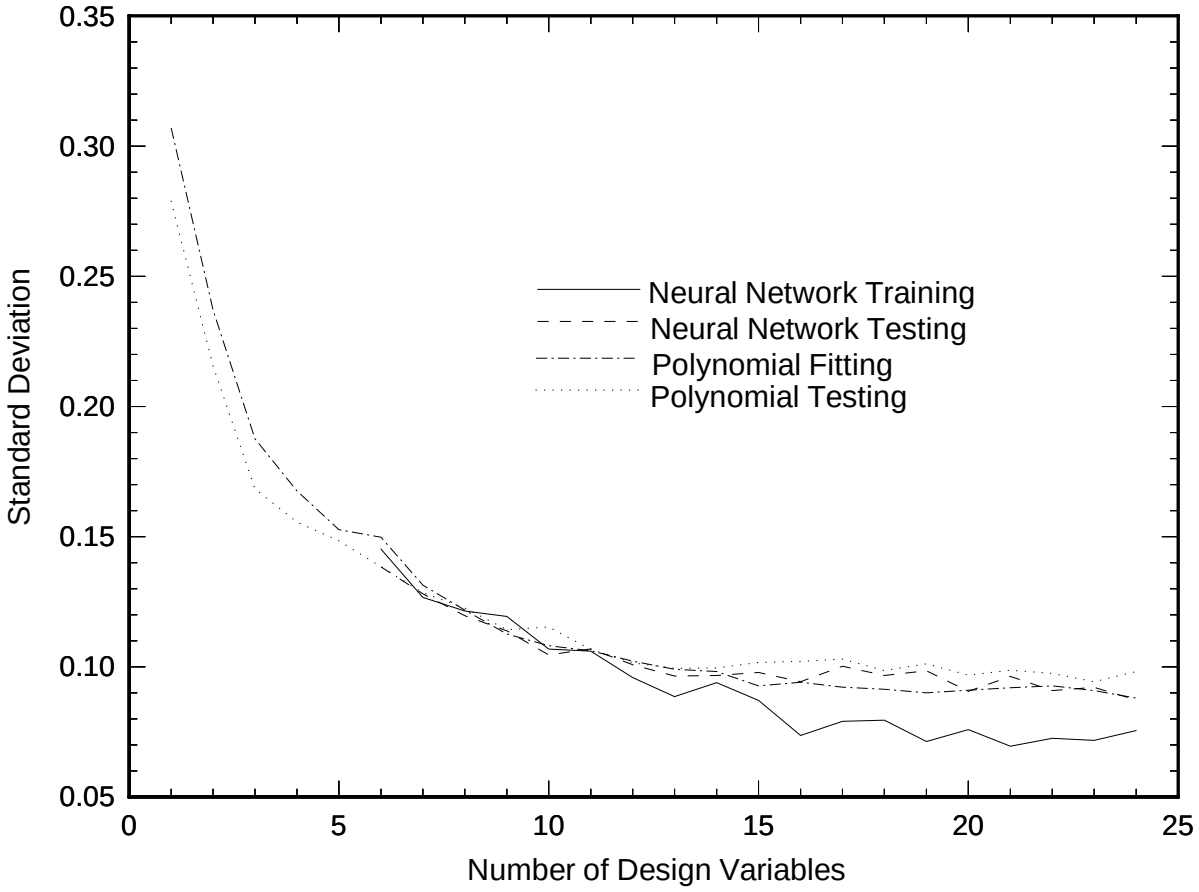


Figure 4.17: Comparison of Polynomial and Neural Network Results for I/O Stiffness

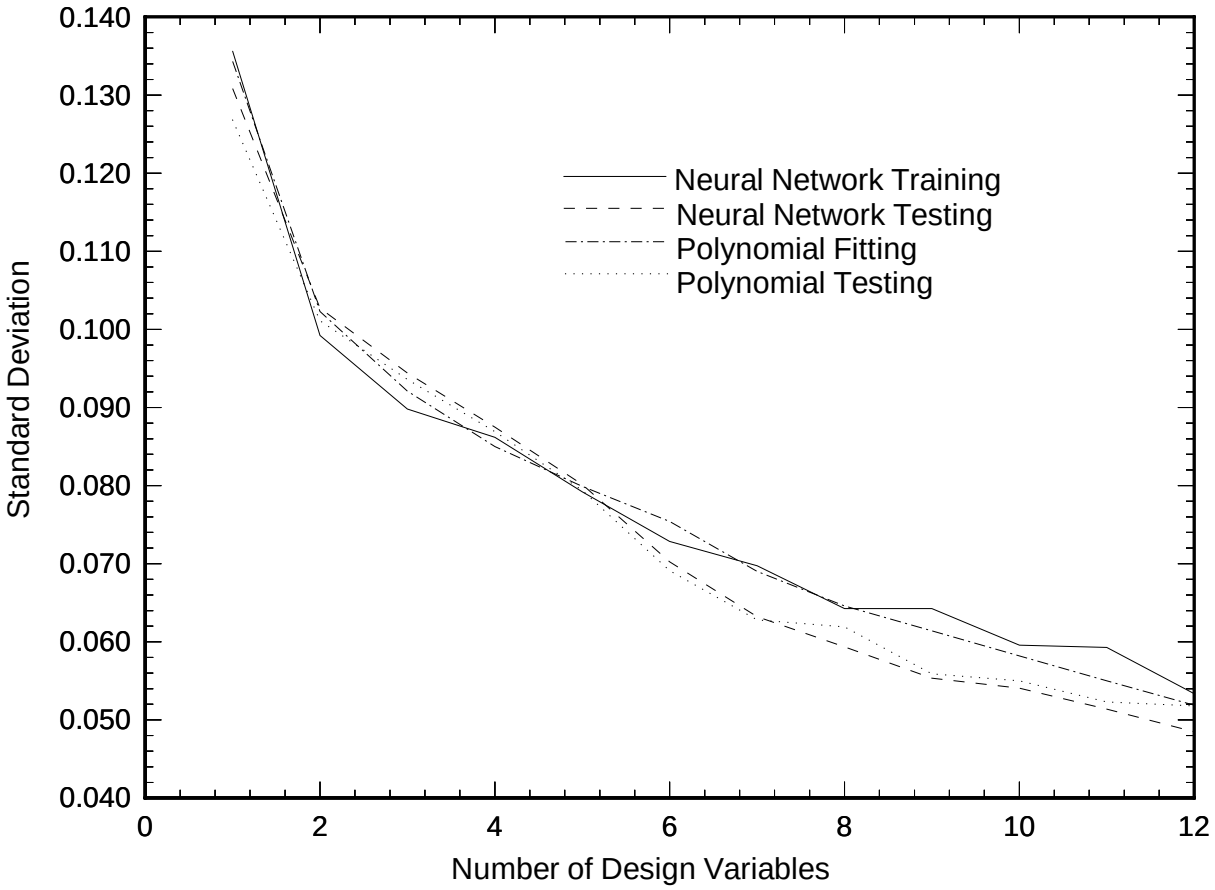


Figure. 4.18: Comparison of Polynomial and Neural Network Results for F/A Stiffness

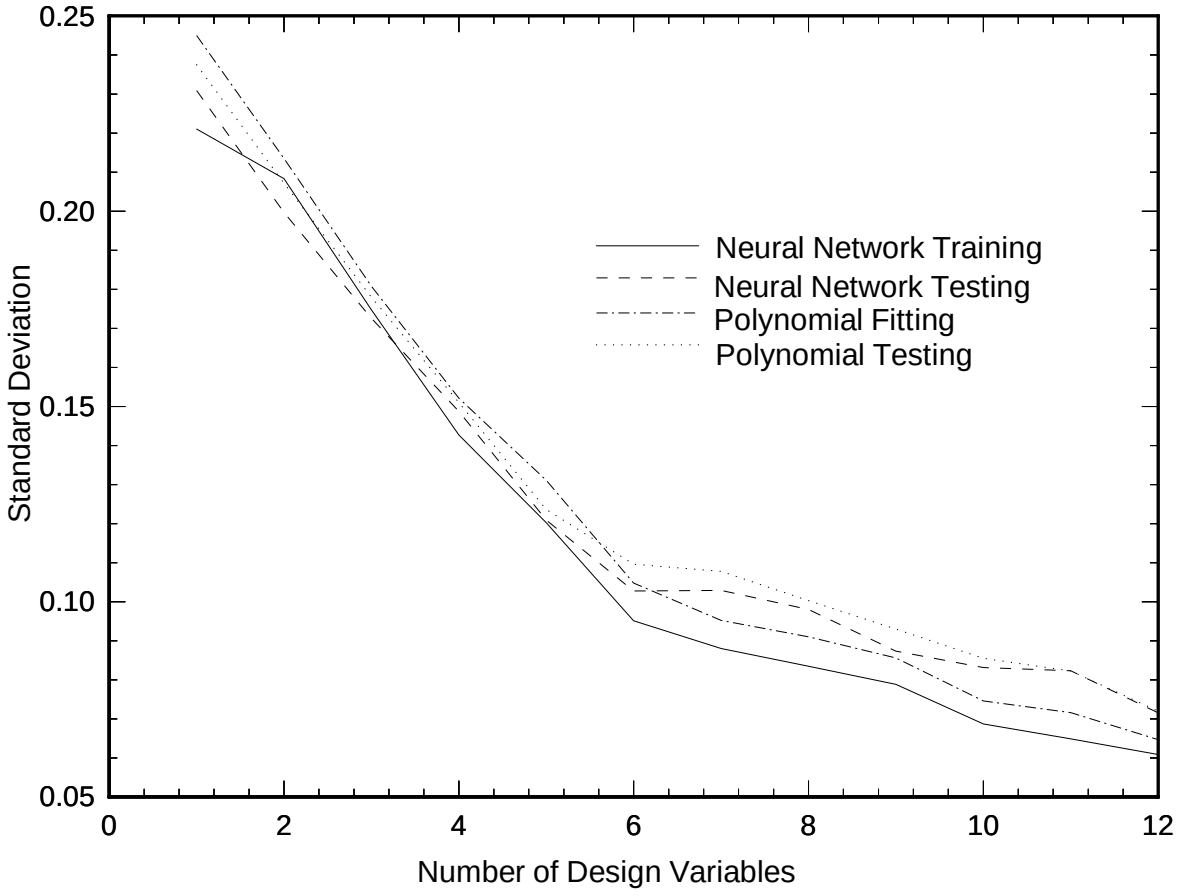


Figure 4.19: Comparison of Polynomial and Neural Network Results for Torsion Stiffness

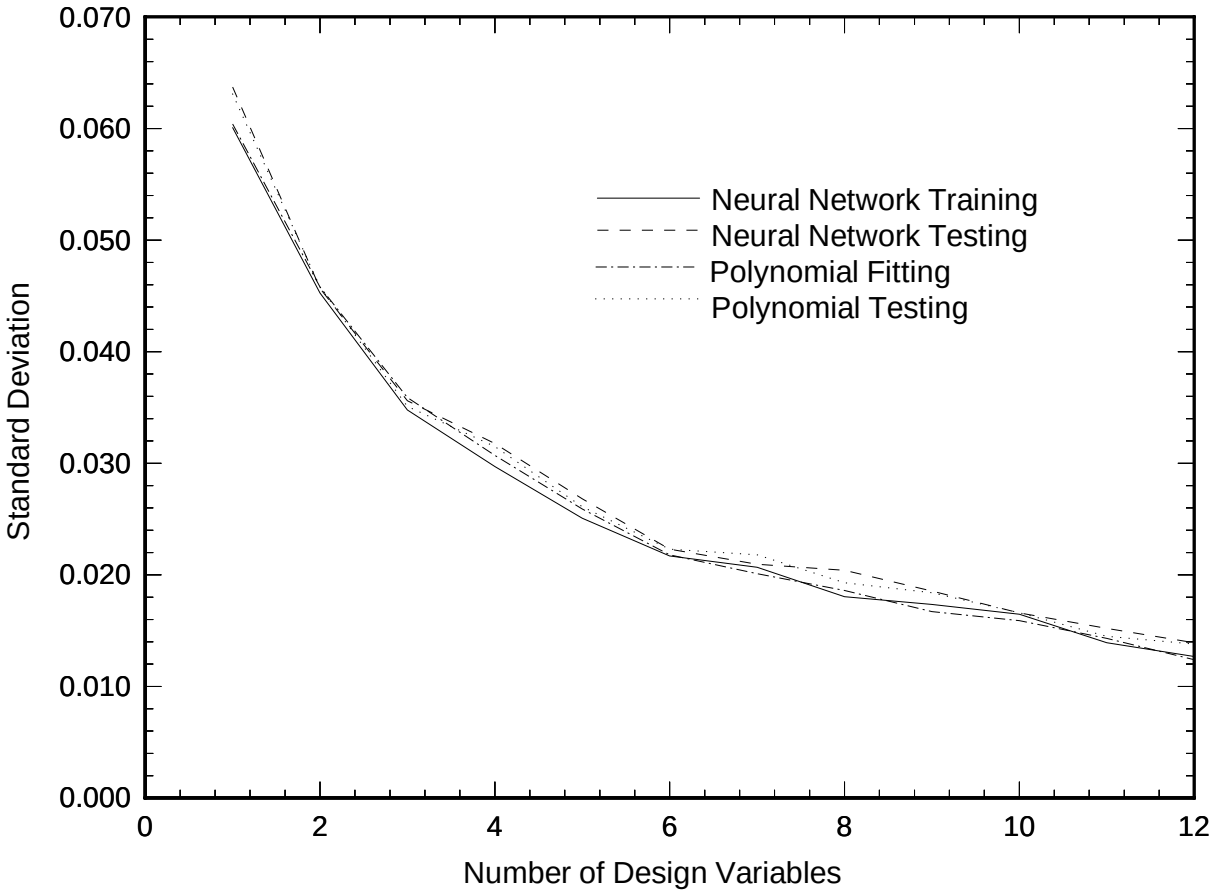


Figure 4.20: Comparison of Polynomial and Neural Network Results for Mass

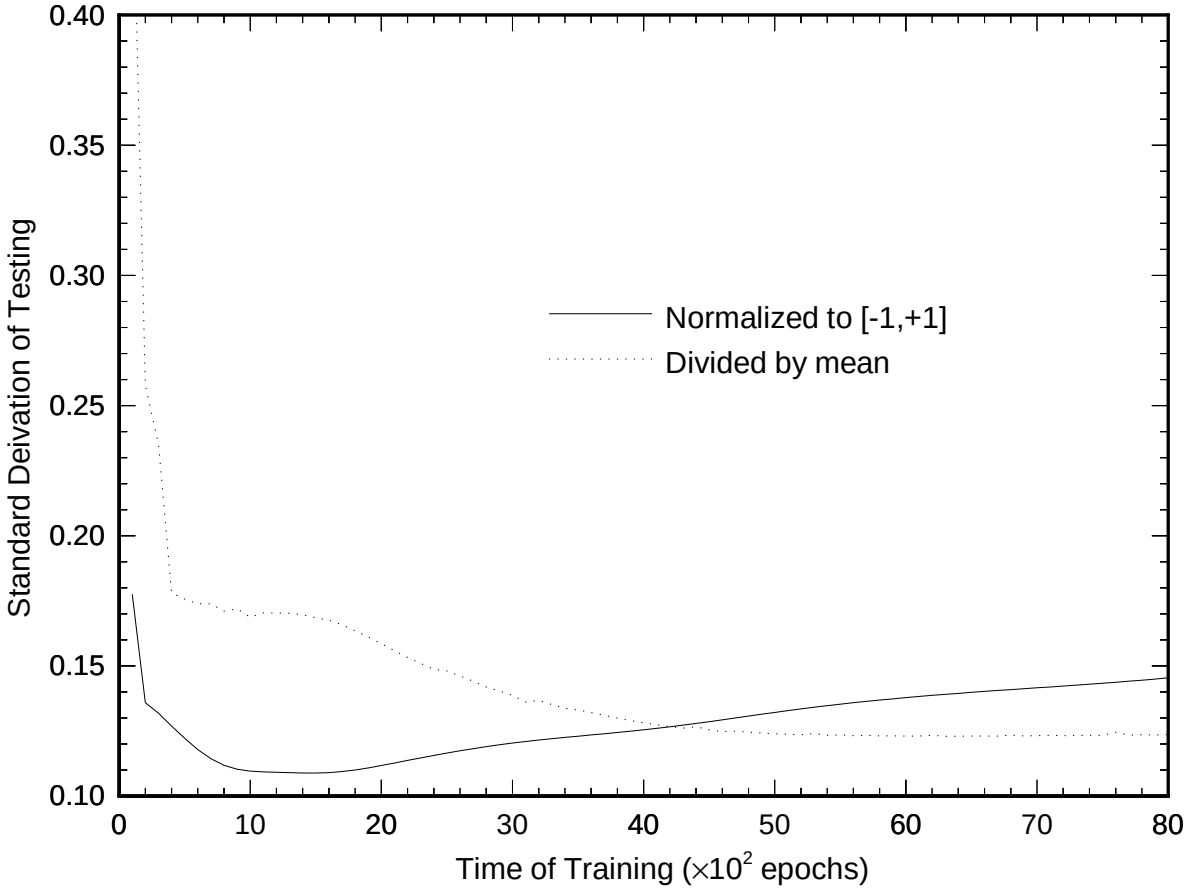


Figure 4.21: Comparison of Two Different Normalization Methods

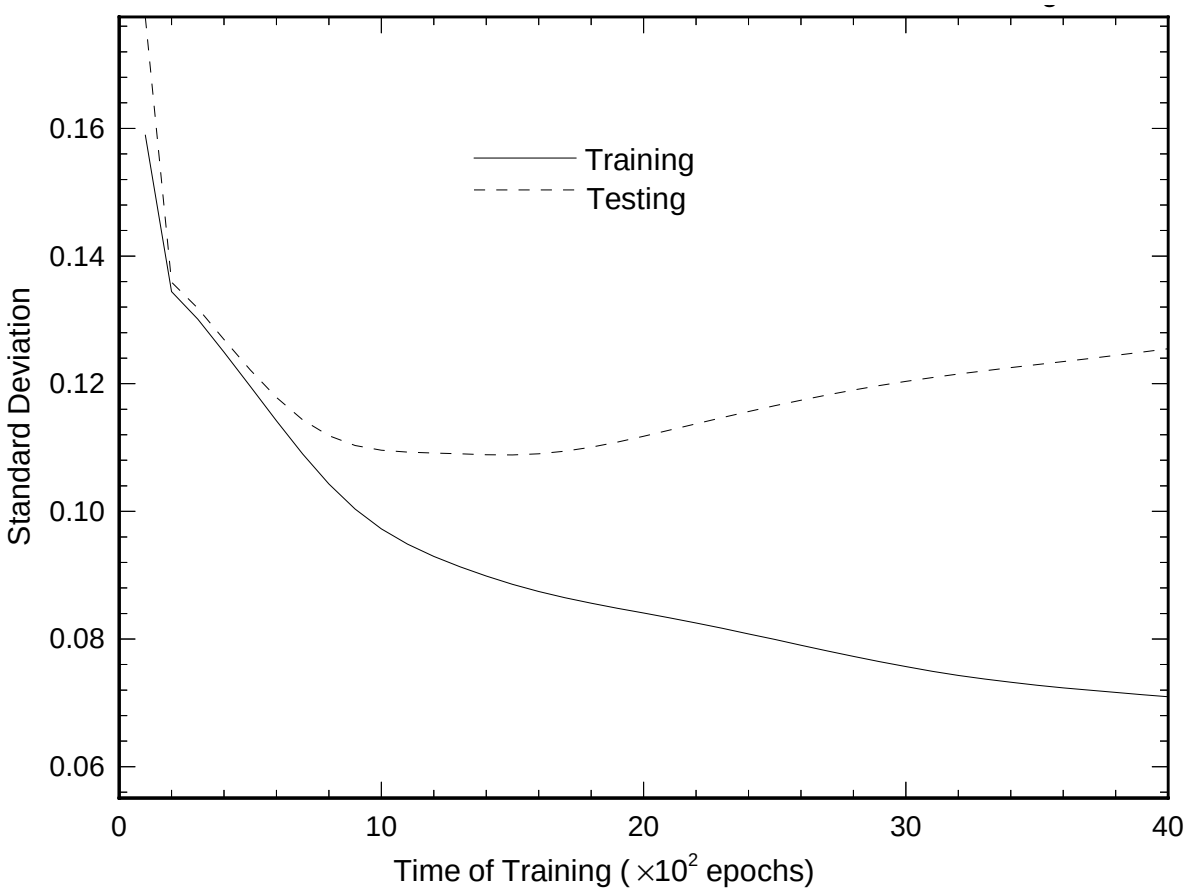


Figure 4.22: Relation Between Standard Deviation and Time of Training

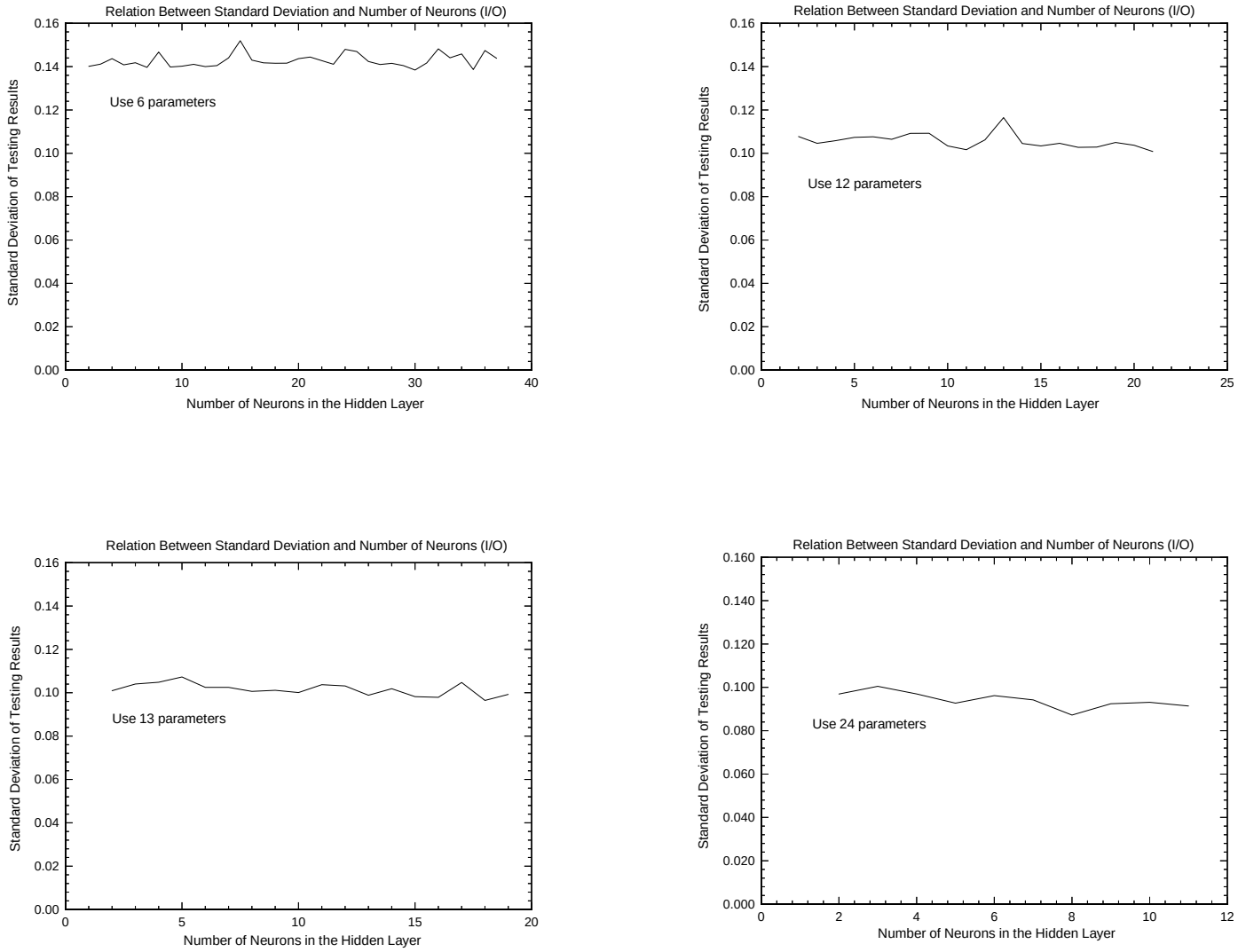


Figure 4.23: Relation Between Standard Deviation and Number of Neurons

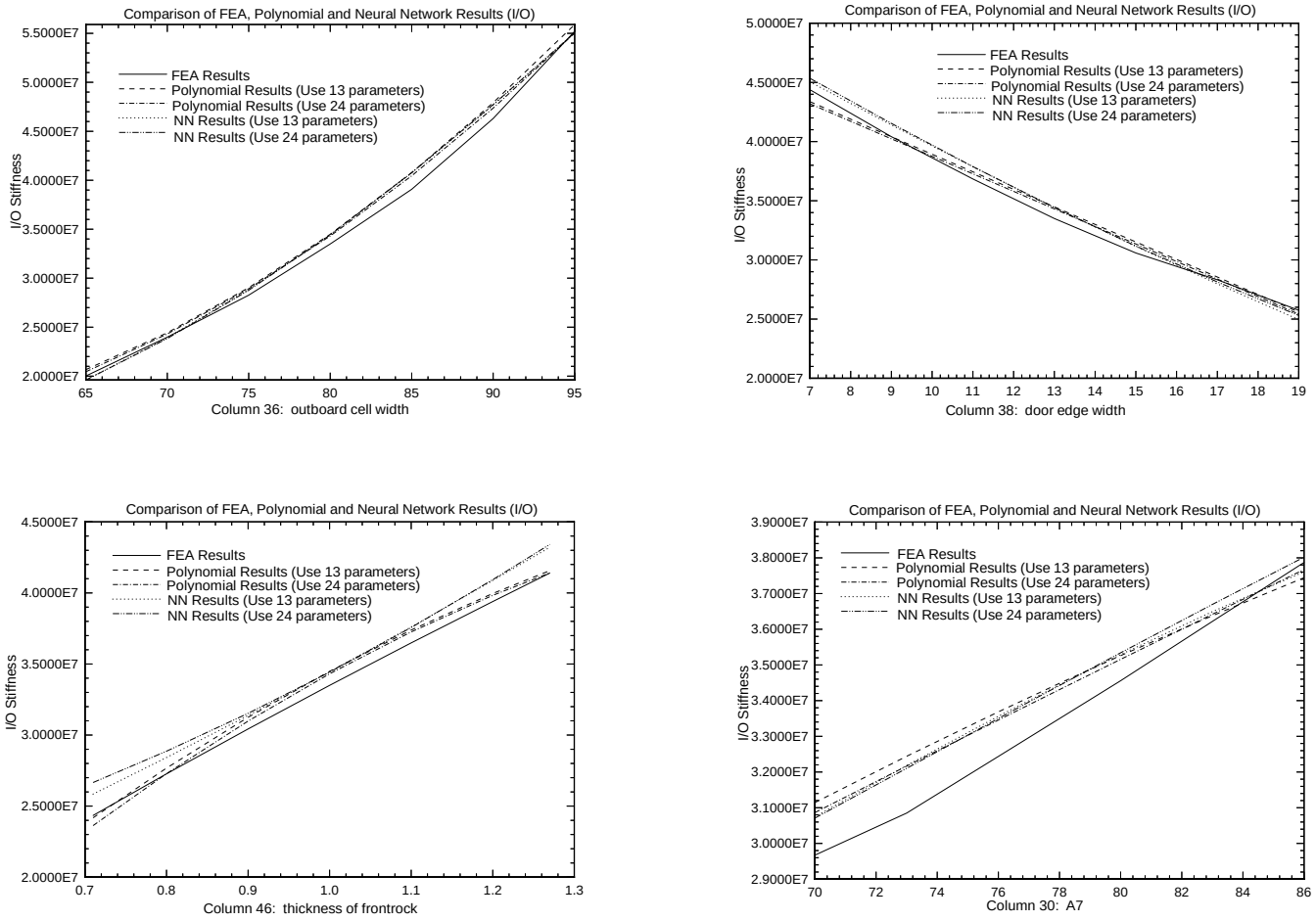


Figure 4.24: Comparison of FEA Results and Predictions of RSP and NN Translators for the I/O Stiffness of B-pillar to Rocker Joint

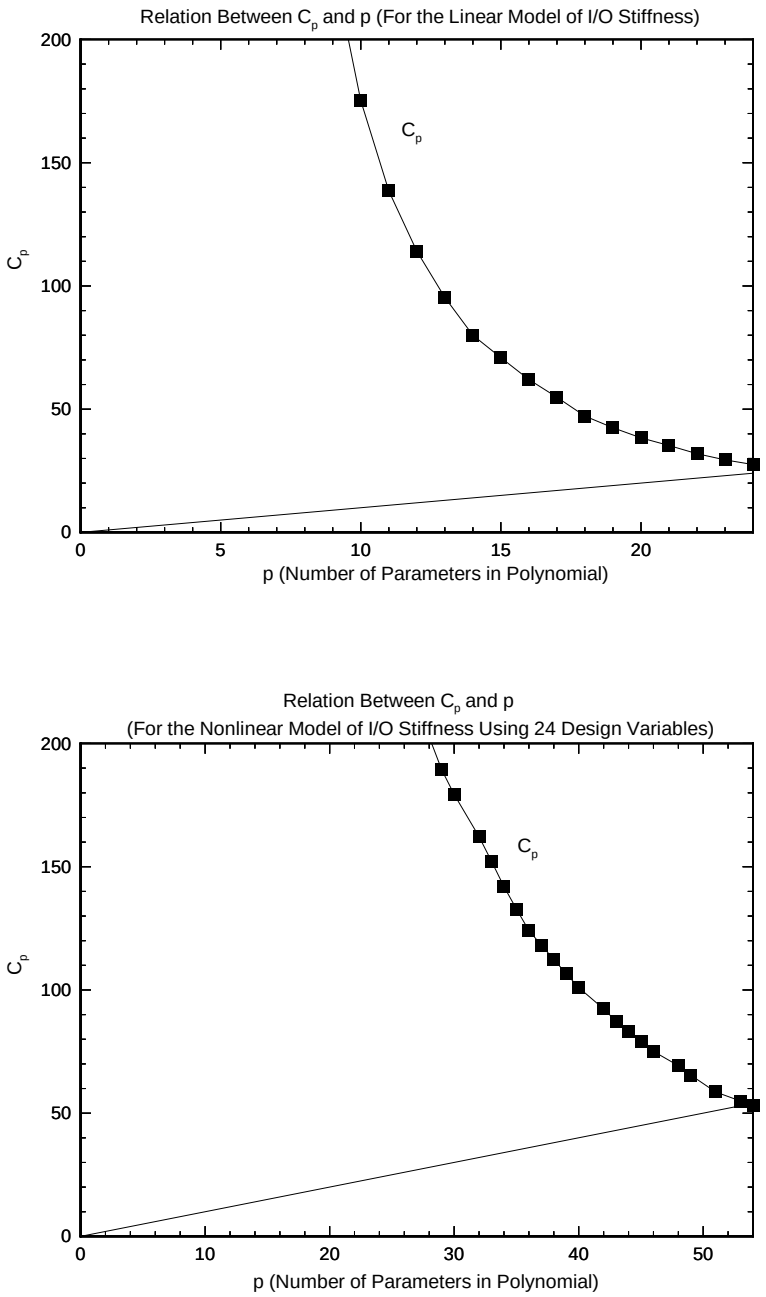


Figure 4.25: Relation Between Cp and p for I/O Stiffness

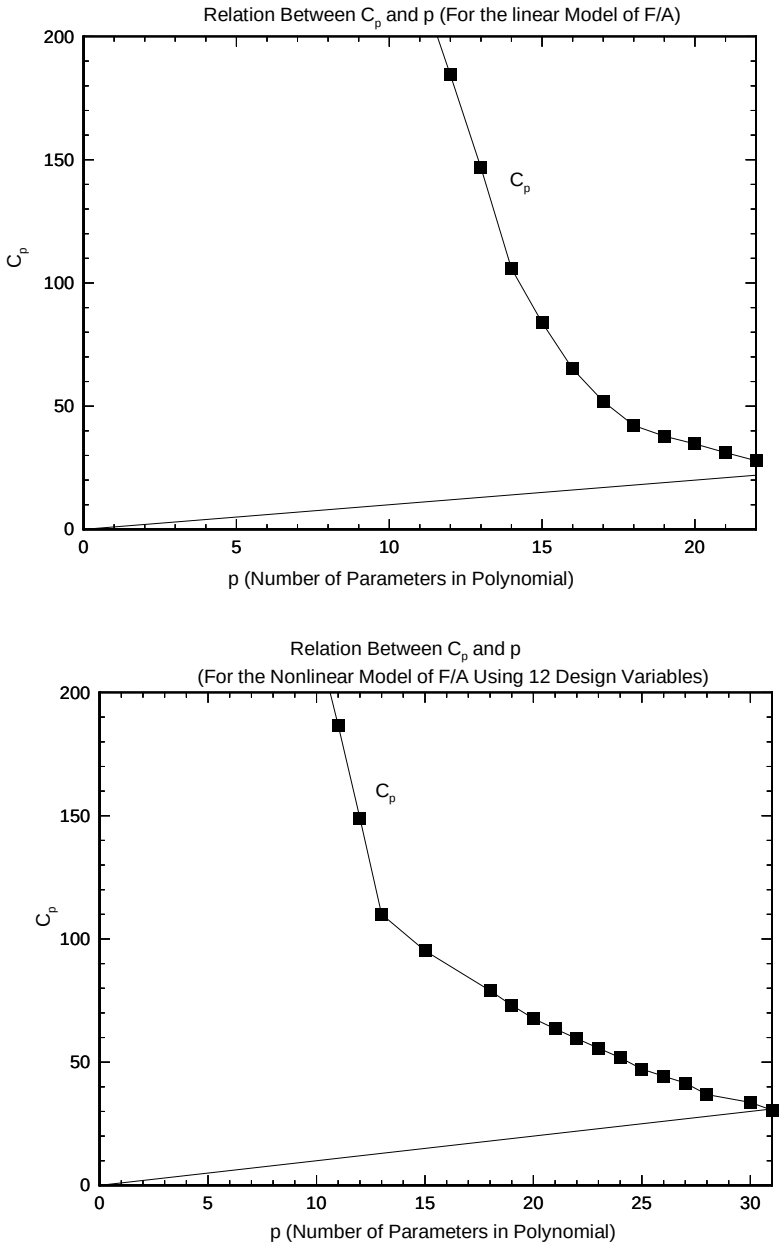
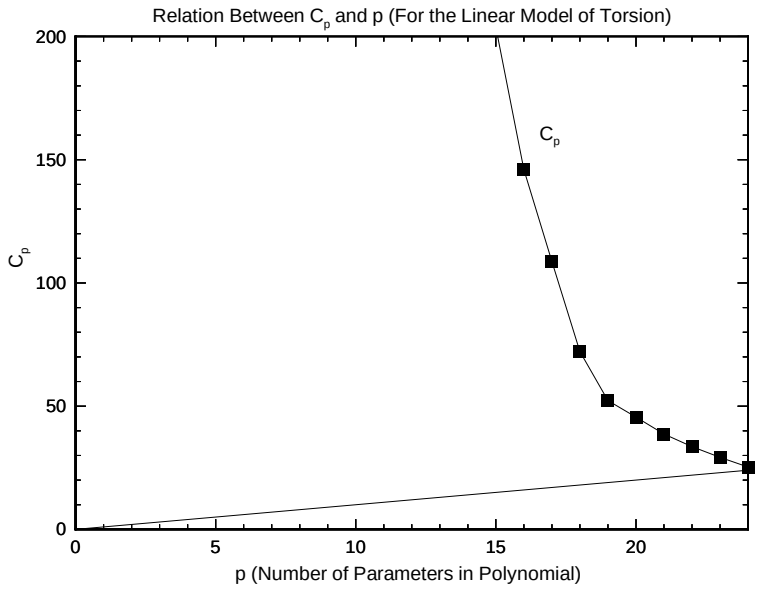


Figure 4.26: Relation Between Cp and p for F/A Stiffness



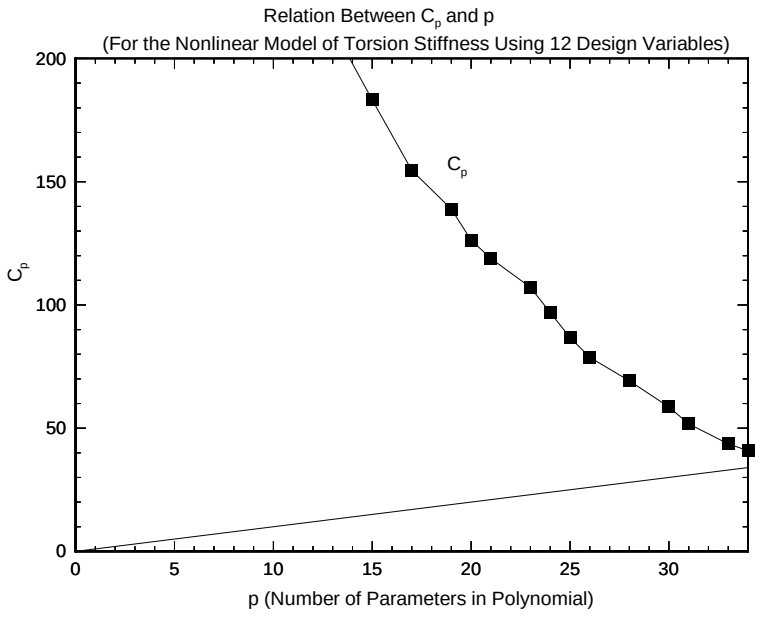
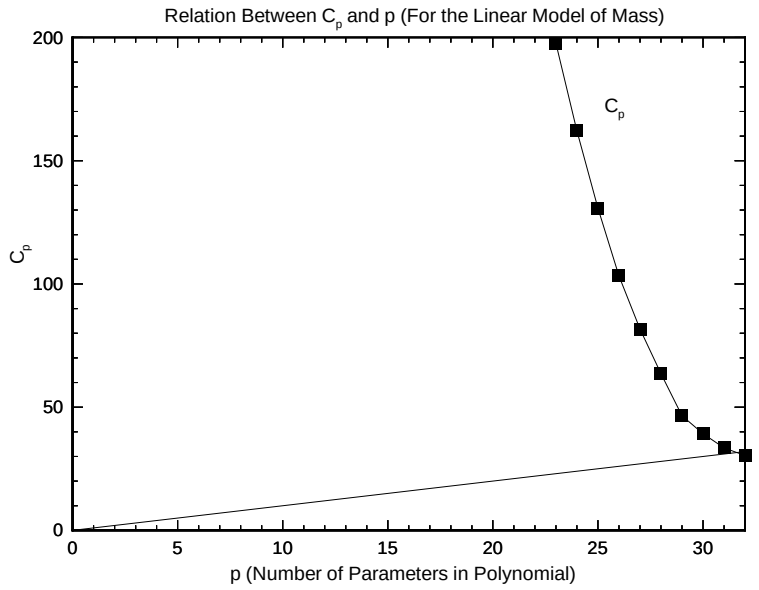


Figure 4.27: Relation Between Cp and p for Torsion Stiffness



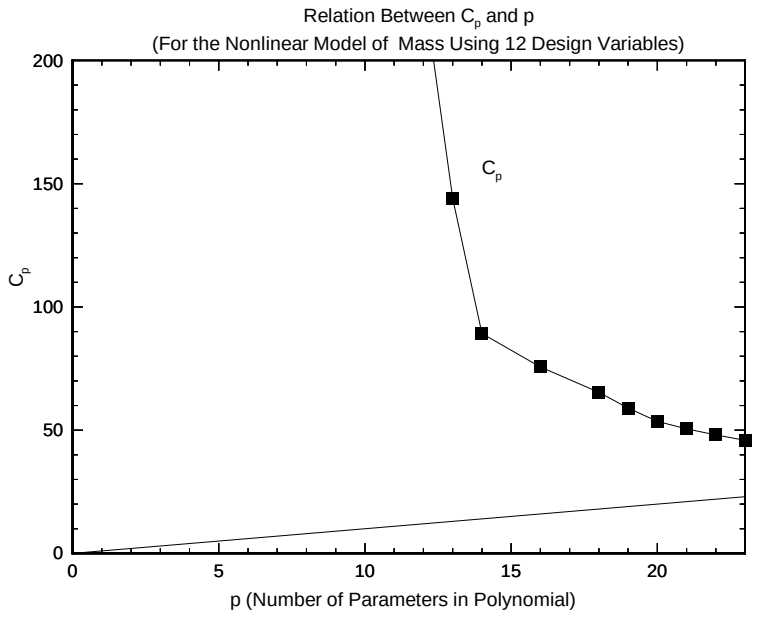


Figure 4.28: Relation Between Cp and p for Mass

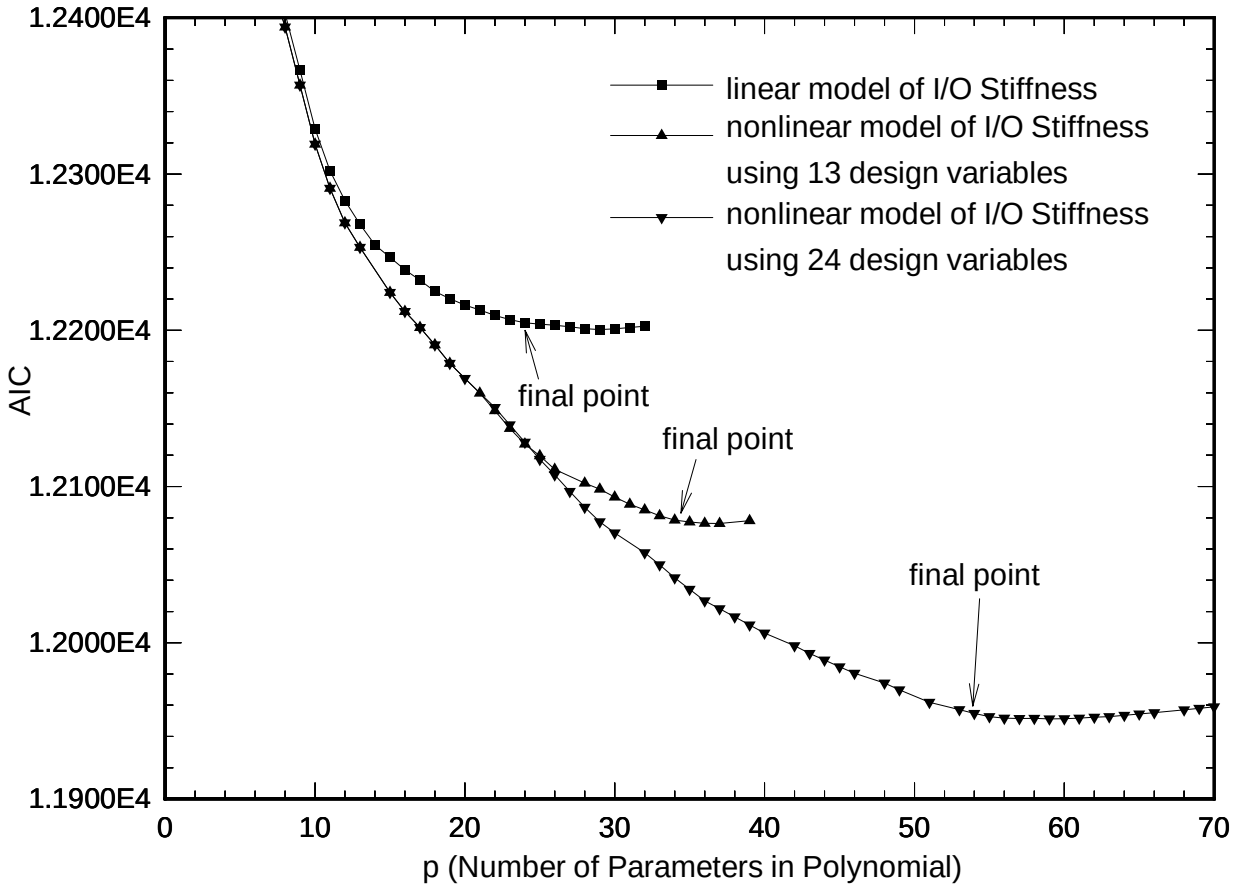


Figure 4.29: Relation Between AIC and p for Different Regression Models for I/O Stiffness

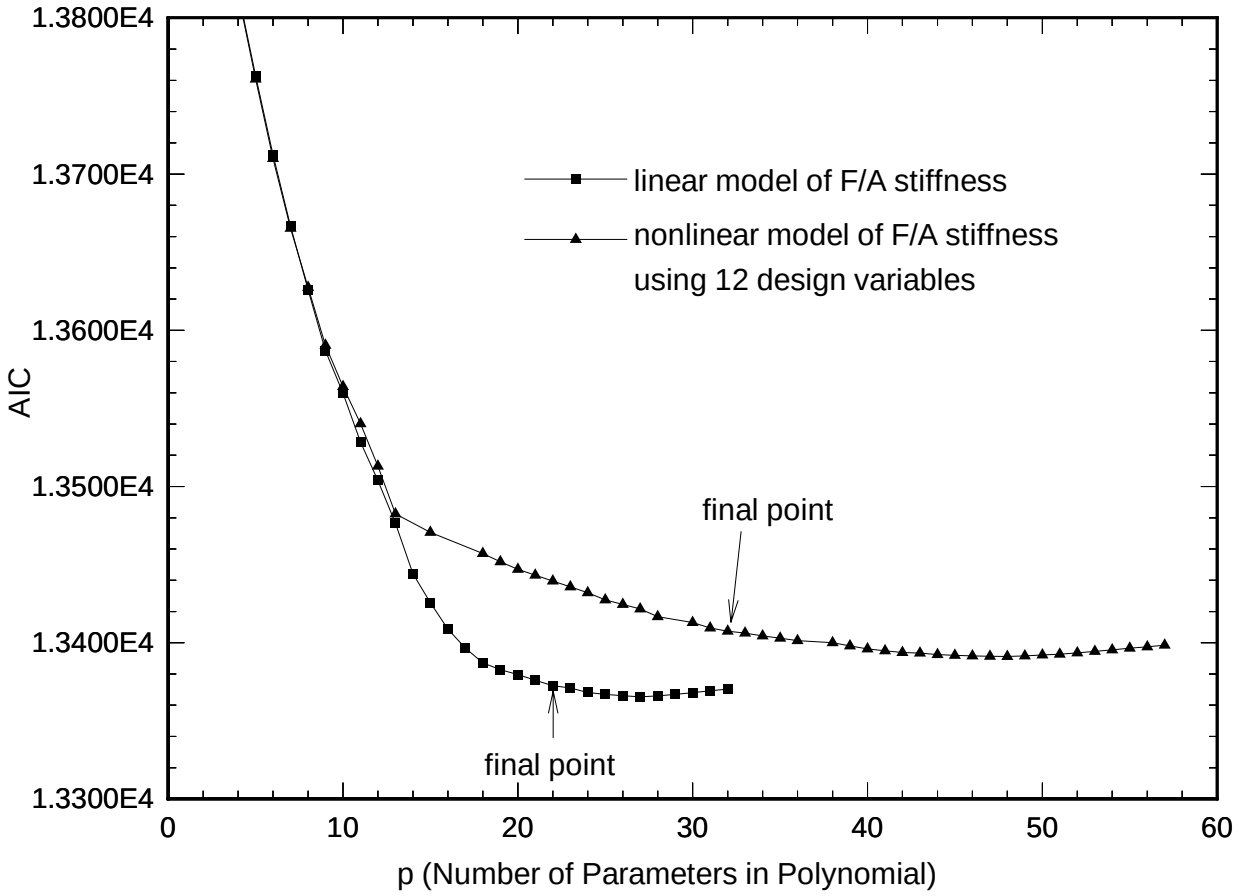


Figure 4.30: Relation Between AIC and p for Different Regression Models for F/A Stiffness

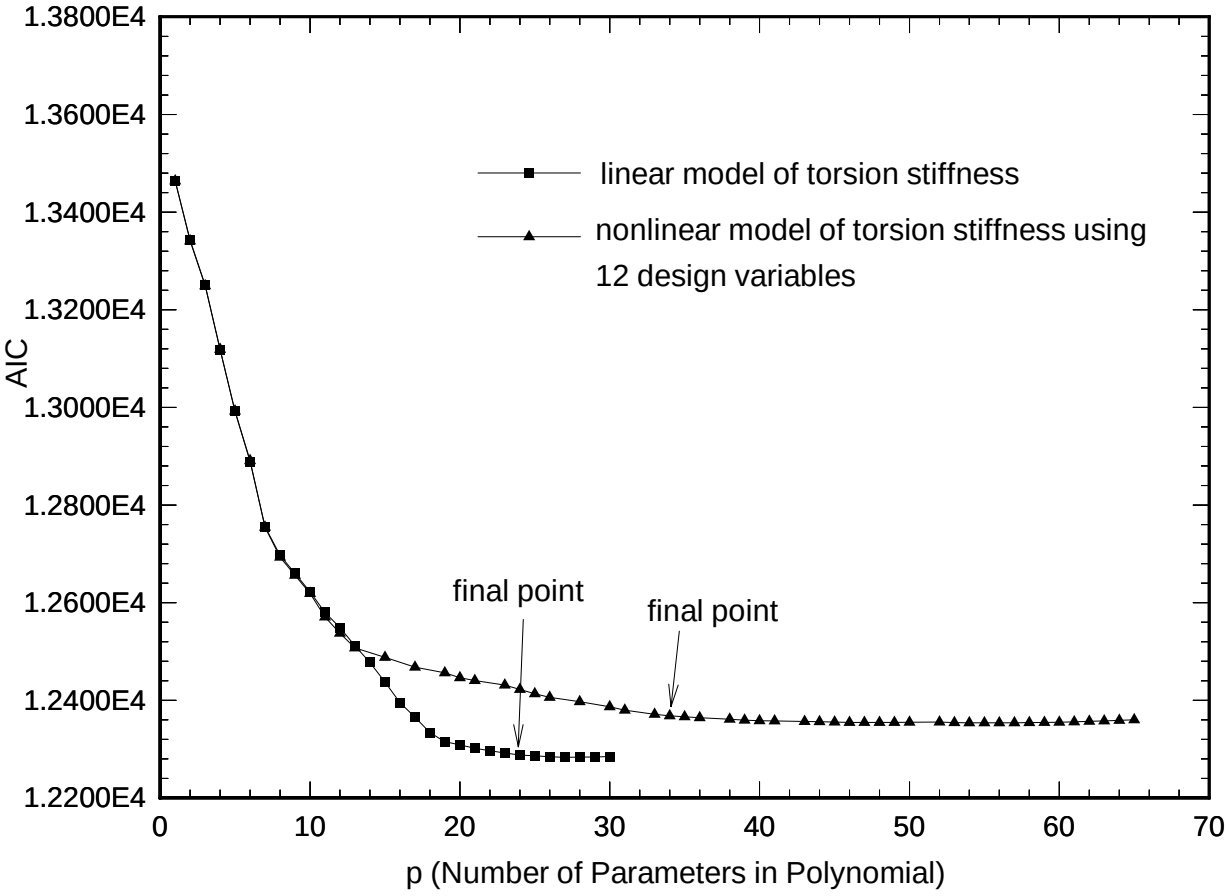


Figure 4.31: Relation Between AIC and p for Different Regression Models for Torsion Stiffness

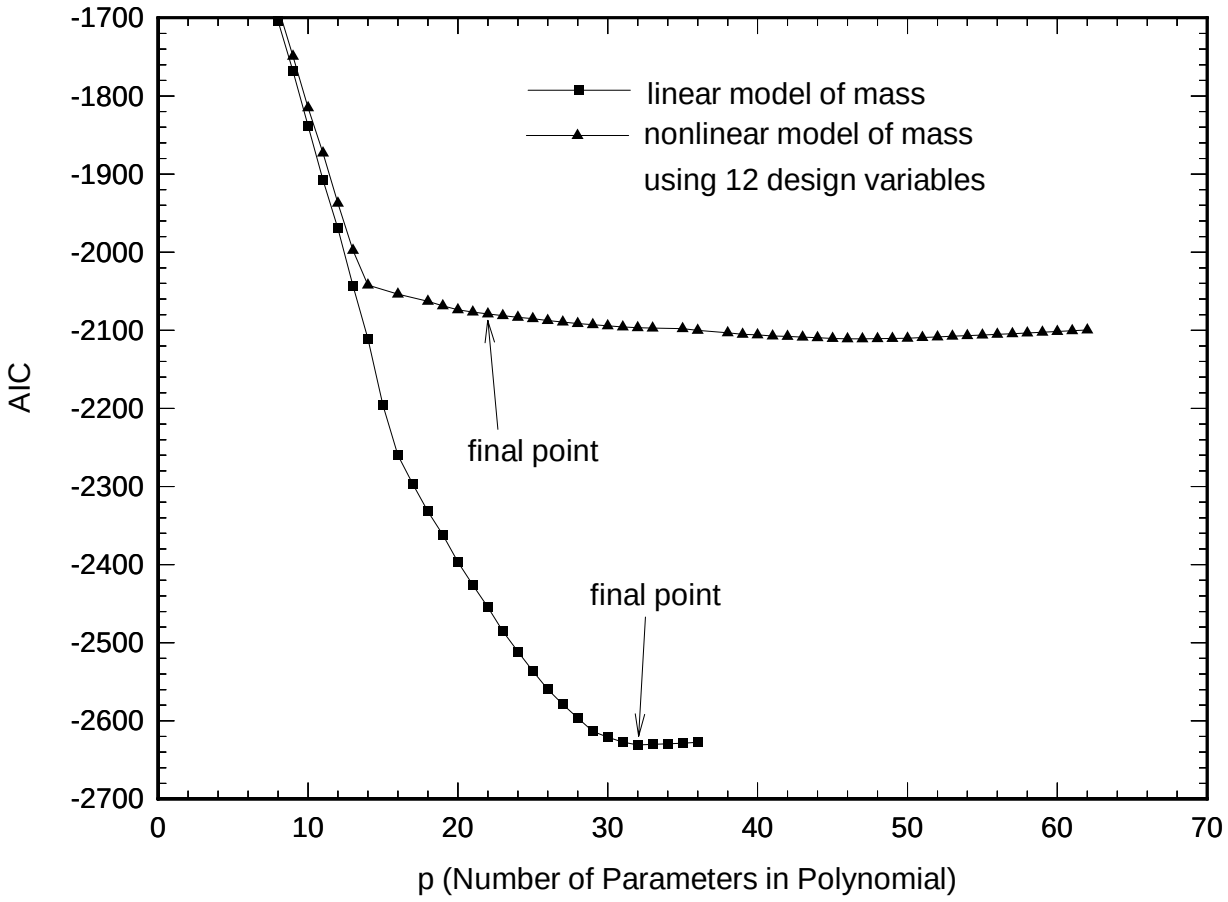


Figure 4.32: Relation Between AIC and p for Different Regression Models for Mass

Chapter 5
Developing Translator A for the A-pillar to Roof Rail Joint

5.1. Introduction

This chapter applies the general methodology for developing translator A to the A-pillar to roof rail joint. The A-pillar to roof rail joint is a "Y" shape joint (Figs. 1.1, 5.1a and 5.1b), located on the upper left and upper right sides of the driver and the passenger, respectively. The A-pillar joint supports the windshield of a car, the roof and part of the front door. The stiffness of the A-pillar to roof rail joint has a big effect on the safety in crash and the overall stiffness of the car body.

Section 5.2 describes the geometry of the A-pillar to roof rail joint. This section also describes a parametric Pro/Engineer model of this joint, and the procedure for FEA of the A-pillar to roof rail joint. Section 5.3 explains how we created a database for developing translator A. Section 5.4 describes the procedure for developing translator A using response surface polynomials and neural networks. Section 5.5 discusses the polynomial and neural network results.

5.2 Description of the A-pillar to Roof Rail Joint

The A-pillar to roof rail joint has three branches. The branch that is perpendicular to the side of a car is called header. The branch that is parallel to the top of the front door is called roof rail. A-pillar is the branch that lies between the windshield and the front door.

The header branch and the roof rail branch are generally perpendicular to each other. The A-pillar branch is inclined at an angle relative to the roof surface (Fig. 5.1a and 5.1b).

Most A-pillar to roof rail joints include four parts (Fig. 5.2):

- *Part 1* is the roof plate.
- *Part 2* is the outer cell of the joint.
- *Part 3* is the inner shell of the joint.
- *Part 4* is the lower shell of the header.

To increase the stiffness of the joint, most A-pillar joints also have reinforcements. The types of reinforcements vary greatly from one joint to another. By examining the hardware joints, we classified reinforcements into the following three types:

- Type 1: A reinforcement that connects the two flanges at A-pillar and roof rail cross sections (Fig. 5.3). This reinforcement is called *part 5* in this study.
- Type 2: A reinforcement that follows the shape of the top plate of the header and is connected to the bottom piece of header (part 4) to create a closed box (Fig. 5.3). This reinforcement is called *part 6.*
- Type 3: A reinforcement that follows the shape of the top plate of the A-pillar and roof rail (Fig. 5.2). This type of reinforcement is called *part 7*.

The A-pillar to roof rail hardware joints examined in this study had considerably different types of construction. The parametric model built in this study was based on some simplifications. The architecture of the parametric model was selected in consultation with engineers of an automotive company. This is close to the architecture of an actual car joint. The parametric model in this study can account for all three types of reinforcements mentioned above. The parametric model was created using Pro/Engineer. The parametric model is generic, and can be easily modified by changing the values of its design variables.

As explained in Chapter 2, we divided the design variables of a joint into six levels. The higher levels define the overall joint shape while the lower levels define the details of the joint geometry. The following sections describe these variables. First, we explain the higher level variables that define the orientation and position of each branch. Then, we define the variables that determine the cross section of each branch. Finally, the design variables defining the blending areas are explained.

5.2.1 Design Variables Defining the Orientation and Position of the Branches

To complete describe the geometry of the A-pillar to roof rail joint, we first define a global coordinate system (Fig. 5.4). The origin of the global coordinate system, *O*, is the intersection of lines tangent to the outboard flange of the A-pillar section and the outboard flange of the roof rail section. The x-axis of the global coordinate system is toward the outboard side of the car. The y-axis is vertical. The z-axis is toward the front direction of the car. All local coordinate systems at the cross sections of the three branches are defined relative to the global coordinate system.

To define the cross section of a branch, we first define a local coordinate system for that branch. The x-axis of this coordinate system is parallel to the axis of this branch, and its y-axis is parallel to the direction of one flange on the cross section of this branch. The z-axis is the cross-product of x and y axes. Each vertex on this cross section is defined relative to the origin of this local coordinate system.

As explained in Section 5.2, the header and the roof rail are perpendicular to each other. The axes of these two branches are defined to be parallel to x-axis and z-axis of the global coordinate system (Fig. 5.4).

The origin of the local coordinate system of the header branch is defined using its three coordinates relative to the global coordinate system. They are called *header_offest, header_vertical_offset,* and *header_horizontal_offset* (Figs. 5.5 and 5.6).

The x-axis of the roof rail branch coincides with the z-axis of the global coordinate system, so only one dimension is needed to define the local coordinate system of roof rail branch. This is called *roof_rail_offset* (Figs. 5.5 and 5.6).

To define the orientation and position of the A-pillar branch, we use two angles and one length (Figs. 5.5 and 5.6), namely, *theta, phi,* and *a_pillar_offset*. The two angles define the orientation of A-pillar relative to the global coordinate system. *A_pillar_offset* is the distance between the origin of the global coordinate system and that of the local coordinate system.

5.2.2 Dimensions Defining the Cross Section of the Header

The y-axis of the local coordinate system of the header is assumed horizontal, that is, parallel to the z-axis of the global coordinate system. Flange 1 and Flange 3 (Fig. 5.8) are also assumed parallel to the z-axis of the global coordinate system. Also, the upper and lower plates of the header are assumed horizontal.

In general, we need $2(n_{edge}-1)$ design variables to define a cross section made up of straight plates, where n_{edge} is the number of edges. The header has seven edges. Therefore, we need 12 design variables. The assumption mentioned in the previous paragraph reduces the number of variables from 12 to eight. The dimensions that define the cross section of the header have the prefix "H_". Eight dimensions (Fig. 5.8) are needed to completely define the cross section of the header.

5.2.3 Dimensions Defining the Cross Section of the Roof Rail

The top plate of the roof rail branch is assumed to be parallel to the x-axis of the global coordinate system. Flange 3 is also assumed to be parallel to the x-axis of the global coordinate system. The dimensions used to define the cross section of roof rail have the prefix "RR_". Twelve dimensions are needed to completely describe the geometry of the roof rail section (Fig. 5.9. *Flange2_width* and *flange3_width* are not shown).

5.2.4 Dimensions Defining the Cross Section of the A-pillar

The cross section of the A-pillar is the most difficult one to parameterize. Two sets of dimensions are used to describe the geometry of the A-pillar cross section. The set of dimensions that have prefix "AP_" define the outer plate shell of the A-pillar section (Fig. 5.10). The other set of dimensions with the prefix "AP_reinf_" define the reinforcement of the A-pillar (Fig. 5.11). Eighteen dimensions (dimensions *flang1_width* and *flange2_width* are not shown in Figs. 5.10 and 5.11) are used for the cross section of the A-pillar.

5.2.5 Blending Radii

Three dimensions, namely, *H_blending_rad*, *RR_blending_rad*, and *AP_blending_rad*, define the blending radii of the joint (Fig. 5.5 and 5.6). Since a joint is created by first extruding the cross section of each branch, and then connecting the cross sections of two branches using blending curves, the extrusion distance of a branch is equal to the value of the offset distance of this branch subtracted by its blending radius. Therefore, the blending radii also determine the extrusion distances of the three branches.

5.2.6 Connections

Ten dimensions are used to define the flanges and the spot welds. These dimensions include the dimensions that define the widths of flanges like *flange1_width*, the dimension defining the location of spot welds on the flange (*distance_from_edge*), the

dimension that defines the distance between two adjacent spot welds (*spot_weld_spacing*), and the angles between the flanges and their adjacent plates like *flange1_angle_down* and *flange1_angle_up* (Fig. 5.7-5.10).

5.2.7 Other Dimensions

Dimensions *part2_cut_distance, part3_cut_distance, part4_extension,* and *part7_cut_distance* complete the definition of the geometry of parts 2-4 and 7 (Figs.5.12-5.15).

5.2.8 Parametric Model and FEA of the A-pillar to Roof Rail Joint

After consulting engineers from an automotive company, we selected one type of construction to develop translator A. This type has the four parts that most A-pillar joints have, e.g., parts 1 to 4. Only one type of reinforcement, part 7, was considered in developing translator A. Parts 5 and 6, which were found in only a few joints, were not considered in the parametric model for developing translator A.

A parametric model of the A-pillar to roof rail joint was developed based on the above variables. Pro/Engineer was used. The model has 48 independent variables, and can simulate any feasible joint design by changing the values of the design variables of these models. The Pro/Engineer model can account for packaging, manufacturing, and styling constraints, and can warn the user if a constraint is violated.

After developing the geometric model corresponding to a joint design, we use Pro/Engineer to generate a FEA model and a NASTRAN bulk data file. The size of mesh in Pro/Engineer controls the size of the elements in the FEA model. We studied the effect of the mesh size on the FEA results of a design. Table 5.1 shows the results. It is observed that the mesh size affects slightly the FEA results. The difference for models with same geometry but different mesh size is less than 3%. A FEA model with smaller

mesh size tends to give smaller predictions for the I/O, F/A, torsion stiffness, and mass than models with larger mesh size.

We also compared the predictions of the stiffnesses of two joints of two cars, called car A and car B, from the Pro/Engineer model with the experimental results (Table 5.2). It is observed that the FEA model predicts the I/O and torsion stiffnesses more accurately than the F/A stiffness. The errors of predictions of I/O and torsion stiffnesses for car A are –4.0% and –5.9%, respectively. For car B, the errors are 5.9% and –19.1%, respectively. The error in FEA result for F/A stiffness is 45.3% for one car and–15.8% for the other car. The prediction of F/A stiffness for the first car is big compared with predictions of stiffness in the other directions. This is due to several reasons. a) The Pro/Engineer model used in this study simplifies the actual joint hardware. Some details such as access holes are neglected. Moreover, part 7 in the parametric model is assumed to follow the shape of the outer shell of the A-pillar, which simplifies the actual part 7. b) In experiments, when measuring the F/A stiffness, engineers apply a moment in the horizontal direction on the cross section of A-pillar. Then, they measure the rotation of A-pillar in the horizontal direction, and calculate the stiffness using the following expression.

$$\text{Stiffness} = \frac{\text{Moment in the given direction}}{\text{Resulting rotation in same direction}} \tag{5.1}$$

Since in practice, it is difficult to define the horizontal direction of the A-pillar section, the actual moment and rotation may not be in the horizontal direction. As a result, the measured F/A stiffness is different from that in the model. Because F/A stiffness is sensitive to the direction, this also contributes to the difference between the predicted and the measured F/A stiffness for one car.

5.3 Developing a Database

To develop translator A, we first develop a database. The database stores the values of dimensions of many designs and their performance characteristics (stiffness and mass) obtained using FEA. The input values of the design variables of a design should make a feasible Pro/Engineer model in order to use the parametric Pro/Engineer model to generate a MSC/NASTRAN bulk data file. Several methods can be used to create those designs, including Taguchi method, linear method, and random method. This study uses the random method to generate these designs as explained in Chapter 2.

Before using a random number generator, we first measured the dimensions of six actual joints that were provided to us by an automotive manufacturer (Table 5.3). From these measurements we found the maximum and minimum values of each design variable. To account for more designs, we increased the ranges of design variables by 10% when developing the database (Table 5.4).

Among all the design variables of the A-pillar joint, some are generally fixed because of manufacturing rules, such as the width of flanges like *flange1_width*, the location of spot welds on flanges (*distance_from_edge*) and the distance between two adjacent spot welds (*spot_weld_spacing*) (Fig. 5.7). Some dimensions are fixed because of styling requirements, such as the orientation of the A-pillar (*theta* and *phi*). To compare the results of different designs fairly, the lengths of the branches should be fixed. Unfortunately, there is not widely accepted rule for defining reference points with respect to which the lengths of the branches are measured. We found that the branches of the actual joints had different lengths. To circumvent this problem, the branches of the three branches were considered as variables and they varied when creating the database for translator A.

First, we used a random number generator to create random designs. The Pro/Engineer model checked if each design was feasible. Infeasible designs were screened out. Then using Pro/Engineer, the model corresponding to each feasible design was generated and a NASTRAN bulk data file was created. Figure 5.16 shows a FEA model of the A-pillar to roof rail joint. 600 models corresponding to feasible designs were developed. These models were analyzed using MSC/NASTRAN to obtain the stiffness and mass. The dimensions of all feasible designs and their performance characteristics were stored in a database that was later used to develop translator A. There were 600 designs in the database for the A-pillar to roof rail joint.

5.4 Developing Translator A

Two different methods, namely, response surface polynomials and neural networks, were used to create translator A.

As described in Chapter 2, the development of translator A involves several steps. First, the design variables are ranked in terms of their effects on the stiffness and mass. Then, different response surface polynomial models are tested. We choose the best regression model according to the results. Finally, neural networks with different architectures are considered, from which we choose the neural network translators. We also compare the response surface polynomial and neural network results.

5.4.1 Ranking Important Design Variables

Many design variables are needed to completely describe the geometry of a joint. However, only a few affect significantly the stiffness and mass of the joint. It is also important for a designer to know the relative importance of each design variable so that the designer can determine the most effective way to improve the performance of a design.

The importance of design variables on the stiffness and mass was ranked using linear polynomial models. Stepwise regression was used in this study to find the best linear polynomial model. See Chapter 2 for details.

Stepwise linear regression calculates the *F*-ratio of each parameter. *F*-ratio is the ratio of the decrease in the sum of squared errors of the results when a parameter is included in the regression model over the mean squared error of the final regression model. The *F*-ratio can be used as a measure of the importance of design variables. The design variable with bigger *F*-ratio has bigger effect on the output. According to the value of the *F*-ratio, the design variables are listed from the one with the biggest *F*-ratio to the one with the smallest *F*-ratio. See Tables 5.5 to 5.8 for the rankings of important design variables for I/O, F/A, torsion stiffness and mass of the A-pillar to roof rail joint.

5.4.2 Polynomial Translators

We developed and tested three polynomial models, including linear polynomial models, second degree polynomial models, and double regression models. The following sections present the procedure for developing these models.

Normalization of the Torsion Stiffness

Before discussing different regression models, we first explain the normalization of the torsion stiffness. Figure 5.17 shows that the torsion stiffness correlated with the ratio of the thickness of part 3 over the offset of the A-pillar (*Thickness of part 3 / A_pillar_offset*). The correlation coefficient is 67.1%. By dividing the torsion stiffness by above quotient, we can reduce the range of the resulting normalized torsion stiffness. It is found that normalizing the torsion stiffness improves the fitting and testing results.

Linear Polynomials

A linear polynomial model was first obtained using stepwise regression. Only the important parameters were included in the regression model. Figure 5.18 shows the

scatter plots of the predictions of the linear model versus FEA results for the I/O, F/A, torsion stiffnesses and mass. There are 25 parameters (24 design variables) in the linear models for both I/O and F/A stiffnesses. The linear models for the torsion stiffness and mass have 29 and 32 parameters, respectively.

The predictions from the linear polynomials were not satisfactory. The standard deviations of the ratio of predicted value over FEA result were 15.8% and 12.3% for the I/O and F/A stiffnesses, respectively. For the torsion stiffness and mass, the standard deviation of the fitting results were 13.9% and 2.2%, respectively.

Second Degree Polynomials

To improve the accuracy of the predictions of the translator, we also considered second degree polynomials. Figure 5.19 shows the scatter plots of the results obtained using second degree polynomials versus the FEA results for the I/O, F/A, torsion stiffnesses and mass.

The results from the second degree polynomials were better than those from the linear polynomials (Tables 5.9-5.12). The standard deviations of fitting results for the I/O, F/A, torsion stiffnesses and mass were reduced to 13.7%, 10.6%, 13.8% and 2.0%, respectively.

Double Regression models

From Figure 5.18, it is observed that the predictions from the linear polynomial models are not evenly distributed along the two sides of a straight line inclined at 45 degrees relative to the horizontal axis. For very low values and very high values of the stiffness from FEA, the translator underestimates the stiffness, whereas for values of the stiffness from FEA close to the mean it overestimates the stiffness. Figure 5.20a shows the relation of the difference between the predictions and FEA results versus the FEA results for I/O stiffness. Figure 5.20b shows the predicted I/O stiffness from the linear

polynomial model versus the FEA results superimposed on a cubic polynomial. It is observed that the predicted results are almost evenly distributed along the two sides of the cubic polynomial. Based on such observation, a second regression is performed which uses a polynomial (cubic polynomials for I/O, F/A, quadratic polynomials for torsion and mass) to simulate the relation between the predicted results from the linear regression model and the FEA results. The following is the formulation for the second regression.

$$\hat{y}_D = \sum_{i=0}^{n_D} c_i \, (\hat{y}_L)^i \tag{5.2}$$

where $\hat{y}_D$ and $\hat{y}_L$ are the values of the dependent variables predicted from the double regression and the linear polynomial models, respectively. n_D is the degree of the double regression model. $n_D = 3$ for the I/O and F/A stiffnesses, and $n_D = 2$ for the torsion stiffness and mass. Coefficients c_i in Eq.5.2 are determined using regression.

Figures 5.21a and 5.21b show the relations between the predictions from the double regression models and the FEA results for the fitting and testing results. It is found that the double regression model significantly improves the prediction compared with the linear regression model (Fig. 5.18). Moreover, the error is still significant for joints with high stiffness. The predictions of the translator are unbiased, with the exception of torsion stiffness.

Explanation of Double Regression

From Tables 5.9 to 5.12, we find that the double regression models are more accurate than the linear and second degree models. The following is an explanation. For simplicity, suppose that there are only two design variables in the linear polynomial model, and a quadratic polynomial is used in the second regression. The linear polynomial is:

$$\hat{y}_L = b_0 + b_1 x_1 + b_2 x_2 \tag{5.3}$$

Substituting the above equation into the equation of double regression (Eq.5.2), we have

$$\begin{aligned}
\hat{y}_D &= c_0 + c_1 \hat{y}_L + c_2 \hat{y}_L^2 \\
&= c_0 + c_1 b_0 + c_1 b_1 x_1 + c_1 b_2 x_2 + c_2 b_0^2 + \\
&\quad c_2 b_1^2 x_1^2 + c_2 b_2^2 x_2^2 + 2c_2 b_0 b_1 x_1 + 2c_2 b_1 b_2 x_1 x_2 + 2c_2 b_0 b_2 x_2 \\
&= (c_0 + c_1 b_0 + c_2 b_0^2) + (c_1 b_1 + 2c_2 b_0 b_1) x_1 + (c_1 b_2 + 2c_2 b_0 b_2) x_2 + \\
&\quad 2c_2 b_1 b_2 x_1 x_2 + c_2 b_1^2 x_1^2 + c_2 b_2^2 x_2^2
\end{aligned} \tag{5.4}$$

A second degree polynomial can be expressed as

$$\hat{y}_Q = q_0 + q_1 x_1 + q_2 x_2 + q_3 x_1 x_2 + q_4 x_1^2 + q_5 x_2^2 \tag{5.5}$$

Comparing the expressions of $\hat{y}_D$ and $\hat{y}_Q$, we find that that both expressions have same number of unknown parameters. It can be easily verified that if more design variables were considered in the linear regression model, there would be more unknowns in the expression of $\hat{y}_Q$. So theoretically, the quadratic regression should give better or at least equally good predictions for fitting. The reason is that if we consider a large number of design variables, say 24, we need to consider totally 325 (=($C_{24}^0 + 2C_{24}^1 + C_{24}^2$), $C_{n_V}^j$ is the combination) parameters. Because this number is too high, we use a procedure that gradually increases the number of design variables when creating the second degree polynomials (see Section 2.4.2 for a detailed explanation). In each step, we use an incomplete second degree polynomial obtained from the previous step, and add a new design variable according to the ranking of important design variables. The new second degree polynomial includes the terms left from last regression, the new design variable, the square of the new variable, and interactions between the new variable and the variables of the previous model. We then use stepwise

regression to get a new second degree polynomial that considers more design variables. This method allows us to circumvent the limitation of the software. However, it sacrifices the accuracy of the second degree polynomial because it misses some terms that may be important. If all the combinations were considered in the regression model, a better results would be found (at least for fitting). Because of the software limitation and the method we use to circumvent this limitation, the double regression model gives better predictions than the quadratic regression model under most situations.

5.4.3 Neural Network Translators

Neural Networks were also used to simulate the mapping relations between the design variables and the stiffness and mass of the A-pillar to roof rail joint.

To develop the neural network translator we used the ranking of important design variables obtained using the linear regression models. First, only a few most important design variables were considered in the neural network according to the importance of each design variable. Then we gradually added more design variables to the neural network.

For each neural network, the input design variables were normalized so that they could only vary in the range [-1, +1]. Chapter 4 showed that this approach worked better than either not normalizing the design variables or normalizing the design variables by their mean values.

If we keep training the neural network over a very long period, we can improve the training results dramatically. However, the trained neural network has poor generalization performance, that is, it can not predict accurately the stiffness or mass of new designs that it has not seen in training. This problem is more frequent with neural networks with a large number of unknown parameters relative to the number of examples. We observed this problem when creating the translator A for the B-pillar to rocker joint.

To avoid this problem, cross-validation is used. Specifically, the 600 designs in the database are split into three groups: 300 designs are used for training; 100 designs are used to determine when to stop training; and the remaining 200 designs are used to test the generalization performance of the trained neural network. Cross-validation prevents a neural network from being over trained, improves the predictions of the trained neural network, and reduces the effect of the number of neurons in the hidden layers.

Figures 5.21c and 5.21d show the scatter plots for the training and testing results of the neural network translators.

5.5 Results and Discussion

Figures 5.22a to 5.22d and Tables 5.9 to 5.12 compare the results obtained using different methods. The following are observed regarding the stiffness in each direction and the mass.

5.5.1 I/O Stiffness

Table 5.9 and Figure 5.22a compare the results from the linear polynomial, the second degree polynomials, the double regression model, and the neural networks. It is observed that neural networks give better training/fitting and testing results compared with the second degree polynomials. The testing results from the linear regression model and the second degree polynomial with 20 design variables are almost equally accurate. On the other hand, the double regression model is the best among the polynomial models. The double regression model is less accurate than the neural network in testing /fitting. However, the accuracy of the neural network in testing deteriorates more than the accuracy of the double regression model. Actually, the two tools have almost the same generalization performance. We selected the neural network translator has 22 design variables (Fig. 5.22a). The double regression model with 24 design variables was selected as the polynomial translator.

5.5.2 F/A Stiffness

Figure 5.22b and Table 5.10 compare the results for the F/A. Similar trends as those for the I/O stiffness are observed: the neural network and the double regression are more accurate than either the second degree polynomials or the linear polynomial. The results of the neural network and the double regression model are equally accurate.

For F/A stiffness, the neural network with 24 design variables was chosen because it had the smallest standard deviation for testing. The double regression model, which also has 24 design variables, had the smallest standard deviation for testing among all the polynomials, and was selected as the polynomial translator for the F/A stiffness.

5.5.3 Torsion Stiffness

Results for torsion stiffness are slightly different than those for I/O and F/A stiffness. Figure 5.22c and Table 5.11 show that the double regression model is the most accurate among all models. The neural network and the second degree polynomials are almost equally accurate. The linear regression model is less accurate than the double regression model, but is slightly better than the neural networks or the second degree polynomials. The reason should be that the linear model has more design variables than the neural network and the second degree polynomial.

Based on the above observations, the double regression model with 28 design variables was selected as the polynomial translator. The neural network with 20 design variables had the smallest standard deviation for testing, and was chosen as the neural network translator.

5.5.4 Mass

The predictions of the mass are significantly more accurate compared with those for the I/O, F/A, and torsion stiffnesses (Fig. 5.22d, and Table 5.12). The standard deviations of the predictions from both neural networks and polynomials are less than 2.5 percent. The neural network is more accurate than the second degree polynomial. The double regression model is slightly better than the linear polynomial model. From the comparison, it is found that the double regression model, which has 31 design variables, gives the best prediction compared with the linear regression model or the second degree polynomial model. Therefore, the double regression model was selected as the polynomial translator. The neural network model with 24 design variables has the smallest standard deviation for testing compared with other neural networks. Therefore, it was selected as the neural network translator for the mass.

5.5.5 Validation

We checked the above polynomial regression models using two criteria, namely, the C_P criterion and *AIC* (Akaike's Information Criterion). Figures 5.23a to 5.23d show the results of validating the regression models using C_P criterion. Figures 5.24a-5.24d show the results when *AIC* criterion was used to validate the previous regression models.

C_P Criterion

As Chapter 2 mentioned, the best model has a C_P value almost equal to the number of terms of the polynomial. A value of C_P greater than the number of terms generally indicates that the polynomial does not have enough terms or too many terms. For the I/O stiffness, the C_P criterion is used to check the linear regression model and the second degree polynomial model with 20 design variables. The linear and the second degree polynomial models have 25 and 37 parameters (terms), respectively. Figure 5.23a shows that the C_P value of the regression model is very close to the number of terms in the polynomial, p. Therefore, according to C_P criterion, both regression models are appropriate.

Figure 5.23b shows the value of C_P for the linear polynomial model and the second degree polynomial model with 21 design variables for the F/A stiffness. The two models have 25 and 37 parameters (terms), respectively. It is observed that both models are appropriate according to the C_P criterion.

Figure 5.23c shows the results for the torsion stiffness. It is found that both the linear polynomial model and the second degree polynomial model with 20 design variables are appropriate according to the C_P criterion.

Figure 5.23d shows the results for the mass using C_P criterion. It is observed that the C_P values for both the linear polynomial model and the second degree model with 20 design variables are very close to the number of parameters in the polynomials. If 36 parameters were used in the nonlinear regression model instead of 38, the value of C_P would be closer to p. According to C_P Criterion, and the regression model with 36 parameters probably would give better prediction than the translator with 38 parameters.

AIC Criterion

Figure 5.24a shows the *AIC* values of the linear and second degree polynomial models for the I/O stiffness. It is observed that the second degree polynomial with 20 design variables (37 parameters in the polynomial) has lower *AIC* value. According to *AIC***,** the second degree polynomial model should have better generalization performance than the linear polynomial model. However, Table 5.9 shows that the linear polynomial is slightly more accurate than the second degree polynomial.

Figure 5.24b shows the *AIC* values for the two polynomial models for the F/A stiffness, which have 25 and 37 parameters, respectively. It is observed that the second degree polynomial model has lower *AIC* value compared with the linear regression model. Therefore, the second degree polynomial should have better generalization

performance than the linear polynomial. Table 5.10 shows that the two models are almost equally accurate.

Figure 5.24c shows the results for the torsion stiffness of A-pillar to roof rail joint. It is observed that the linear regression model and the second degree polynomial model with 20 design variables have almost the equal AIC value. Therefore, the two models should have same accurate generalization performance. Table 5.11 shows that the second degree polynomial is more accurate than the linear polynomial, but the difference of the two models is very small.

Figure 5.23d shows the results for the mass of the A-pillar to roof rail joint. The linear polynomial model and the second polynomial model with 20 design variables have almost same *AIC* value. Although the second degree polynomial is slightly more accurate, both regression models will have almost same accurate generalization performance according to *AIC* criterion. The same conclusion can be drawn from Table 5.12.

Discussion About the *AIC* Criterion

It is found that the conclusions obtained using *AIC* criterion generally agree well with those obtained by using standard deviation of the bias corresponding to the testing set of the database. However, there are cases where the *AIC* criterion gives different conclusion from that obtained using standard deviation. For example, the fitting results of the linear regression model for I/O stiffness are worse than the second degree polynomial model. However, its testing results are better than the second degree polynomial model. The second degree polynomial model has smaller *AIC* value comparing with the linear model. According to the *AIC* criterion, the second degree polynomial model should give better prediction. The following are possible reasons for the conflicts. First, the linear and second degree models have almost the same accuracy (Tables 5.10-5.12) so even small numerical errors or statistical errors can affect the conclusions. Second, when we derive

the formulation of *AIC*, some assumptions are made, such as the distribution of each design variable, which may not be accurate.

5.5.6 Conclusions

In this chapter, translators A for the A-pillar to roof rail joint were developed. Several regression models were studied and compared. C_P and *AIC* criteria were used to validate obtained results. From the comparison, it was found that the double regression models were more accurate than the other polynomial models. Neural networks were also used to simulate the mapping between the design variables and the stiffness and mass of the joint. The double regression model and the neural network model seemed to have the same generalization performance.

Table 5.1: Effects of Mesh Size on FEA Results

Seq. No	No. of Nodes	No. of Elements	I/O ($\times 10^7$ Nmm)	F/A ($\times 10^7$ Nmm)	Torsion ($\times 10^7$ Nmm)	Mass (kg)
2	2098	2079	6.2716	7.1630	1.1847	2.0302
3	3045	3017	6.2592	7.0685	1.1851	2.0304
4	3635	3537	6.1042	7.0145	1.1673	2.0297

Table 5.2: Comparison of FEA Results and Experimental Results

Cars	Stiffness/Mass	Experimental (Nmm, kg)	FEA (Nmm, kg)	error
Car A	I/O ($\times 10^7$)	6.531	6.2716	-4.0 %
	F/A($\times 10^7$)	4.929	7.1630	45.3%
	Torsion ($\times 10^7$)	1.259	1.1847	-5.9%
	Mass		2.0302	
Car B	I/O ($\times 10^7$)	3.399	3.6003	5.9%
	F/A ($\times 10^7$)	5.981	5.0370	-15.8%
	Torsion ($\times 10^7$)	1.044	0.8449	-19.1%
	Mass		2.4396	

Table 5.3: Measured Dimensions for A-pillar to Roof Rail Joints

Cars and Dimensions	Car 1	Car 2	Car 3	Car 4	Car 5	Car 6	Min	Max
Part1_part2_assmble	Yes	Yes	Yes	Yes	Yes	Yes		
Part2_part3_assmble	Yes	Yes	Yes	Yes	Yes	Yes		
Part3_part4_assmble	Yes	Yes	Yes	Yes	Yes	Yes		
Part5_part1_assmble	No	No	No	Yes	No	No		
Part6_part4_assmble	No	No	Yes	No	No	No		
Part7_part2_assmble	Yes	Yes	Yes	No	Yes	No		
Distance_from_edge	10	8	10	8	7	8	7	10
Header_offset	202	255	176	238	190	285	176	285
Roof_rail_offset	165	196	105	158	137	147	105	196
A_pillar_offset	163	179	225	207	203	240	163	240
Theta	23	25	20	20	30	21	20	30
Phi	76	76	77	81	71	76	71	81
Header_horizontal_offset	15	37	75	78	72	53	15	78
Same_flange_width	NO	NO	NO	NO	NO	NO		
Flange1_width	20	24	18	20	24	19	18	24
Flange2_width	15	16	16	14	16	16	14	16
Flange3_width	16	17	19	12	14	16	12	19
Spot_weld_spacing	35	45	35	50	33	40	33	50
Flange1_angle_up	90	103	90	90	90	113	90	113
Flange1_angle_down	90	90	102	110	100	104	90	110
Flange2_angle_up	90	130	110	90	83	110	90	130
Flange2_angle_down	115	112	96	122	123		96	123
Flange3_angle_down	100	90	87	90	98	107	87	107
Door_allowance	38	62	27	63	54	63	27	63
Angleback	30	23	46	14	20	8	8	46
H_width	80	90	78	108	109	96	78	109
H_height	25	19	30	34	28	28	19	34
H_window_depth	16	15	13	15	17	12	12	17
H_blending_rad	95	140	111	146	159	210	96	210
Rr_width	65	73	56	83	84	146	56	146
Rr_height	55	75	47	77	63	79	47	79
Rr_h1	40	52	39	60	60	108	39	108
Rr_h2	35	54	51	53	60	107	35	107
Rr_angle_bottom	35	30	24	42	35	20	20	42
Rr_blending_rad	90	88	60	53	108	105	53	108
Ap_height	60	75	52	78	60	66	52	78
Ap_inboard_width	26	31	20	23	23	33	20	33
Ap_inboard_depth	25	27	36	29	36	36	25	36
Ap_door_ws_distance	35	19	24	34	23	21	19	35
Ap_window_depth	22	26		20	19	17	17	26
Ap_flange1_angle	0	10	0		7	6	0	10
Ap_reinf_depth	9	9					9	9
Ap_reinf_flange1_angle	155	140					140	155
Ap_reinf_inner_door_allowance	38	29	22				22	38
Ap_reinf_flange2_angle	91	130	116				91	130
Ap_blending_rad	55	100	105	120	113	138	55	138
Part2_cut_distance	90	55	85	96	110		55	110

Part3_cut_distance	50	45	24	36	56		24	56
Part4_extension	90	50	46		68		46	90
Part7_cut_distance	90	90					90	90
Thickness of Part1	0.94	0.89	1.19	0.94	0.89	0.81	0.81	1.19
Thickness of Part2	1.02	0.89	0.89	1.14	1.02	1.14	0.89	1.14
Thickness of Part3	1.65	1.12	1.68	1.91	0.89	1.40	0.89	1.68
Thickness of Part4	1.02	0.89	0.71		1.02	1.07	0.89	1.07
Thickness of Part7	1.24	1.83	1.30	1.17			1.17	1.83

Table 5.4: Ranges of Design Variables for A-pillar to Roof Rail Joint

No	Cars and Dimensions	Bounds (mm, degree)	
		Lower	Upper
1	Distance_from_edge #	8	8
2	Header_offset	160	314
3	Roof_rail_offset	95	216
4	A_pillar_offset	148	264
5	Theta	18	33
6	Phi	65	89
7	Header_horizontal_offset	14	86
8	Flange1_width #	20	20
9	Flange2_width #	16	16
10	Flange3_width #	16	16
11	Spot_weld_spacing #	50	50
12	Flange1_angle_up	90	124
13	Flange1_angle_down	90	121
14	Flange2_angle_up	90	143
15	Flange2_angle_down	90	135
16	Flange3_angle_down	90	118
17	Door_allowance	25	69
18	Angleback	7	51
19	H_width	71	120
20	H_height	17	37
21	H_window_depth	11	19
22	H_blending_rad	87	231
23	Rr_width	51	161
24	Rr_height	43	87
25	Rr_h1	35	119
26	Rr_h2	32	118
27	Rr_angle_bottom	18	46
28	Rr_blending_rad	48	119
29	Ap_height	47	86
30	Ap_inboard_width	18	36
31	Ap_inboard_depth	23	40
32	Ap_door_ws_distance	17	39
33	Ap_window_depth	15	29
34	Ap_flange1_angle	0	11
35	Ap_reinf_depth	1	29
36	Ap_reinf_flange1_angle	91	171
37	Ap_reinf_inner_door_allowance	20	69
38	Ap_reinf_flange2_angle	91	144

39	Ap_blending_rad	50	151
40	Part2_cut_distance	50	121
41	Part3_cut_distance	22	62
42	Part4_extension	42	99
43	Part7_cut_distance	40	121
44	Thickness of Part1	0.74	1.31
45	Thickness of Part2	0.81	1.25
46	Thickness of Part3	0.81	1.85
47	Thickness of Part4	0.81	1.18
48	Thickness of Part7	1.06	2.01

Fixed

Table 5.5: Ranking of Important Design Variables for the I/O Stiffness of A-pillar to Roof Rail Joint

Rank	Design Variable
1	A_pillar_offset
2	AP_blending_rad
3	Thickness of part3
4	AP_inboard_width
5	Thickness of part7
6	AP_flange1_angle
7	AP_door_ws_distance
8	Phi
9	Flange1_angle_up
10	Flange1_angle_down
11	AP_reinf_depth
12	Theta
13	Thickness of part2
14	AP_reinf_flange2_angle
15	Angleback
16	Header_horizontal_offset
17	AP_window_depth
18	Roof_rail_offset
19	Flange2_angle_up
20	RR_height

Table 5.6: Ranking of Important Design Variables for the F/A Stiffness of A-pillar to Roof Rail Joint

Rank	Design Variable
1	Thickness of part3
2	A_pillar_offset
3	RR_height
4	AP_height
5	Theta
6	Roof_rail_offset
7	Thickness of part2
8	AP_door_ws_distance
9	Thickness of part7
10	Flange1_angle_up
11	Flange1_angle_down
12	Flange2_fangle_up
13	AP_reinf_inner_door_allowance
14	AP_reinf_depth
15	Door_allowance
16	Phi
17	AP_inboard_width
18	RR_width
19	AP_inboard_depth
20	AP_window_depth

Table 5.7: Ranking of Important Design Variables for the Torsion Stiffness of A-pillar to Roof Rail Joint

Rank	Design Variable
1	A_pillar_offset
2	Thickness of part3
3	Flange1_angle_down
4	AP_inboard_width
5	AP_blending_rad
6	AP_door_ws_distance
7	AP_window_depth
8	AP_inboard_depth
9	Roof_rail_offset
10	Phi
11	AP_flange1_angle
12	Thickness of part2
13	Door_allowance
14	Thickness of part7
15	RR_width
16	RR_h1
17	AP_reinf_depth
18	Theta
19	Header_horizontal_offset
20	RR_height

Table 5.8: Ranking of Important Design Variables for the Mass of A-pillar to Roof Rail Joint

Rank	Design Variable
1	Roof_rail_offset
2	Thickness of part3
3	Header_offset
4	Thickness of part7
5	A_pillar_offset
6	Thickness of part1
7	RR_width
8	Thickness of part2
9	Door_allowance
10	H_width
11	AP_blending_rad
12	Part7_cut_distance
13	Angleback
14	Header_horizontal_offset
15	Part4_extension
16	AP_door_ws_distance
17	H_blending_rad
18	Part2_cut_distance
19	AP_height
20	AP_reinf_depth

Table 5.9: Comparison of Results from Different Methods For the I/O Stiffness of A-pillar to Roof Rail Joint

Method	Std of Fitting/Training	R^2	Std of Testing
Linear	0.1583	83.5%	0.1625
2nd Degree Polynomial	0.1366	87.7%	0.1717
Double Regression (Cubic)	0.1231	88.1%	0.1210
Neural Networks	0.1087		0.1208

Table 5.10: Comparison of Results from Different Methods For the F/A Stiffness of A-pillar to Roof Rail Joint

Method	Std of Fitting/Training	R^2	Std of Testing
Linear	0.1233	88.2%	0.1320
2nd Degree Polynomial	0.1056	90.5%	0.1312
Double Regression (Cubic)	0.0960	91.5%	0.1061
Neural Networks	0.0920		0.1011

Table 5.11: Comparison of Results from Different Methods For the Torsion Stiffness of A-pillar to Roof Rail Joint

Method	Std of Fitting/Training	R^2	Std of Testing
Linear	0.1391	82.6%	0.1488
2nd Degree Polynomial	0.1380	82.4%	0.1568
Double Regression (Quadratic)	0.1229	84.7%	0.1362
Neural Networks	0.1332		0.1517

Table 5.12: Comparison of Results from Different Methods For the Mass of A-pillar to Roof Rail Joint

Method	Std of Fitting/Training	R^2	Std of Testing
Linear	0.0215	97.9%	0.0228
2nd Degree Polynomial	0.0197	98.2%	0.0242
Double Regression (Quadratic)	0.0185	98.3%	0.0211
Neural Networks	0.0155		0.0219

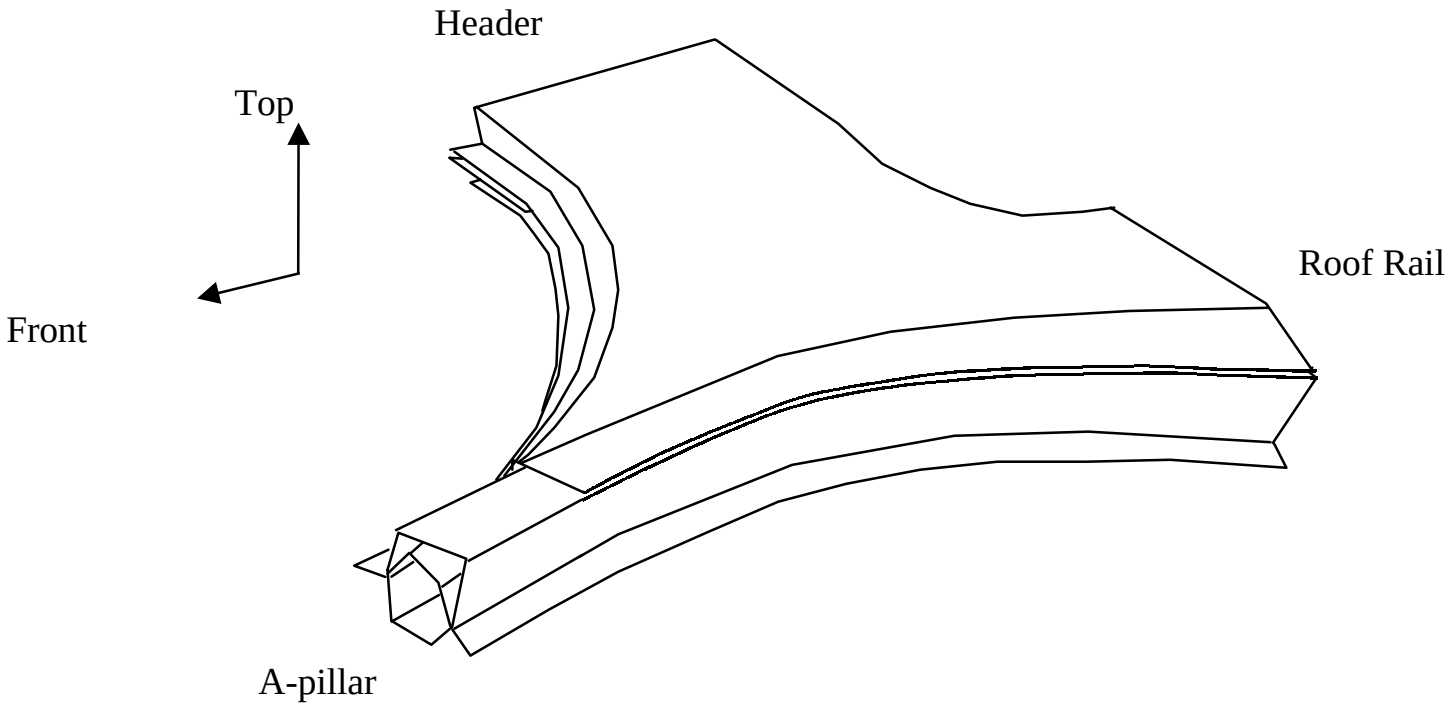


Figure 5.1a: A-pillar to Roof Rail Joint

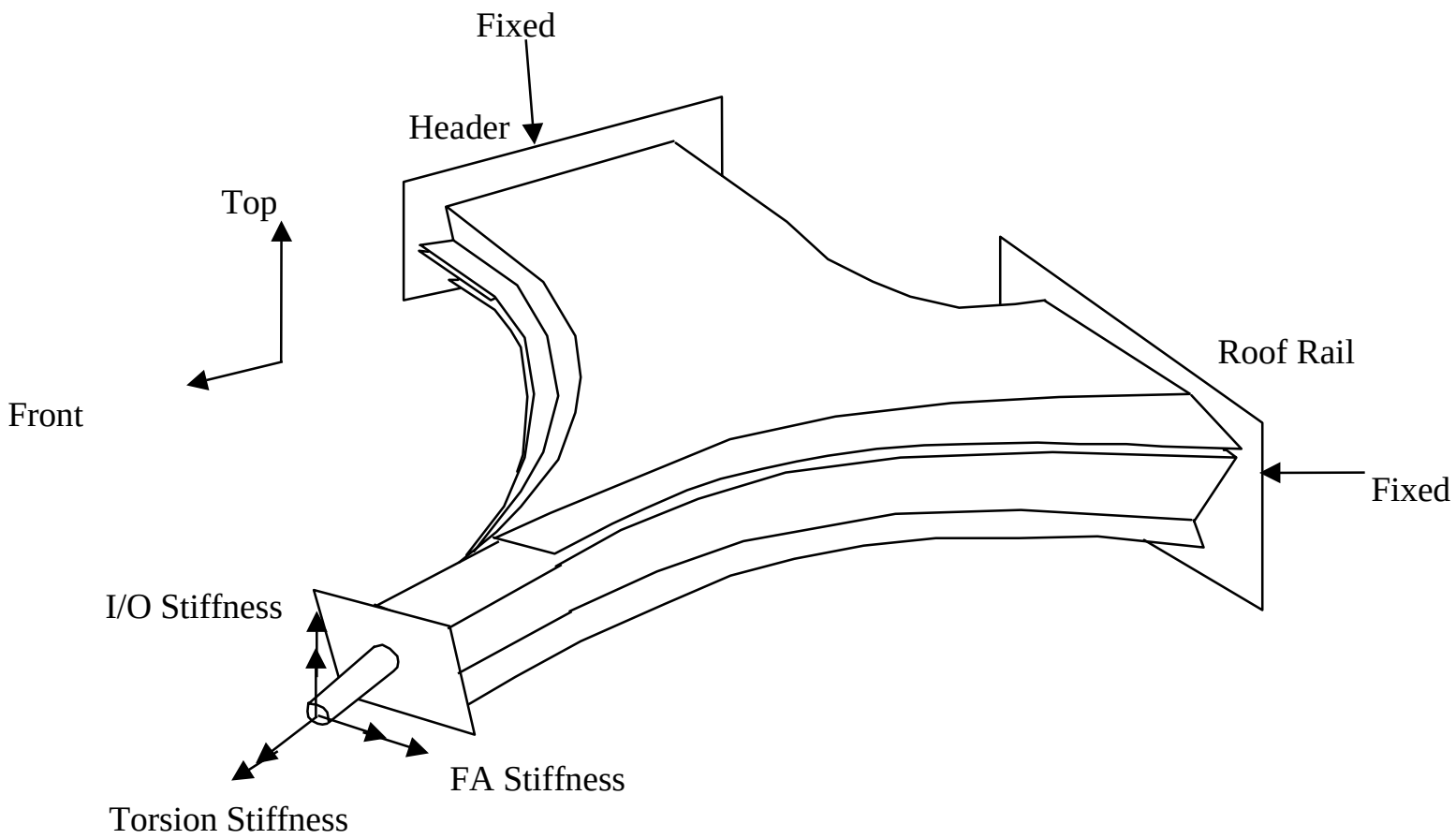


Figure 5.1b: Definition of Stiffness for A-pillar to Roof Rail Joint

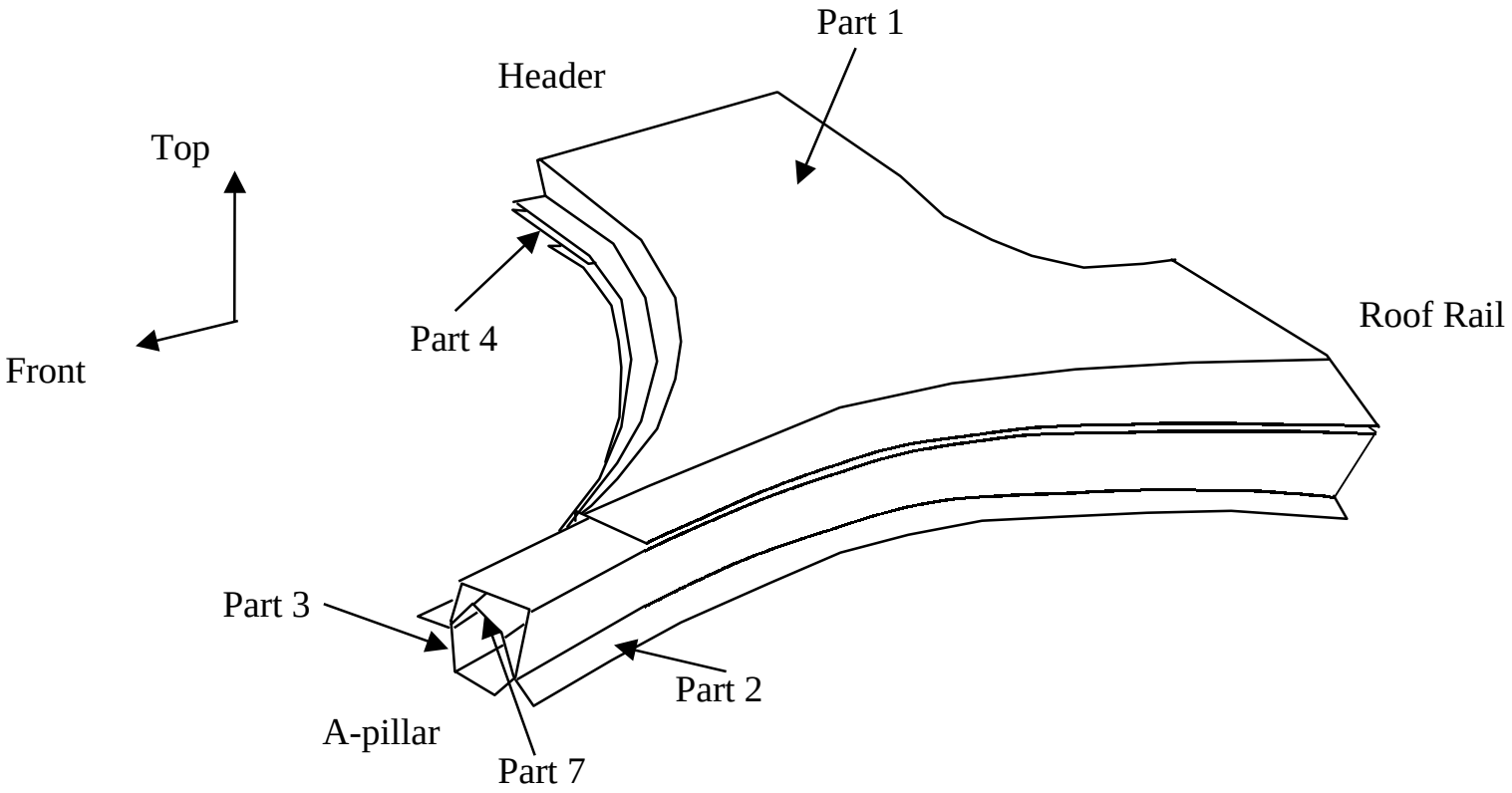


Figure 5.2: Parts of A-pillar to Roof Rail Joint

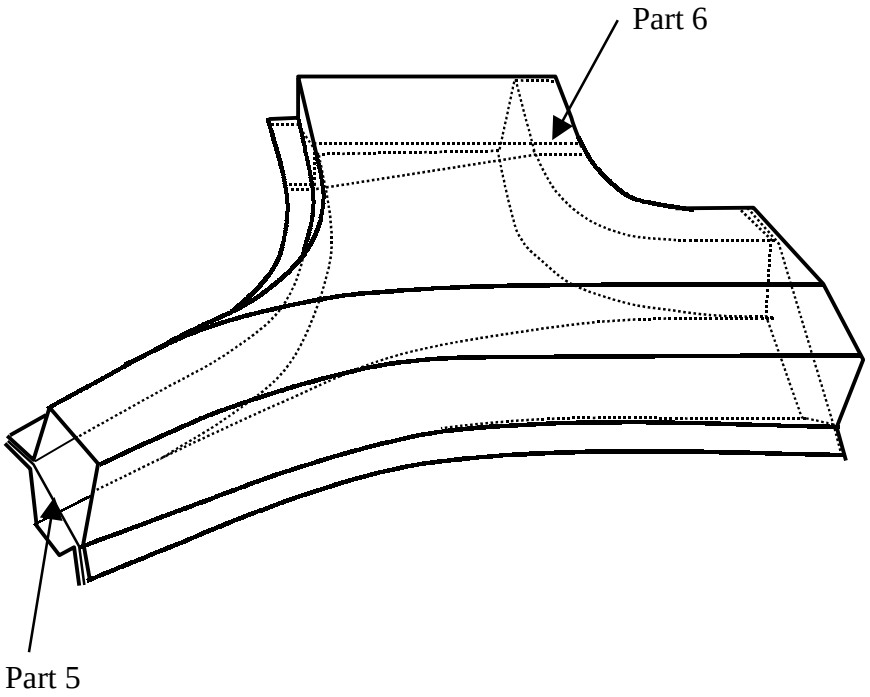


Figure 5.3: Parts of A-pillar to Roof Rail Joint

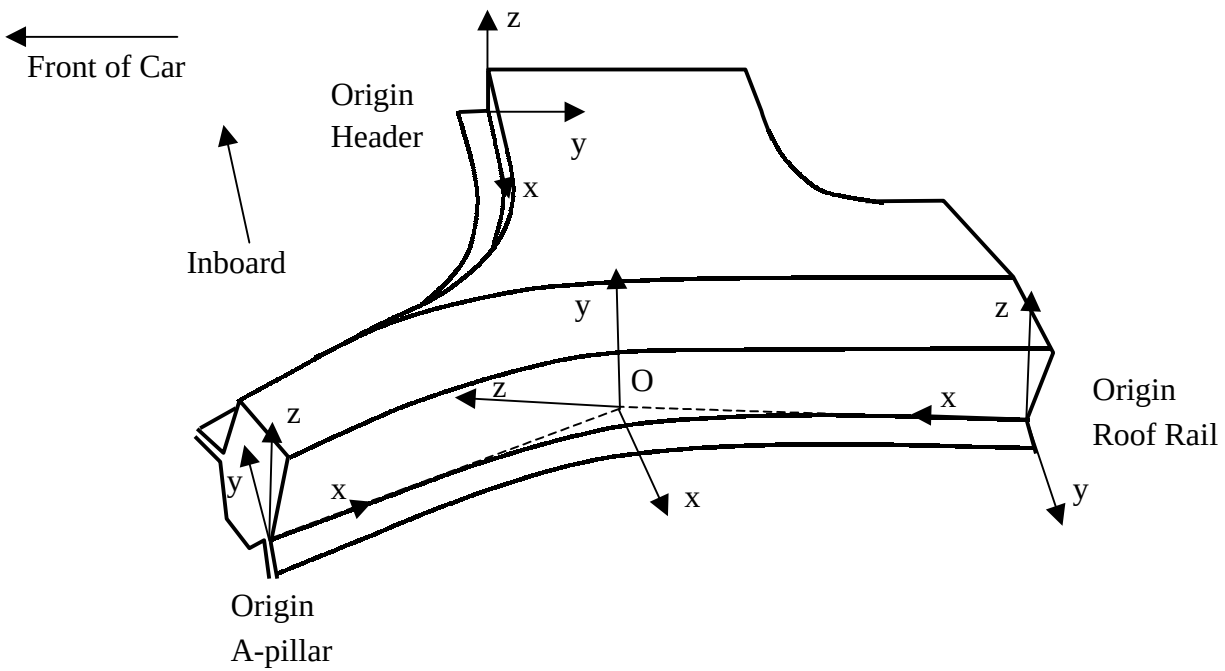


Figure 5.4: Global and Local Coordinate Systems

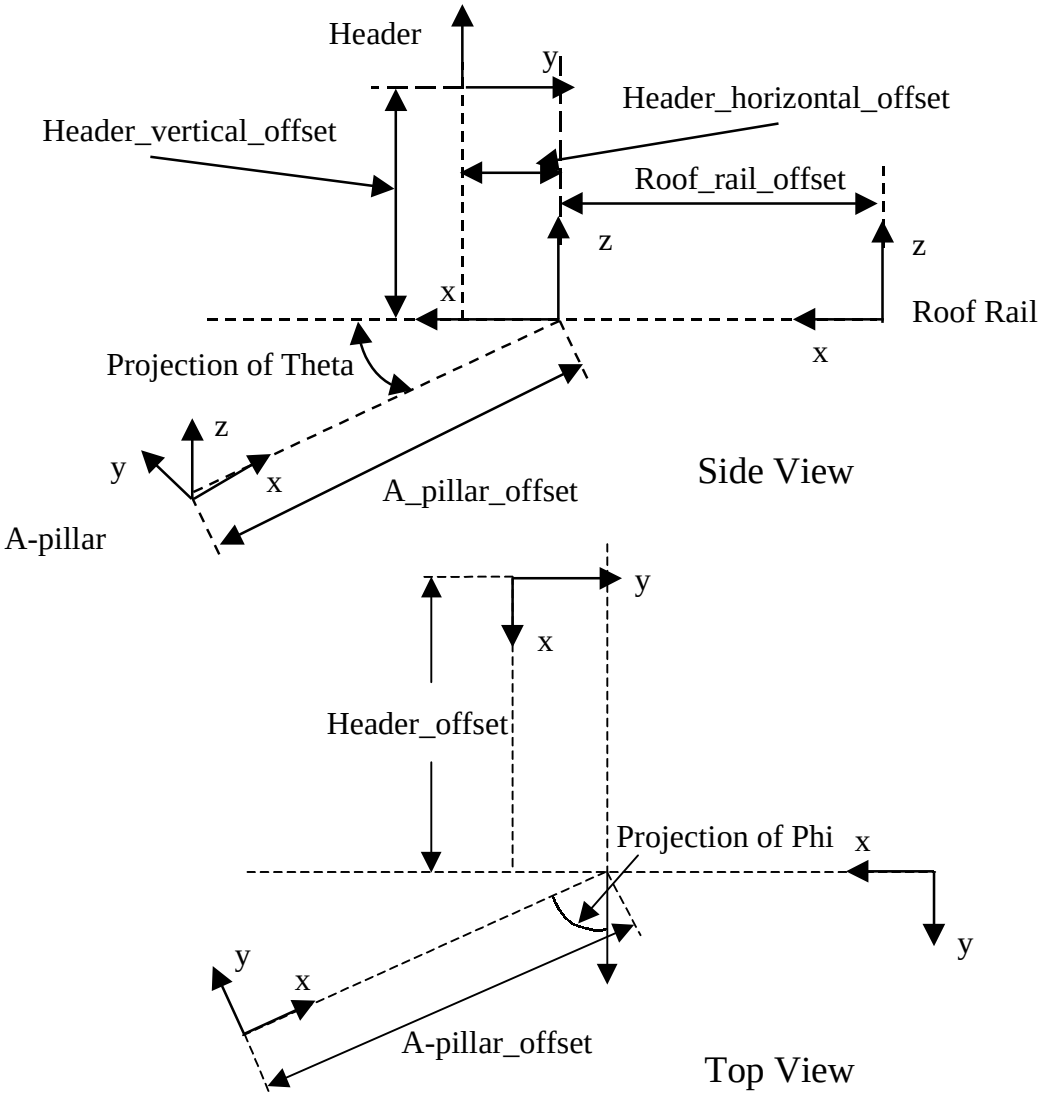


Figure 5.5: Orientation and Position of Each Branch

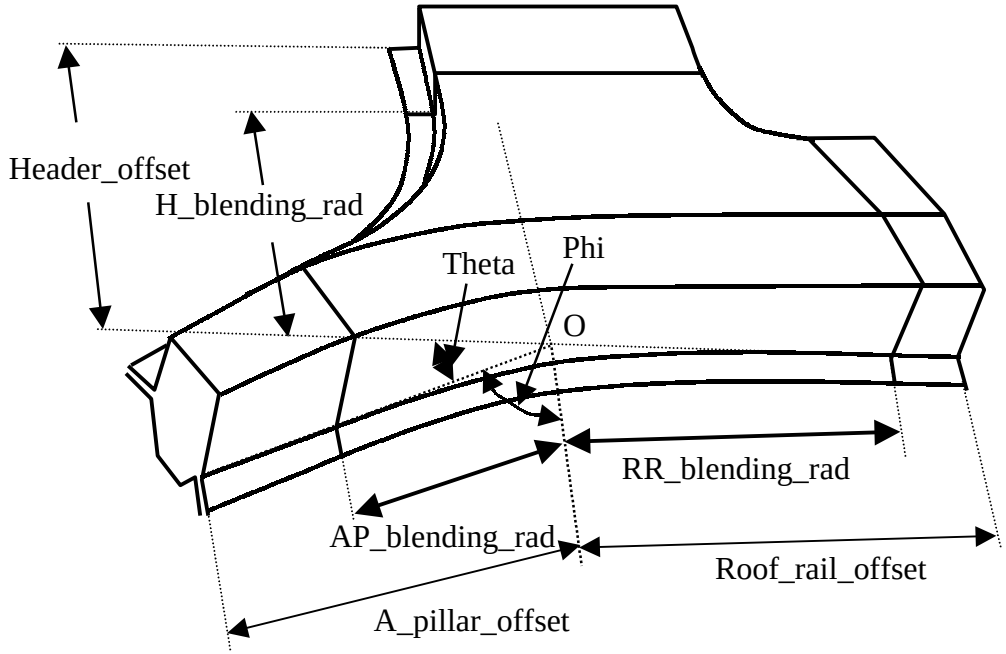


Figure 5.6: A-pillar to Roof Rail Joint Parameters

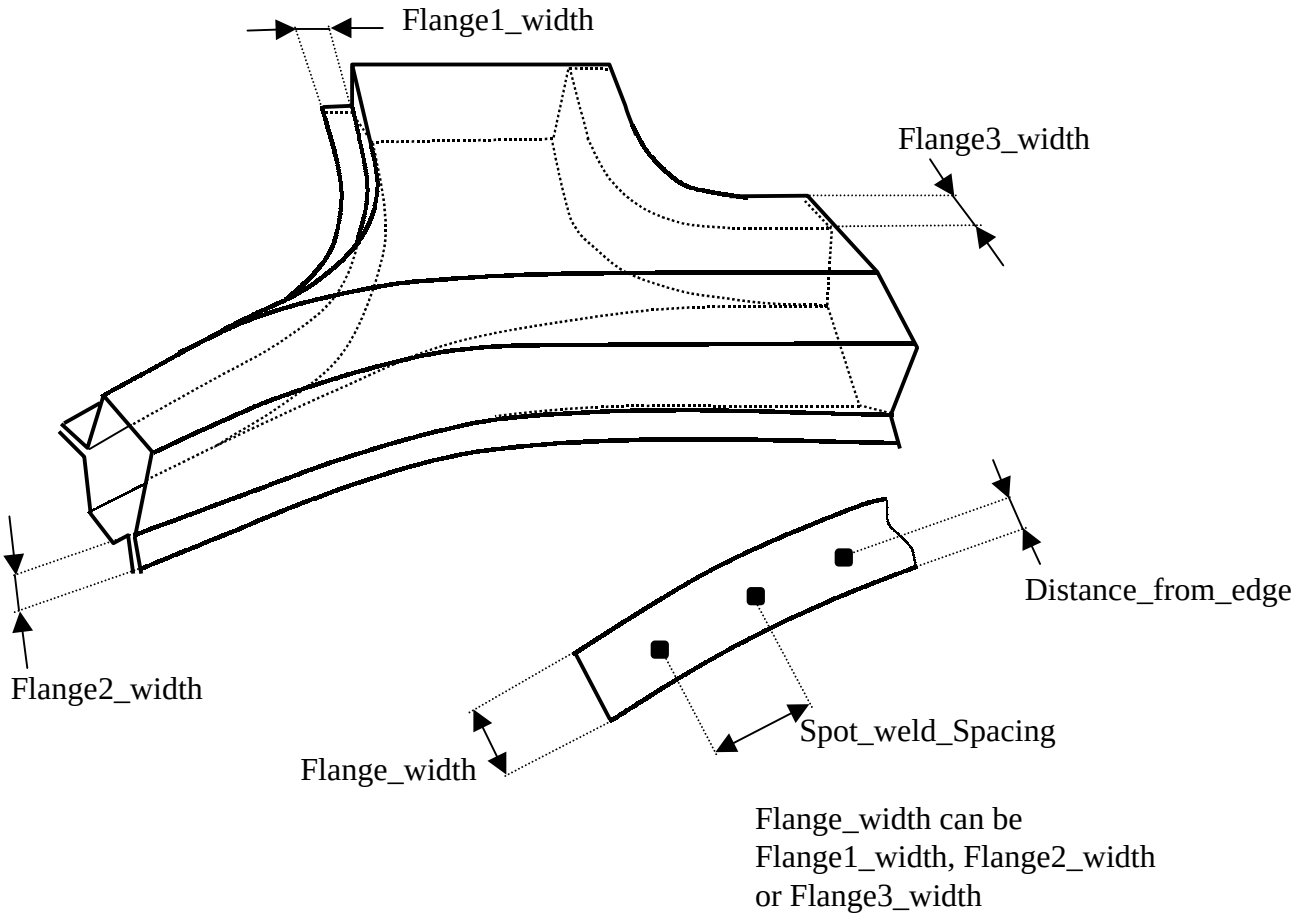


Figure 5.7: Flange and Welding Dimensions

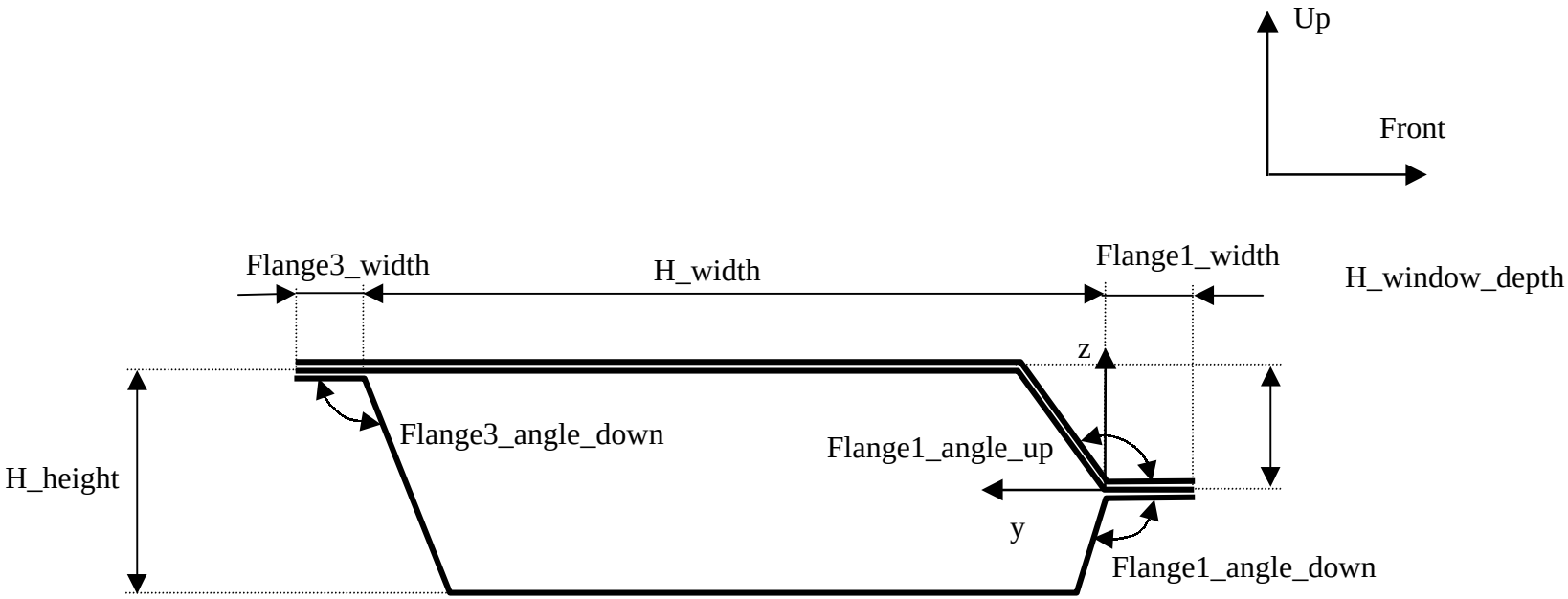


Figure 5.8: Physical Parameters of Header

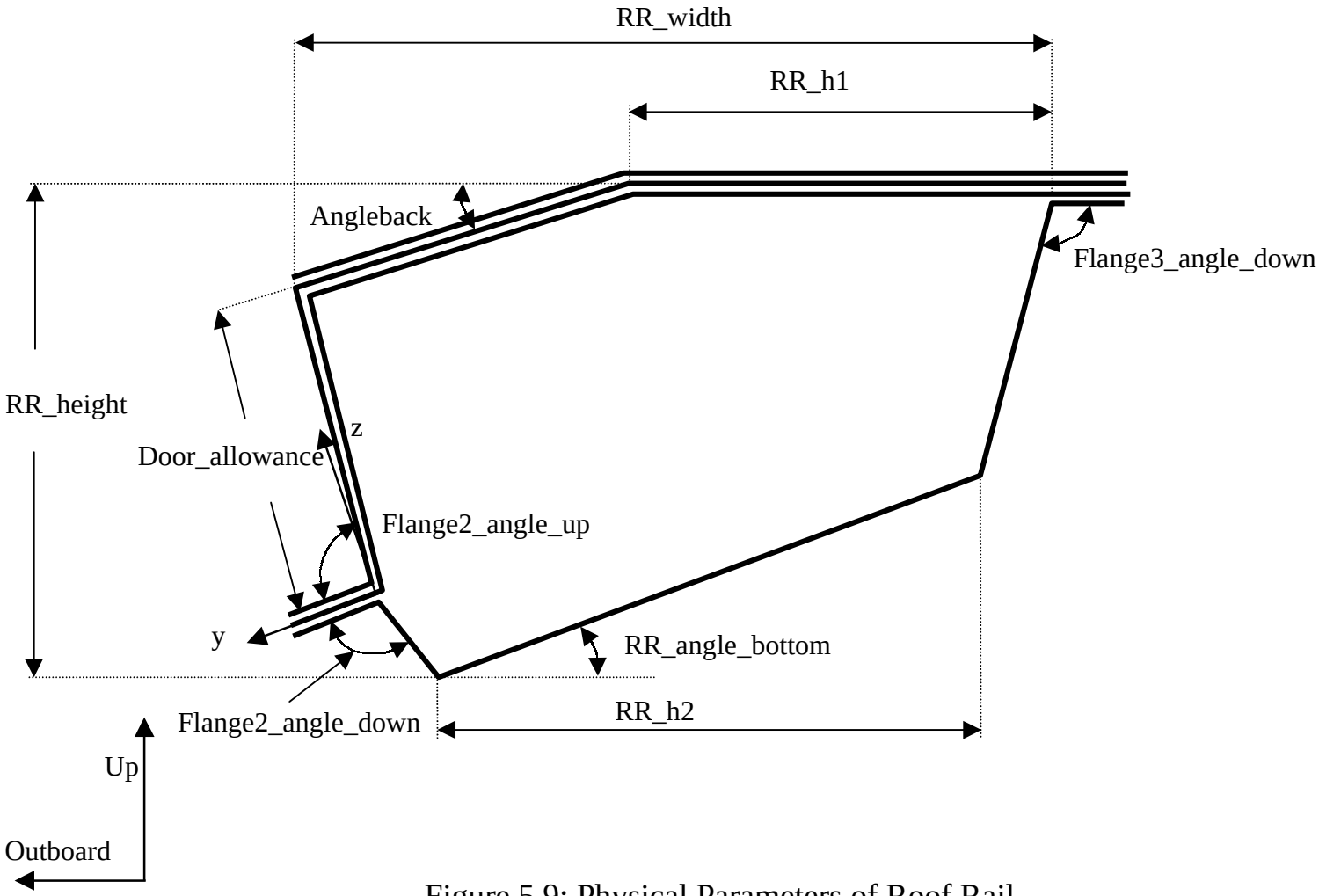


Figure 5.9: Physical Parameters of Roof Rail

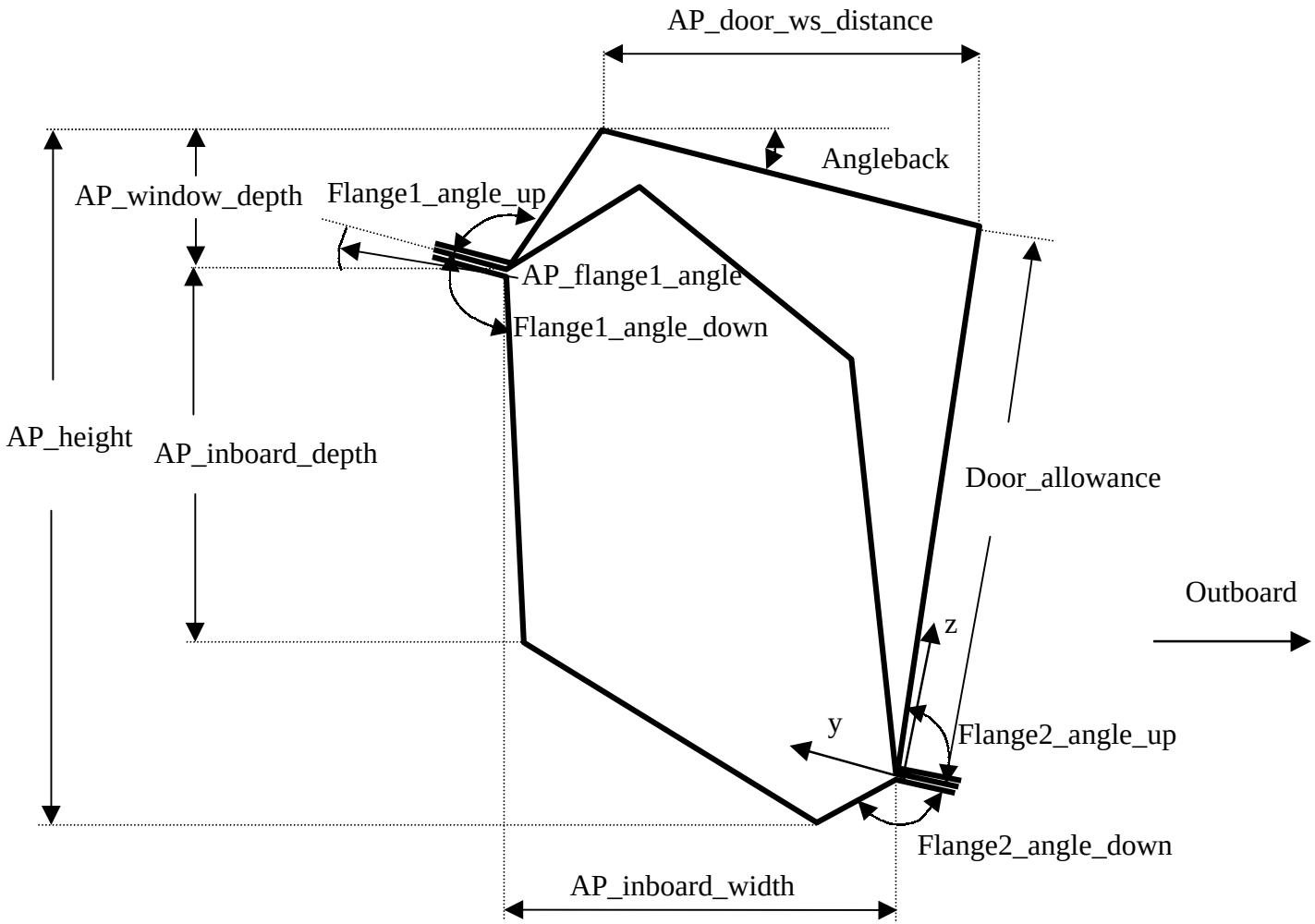


Figure 5.10: Physical Parameters of A-pillar

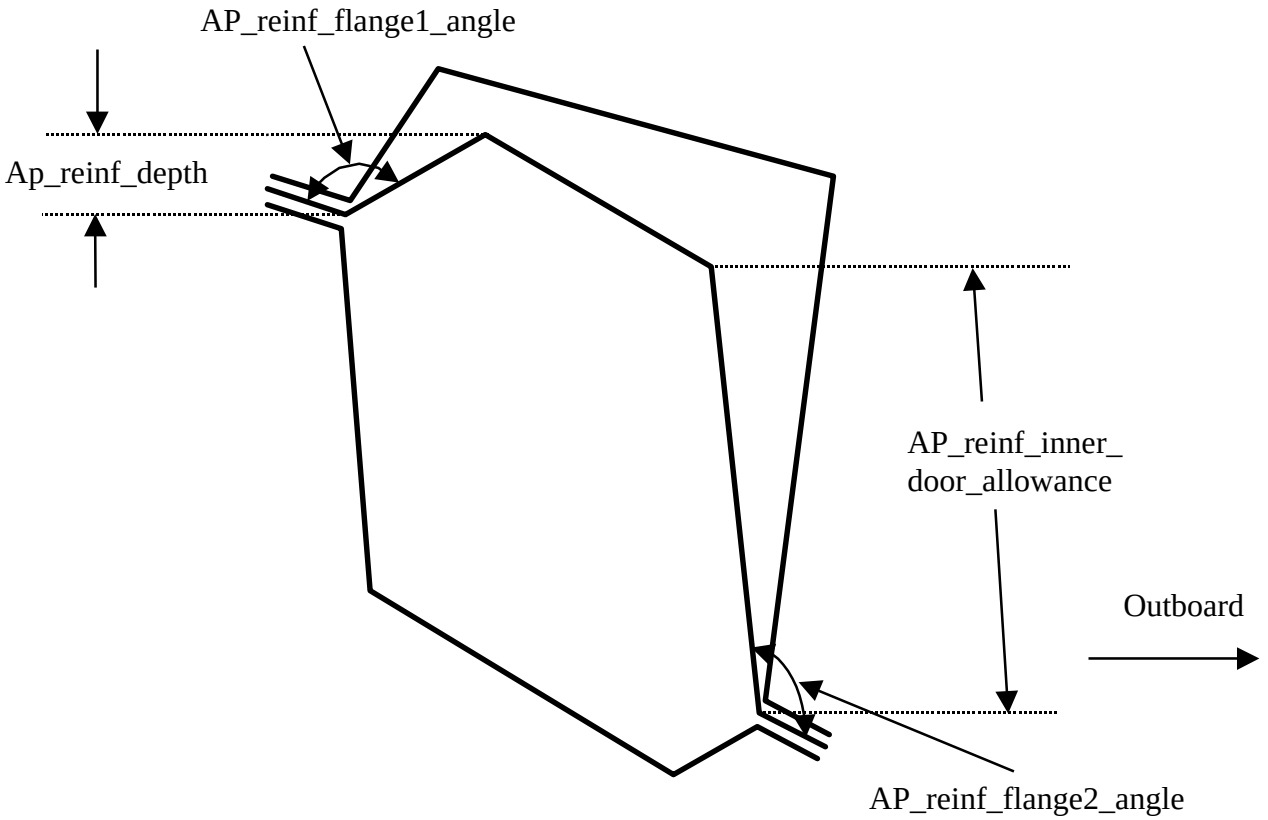


Figure 5.11: Physical Parameters for A-pillar Reinforcement

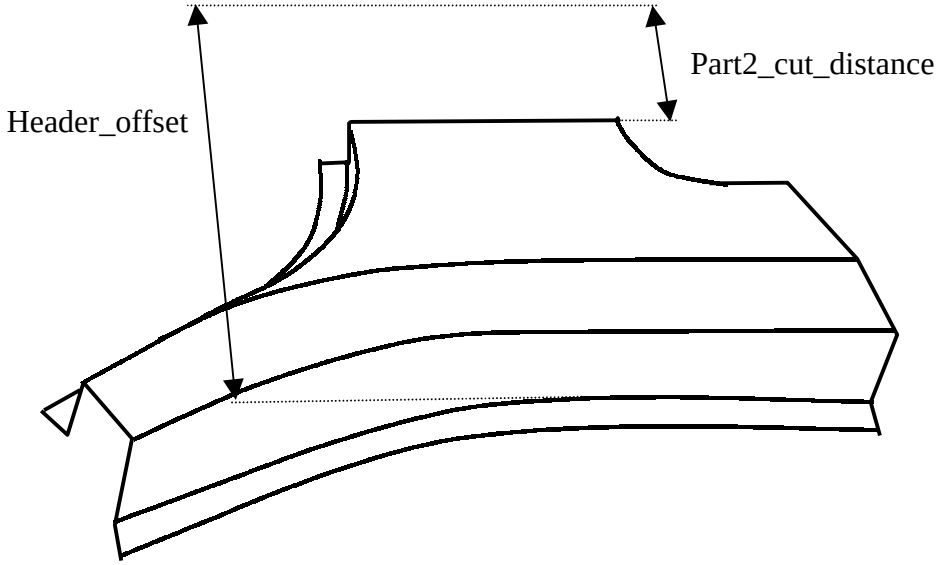


Figure 5.12: Physical Parameters for Part 2

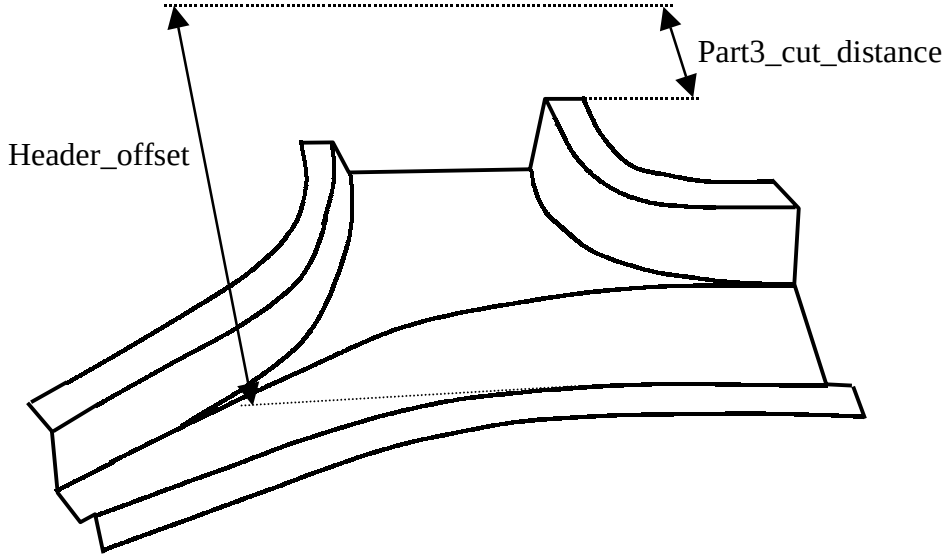


Figure 5.13: Physical Parameters for Part 3

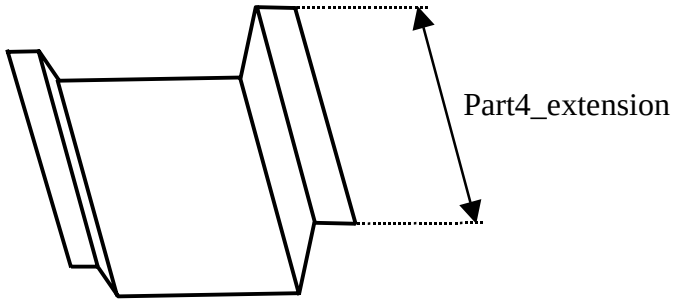


Figure 5.14: Physical Parameter for Part 4

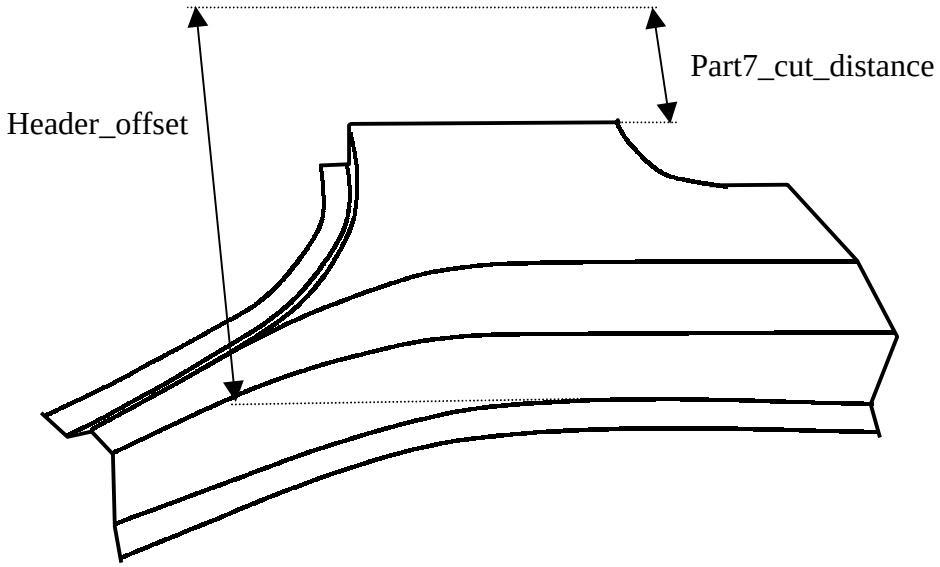


Figure 5.15: Physical Parameters for Part 7

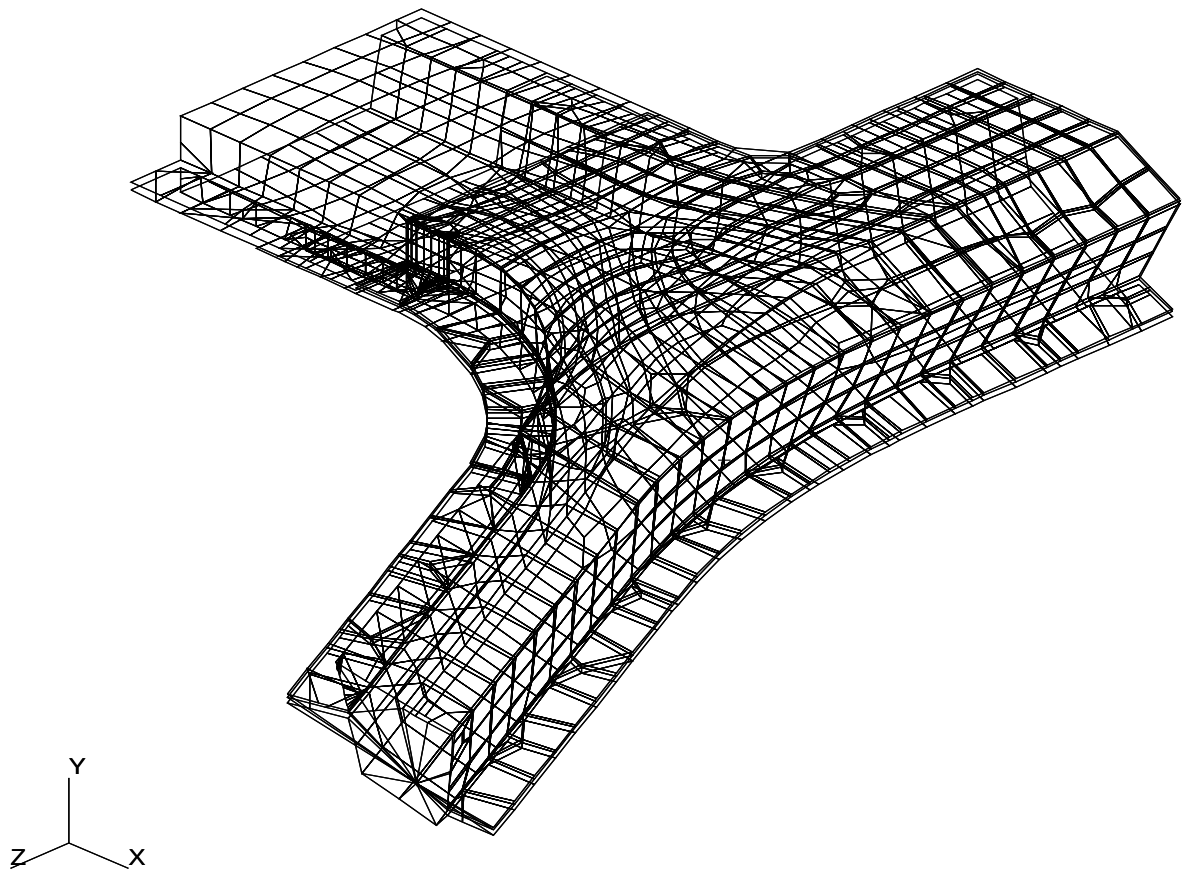


Figure 5.16: FEA Model for A-pillar to Roof Rail Joint

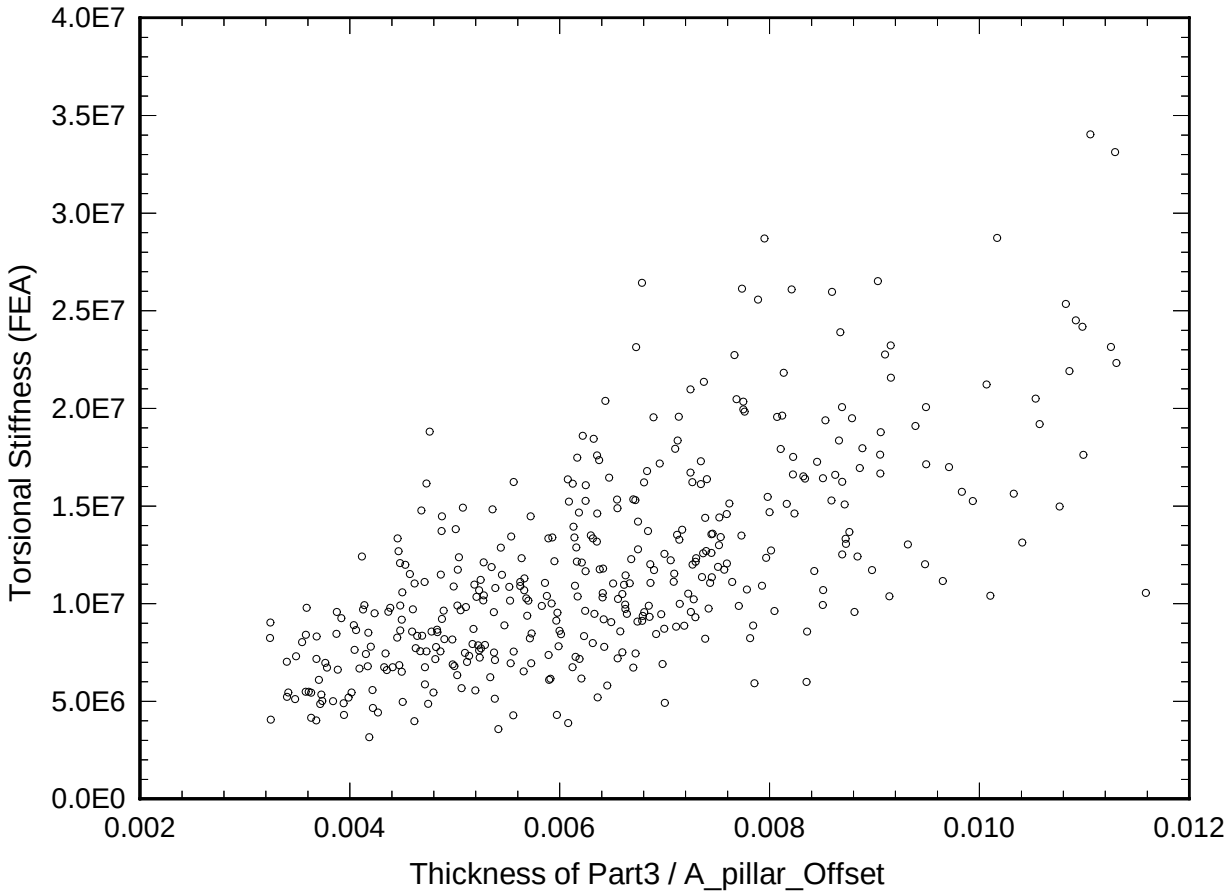


Fig. 5.17: Correlation Between Torsion Stiffness and the Value of (Thickness of Part3/A_pillar_Offset) for A-pillar to Roof Rail Joint

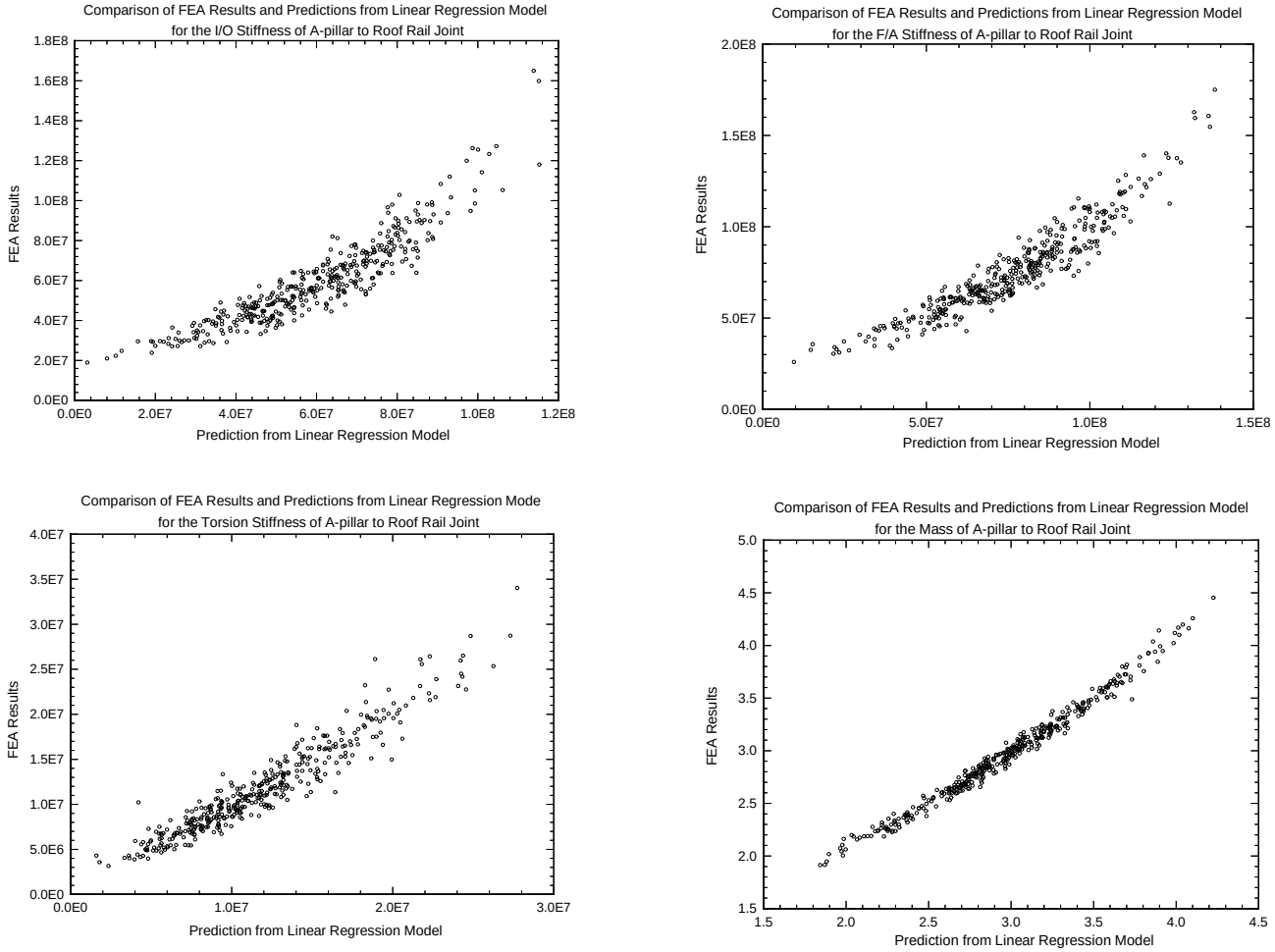


Figure 5.18:. Comparison of FEA Results and Predictions from Linear Regression Model

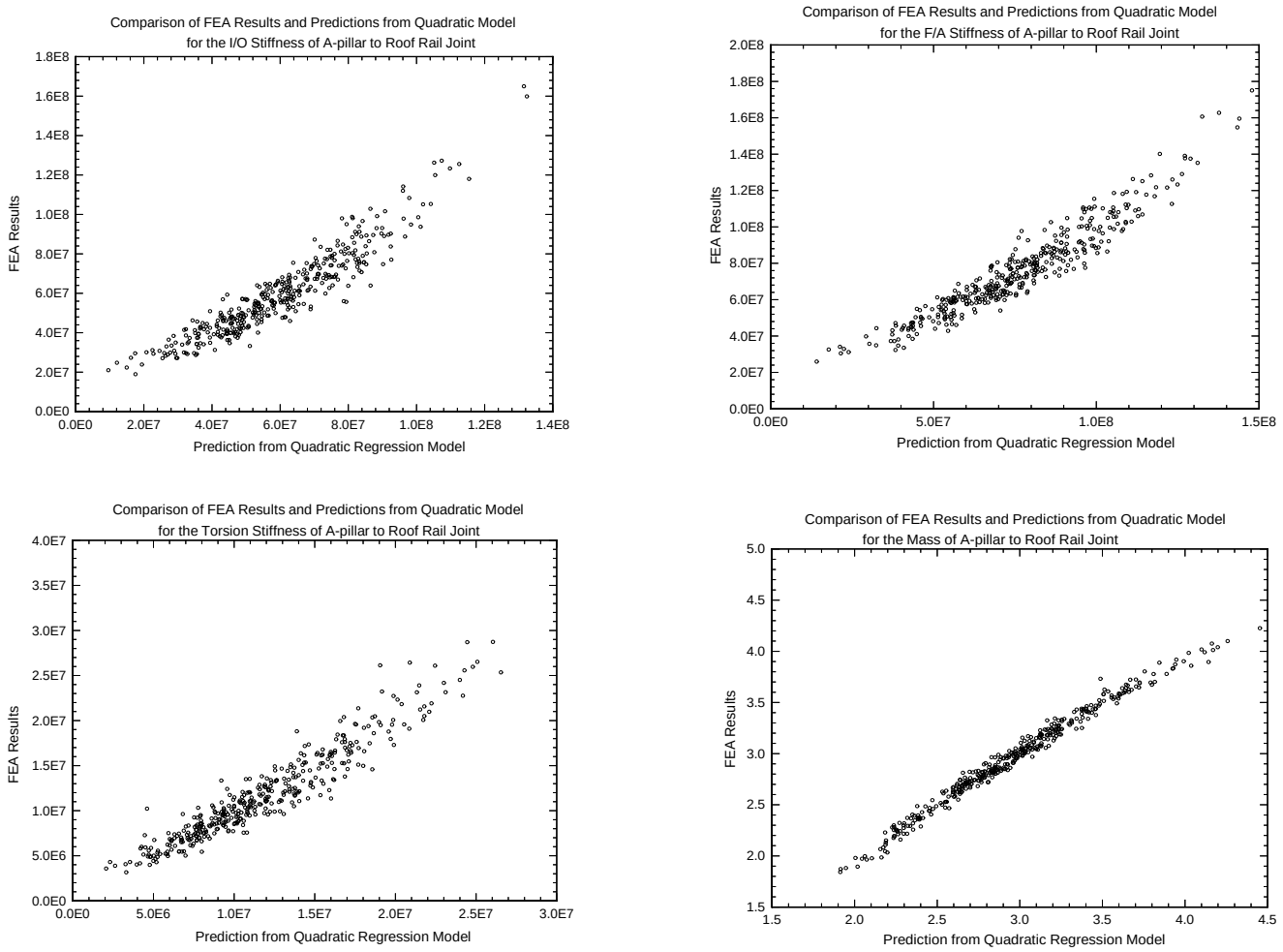


Figure 5.19: Comparison of FEA Results and Predictions from Second Degree Polynomial Model

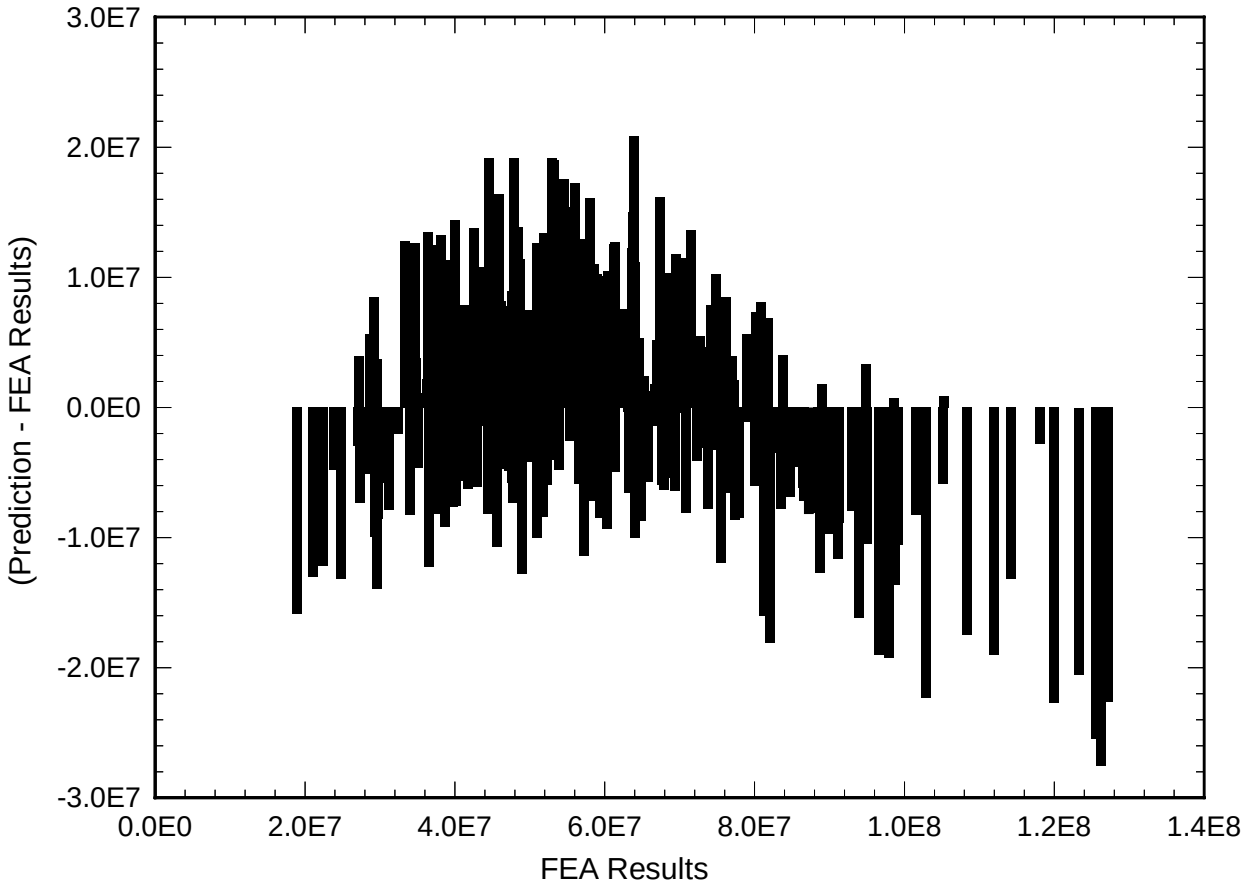


Fig 5.20a: Relation Between FEA Results and Error of Prediction for the Linear Polynomial Model of I/O Stiffness

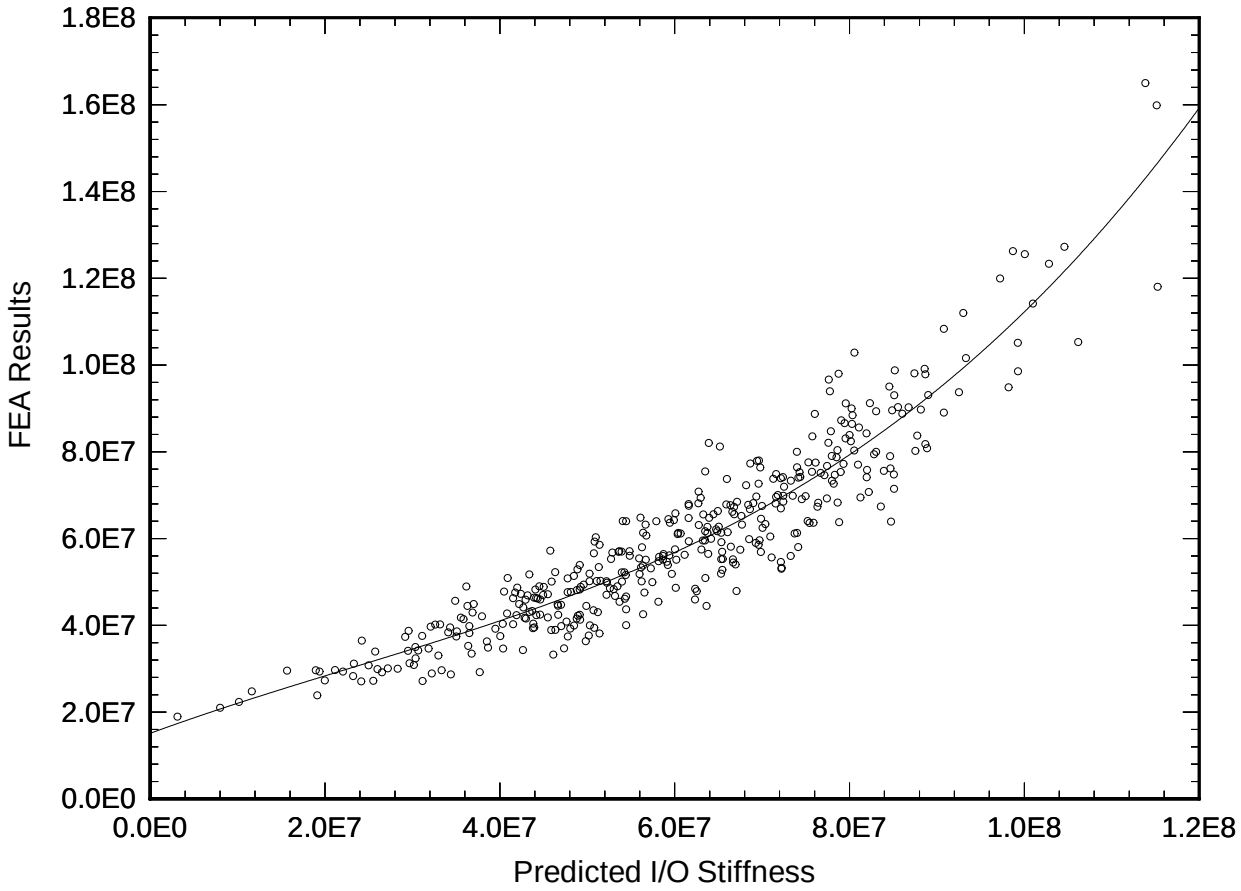
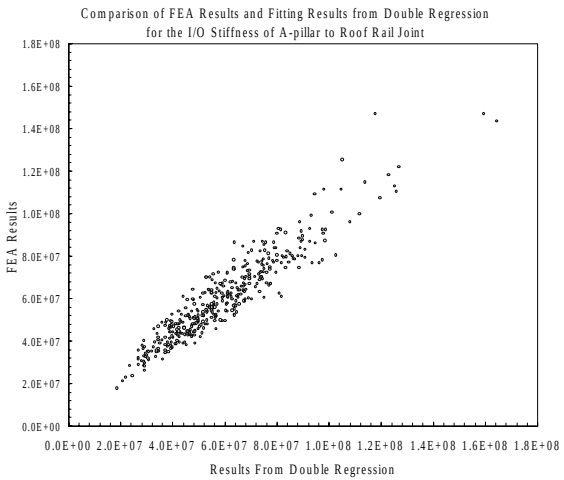
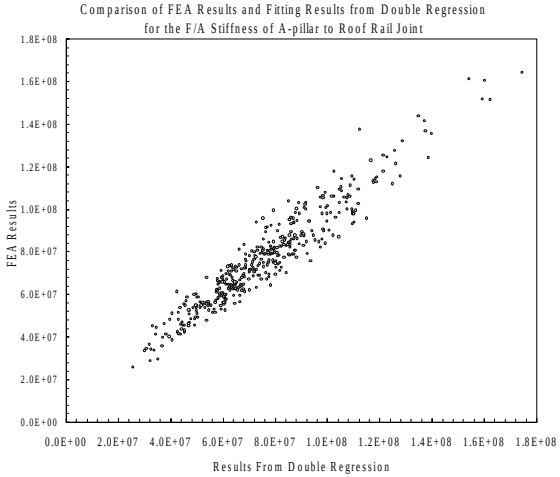
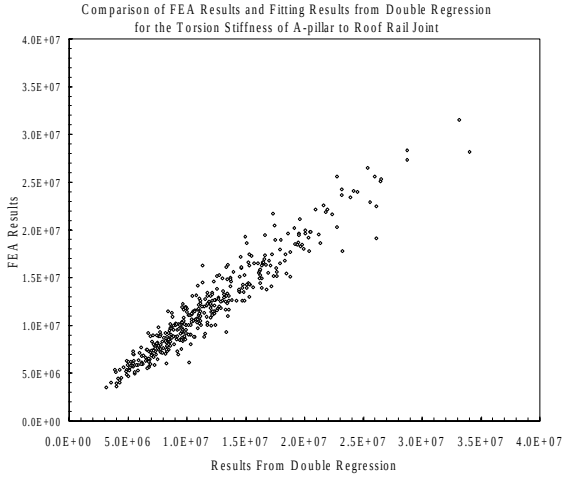


Figure 5.20b: Comparison of FEA Results and Prediction
from Linear Polynomial Model
and Explanation of Double Regression







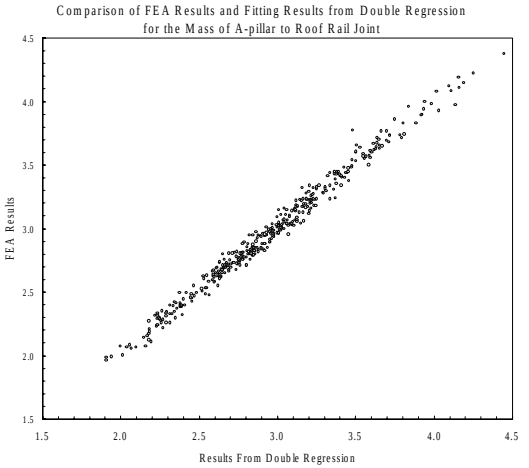
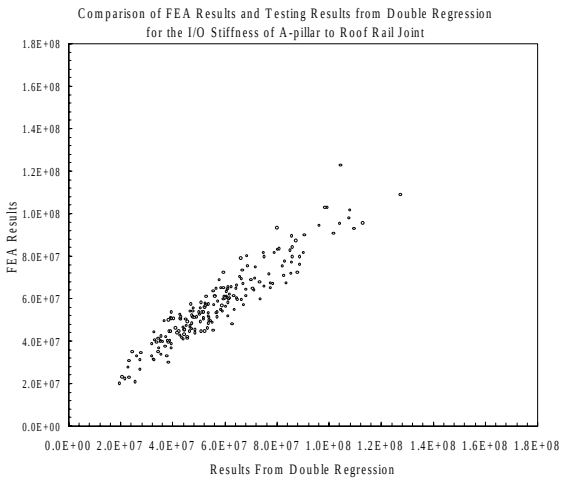
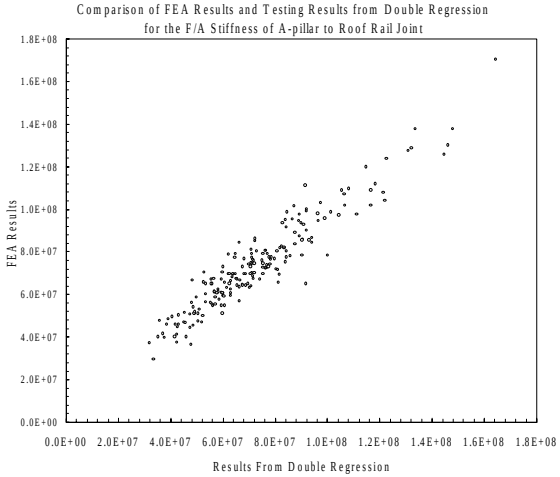
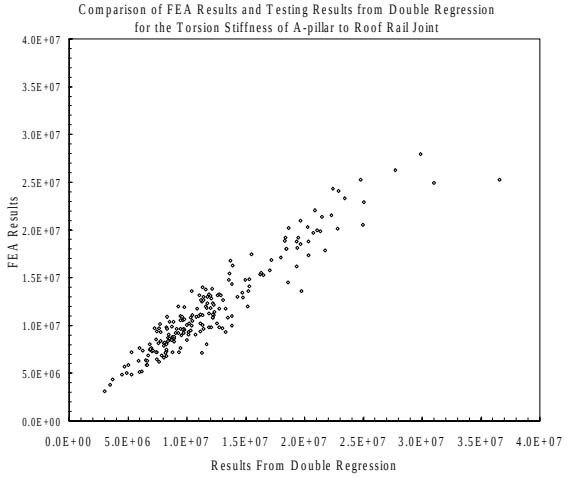


Figure 5.21a: Scatter Plots for the Fitting Results of Polynomial Translators







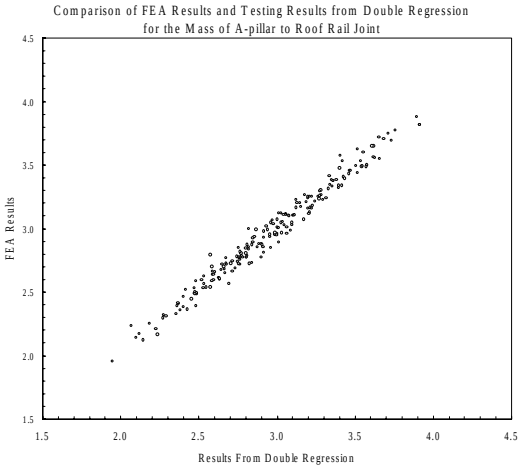
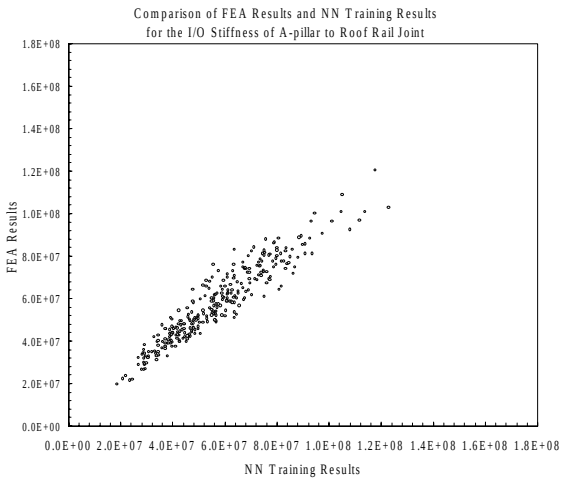
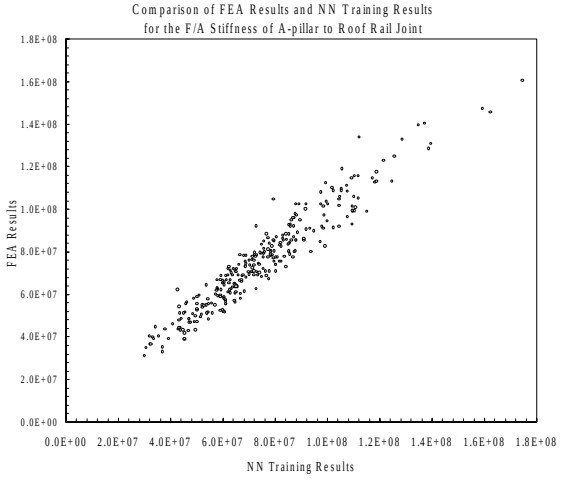
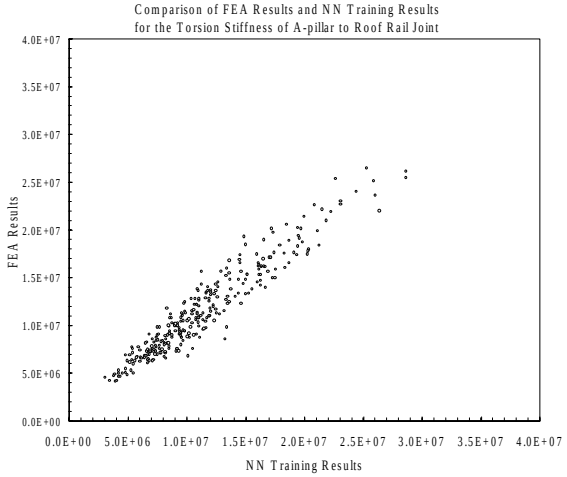


Figure 5.21b: Scatter Plots for the Testing Results of Polynomial Translators







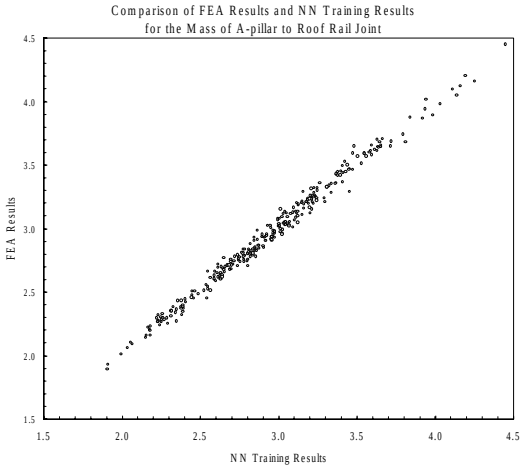
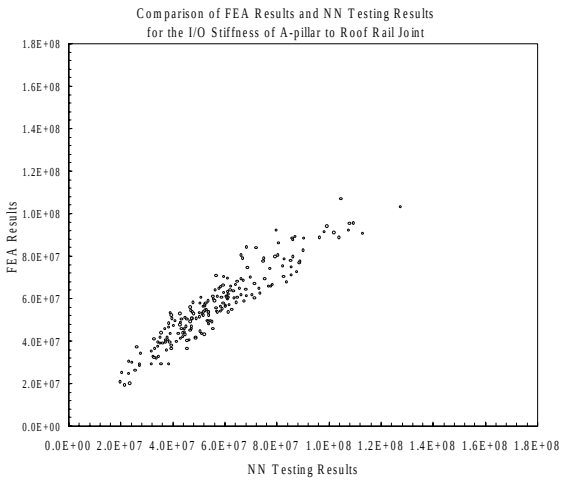
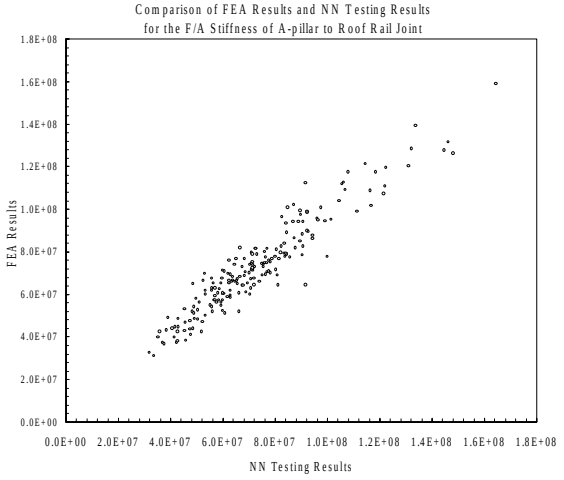
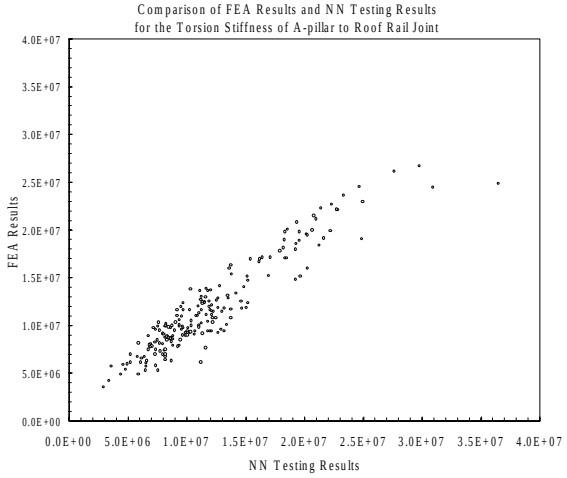


Figure 5.21c: Scatter Plots for the Training Results of Neural Network Translators







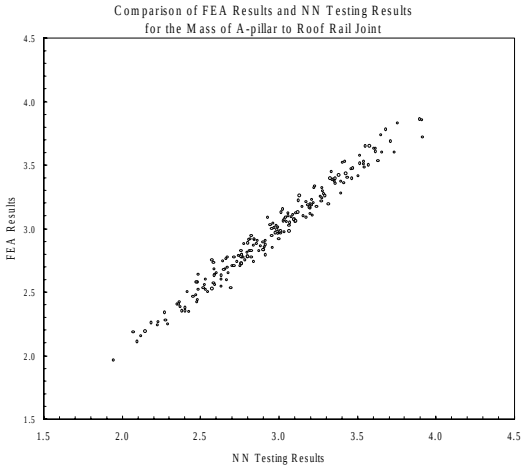


Figure 5.21d: Scatter Plots for the Testing Results of Neural Network Results

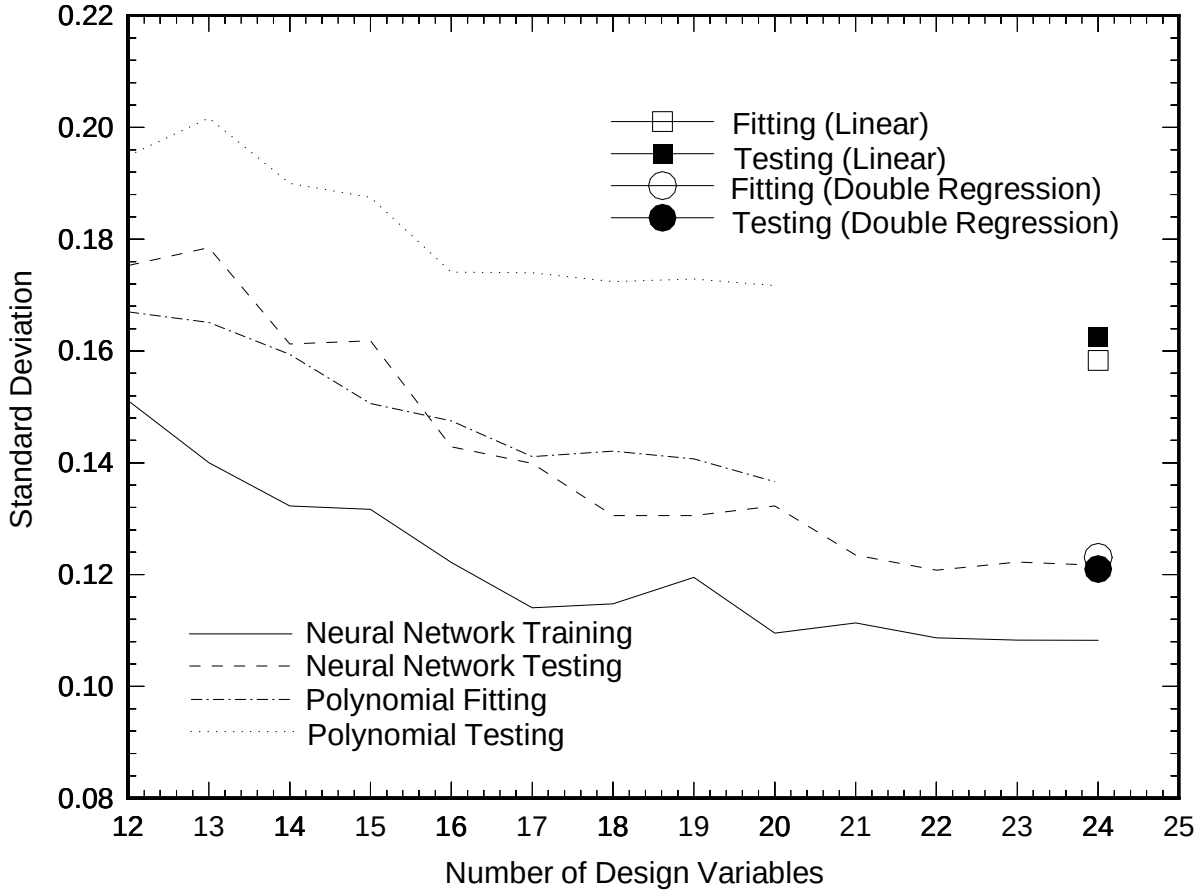


Fig. 5.22a: Comparison of Results for I/O Stiffness
of A-pillar to Roof Rail Joint

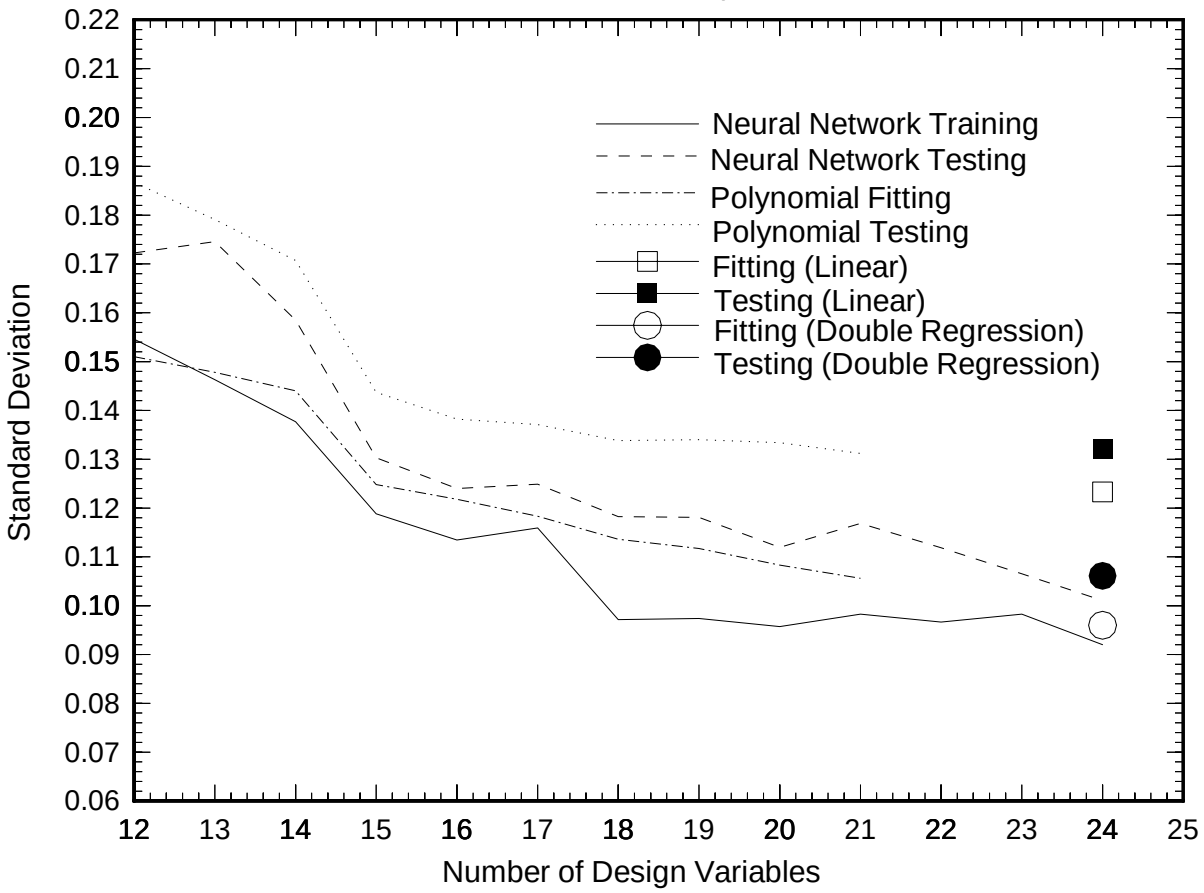


Figure 5.22b: Comparison of Results for F/A Stiffness of A-pillar to Roof Rail Joint

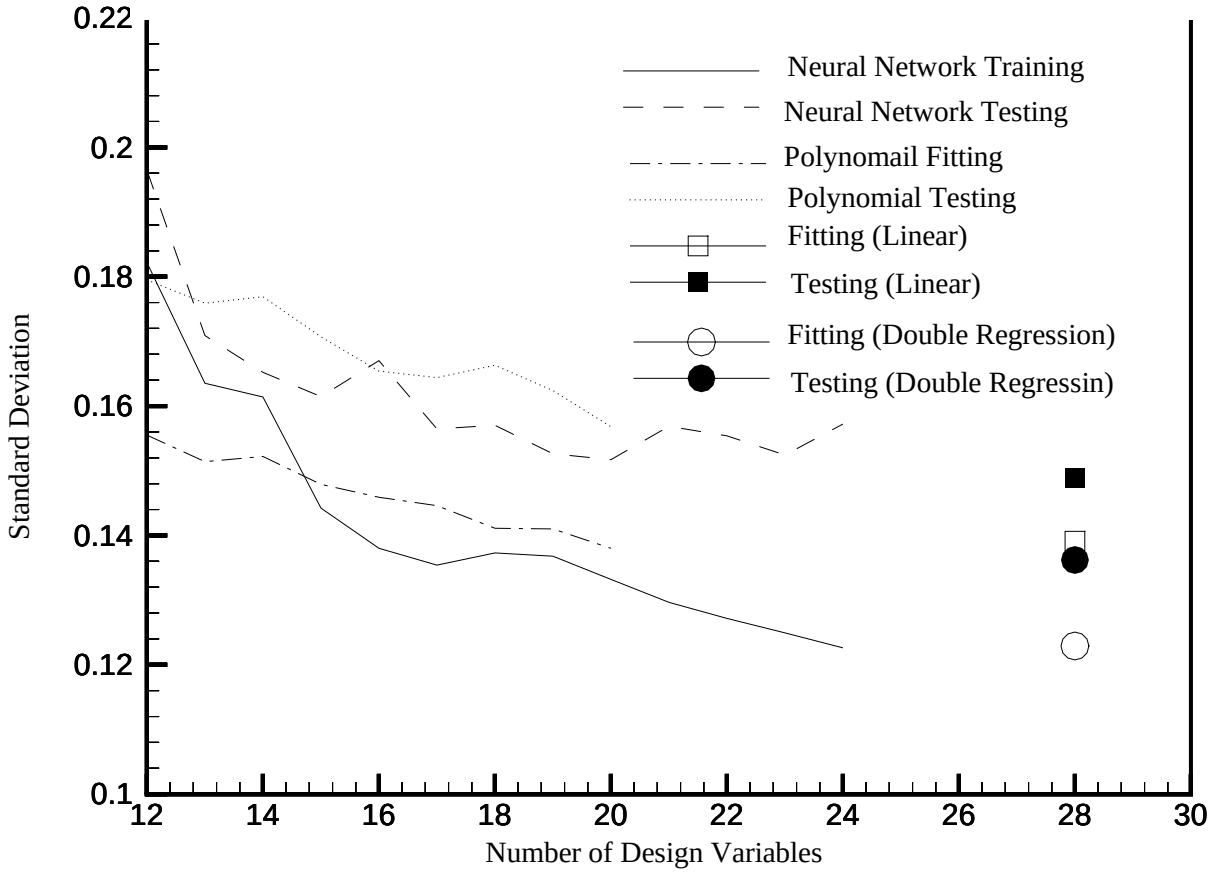


Figure 5.22c: Comparison of Results for Torsion Stiffness of A-pillar to Roof Rail Joint

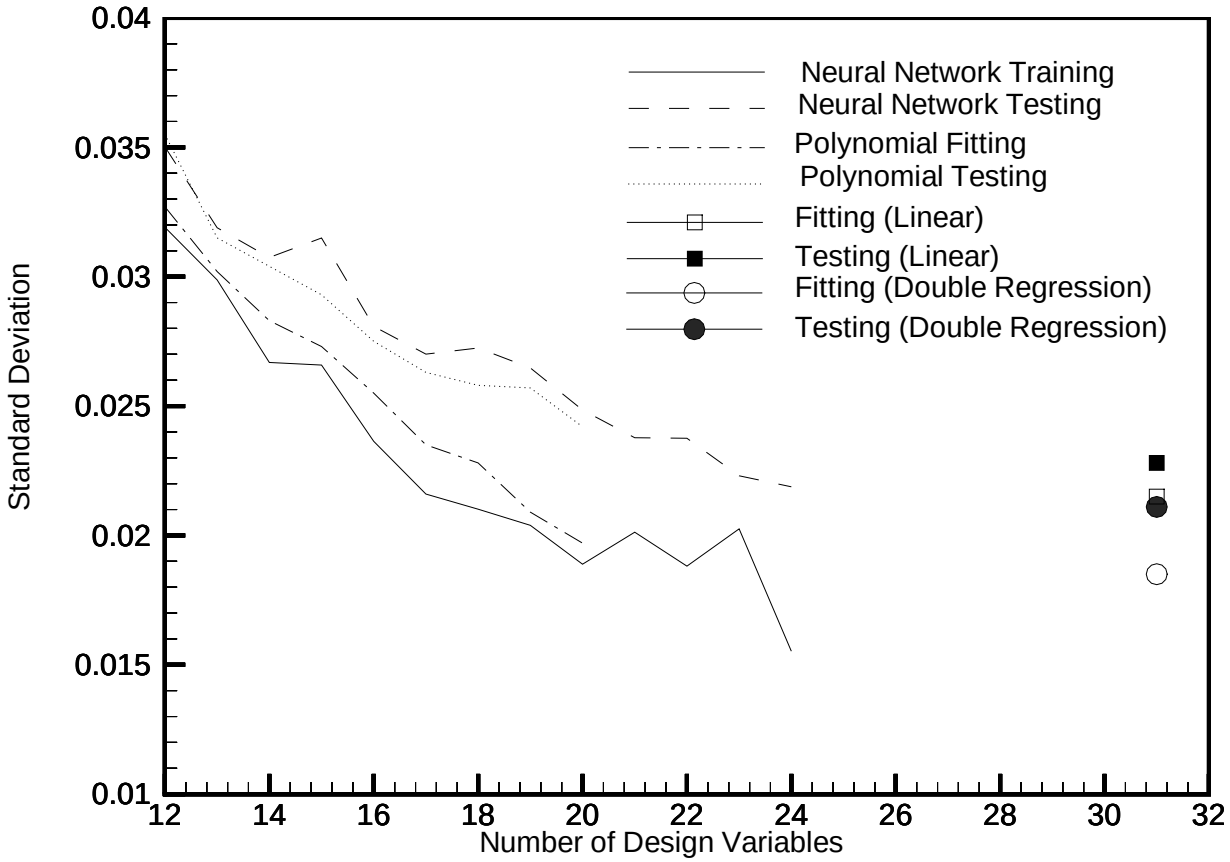
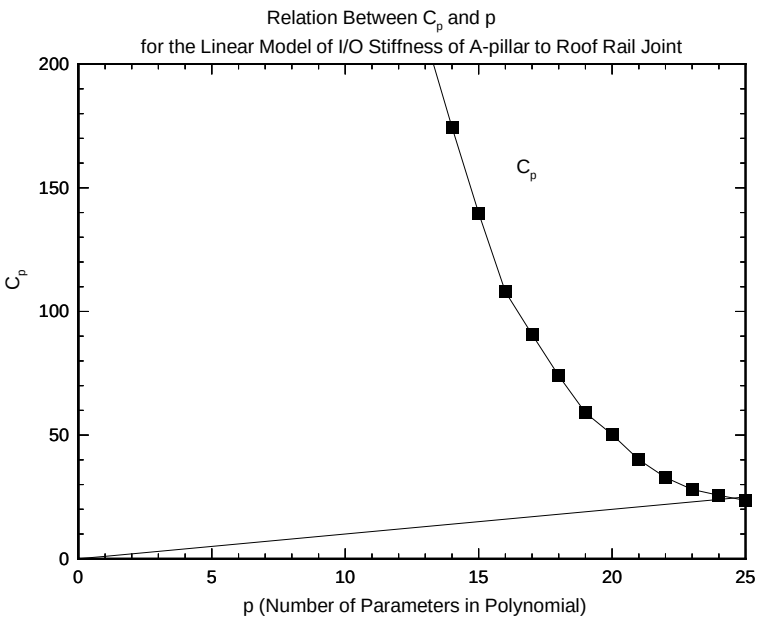


Figure 5.22d: Comparison of Results for the Mass of A-pillar to Roof Rail Joint



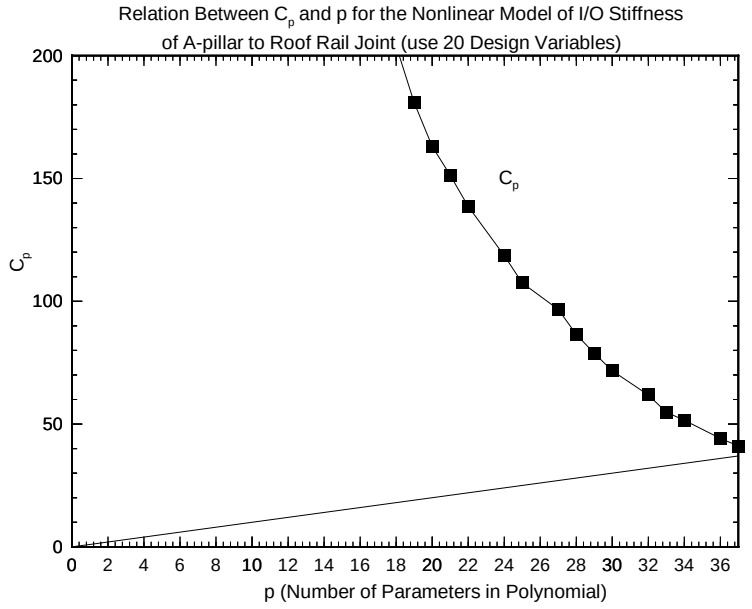
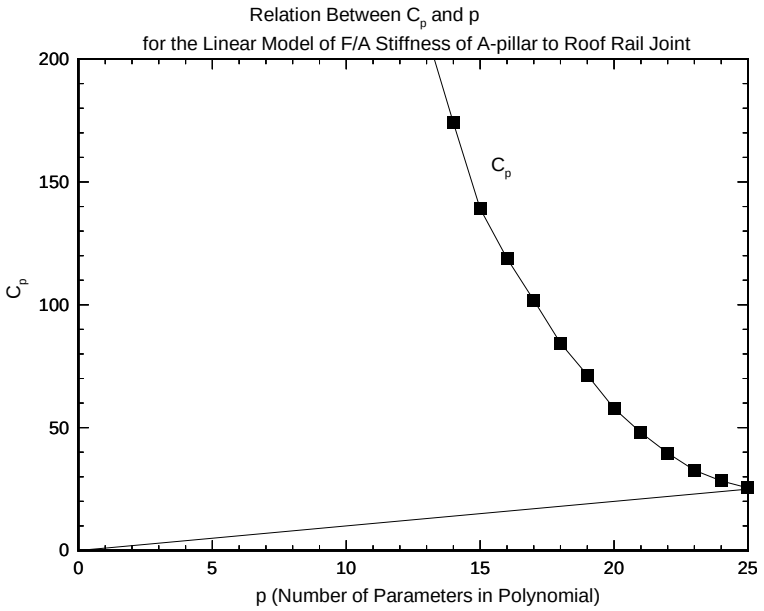


Figure 5.23a: Validating Using Cp Criterion for I/O Stiffness



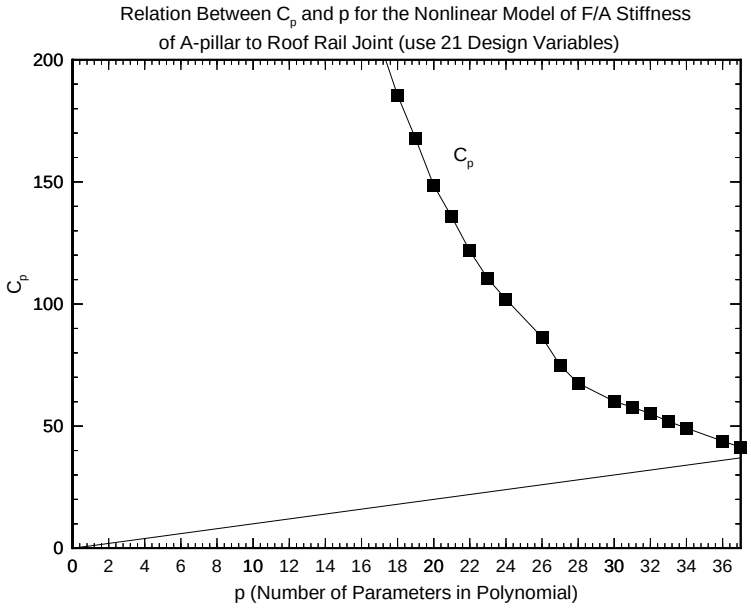


Figure 5.23b: Validating Using Cp Criterion for F/A Stiffness

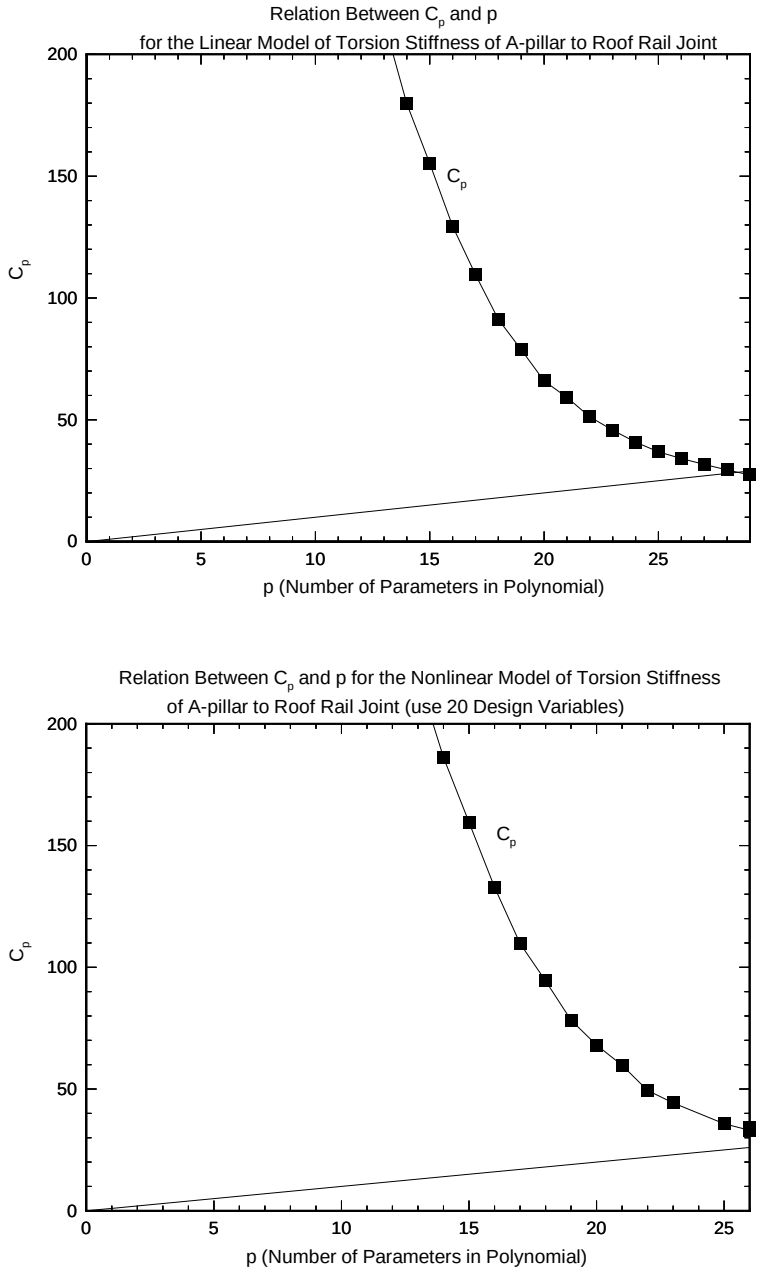


Figure 5.23c: Validating Using Cp Criterion for Torsion Stiffness

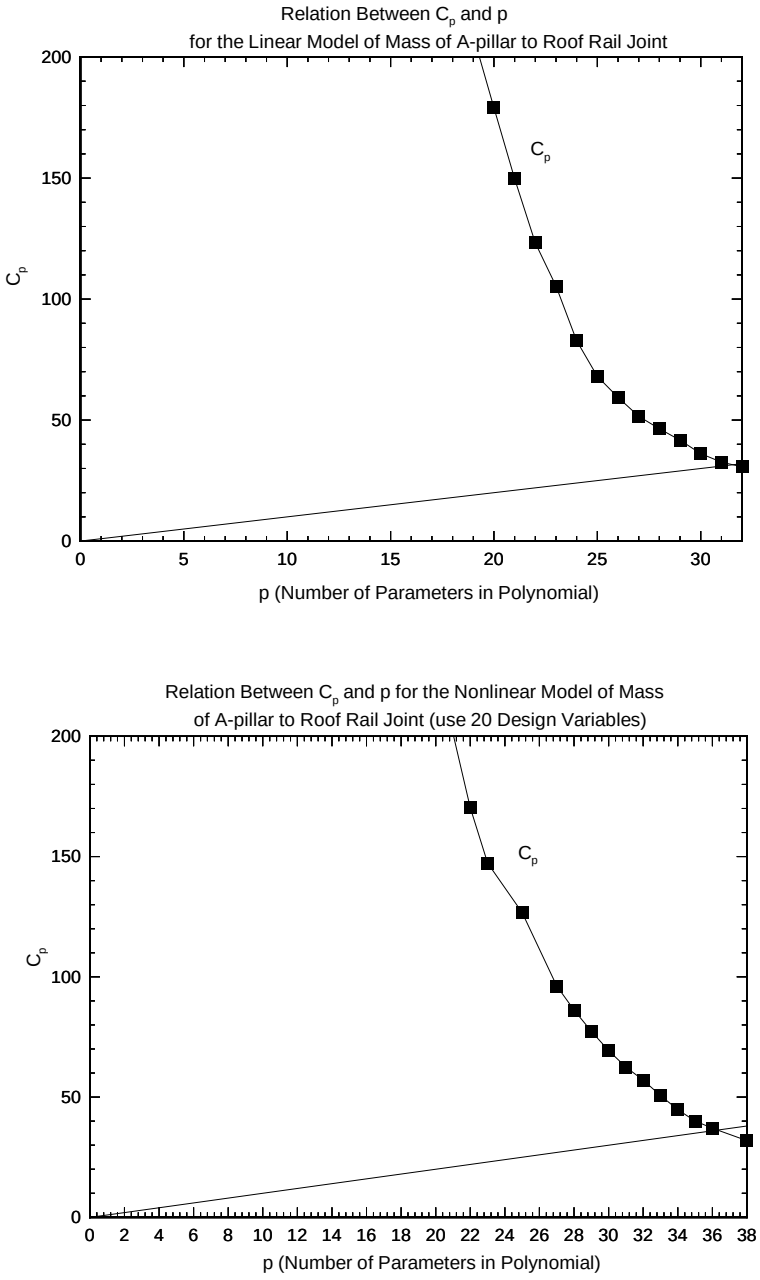


Figure 5.23d: Validating Using Cp Criterion for Mass

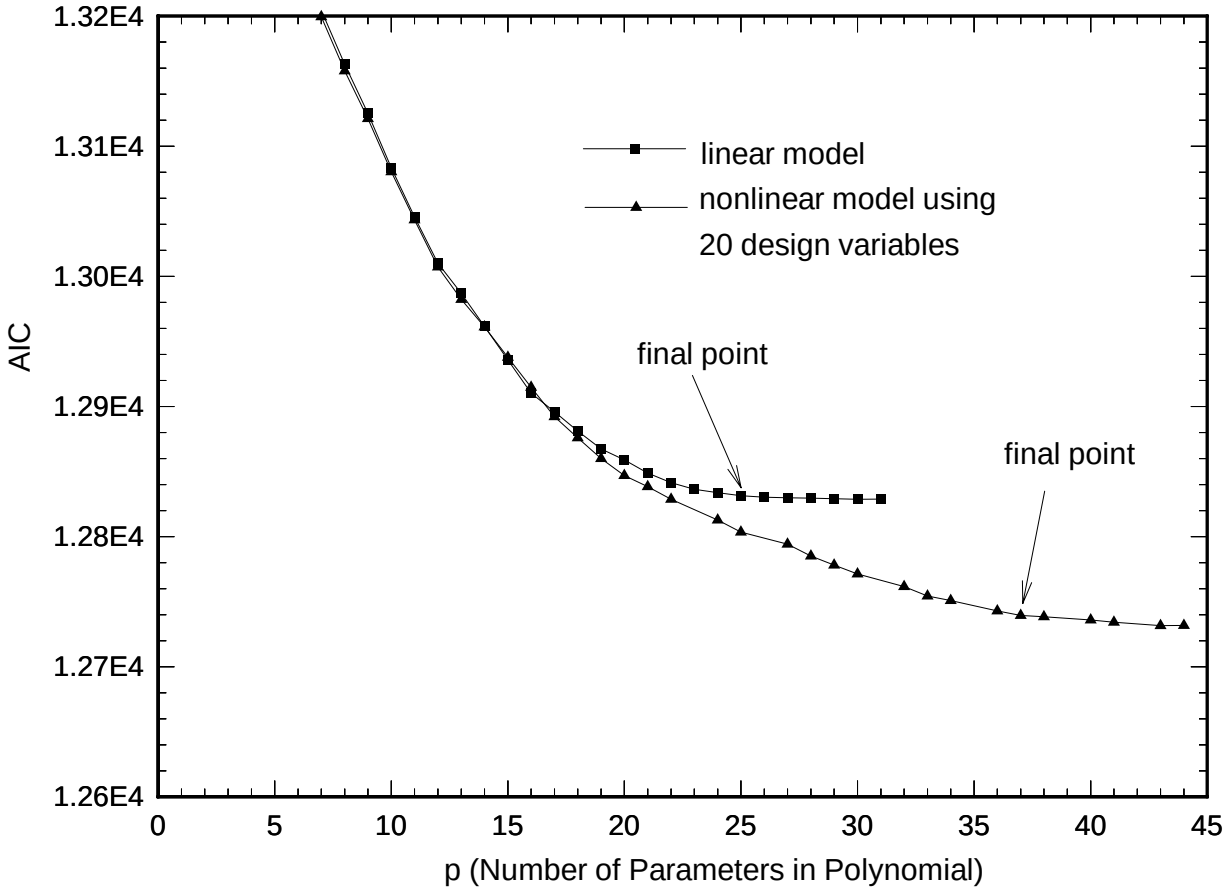


Figure 5.24a: Validating Using AIC Criterion for the I/O Stiffness of A-pillar to Roof Rail Joint

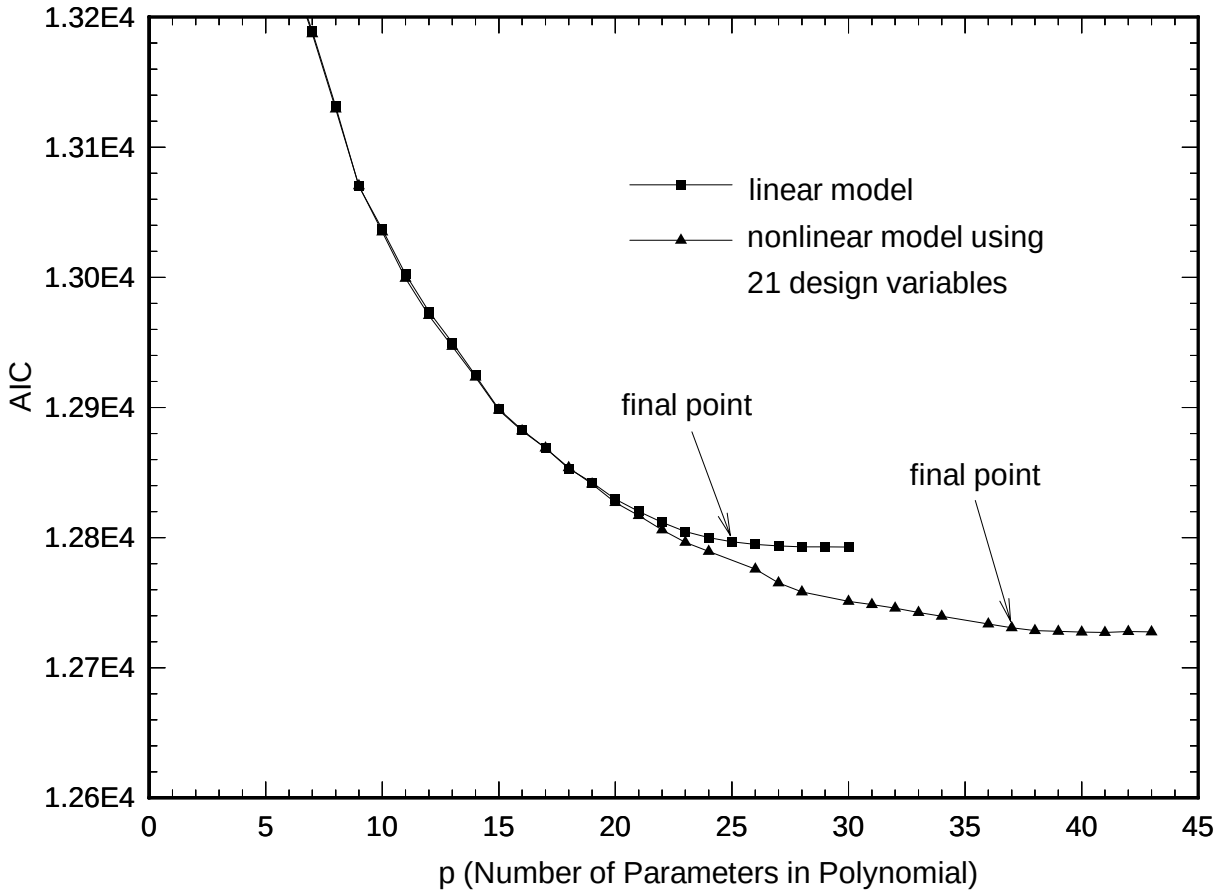


Figure 5.24b Validating Using AIC Criterion for the F/A Stiffness of A-pillar to Roof Rail Joint

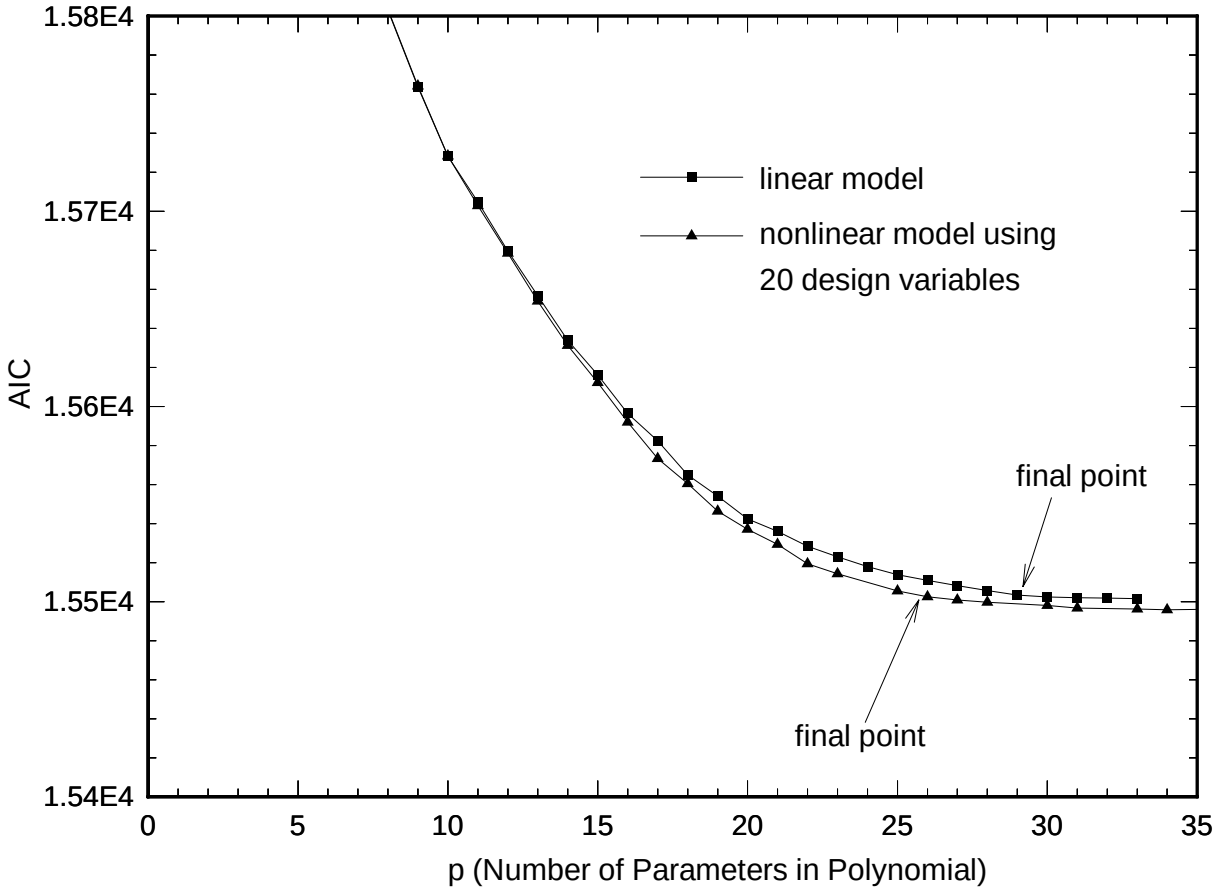


Figure 5.24c Validating Using AIC Criterion for the Torsion Stiffness of A-pillar to Roof Rail Joint

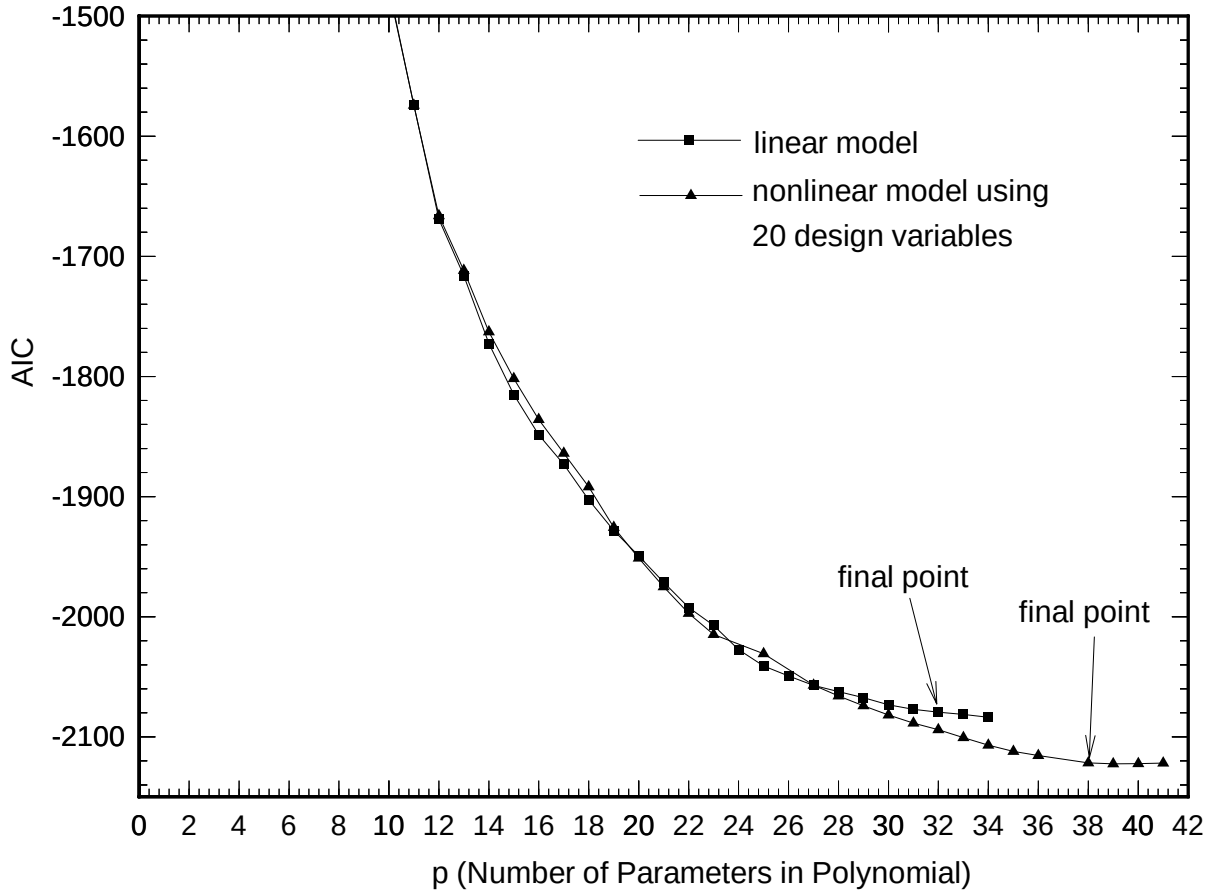


Figure 5.24d: Validating Using AIC Criterion for the Mass of A-pillar to Roof Rail Joint

Chapter 6
Developing Translator B for an Actual B-pillar to Rocker Joint

6.1 Introduction

This chapter applies the general methodology for developing translator B in Chapter 3 to a B-pillar to rocker joint. The B-pillar to rocker joint, which lies between the front and rear doors, is a T- like joint (see Chapter 4 for a detailed description of this joint). The vertical branch of the joint is called B-pillar, and the horizontal branch is called rocker. This chapter presents work that was completed in close collaboration with Mr. Ling. Therefore, parts of this chapter appear in his thesis (Ling, 1998).

This chapter is organized as follows:

- Section 6.2 formulates the optimization problem used in developing translator B. This section explains the selection and classification of design variables, the objective function, and the constraints.
- Section 6.3 presents some applications of translator B to some design problems. First, we use several methods to check the convergence of translator B. Then, we compare the optimum results from translator B with FEA results. We also compare the *response surface polynomial* (RSP) and *neural network* (NN) results since we use both to develop translator B. Then, we perform a parametric study to examine the effects of stiffness requirements on the mass of the optimum design, and the relation between objective function and the lower and upper bounds of some important design variables. Design guidelines for improving the design of the B-pillar to rocker joint are presented. The results are discussed and compared with those in other studies.

6.2 Formulation of the Optimization Problem for Developing a Translator B for a B-pillar to Rocker Joint

6.2.1 Definition of the Problem

Translator B uses optimization to find the most efficient design that meets given performance target (stiffness requirements), and satisfies all the constraints. Unlike translator A, which calculates the response of a joint whose design variables are specified, translator B solves the inverse problem. The user specifies performance requirements for the joint, including, stiffness requirements and constraints on the design variables, and translator B finds the most efficient design that satisfies all these requirements.

Structural optimization is an iterative process, and it would be very time-consuming if during the iteration, each new design were analyzed using FEA. To avoid this problem, a translator A is used to simulate FEA. As explained in Chapters 2 and 4, translator A is created using a continuous function to simulate the relation between the performance characteristics (stiffness and mass) and the values of the design variables of a design. Translator A predicts stiffness and mass almost instantaneously for a new design once its dimensions are specified. Thus, use of translator A enables the user to optimize the joint design. Two types of translator A are used in this study. One is created using response surface polynomials, and the other is created using neural networks.

There are different types of construction for B-pillar to rocker joints. As explained in Chapter 4, different types of construction are primarily characterized by different reinforcements. Some reinforcements are more common than others. For example, an extended pillar reinforcement is common. After consulting engineers from an automotive company, we decided to consider only one type of construction (Figs.4.2 and 4.3a). This type of B-pillar to rocker joint has an extended pillar reinforcement, and the center plate is connected to the bottom of the front rocker and the rear plate of the rocker (This rocker section is called generic type of rocker cross section. Fig. 4.6). Bulkheads are not considered in the model. Ling (1998) studied the effects of bulkheads on the stiffness.

Pillar bridge is not considered in the model because it is not effective (Murphy, 1995). Translator B uses 48 design variables.

6.2.2 Design Variables

The design variables for the B-pillar to rocker joint are shown in Table 6.1. As explained in Chapter 3, the design variables are divided into four types:

1) Design variables that are *fixed by the optimization program* because of design rules or conventions. An example is the length of rocker.
2) Design variables that are *fixed by the user*.
3) *Dependent design variables* whose values can be expressed in terms of other design variables.
4) *Independent design variables* that can change in optimization.

As explained in Chapter 4, some design variables are generally fixed because of packaging, and manufacturing requirements. The values of some design variables are determined from styling requirements. To compare the stiffness of different joint designs fairly, the rocker branch and B-pillar branch should be cut at the same length for different joint designs.

There is one design variable in the first type (fixed by the program), namely, *rocker_length* (Fig. 4.10). We fix it at 485 mm in order to compare fairly the stiffness of different joints.

There are seven design variables in the second type (fixed by the user), namely, *length_of_flange, spot_weld_spacing, spot_weld_placement, pillar_angle, pillar_io_angle, pillar_location,* and *aft_pillar_hole* (Figs.4.13, 4.7, 4.10, and 4.14). Because of manufacturing considerations, *length_of_flange, spot_weld_spacing,* and *spot_weld_placement* are fixed at 19 mm, 47.5 mm, and 9.5 mm, respectively. *Pillar_angle* and *pillar_io_angle* are fixed because the designer generally fixes the orientation of the B-pillar. *Pillar_location* is fixed because of door packaging considerations.

There are six design variables whose values depend on other independent design variables, namely, *pillar_height, fwd_outer_ver_blending_rad, fwd_outer_hor_blending_rad, aft_outer_ver_blending_rad, aft_outer_hor_blending_rad,* and *pillar_reinforcement_depth* (Figs. 4.10, 4.14, and 4.11). The length of the B-pillar branch is defined as the distance from the center of rocker central line to the top of the B-pillar (Fig. 6.1). The length of the B-pillar branch is fixed at 269 mm to allow for fair comparison of the stiffness and mass of different joints. *Pillar_height* is a dependent variable, which can be expressed in terms of the length of B-pillar branch and other design variables. The blending radii at the outer side of B-pillar, such as *fwd_outer_ver_blending_rad,* are assumed equal to their corresponding blending radii at the inner side of B-pillar, such as *fwd_inner_ver_blending_rad.* In this study, *pillar_reinforcement_depth* is a dependent design variable because we assume that the pillar reinforcement touches the bottom of the rocker. Table 6.1 classifies the design variables.

The optimization program does not directly change the design variables that are fixed by the program or by the user, and design variables that are dependent. Only 34 independent design variables are considered. The ranges of the design variables are shown in Table 6.1.

6.2.3 Objective Function

The objective function of translator B for the B-pillar to rocker joint can be expressed as:

$$F = \alpha M + (1-\alpha)\sqrt{((K_{I/O} - \hat{K}_{I/O})/\overline{K}_{I/O})^2 + ((K_{F/A} - \hat{K}_{F/A})/\overline{K}_{F/A})^2 + ((K_{Tor} - \hat{K}_{Tor})/\overline{K}_{Tor})^2} \quad (6.1)$$

where α is a weighting factor. M is the mass of the joint. $K_{I/O}, K_{F/A}$, and K_{Tor} are the I/O, F/A and torsion stiffnesses of the joint. $\hat{K}_{I/O}, \hat{K}_{F/A}$, and $\hat{K}_{Tor}$ are the user-specified requirements (targets) for I/O, F/A and torsion stiffness. $\overline{K}_{I/O}, \overline{K}_{F/A}$, and $\overline{K}_{Tor}$ are the stiffness values used to normalize the stiffness. Generally, mass varies from 2 to 8 kg. Each stiffness term varies from 0 to 3. We normalize the stiffness so that the mass and the stiffnesses can have compatible magnitude.

In this study, α is set to be 1. The objective function is equal to the mass. However, α can be set to be any value between 0 and 1. In this case, the objective function will be the sum of mass and some measure of difference between the stiffness of a design and the given stiffness targets. The optimum design obtained using this objective function will be a joint that not only has low mass but also its stiffness is close to the stiffness requirement.

6.2.4 Constraints

There are five types of constraints, namely, packaging, manufacturing, styling, mathematical, and performance target constraints.

Packaging constraints are related to the arrangement of car components in space. Manufacturing constraints are due to manufacturing limitations. Styling constraints are due to styling requirements. Mathematical constraints are used to control the range of different design variables to ensure that a design has a feasible geometry. Performance constraints dictate that the stiffnesses should exceed given minimum values.

The constraints used in translator B are explained in the following section. The number in the parenthesis of each constraint is the corresponding constraint number in the program for translator B.

- Packaging Constraints

1) Space must be provided for the door latch (Fig. 4.9) (4).

$$Pillar\_outer\_length - pillar\_inner\_length \leq 0 \qquad (6.2)$$

2) There should be enough room between the front and rear doors (Fig. 4.9)(52).

$$Minimum\ value\ of\ pillar\_outer\_length - pillar\_outer\_length \leq 0 \qquad (6.3)$$

3) There is a minimum requirement for the thickness of door (Fig. 4.9)(53).

$$\textit{Minimum value of outer\_pillar\_width - outer\_pillar\_width} \leq 0 \qquad (6.4)$$

4-7) Orientation of B-pillar must be within a range so that it is easy to get in and out of the car (Fig.4.7)(54-57).

$$\textit{Minimum value of pillar\_angle - pillar\_angle} \leq 0 \qquad (6.5)$$

$$\textit{Pillar\_angle - maximum value of pillar\_angle} \leq 0 \qquad (6.6)$$

$$\textit{Minimum value of pillar\_io\_angle - pillar\_io\_angle} \leq 0 \qquad (6.7)$$

$$\textit{Pillar\_io\_angle - maximum value of pillar\_io\_angle} \leq 0 \qquad (6.8)$$

8) Pillar reinforcement should not intersect with the bottom and/or and side of the rocker (Fig. 6.2)(5).

Case 1:

Pillar reinforcement intersects the front part of rocker (6.9a)

Case 2:

Pillar reinforcement intersects the bottom part of rocker (6.9b)

9-10) *Pillar_reinforcement_depth* must be within its range (Fig. 4.11) (84,85).

$$\textit{Minimum value of pillar\_reinforcement\_depth - Pillar\_reinforcement\_depth} \leq 0 \qquad (6.10)$$

$$\textit{Pillar\_reinforcement\_depth – maximum value of pillar\_reinforcement\_depth} \leq 0 \qquad (6.11)$$

11-18) Blending radii should be within the range [*Bld_Mi*, *Bld_Mx*] (Fig. 4.10)(58-65) because of door packaging considerations. *Bld_Mi* and *Bld_Mx* are the minimum and maximum values given by the user. Their default values are 30mm and 200mm, respectively.

$$Bld\_Mi - Fwd\_inner\_ver\_blendign\_rad \leq 0 \quad (6.12)$$

$$Fwd\_inner\_ver\_blendign\_rad - Bld\_Mx \leq 0 \quad (6.13)$$

$$Bld\_Mi - Aft\_inner\_ver\_blendign\_rad \leq 0 \quad (6.14)$$

$$Aft\_inner\_ver\_blendign\_rad - Bld\_Mx \leq 0 \quad (6.15)$$

$$Bld\_Mi - Fwd\_inner\_hor\_blendign\_rad \leq 0 \quad (6.16)$$

$$Fwd\_inner\_hor\_blendign\_rad - Bld\_Mx \leq 0 \quad (6.17)$$

$$Bld\_Mi - Aft\_inner\_hor\_blendign\_rad \leq 0 \quad (6.18)$$

$$Aft\_inner\_hor\_blendign\_rad - Bld\_Mx \leq 0 \quad (6.19)$$

19-20) Rocker height and width must not exceed a maximum value (Fig.4.8) (40,41).

$$Rocker\_height - maximum\ value\ of\ rocker\_height \leq 0 \quad (6.20)$$

$$Rocker\_width - maximum\ value\ of\ rocker\_width \leq 0 \quad (6.21)$$

21,22) The door edge height and width must be large enough to accommodate the sealant and the door edge (Fig. 4.8) (42,43).

$$Minimum\ value\ of\ door\_edge\_height - door\_edge\_height \leq 0 \quad (6.22)$$

$$Minimum\ value\ of\ door\_edge\_width - door\_edge\_width \leq 0 \quad (6.23)$$

23,24) There is a range for *A1* to allow for water drainage (Fig. 4.8) (44,45).

$$Minimum\ value\ of\ A1 - A1 \leq 0 \quad (6.24)$$

$$A1 - 90 \leq 0 \quad (6.25)$$

25,26) There is a range for A5 from seat packaging considerations (Fig.4.8)(46,47).

$$Minimum\ value\ of\ A5 - A5 \leq 0 \quad (6.26)$$

$$A5 - maximum\ value\ of\ A5 \leq 0 \quad (6.27)$$

27,28) There is a range for A3 from packaging consideration (Fig. 4.8) (49,50).

$$\textit{Minimum value of A3-A3} \leq 0 \tag{6.28}$$

$$\textit{A3- maximum value of A3} \leq 0 \tag{6.29}$$

29) There must be a slope for water to run off (Fig. 6.3a) (51).

$$\textit{A6Mi (Minimum value of A6)} - \textit{A6} \leq 0 \tag{6.30}$$

where *A6Mi* is given by the user. A default value of 3 degree was used in the examples of this chapter.

30) The slope of top plate of inner rocker should not be too high (Fig.6.5) (72).

$$\textit{Atan(d7/h4)} - \textit{SRMx (maximum value of the slope)} \leq 0 \tag{6.31}$$

The user can define *SRMx*. A default value of 10 degree was used in the following examples.

31) The width of inner rocker cell can not be too large because of seat packaging considerations (Fig. 4.8) (83).

$$\textit{Rocker\_width - outboard\_cell\_width} - \textit{widInRkMx (maximum value)} \leq 0 \tag{6.32}$$

where *widInRkMx* is defined by the user. A default value of 82mm was used in the following examples.

32) There should be enough room to put afterward bulkhead (Figs. 4.10 an 4.14) (9).

$$Aft\_bulk\_head\_position + aft\_inner\_hor\_blendign\_rad - pillar\_location \leq 0 \quad (6.33)$$

33) There should be room to put forward bulkhead (Figs. 4.10 and 4.14)(6,7)

$$Pillar\_location + fwd\_inner\_hor\_blendign\_rad + pillar\_base + pillar\_inner\_length - pillar\_outer\_length - fwd\_bulk\_head\_position \leq 0 \quad (6.34)$$

34) *Inner_pillar_base_width* should be smaller than the horizontal projection of *BR_L8* to connect the pillar back with the inner plate of rocker (Figs. 4.15 and 6.3) (71).

$$Inner\_pillar\_base\_width - horizontal\ distance\ of\ BR\_L8 \leq 0 \quad (6.35)$$

- Manufacturing Constraints

Manufacturing constraints include stamping and welding constraints. One needs very detailed information about the geometry to determine if a given design is feasible. In the early design stages this information is not available. We have used crude equations to check if a design is manufacturable. Stamping constraints include constraints for strains, spring back, and die lock. To ensure that plastic strains are low, we impose low limits on the edges of each section, and the angles between two adjacent plates of a section. We also impose an upper limit on the depth of draw to avoid overstretching a plate (Fig.6.8a). Spring back constraints ensure that a plate can be permanently bent. For this reason, the deformation angle should exceed a minimum value (Fig.6.6). Constraints on die lock ensure that the die and block can be separated after the plate has been stamped. For this purpose, the vertical walls of the hat-shape section should be slanted away from the center of the section (Fig. 6.7a).

35-42) The length of each edge on the rocker section should be greater than a minimum value, so that the rocker inner and outer shells can be stamped (Fig. 6.3) (12-19).

$$RMin\_L \text{ (minimum length)} - BR\_Lj \leq 0 \qquad j=1,\ldots,8 \tag{6.36}$$

where *RMin_L* is minimum length. The user can define this length. A default value of 10 mm was used in the examples of this chapter.

43-50) The angle between two adjacent edges of the rocker section must be larger than a minimum value to avoid a sharp angle between the plates (Fig. 6.4) (20-27).

$$RMin\_A \text{ (minimum angle)} - BR\_Anglej \leq 0 \qquad j=1,\ldots,8 \tag{6.37}$$

where *RMin_A* is minimum angle. The user can define this angle. A default value of 20 degrees was used in the examples of this chapter.

51,52) The top and bottom plates of the outer rocker shell should be able to avoid die lock (Figs. 4.8, 6.7a and 6.7b)(28,29). Die lock angle is the minimum requirement on the angle between the moving direction of a die and the side shell of a die casting part. This constraint is due to manufacturing consideration.

$$ADieMi \text{ (minimum value of die angle)} - (90\text{-}Aj) \leq 0 \qquad j=1,2 \tag{6.38}$$

where the minimum angle *ADieMi* is defined by the user. A default value of 3 degrees was used in the examples of this chapter.

53) The draw angle of the front rocker should be smaller than a maximum value from stamping considerations (Fig.6.6) (30). Draw angle is the maximum allowable angle between the moving direction of a die and normal direction of a shell metal from which a die casting part is made.

$$\textit{Draw angle of front rocker} - \textit{AdrawMx (maximum value)} \leq 0 \qquad (6.39)$$

where *AdrawMx* is a maximum angle. The user can define this angle. A default value of 30 degrees was used in the examples of this chapter.

54) The ratio of depth of draw over the width of the draw should be smaller than a given value to avoid excessive plastic strains (Figs. 6.8a and 6.8b) (31).

$$\textit{Depth of draw (outer rocker cell)/Width of draw (outer rocker cell)} - \textit{RaDraw (maximum ratio)} \leq 0 \qquad (6.40)$$

The maximum value of *RaDraw* is defined by the user. A default value of 1.5 was used in the examples of this chapter.

55) There should be no sudden change in depth of draw (Fig. 6.8b) (32).

$$\textit{Changes in depth of draw} - \textit{DpDrawMx (maximum value)} \leq 0 \qquad (6.41)$$

where *DpDrawMx* is a maximum value. The user can define this value. A default value of 40 mm was used in the examples of this chapter.

56,57) The upper and lower plates of the inner rocker should be able to avoid die lock (Figs. 4.8, 6.7a and 6.7b) (33,34).

$$\textit{ADieMi (minimum value)} - (90 - \textit{A8}) \leq 0 \qquad (6.42)$$

$$\textit{ADieMi (minimum value)} - \textit{BR\_1Ang} \leq 0 \qquad (6.43)$$

58) The draw angle for the inner rocker shell should be smaller than a maximum value from stamping requirement (Figs. 6.8a and 6.8b) (35).

$$\textit{Draw angle of back rocker} - \textit{AdrawMx (maximum value)} \leq 0 \qquad (6.44)$$

59) Depth of draw for the inner rocker cell should not be too large relative to its width to avoid excessive plastic strains (Figs. 6.8a and 6.8b) (36).

$$\textit{Depth of draw (inner rocker cell)/width of draw (inner rocker cell)} - \textit{RaDraw (maximum ratio)} \leq 0 \qquad (6.45)$$

60-69) Spring back angle of the plates of rocker must be smaller than a maximum value (Fig. 6.4) (73-82). Spring back angle is the maximum allowable angle between two plates to create a permanent deformation.

$$\textit{SpBkAngj} - (180 - \textit{spBkMi (minimum value)}) \leq 0 \qquad j=0,\ldots,9 \qquad (6.46)$$

The user can define *spBkMi*, which is a minimum value. A default value of 20 degrees was used to obtain the results in this chapter.

70,71) Depth of pillar plates should not be too large relative to their width to avoid excessive plastic strains (Fig.4.9) (37,38).

$$\textit{Outer\_pillar\_width/Pillar\_outer\_} - \textit{RaDraw (maximum ratio)} \leq 0 \qquad (6.47)$$

$$\textit{Inner\_pillar\_width/Pillar\_inner\_length} - \textit{RaDraw (maximum ratio)} \leq 0 \qquad (6.48)$$

72) The distance between spot welds and the root of a flange should be larger than a minimum value from manufacturing considerations (Fig. 4.13) (66).

$$\textit{Weld\_root\_Mi(minimum value of weld\_root\_distance} - (\textit{length\_of\_flange-spot\_weld\_placement}) \leq 0 \qquad (6.49)$$

The user can define the value of *Weld_root_Mi*. A default value of 6 mm was used to get the results of examples in this chapter.

73) *Spot_weld_spacing* should be larger than a minimum value from manufacturing considerations (Fig.4.13) (67).

$$Spot\_W\_Mi \text{ (minimum value of spot\_weld\_spacing-spot\_weld\_spacing} \leq 0 \tag{6.50}$$

The user can define the value of *spot_W_Mi*. A default value of 35 mm was used to get the results of examples in this chapter.

- Styling Constraints

74) The bottom flange of the rocker should not be visible to a person standing at one side of the car (Fig. 6.9) (39).

$$BR\_alpha - BR\_beta \leq 0 \tag{6.51}$$

75) The slope of the outer rocker cell, A3, should be approximately equal to the slope of the door (Fig. 4.8) (48).

$$|-A3 + Ang\_Door| - ABDrMx \leq 0 \tag{6.52}$$

The user can define the values of *ABDrMx* and *Ang_Door*. Their default values are 15 degree and 20 degrees, respectively.

- Mathematical Constraints

76) The summation of *inner_rocker_height* and height of *BR_L6* should be less than the overall height of rocker (Figs. 4.8 and 6.3) (1).

$$Inner\_rocker\_height + vertical\ distance\ of\ L6\text{-}rocker\_height \leq 0 \quad (6.52)$$

77) The summation of *inner_flange_distance* and *outboard_cell_width* should be less than the overall width of rocker to ensure a meaningful rocker section (Fig. 4.8) (2).

$$Inner\_flange\_distance + outboard\_cell\_width - rocker\_width \leq 0 \quad (6.53)$$

78) The summation of *low_door_ht_minus_clearance* and *door_edge_height* should be less than overall height of rocker to ensure that a design has a meaningful rocker section (Fig. 4.8) (3).

$$Low\_door\_ht\_minus\_clearance + door\_edge\_height - rocker\_height \leq 0 \quad (6.54)$$

79,80) The inner vertical blending radii must no exceed the pillar height (Fig.4.10) (67).

$$aft\_inner\_ver\_blending\_rad - pillar\_height \leq 0 \quad (6.55)$$

$$fwd\_inner\_ver\_blending\_rad - pillar\_height \leq 0 \quad (6.56)$$

81) The summation of *pillar_location, pillar_base, fwd_inner_hor_blending_rad, pillar_inner_length, pillar_outer_length,* and *rocker_length* should be less than the overall length of rocker (Fig. 4.10)(11).

$$Pillar\_location + Pillar\_base + fwd\_inner\_hor\_blendign\_rad + pillar\_inner\_length - pillar\_outer\_length - rocker\_length + 20 \leq 0 \quad (6.57)$$

where 20 is small margin to avoid finding a design that can not be created by the parametric CAD model.

82) The afterward inner horizontal blending radius must be smaller than the length of pillar_location (Fig. 4.10) (9).

$$Aft\_inner\_hor\_blending\_rad - pillar\_location \leq 0 \tag{6.58}$$

- Performance Target Constraints

83-85) Typically, the stiffness of a joint must meet two types of requirements: it should not be less than a minimum value and/or should be close to a target value. The first requirement is expresses as follows:

$$(\hat{K}_{I/O} - K_{I/O}) / \overline{K}_{I/O} \leq 0 \tag{6.59}$$
$$(\hat{K}_{F/A} - K_{F/A}) / \overline{K}_{F/A} \leq 0 \tag{6.60}$$
$$(\hat{K}_{Tor} - K_{Tor}) / \overline{K}_{Tor} \leq 0 \tag{6.61}$$

6.3 Results and Discussion

Above optimization problem was solved several times by changing the design requirements for a joint using nonlinear optimization program DOT (1995). First, optimization results were validated by comparing the stiffness and mass of the obtained joint designs with FEA results. Then, results obtained using response surface polynomial and neural network translators were presented and compared.

6.3.1 Checking the Convergence of the Optimization Program

The optimization of B-pillar to rocker joint is a nonlinear problem. The optimizer may converge to a local minimum instead of a global minimum. We can use three methods to check whether the optimization program converges to a global minimum.

- We solve the problem from different initial points to see if the optimization program converges to the same optimum design.
- We use the final optimum design as an initial point, and solve the problem again to see if the optimum design changes.
- We also solve the same problem using different methods to see if the optimization program converges to the same point. In this study, we used the method of modified feasible direction (MFD) and sequential linear programming (SLP).

Eight randomly generated designs were used as the initial points. The measured stiffness of an actual car joint was used as the target stiffness. We only considered mass in the objective function, e.g., α in Eq.6.1 was set to be one.

Table 6.2 shows the objective functions of the optimum designs when we start from different initial points. We used sequential linear programming (SLP) to obtain the above results. Both RSP and NN translators were tested. According to Table 6.2, the maximum relative difference between the masses of optimum designs is 0.1%. The first 2 to 3 digits of the objective function are the same when starting from different initial designs. For the final design, the first 2 to 3 digits of design variables are found to be the same. Practically no improvement in the objective function (less than 0.1%) was achieved by solving the problem again starting from the obtained optimum designs.

We also used modified feasible direction method (MFD) to check the convergence of the optimization program when we start from different designs. We found that the optimum design obtained using MFD was the same as that obtained using SLP method. However, performing optimization using MFD method needed more iterations than using SLP method. SLP was more efficient than MFD for the optimization problems in this study. Because of these reasons, we chose SLP as the default optimization method in the translator B for B-pillar to rocker joint. All the optimization results presented in this chapter were obtained using SLP method. The user can easily change to the MFD method by changing the identification number of the optimization method in the program of translator B.

6.3.2 Comparison of Results of Translator B with FEA Results

As mentioned earlier, translator B finds the joint with the smallest mass whose stiffnesses are equal to or higher than given target values. To validate translator B, we compared the mass and stiffnesses of optimum designs calculated from translator B and from FEA. To generate some random performance targets (stiffness requirements), we first needed to select ranges in which I/O, F/A, and torsion stiffnesses could vary. We observed that the stiffnesses in different directions were correlated. The user should not ask for a joint that has very high I/O stiffness, but very low F/A, and torsion stiffness because the user may get an optimum design whose stiffness correlates poorly with FEA results. The following is the main reason. Since both the RSP and NN were created using a database obtained using FEA, each stiffness in this database has a range in which it can vary. Translator A is accurate for designs that are close to the center of the feasible domain of the design variables in the database. The predictions from translator A deteriorate when a design approaches the boundary of the database. If the stiffness requirement is high in one direction, but low in another, the optimization program will find designs that are close to the boundary of the region or even outside this region. As a result, translators A and B will not predict the stiffness and mass of the optimum design accurately.

To avoid the above problem, we divided the database into several zones by observing the scatter plots of I/O stiffness versus F/A stiffness, and I/O stiffness versus torsion stiffness of the database. Figures 6. 10a and 6.10b show that the I/O, F/A, and torsion stiffness were actually correlated. For example, a design with a large I/O stiffness often has large F/A and torsion stiffness. Therefore, to ensure that optimization results from the translator B correlate well with FEA results, the combination of stiffness targets should be within these zones. Table 6.3 shows eight zones of the database. We have implemented these zones in the translator B for B-pillar to rocker joint. The translator B will warn the user that the optimization results may be invalid if the combination of stiffness targets is outside the zones.

We selected a number of random performance targets (stiffness requirements) within the zones. Using translator B, the optimum design corresponding to each set of stiffness

requirements was found. Then Pro/Engineer was used to create the models corresponding to the optimum designs, and the NASTRAN bulk data files. We analyzed the model using MSC/NASTRAN to find the stiffnesses and masses of the optimum designs. Table 6.4a and Figures 6.11a-6.11d compare the predictions of translator B for the I/O, F/A, torsion stiffnesses and mass with FEA results when response surface polynomial translators were used. Table 6.4b and Figs. 6.12a-6.12d show the corresponding results for neural network translators.

The *correlation coefficient* is used to measure the correlation between the FEA and the optimum results. Correlation coefficient is:

$$\rho_{Y_o, Y_{FEA}} = \frac{cov(\boldsymbol{Y}_O, \boldsymbol{Y}_{FEA})}{\sigma_{Y_O} \times \sigma_{Y_{FEA}}} \tag{6.62}$$

where

$$\mathrm{cov}(\mathbf{Y}_O, \mathbf{Y}_{FEA}) = \frac{1}{n_c} \sum (y_{O,j} - \bar{y}_O)(y_{FEA,j} - \bar{y}_{FEA}) \tag{6.63a}$$

$$\sigma^2_{Y_O} = \frac{1}{n_c} \sum (y_{O,j} - \bar{y}_O)^2 \tag{6.63b}$$

$$\sigma^2_{Y_{FEA}} = \frac{1}{n_c} \sum (y_{FEA,j} - \bar{y}_{FEA})^2 \tag{6.63c}$$

where $y_{O,j}$ and $y_{FEA,j}$ are the optimization and FEA results, respectively. $\bar{y}_O$ and $\bar{y}_{FEA}$ are the mean values of the optimization and FEA results. n_C is the number of designs used in comparison. $\boldsymbol{Y}_O$ and $\boldsymbol{Y}_{FEA}$ are the vectors of the optimization and FEA results, respectively. In general, a correlation coefficient larger than 0.9 shows good agreement between predictions of translator B and FEA results. Note that the square of the correlation coefficient is equal to the coefficient of determination, R^2, which was defined in Eq. 2.5

The correlation coefficients for stiffnesses in different directions are given in the Figs. 6.11a - 6.12d and Table 6.4c. The correlation coefficients of RSP translators

range from 0.9677 to 0.9920. For the NN translators, the correlation coefficients range from 0.8809 to 0.9810.

6.3.3 Redesign of the Joints of two Cars Using Translator B

We also used the translator B to redesign two actual car joints. We set the stiffness targets equal to the stiffnesses of the actual joints. We obtained the optimum designs using the translator B. Tables 6.5a to 6.5d show the optimization and FEA results.

From the comparison of the optimization and FEA results of the two joints, we observed the following trends when optimizing the shapes of the joint components:

B-pillar

- *Pillar_outer_length* reaches its upper bound at the optimum design. *Outer_pillar_width* increases when the stiffness requirements increase. Both help increase the cross section of B-pillar and the stiffness.
- *Pillar_base* increases as the stiffness requirements increase. This is reasonable because the stiffness of a joint is expect to increase rapidly as the section modulus of the beam leading to the joint increases.

Pillar reinforcement

- Because *pillar_reinf_base_width* and *pillar_reinf_expansion* only have little effects on the stiffness of the joint, they take values close to their lower bounds to reduce the mass of the joint.

Cross section of rocker

- *Rocker_height* and *low_door_ht_minus_clearance* reach their lower bounds. *outboard_cell_width* is close to its lower bound. *Rocker_width, inner_rocker_height* and *inner_flange_distance* assume values between their lower and upper bounds.
- *Door_edge_height* tends to be as large as possible, and *door_edge_width* tends to be small as possible. This trend of the translator to minimize the door edge width also has been observed by Zhu (1994). The reason for reducing the

door_edge_width is that the stiffness increases rapidly as the *door_edge_width* decreases.

- A1, A3, and A8 reach their lower bounds. A2 and A7 are close to their upper bounds. As a result of these trends, the front part of rocker tends to become vertical, and the bottom of rocker tends to become horizontal.

Pillar hole

- Among the four dimensions that define the opening of B-pillar, *aft_pillar_hole* is fixed. *Bottom_pillar_hole* and *fwd_pillar_hole* reach their upper bounds. This makes the opening of B-pillar as small as possible.

Plate thickness

- The value of *thickness of front rocker* assumes the lower bound for car B and a value between its lower and upper bounds for car A. The parametric study in Section 6.3.4 shows that increasing the *thickness of front rocker* greatly increases both the mass and stiffness of a design. Therefore, determining a suitable value between the lower and upper bounds is a trade-off for an optimum design between high stiffness and low mass. For smaller stiffness requirements, the thickness of front rocker reaches its lower bounds to reduce the mass. For high stiffness requirements, it takes a value between its lower and upper bounds.
- *Thicknesses of pillar reinforcement, center plate* and *back rocker* do not have big effects on the stiffness. They reach their lower bounds to reduce the mass of the joint.
- *Thickness of pillar back* increases as the stiffness requirements increase. The study in Section 6.3.4 also shows that increasing the thickness of pillar back increases the stiffness of the joint.

Table 6.5d compares the mass of the two joints before and after optimization. It is found that improvements range from 8.1% to 27.5%. It should be noted that we used the bounds measured from actual cars to obtain the above results. In practice, the ranges of the design variables may be smaller, and some design variables may be fixed because of styling, packaging, or manufacturing considerations. This will reduce the region in

which the optimizer can search for an optimum design. Therefore, actual improvement of two joints may be smaller than that in Table 6.5d. Figures 6.13 compares the shapes of the initial and the optimum designs.

We also observed the 12 designs compared in Section 6.3.2, and found the trends were the same as the above.

6.3.4 Parametric Study

Figures 6.14 to 6.16 show the effect of I/O, F/A, and torsion stiffness requirements on the mass of the optimum design. To get the above results, we only changed one stiffness requirement at a time and fixed the other two stiffness requirements at the values corresponding to an actual joint. It is observed that for small stiffness requirements, the mass of the final design is almost constant or increases very little. After a certain point, the mass begins to increase at a considerable rate with the stiffness. It is found that the F/A stiffness requirement has larger effects on the mass than the I/O and torsion stiffness requirements. Thus, for the range of stiffness considered in this project, the mass of the optimum design is mainly determined by the F/A stiffness requirement. The I/O stiffness requirement has larger effects on the mass compared with the torsion stiffness requirement.

It is found that some design variables in an optimum design tended to reach their lower or upper bounds, and some others had values between the lower and upper bounds. The mass of the optimum design changes when we change the lower and upper bounds of these design variables. Figures 6.17 to 6.25 show the effects of the lower and upper bounds of some design variables on the mass of the optimum design. The measured stiffness of an actual joint was used as the stiffness targets in this study. The design variables are *thickness of front rocker, thickness of pillar back, pillar_base, outer_pillar_width, pillar_inner_length, door_edge_width, rocker_width,* and *outboard_cell_width.*

From these figures, we find that the mass of optimum design increases when we increase the lower bound or decrease the upper bound of these design variables because

these changes reduce the region in which the optimizer can search for an optimum design.

Thickness affects significantly the mass of the optimum design. Figures 6.17-6.18 show the relation between the lower bounds of thickness of the front rocker and pillar back on the mass. The thickness of pillar back of the optimum design contributes comparatively less to the mass while increasing its value significantly increases the stiffness. When we reduce its upper bound, the optimum design tends to increase the thickness of the front rocker to increase the overall stiffness of the joint.

Pillar_base is one of the most important design variables. For the optimum design, the value of *pillar_base* falls between its lower and upper bounds. Figures 6.19 and 6.20 show that the effects of the lower and upper bounds of *pillar_base* on the mass can be significant.

Figures 6.21-6.25 show the effects of the bounds of *outer_pillar_width, pillar_inner_length, door_edge_width, rocker_width,* and *outboard_cell_width* on the mass of the optimum design.

6.3.3 Discussion of Results

For the B-pillar to rocker joint, we developed the translator B using both the response surface polynomials and neural networks. Both MFD and SLP methods were tested. SLP was chosen as the default optimization algorithm because of its efficiency when applied to the problems considered in this project. The user can easily switch to the MFD method by changing a variable in the program for translator B.

From the comparison of the optimum and FEA results, we found that it is important to choose a combination of stiffness targets for which translator B is valid. For this purpose, we defined eight zones in which the stiffness requirements should fall.

The results of both NN and RSP translators correlated well with FEA results, except when the stiffness targets were very high (Figs. 6.11a to 6.12d). By comparing the

results obtained using RSP and NN, we found that the mass of the optimum design obtained using NN translators was typically about 10% higher than that obtained using RSP translators. This is because the NN tended to underestimate stiffness (Figs 6.12a-6.12c). The correlation coefficients between the FEA results and the optimization results obtained using RSP and NN translators were compared in Table 6.4c. We found that the RSP translator B predicted F/A stiffness and mass more accurately than the NN translator B. Predictions for I/O and torsion stiffness from both types of translators were equally accurate. Note that when we used the NN translator, we considered several design variables that do not affect stiffness and mass. More accurate results could be obtained if these variables were fixed. Even if the testing results from RSP translators seem to be slightly better than those from NN translators for the 12 design compared, it is difficult to conclude whether the RSP or the NN translator is better just from the 12 designs we compared. This is different from the Zhu's conclusion (1994), who found that the predictions of NN were slightly more accurate than those of RSP for a set of 29 randomly generated designs. This is because he used a completed second degree polynomial to create the RSP translator that should include some statistically unimportant terms. As a result, Zhu's RSP translator performed more poorly than a translator developed using stepwise regression.

Some observations about the effects of the important design variables should help designers to improve the joint stiffness, and design more efficient joints. For example, this study found that the value of *pillar_base* of the optimum design increases as the stiffness requirements increase. This shows that the optimum design tends to make the joint connection between the rocker and the B-pillar as large as possible, which agrees with rule of thumb presented in internal reports of an automotive manufacturer.

In most optimum designs, the *bottom_pillar_hole* and the *fwd_pillar_hole* tended to reach their upper bounds. This made the pillar hole of the optimum design as small as possible. This is also consistent with rules of thumb presented in an internal report.

It was found that *pillar_reinf_base_width* and *pillar_reinf_expansion* did not have big effects on the stiffness of the joint. They assumed values equal to their lower bounds.

We found that *rocker_height* and *outboard_cell_width* reached their lower bounds. This is different from Zhu's conclusion (1994). In his study, he found that rocker section tended to take the maximum allowable height for the optimum design. This may be caused by the difference of parametric model and constraints between this study and that of Zhu's. *Rocker_width* and *inner_flange_width* took values between their lower and upper bounds. *A1*, *A3*, and *A8* reached their lower bounds, and *A2* and *A7* were close to their upper bounds. The front part of rocker tended to become vertical, and the bottom part of rocker tended to become horizontal. This trend increased the area of the rocker cross section and the stiffness of the joint.

This study found that when the I/O and F/A stiffness requirements were low, the RSP translator B tended to overestimate the stiffness of the optimum design compared to FEA results. When the stiffness requirements were high, the translator B tended to underestimate stiffnesses (Figs. 6.11a-6.11c). The RSP translator consistently underestimated the mass of the optimum designs (Fig. 6.11d). On the other hand, in general, the NN translator B underestimated stiffnesses. For the 12 designs studied, the NN translator B tended to underestimate F/A, and torsion stiffness compared to the RSP translator B. The NN translator also underestimated the mass of a few designs, but its bias was smaller than that of the RSP translator.

Zhu (1994) suggested increasing the minimum allowable stiffness by 10% to ensure translator B yields designs whose stiffness is equal to or exceeds the given target values. In this study, we did not notice the phenomenon that translator B consistently overestimates the design. Therefore, we do not suggest to slightly increase the stiffness targets for the optimum designs.

Table 6.1: Ranges and States of Design Variables

Seq. No	Name of Design Variables	Bounds (mm, degree)		States
		Lower	Upper	
1	Length_of_flange	19	19	# Fixed by user
2	Spot_weld_spacing	47.5	47.5	# Fixed by user
3	Spot_weld_placement	9.5	9.5	# Fixed by user
4	Pillar_base	157	215	Independent
5	Pillar_angle	90	90	# Fixed by user
6	Pillar_io_angle	90	90	# Fixed by user
7	Pillar_height	210	250	## Dependent
8	Pillar_location	150	150	# Fixed by user
9	Outer_pillar_width	50	83	Independent
10	Inner_pillar_width	6	25	Independent
11	Pillar_outer_length	77	110	Independent
12	Pillar_inner_length	122	160	Independent
13	Fwd_inner_ver_blending_rad	120	155	Independent
14	Fwd_inner_hor_blending_rad	80	125	Independent
15	Fwd_outer_ver_blending_rad	120	155	## Dependent
16	Fwd_inner_hor_blending_rad	80	125	## Dependent
17	Aft_inner_ver_blending_rad	95	133	Independent
18	Aft_inner_hor_blending_rad	100	135	Independent
19	Aft_outer_ver_blending_rad	95	140	## Dependent
20	Aft_outer_hor_blending_rad	100	135	## Dependent
21	Inner_pillar_base_width	2	15	Independent
22	Rocker_length	485	485	## Fixed by program
23	Pillar_reinforcement_depth	10	100	## Dependent
24	Pillar_reinf_base_width	20	60	Independent
25	Pillar_reinf_expansion	2	15	Independent
26	A1	83	90	Independent
27	A2	65	78	Independent
28	A3	10	36	Independent
29	A5	75	87	Independent
30	A7	70	86	Independent
31	A8	80	90	Independent
32	Rocker_height	110	120	Independent
33	Inner_flange_distance	18	45	Independent
34	Inner_rocker_height	90	110	Independent
35	Rocker_width	115	147	Independent
36	Outboar_cell_width	65	95	Independent
37	Door_edge_height	6	38	Independent
38	Door_edge_width	7	19	Independent

39	Low_door_ht_minus_clearance	60	75	Independent
40	Fwd_bulk_head_position	470	470	Not used
41	Aft_bulk_head_position	10	10	Not used
42	Top_pillar_hole	22	80	Independent
43	Bottom_pillar_hole	28	50	Independent
44	Fwd_pillar_hole	17	30	Independent
45	Aft_pillar_hole	15	15	## Fixed by user
46	Thickness of frontrock	0.71	1.27	Independent
47	Thickness of pillar_reinf	0.71	1.52	Independent
48	Thickness of pillarback	0.89	1.27	Independent
49	Thickness of backrock	1.27	1.98	Independent
50	Thickness of centerplate	0.71	1.27	Independent
51	Thickness of pillar_bridge	1.0	1.0	Not used
52	Thickness of bulkheads	1.4	1.78	Not used

& #, excluded from optimization
Length of B-pillar branch (Fig. 6.1) is fixed at 269 mm.

Table 6.2: Comparison of Optimum Results
When Starting From Different Initial Points
($K_{I/O} > 4.3899E7Nmm, K_{F/A} > 5.2297E8Nmm, K_{Tor} > 7.9788E7Nmm$)

Seq. No	Use Polynomial Translators		Use Neural Network Translators	
	No. of Function Evaluations	Object Function (kg)	No. of Function Evaluations	Objective Function (kg)
1	527	4.54870	740	4.97923
2	527	4.54962	740	4.97973
3	527	4.54821	705	4.98170
4	492	4.54730	635	4.98150
5	492	4.54704	600	4.97854
6	492	4.54812	706	4.97915
7	527	4.55183	705	4.98036
8	528	4.55183	673	4.97942

Table 6.3: Zones of Database (According to the Combination of Stiffness)

Zone No.	Range of $K_{I/O}$ ($\times 10^7$ Nmm)	Range of $K_{F/A}$ ($\times 10^8$ Nmm))	Range of K_{Tor} ($\times 10^7$ Nmm)
1	1.0~1.5	1.8~4.0	2.5~8.0
2	1.5~2.0	2.0~4.5	3.0~10.0
3	2.0~3.0	2.0~5.5	3.5~12.5
4	3.0~4.0	2.0~5.5	4.0~13.5
5	4.0~5.0	2.5~5.5	4.0~14.5
6	5.0~6.0	2.5~5.0	5.0~13.0
7	6.0~7.0	3.5~5.0	7.5~13.0
8	7.0~8.0	3.5~5.0	9.0~13.0

Table 6.4a: Comparison of Optimization Results and FEA Results for B-pillar to Rocker Joint (using RSP Translators)

Design No	Stiffness Requirement (Nmm)			Optimization Results (Nmm, kg)				FEA Results (Nmm, kg)			
	I/O $\times 10^7$	F/A $\times 10^8$	Tor $\times 10^7$	I/O $\times 10^7$	F/A $\times 10^8$	Torsion $\times 10^7$	Mass	I/O $\times 10^7$	F/A $\times 10^8$	Torsion $\times 10^7$	Mass
1	1.25	2.5	5.0	1.4175	2.5000	4.9996	3.4099	1.0990	2.3485	4.1962	3.6599
2	1.75	3.0	6.5	1.7504	3.0001	6.5001	3.5445	1.5428	2.8517	6.0755	3.7395
3	2.5	3.5	8.0	2.5003	3.5002	8.0007	3.6865	2.0414	3.0754	9.0905	3.8463
4	3.3	3.5	8.5	3.3002	3.5000	8.5000	3.7138	2.5715	3.2920	9.6463	3.8601
5	3.6	4.5	10.	3.6000	4.5000	10.001	3.9397	3.1401	4.3361	11.245	4.0734
6	4.3	4.0	9.0	4.3000	4.0000	9.0000	3.8364	4.4447	4.2336	10.483	3.9484
7	4.6	5.0	10.	4.6000	5.0000	12.168	4.3960	5.1175	5.4079	14.206	4.5069
8	5.3	3.5	9.0	5.3005	3.4999	9.0004	3.8585	5.1304	3.4553	9.2918	3.9485
9	5.6	4.0	10.	5.6002	4.0000	10.001	3.9898	5.2545	3.7779	11.264	4.0638
10	6.3	4.0	10.	6.3000	4.0000	10.000	4.0835	7.2950	4.3145	10.337	4.2581
11	6.6	4.5	11.	6.6000	4.5000	14.260	4.3461	8.5703	4.9361	15.774	4.4874
12	7.5	4.5	11.	7.4996	4.5000	15.379	4.4949	9.9149	4.9618	16.824	4.6146

Table 6.4b: Comparison of Optimization Results and FEA Results for B-pillar to Rocker Joint (using NN Translators)

Design No	Stiffness Requirement (Nmm)			Optimization Results (Nmm, kg)				FEA Results (Nmm, kg)			
	I/O $\times 10^7$	F/A $\times 10^7$	Tor $\times 10^7$	I/O $\times 10^7$	F/A $\times 10^8$	Torsion $\times 10^7$	Mass	I/O $\times 10^7$	F/A $\times 10^8$	Torsion $\times 10^7$	Mass
1	1.25	2.5	5.0	2.9688	3.2061	5.6278	3.9192	3.2403	3.5276	6.9522	3.8604
2	1.75	3.0	6.5	3.7676	3.3338	6.5004	3.9235	5.1058	4.4350	8.3798	3.9105
3	2.5	3.5	8.0	3.8915	3.5000	7.9997	3.9540	5.1183	4.4917	9.8755	3.9556
4	3.3	3.5	8.5	4.0418	3.5000	8.4997	3.9676	5.1352	4.5034	10.284	3.9627
5	3.6	4.5	10.	3.6807	4.4998	11.995	4.4386	3.4885	5.0269	13.593	4.4863
6	4.3	4.0	9.0	4.3000	4.0000	11.392	4.1179	5.0931	5.1805	12.860	4.1348
7	4.6	5.0	10.	4.6001	5.0001	13.186	4.8121	4.8263	5.7996	15.905	4.8888
8	5.3	3.5	9.0	5.3004	3.5000	8.9996	3.9917	6.9335	4.1704	11.130	4.1492
9	5.6	4.0	10.	5.6001	4.0000	12.116	4.1656	6.7701	4.6281	14.047	4.2264
10	6.3	4.0	10.	6.3001	4.0000	12.644	4.2252	7.8590	4.5805	15.256	4.3893
11	6.6	4.5	11.	6.6000	4.5000	14.216	4.5564	9.0910	5.2361	18.610	4.7582
12	7.5	4.5	11.	7.5000	4.4999	14.561	4.7961	9.6724	4.8660	18.543	4.8464

Table 6.4c: Comparison of Correlation Coefficients Obtained using RSP and NN translators

Stiffness/Mass	RSP translators	NN translator
I/O	0.9733	0.9703
F/A	0.9677	0.8809
Torsion	0.9872	0.9831
Mass	0.9920	0.9810

Table 6.5a: States of Design Variables and Optimum Designs for Car A

No	Name of Design Variables	State	Car A (mm, degree)	
			RSP	NN
1	Length_of_flange	# Fixed by user	19.00	19.00
2	Spot_weld_spacing	# Fixed by user	47.50	47.50
3	Spot_weld_placement	# Fixed by user	9.50	9.50
4	Pillar_base	Independent	185.15	185.00
5	Pillar_angle	# Fixed by user	90.00	90.00
6	Pillar_io_angle	# Fixed by user	90.00	90.00
7	Pillar_height	## Dependent	218.45	218.37
8	Pillar_location	# Fixed by user	150.00	150.00
9	Outer_pillar_width	Independent	82.99	83.00
10	Inner_pillar_width	Independent	6.00	23.00
11	Pillar_outer_length	Independent	109.99	110.00
12	Pillar_inner_length	Independent	159.84	159.99
13	Fwd_inner_ver_blending_rad	Independent	120.00	134.99
14	Fwd_inner_hor_blending_rad	Independent	80.00	80.00
15	Fwd_outer_ver_blending_rad	## Dependent	120.00	134.99
16	Fwd_inner_hor_blending_rad	## Dependent	80.00	80.00
17	Aft_inner_ver_blending_rad	Independent	132.99	133.00
18	Aft_inner_hor_blending_rad	Independent	135.00	100.00
19	Aft_outer_ver_blending_rad	## Dependent	132.99	133.00
20	Aft_outer_hor_blending_rad	## Dependent	135.00	100.00
21	Inner_pillar_base_width	Independent	15.00	2.32
22	Rocker_length	## Fixed by program	485.00	485.00
23	Pillar_reinforcement_depth	## Dependent	38.16	46.82
24	Pillar_reinf_base_width	Independent	20.00	20.00
25	Pillar_reinf_expansion	Independent	2.00	2.00
26	A1	Independent	83.00	83.00
27	A2	Independent	77.94	77.99
28	A3	Independent	10.00	10.00
29	A5	Independent	75.00	75.00
30	A7	Independent	85.95	85.95
31	A8	Independent	80.03	80.01
32	Rocker_height	Independent	110.01	110.01
33	Inner_flange_distance	Independent	37.57	36.94
34	Inner_rocker_height	Independent	106.16	106.19
35	Rocker_width	Independent	120.17	119.75
36	Outboar_cell_width	Independent	67.83	68.03
37	Door_edge_height	Independent	35.66	27.07
38	Door_edge_width	Independent	7.00	7.00
39	Low_door_ht_minus_clearance	Independent	60.05	60.00

40	Fwd_bulk_head_position	Not used		
41	Aft_bulk_head_position	Not used		
42	Top_pillar_hole	Independent	80.00	80.00
43	Bottom_pillar_hole	Independent	49.99	50.00
44	Fwd_pillar_hole	Independent	30.00	30.00
45	Aft_pillar_hole	## Fixed by user	15.00	15.00
46	Thickness of frontrock	Independent	1.14	1.27
47	Thickness of pillar_reinf	Independent	0.71	0.71
48	Thickness of pillarback	Independent	1.27	1.27
49	Thickness of backrock	Independent	1.27	1.39
50	Thickness of centerplate	Independent	0.71	0.71
51	Thickness of pillar_bridge	Not used		
52	Thickness of bulkheads	Not used		

Table 6.5b: States of Design variables and Optimum Designs for Car B

No	Name of Design Variables	State	Car B (mm, degree)	
			RSP	NN
1	Length_of_flange	# Fixed by user	19.00	19.00
2	Spot_weld_spacing	# Fixed by user	47.50	47.50
3	Spot_weld_placement	# Fixed by user	9.50	9.50
4	Pillar_base	Independent	157.15	160.24
5	Pillar_angle	# Fixed by user	90.00	90.00
6	Pillar_io_angle	# Fixed by user	90.00	90.00
7	Pillar_height	## Dependent	218.72	217.62
8	Pillar_location	# Fixed by user	150.00	150.00
9	Outer_pillar_width	Independent	79.34	50.00
10	Inner_pillar_width	Independent	7.54	24.99
11	Pillar_outer_length	Independent	109.93	109.96
12	Pillar_inner_length	Independent	122.07	159.89
13	Fwd_inner_ver_blending_rad	Independent	120.01	155.00
14	Fwd_inner_hor_blending_rad	Independent	125.00	80.03
15	Fwd_outer_ver_blending_rad	## Dependent	120.01	155.00
16	Fwd_inner_hor_blending_rad	## Dependent	125.00	80.03
17	Aft_inner_ver_blending_rad	Independent	113.06	133.00
18	Aft_inner_hor_blending_rad	Independent	135.00	100.00
19	Aft_outer_ver_blending_rad	## Dependent	113.06	133.00
20	Aft_outer_hor_blending_rad	## Dependent	135.00	100.00
21	Inner_pillar_base_width	Independent	14.99	14.99
22	Rocker_length	## Fixed by program	485.00	485.00
23	Pillar_reinforcement_depth	## Dependent	43.17	47.88
24	Pillar_reinf_base_width	Independent	20.00	25.00
25	Pillar_reinf_expansion	Independent	2.00	10.00
26	A1	Independent	83.00	83.00
27	A2	Independent	77.94	78.00
28	A3	Independent	10.00	10.00
29	A5	Independent	75.00	80.12
30	A7	Independent	81.21	85.98
31	A8	Independent	80.04	80.00
32	Rocker_height	Independent	110.05	110.01
33	Inner_flange_distance	Independent	40.28	29.93
34	Inner_rocker_height	Independent	102.59	105.81
35	Rocker_width	Independent	120.10	119.64
36	Outboar_cell_width	Independent	65.05	74.93
37	Door_edge_height	Independent	30.73	26.65
38	Door_edge_width	Independent	7.00	7.00
39	Low_door_ht_minus_clearance	Independent	60.05	60.01
40	Fwd_bulk_head_position	Not used	470.00	470.00

41	Aft_bulk_head_position	Not used	10.00	10.00
42	Top_pillar_hole	Independent	22.00	79.94
43	Bottom_pillar_hole	Independent	49.96	47.60
44	Fwd_pillar_hole	Independent	30.00	29.97
45	Aft_pillar_hole	## Fixed by user	15.00	15.00
46	Thickness of frontrock	Independent	0.71	0.71
47	Thickness of pillar_reinf	Independent	0.71	0.71
48	Thickness of pillarback	Independent	0.89	0.89
49	Thickness of backrock	Independent	1.27	1.27
50	Thickness of centerplate	Independent	0.71	0.71
51	Thickness of pillar_bridge	Not used	1.00	1.00
52	Thickness of bulkheads	Not used	1.78	1.78

Table 6.5c: Comparison of FEA Results and Results from Translator B for B-pillar to Rocker Joint

Cars/ Stiff. Requirements (Nmm)	Stiffness / Mass	RSP Translators (Nmm, kg)		NN Translators (Nmm, kg)	
		FEA	Translator (err)	FEA	Translator (err)
Car A I/O>4.76E7 F/A >5.684E8 Tor >7.649E7	I/O	5.4527E7	4.7600E7 (-12.7%)	4.5713E7	4.7602E7 (4.1%)
	F/A	6.2650E8	5.6840E8 (-9.3%)	6.6915E8	5.6840E8 (-15.1%)
	Torsion	1.6285E8	1.3799E8 (-15.3%)	1.7093E8	1.3924E8 (-18.5%)
	Mass	5.0450	4.9475 (-1.9%)	5.4505	5.3570 (-1.7%)
Car B I/O>2.567E7 F/A>2.527E8 Tor > 6.521E7)	I/O	1.9929E7	2.5674E7 (28.8%)	4.0275E7	3.6564E7 (-9.2%)
	F/A	2.4668E8	2.5270E8 (2.4%)	3.6429E8	3.2990E8 (-9.4%)
	Torsion	5.7217E7	6.5211E7 (14.0%)	8.6064E7	6.5210E7 (-24.2%)
	Mass	3.7179	3.5204 (-5.3%)	3.9322	3.9231 (-0.2%)

Table 6.5d: Comparison of Mass of the Initial and Optimum Designs for Car A and Car B

Cars	Initial Design (kg)	Optimum Design (kg) and Percentage Improvement	
		RSP	NN
Car A	5.9312	5.0450 (14.9%)	5.4505 (8.1%)
Car B	5.1268	3.7179 (27.5%)	3.9322 (23.3%)

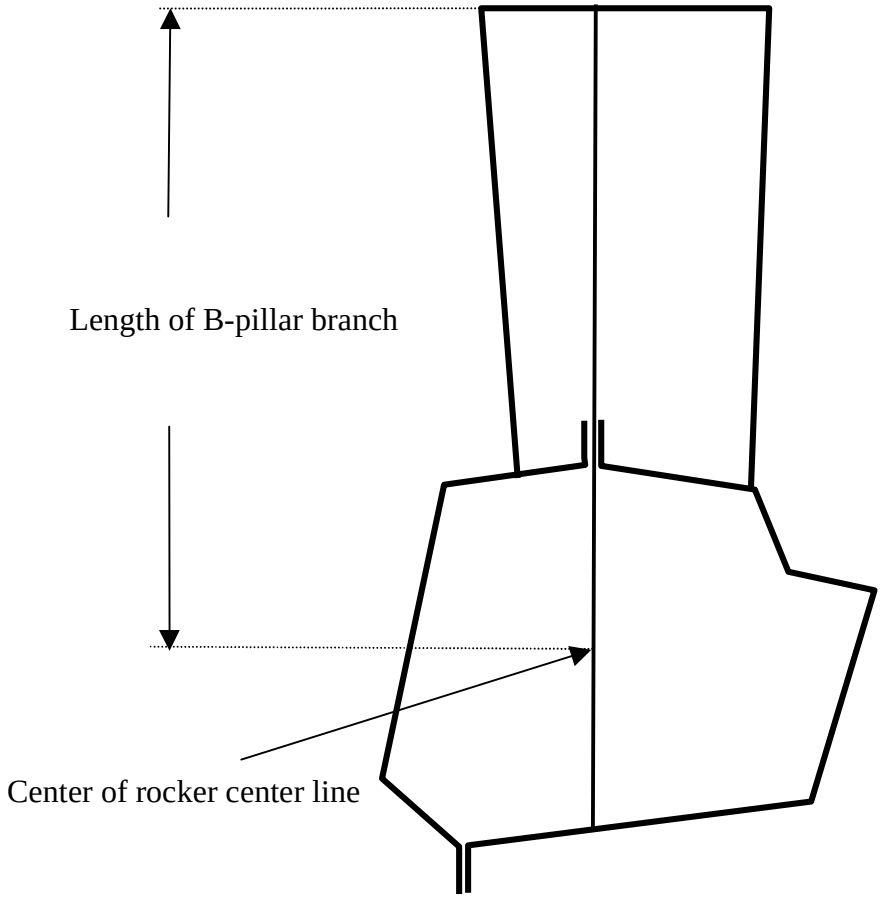


Figure 6.1: Length of B-pillar Branch

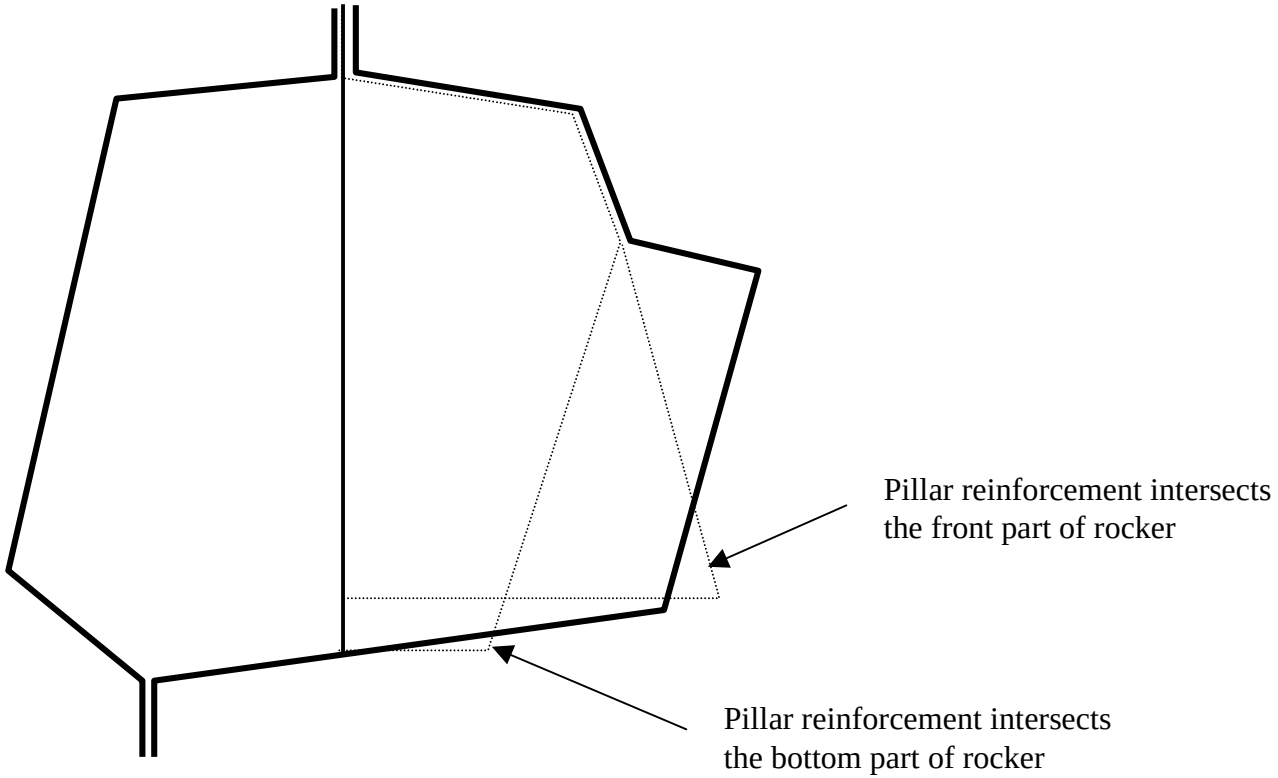


Figure 6.2: Constraints on the Pillar Reinforcement

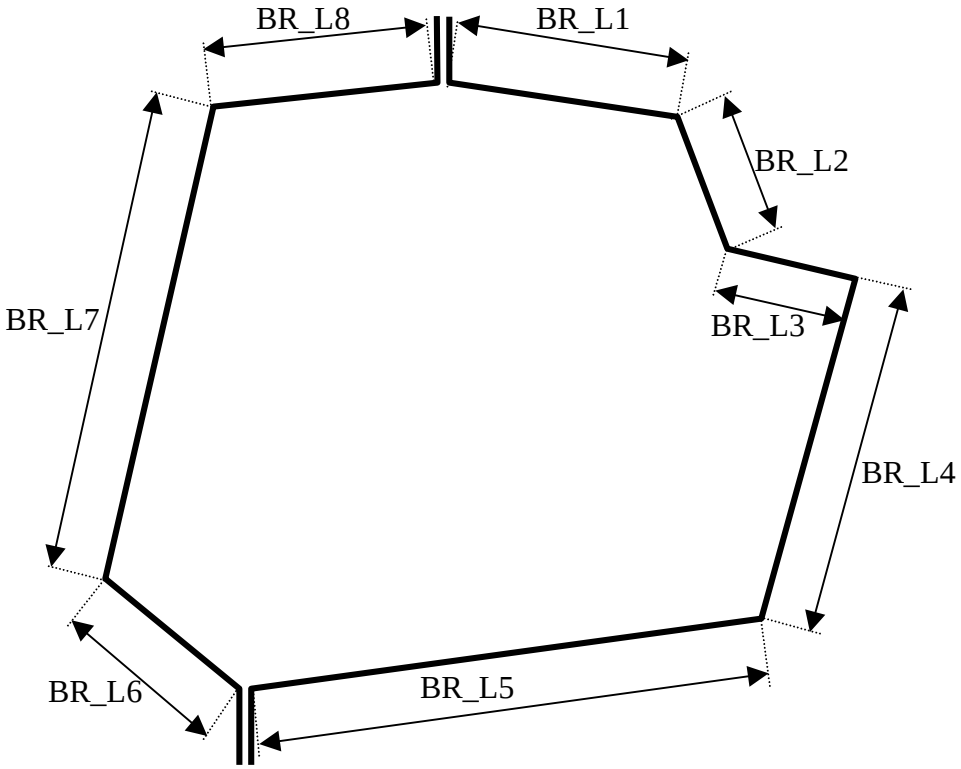


Figure 6.3: Manufacturing Constraints on the Length of Rocker Cross Section

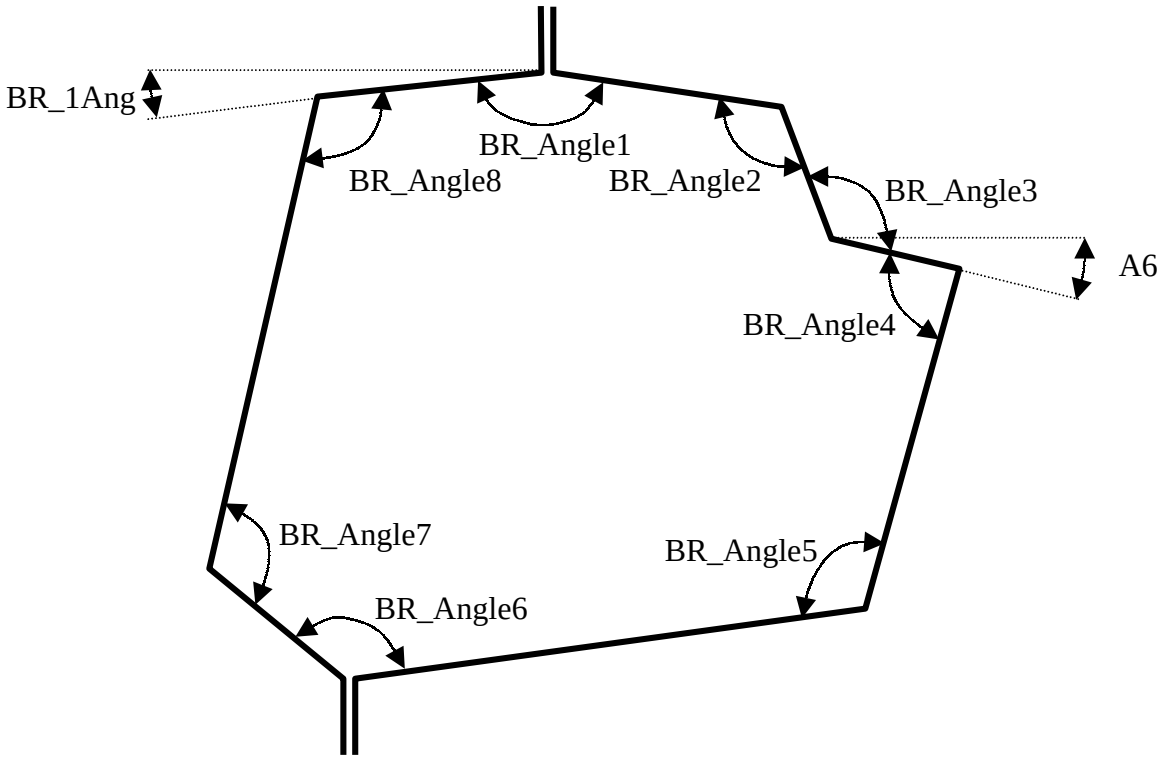


Figure 6.4: Manufacturing Constraints on Angles of Rocker Cross Section

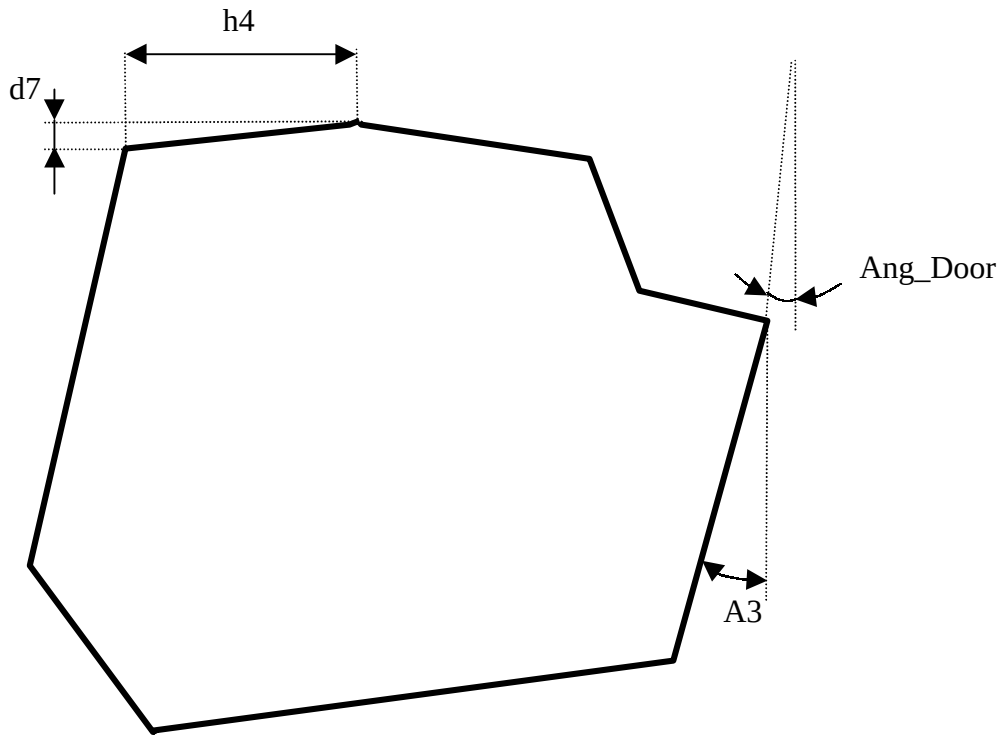


Figure 6.5: Constraints on the Rear Plate of Rocker and Angle of Outer Rocker Shell

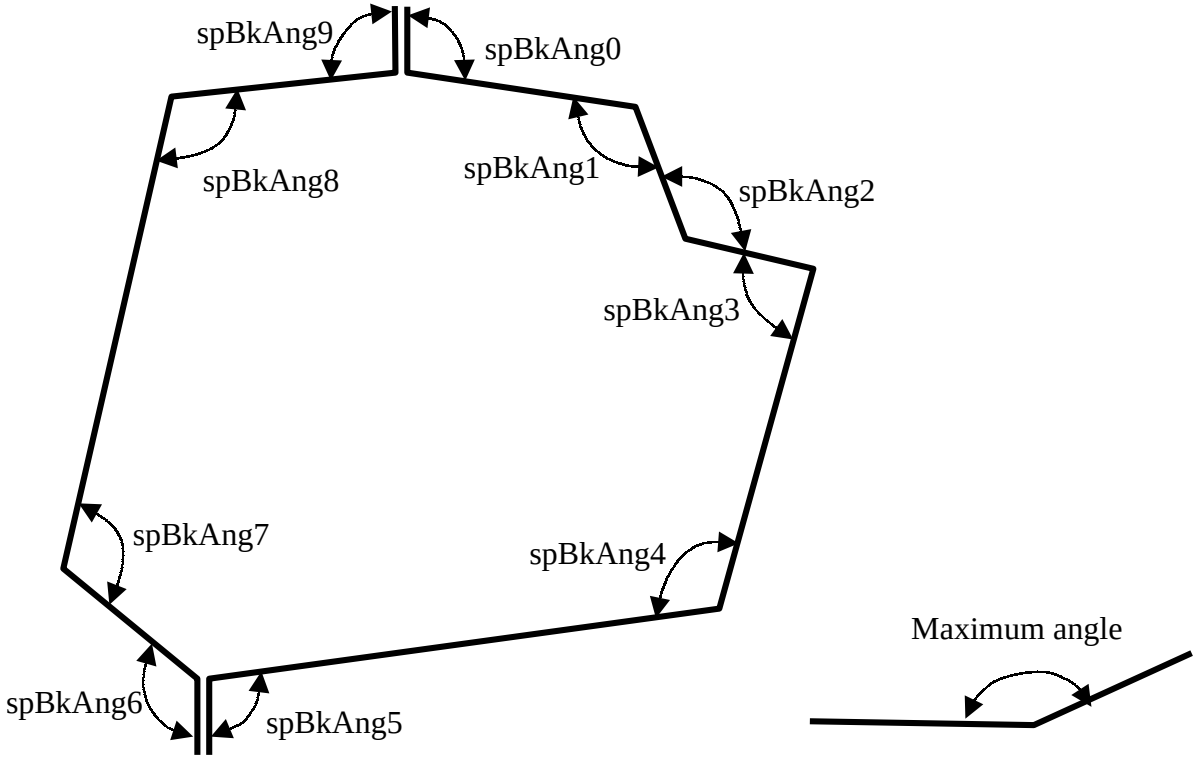


Figure 6.6: Spring Back Angles on Rocker Cross Section

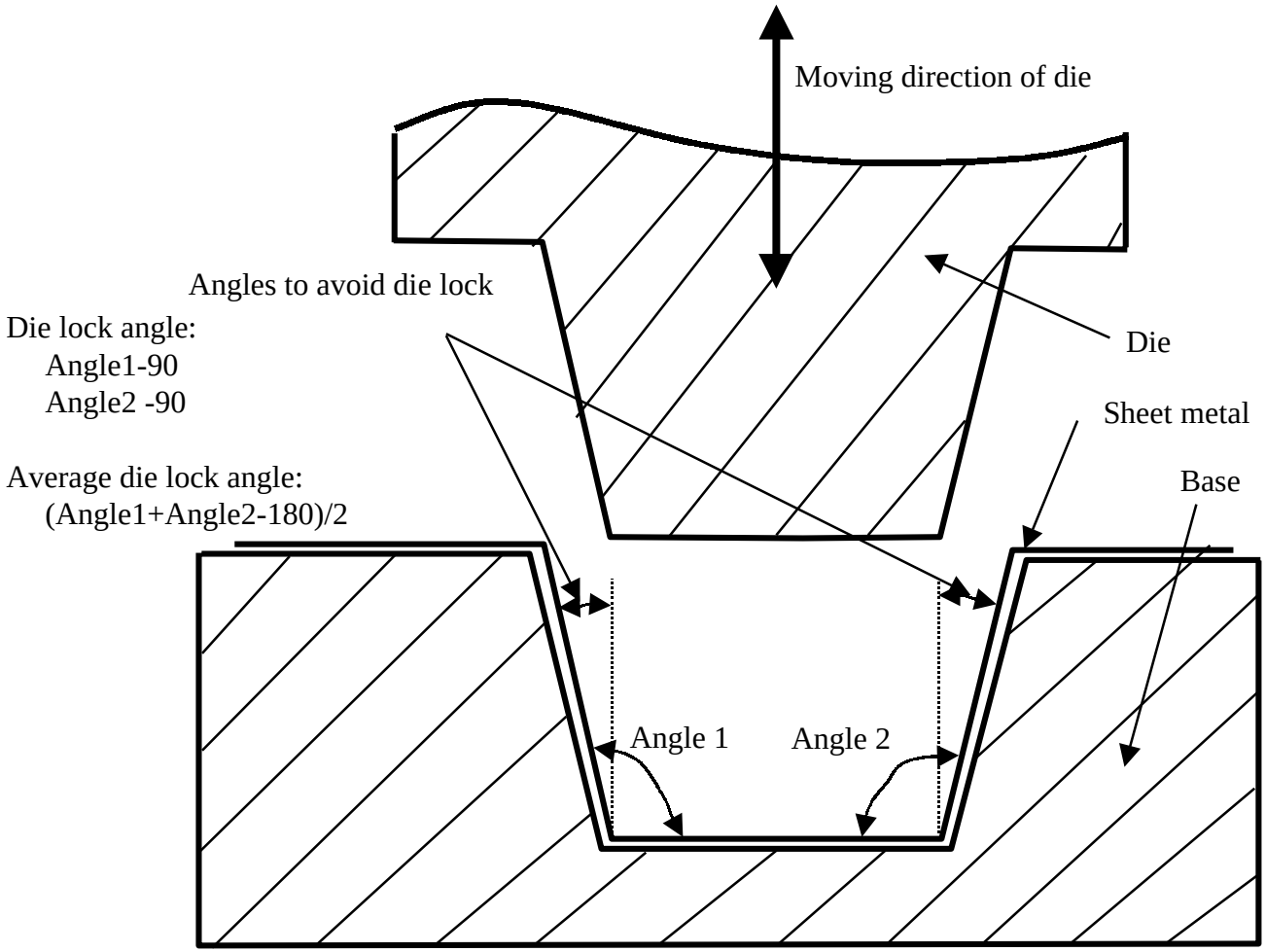


Figure 6.7a: Explanation of Die Lock Angle

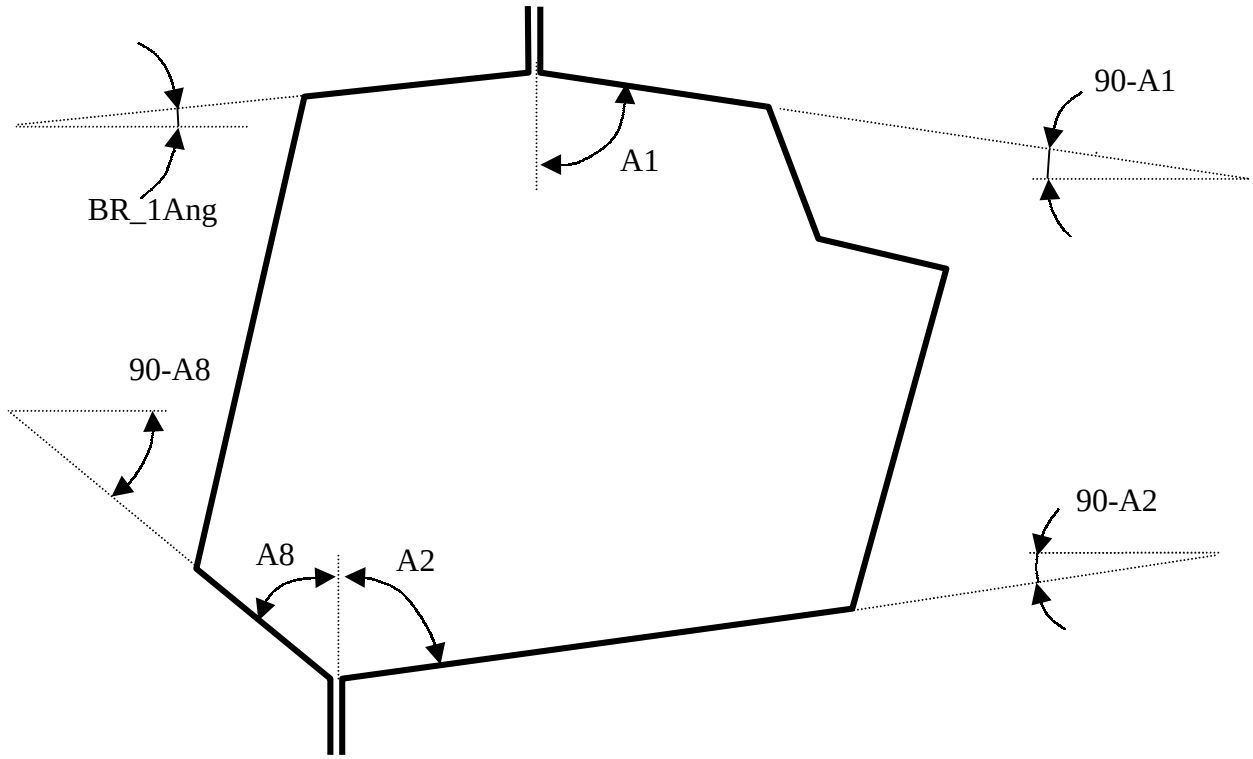


Figure 6.7b: Die Lock Angles on the Rocker Cross Section

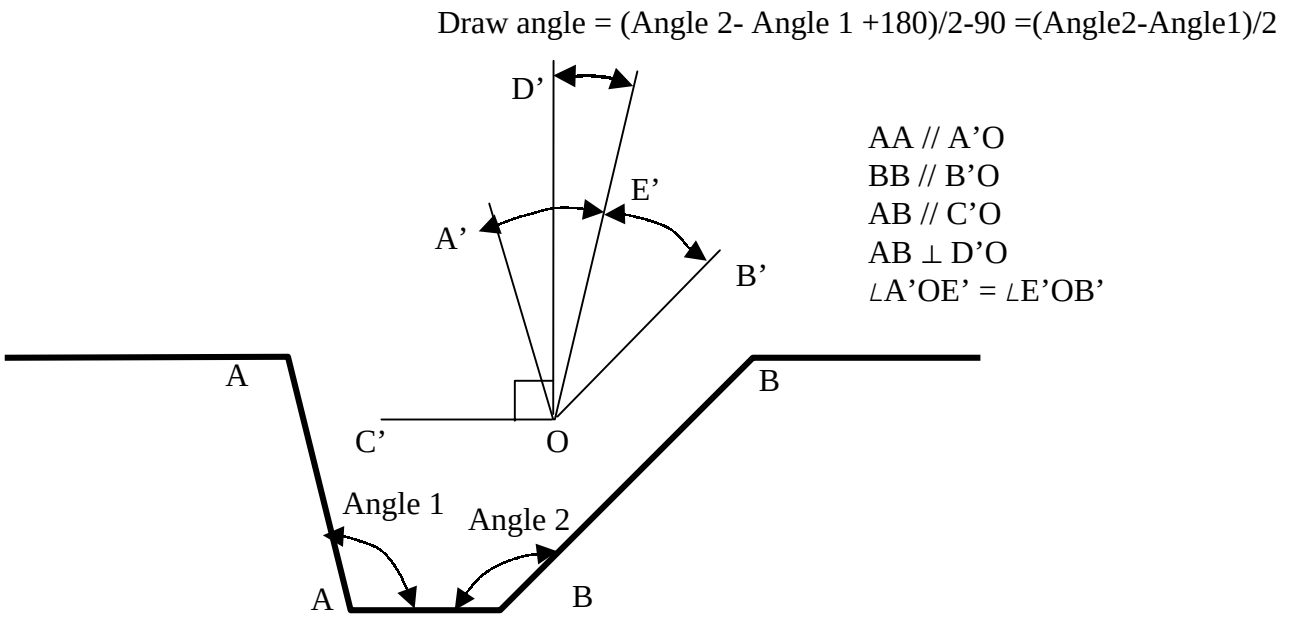


Figure 6.8a: Explanation of Draw Angle

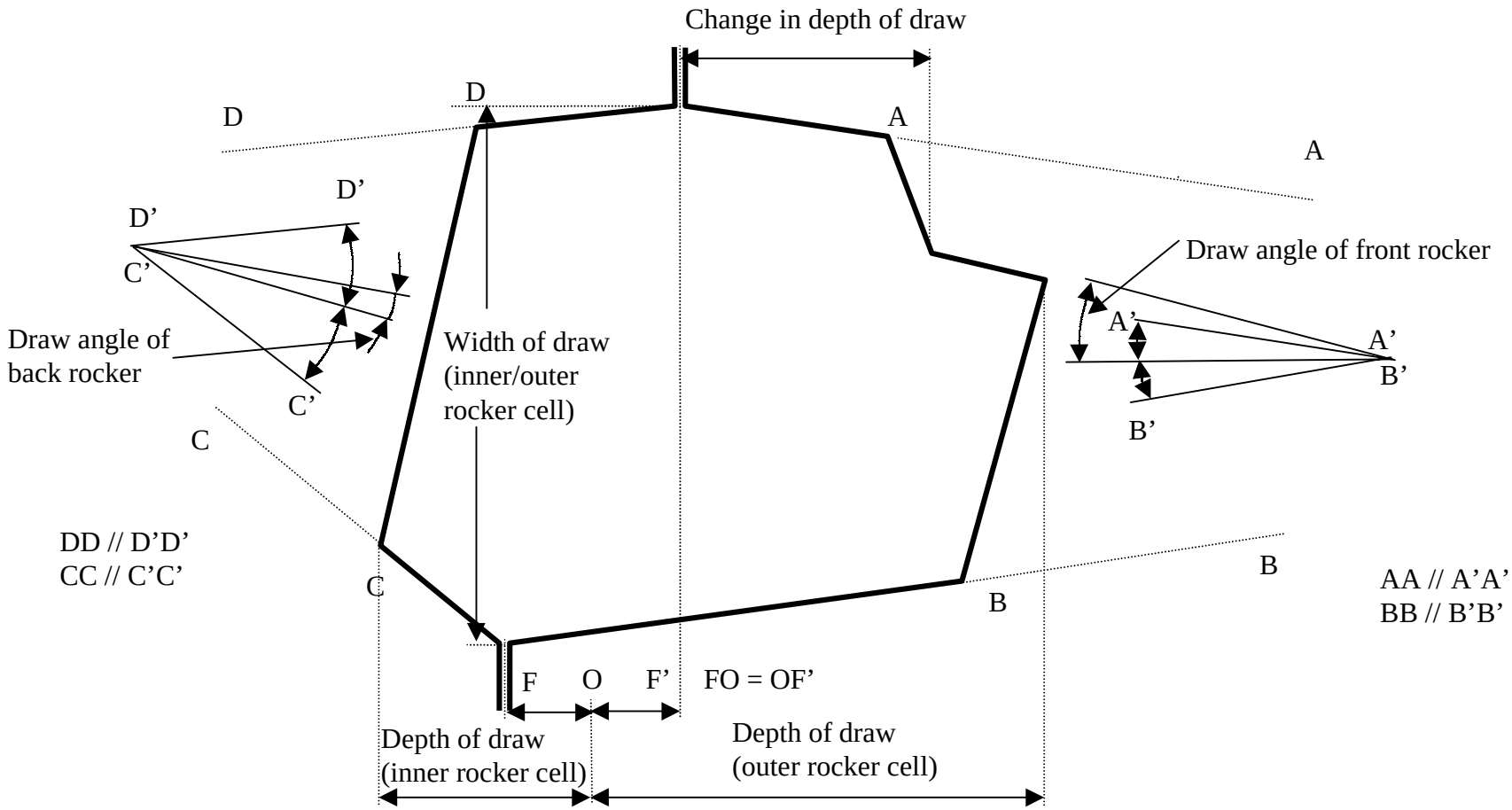


Figure 6.8b: Constraints on Draw Angle and Draw Steps

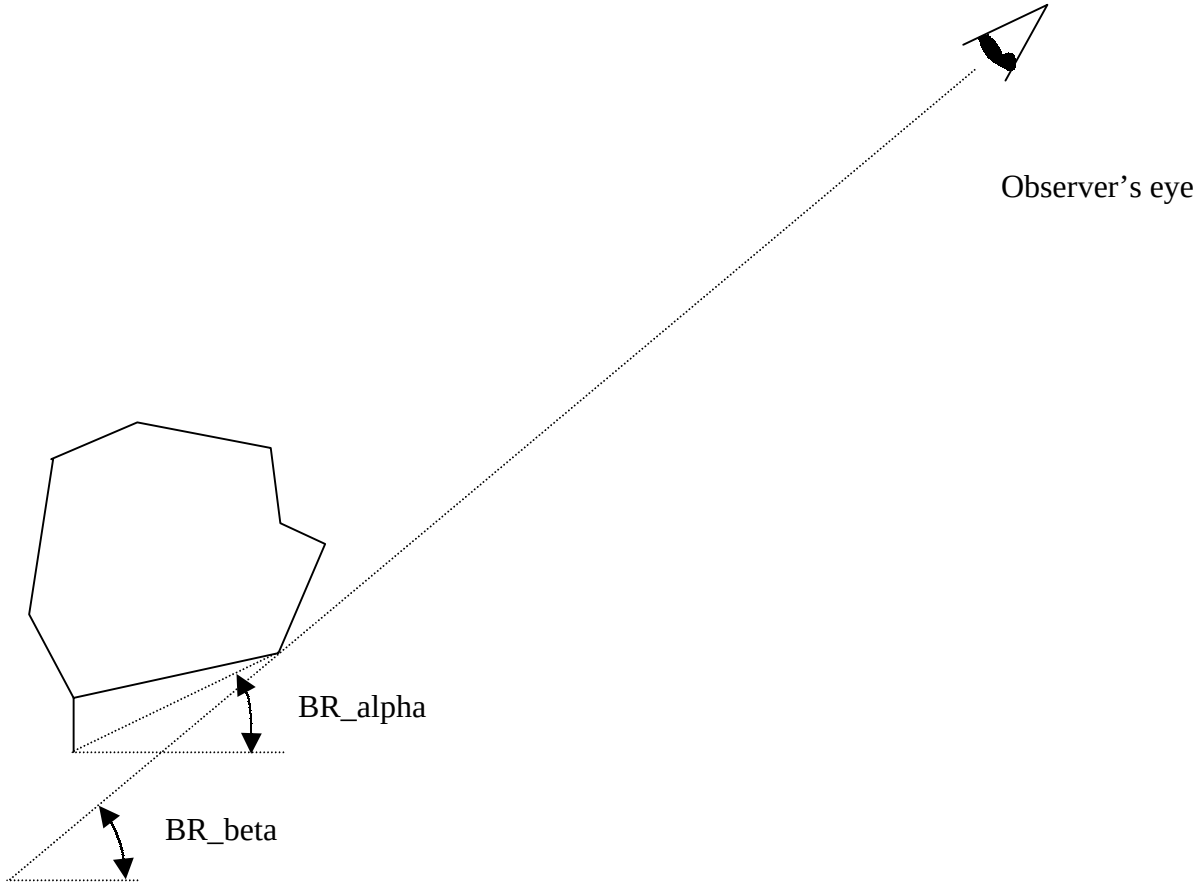


Figure 6.9: Constraint on the Cross Section of Rocker

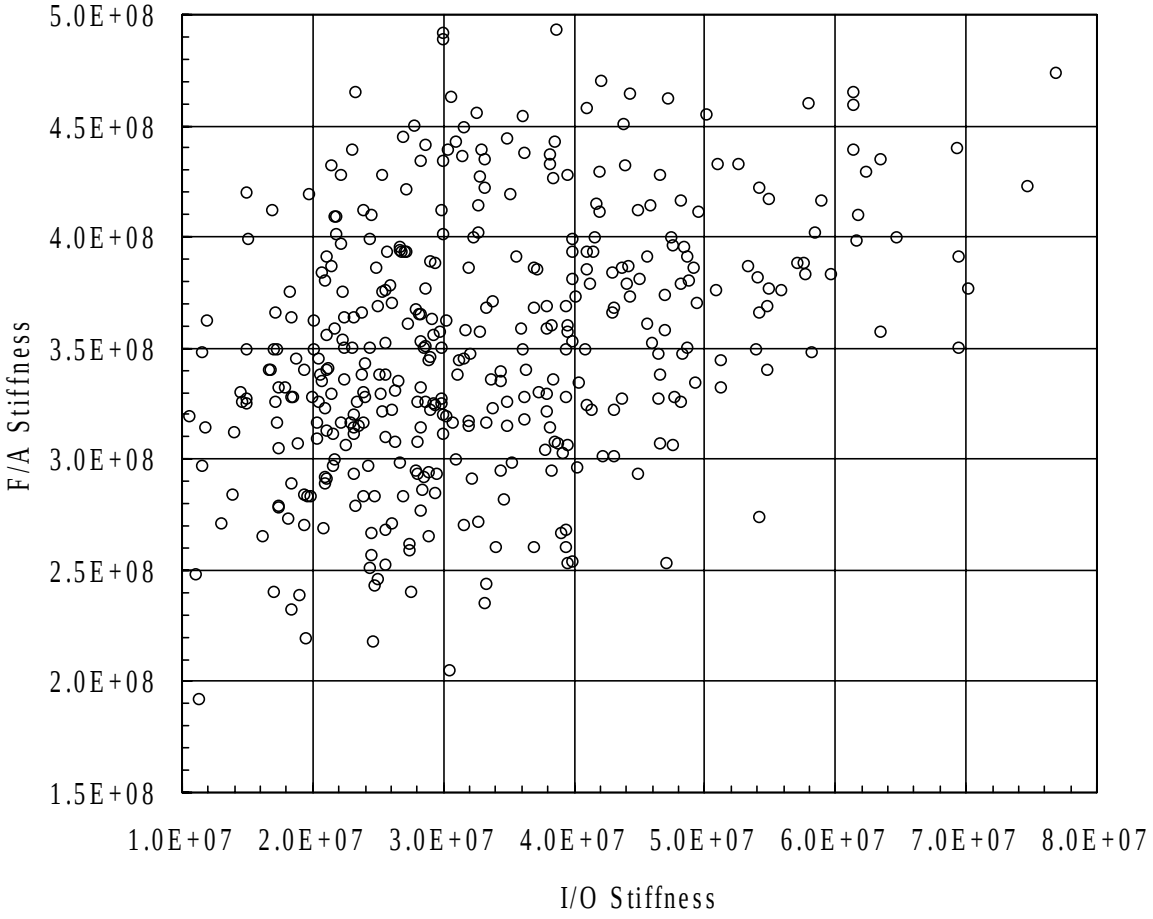


Figure 6.10a: Correlation Between I/O and F/A Stiffness
for Designs in the Database

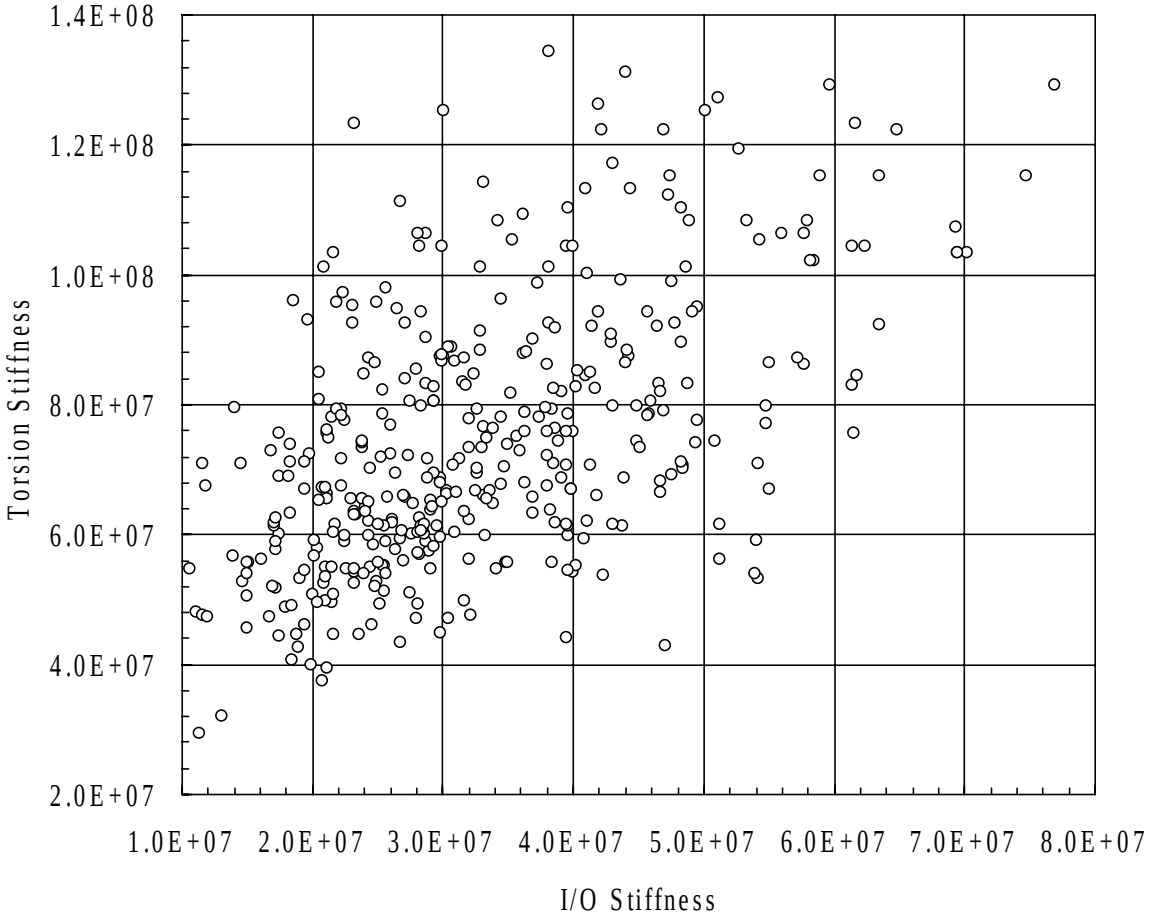


Figure 6.10b: Correlation Between I/O and Torsion Stiffness for Designs in the Database

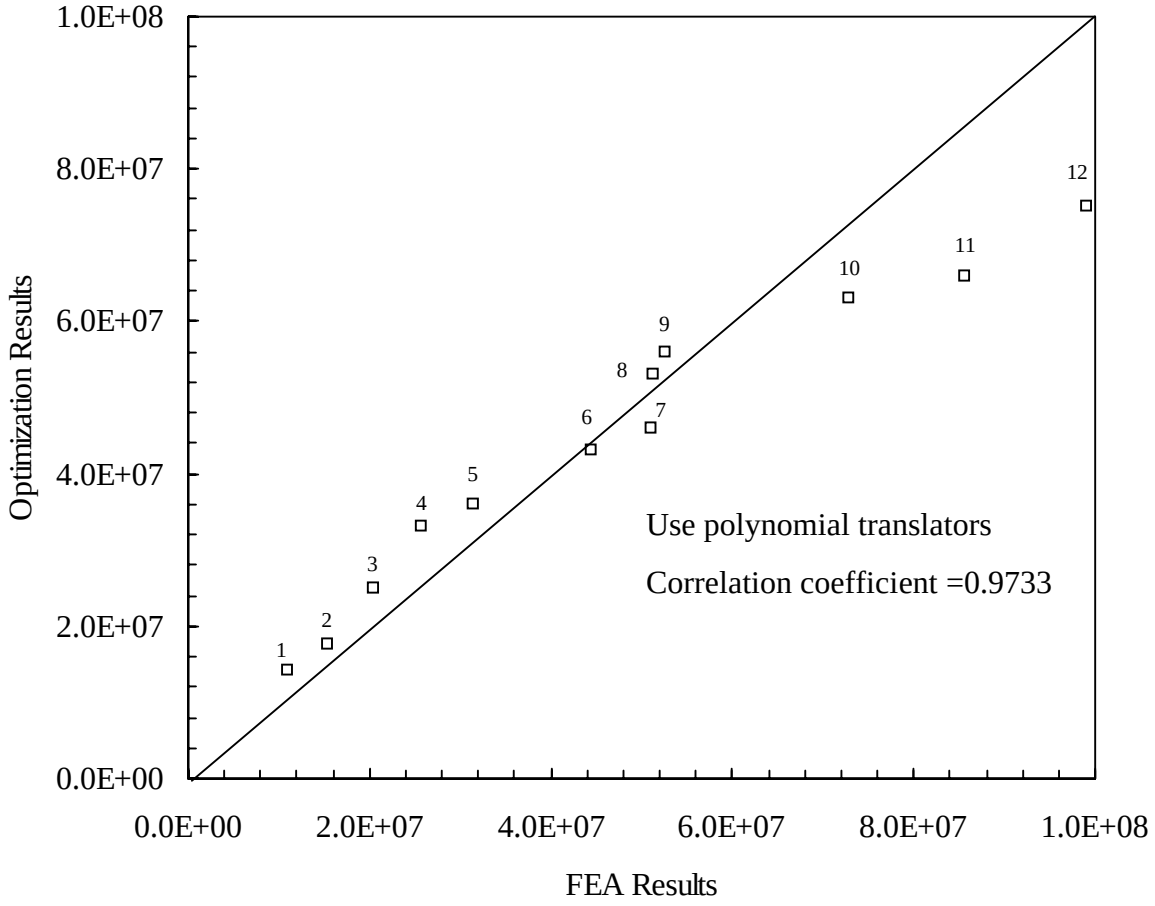


Figure 6.11a: Comparison of FEA Results and the Optimization Results Obtained using RSP Translators for the I/O Stiffness of B-pillar to Rocker Joint

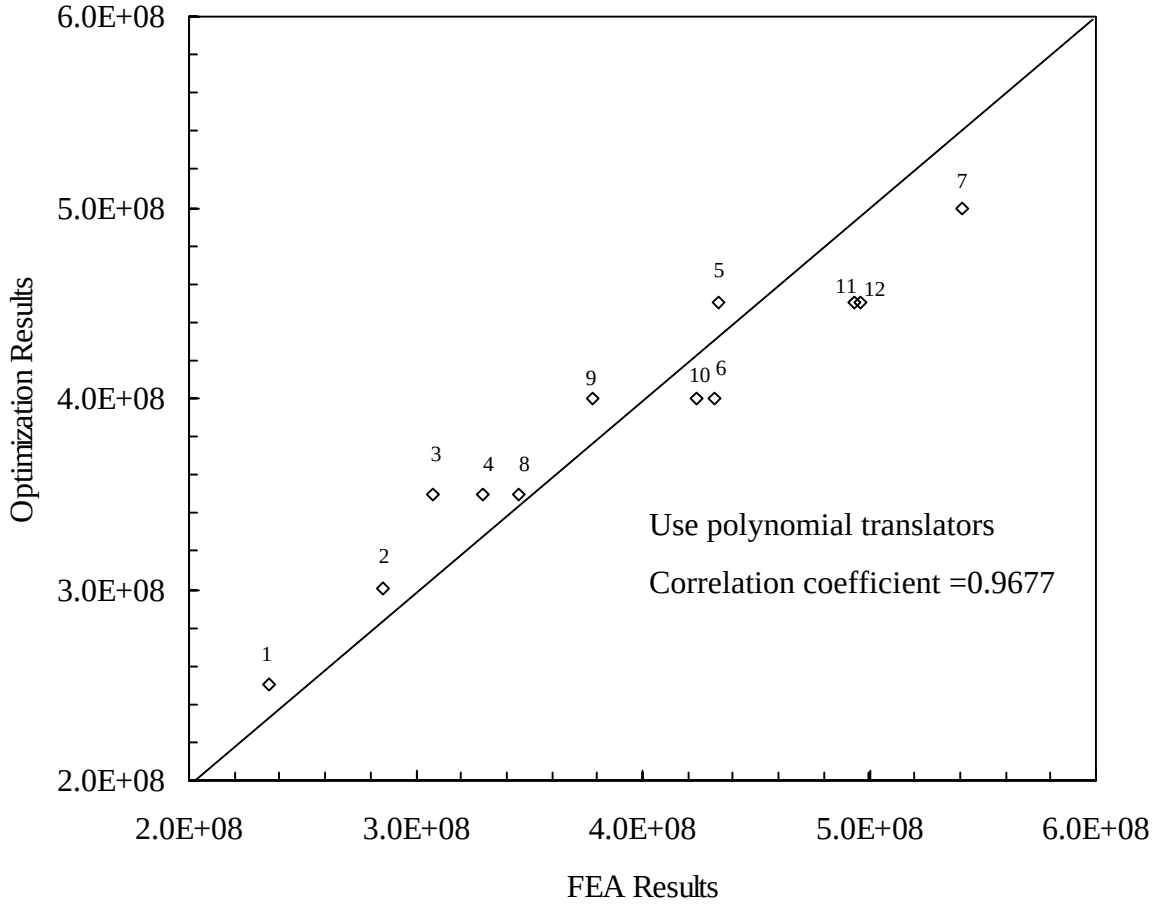


Figure 6.11b: Comparison of FEA Results and the Optimization Results Obtained using RSP Translators for the F/A Stiffness of B-pillar to Rocker Joint

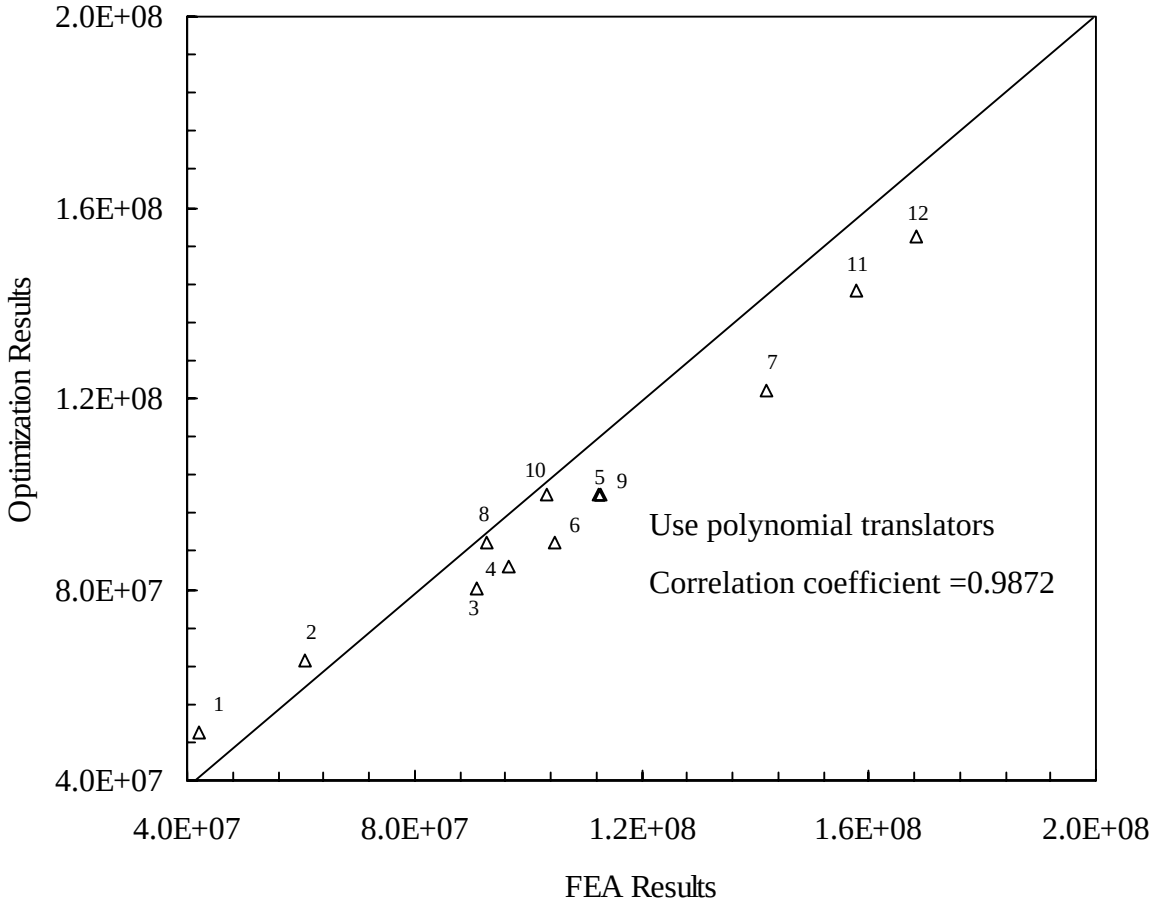


Figure 6.11c: Comparison of FEA Results and the Optimization Results Obtained using RSP Translators for the Torsion Stiffness of B-pillar to Rocker Joint

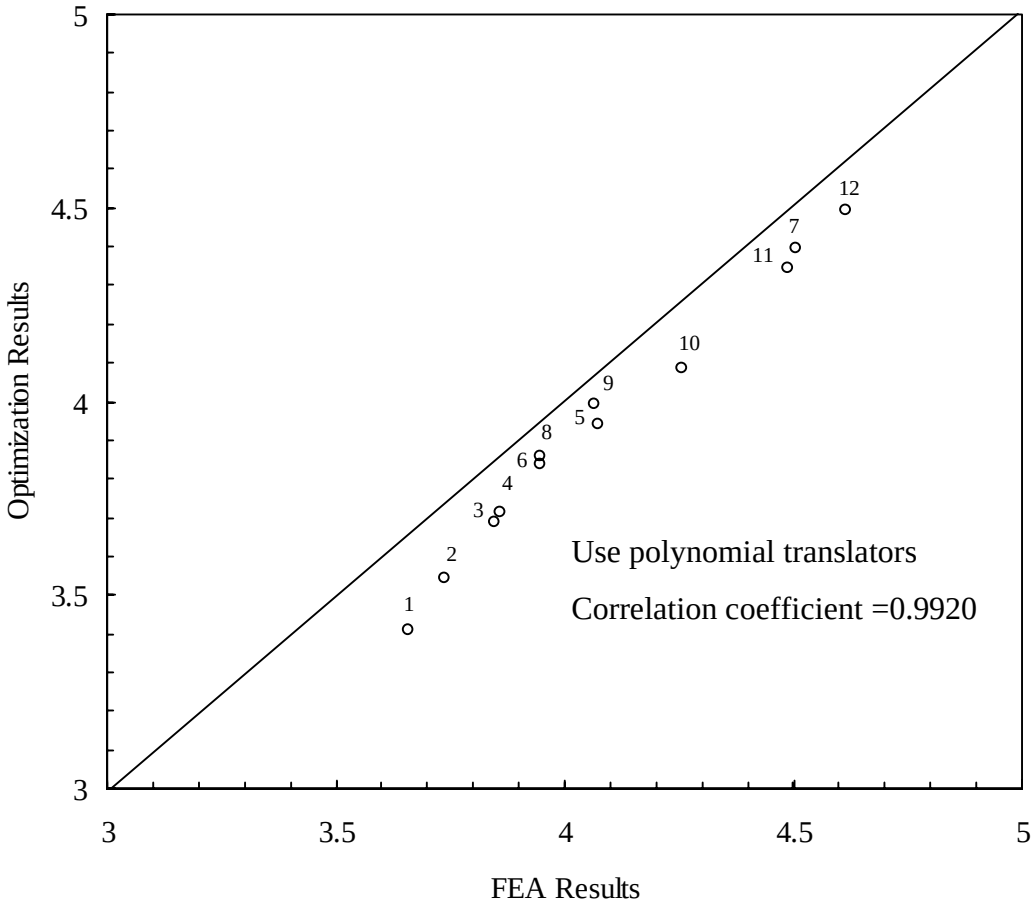


Figure 6.11d: Comparison of FEA Results and the Optimization Results Obtained using RSP Translators for the Mass of B-pillar to Rocker Joint

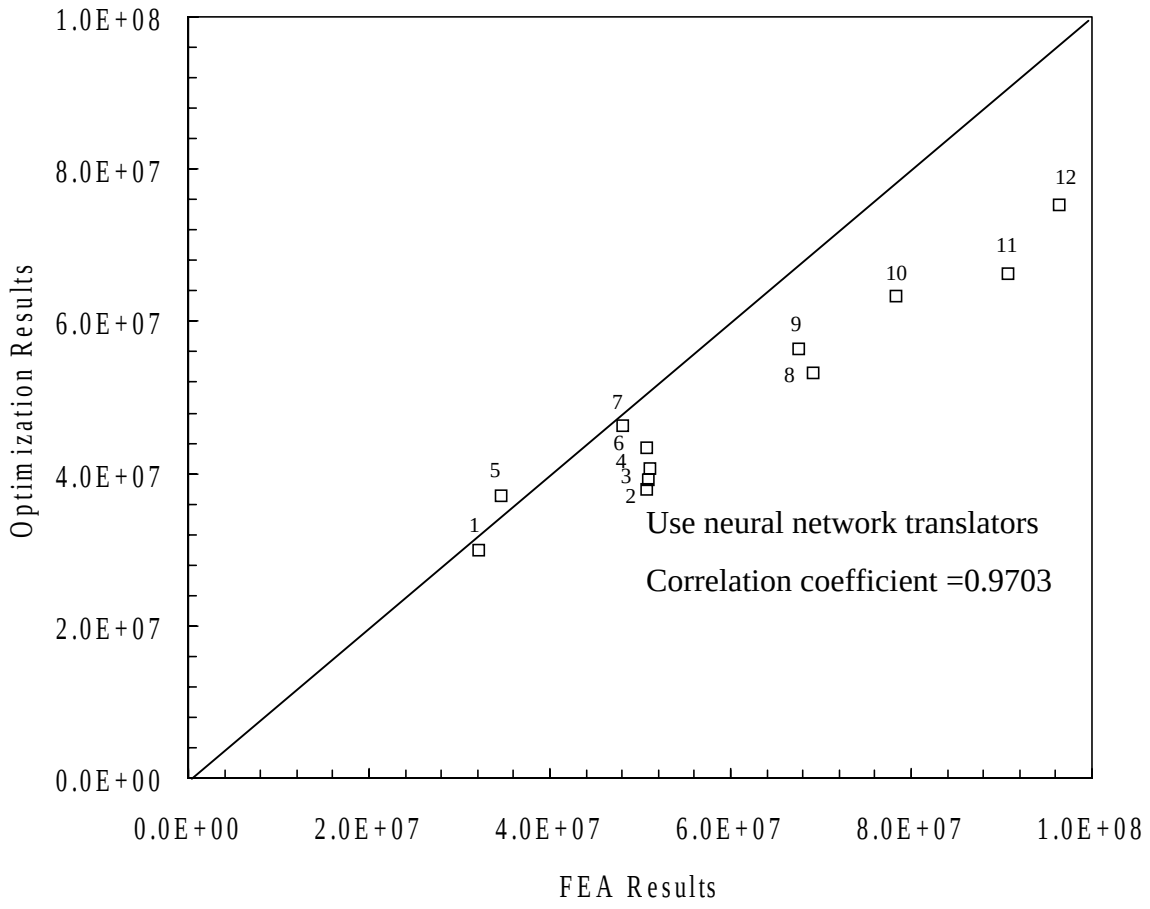


Figure 6.12a: Comparison of FEA Results and the Optimization Results Obtained using NN Translators for the I/O Stiffness of B-pillar to Rocker Joint

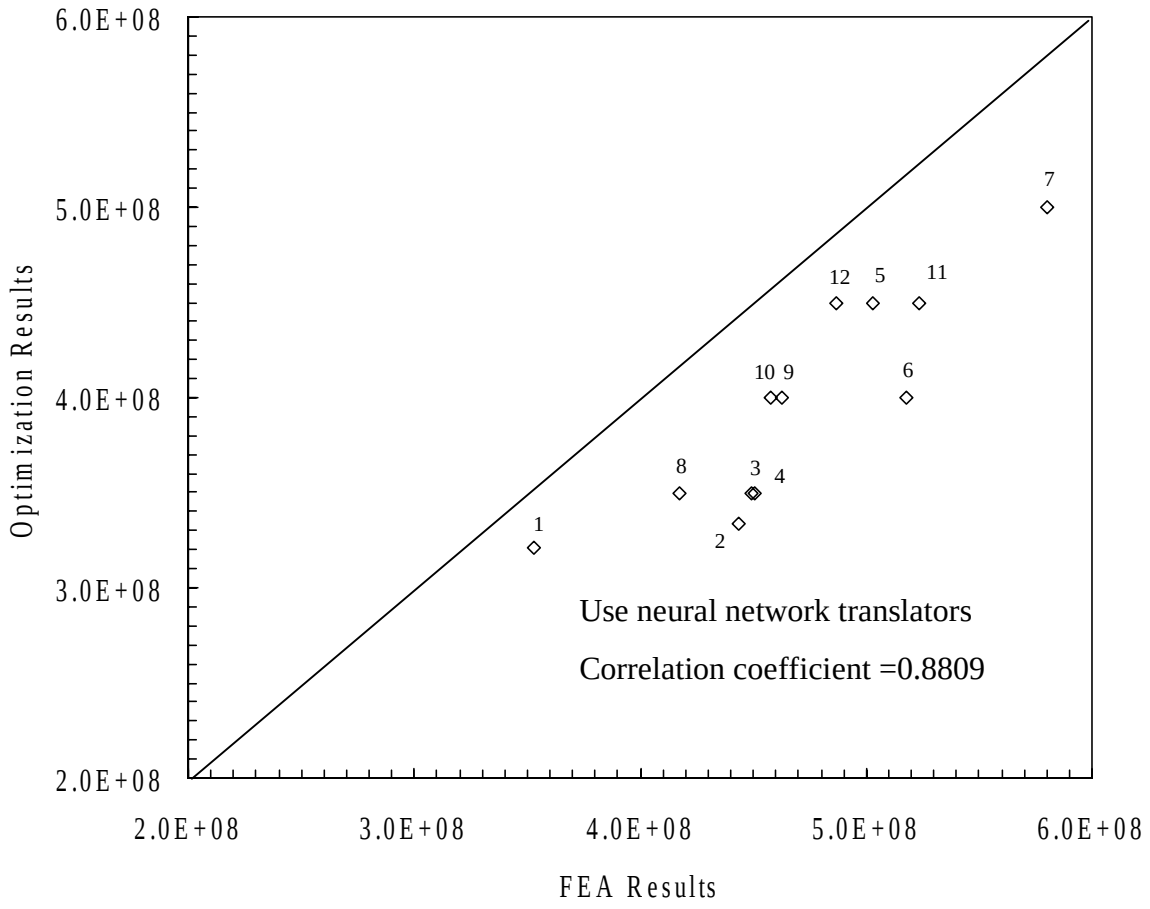


Figure 6.12b: Comparison of FEA Results and the Optimization Results Obtained using NN Translators for the F/A Stiffness of B-pillar to Rocker Joint

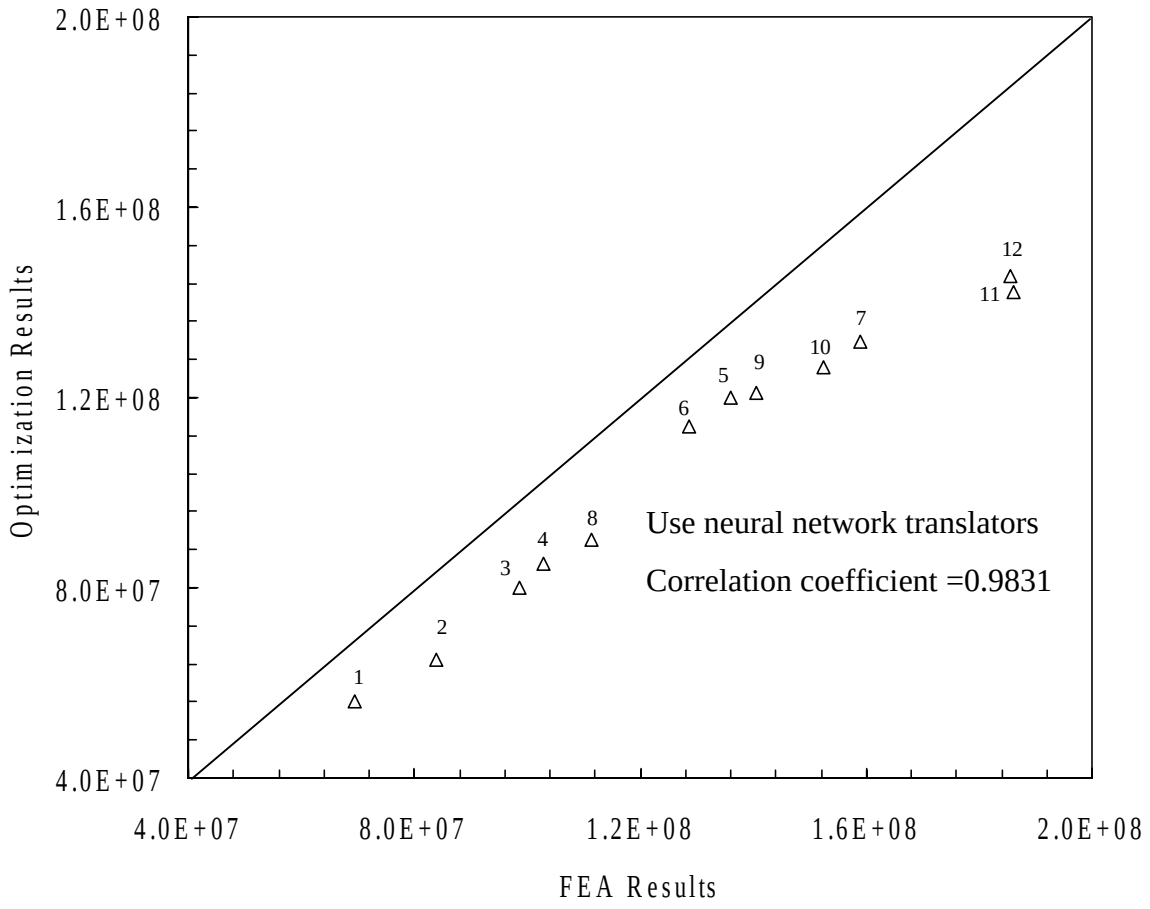


Figure 6.12c: Comparison of FEA Results and the Optimization Results Obtained using NN Translators for the Torsion Stiffness of B-pillar to Rocker Joint

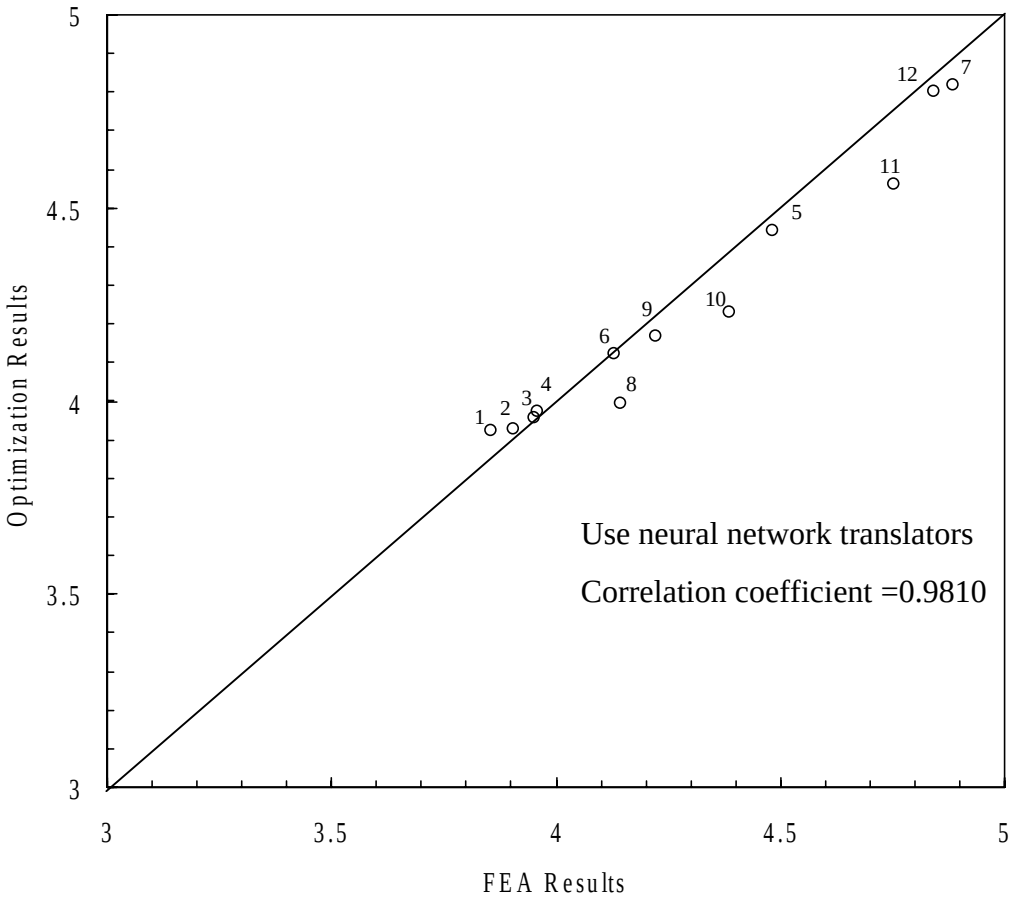


Figure 6.12d: Comparison of FEA Results and the Optimization Results Obtained using NN Translators for the Mass of B-pillar to Rocker Joint

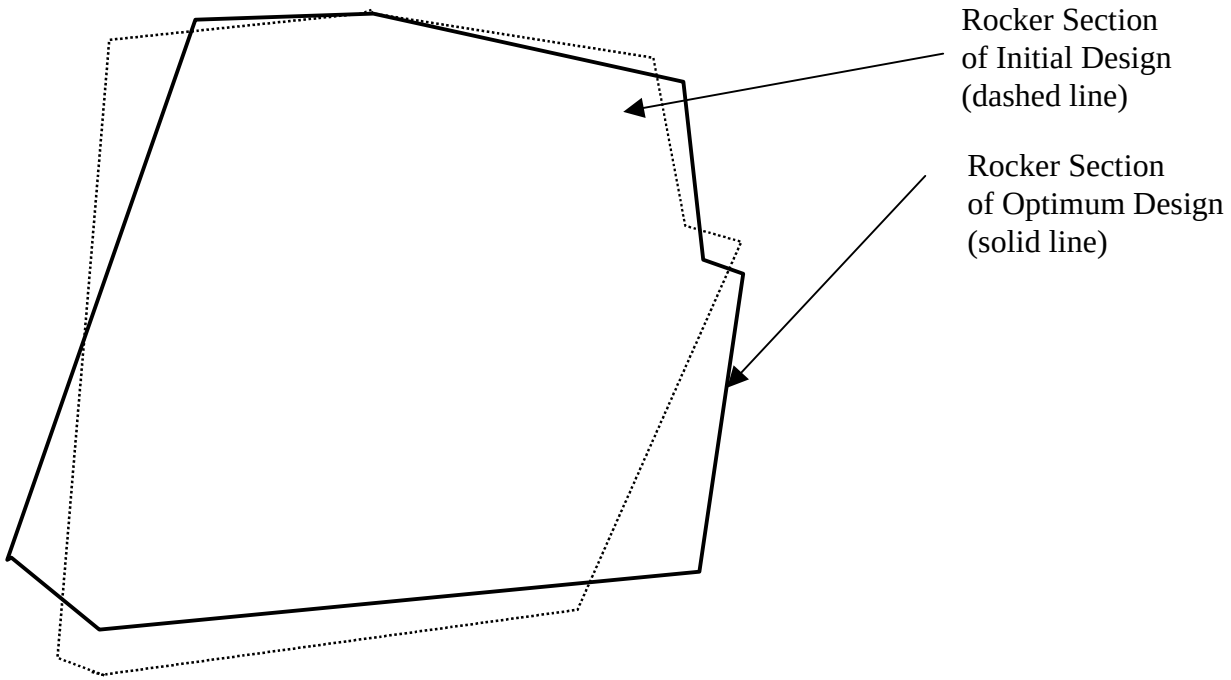


Figure 6.13: Comparison of Rocker Cross Sections of the Initial Design and Optimum Design

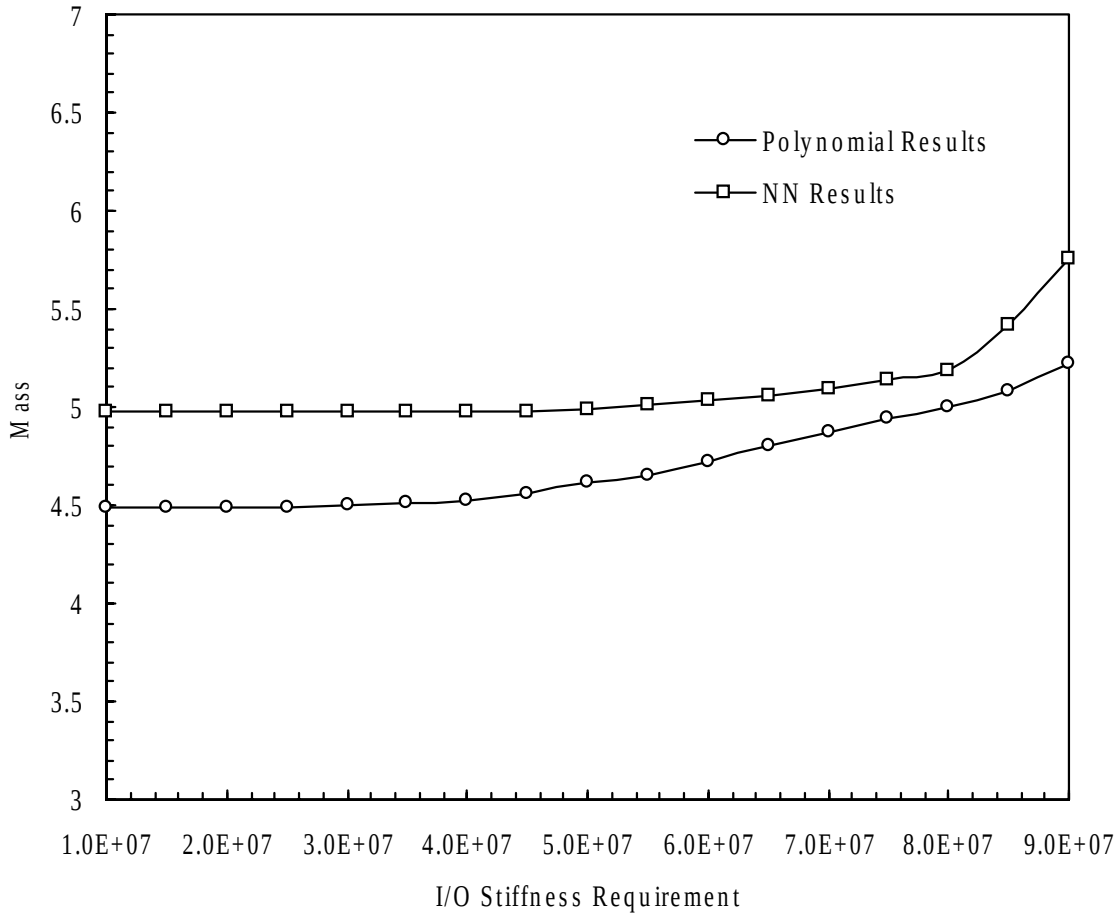


Figure 6.14: Relation Between Mass and I/O Stiffness Requirement for B-pillar to Rocker Joint ($K_{F/A}$>5.2297E8, K_{tor}>7.9788E7)

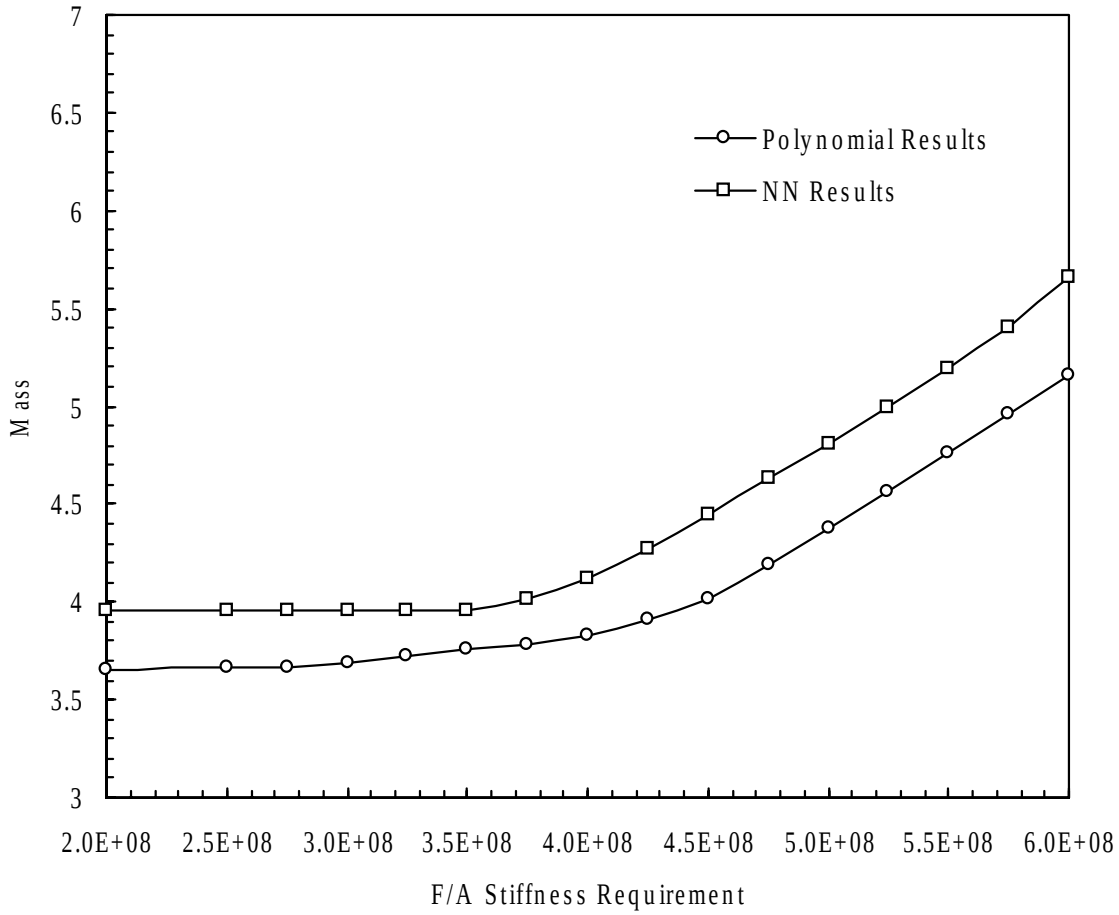


Figure 6.15: Relation Between Mass and F/A Stiffness Requirement for B-pillar to Rocker Joint ($K_{I/O}$>4.3899E7, K_{tor}>7.9788E7)

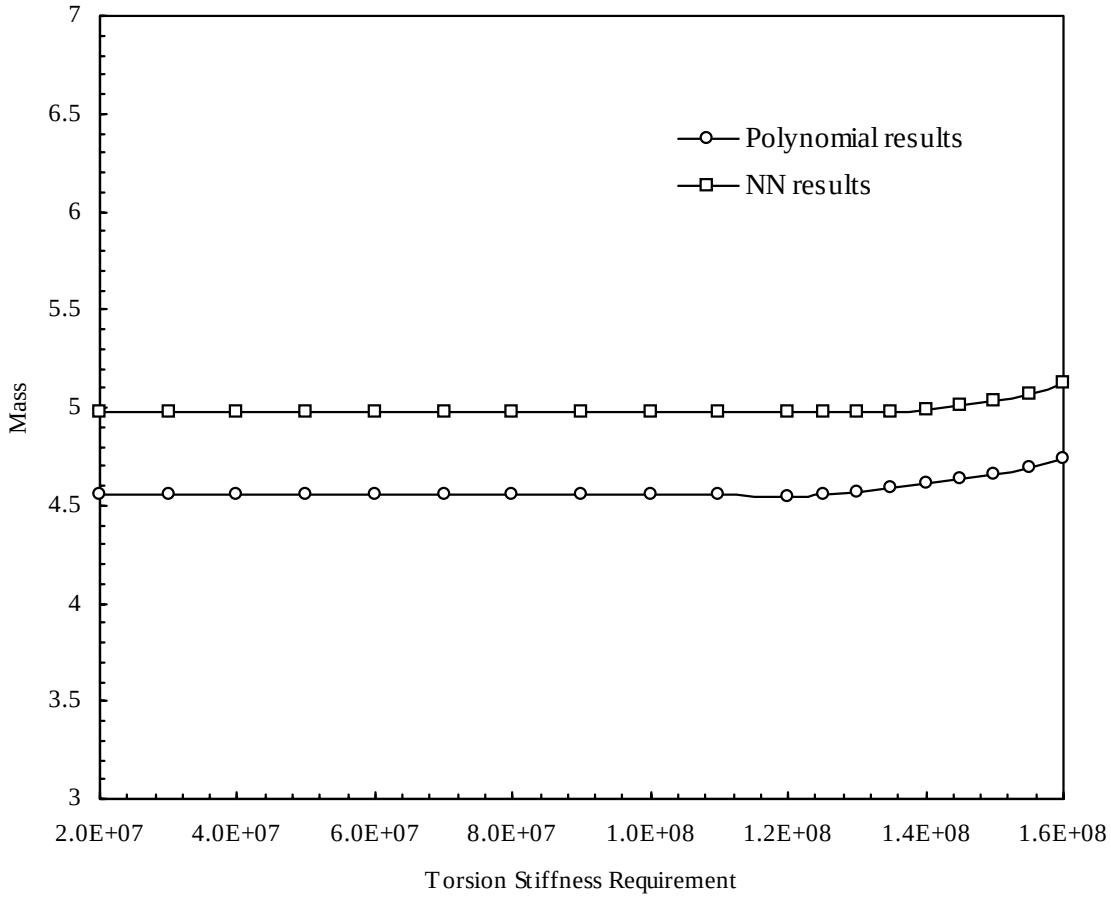


Figure 6.16: Relation Between Mass and Torsion Stiffness Requirement for B-pillar to Rocker Joint ($K_{I/O}$>4.3899, $K_{F/A}$>5.2297E7)

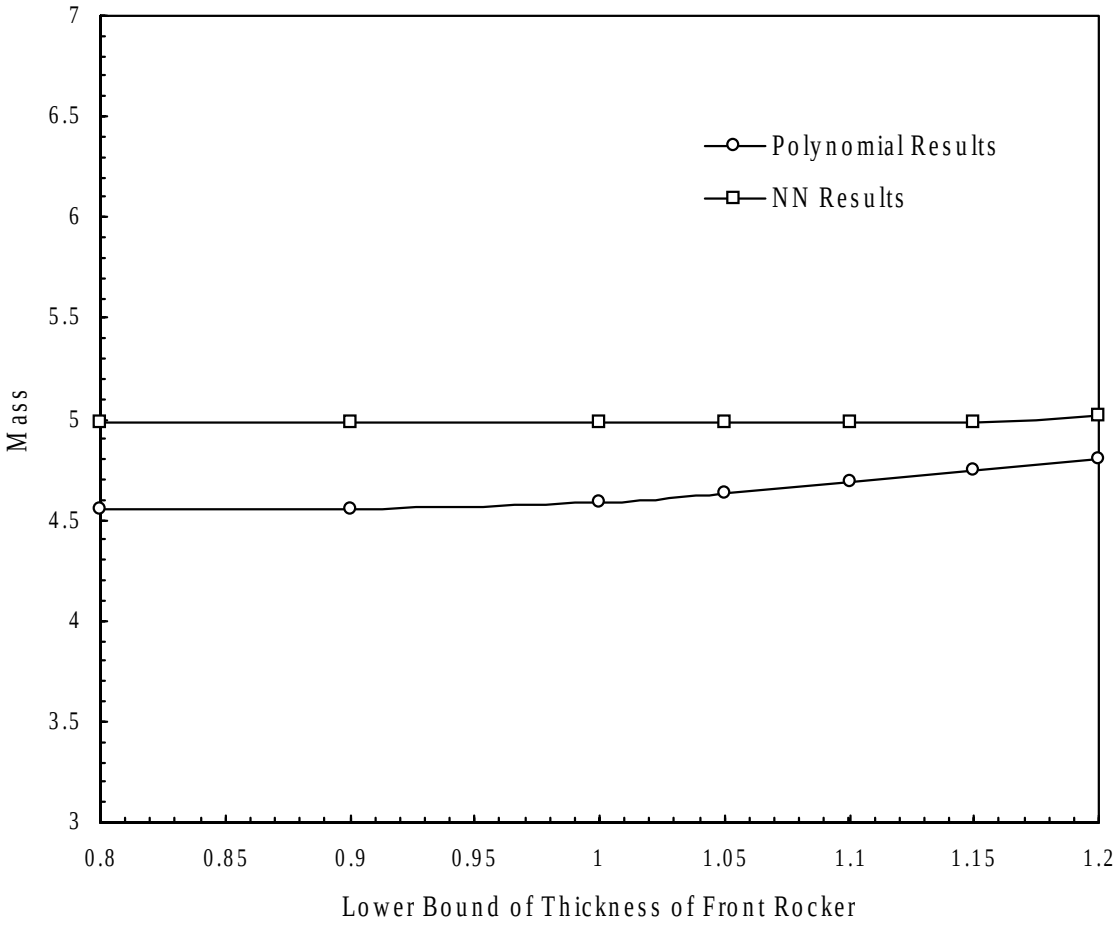


Figure 6.17: Relation Between the Lower Bound of *Thickness of Front Rocker* and the Mass of Optimum Design of B-pillar to Rocker Joint ($KI_{/O}$>4.3899E7, $K_{F/A}$>5.2297E8, K_{Tor}>7.9788E7)

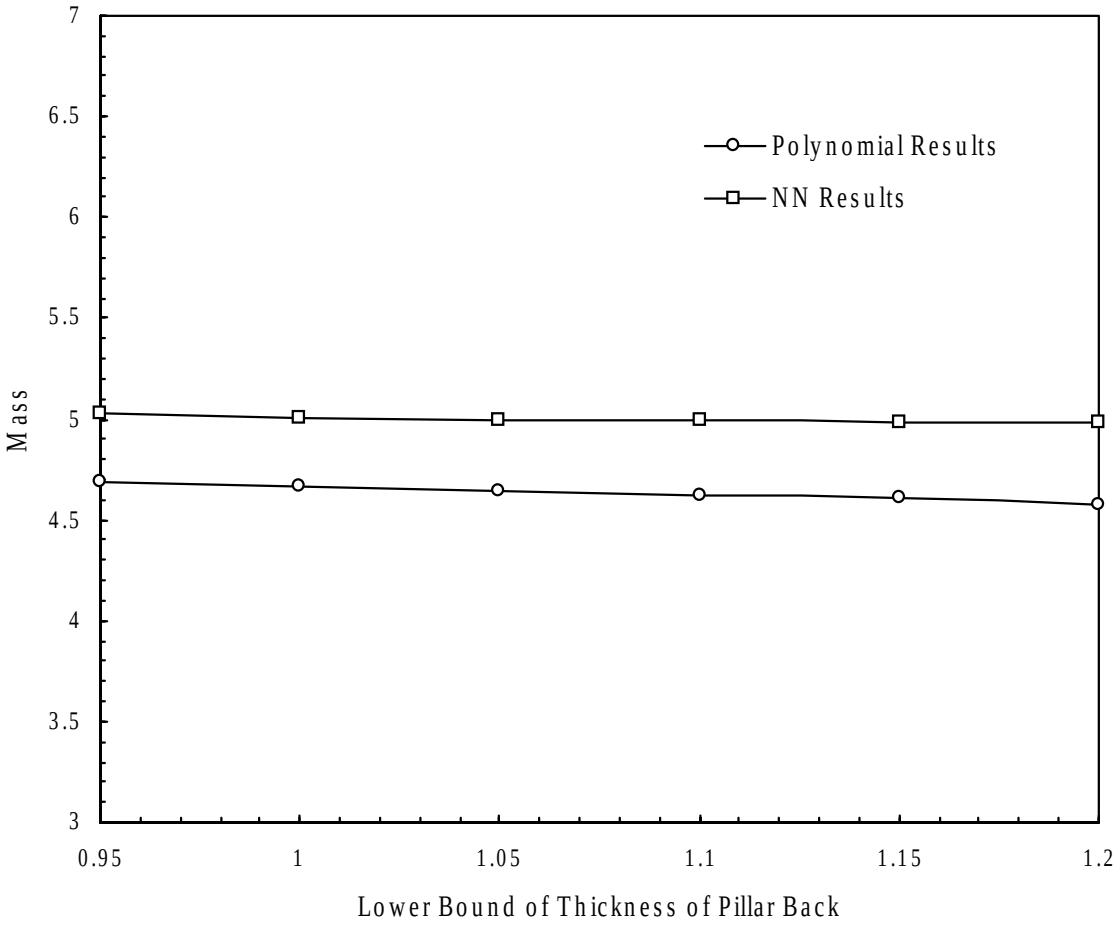


Figure 6.18: Relation Between the Lower Bound of *Thickness of Pillar Back* and the Mass of Optimum Design of B-pillar to Rocker Joint ($KI_{/O}$>4.3899E7, $K_{F/A}$>5.2297E8, K_{Tor}>7.9788E7)

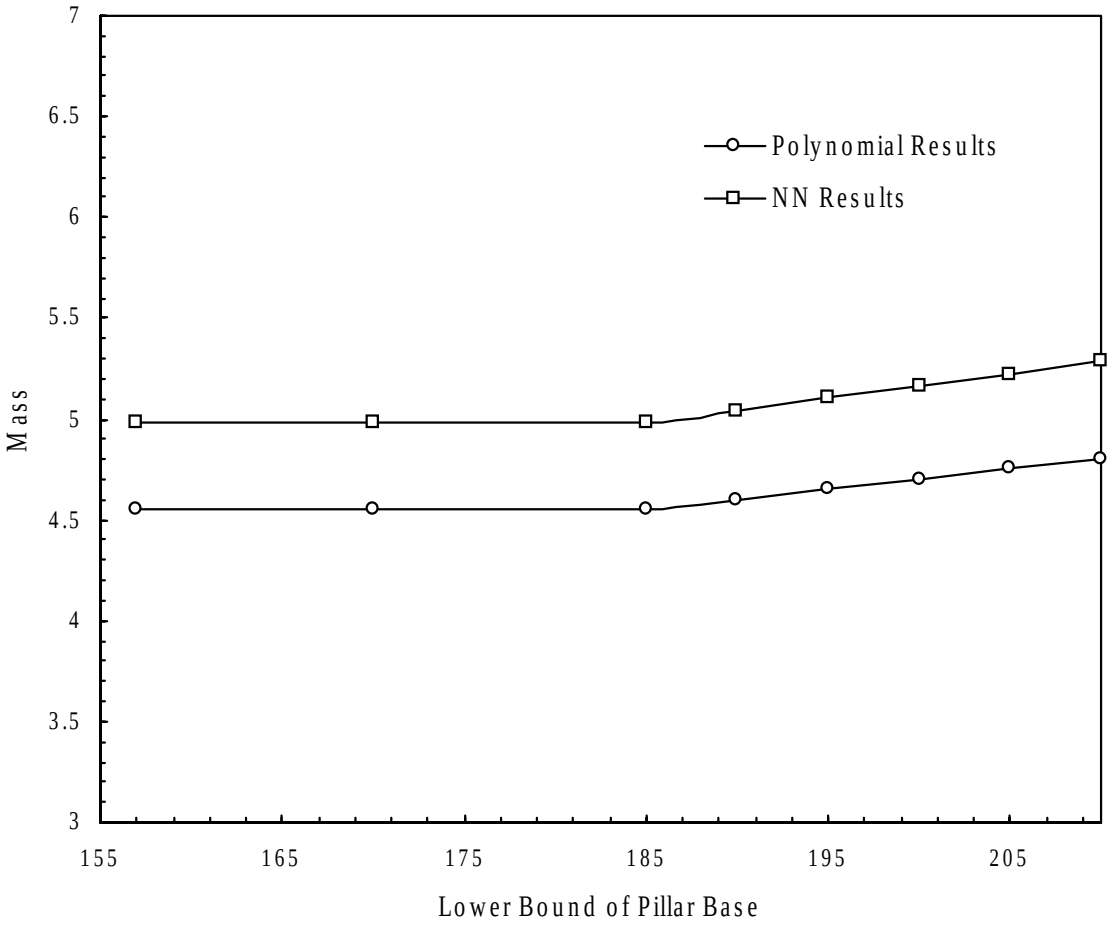


Figure 6.19: Relation Between the Lower Bound of *Pillar_base* and the Mass of Optimum Design of B-pillar to Rocker Joint ($KI_{/O}$>4.3899E7, $K_{F/A}$>5.2297E8, K_{Tor}>7.9788E7)

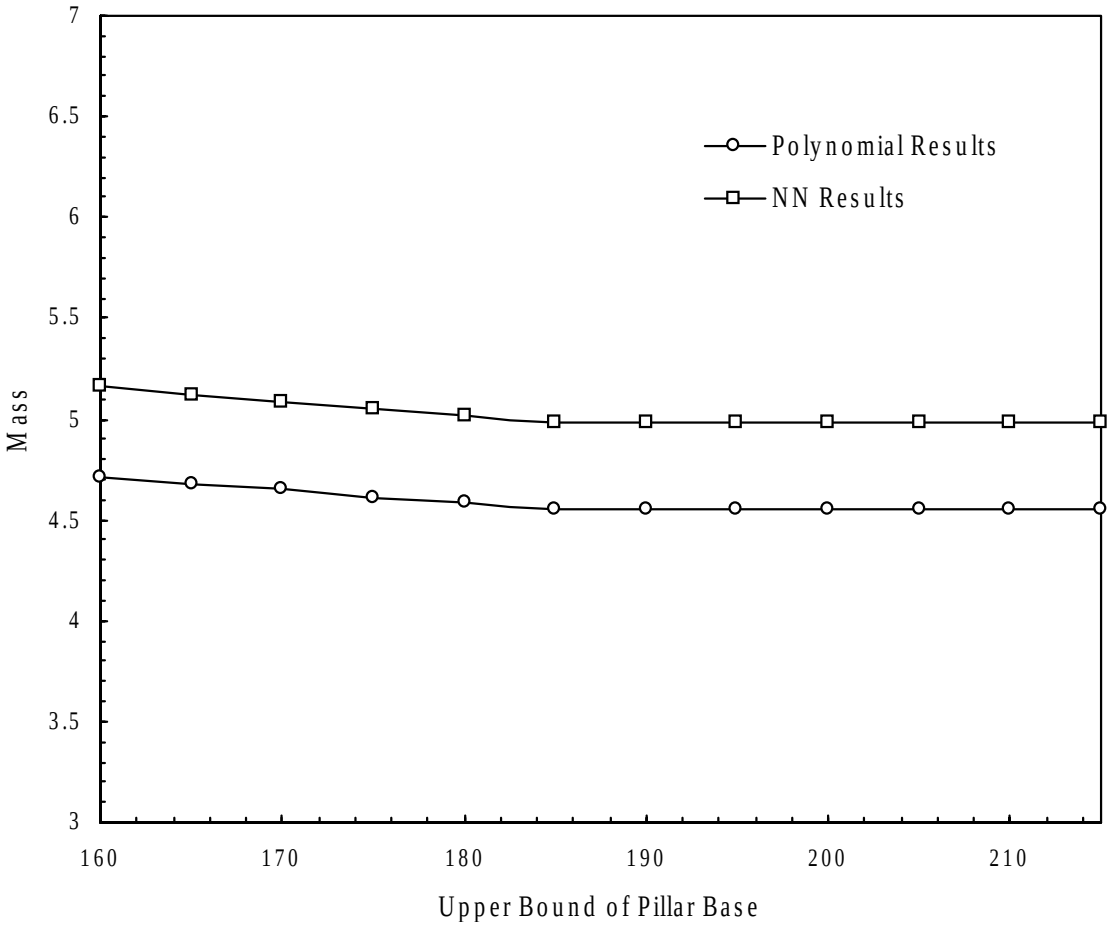


Figure 6.20: Relation Between the Upper Bound of *Pillar_base* and the Mass of Optimum Design of B-pillar to Rocker Joint ($KI_{/O}$>4.3899E7, $K_{F/A}$>5.2297E8, K_{Tor}>7.9788E7)

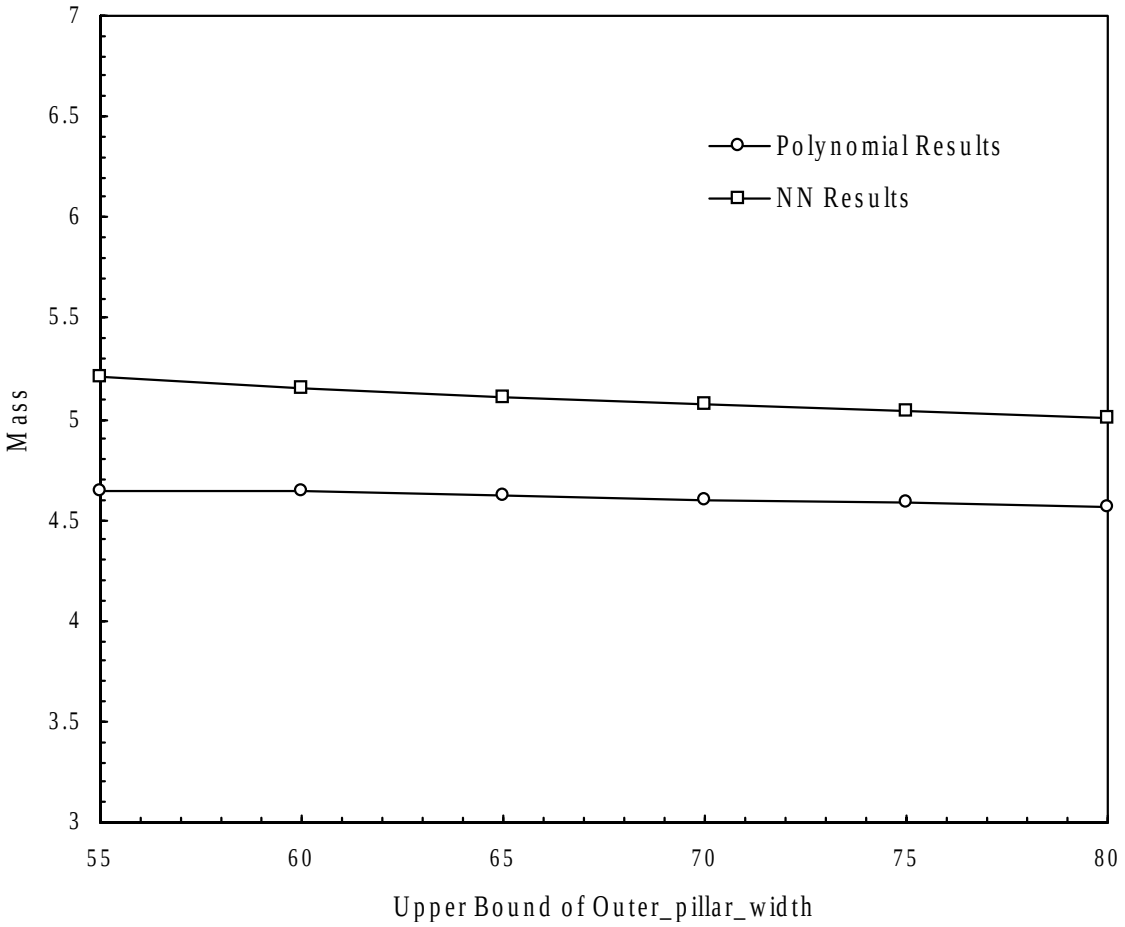


Figure 6.21: Relation Between the Upper Bound of *Outer_pillar_width* and the Mass of Optimum Design of B-pillar to Rocker Joint ($KI_{/O}$>4.3899E7, $K_{F/A}$>5.2297E8, K_{Tor}>7.9788E7)

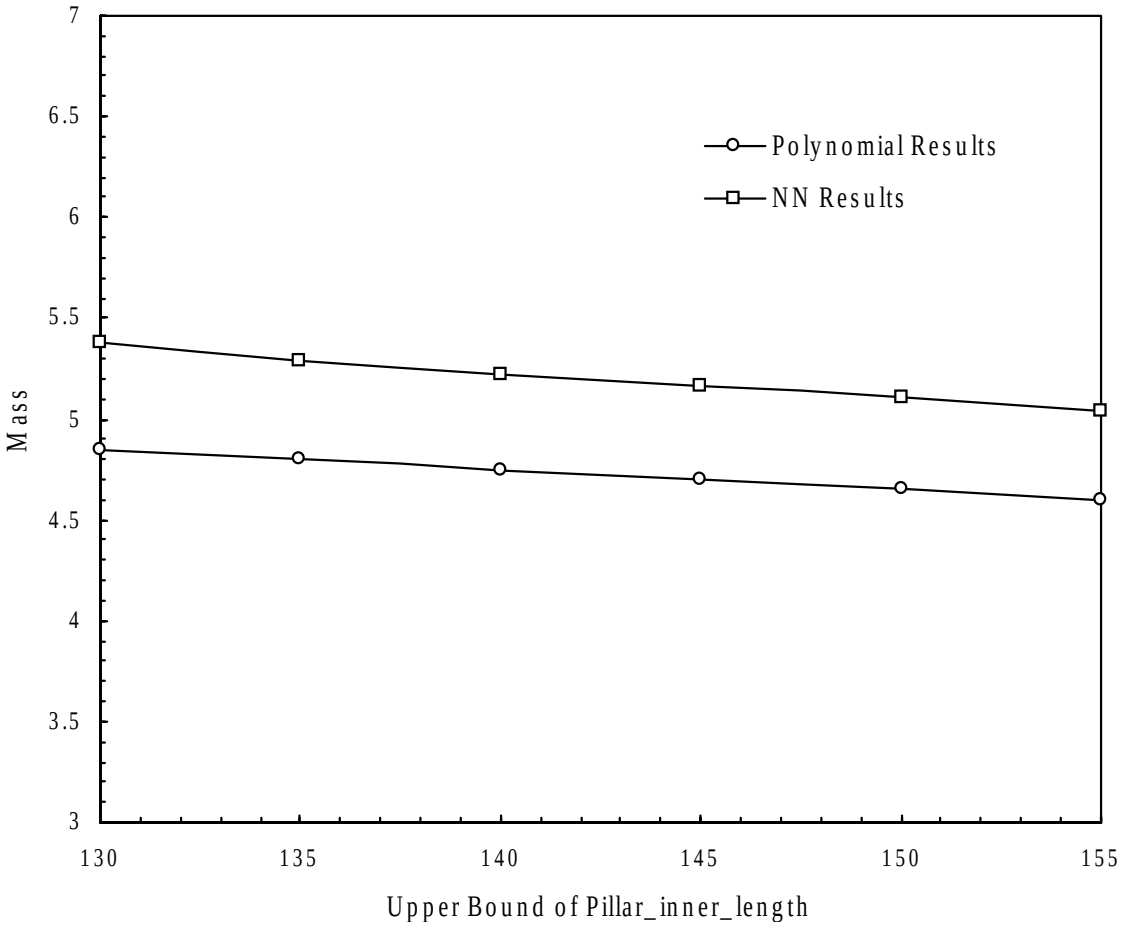


Figure 6.22: Relation Between the Upper Bound of *Pillar_inner_length* and the Mass of Optimum Design of B-pillar to Rocker Joint ($KI_{/O}$>4.3899E7, $K_{F/A}$>5.2297E8, K_{Tor}>7.9788E7)

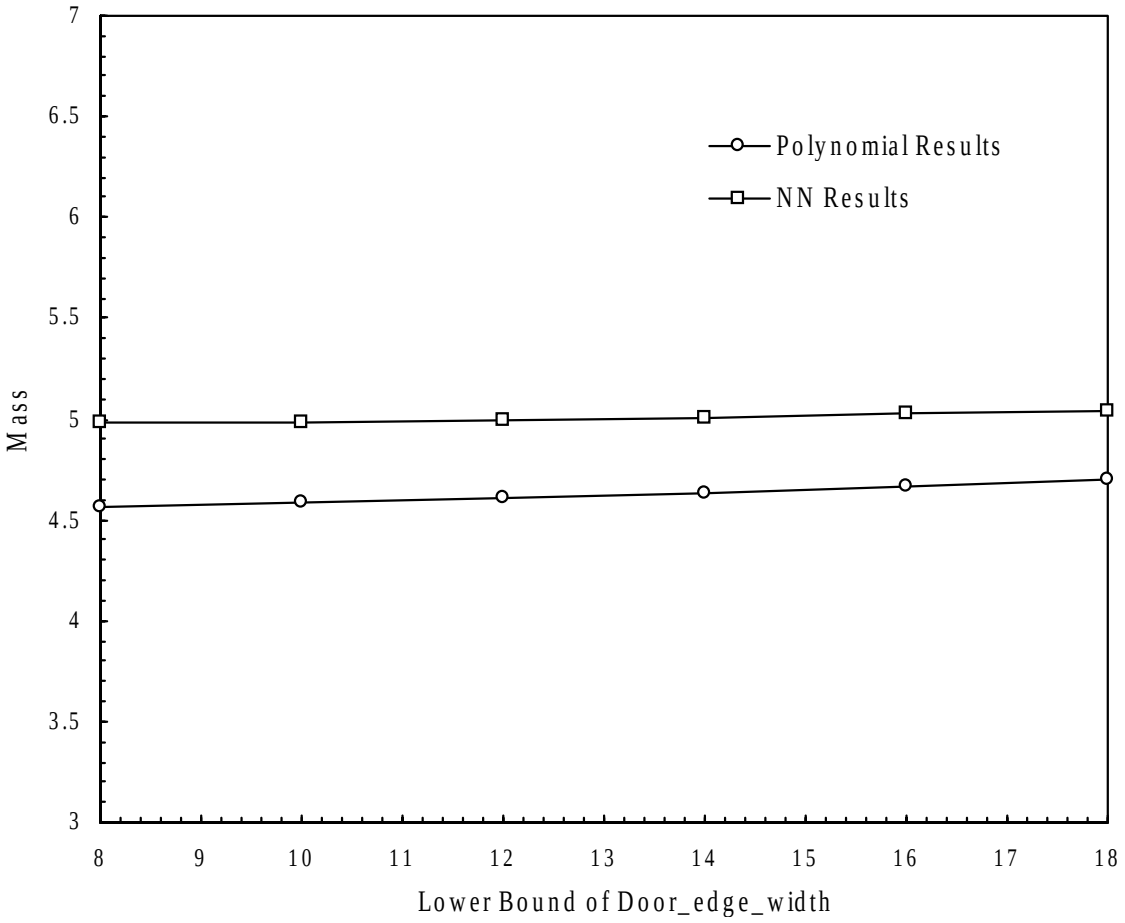


Figure 6.23: Relation Between the Lower Bound of *Door_edge_width* and the Mass of Optimum Design of B-pillar to Rocker Joint ($KI_{/O}$>4.3899E7, $K_{F/A}$>5.2297E8, K_{Tor}>7.9788E7)

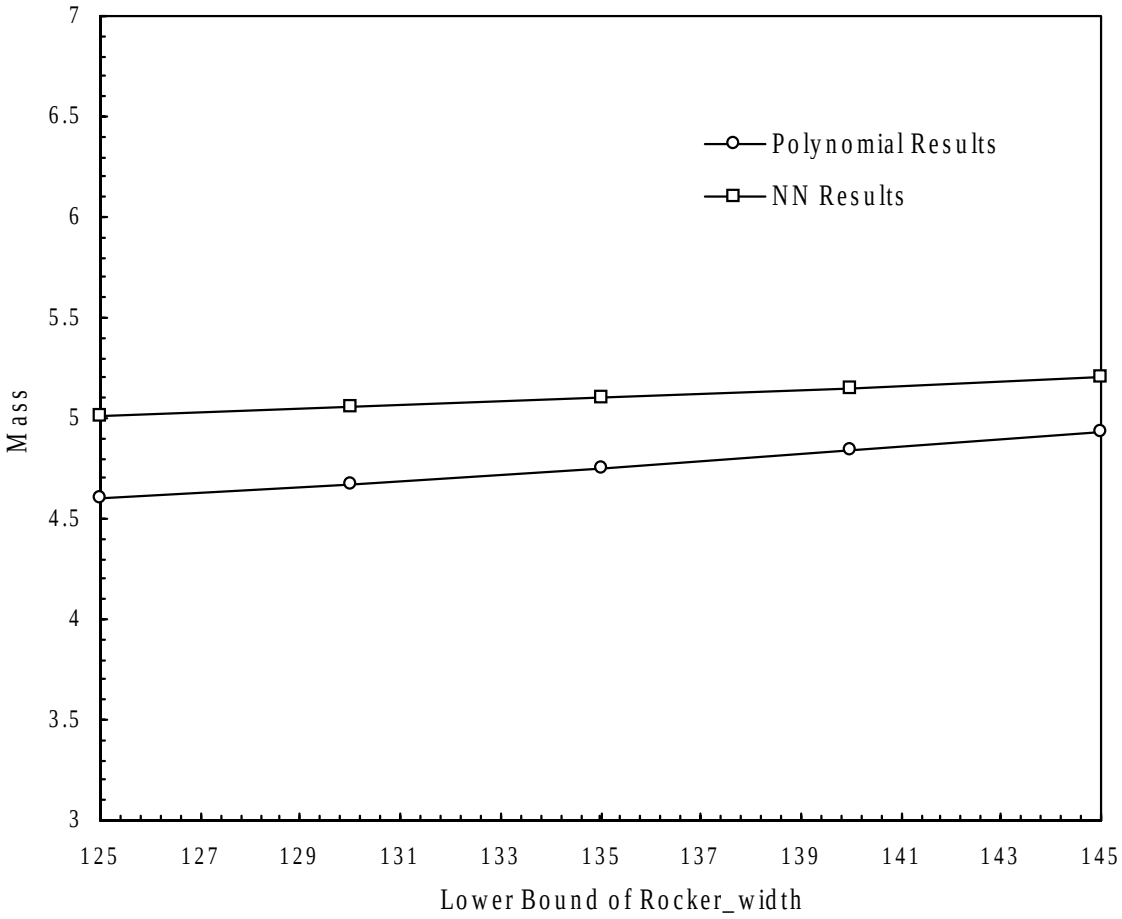


Figure 6.24: Relation Between the Lower Bound of *Rocker_width* and the Mass of Optimum Design of B-pillar to Rocker Joint ($KI_{/O}$>4.3899E7, $K_{F/A}$>5.2297E8, K_{Tor}>7.9788E7)

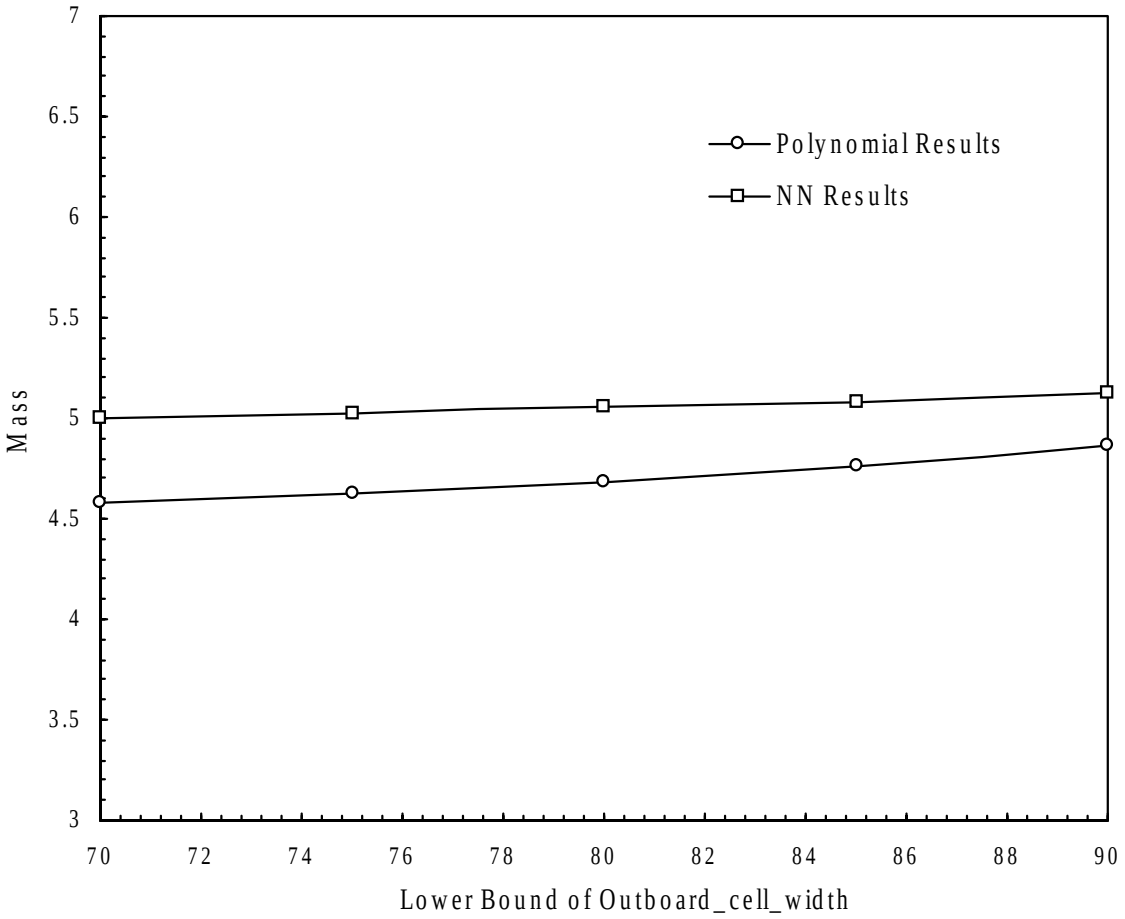


Figure 6.25: Relation Between the Lower Bound of *Outboard_cell_width* and the Mass of Optimum Design of B-pillar to Rocker Joint ($KI_{/O}$>4.3899E7, $K_{F/A}$>5.2297E8, K_{Tor}>7.9788E7)

Chapter 7
Developing Translator B for an Actual A-pillar to Roof Rail Joint

7.1 Introduction

This chapter applies the general methodology for developing translator B in Chapter 3 to an A-pillar to roof rail joint. The A-pillar to roof rail joint is a "Y" shape joint, located on the upper left and upper right sides of the driver and the passenger, respectively (Fig.1.1). The A-pillar joint has three branches. The branch that is perpendicular to the side of the car is called header. The branch that is parallel to the top of the front door is called roof rail. A-pillar is the branch that lies between the windshield and the front door. A-pillar joint supports the windshield of a car, the roof and part of the front door. Chapter 5 describes this joint in detail.

This chapter is organized as follows:

- Section 7.2 formulates the optimization problem used in developing translator B for the A-pillar to roof rail joint. This section explains the selection and classification of design variables, the objective function and the constraints.
- Section 7.3 presents some applications of translator B to some actual design problems. First, we use several methods to check the convergence of translator B. Then, we compare the optimization results with the FEA results. We also use translator B to redesign two actual joints. Since both response surface polynomials and neural networks are used in developing translator B, we also compare the results of both translators. Finally, we perform a parametric study to examine the effects of stiffness requirements and the lower and upper bounds of several important design variables on the mass of the optimum design. Design guidelines for improving the design of the A-pillar to roof rail joint are presented and discussed.

7.2 Formulation of the Optimization Problem for Developing a Translator B for an A-pillar to Roof Rail Joint

7.2.1 Definition of the Problem

Translator B finds the most efficient design that meets given performance targets (stiffness requirements), and satisfies packaging, manufacturing, styling and performance constraints.

Translator B uses optimization. The optimization program searches through the feasible domain to find the most efficient design that satisfies the constraints and performance targets. During each iteration of the optimization process, a new design is analyzed to get its stiffness and mass. It would be very time-consuming to use FEA to get these responses. Moreover, the intermediate designs created in the optimization process are not always feasible, and can not be generated using a generic CAD model. To circumvent these problems, translator A, which was developed in Chapter 5, is used to predict the stiffnesses and masses of new designs during the optimization process. Translator A is encoded in the program of translator B. Translator B uses two types of translators A, a *response surface polynomial* (RSP) translator A and a *neural network* (NN) translator A. Chapter 5 describes in detail the translator A for the A-pillar to roof rail joint.

There are different types of construction for the A-pillar to roof rail joints. As Chapter 5 explained, different construction types have different reinforcements. Some reinforcements are more common than others. For example, an A-pillar reinforcement that follows the shape of the upper plate of the A-pillar (part 7 in Fig. 5.2) is common. After consulting with engineers from an automotive company, we decided to consider only one type of construction shown in Fig. 5.2. This type of A-pillar to roof rail joint consists of the following parts: the roof (part 1), the top plate of the A-pillar joint (part 2), the bottom plate of the A-pillar joint (part 3), the bottom plate of the header (part 4), and the A-pillar reinforcement (part7). The longitudinal plates that connect two flanges at the

three sections of the A-pillar joint (part 5), and the header reinforcement (part 6) are not considered. Translator B has 48 design variables.

7.2.2 Design Variables

Table 7.1 shows the design variables for the A-pillar to roof rail joint. As explained in Chapter 3, design variables are divided into four types:

1) Design variables that are *fixed by the program* because of design rules or conventions.
2) Design variables that are *fixed by the user*.
3) *Dependent design variables* whose values can be expressed in terms of other design variables.
4) *Independent design variables* that can change in optimization.

As explained in Chapter 5, some design variables, such as the widths of flanges, and the distance between two adjacent spot welds, are almost always fixed because of manufacturing requirements. These variables are fixed because the user is not expected to change these variables. Values of some design variables, such as the orientation of A-pillar, are determined from styling requirements. For a particular joint, the lengths of its three branches are generally fixed because of styling and manufacturing considerations. This allows us to compare the stiffnesses of different joint designs fairly. The user fixes the values of these variables in the input file. These variables do not change in optimization.

Six design variables are fixed by the program, namely, *distance_from_edge, flange1_width, flange2_width, flange3_width, spot_weld_spacing* (Fig. 5.7) and *ap_flange1_angle* (Fig.5.10). The values of the first five design variables are determined from manufacturing considerations. *Flange1_width, flange2_width,* and *flange3_width* are fixed at 20 mm, 16 mm, and 16 mm, respectively. *Spot_weld_spacing* and *distance_from_edge* are fixed at 50 mm and 8 mm, respectively. As mentioned in Section 5.2.1, *Ap_flange1_angle* is set to be zero. This ensures that stiffnesses are defined in the same way for different joint designs, which allows us to compare the stiffnesses fairly.

There are two design variables of the second type (fixed by the user), namely, *theta*, and *phi*. Because the orientation of A-pillar is determined from styling requirements, *theta* and *phi* are fixed at 23 degrees and 76 degrees, respectively. These two values were determined by examining hardware joints.

Unfortunately, there is no consistent definition of the center of a joint. In practice, the lengths of the joint branches vary because there is no widely acceptable definition of a reference point that should be used to measure the length of a branch. Indeed, we found a considerable variation in the lengths of the A-pillar, header and roof rail among several hardware joints. It is important, to have a consistent set of reference points for measuring the lengths of all the joint branches and to fix these lengths so that we can compare fairly alternative joint designs. Moreover, we need to define the orientations of the planes of the sections at the ends of the three branches.

Figure 7.1 shows a proposed definition of the lengths of the branches of the A-pillar to roof joint. First, we define reference point, O, which is at the intersection of two straight lines tangent to the bases of the outboard flanges of the A-pillar and the roof rail. The length of the header is the distance from point O to the end of the header. A reference point is similarly defined for measuring the lengths of the A-pillar and the roof rail. Figure 7.1 shows that this is the intersection of the lines tangent to the roots of the flanges of the header and the A-pillar. We also defined the planes of the sections at the ends of the three branches to be normal to axes of these members. The axis of each member is tangent to one of the flanges. Specifically, the axis of the A-pillar is tangent to the outboard flange of the A-pillar. The axis of the roof rail is tangent to the outboard flange of the roof rail. The axis of the header is tangent to the forward flange of the header (Fig.5.4)

According to our measurements on hardware joints, we fixed the lengths of the header, the roof rail and the A-pillar at 260 mm, 213 mm and 161 mm, respectively. We observe from Fig. 7.1 that the *A_pillar_offset, Roof_rail_offset* and *Header_offset* are

dependent variables because they can be expressed in terms of the lengths of the branches and other independent variables.

Table 7.1 classifies the design variables. The optimization program does not directly change these design variables that are fixed by the program or by the user, or are dependent. Only 37 independent design variables are considered in the translator B. The states and ranges of the design variables are shown in Table 7.1.

7.2.3 Objective Function

The objective function of the translator B for the A-pillar to roof rail joint can be expressed as

$$F = \alpha M + (1-\alpha)\sqrt{((K_{I/O} - \hat{K}_{I/O})/\overline{K}_{I/O})^2 + ((K_{F/A} - \hat{K}_{F/A})/\overline{K}_{F/A})^2 + ((K_{Tor} - \hat{K}_{Tor})/\overline{K}_{Tor})^2} \quad (7.1)$$

where α is a weighting factor. M is the mass of the joint. $K_{I/O}, K_{F/A}$, and K_{Tor} are the I/O, F/A and torsion stiffnesses of the joint. $\hat{K}_{I/O}, \hat{K}_{F/A}$, and $\hat{K}_{Tor}$ are the user-specified requirements (targets) for I/O, F/A and torsion stiffness. $\overline{K}_{I/O}, \overline{K}_{F/A}$, and $\overline{K}_{Tor}$ are the stiffness values used to normalize the stiffness. Generally, mass varies from 2 to 8 kg. Each stiffness term varies from 0 to 3. We normalize the stiffness so that the mass and the stiffnesses can have compatible magnitude.

In this study, α is set to be 1. Only mass is considered in the objective function. However, α can assume any value between 0 and 1. In this case, the objective function will be the sum of mass and some measure of difference between the stiffness of a design and the given stiffness targets. The optimum design obtained using this objective function will be a joint that not only has low mass but also its stiffnesses are close to given targets.

7.2.4 Constraints

The constraints are divided into five types, including packaging, manufacturing, styling, mathematical, and performance constraints.

Packaging constraints are related to the arrangement of the car components in space. Manufacturing constraints are due to manufacturing limitations. Styling constraints are due to styling requirements. Mathematical constraints are used to control the ranges of different design variables to ensure that a design has a feasible geometry. Performance constraints dictate that the stiffnesses should exceed given minimum values.

The constraints used in translator B are explained in the following section. The number in the parenthesis of each constraint is the corresponding constraint number in the program for translator B.

- Packaging Constraints

1) *H_blending_rad* must be larger than the overall width of roof rail so that the header does not intersect roof rail branch (Figs. 5.6, 5.9, and 7.2) (2).

$$Width\ of\ RR\ cross\ section + flange3\_width - h\_blending\_rad \leq 0 \qquad (7.2)$$

2) *RR_blending_rad* should be larger than the overall width of header cross section to prevent the two branches from intersecting (Figs.5.5, 5.6, and 5.8)(3).

$$H\_width + flange3\_width - header\_horizontal\_offset + flange3\_width - RR\_blending\_rad \leq 0 \qquad (7.3)$$

3-5) The cut distances for parts 2 and 7, and the extension distance of part 4 at the header should be smaller than the overall width of roof rail cross section to avoid intersecting the roof rail branch (Figs. 5.6, 5.12, 5.14, 5.15and 7.2) (7-9).

$$Width\ of\ RR\ cross\ section - (header\_offset - Part2\_cut\_distance) \leq 0 \qquad (7.4)$$

Width of RR cross section –(header_offset-Part4_extension_distance) ≤ 0 (7.5)

Width of RR cross section – (header_offset - Part7_cut_distance) ≤ 0 (7.6)

6) The cut distance for part 3 should be smaller than the extension distance of part 4 in order for the two parts to be welded at the bottom (Figs. 5.13 and 5.14)(10).

Part3_cut_distance - part4_extension_distance ≤0 (7.7)

7,8) The edge at the bottom of roof rail section, *RLV2*, must be in a range (Fig.7.2) (19,20).

Minimum value of RLV2 - RLV2 ≤ 0 (7.8)

RLV2- maximum value of RLV2 ≤ 0 (7.9)

10,11) The angle defining the orientation of flange2 of the roof rail section, *FA2*, should be in the range [0,90] to accommodate door (Fig.7.3) (13,14).

-FA2≤0 (7.10)

FA2 – 90 ≤ 0 (7.11)

12,13) *RR_h1* should vary in a range (Fig. 5.9) (107,108).

Minimum value of RR_h1 - RR_h1 ≤ 0 (7.12)

RR_h1- maximum value of RR_h1 ≤0 (7.13)

14,15) *RR_h2* should vary in a range (Fig. 5.9) (109,110).

Minimum value of RR_h2 - RR_h2 ≤ 0 (7.14)

RR_h2 - Maximum value of RR_h2 ≤0 (7.15)

16) The vertical projection of the door thickness at the roof rail section must be nonnegative (Fig. 7.2) (18).

$$- \ Roof\_rail\_height1 \leq 0 \tag{7.16}$$

17,19) A-pillar reinforcement must be able to fit into the outer shell of the A-pillar (Figs. 5.10 and 5.11) (21-23).

$$Flange1\_angle\_up - AP\_reinf\_flange1\_angle \leq 0 \tag{7.17}$$

$$Flange2\_angle\_up - AP\_reinf\_flange2\_angle \leq 0 \tag{7.18}$$

$$AP\_reinf\_depth - AP\_window\_depth \leq 0 \tag{7.19}$$

20,21) The small edge at the bottom of the A-pillar section should vary in a range (Fig. 7.4) (27,28).

$$Minimum\ value\ of\ ALV2 - ALV2 \leq 0 \tag{7.20}$$

$$ALV2 - maximum\ value\ of\ ALV2 \leq 0 \tag{7.21}$$

22,23) The angle defining the orientation of flange2 at A-pillar section, *AP_2angle*, should be in the range [0, 90] to accommodate door (Fig.7.5) (29,30).

$$-AP\_2angle \leq 0 \tag{7.22}$$

$$AP\_2angle - 90.0 \leq 0 \tag{7.23}$$

24) The vertical projection of the door thickness at the A-pillar section must be nonnegative (Fig. 7.4) (26).

$$-A\_pillar\_door\_allowance \leq 0 \tag{7.24}$$

25,26) The A-pillar reinforcement should not intersect with outer shell of the A-pillar section (Fig. 7.6) (31,32).

Case 1: Inner plate of A-pillar reinforcement intersects the top plate of A-pillar (7.25)

Case 2: Outer plate of A-pillar reinforcement intersects the top plate of A-pillar (7.26)

The equations corresponding to these cases are omitted because they are too complicated.

27,28) *Header_offset* should not be too large or too small (Fig. 5.6) (101,102).

$$\textit{Minimum value of Header offset- header\_offset} \leq 0 \quad (7.27)$$

$$\textit{Header\_offset - maximum value of header\_offset} \leq 0 \quad (7.28)$$

29,30) *Header_horizontal_offset* must be in a range (Fig. 5.5) (103,104).

$$\textit{Minimum value of header\_horizontal\_offset- header\_horizontal\_offset} \leq 0 \quad (7.29)$$

$$\textit{Header\_horizontal\_offset-maximum value of header\_horizontal\_offset} \leq 0 \quad (7.30)$$

31,32) *Roof_rail_offset* should be in a range so that the roof rail branch is not too long or too short (Fig. 5.6) (105,106).

$$\textit{Minimum value of roof\_rail\_offset - roof\_rail\_offset} \leq 0 \quad (7.31)$$

$$\textit{Roof\_rail\_offset - maximum value of roof\_rail\_offset} \leq 0 \quad (7.32)$$

33,34) *A_pillar_offset* should be within a range so that the A-pillar branch is not too long or too short (Fig. 5.6) (111,112).

$$\textit{Minimum value of A\_pillar\_offset} - \textit{A\_pillar\_offset} \leq 0 \quad (7.33)$$

$$\textit{A\_pillar\_offset} - \textit{maximum value of A\_pillar\_offset} \leq 0 \quad (7.34)$$

35,36) There are minimum and maximum values for *angleback* (Fig. 5.9) (113,114).

$$\textit{Minimum value of angleback} - \textit{angleback} \leq 0 \quad (7.35)$$

$$\textit{Angleback} - \textit{maximum value of angleback} \leq 0 \quad (7.36)$$

37,38) The distance between the windshield and the door should not be too large or too small (Fig. 5.10) (115,116).

$$\textit{Minimum value of AP\_door\_ws\_distance} - \textit{AP\_door\_ws\_distance} \leq 0 \quad (7.37)$$

$$\textit{AP\_door\_ws\_distance} - \textit{maximum value of AP\_door\_ws\_ distance} \leq 0 \quad (7.38)$$

39) The ratio of the vertical distance of the vertex to the line connecting the two flanges of A-pillar cross section over the length of the line should be smaller than a maximum value so that inner plate of the A-pillar fits into a cover (Fig.7.7) (98).

$$\textit{APVert/APD2f} - \textit{APRaMx (maximum value)} \leq 0 \quad (7.39)$$

where *APRaMx* is a maximum ratio. The user can define its value. A default value of 0.5 was used in this chapter.

- Manufacturing Constraints

Manufacturing constraints include stamping and welding constraints. One needs very detailed information about a joint to determine if a given design is feasible. This information is not available in the early design stage. This study uses crude equations to check if a design is manufacturable. Stamping constraints include constraints for strains, spring back, and die lock. To ensure that plastic strains are low, we impose low limits on the edges of each section, and the angles between two adjacent plates of a section. We also impose an upper limit on the depth of draw to avoid overstretching a plate (Fig.6.8a). Spring back constraints ensure that a plate can be permanently bent. For this reason, the deformation angle should exceed a minimum value (Fig. 7.8). Constraints on die lock ensure that the die and block can be separated after the plate has been stamped. For this purpose, the vertical walls of the hat-shape section should be slanted away from the center of the section (Fig. 6.7a).

40-44) The length of each edge on the header section should be larger than a minimum value so that the header top and bottom shells can be stamped (Fig. 7.9) (39-43).

$$R\_L\_Mi \text{ (a minimum value)} - H\_Lj \leq 0 \quad j=1,\ldots,5 \tag{7.40}$$

where *R_L_Mi* is a minimum length. The user can define this length. A default value of 5 mm was used in the examples of this chapter.

45-47) The angles between two adjacent edges on the header section must be larger than a minimum value to avoid a sharp angle between the plates (Fig. 7.10) (44-46).

$$R\_A\_Mi \text{ (a minimu value)} - H\_Aj \leq 0 \quad j=1,2,3 \tag{7.41}$$

where *R_A_Mi* is a minimum angle. The user can define this angle. A default value of 20 degrees was used in the examples of this chapter.

48-53) The length of each edge on the roof rail cross section must be larger than a minimum value because of stamping requirement (Fig. 7.11) (47-52).

$$R\_L\_Mi \text{ (a minimum value)} - RR\_Lj \leq 0 \quad j = 1,\ldots,6 \tag{7.42}$$

54-60) The angle between two adjacent edges on the roof rail cross section must be larger than a minimum value to avoid sharp angle between the plates (Fig. 7.12) (53-59).

$$R\_A\_Mi \text{ (a minimum value)} - RR\_Aj \leq 0 \quad j = 1,\ldots,7 \tag{7.43}$$

61-62) The spring back angles of the roof rail section must be larger than a minimum value (Fig. 7.13) (60-62). This constraint on spring back angle ensures that we can create permanent deformation

$$SpBkMi \text{ (a minimum value)} - RR\_Spj \leq 0 \quad j = 1,2,3 \tag{7.44}$$

The user can define the minimum value *SpBkMi*. A default value of 10 degrees was used.

63) The angle between the inner and outer sides of part 3 must be large enough to avoid die lock (Fig. 7.13) (63).

$$ADieMi \text{ (minimum value)} - RR\_AD1 \leq 0 \tag{7.45}$$

where *ADieMi* is a minimum angle. The user can define this angle. A default value of 3 degrees was used.

64-71) The length of each edge of the A-pillar cross section should be larger than the minimum value because of stamping requirements (Fig. 7.14) (64-71).

$$R\_L\_Mi \text{ (a minimum value)} - AP\_L_i \leq 0 \quad i = 1,\ldots,9 \tag{7.46}$$

72-84) The angle between two adjacent edges on the A-pillar cross section must be larger than a minimum value to avoid bending a plate at a sharp angle (Figs. 7.15 and 7.16). *AP_A7* should be smaller than 180 degrees (72-84).

$$R\_A\_Mi \text{ (a minimum value)} - AP\_Aj \le 0 \qquad j=1, \ldots,12 \tag{7.47}$$

$$AP\_A7 \le 180 \tag{7.48}$$

85-91) The spring back angles of the A-pillar cross section should be larger than a minimum allowable value to create a permanent shape (Fig. 7.17) (85-91).

$$SpBkMi \text{ (a minimum value)} - AP\_Spj \le 0 \qquad j=1,\ldots, 7 \tag{7.49}$$

92,93) The angle between the two side edges of the A-pillar should be larger than a minimum value to avoid die lock (Fig. 7.17) (92,93).

$$ADieMi \text{ (a minimum value)} - AP\_ADj \le 0 \qquad j=2,3 \tag{7.50}$$

94-96) The distance between the spot welds and the roots of flanges must exceed a minimum value so that the plates of the joint can be welded (Fig. 5.7) (94-96).

$$W\_R\_DMi \text{ (a minimum value)} - Flangej\_width + Distance\_from\_edge \le 0 \qquad j=1,2,3 \tag{7.51}$$

The user can define the minimum value *W_R_DMi*. A default value of 8 mm was used.

97) *Spot_weld_spacing* must exceed a minimum value because of manufacturing requirements (Fig. 5.7) (97).

$$\text{Minimum value of } spot\_weld\_spacing - spot\_weld\_spacing \le 0 \tag{7.52}$$

A default value of 35 mm was used for the minimum weld spacing.

- Styling Constraints

98,99) The difference between the length of the top piece of the door at the roof rail section and that at the A-pillar section should not be too big because of styling considerations (Fig. 7.18). *RAraMi* and *RAraMx* are the minimum and maximum ratios specified by the user. Their default values are 0.5 and 2.0, respectively (99,100).

$$RAraMi*AP\_edge - RR\_edge \leq 0 \quad (7.53)$$

$$RR\_edge - RAraMx *AP\_edge \leq 0 \quad (7.54)$$

100) The A-pillar should be slim and should not obstruct driver's vision (Fig.7.19) (117).

$$Blind\ angle - AInvMx\ (a\ maximum\ value) \leq 0 \quad (7.55)$$

The user can define the maximum value *AInvMx*. A default value of 10 degrees was used.

- Mathematical Constraints

101) The sum of *theta* and *phi* must be larger than 90 degrees to determine the orientation of A-pillar (Figs. 5.5 and 5.6) (1).

$$90 - (Theta+Phi\ ) \leq 0 \quad (7.56)$$

102) *H_window_depth* must be smaller than the overall height of header (Fig.5.8) (11).

$$H\_window\_depth - H\_height \leq 0 \quad (7.57)$$

103) The sum of *AP_inboard_depth* and *AP_window_depth* must be smaller than the overall height of A-pillar so that the shape of the A-pillar cross section is reasonable (Fig. 5.10) (38).

$$AP\_inboard\_depth + AP\_window\_depth - AP\_height \leq 0 \quad (7.58)$$

104-106) The length of the blending regions of a branch should be smaller than the corresponding offset distance from the origin of the global coordinate system to the cross section of that branch (Figs. 5.5 and 5.6) (4-6).

$$H\_blending\_rad - Header\_offset \leq 0 \quad (7.59)$$

$$RR\_blending\_rad - RR\_offset \leq 0 \quad (7.60)$$

$$AP\_blending\_rad - AP\_offset \leq 0 \quad (7.61)$$

107) The value of *sin(BRang)* should be less than 1.0 (Fig. 7.20) (12).

$$sBRang \leq 1.0 \quad (7.62)$$

where *sBRang=sin(BRang).*

108) Angle *RR_INang* should be positive (Fig. 7.3) (15).

$$-RR\_INang \leq 0 \quad (7.63)$$

109) *RR_edge* should be bigger than dimension *distance_from_edge* to be able to weld the parts 1 and 2 at the top (Figs. 5.7 and 5.9) (16).

$$RR\_h1 + Distance\_from\_edge*\cos(angle\_back) - RR\_width \leq 0 \quad (7.64)$$

110) *RR_h2* should be smaller than the overall width of roof rail cross section (Fig. 5.9) (17).

$$RR\_h2 - RR\_width \leq 0 \tag{7.65}$$

111,112) To make the angle between the side shell of door and the horizontal direction in the range [-90,90]. Therefore, the absolute value of the cosine of this angle, *AP_c*, should be less than 1.0 (Fig. 7.5) (24,25).

$$-1.0 - AP\_c \leq 0 \tag{7.66}$$

$$AP\_c - 1.0 \leq 0 \tag{7.67}$$

113,114) *A_H*, and *A_V* should be positive so that the A-pillar reinforcement has reasonable shape (Fig. 7.4) (33,34).

$$-A\_H \leq 0 \tag{7.68}$$

$$-A\_V \leq 0 \tag{7.69}$$

115) The projection of *A_pillar_blending_rad* on global z-axis should be bigger than the horizontal distance measured from the global origin to the edge of flange 1 at the cross section of the header (Fig. 7.21) (35).

Projection of A_pillar_blending_rad on z axis – projection of distance between global origin and edge of flange 1 on z ≤ 0 (7.70)

116,117) Both *AP_d1*, and *AP_h2* should be positive so that the shape of the A-pillar is reasonable (Fig. 7.4) (36,37).

$$-AP\_d1 \leq 0 \tag{7.71}$$

$$-AP\_h2 \leq 0 \tag{7.72}$$

- Performance Targets

  118-120) The three stiffnesses of the design should be larger than the minimum values.

$$(\hat{K}_{I/O} - K_{I/O}) / \overline{K}_{I/O} \le 0 \tag{7.73}$$

$$(\hat{K}_{F/A} - K_{F/A}) / \overline{K}_{F/A} \le 0 \tag{7.74}$$

$$(\hat{K}_{Tor} - K_{Tor}) / \overline{K}_{Tor} \le 0 \tag{7.75}$$

7.3 Results and Discussion

The above optimization problem was solved several times using the nonlinear optimization program DOT (1995). First, we checked if the optimization program converged to a global minimum. Then, we validated the optimization program by comparing the optimization results with FEA results. Finally, results obtained using RSP and NN translators were compared.

7.3.1 Checking the Convergence of the Optimization Program

We can use three methods to check whether the optimization program converges to a global minimum.

- We solve the problem from different initial points to see if the optimization program converges to the same optimum design.
- We use the final optimum design as an initial point, and solve the problem again to see if the design changes.
- We also solve the same problem using both the *modified feasible direction* method (MFD) and the *sequential linear programming* (SLP) to see if the optimization program converges to the same point.

Eight randomly generated designs were used as initial points. The measured stiffness of an actual car joint was used as target. We only considered the mass in the objective function. E.g., α in Eq.7.1 was set to be one. Table 7.2 shows the objective functions of

the optimum designs when we start from different initial points. We used SLP to obtain the above results. Both RSP and NN translators were tested. It is observed that the maximum relative difference between the masses of optimum designs is 0.3%. The first 2 to 3 digits of the objective function are the same when starting from different initial designs. In general, the first 2 to 3 digits of design variables of the final designs are found to be the same. Practically, no improvement (less than 0.3%) was achieved by solving the problem again starting from the obtained optimum designs. Finally, it is observed that the optimum design from the NN translator has about 7% larger mass than the designs obtained from the RSP translator. This observation is the same as that for the B-pillar to rocker joint.

We also used the modified feasible direction method (MFD) to check the convergence of the optimization program by starting from different initial points. We found that the optimum design obtained using the MFD method was practically the same as that obtained using the SLP method. We chose SLP as the default optimization method in the translator B for the A-pillar to roof rail joint because the SLP method needed fewer iterations than the MFD method. All the optimization results presented in this chapter were obtained using SLP method. The user can easily switch to the MFD method by changing a variable in the optimization program of translator B.

7.3.2 Comparison of the Results of Translator B with FEA Results

To validate translator B, we compared the optimization results with FEA results. First, we randomly generated 12 sets of performance targets (Each set contained minimum values for I/O, F/A, and torsion stiffness). Using translator B, we obtained the optimum design corresponding to each set of stiffness requirements. The optimum designs were visually checked using the Pro/Engineer model, and MSC/NASTRAN bulk date files were created. Each model was analyzed and its performance characteristics (stiffnesses and mass) were obtained. The optimization results were compared with the FEA results. Table 7.3a and Figures 7.22a – 7.22d show the results for the I/O, F/A, torsion stiffnesses and the mass when RSP translators were used. The values of design variables of four optimum designs are shown in Table 7.3b. Table 7.3c and Figures 7.23a-7.23d show the results when NN translators were used. Table 7.3d shows the

values of design variables of four optimum designs. Table 7.3e shows the correlation coefficients of the 12 designs for the RSP and NN translators. The cross sections of the initial and the optimum designs are compared in Fig. 7.24.

By observing the optimum designs, we found the following trends when optimizing the shape of the joint components:

A-pillar

In general, the size of the section of the A-pillar increases (Fig. 7.24). The shape of the A-pillar section approaches to the shape of a rectangle. The following trends about the values of the design variables are observed (Tables 7.3b and 7.3d)

- *AP_inboard_width* reaches its upper bound. This makes the A-pillar section close to a square, and thus increases the stiffness of A-pillar.
- *AP_reinf_inner_door_allowance* is close to its upper bound.
- *AP_reinf_flange2_angle* assumes a value that makes the outer piece of A-pillar reinforcement almost vertical. Thus, the A-pillar reinforcement approaches the outer shell of the A-pillar. This increases the cross section area and the stiffness of A-pillar.
- *Angleback* takes its minimum allowable value that is determined by the constraint on spring back angle.
- *AP_blending_rad* reaches its lower bound for most designs to reduce the mass of the joint.

Header

- *H_width*, *H_window_depth* (Fig. 5.8) and *H_blending_rad* (Fig. 5.6) are close to their lower bounds at the optimum. The optimizer tends to reduce the cross section of header to reduce the mass of the joint. This shows that the header branch has higher stiffness compared with the A-pillar branch and it has little effect on the overall stiffness of the joint.

Roof rail

- *RR_width* reaches its lower bound. *RR_angle_bottom* is close to its upper bound. Most dimensions at the roof rail cross section take values between their lower and upper bounds. This shows that the roof rail section is determined by a trade-off between high stiffness and low mass.

Global dimensions

- *Flange1_angle_down* (Fig. 5.10) reaches its lower bound. As a result of this trend, the inner plate of part 3 tends to be vertical, which increases the cross section of the A-pillar and the stiffness of the joint (Fig. 7.24).

Extrusion/cutting distance

- *Part3_cut_distance* and *part4_extension* take their minimum allowable values. Since part 3 is thicker than part 4, this will increase the thickness at the bottom of header.
- *Part7_cut_distance* reaches its upper bound. This indicates that part 7 (reinforcement) contributes little to the stiffness of header. The optimizer tries to increase this dimension to reduce the mass of the joint.

Thickness

- All thicknesses reach their lower bounds. This is reasonable because the optimizer tries to reduce the thickness of each part, and increase the cross section of the A-pillar to achieve the maximum stiffness with the smallest increase of mass.

From the comparison of the results of translator B and the FEA results (Figs.7.22a-7.23d), we observe that translator B predicts the stiffness and mass accurately when the stiffness requirements are lower ($\hat{K}_{I/O} \le 6.5E7Nmm, \hat{K}_{Tor} \le 1.0E7Nmm$, Figs. 7.22a-7.23d). The F/A stiffness requirement, $\hat{K}_{F/A}$, does not significantly affect the correlation between the results of translator B and FEA results. Both RSP and NN translators predict F/A stiffness accurately. The RSP translator slightly overestimates the F/A stiffness for most designs. On the other hand, the NN translator slightly underestimates the F/A

stiffness for most designs. Both RSP and NN translators B underestimate the mass of the optimum design. E.g., the predicted mass of translator B is lower than the actual mass of the joint. The difference between the predictions of translator B and the FEA results increases when the stiffness requirements are high ($\hat{K}_{I/O} > 6.5E7Nmm, \hat{K}_{Tor} > 1.0E7Nmm$). Both the RSP and NN translators B tend to overestimate the stiffness of joint when the stiffness requirements are high. For the RSP translator B, the error of the predicted I/O stiffness can be 24.7% higher than the FEA result (design 6), and error of the predicted torsion stiffness can be 51% higher than the FEA result (design 6). For the NN translator B, the errors of predicted I/O and torsion stiffnesses can be as high as 30.8% (design 12) and 24.3% (design 11), respectively. Both the RSP and NN translators B predict the F/A stiffness and the mass accurately. The errors of the predicted F/A stiffness and the mass are less than 10% and 5%, respectively.

The correlation coefficient defined in Eq.6.62 is used to assess the correlation between the FEA and optimization results. Table 7.3e shows the correlation coefficients for the stiffnesses and mass when RSP and NN translators are used. It is found that the correlation coefficients for RSP translators range from 0.8074 to 0.9964. For NN translators, the correlation coefficients range from 0.8919 to 0.9967. The NN translator predicts torsion stiffness more accurately than RSP translator. RSP translator predicts the mass with slightly better accuracy. The predictions of I/O and F/A stiffnesses from RSP and NN translators are equally accurate. Overall, the testing results from NN translators are slightly better than those from RSP translators for the 12 designs considered.

By comparing the results obtained using the RSP and NN translators, we observe that the mass of the optimum design obtained using NN translators is typically about 7% higher than that obtained using RSP translators.

7.3.3 Redesign of the Joints of two Cars Using Translator B

We used translator B to redesign two actual joints. The measured stiffnesses of the two joints were used as targets. Since the shape of the outer shells of A-pillar to roof rail joint (parts 1 and 2) are generally determined by styling requirements, we fixed the

design variables that define the outer shape of A-pillar joint at their values for the actual joint, and optimized only those design variables that define the inner part and reinforcement of the joint. Tables 7.4a and 7.4c show the states and ranges of the design variables of cars A and B. Tables 7.4b and 7.4d present the dimensions of the optimum designs obtained using translator B. Tables 7.4e and 7.4f compare the optimization results with FEA results. We observed that the predictions for F/A stiffness and mass are close to the FEA results if the RSP translators are used. The errors are less than 3% and 2%, respectively. The predictions of torsion stiffness are worse than those for F/A stiffness and mass. The errors are 14.8% and 20.1% for the two cars, respectively. The prediction of I/O stiffness correlates well with FEA results for car A (error is -1.9%), but correlates poorly for car B (error is 16.8%). The NN translators predict the I/O stiffness and mass accurately. The biggest errors are -4.7% and -4.2%, respectively. The errors of predictions of the F/A and torsion stiffnesses for car A are –10.8% and –3.5%, respectively. For car B, they are –6.2% and –11.0%, respectively.

Table 7.4f compares the masses of the initial and optimum designs of the two cars. Because we fixed design variables defining the outer shape of the A-pillar joint, the improvement of the optimum design compared with the initial design was small. When RSP translators were used, 7.2% and 11.2% improvement were achieved through optimization for car A and car B. When using NN translators, the improvement for car B was 7.8%. There was practically no improvement for car A.

7.3.4 Parametric Study

Figures 7.25 –7.27 show the effects of the I/O, F/A, and torsion stiffness requirements on the mass of the optimum design. To get the above results, we only changed one stiffness requirement at a time and fixed the other two stiffness requirements at the values corresponding to an actual joint. It is observed that, when the required stiffnesses are low, the mass of the optimum design is almost constant or increases very little. The mass begins to increase at a considerable rate with the stiffness after a certain point. We also observe that torsion stiffness requirement has bigger effect on the mass of the optimum designs than the I/O and F/A stiffness requirements.

As mentioned in Section 7.2.3, we fixed the lengths of the three branches of A-pillar joint to obtain the previous results. We also studied the effects of the lengths of each branch on the mass of the optimum design. Figures 7.28-7.30 show the effect of lengths of header, roof rail, and A-pillar branches on the mass. It is found that increasing the length of a branch increases the mass of the optimum design.

Figures 7.31 and 7.32 show the effects of *theta* and *phi*, which define the orientation of the A-pillar, on the mass. It is found that the mass of the optimum designs is insensitive to changes in these angles.

We observed in Section 7.3.2 that the thicknesses of several parts reached their lower bounds. Figures 7.33 to 7.36 show the relations between the mass of the optimum design and the lower bounds of thicknesses of parts 1, 2, 3, and 7. We observe that increasing the lower bound of thickness increases the mass of the optimum design.

Figure 7.37 shows the effect of increasing the lower bound of *RR_width* on the mass of the optimum design. As we observed in section 7.3.2, *RR_width* of the optimum design reached its lower bound. Therefore, increasing its lower bound increases the mass of the optimum design because the region in which the optimizer could search decreases.

Door_allowance took a value between its lower and upper bounds at the optimum design. Figures 7.38 and 7.39 show the effects of increasing the lower bound or decreasing the upper bound of *door_allowance,* respectively, on the mass of the optimum design. As expected, both increase the mass of the optimum design. It is also found that *door_allowance* contributes little to the mass.

Figures 7.40 and 7.41 show the relations between the mass of the optimum design and the lower bounds of *H_width* and *AP_blending_rad.* Since both design variables are close to their lower bounds for the optimum design, increasing their lower bounds increases the mass of the optimum design.

Finally, it is observed from Figs. 7.25-41 that the NN translator yields heavier designs than the RSP translator for the same stiffness targets.

7.3.5 Discussion of Results

Some observations of this study should help designers improve the joint design. For example, this study found that the dimensions of the header branch had little effect on the stiffness of the joint. Therefore, reducing the cross section of header reduces the mass of the joint without reducing the stiffness of the joint. We found that the thickness of every part reached its minimum. As a result of this trend, the mass of the joint decreased. It was also found that most design variables defining the cross section of the roof rail assumed values between their lower and upper bounds. This shows that those design variables were determined by a trade-off between the high stiffness and low mass.

In general, the translator B increased the size of the cross section of the A-pillar and reduced the size of the cross sections of other two beams. The reason is that the stiffness of the joint is more sensitive to the properties of the A-pillar than the properties of the other beams. Moreover, the properties of all three beams affect the mass.

By comparing the results of the translator B with FEA results, we observed that both the RSP and NN translators B predicted the stiffness and mass accurately when the stiffness requirements were lower. The difference between the predictions of the translators and the FEA results increased as the stiffness requirements increased. Both RSP and NN translators B tended to overestimate the stiffnesses, and underestimate the mass of the optimum design when the stiffness requirements were high.

It was found that the NN translator predicted torsion stiffness more accurately than the RSP translator. On the other hand, the RSP translator gave slightly more accurate predictions of the mass. The predictions of the I/O and F/A stiffnesses from RSP and NN translators were equally accurate.

The above observations are the same as the conclusion of Chapter 5 for the I/O, F/A stiffnesses and the mass, but different from the conclusion of Chapter 5 for the torsion

stiffness. Chapter 5 found that, the RSP translator A was slightly more accurate for the mass. The RSP and NN translators A predicted I/O and F/A stiffnesses equally accurately. The RSP translator A predicted the torsion stiffness more accurately than the NN translator A. However, Chapter 7 found that the prediction of translator B correlated better with the FEA results when the NN translator A was used than when the RSP translator A was used. The conclusion of Chapter 5 was drawn based on the 600 designs in the database for developing translator A. Since we fixed the lengths of the three branches when developing the translator B, only a portion of the 600 designs were in the feasible domain of the translator B. Therefore, the conclusion of this chapter that was drawn from a portion of the 600 designs may be different from the overall conclusion in Chapter 5.

It was found that the NN translator yielded heavier designs than the RSP translator. This is because the mass of the optimum design was mainly determined by the torsion stiffness requirement. The difference between the predicted torsion stiffness and the FEA results was bigger when the RSP translator was used than when the NN translator was used. The error of the predicted torsion stiffness from the RSP translator B was much higher than that from the NN translator B when the torsion stiffness requirement was high. According to these results, we recommend to use the NN translator B for the A-pillar to roof rail joint.

This chapter observed the same trend as that in Chapter 6 for the mass. That is, translator B underestimated the mass of the optimum design. On the other hand, the general trends observed in this chapter are different from those in Chapter 6 for the I/O, F/A and torsion stiffness. In Chapter 6, we found that when the RSP translators were used, the translator B overestimated the stiffness when the stiffness requirements were lower, and underestimated the stiffness when the stiffness requirements were high. When NN translators were used, the translator B tended to overestimate the I/O stiffness when the stiffness requirements were low, but underestimated the I/O stiffness when the stiffness requirements were high. In general, the NN translators B underestimated the F/A and torsion stiffnesses. We found that the RSP translators B gave slightly better predictions than the NN translator B in Chapter 6. The difference of the conclusions of

this chapter and Chapter 6 is due to several reasons. First, B-pillar to rocker joint and A-pillar to roof rail joint are completely different. A-pillar to roof rail joints vary more from joint to joint than B-pillar to rocker joints. Second, since translator B uses translator A to obtain the responses of a new design in optimization, the accuracy of translator A affects significantly the accuracy of translator B. The optimum design of the B-pillar to rocker joint is mainly determined by the F/A stiffness requirement. Both RSP and NN translators A predict accurately the F/A stiffness of the B-pillar to rocker joint. For the A-pillar to roof rail joint, the predictions of both the RSP and NN translators A are less accurate than those for the B-pillar to rocker joint. The optimizer tends to take advantage of the error in the predictive models (translator A). Therefore, the translator B for the A-pillar to roof rail joint is less accurate than that for the B-pillar to rocker joint. Third, when we create the database for developing the translator A of A-pillar joint, we leave the lengths of the three branches be design variables because there is no consistent definition of the length of branches of joint. Since the lengths of the three branches of the A-pillar to roof rail joint affect significantly the stiffnesses and the mass (See Tables 5.5-5.8), considering the lengths of the three branches as design variables allows the translator A to cover a large range of joint designs. However, this sacrifices the accuracy of the translator A for the A-pillar to roof rail joint. When we develop translator B for A-pillar, we fix the lengths of the three branches in order to compare the stiffnesses of different designs fairly. The predictions of translator B would be more accurate if we fixed these three lengths when developing the database and translator A.

Table 7.1: Ranges and States of Design Variables

Seq. No	Name of Design Variables	Bounds (mm,degree)		State
		Lower	Upper	
1	Distance_from_edge	8.0	8.0	## Fixed by program
2	Header_offset	260.00	260.00	## Dependent
3	Roof_rail_offset	105	196	## Dependent
4	A_pillar_offset	163	240	## Dependent
5	Theta	23.00	23.00	# Fixed by user
6	Phi	76.00	76.00	# Fixed by user
7	Header_horizontal_offset	15	78	Independent
8	Flange1_width	20	20	## Fixed by program
9	Flange2_width	16	16	## Fixed by program
10	Flange3_width	16	16	## Fixed by program
11	Spot_weld_spacing	50	50	## Fixed by program
12	Flange1_angle_up	90	113	Independent
13	Flange1_angle_down	90	110	Independent
14	Flange2_angle_up	90	130	Independent
15	Flange2_angle_down	96	123	Independent
16	Flange3_angle_down	90	107	Independent
17	Door_allowance	27	63	Independent
18	Angleback	8	46	Independent
19	H_width	78	109	Independent
20	H_height	19	34	Independent
21	H_window_depth	12	17	Independent
22	H_blending_rad	96	210	Independent
23	RR_width	56	146	Independent
24	RR_height	47	79	Independent
25	RR_h1	39	108	Independent
26	RR_h2	35	107	Independent
27	RR_angle_bottom	20	42	Independent
28	RR_blending_rad	53	108	Independent
29	AP_height	52	78	Independent
30	AP_inboard_width	20	33	Independent
31	AP_inboard_depth	25	36	Independent
32	AP_door_ws_distance	19	35	Independent
33	AP_window_depth	17	26	Independent
34	AP_flange1_angle	0	0	## Fixed by program
35	AP_reinf_depth	1	9	Independent
36	AP_reinf_flange1_angle	91	155	Independent
37	AP_reinf_inner_door_ allowance	22	38	Independent
38	AP_reinf_flange2_angle	91	116	Independent
39	Ap_blending_rad	55	138	Independent

40	Part2_cut_distance	55	110	Independent
41	Part3_cut_distance	24	56	Independent
42	Part4_extension	46	90	Independent
43	Part7_cut_distance	44	90	Independent
44	Thickness of Part1	0.81	1.19	Independent
45	Thickness of Part2	1.03	1.14	Independent
46	Thickness of Part3	1.31	1.68	Independent
47	Thickness of Part4	1.00	1.07	Independent
48	Thickness of Part7	1.53	1.83	Independent

Lengths of header, roof rail and A-pillar branches are fixed at 260 mm, 213 mm and 161mm, respectively.

Table 7.2: Comparison of Optimum Results
When Starting From Different Initial Points
($K_{I/O} > 3.16E7Nmm, K_{F/A} > 7.93E7Nmm, K_{Tor} > 1.38E7Nmm$)

Initial Design No.	Using RSP Translators		Using NN Translators	
	No. of Function Evaluations	Object Function (kg)	No. of Function Evaluations	Objective Function (kg)
1	574	2.3380	537	2.4914
2	344	2.3433	804	2.4917
3	497	2.3385	727	2.4914
4	500	2.3381	695	2.4914
5	499	2.3382	614	2.4916
6	458	2.3428	538	2.4915
7	575	2.3384	2403	2.4918
8	423	2.3383	619	2.4920

Table 7.3a: Comparison of Optimization and FEA Results
for A-pillar to Roof Rail Joint(using RSP Translators)

Set No.	Stiffness Requirement (Nmm)			Optimization Results (Nmm,kg)				FEA Results (Nmm,kg)			
	I/O $\times 10^7$)	F/A $\times 10^7$	Tor $\times 10^7$	I/O $\times 10^7$	F/A $\times 10^7$	Torsion $\times 10^7$	Mass	I/O $\times 10^7$	F/A $\times 10^7$	Torsion $\times 10^7$	Mass
1	3.5	5.25	0.75	5.8419	5.2511	0.7500	2.1491	5.8401	5.0244	0.8474	2.2473
2	6.5	6.75	1.00	6.8226	6.7501	1.0000	2.2377	6.8802	6.3976	0.9311	2.3442
3	5.7	6.0	0.8	5.7024	6.0018	0.8003	2.1877	5.7973	5.8579	0.9137	2.2910
4	6.0	8.25	0.8	6.0012	8.2501	0.9857	2.2925	6.3396	8.2430	1.0048	2.4004
5	5.0	5.0	0.9	6.0015	5.0007	0.9000	2.1689	5.8833	4.9933	0.8379	2.2796
6	7.5	9.0	1.8	9.3395	9.0000	1.8001	2.4828	7.4874	8.6355	1.1917	2.5443
7	4.25	4.75	0.7	5.6450	4.7500	0.7000	2.1224	5.5783	4.4314	0.6969	2.2187
8	5.5	6.5	0.85	5.8831	6.5001	0.8500	2.2138	6.1164	6.4338	0.9684	2.3199
9	7.25	10.	1.4	7.2816	10.0020	1.4006	2.3897	7.2926	9.1500	1.2760	2.4785
10	5.1	5.9	0.85	6.2110	5.9000	0.8501	2.1868	6.2039	5.6297	0.8365	2.2887
11	3.4	8.0	1.43	7.1681	7.9994	1.4298	2.3549	6.7341	7.7269	1.0084	2.4383
12	3.4	5.9	1.1	7.2490	5.8996	1.0998	2.2312	6.7114	5.4363	0.8483	2.3261

Table 7.3b: Values of Design Variables of Four Optimum Designs From RSP Translator

No	Name of Design Variables	Design and Value of Design Variable (mm,degree)			
		1	5	9	12
1	Distance_from_edge	8.00	8.00	8.00	8.00
2	Header_offset	260.00	260.00	260.00	260.00
3	Roof_rail_offset	181.65	181.65	190.45	196.00
4	A_pillar_offset	189.86	189.86	181.76	176.64
5	Theta	23.00	23.00	23.00	23.00
6	Phi	76.00	76.00	76.00	76.00
7	Header_horizontal_offset	31.35	31.35	22.55	17.00
8	Flange1_width	20.00	20.00	20.00	20.00
9	Flange2_width	16.00	16.00	16.00	16.00
10	Flange3_width	16.00	16.00	16.00	16.00
11	Spot_weld_spacing	50.00	50.00	50.00	50.00
12	Flange1_angle_up	112.91	92.35	113.00	103.84
13	Flange1_angle_down	90.00	90.00	90.00	90.04
14	Flange2_angle_up	115.00	114.91	114.97	114.97
15	Flange2_angle_down	96.05	96.00	96.06	96.22
16	Flange3_angle_down	106.95	107.00	106.97	106.95
17	Door_allowance	37.94	39.02	57.19	42.89
18	Angleback	10.00	10.00	10.00	10.00
19	H_width	78.05	78.01	78.06	78.01
20	H_height	19.00	19.00	19.00	19.00
21	H_window_depth	12.27	12.26	12.22	12.26
22	H_blending_rad	96.03	96.02	100.27	96.00
23	RR_width	56.00	56.50	56.52	56.49
24	RR_height	53.53	54.84	73.23	58.63
25	RR_h1	43.24	39.00	39.01	39.02
26	RR_h2	55.99	56.49	56.52	56.49
27	RR_angle_bottom	37.58	39.21	41.98	41.99
28	RR_blending_rad	78.70	78.66	87.51	93.01
29	AP_height	55.35	52.00	71.89	52.01
30	AP_inboard_width	33.00	32.99	33.00	33.00
31	AP_inboard_depth	26.30	25.01	36.00	25.02
32	AP_door_ws_distance	25.50	34.99	35.00	34.97
33	AP_window_depth	17.00	21.38	26.00	24.60
34	AP_flange1_angle	0.00	0.00	0.00	0.00
35	AP_reinf_depth	9.00	7.99	7.98	9.00
36	AP_reinf_flange1_angle	113.91	114.00	119.42	113.95
37	AP_reinf_inner_door_ allowance	34.44	31.74	37.99	33.17

38	AP_reinf_flange2_angle	116.00	115.91	115.97	115.97
39	Ap_blending_rad	55.01	55.02	55.03	55.02
40	Part2_cut_distance	110.00	110.00	110.00	110.00
41	Part3_cut_distance	26.00	26.00	26.00	26.00
42	Part4_extension	46.00	46.00	46.00	46.00
43	Part7_cut_distance	90.00	89.91	90.00	90.00
44	Thickness of Part1	0.81	0.81	0.81	0.81
45	Thickness of Part2	1.03	1.03	1.03	1.03
46	Thickness of Part3	1.31	1.31	1.31	1.31
47	Thickness of Part4	1.00	1.00	1.00	1.00
48	Thickness of Part7	1.53	1.53	1.53	1.53

Table 7.3c: Comparison of Optimization and FEA Results for A-pillar to Roof Rail Joint(using NN Translators)

No	Stiffness Requirement (Nmm)			Optimization Results (Nmm,kg)				FEA Results (Nmm,kg)			
	I/O $\times 10^7$	F/A $\times 10^7$	Tor $\times 10^7$	I/O $\times 10^7$	F/A $\times 10^7$	Torsion $\times 10^7$	Mass	I/O $\times 10^7$	F/A $\times 10^7$	Torsion $\times 10^7$	Mass
1	3.5	5.25	0.75	5.3399	5.2503	0.7505	2.2756	5.6092	5.5042	0.8079	2.3966
2	6.5	6.75	1.00	6.4990	6.7492	1.0000	2.3655	6.7411	7.0129	0.9560	2.4518
3	5.7	6.0	0.8	5.6990	6.0001	0.8070	2.3134	6.1530	6.3124	0.8459	2.4194
4	6.0	8.25	0.8	5.9985	8.2526	0.9997	2.4072	6.3693	8.7465	1.0502	2.4777
5	5.0	5.0	0.9	7.2137	4.9990	0.9004	2.3080	6.4926	5.4398	0.8649	2.4096
6	7.5	9.0	1.8	9.7608	9.1894	1.8005	2.6286	8.8787	9.2253	1.6295	2.6582
7	4.25	4.75	0.7	5.2906	4.7514	0.7002	2.2515	5.4101	5.0016	0.7654	2.3668
8	5.5	6.5	0.85	5.4997	6.4997	0.8500	2.3347	5.8346	6.9671	9.1464	2.4371
9	7.25	10.	1.4	7.5898	10.0020	1.3998	2.5211	7.2727	10.2960	1.3939	2.5965
10	5.1	5.9	0.85	5.3679	5.8990	0.8501	2.3116	5.5333	6.3184	0.8786	2.4127
11	3.4	8.0	1.43	9.1975	8.3037	1.4297	2.5074	7.3289	8.3920	1.1501	2.5270
12	3.4	5.9	1.1	8.8012	5.9000	1.1000	2.3892	6.7297	6.1323	0.9872	2.4481

Table 7.3d: Values of Design Variables of Four Optimum Designs From NN Translators

No	Name of Design Variables	Design and Value of Design Variable (mm,degree)			
		1	5	9	12
1	Distance_from_edge	8.00	8.00	8.00	8.00
2	Header_offset	260.00	260.00	260.00	260.00
3	Roof_rail_offset	181.77	193.72	181.66	196.00
4	A_pillar_offset	189.75	178.75	189.85	176.65
5	Theta	23.00	23.00	23.00	23.00
6	Phi	76.00	76.00	76.00	76.00
7	Header_horizontal_offset	31.23	19.28	31.34	17.00
8	Flange1_width	20.00	20.00	20.00	20.00
9	Flange2_width	16.00	16.00	16.00	16.00
10	Flange3_width	16.00	16.00	16.00	16.00
11	Spot_weld_spacing	50.00	50.00	50.00	50.00
12	Flange1_angle_up	112.92	112.90	113.00	112.93
13	Flange1_angle_down	90.04	90.02	90.03	90.00
14	Flange2_angle_up	114.95	114.94	114.89	111.88
15	Flange2_angle_down	96.02	96.04	96.02	105.44
16	Flange3_angle_down	90.03	90.09	90.08	90.01
17	Door_allowance	46.59	56.78	52.62	54.17
18	Angleback	10.00	10.00	10.00	10.00
19	H_width	78.00	78.00	78.00	78.05
20	H_height	30.00	30.00	30.00	30.00
21	H_window_depth	13.92	13.92	13.92	13.92
22	H_blending_rad	96.00	96.00	96.09	96.01
23	RR_width	56.03	56.04	56.00	56.04
24	RR_height	61.62	71.74	67.62	69.29
25	RR_h1	46.53	46.53	46.49	41.50
26	RR_h2	56.03	56.03	55.99	56.04
27	RR_angle_bottom	42.00	42.00	42.00	42.00
28	RR_blending_rad	108.00	107.99	108.00	94.03
29	AP_height	54.09	73.30	62.92	62.61
30	AP_inboard_width	32.67	33.00	32.93	32.97
31	AP_inboard_depth	32.48	34.08	35.99	36.00
32	AP_door_ws_distance	19.00	19.00	19.00	29.07
33	AP_window_depth	17.01	25.99	19.85	23.11
34	AP_flange1_angle	0.00	0.00	0.00	0.00
35	AP_reinf_depth	5.05	3.82	2.11	8.99
36	AP_reinf_flange1_angle	114.02	130.34	155.00	119.75
37	AP_reinf_inner_door_ allowance	37.99	37.96	37.99	38.00

38	AP_reinf_flange2_angle	115.95	115.94	115.89	113.26
39	Ap_blending_rad	55.03	55.01	55.03	124.19
40	Part2_cut_distance	58.91	109.94	79.14	110.00
41	Part3_cut_distance	26.00	26.00	26.00	26.00
42	Part4_extension	46.00	46.00	46.00	46.00
43	Part7_cut_distance	90.00	90.00	90.00	90.00
44	Thickness of Part1	0.81	0.81	0.81	0.81
45	Thickness of Part2	1.03	1.03	1.03	1.03
46	Thickness of Part3	1.31	1.31	1.31	1.31
47	Thickness of Part4	1.00	1.00	1.00	1.00
48	Thickness of Part7	1.53	1.53	1.53	1.53

Table 7.3e: Comparison of Correlation Coefficients
Obtained using RSP and NN Translators

Stiffness/Mass	RSP translators	NN translator
I/O	0.8984	0.8919
F/A	0.9919	0.9967
Torsion	0.8074	0.9634
Mass	0.9964	0.9841

Table 7.4a: State and Ranges of Design Variables of Car A

No	Name of Design Variables	State	Car A Bounds (mm,degree)	
			Lower	Upper
1	Distance_from_edge	## Fixed by program	8.00	8.00
2	Header_offset	## Dependent	202.00	202.00
3	Roof_rail_offset	## Dependent	165.00	165.00
4	A_pillar_offset	## Dependent	163.00	163.00
5	Theta	# Fixed by user	23.00	23.00
6	Phi	# Fixed by user	76.00	76.00
7	Header_horizontal_offset	# Fixed by user	15.00	15.00
8	Flange1_width	## Fixed by program	20.00	20.00
9	Flange2_width	## Fixed by program	16.00	16.00
10	Flange3_width	## Fixed by program	16.00	16.00
11	Spot_weld_spacing	## Fixed by program	50.00	50.00
12	Flange1_angle_up	# Fixed by user	90.00	90.00
13	Flange1_angle_down	Independent	90.00	121.00
14	Flange2_angle_up	# Fixed by user	90.00	90.00
15	Flange2_angle_down	Independent	90.00	135.00
16	Flange3_angle_down	Independent	90.00	118.00
17	Door_allowance	# Fixed by user	38.00	38.00
18	Angleback	# Fixed by user	30.00	30.00
19	H_width	# Fixed by user	80.00	80.00
20	H_height	Independent	17.00	37.00
21	H_window_depth	# Fixed by user	16.00	16.00
22	H_blending_rad	# Fixed by user	95.00	95.00
23	RR_width	# Fixed by user	65.00	65.00
24	RR_height	Independent	43.00	87.00
25	RR_h1	# Fixed by user	40.00	40.00
26	RR_h2	Independent	32.00	118.00
27	RR_angle_bottom	Independent	18.00	46.00
28	RR_blending_rad	# Fixed by user	90.00	90.00
29	AP_height	Independent	47.00	86.00
30	AP_inboard_width	# Fixed by user	26.00	26.00
31	AP_inboard_depth	Independent	23.00	40.00
32	AP_door_ws_distance	# Fixed by user	35.00	35.00
33	AP_window_depth	# Fixed by user	22.00	22.00
34	AP_flange1_angle	# Fixed by user	0.00	0.00
35	AP_reinf_depth	Independent	1.00	29.00
36	AP_reinf_flange1_angle	Independent	91.00	171.00
37	AP_reinf_inner_door_ allowance	Independent	20.00	69.00
38	AP_reinf_flange2_angle	Independent	91.00	144.00

39	Ap_blending_rad	# Fixed by user	55.00	55.00
40	Part2_cut_distance	Independent	50.00	121.00
41	Part3_cut_distance	Independent	22.00	62.00
42	Part4_extension	Independent	42.00	99.00
43	Part7_cut_distance	Independent	40.00	121.00
44	Thickness of Part1	# Fixed by user	0.94	0.94
45	Thickness of Part2	# Fixed by user	1.02	1.02
46	Thickness of Part3	Independent	0.81	1.85
47	Thickness of Part4	Independent	0.81	1.18
48	Thickness of Part7	Independent	1.06	2.01

The lengths of header, roof rail, and A-pillar branches are fixed at 202 mm, 180 mm, and 149.2 mm, respectively.

Table 7.4b: Optimum Design for Car A

No	Name of Design Variables	Optimum Design for Car 1 (mm,degree)	
		Use RSP	Use NN
1	Distance_from_edge	8.00	8.00
2	Header_offset	202.00	202.00
3	Roof_rail_offset	165.00	165.00
4	A_pillar_offset	163.01	163.01
5	Theta	23.00	23.00
6	Phi	76.00	76.00
7	Header_horizontal_offset	15.00	15.00
8	Flange1_width	20.00	20.00
9	Flange2_width	16.00	16.00
10	Flange3_width	16.00	16.00
11	Spot_weld_spacing	50.00	50.00
12	Flange1_angle_up	90.00	90.00
13	Flange1_angle_down	90.00	90.00
14	Flange2_angle_up	90.00	90.00
15	Flange2_angle_down	135.00	90.07
16	Flange3_angle_down	117.97	92.91
17	Door_allowance	38.00	38.00
18	Angleback	30.00	30.00
19	H_width	80.00	80.00
20	H_height	21.00	30.00
21	H_window_depth	16.00	16.00
22	H_blending_rad	95.00	95.00
23	RR_width	65.00	65.00
24	RR_height	55.73	67.36
25	RR_h1	40.00	40.00
26	RR_h2	56.92	61.94
27	RR_angle_bottom	42.03	45.20

28	RR_blending_rad	90.00	90.00
29	AP_height	59.73	61.98
30	AP_inboard_width	26.00	26.00
31	AP_inboard_depth	32.30	36.71
32	AP_door_ws_distance	35.00	35.00
33	AP_window_depth	22.00	22.00
34	AP_flange1_angle	0.00	0.00
35	AP_reinf_depth	1.00	1.00
36	AP_reinf_flange1_angle	170.01	168.57
37	AP_reinf_inner_door_allowance	37.03	31.47
38	AP_reinf_flange2_angle	91.00	91.00
39	Ap_blending_rad	55.00	55.00
40	Part2_cut_distance	121.00	50.00
41	Part3_cut_distance	22.01	22.00
42	Part4_extension	42.01	42.00
43	Part7_cut_distance	121.00	121.00
44	Thickness of Part1	0.94	0.94
45	Thickness of Part2	1.02	1.02
46	Thickness of Part3	1.73	1.68
47	Thickness of Part4	0.91	0.91
48	Thickness of Part7	1.13	1.17

Table 7.4c: States and Ranges of Design Variables of Car B

No	Name of Design Variables	State	Car B Bounds (mm,degree)	
			Lower	Upper
1	Distance_from_edge	## Fixed by program	8.00	8.00
2	Header_offset	## Dependent	255.00	255.00
3	Roof_rail_offset	## Dependent	196.00	196.00
4	A_pillar_offset	## Dependent	179.00	179.00
5	Theta	# Fixed by user	25.00	25.00
6	Phi	# Fixed by user	76.00	76.00
7	Header_horizontal_offset	# Fixed by user	37.00	37.00
8	Flange1_width	## Fixed by program	20.00	20.00
9	Flange2_width	## Fixed by program	16.00	16.00
10	Flange3_width	## Fixed by program	16.00	16.00
11	Spot_weld_spacing	## Fixed by program	50.00	50.00
12	Flange1_angle_up	# Fixed by user	103.00	103.00
13	Flange1_angle_down	Independent	90.00	121.00
14	Flange2_angle_up	# Fixed by user	130.00	130.00
15	Flange2_angle_down	Independent	90.00	135.00
16	Flange3_angle_down	Independent	90.00	118.00
17	Door_allowance	# Fixed by user	62.00	62.00
18	Angleback	# Fixed by user	23.00	23.00
19	H_width	# Fixed by user	90.00	90.00
20	H_height	Independent	17.00	37.00
21	H_window_depth	# Fixed by user	15.00	15.00
22	H_blending_rad	# Fixed by user	140.00	140.00
23	RR_width	# Fixed by user	73.00	73.00
24	RR_height	Independent	43.00	87.00
25	RR_h1	# Fixed by user	52.00	52.00
26	RR_h2	Independent	32.00	118.00
27	RR_angle_bottom	Independent	18.00	46.00
28	RR_blending_rad	# Fixed by user	88.00	88.00
29	AP_height	Independent	47.00	86.00
30	AP_inboard_width	# Fixed by user	31.00	31.00
31	AP_inboard_depth	Independent	23.00	40.00
32	AP_door_ws_distance	# Fixed by user	19.00	19.00
33	AP_window_depth	# Fixed by user	26.00	26.00
34	AP_flange1_angle	# Fixed by user	10.00	10.00
35	AP_reinf_depth	Independent	1.00	29.00
36	AP_reinf_flange1_angle	Independent	91.00	171.00
37	AP_reinf_inner_door_allowance	Independent	20.00	69.00
38	AP_reinf_flange2_angle	Independent	91.00	144.00
39	Ap_blending_rad	# Fixed by user	100.00	100.00

40	Part2_cut_distance	Independent	50.00	121.00
41	Part3_cut_distance	Independent	22.00	62.00
42	Part4_extension	Independent	42.00	99.00
43	Part7_cut_distance	Independent	40.00	121.00
44	Thickness of Part1	# Fixed by user	0.89	0.89
45	Thickness of Part2	# Fixed by user	0.89	0.89
46	Thickness of Part3	Independent	1.12	1.12
47	Thickness of Part4	Independent	0.81	1.18
48	Thickness of Part7	Independent	1.06	2.01

The lengths of header, roof rail and A-pillar branches are fixed at 255 mm, 233 mm, and 145.5 mm, respectively.

Table 7.4d: Optimum Design for Car B

No	Name of Design Variables	Optimum Design for Car 2 (mm,degree)	
		Use RSP	Use NN
1	Distance_from_edge	8.00	8.00
2	Header_offset	255.00	255.00
3	Roof_rail_offset	196.00	196.00
4	A_pillar_offset	179.03	179.03
5	Theta	25.00	25.00
6	Phi	76.00	76.00
7	Header_horizontal_offset	37.00	37.00
8	Flange1_width	20.00	20.00
9	Flange2_width	16.00	16.00
10	Flange3_width	16.00	16.00
11	Spot_weld_spacing	50.00	50.00
12	Flange1_angle_up	103.00	103.00
13	Flange1_angle_down	98.65	90.89
14	Flange2_angle_up	130.00	130.00
15	Flange2_angle_down	135.00	134.98
16	Flange3_angle_down	118.00	105.94
17	Door_allowance	62.00	62.00
18	Angleback	23.00	23.00
19	H_width	90.00	90.00
20	H_height	19.94	30.00
21	H_window_depth	15.00	15.00
22	H_blending_rad	140.00	140.00
23	RR_width	73.00	73.00
24	RR_height	67.14	73.01
25	RR_h1	52.00	52.00
26	RR_h2	73.00	73.00
27	RR_angle_bottom	18.00	25.08

28	RR_blending_rad	88.00	88.00
29	AP_height	70.57	70.57
30	AP_inboard_width	31.00	31.00
31	AP_inboard_depth	39.90	38.12
32	AP_door_ws_distance	19.00	19.00
33	AP_window_depth	26.00	26.00
34	AP_flange1_angle	10.00	10.00
35	AP_reinf_depth	1.00	1.00
36	AP_reinf_flange1_angle	159.25	158.46
37	AP_reinf_inner_door_allowance	37.26	35.30
38	AP_reinf_flange2_angle	144.00	131.00
39	Ap_blending_rad	100.00	100.00
40	Part2_cut_distance	121.00	83.28
41	Part3_cut_distance	22.00	22.00
42	Part4_extension	42.00	42.00
43	Part7_cut_distance	121.00	121.00
44	Thickness of Part1	0.89	0.89
45	Thickness of Part2	0.89	0.89
46	Thickness of Part3	1.12	1.12
47	Thickness of Part4	0.91	0.91
48	Thickness of Part7	1.06	1.06

Table 7.4e: Comparison of Optimization and FEA Results for A-pillar to Roof Rail Joint using RSP and NN Translators

Cars/ Stiff Req.	Stiffness / Mass	Use RSP Translators (Nmm,kg)		Use NN Translators (Nmm,kg)	
		FEA	Predictions (err)	FEA	Predictions (err)
Car A I/O>6.531E7 F/A>4.929E7 Tor>1.259E7	I/O	6.6568E7	6.5329E7 (-1.9%)	6.6554E7	6.5301E7 (-1.9%)
	F/A	6.5412E7	6.3504E7 (-2.9%)	7.7032E7	6.8737E7 (-10.8%)
	Torsion	1.1171E7	1.2826E7 (14.8%)	1.3047E7	1.2587E7 (-3.5%)
	Mass	1.8835	1.8478 (-1.9%)	2.0479	2.0279 (-1.0%)
Car B I/O>3.399E7 F/A>5.981E7 Tor>1.044E7	I/O	4.4762E7	5.2274E7 (16.8%)	5.4829E7	5.2274E7 (-4.7%)
	F/A	5.8044E7	5.9812E7 (3.0%)	6.3793E7	5.9812E7 (-6.2%)
	Torsion	0.8766E7	1.0524E7 (20.1%)	1.1822E7	1.0524E7 (-11.0%)
	Mass	2.1667	2.1528 (-0.6%)	2.2477	2.1528 (-4.2%)

Table 7.4f: Comparison of Mass of the Initial and Optimum Designs for Car A and Car B

Cars	Initial Design (kg)	Optimum Design (kg) and Percentage Improvement	
		RSP	NN
Car A	2.0302	1.8835 (7.2%)	2.0479 (-0.9%)
Car B	2.4396	2.1667 (11.2%)	2.2477 (7.8%)

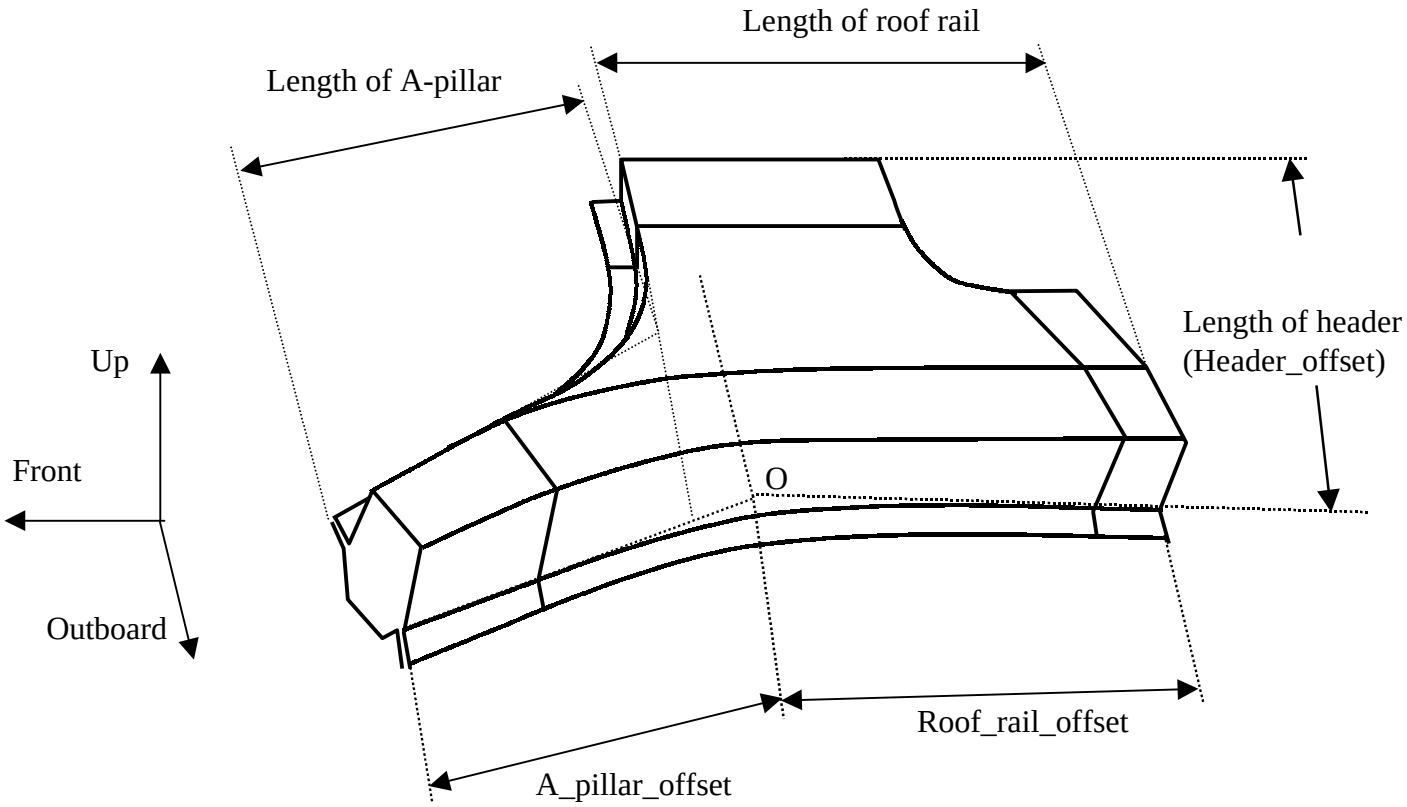


Figure 7.1: Definition of Lengths of Three Branches for A-pillar to Roof Rail Joint

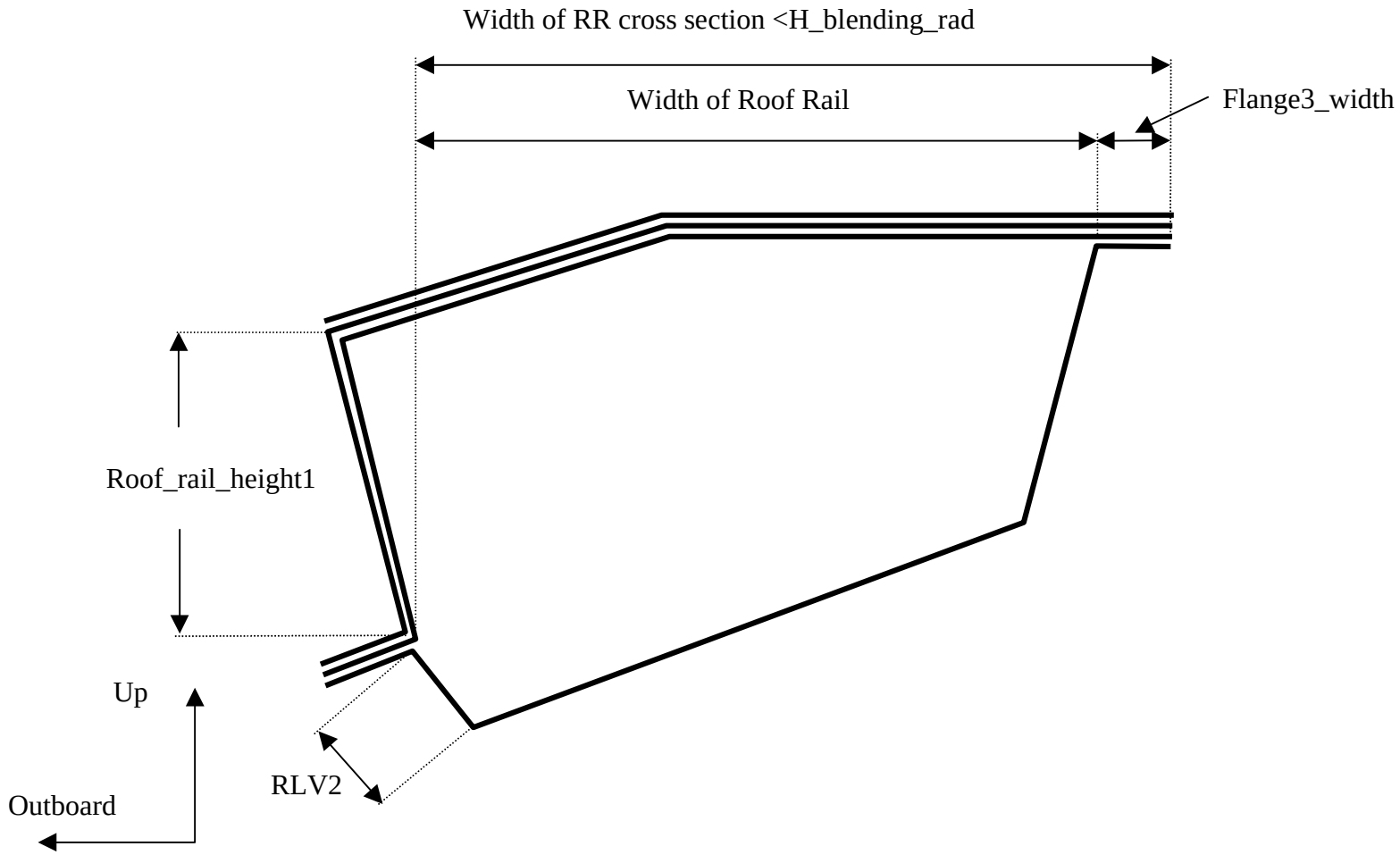


Figure 7.2: Constraints on Dimensions of Roof Rail Cross Section

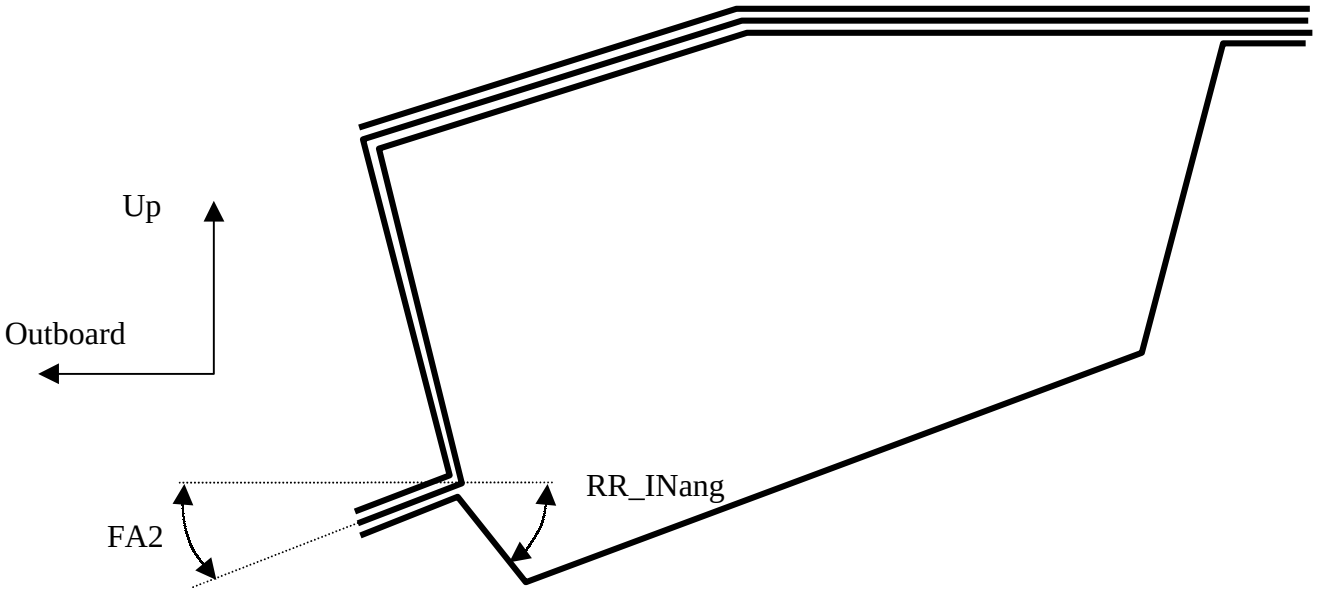


Figure 7.3: Constraints on Angles of Roof Rail Cross Section

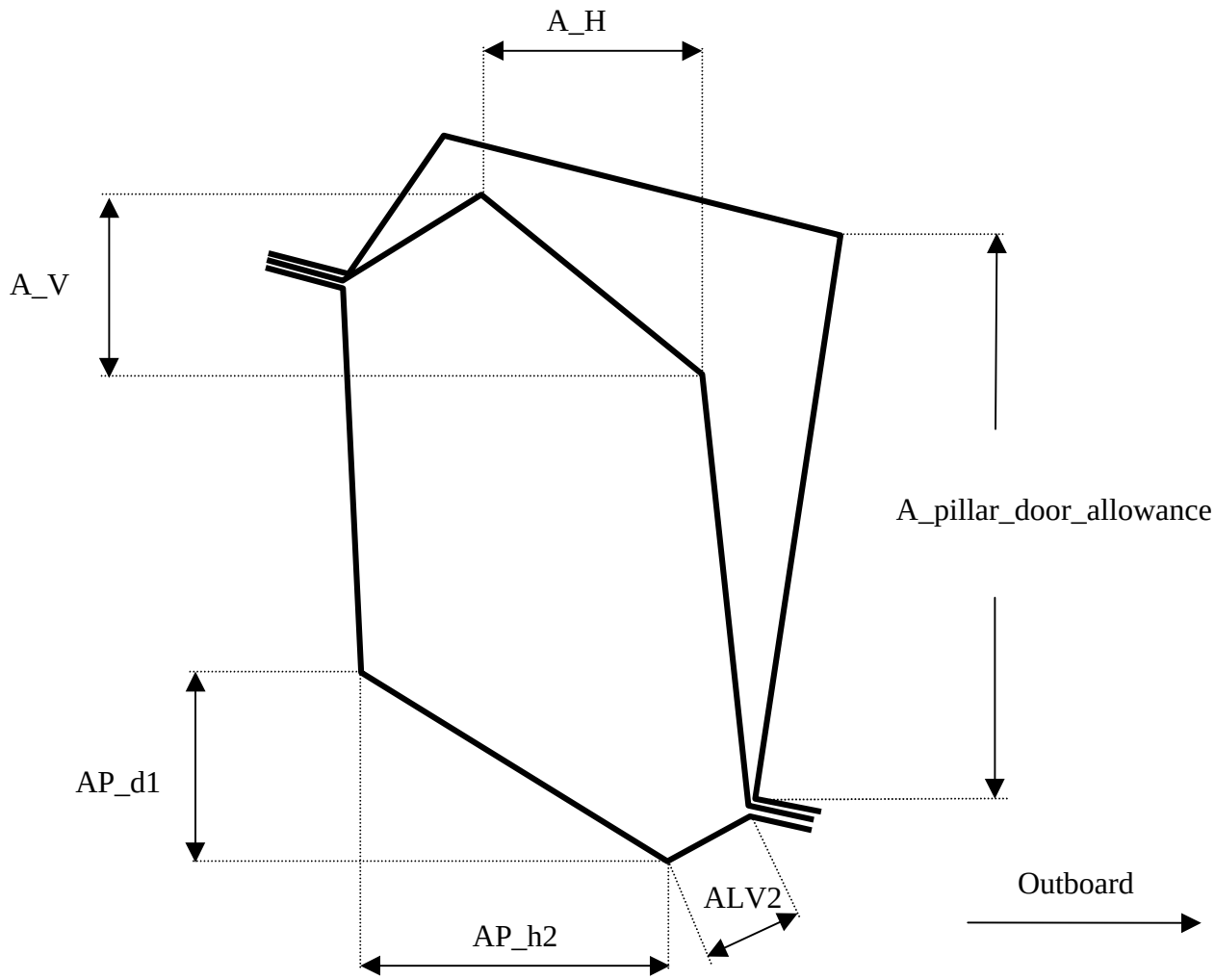


Figure 7.4: Constraints on Dimensions of A-pillar Cross Section

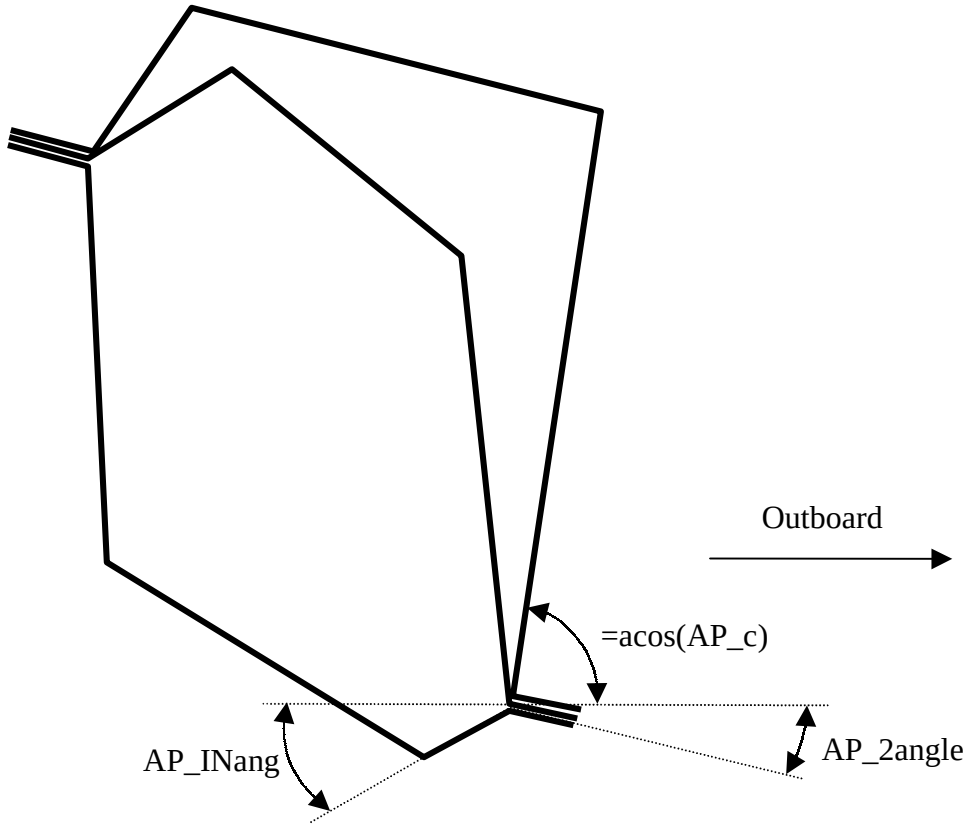


Figure 7.5: Constraints on Angles of A-pillar Cross Section

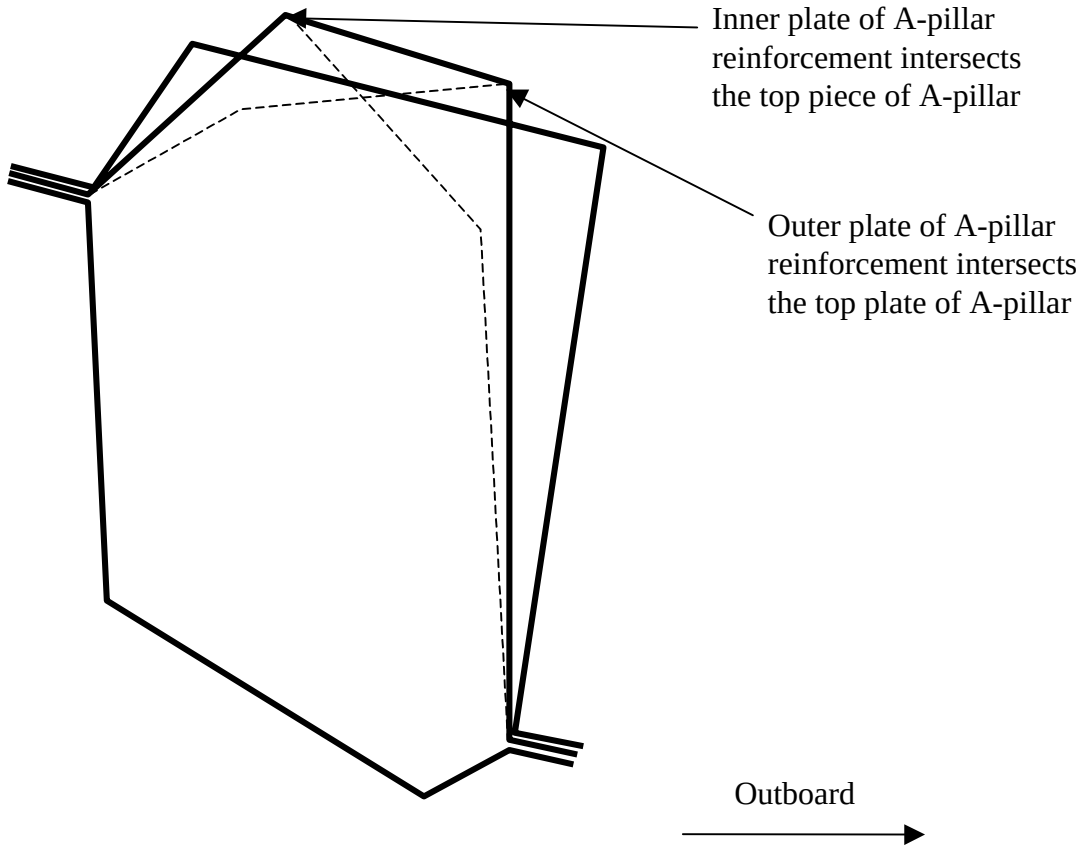


Figure 7.6: Constraints on A-pillar Reinforcement

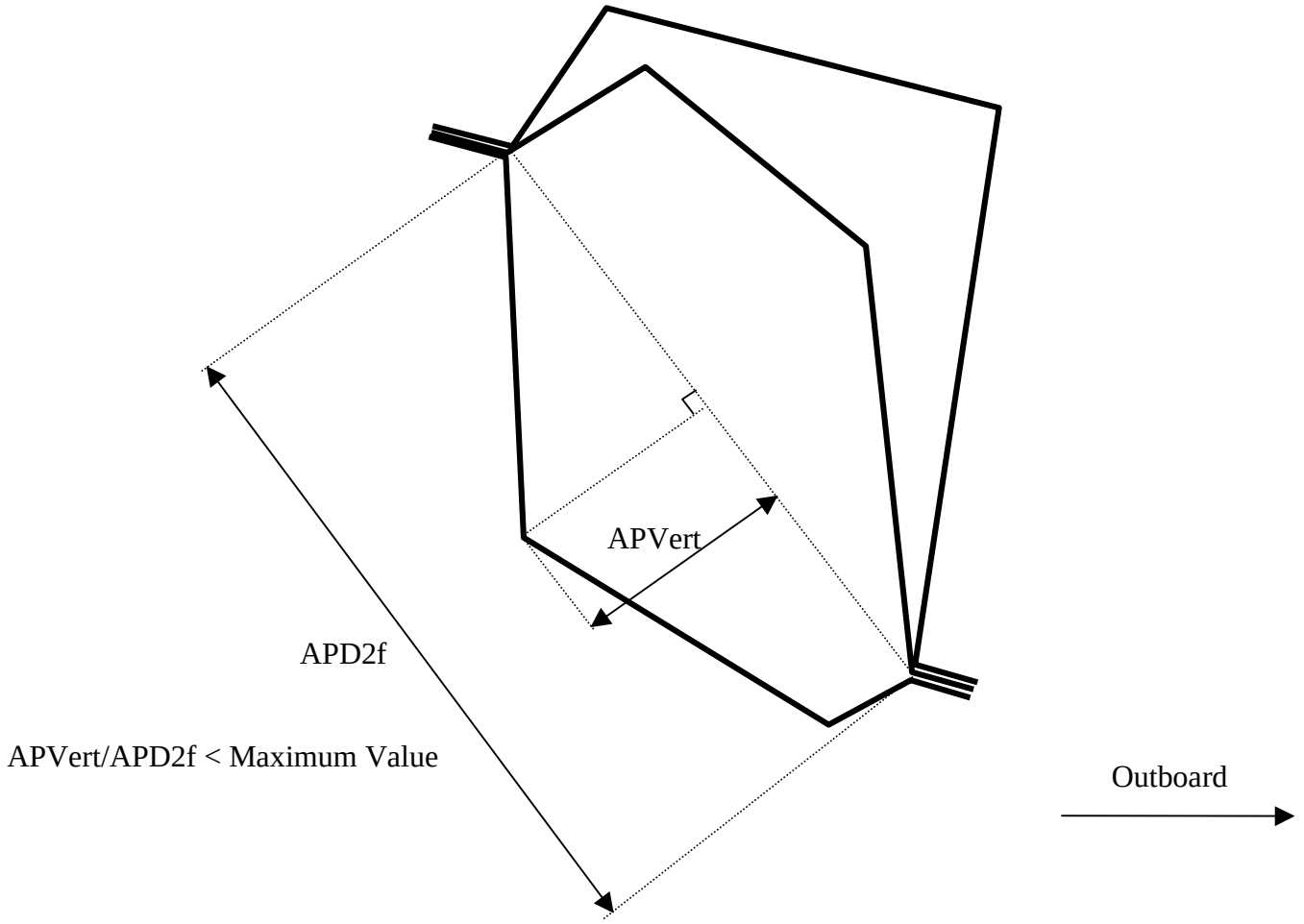


Figure 7.7: Constraints on the Inner Plate of A-pillar

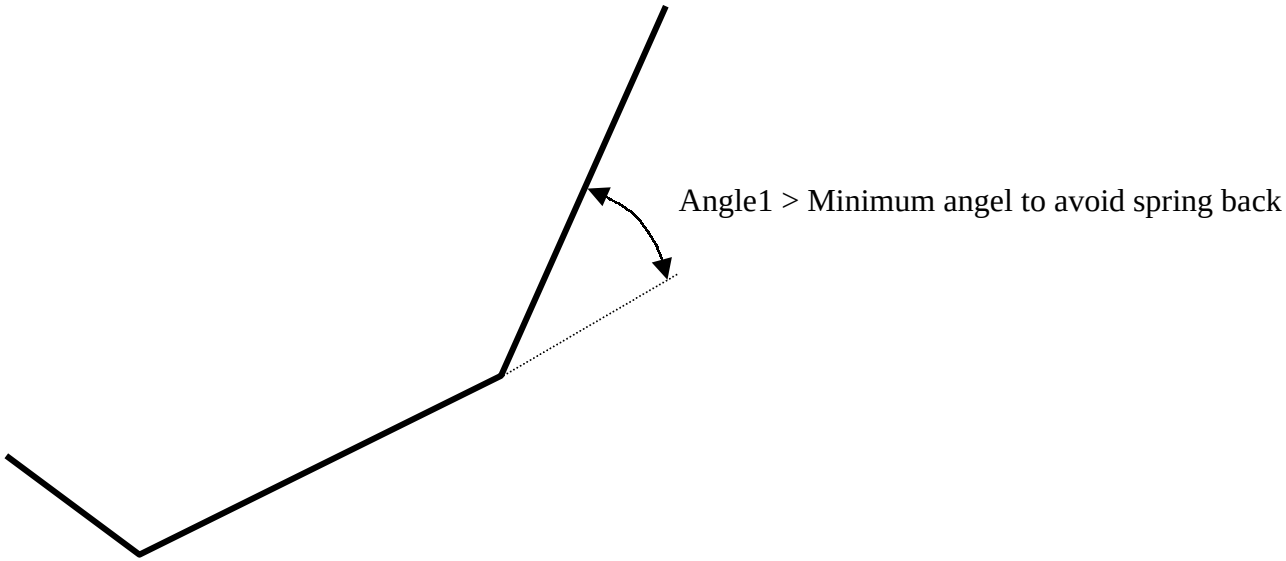


Figure 7.8: Explanations of Spring Back Angle

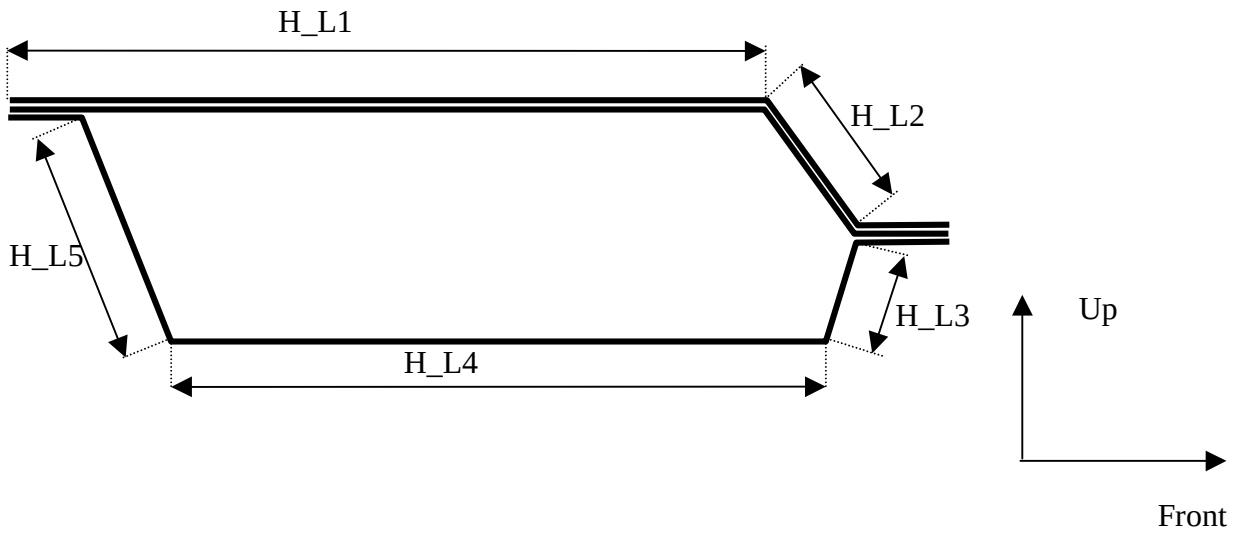


Figure 7.9: Manufacturing Constraints on Lengths of Cross Section of Header

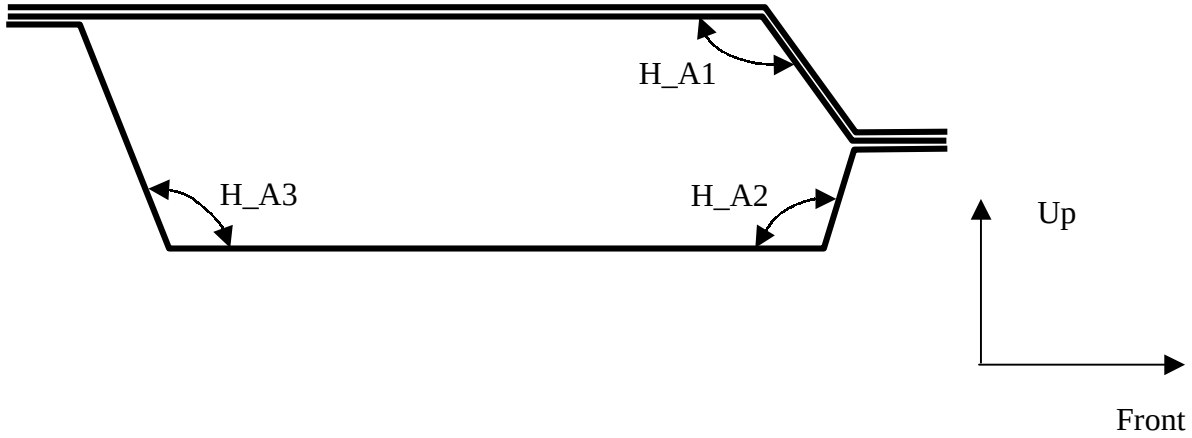


Figure 7.10: Manufacturing Constraints on Angles of Cross Section of Header

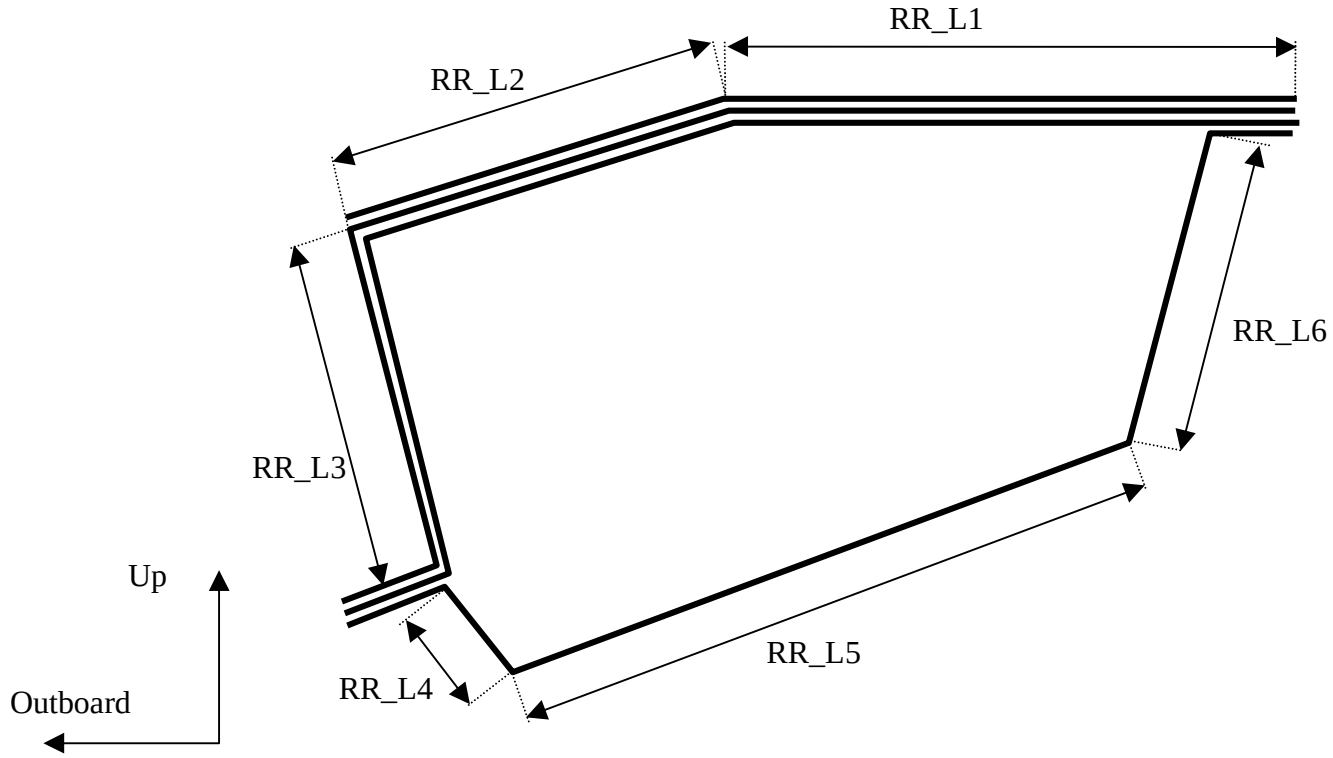


Figure 7.11: Manufacturing Constraints on Lengths of Cross Section of Roof Rail

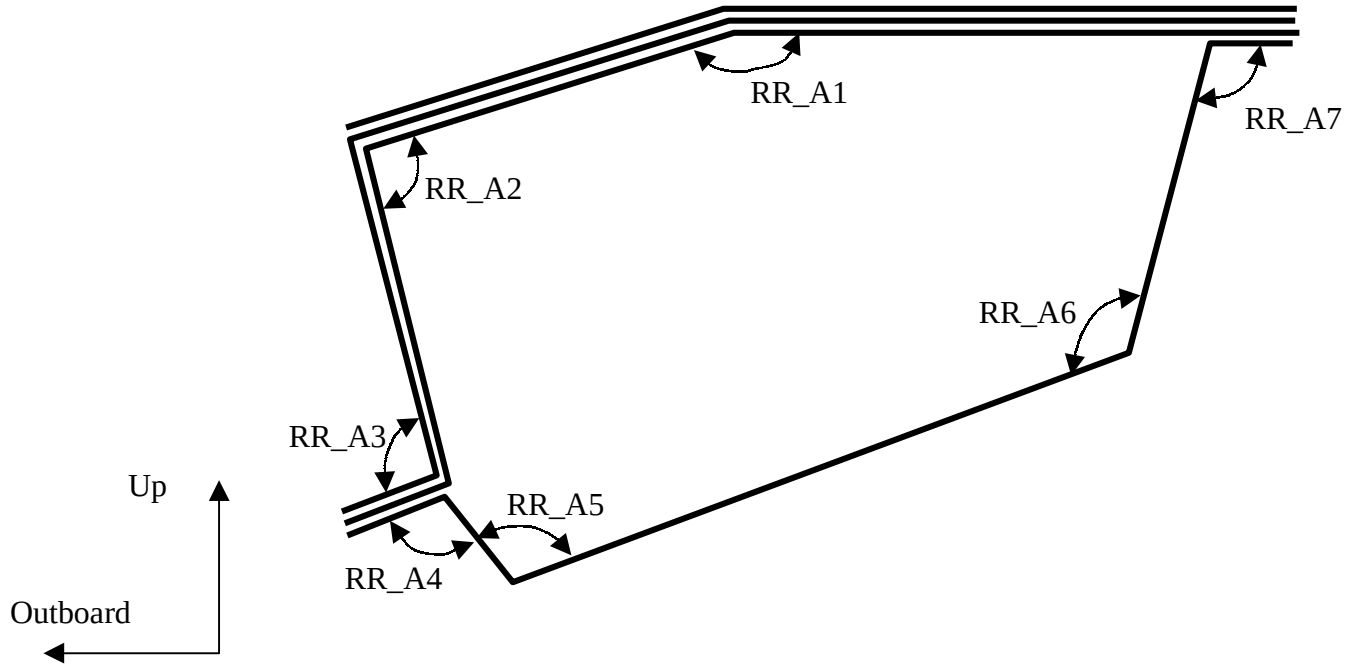


Figure 7.12: Manufacturing Constraints on Angles of Cross Section of Roof Rail

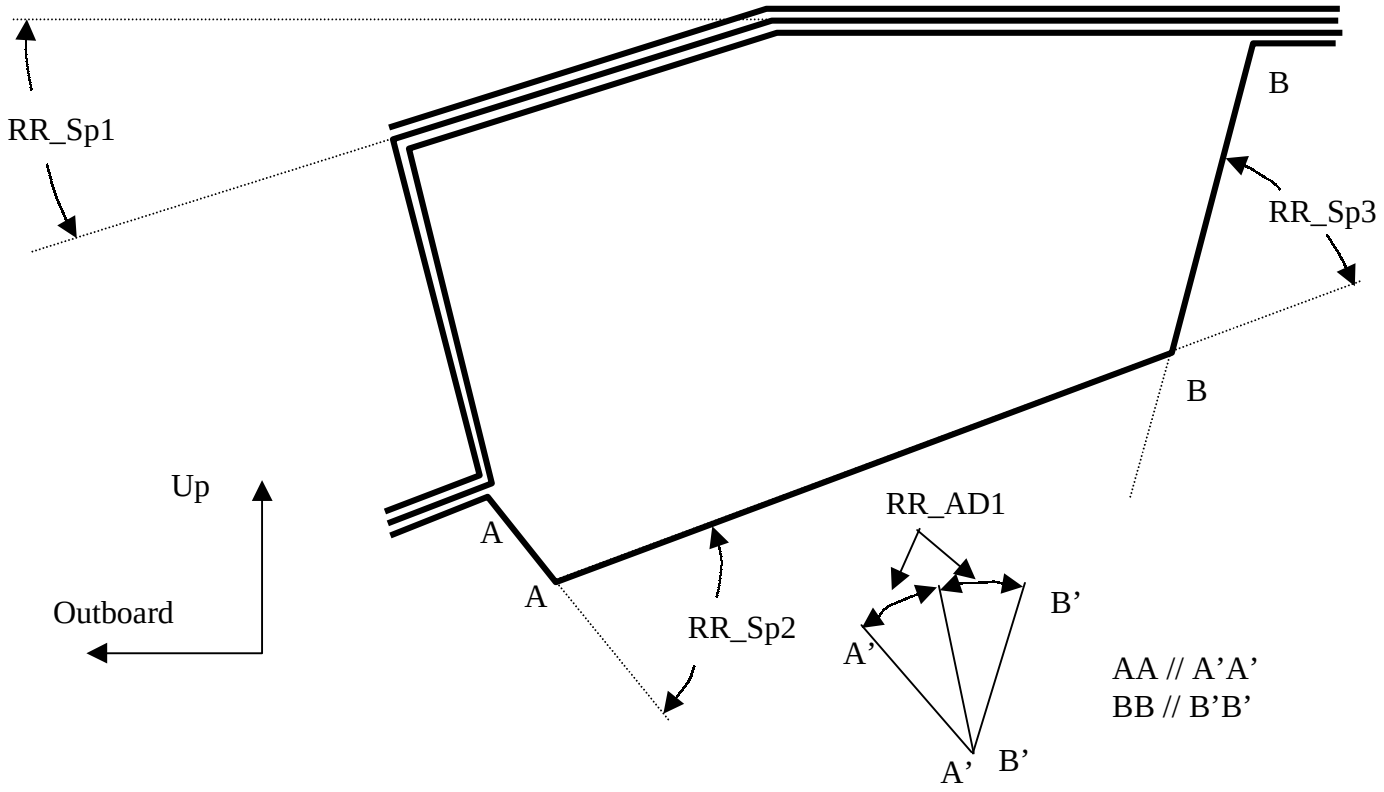


Figure 7.13: Constraints on the Spring Back Angles and Die Lock Angle of Roof Rail Cross Section

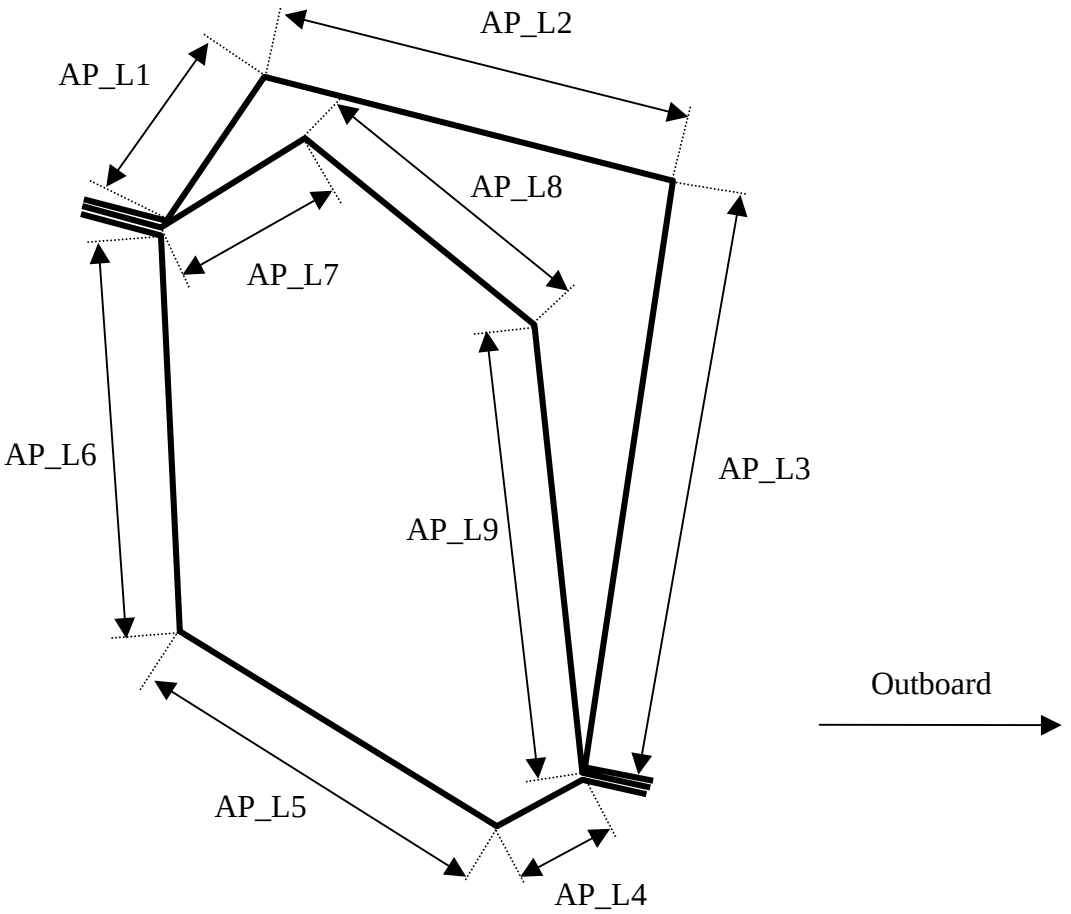


Figure 7.14: Manufacturing Constraints on Lengths of A-pillar Section

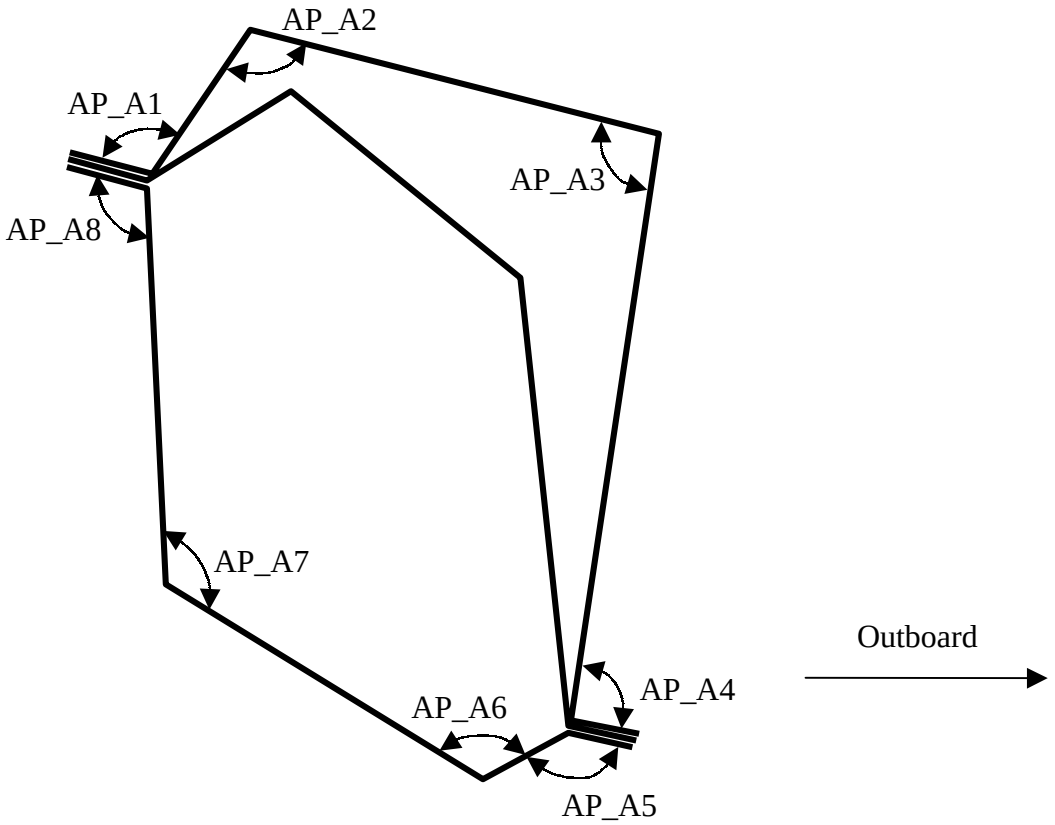


Figure 7.15: Manufacturing Constraints on Angles of the Outer Shell of A-pillar Cross Section

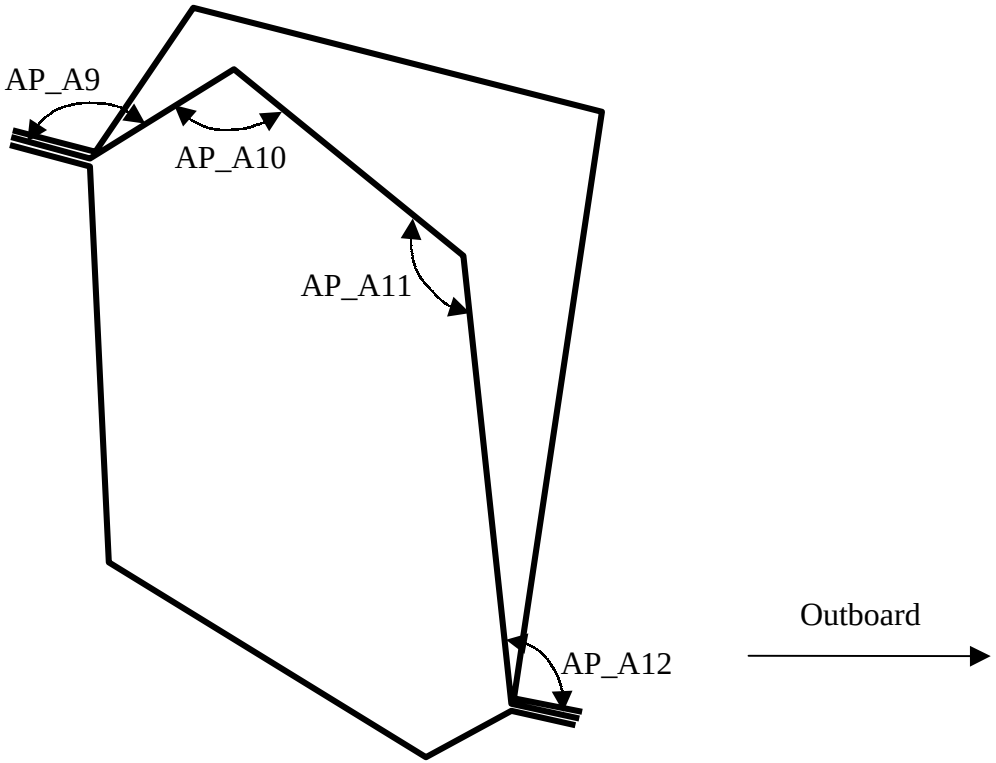


Figure 7.16: Manufacturing Constraints on Angles of A-pillar Reinforcement

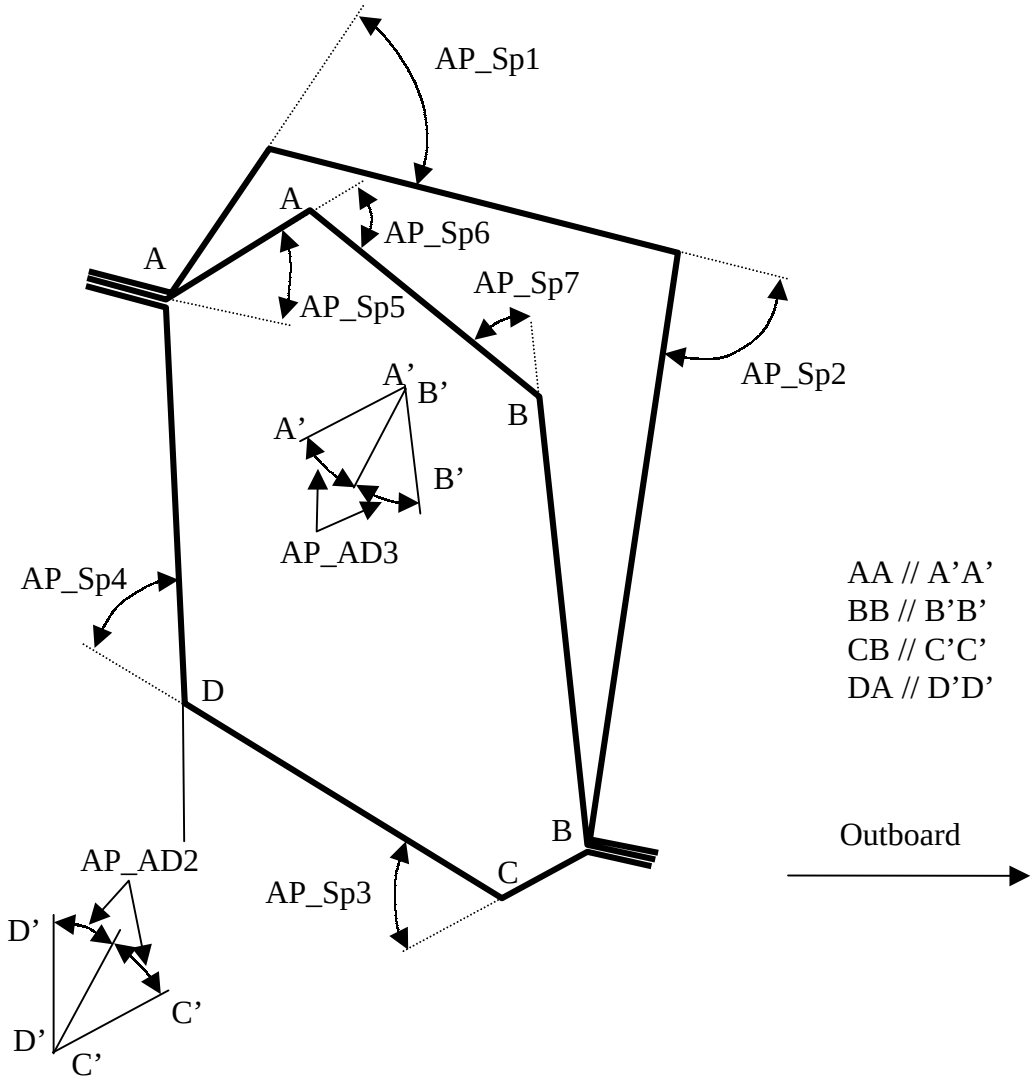


Figure 7.17: Constraints on Spring Back Angles and Die Lock Angles of A-pillar Cross Section

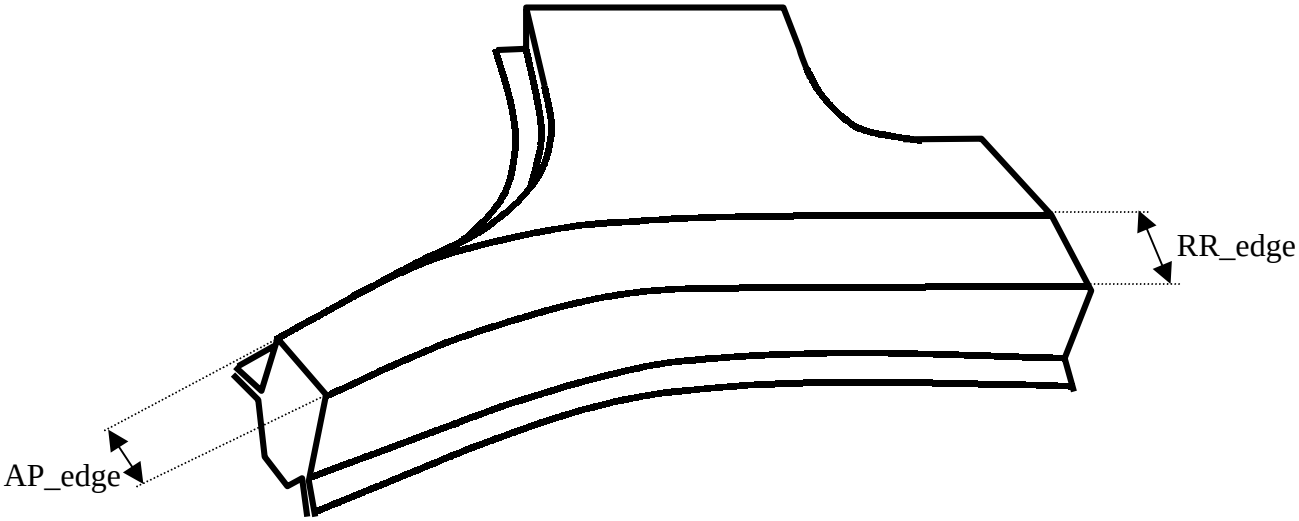


Figure 7.18: Styling Constraint

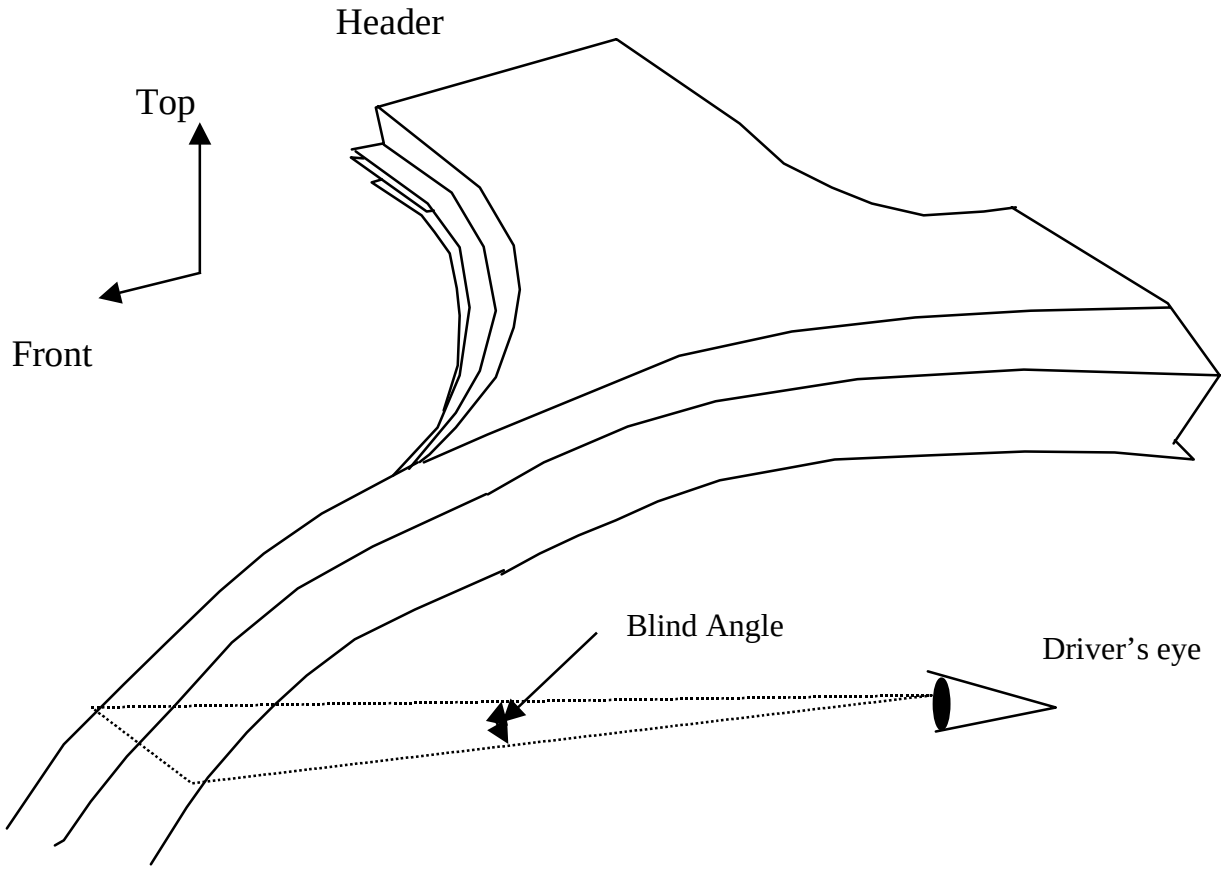


Figure 7.19: Explanation of Safety Constraint

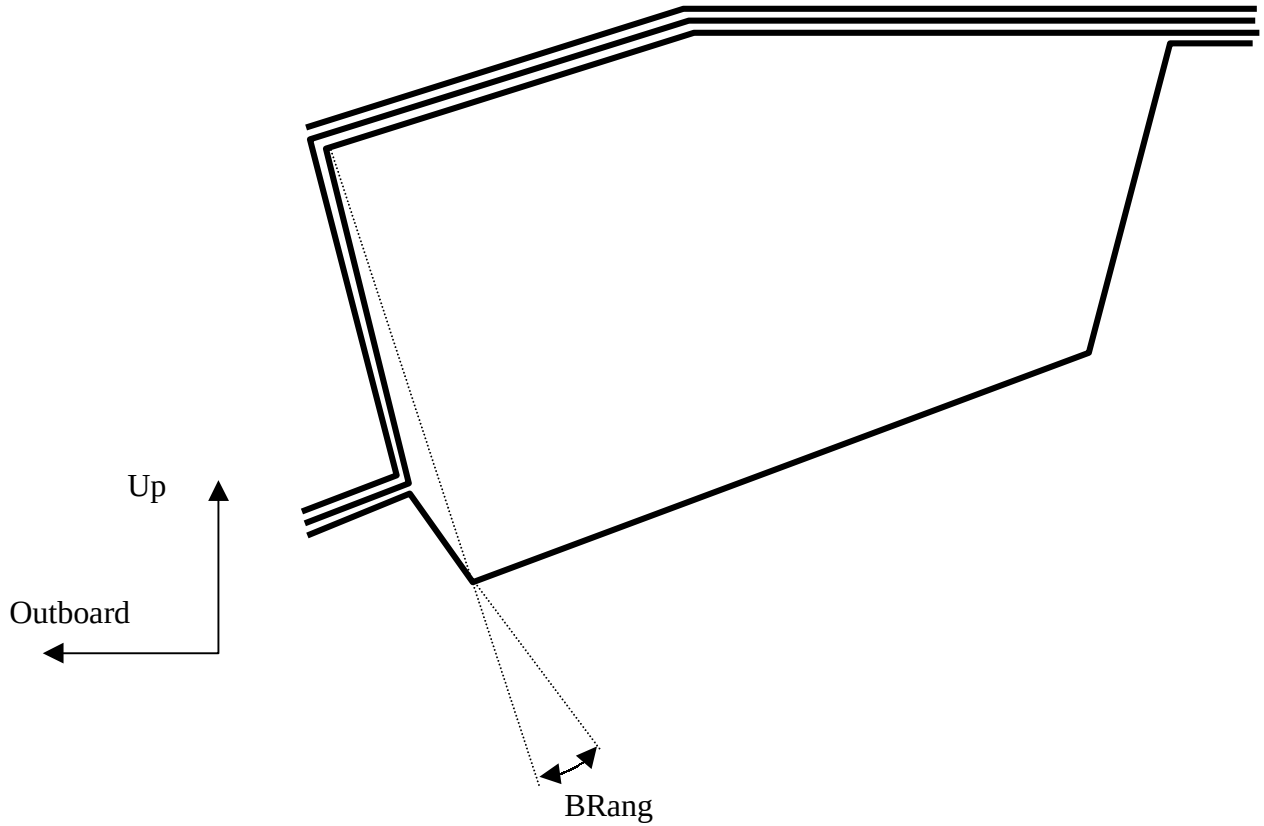


Figure 7.20: Constraints on an Angle of Roof Rail Cross Section

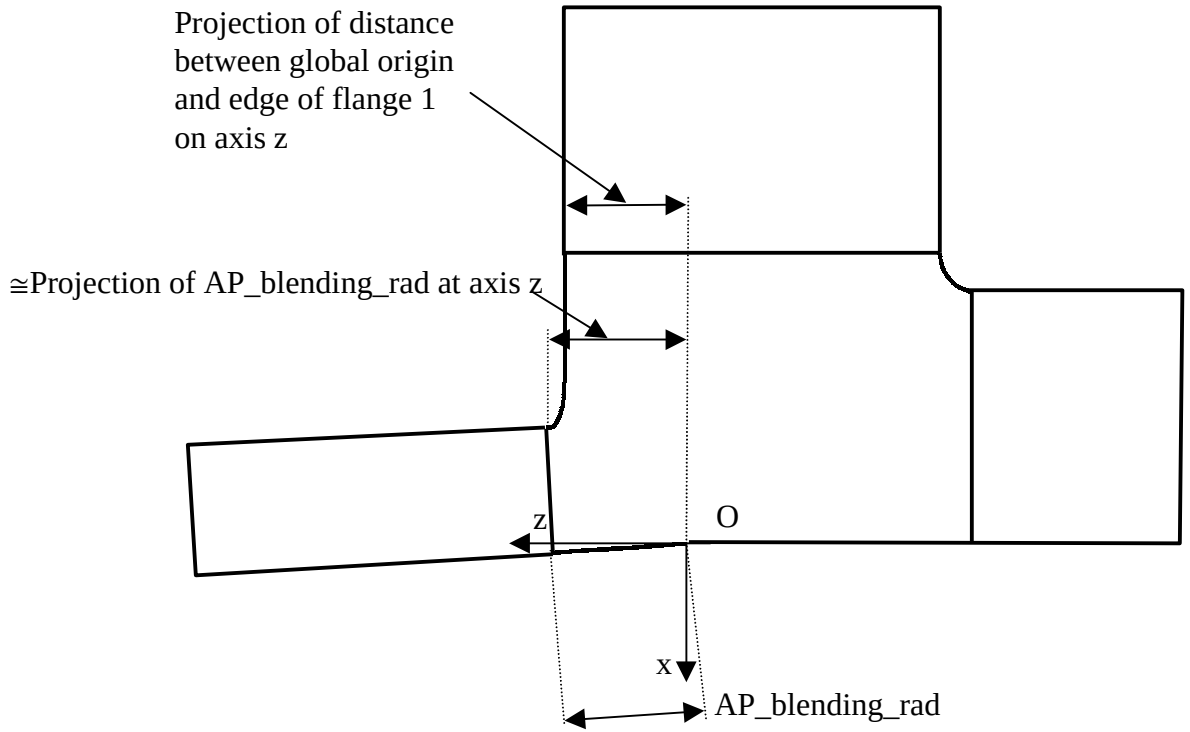


Figure 7.21: Constraint on A-pillar Blending Radius

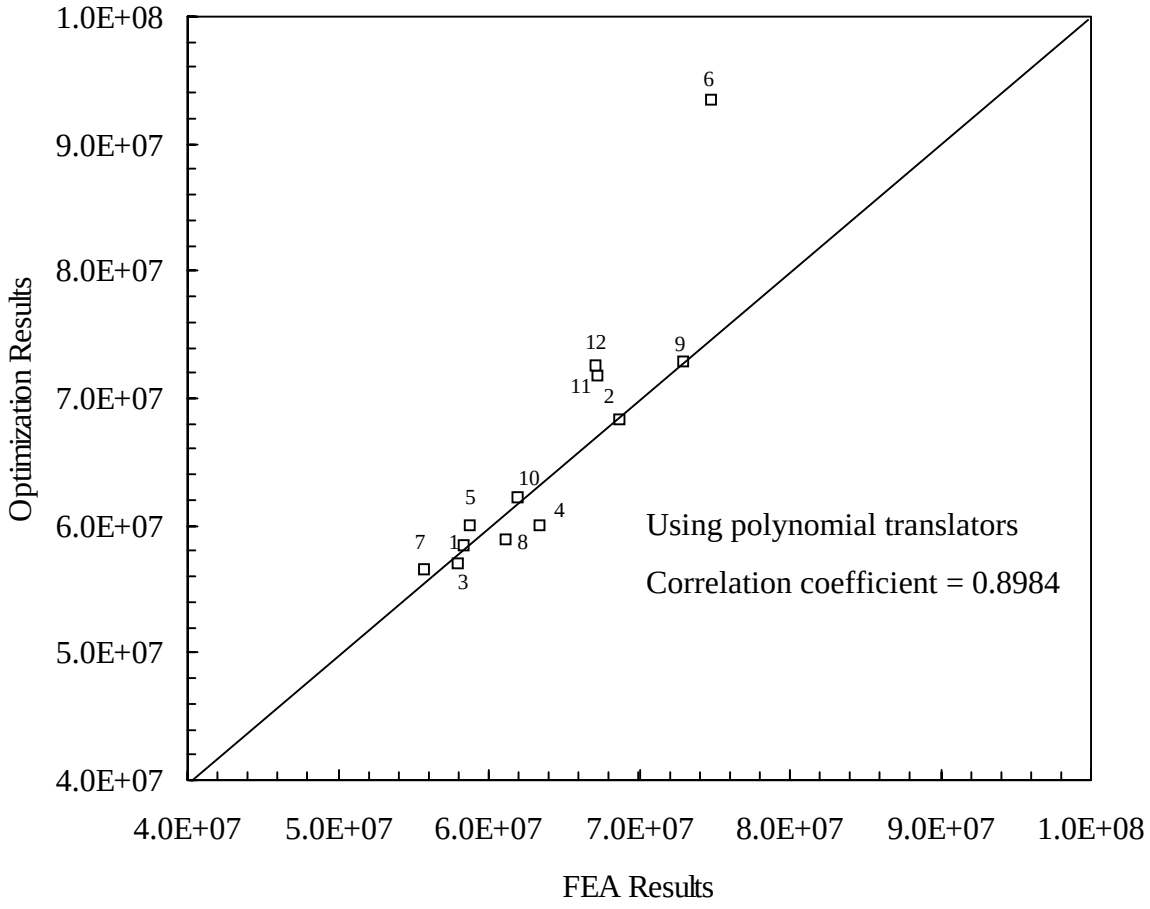


Figure 7.22a: Comparison of FEA Results and Optimization Results Obtained using RSP Translator for the I/O Stiffness of A-pillar to Roof Rail Joint

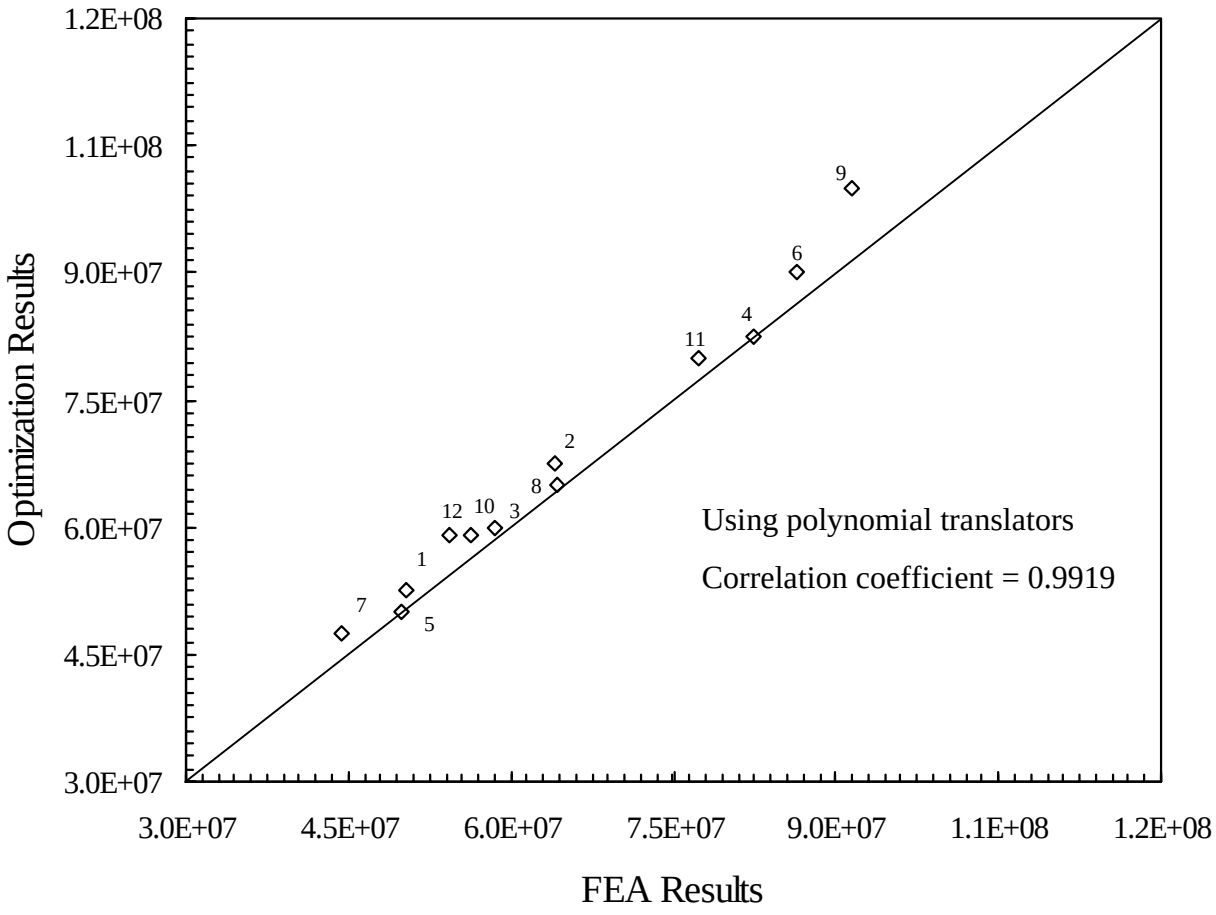


Figure 7.22b: Comparison of FEA Results and Optimization Results Obtained using RSP Translator for the F/A Stiffness of A-pillar to Roof Rail Joint

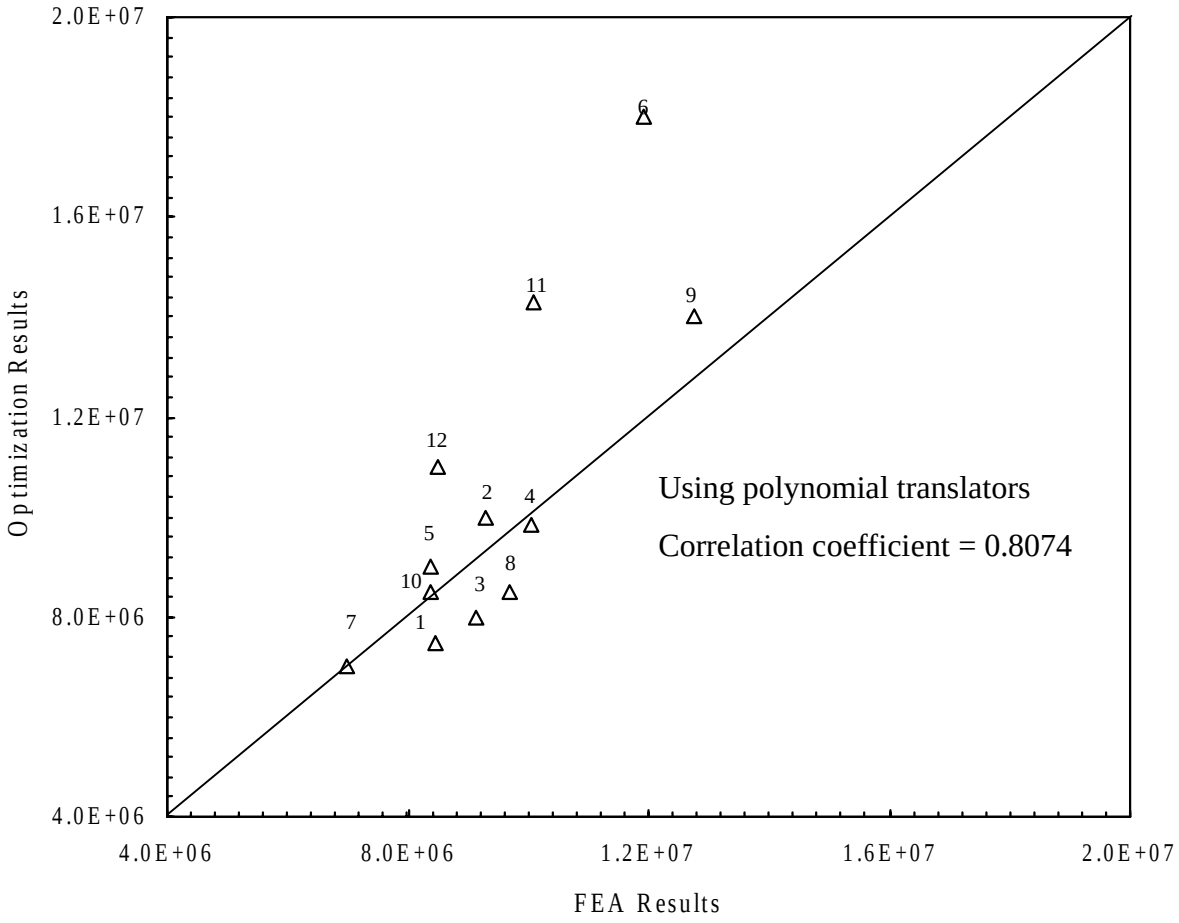


Figure 7.22c: Comparison of FEA Results and Optimization Results Obtained using RSP Translator for the Torsion Stiffness of A-pillar to Roof Rail Joint

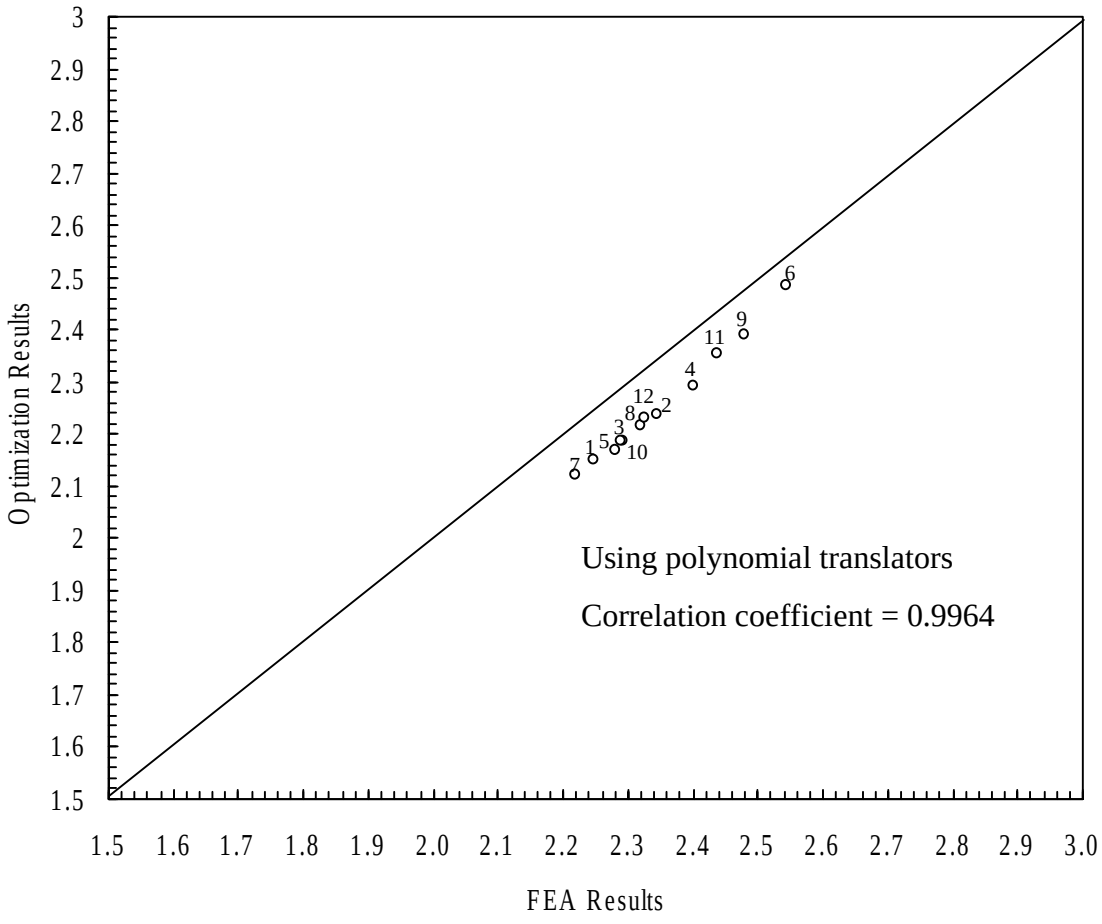


Figure 7.22d: Comparison of FEA Results and Optimization Results Obtained using RSP Translator for the Mass of A-pillar to Roof Rail Joint

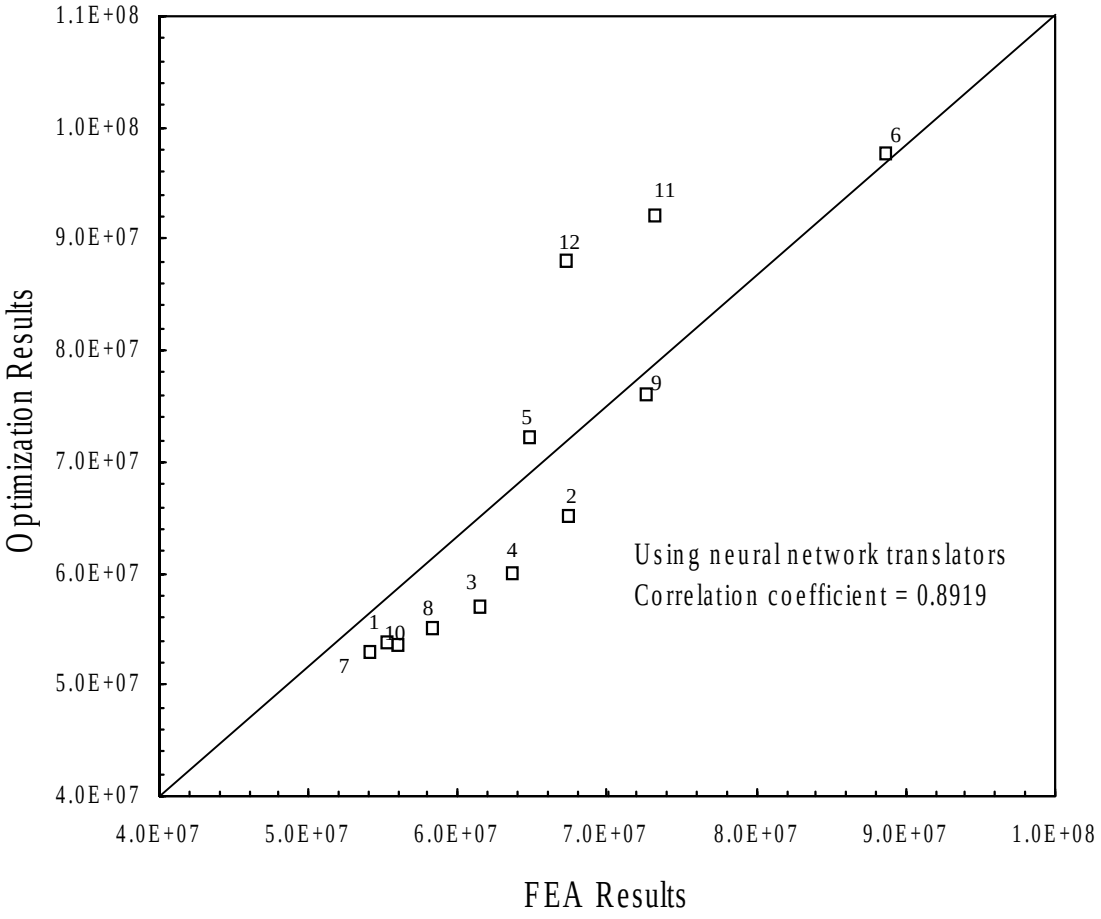


Figure 7.23a: Comparison of FEA Results and Optimization Results Obtained using NN Translator for the I/O Stiffness of A-pillar to Roof Rail Joint

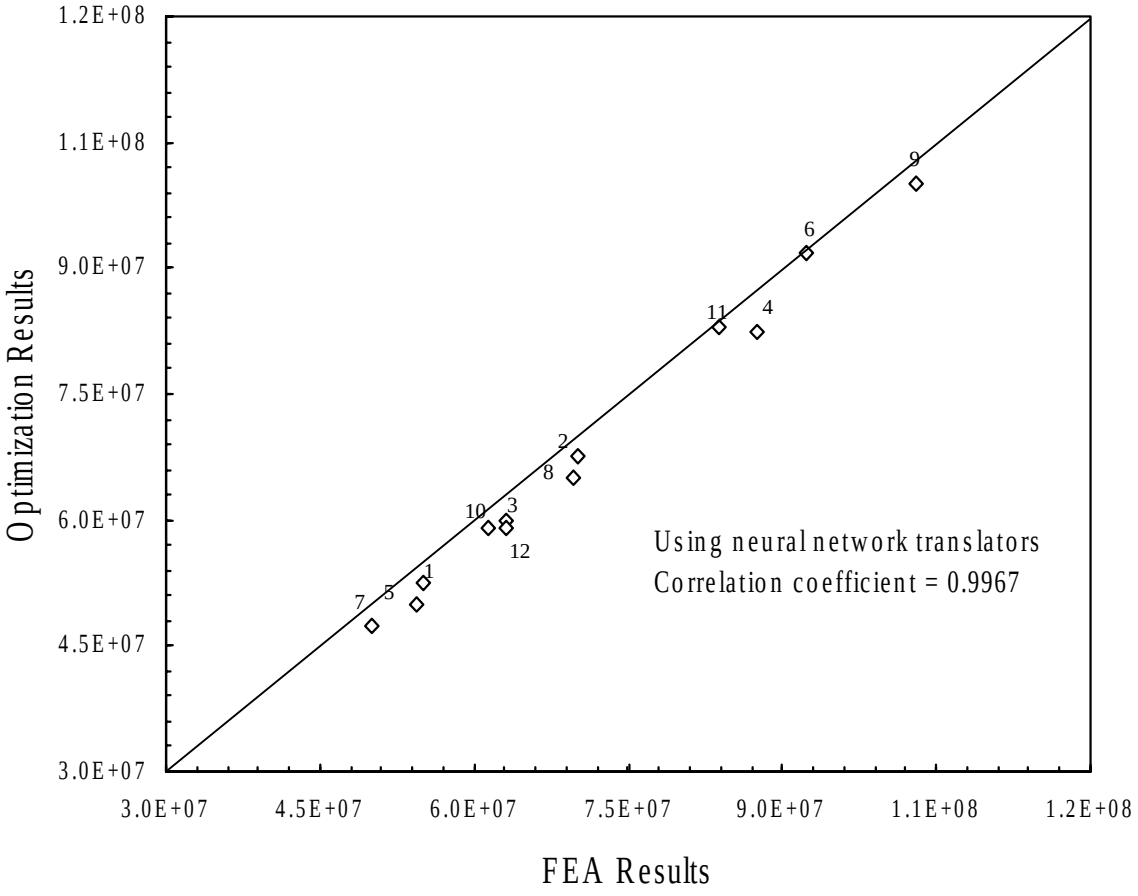


Figure 7.23b: Comparison of FEA Results and Optimization Results Obtained using NN Translator for the F/A Stiffness of A-pillar to Roof Rail Joint

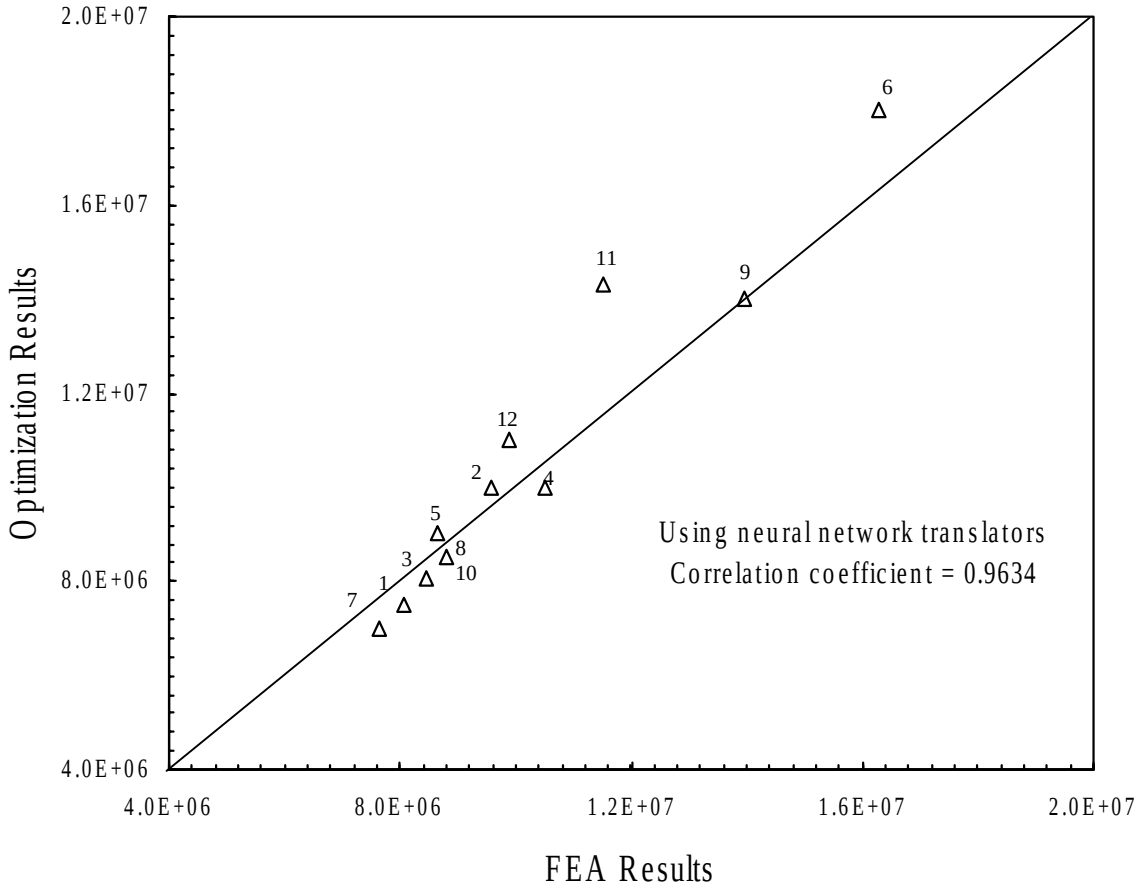


Figure 7.23c: Comparison of FEA Results and Optimization Results Obtained using NN Translator for the Torsion Stiffness of A-pillar to Roof Rail Joint

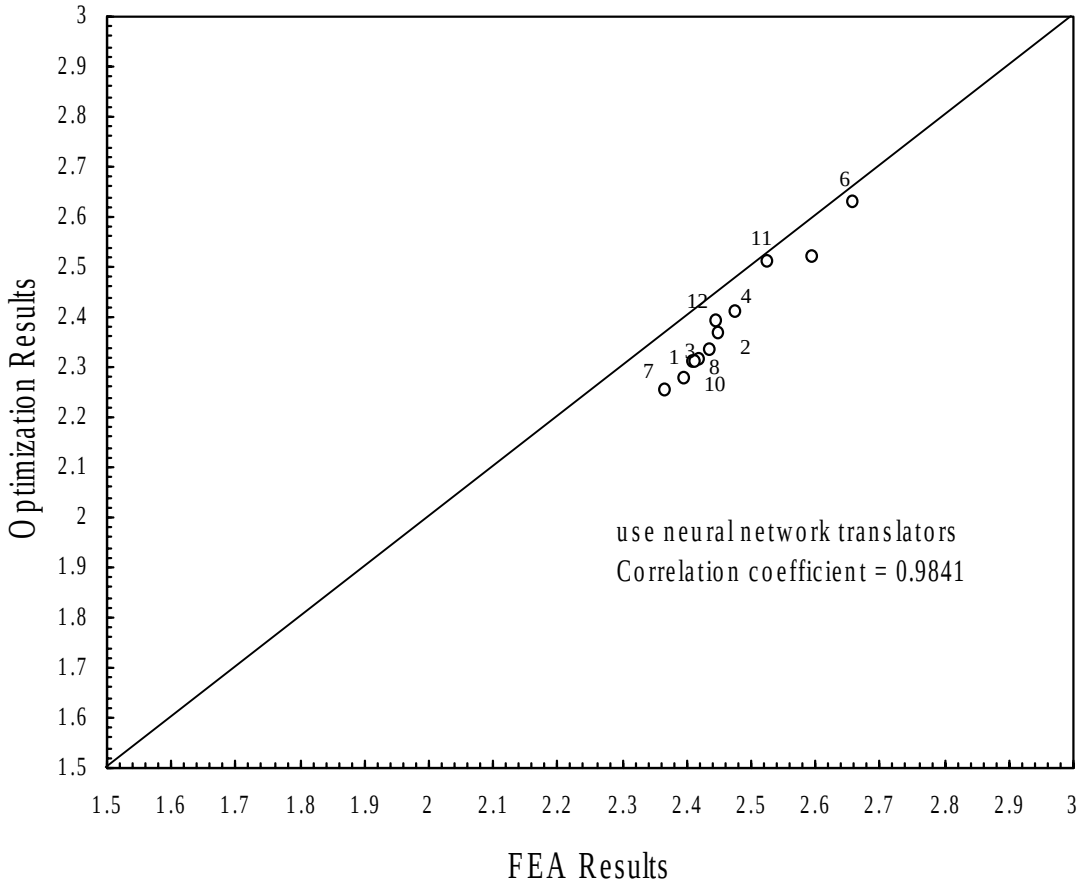


Figure 7.23d: Comparison of FEA Results and Optimization Results Obtained using NN Translator for the Mass of A-pillar to Roof Rail Joint

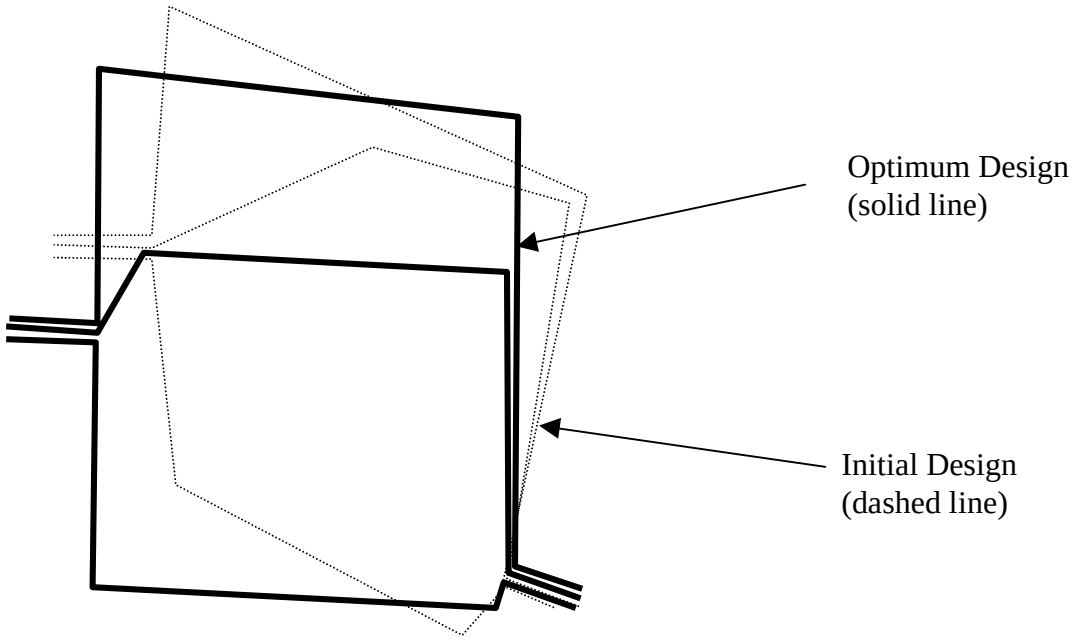


Figure 7.24: Comparison of A-pillar Cross Sections of Actual Joint and Optimum Joint from Translator B

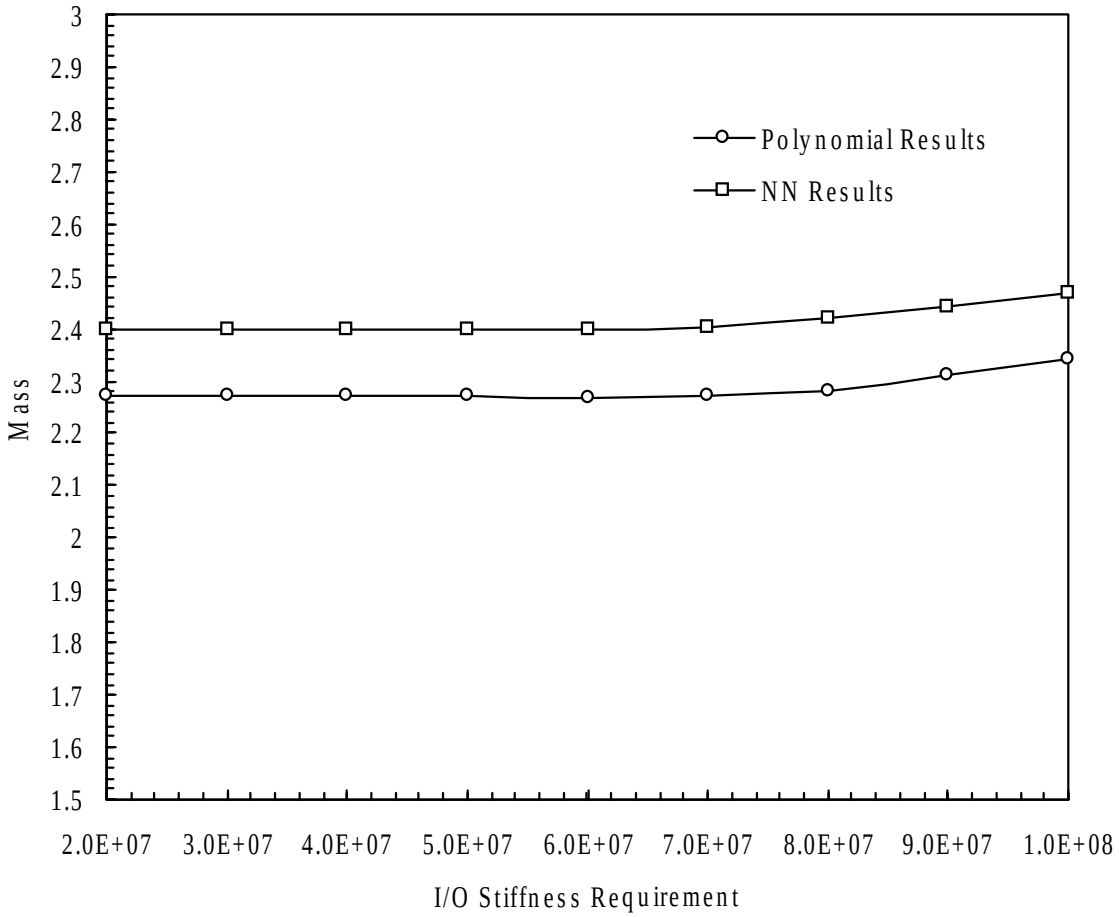


Figure 7.25: Mass of Optimum Design vs. I/O Stiffness Requirement
($K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

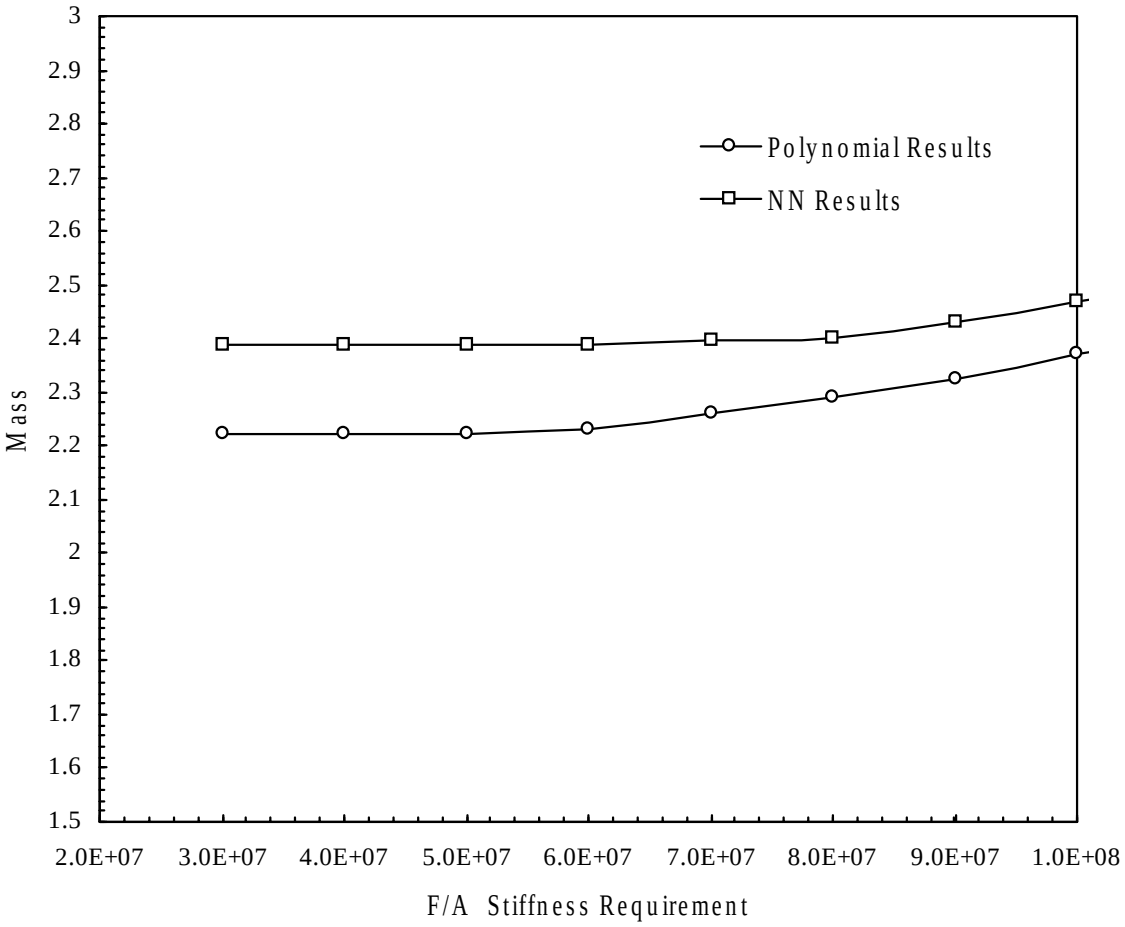


Figure 7.26: Mass of Optimum Design vs. F/A Stiffness Requirement
($K_{I/O} \geq 5.3936E7$, $K_{Tor} \geq 1.0945E7$)

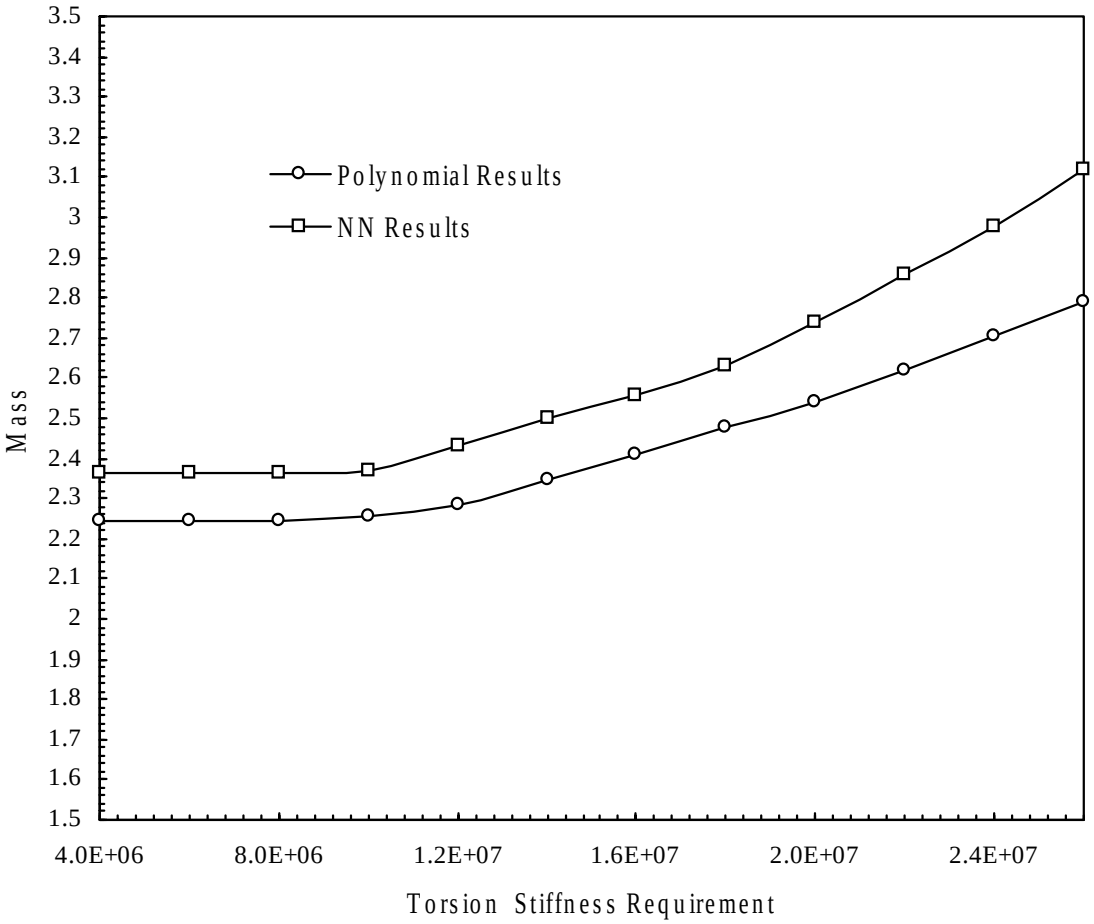


Figure 7.27: Mass of Optimum Design vs. Torsion Stiffness Requirement ($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$)

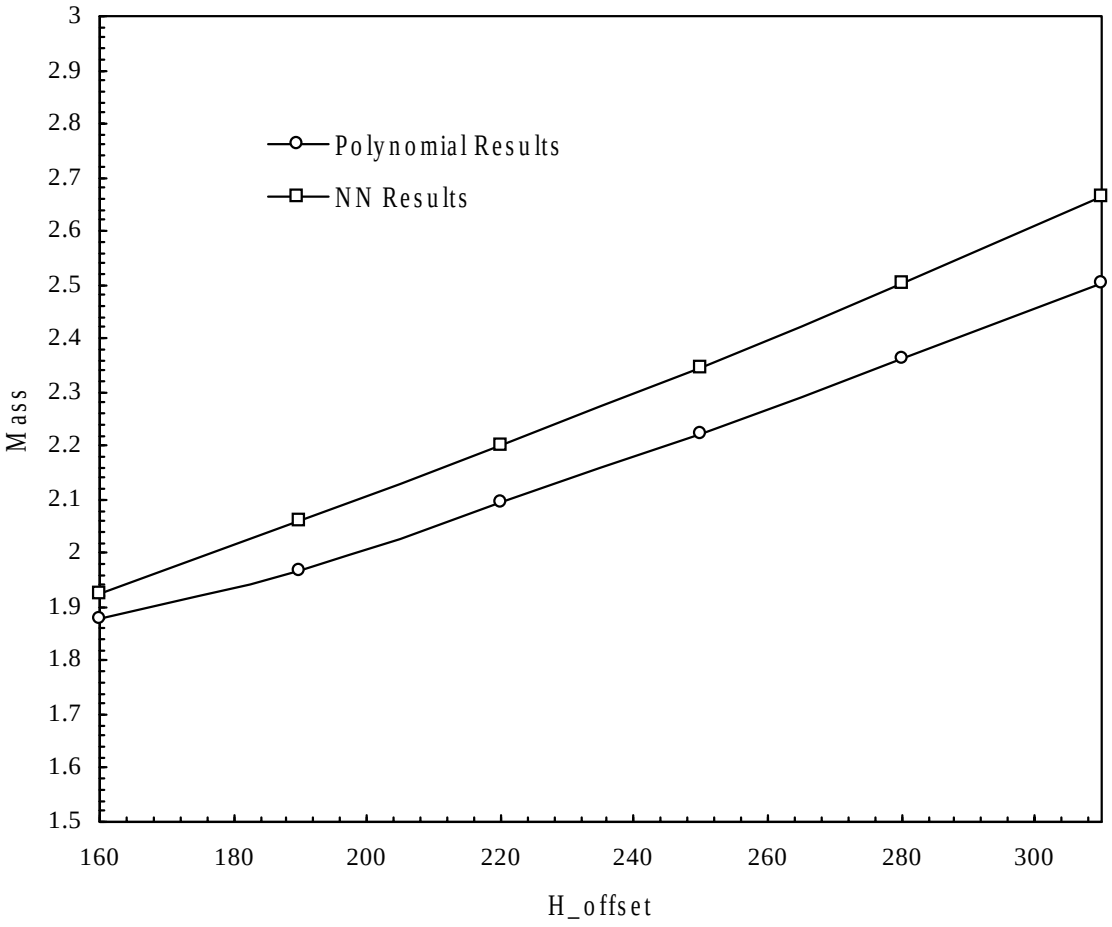


Figure 7.28: Mass of Optimum Design vs.. *H_offset*
($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

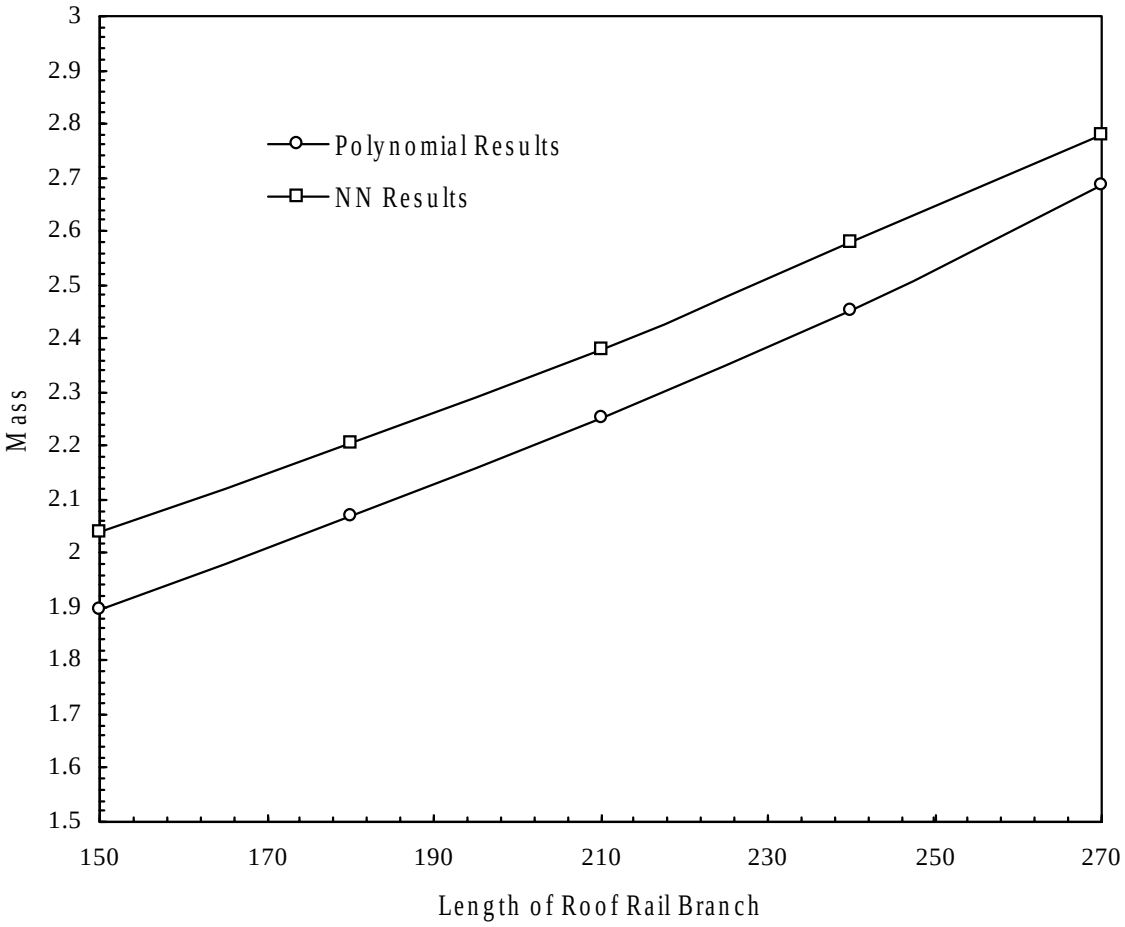


Figure 7.29: Mass of Optimum Design vs. *Length of Roof Rail Branch* ($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

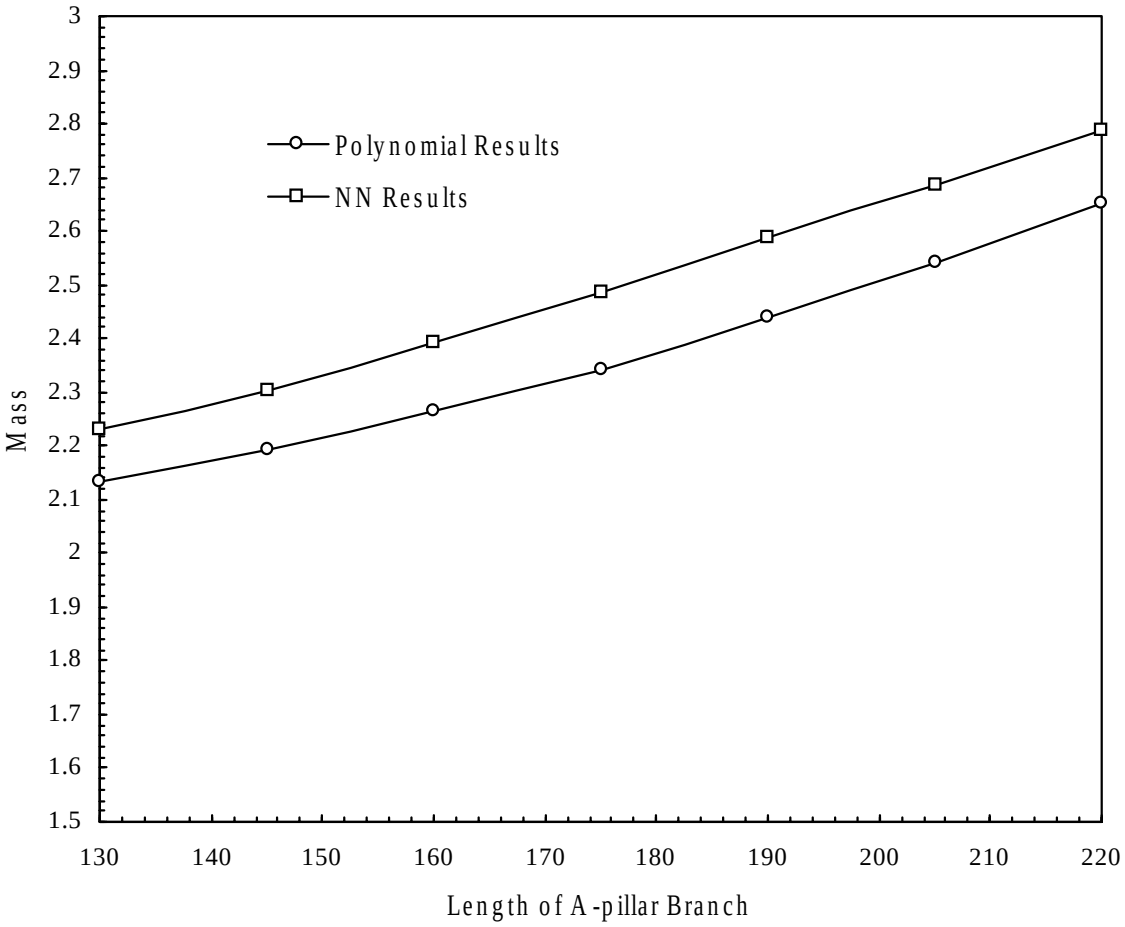


Figure 7.30: Mass of Optimum Design vs. *Length of A-pillar Branch* ($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

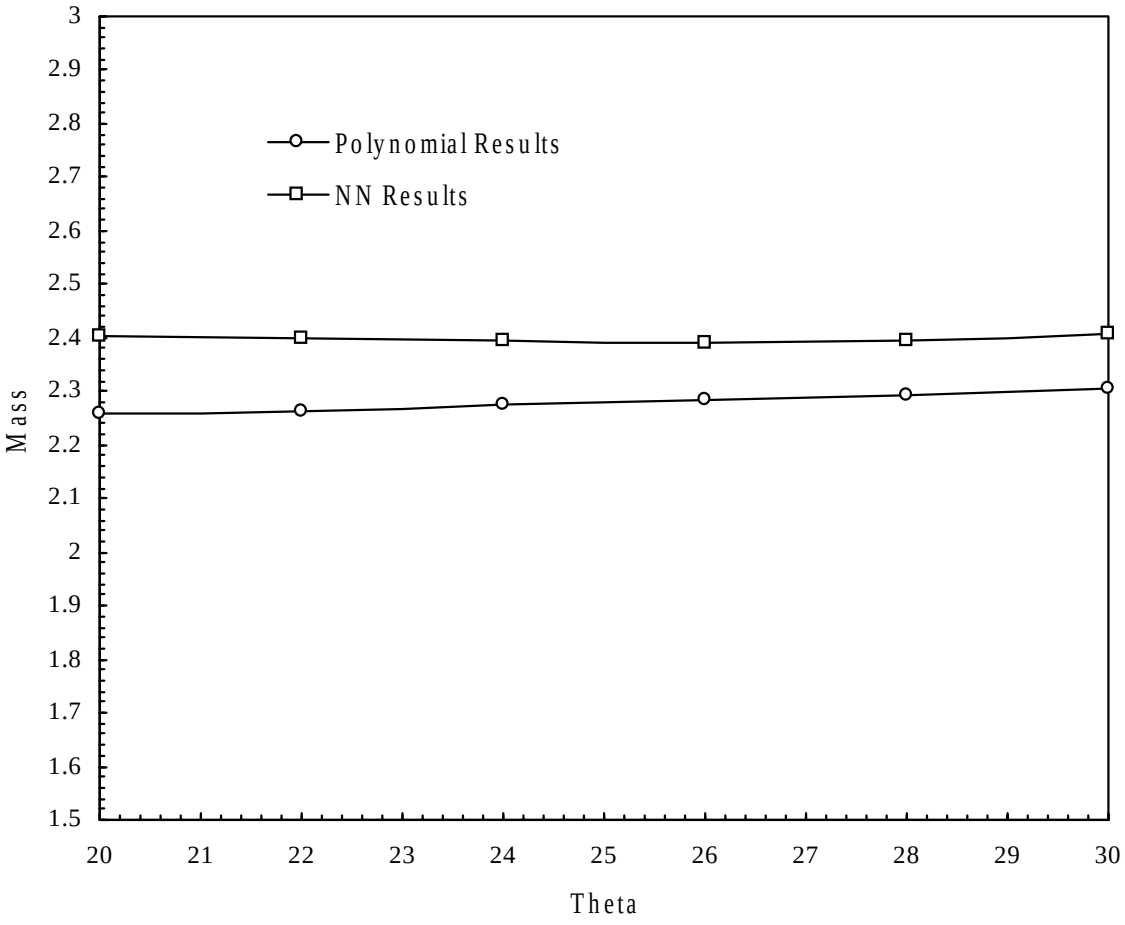


Figure 7.31: Mass of Optimum Design vs. Angle *Theta*
($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

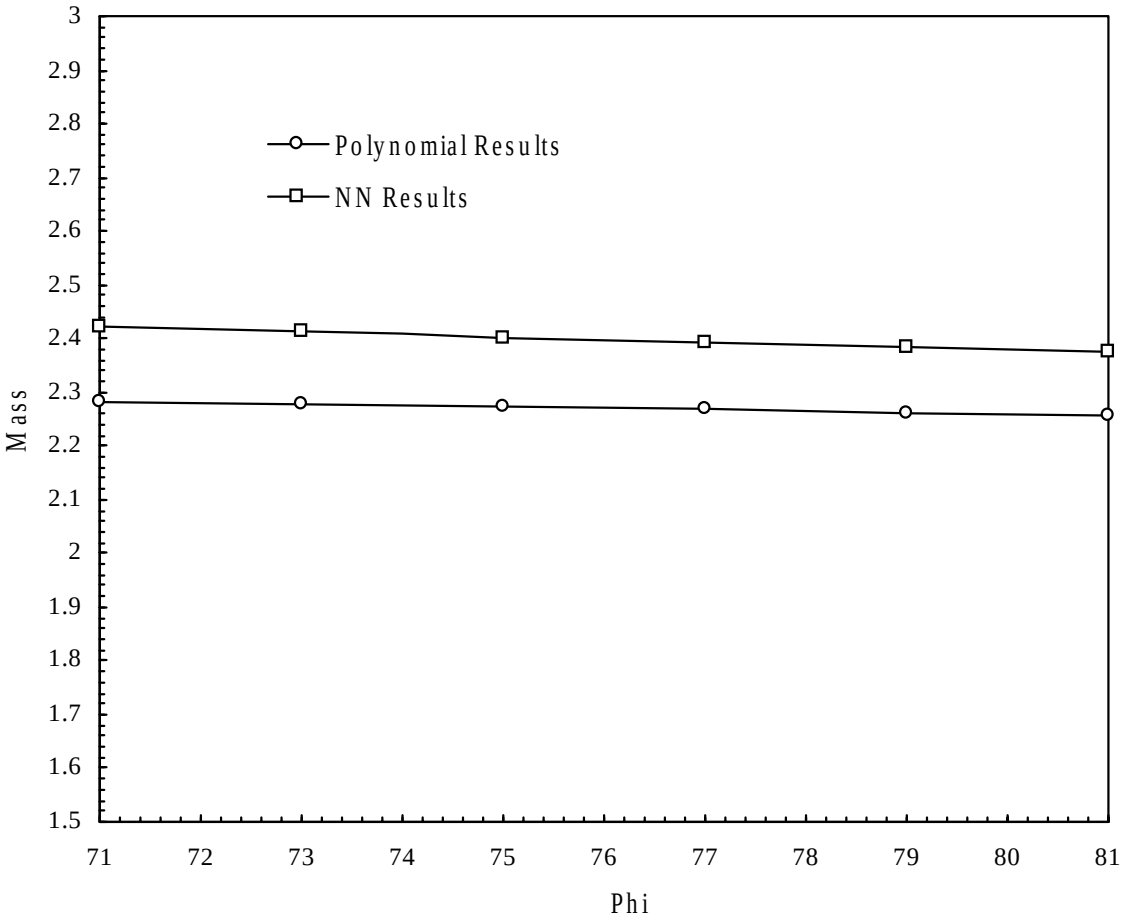


Figure 7.32: Mass of Optimum Design vs. Angle *Phi*
($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

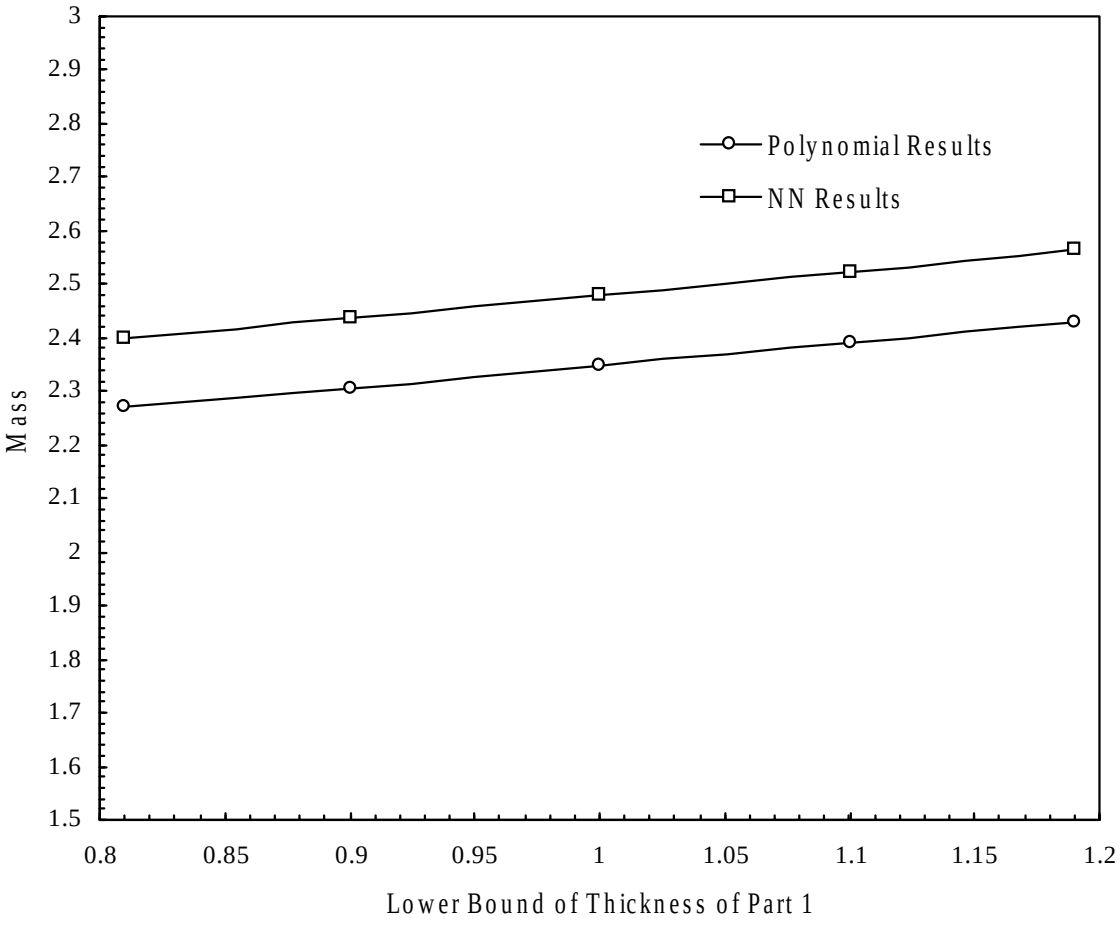


Figure 7.33: Mass of Optimum Design vs. *Thickness of Part* 1
($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

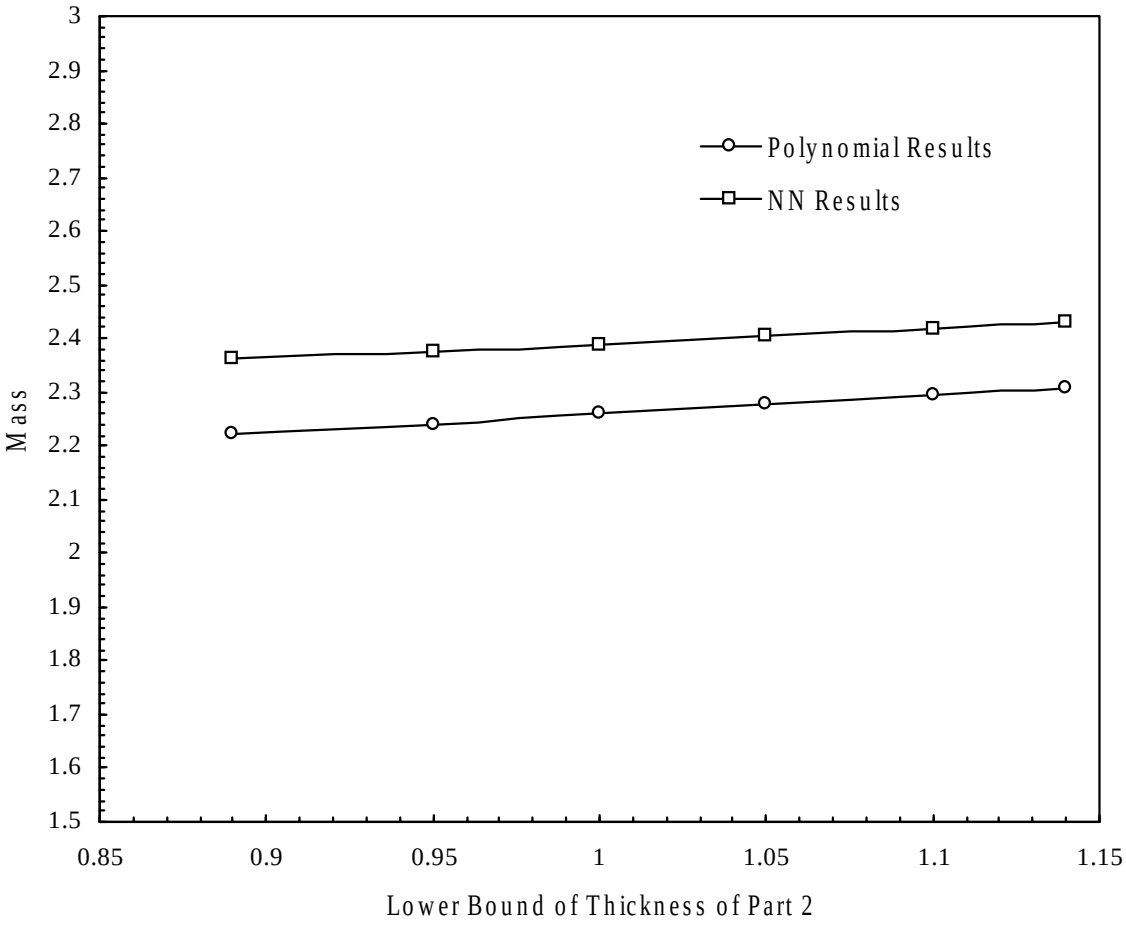


Figure 7.34: Mass of Optimum Design vs. *Thickness of Part 2*
($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

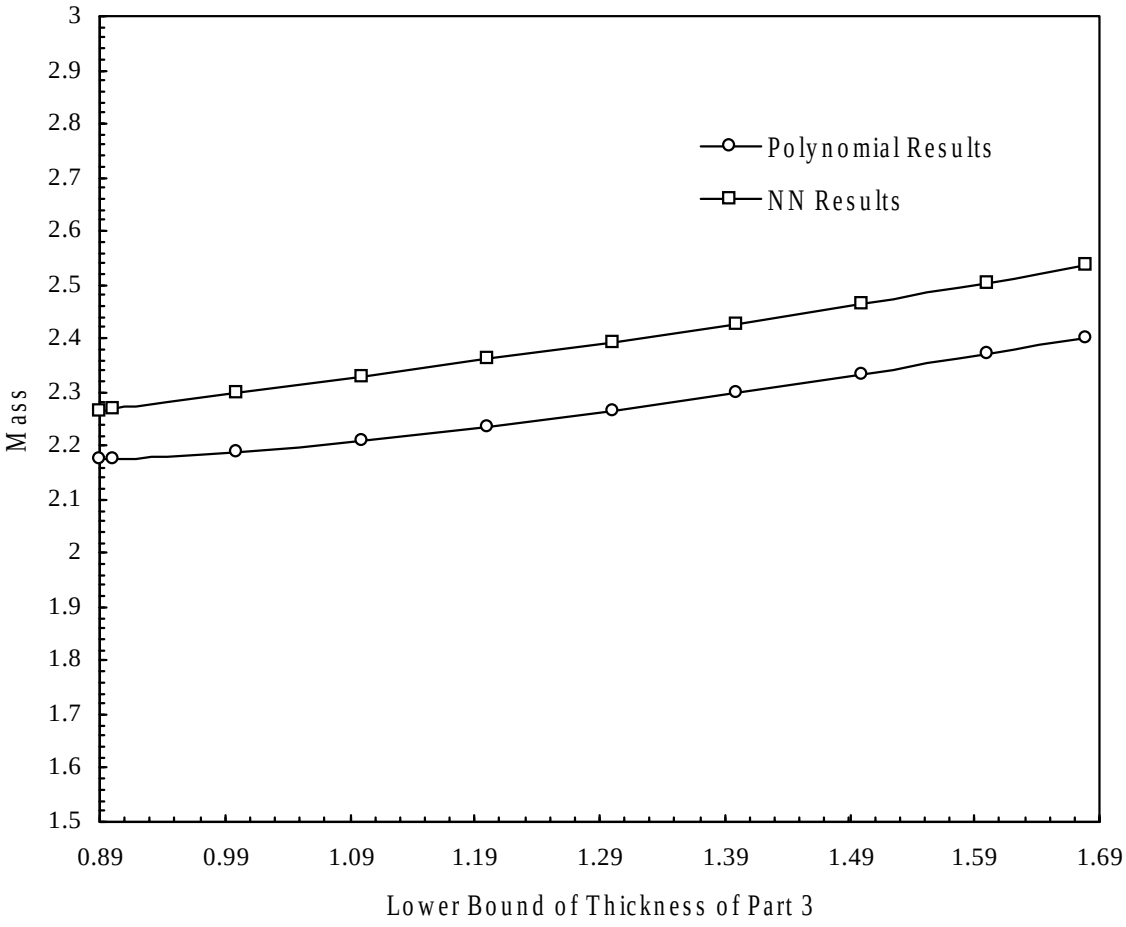


Figure 7.35: Mass of Optimum Design vs. *Thickness of Part 3*
($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

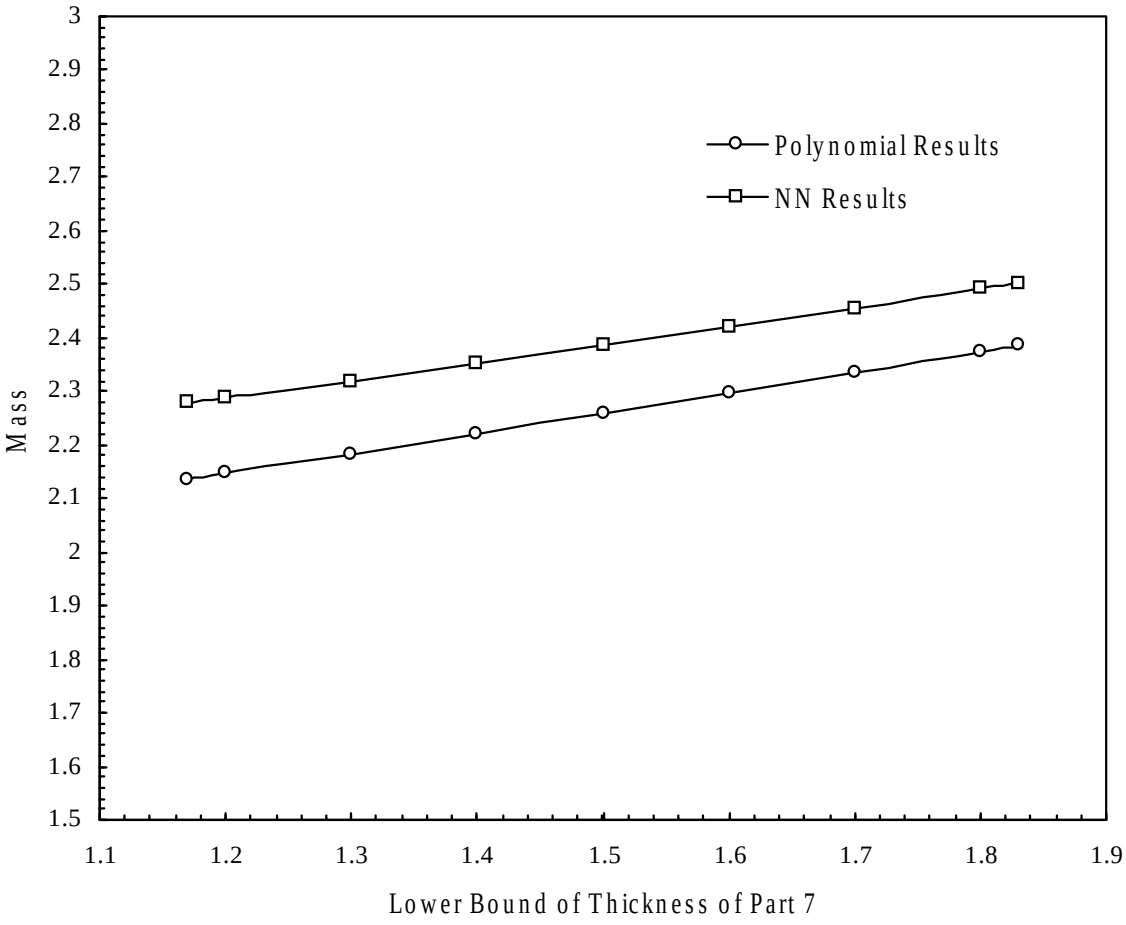


Figure 7.36: Mass of Optimum Design vs. *Thickness of Part 7*
($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

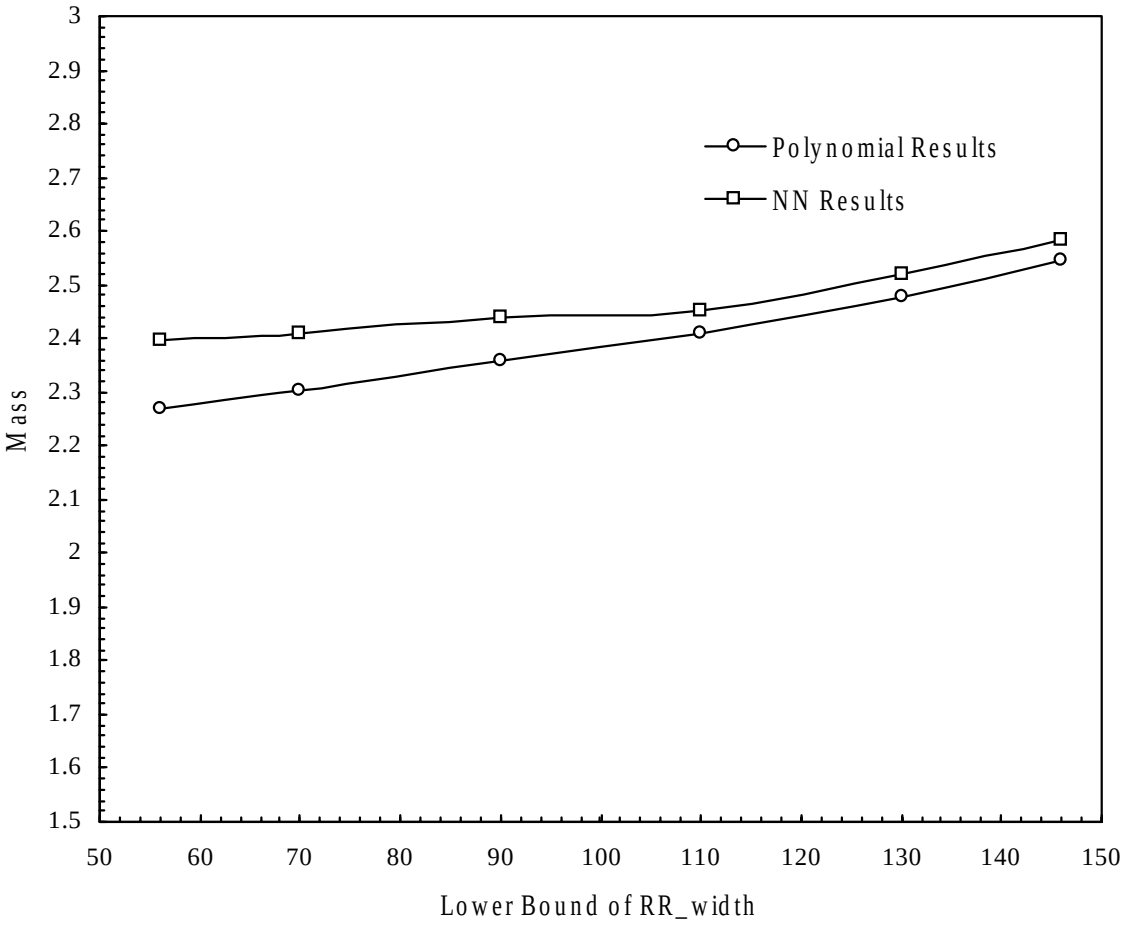


Figure 7.37: Mass of Optimum Design vs. the Lower Bound of *RR_width*
($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

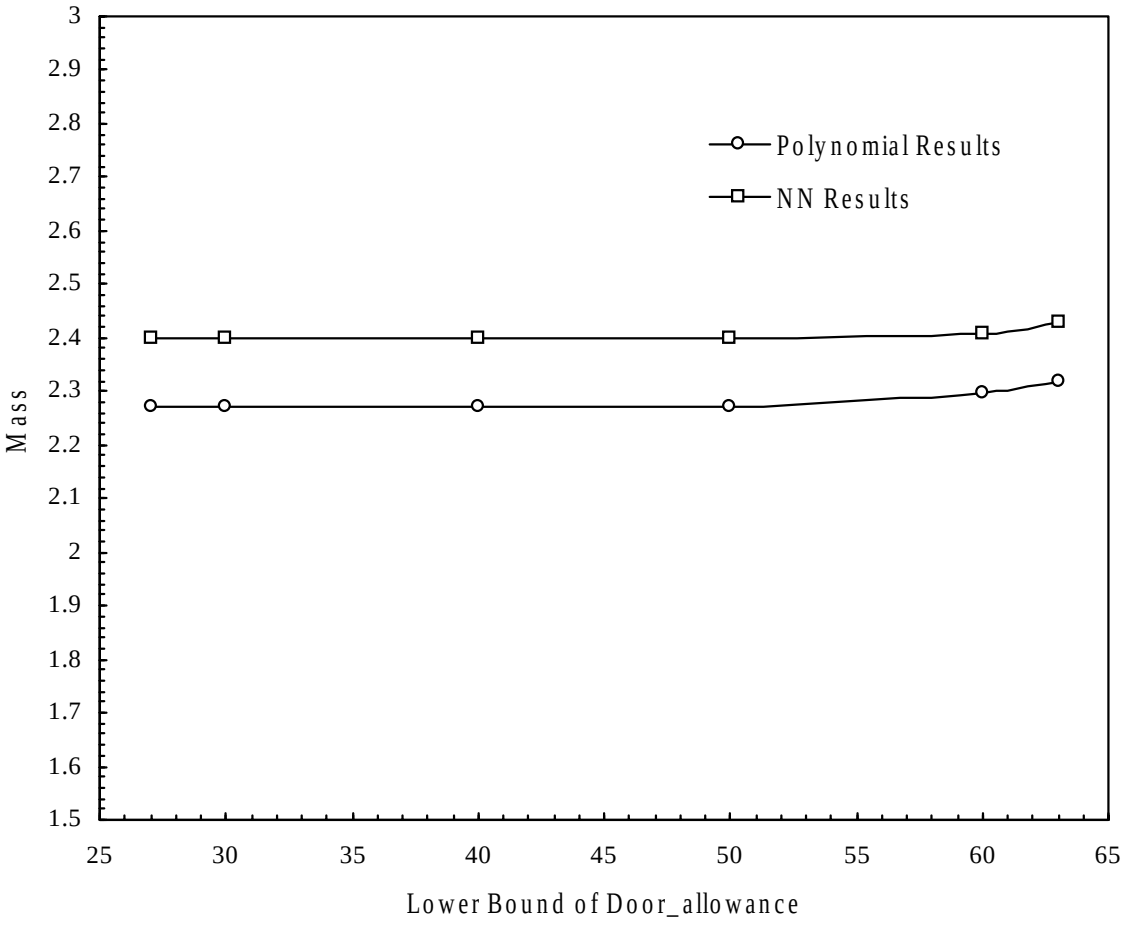


Figure 7.38: Mass of Optimum Design vs. the Lower Bound of *Door_allowance* ($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

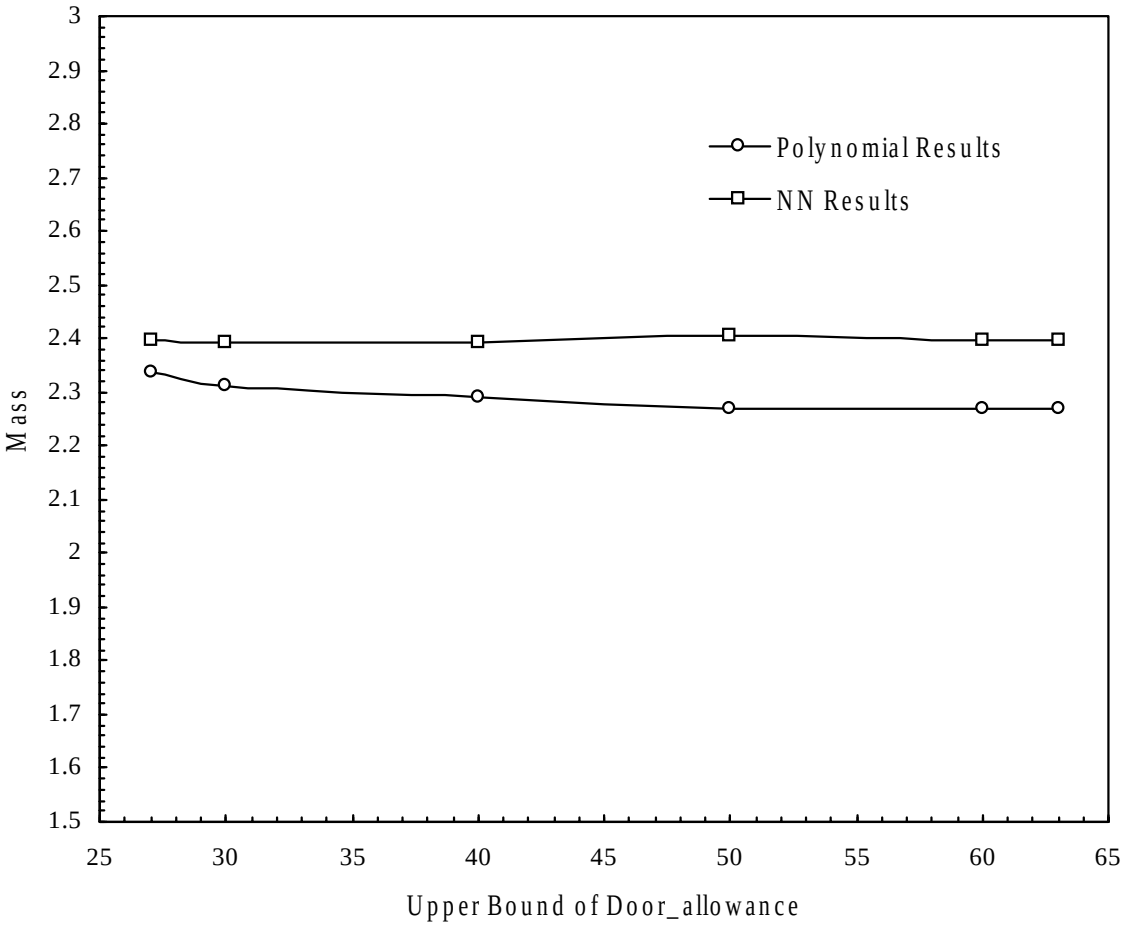


Figure 7.39: Mass of Optimum Design vs. the Upper Bound of *Door_allowance* ($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

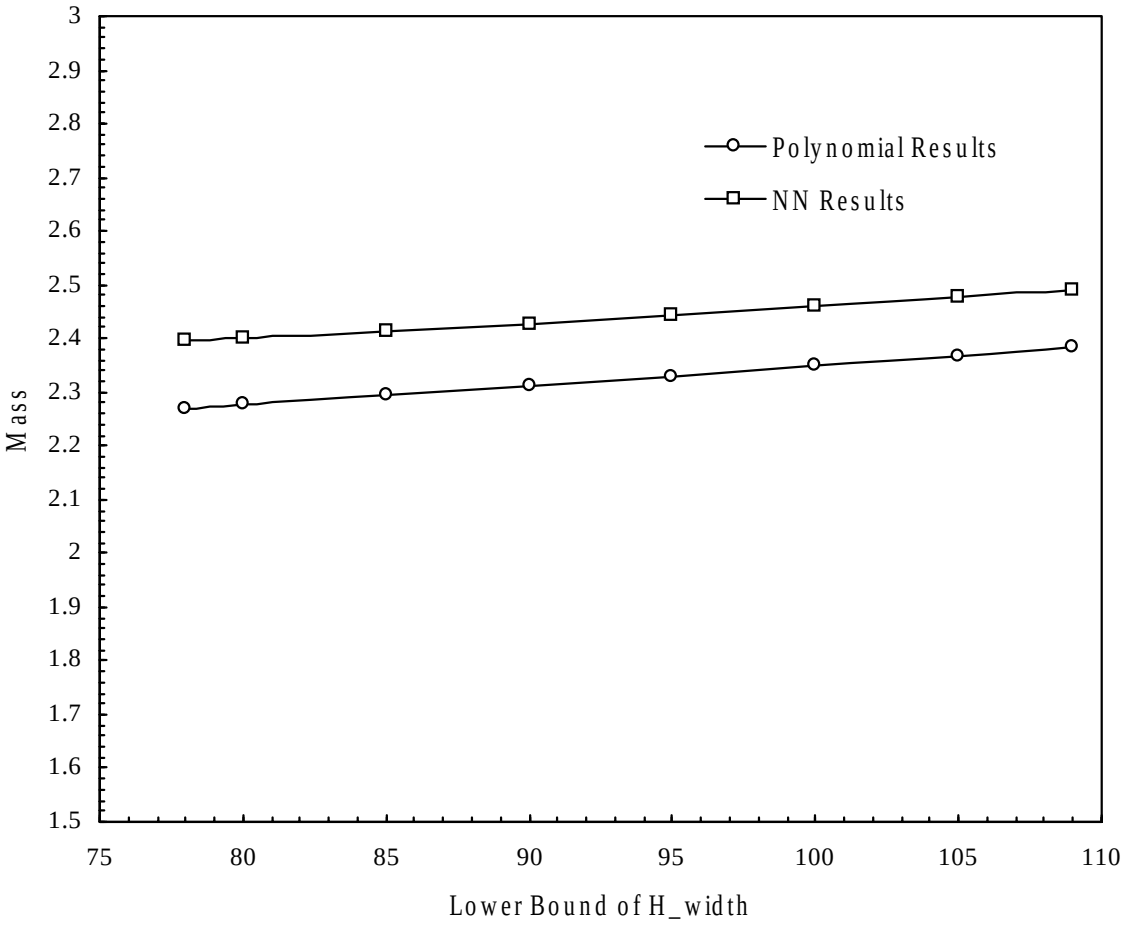


Figure 7.40: Mass of Optimum Design vs. the Lower Bound of *H_width* ($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

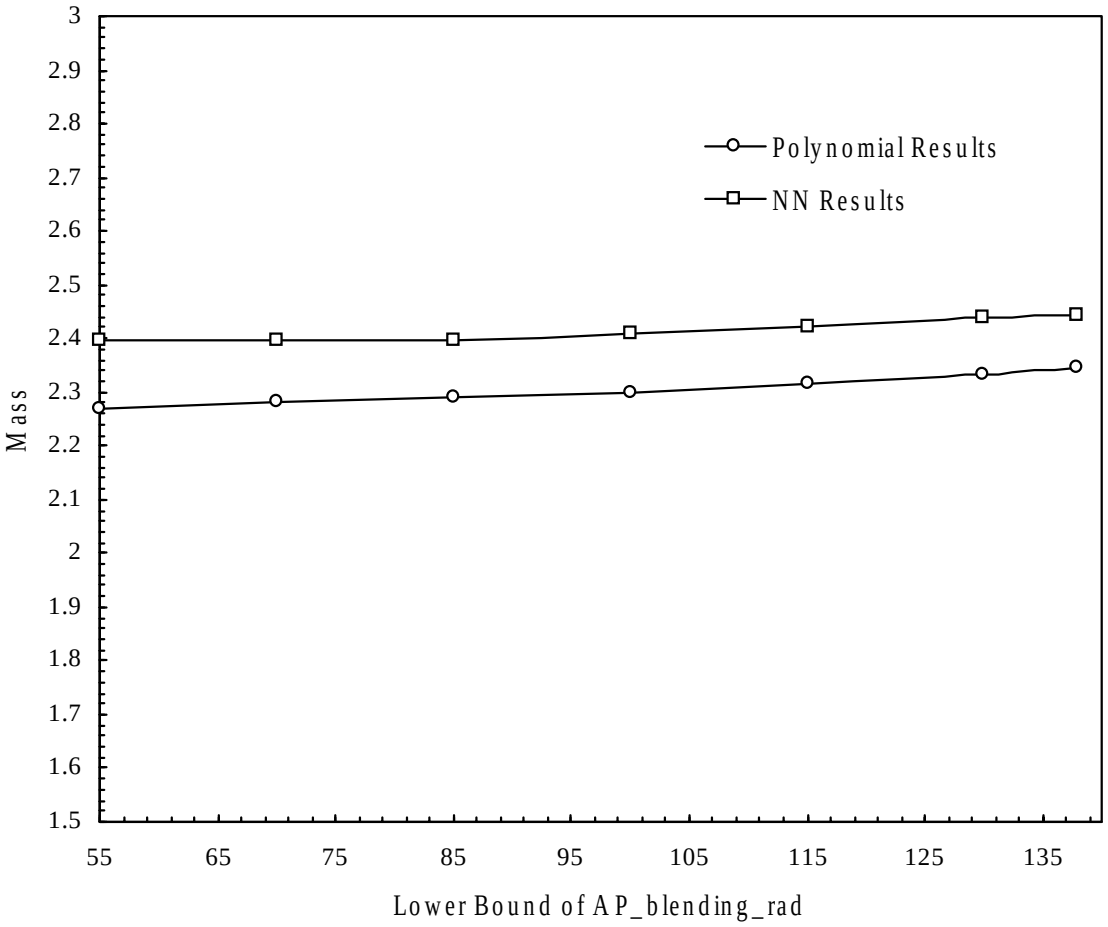


Figure 7.41: Mass of Optimum Design vs. the Lower Bound of *AP_blending_rad* ($K_{I/O} \geq 5.3936E7$, $K_{F/A} \geq 7.3234E7$, $K_{Tor} \geq 1.0945E7$)

Chapter 8
Conclusions

8.1 Conclusions

This study focused on the development of two translators for the design guidance of joints. General methodologies for developing translator A and translator B for all car joints were presented. The methodologies were demonstrated on the B-pillar to rocker and A-pillar to roof rail joints. Translator A can predict the stiffness and mass of a given joint almost instantaneously with good accuracy. Translator B designs the most efficient feasible joint that satisfies given performance (stiffness) targets.

The parametric model of the B-pillar to rocker joint created by Murphy (1995) was improved to make it more robust. A parametric model for the A-pillar to roof rail joint was created using Pro/Engineer. Using the B-pillar to rocker and the A-pillar to roof rail models, the stiffness and mass of a few actual joints were estimated and compared with experimental results. For the B-pillar joint, the error in the predictions of the I/O, F/A, and torsion stiffnesses were within the range in which measurements can vary due to hardware variability and experimental errors. For the A-pillar to roof rail joint, the Pro/Engineer model approximated the I/O and torsion stiffness well. The estimated values of F/A stiffness from the FEA model did not agree well with experimental results (The error was about 40% higher than the measured stiffness for a joint). This is due to several reasons. First, since there is no widely accepted definition of length of a branch, the three branches of a joint may be cut at different lengths than those used in the analysis. Second, since the orientation of I/O and F/A stiffnesses are difficult to measure exactly, test engineers often assume them to be in the horizontal and vertical directions. As a result, stiffnesses are defined differently in FEA and tests. Finally, the model used in FEA includes many simplifications.

A database was created for each joint by modifying the design variables of the joint randomly and by calculating the stiffnesses and masses of the resulting designs. Methods for *design of experiments* (DOE) are generally more efficient than a random method for creating designs. However, they cannot be directly applied to problems that involve many constraints that restrict the region of the design variables.

We studied the ranking of design variables in terms of importance on the stiffness and mass of the joint using the database and stepwise regression. *F*-ratio, which measures the relative importance of design variables, was obtained using a linear polynomial model. The design variables with the biggest *F*-ratio affect significantly the stiffness or mass. The important design variables were identified and ranked according to the value of *F*-ratio. These results can help improve design of joints.

Translator A was created using both response surface polynomials and neural networks. Different polynomial models, including, linear polynomial models, second degree polynomial models, and double regression models, were studied. A second regression is performed in the double regression using the results from the linear polynomial model to improve the prediction. Three layer neural networks, which have one input layer, one hidden layer and one output layer, were used to create the neural network translators. Cross-validation was used in this study to avoid overtraining the neural networks. Specifically, all designs were split into three groups. The first group was used to train a neural network. The second group was used to determine when to stop training the neural network. The training process stopped when the error corresponding to the second group reaches its minimum. The third group was used to test the generalization performance of the trained neural network. Different neural network architectures (the number of input design variables, and the number of neurons in the hidden layer) were studied, from which the one with the best generalization performance was chosen to be the neural network translator. Translator A can greatly reduce the time of analyzing a joint design and thus enables designers to optimize a joint.

Two optimization programs (translators B) were developed. These tools find the best, feasible design when the user specifies the stiffness requirements and the ranges of the

design variables. Translator A was incorporated into the program of translator B. The user can use the polynomial translator A or the neural network translator A. Translator B accounts for manufacturing, packaging, styling, performance and mathematical constraints. These constraints help ensure that the optimum design from translator B is feasible and its stiffness and mass are predicted with reasonable accuracy by the translator A. Two optimization algorithms, namely, the modified feasible direction method (MFD) and sequential linear programming (SLP), were tested in translator B. SLP is the default algorithm since it generally converges faster than the MFD in the problems considered. The user can easily switch optimization methods in the program.

Several methods were used to validate translator B. First, optimum designs from translator B were checked to make sure that they corresponded to a global minimum by starting from different initial points. In all cases, the translator B yielded the same design. Both MFD, SLP and another optimization program written in Mathematica were used to solve the same problem. The solutions were almost identical in all cases considered. Second, a number of random stiffness requirements were generated. The optimum designs corresponding to each set of stiffness requirements were obtained using translator B. Then, the optimum results were compared with FEA results

The two translators developed in this study should help set up meaningful performance targets for joints. They should also improve designs of new car models because they allow designers analyze many designs rapidly.

This study shows that:

a) We can simulate FEA for predicting the stiffness of joints using a polynomial or a neural network.

b) It is easier to predict the stiffness and mass of the B-pillar to rocker joint than the stiffness and mass of the A-pillar to roof rail joint.

c) It is much easier to predict the mass than the stiffness.

d) The neural network fitted the data used for training slightly better than the polynomial. However, the accuracy of polynomial deteriorated less than that of the neural network when both models were used to analyze designs they had not seen before.

f) Based on the results, we cannot conclude that either the response surface polynomials or the neural networks were consistently more accurate in predicting the stiffness and mass of designs that they had not seen before.

The above conclusions are different from these reached by Zhu (1994). The reason is that Zhu used complete second degree polynomials, which contained redundant terms. This study used stepwise regression, which filtered out these terms.

g) In general, when the stiffness target are too high, translator B yields designs whose true stiffness can be significantly different from the predicted and the target stiffnesses. However, there is not such problem when the stiffness targets are reasonable.

h) The neural network translator B consistently yielded heavier designs than the polynomial translator B.

i) Translator B was found to be efficient and robust. The reason is that it uses polynomial or neural network models to analyze a design, which are much more efficient than FEA

8.2 Recommendations for Future Work

Following are a few tasks that can improve the methodologies for the translators and facilitate their integration to the design process in the automotive industry.

a) Develop consistent definitions of some joint parameters

There is no widely acceptable definition of many parameters of joints. In many cases, these parameters affect seriously stiffness calculations and test results. This lack of consistency can be a serious problem when comparing joints and setting stiffness targets.

It is important to establish a protocol for analysis and testing of joints. Specifically, we need establish consistent definitions of the following:

- The joint center
- The length(s) of the joint branches
- Reference point(s) relative to which the lengths of the branches must be measured

- The orientations of cutting planes
- The orientations of the planes about which the branches are assumed to rotate when calculating stiffness

The parameters used to define the above quantities should be meaningful to designers and easy to measure.

b) Automating the process for developing a database for translator A

Developing a database of examples that are used to develop translator A was very time consuming. It took two to three months to develop such a database. The reason is mainly that many designs created by randomly changing the design variables are found infeasible. Moreover, some designs cannot be simulated by the CAD software although they are feasible. Since in industries, the designer may need to design and/or improve a joint in a much shorter time, it will be helpful to automate the process of developing the database. The automation of such process requires to interface different software packages. Because of the limitations of CAD software, it is still a difficult task to develop an error free program that can automate such work. Some progress has been achieved that can get the FEA information for more than 50 designs in a few hours. Once the automating program is finished, it will reduce the time of developing a database to a few days.

c) Universal translators

The methodology developed in this dissertation is for constructing a translator that works only for a specific joint architecture. Thus, an engineer has to develop a new translator when he/she wants to consider a new architecture. A *universal translator*, which can process many types of joint architectures, should be more effective and economical for design of joints.

This task should establish a methodology for a universal translator that can predict the performance characteristics of most types of joint architectures that are important to the automotive industry. The translator should be fast (analyze a joint at a small fraction of a

second), and accurate (average error in stiffness and mass should be less than 10% to 15%).

8.3 Deliverables

The deliverables of the study presented in this dissertation include the methodology for developing translators, the parametric models of two joints and the translators A and B for these joints. Our B-pillar to rocker model and translators A and B for the B-pillar to rocker joint are used by an automotive company. The model and the two translators can rapidly give engineers the information needed in the preliminary design stage of a car, help them improve the design of joints and minimize the total weight of the car. This in turn helps improve the overall static and dynamic characteristics of the car. The methodology used in this project can be applied for the parameterization and modeling of other joints or other car components such as beams.

Bibliography

Akaike, H. "A New Look at the Statistical Model Identification," IEEE Transactions on Automatic Control, Vol. 19, 1974, pp.716-723.

Alon, N., Dewdney, A. K., Ott, T. J. "Efficient Simulation of Finite Automata by Neural Nets," Journal of the Association for Computing Machinery, Vol. 38, No. 2, April 1991, pp.495-514.

Amari, S., Murata, N., Müller, K. R., Finke M. and Yang H. "Statistical Theory of Overtraining-Is Cross-Validation Asymptotically Effective?" Advances in Neural Information Process Systems 8, Proceeding of the 1995 Conference, 1995, pp.176-182

Balabanov, V. O. "Development of Approximations for HSCT Wing Bending Material Weight Using Response Surface Methodology," Ph.D. Dissertation, Department of Aerospace and Ocean Engineering, Virginia Polytechnic Institute and State University, 1997.

Belegundu, A. D. "Optimizing the Shapes of Mechanical Components," Mechanical Engineering, Vol. 115, No. 1, 1993, pp. 46-48.

Berke, L., and Hajela, P. "Application of Artificial Neural Nets in Structural Mechanics," Structural Optimization, Vol. 4, 1992, pp. 90-98.

Botkin, M. E. " Three-Dimensional Shape Optimization Using Fully Automatic Mesh Generation," AIAA Journal, Vol. 30, No. 7, 1992, pp. 1932-1934.

Botkin, M. E., and Lust, R. V. "A Neural Network Application to Shape Optimization," 961102, General Motors Research and Development Center.

Box, G. E. P. and Behnken, D. W. “Some New Three Level Designs for the Study of Quantitative Variables,” Technometrics, Vol. 2, No. 4, 1960, pp. 455-475.

Box, G. E. P., Hunter, W. G. and Hunter, J. S. “Statistics for Experimenters,” John Wiley and Sons Inc., New York, 1978.

Carpenter, W. C., and Barthelemy, J. M. “A Comparison of Polynomial Approximations and Artificial Neural Nets as Response Surface,” Proceedings of the 33rd AIAA/ASME/ASCE/AHS/ASC SDM Meeting, Dallas, Texas, 1992, pp. 2474-2482.

Chang, D.C. “Effects of Flexible Connections on Body Structural Response,” SAE Transactions, Vol. 83, 1974, pp.233-244.

Choi, K. K., Santos, J. L. T., and Frederick, M. C. “Implementation of Design Sensitivity Analysis with Existing Finite Element Codes,” Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 109, September 1987, pp. 385-391.

Cohn, D. A. “Neural Network Exploration Using Optimal Experiment Design,” Advances in Neural Network Information Processing Systems 6, 1994, pp. 679-686.

Cook, D. F., and Shannon, R. E. “A Predictive Neural Network Modeling System for Manufacturing Process Parameters,” International Journal of Production Research, Vol. 30, No. 7, 1992, pp. 1537-1550.

Cybenko, G. “Approximation by Superpositions of a Sigmoidal Function,” Mathematics of Control, Signals, and Systems, Vol. 2, No. 4, 1989, pp. 303-314.

Dodier, R. “Geometry of Early Stopping in Linear Networks,” Advances in Neural Information Processing Systems 8, Proceedings of the 1995 Conference, 1995, pp.365-371.

Fenyes, F. “Structural Optimization with Manufacturing Considerations,” Structural Optimization, Vol. 5, 1992, pp.116-122.

Ghahramani, Z., and Jordan, M. I. “Supervised Learning from Incomplete Data via an EM approach,” Advance in Neural Information Processing Systems 6, 1994, pp.120-127.

Giunta, A. A. “Aircraft Multidisciplinary Design Optimization Using Design of Experiments Theory and Response Surface Modeling Methods,” Ph.D. Dissertation, Department of Aerospace and Ocean Engineering, Virginia Polytechnic Institute and State University, 1997.

Gorman, J. W., and Toman, R. J. “Selection of Variables for Fitting Equations to Data,” Technometrics, Vol. 8, No. 1, 1966, pp.27-51.

Guyot, N. “ Fuzzy Logic and Utility Theory for Multi-objective Optimization of Automotive Joints,” Master Thesis, Department of Aerospace and Ocean Engineering, Virginia Polytechnic Institute and State University, 1996.

Haftka, R. T., and Gürdal, Z. “Elements of Structural Optimization,” Kluwer Academic Publishers, 1992.

Hajela, P., and Berke, L. “Neurobiological Computational Models in Structural Analysis and Design,” AIAA/ASME/ASCE/AHS/ASC 31st Structures, Structural Dynamics, and Materials Conference, A Collection of Technical Papers, 1990, pp.345-353.

Hardee, E., Chang, H., Choi, K. K., Yu, X., and Grindeanu, I. “A CAD-Based Design Sensitivity Analysis and Optimization for Structural Shape Design Applications,” 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1996, pp.77-87.

Huang, S. C., and Huang, Y. F. “Bounds on the Number of Hidden Neurons in Multilayer Perceptrons,” IEEE Transactions on Neural Networks, 1991, Vol. 2, No. 1, January 1991, pp. 47-55.

Hugh, D. R., and Horne, B. G. “Progress in Supervised Neural Network,” IEEE Signal Processing Magazine, January 1993, pp. 8-38.

Hughes, O. F. “Ship structural design : a rationally-based, computer-aided, optimization approach,” Wiley, 1983.

Jagota, A. “Neural Network Methods for Optimization Problems,” Advances in Neural Information Processing Systems 6, 1994, pp.1184-1185.

Keeler, J. “Vision of Neural Networks and Fuzzy Logic for Prediction and Optimization of Manufacturing Processes,” Application of Artificial Neural Networks, Vol. 3, 1992, pp.447-455.

Kennard, R. W. “A Note on the C_p Statistic,” Technometrics, Vol. 13, No. 4, 1971, pp.899-900.

Li, J., and Parsons, M. G. “Forecasting Tanker Freight Rate Using Neural Networks,” submitted for Maritime Policy and Management, University of Michigan, 1995.

Liu, Y. “Robust Parameter Estimation and Model Selection for Neural Network Regression,” Advances in Neural Information Processing Systems 6, 1994, pp.192-199.

Liu, Y., Kapania, R. K., and VanLandingham, H. F. “Simulating and Synthesizing Substructures Using Neural Network and Generic Algorithms,” International Conference on Computational Engineering Science, Manuscript, 1998.

Lindby, T., and Santos, J. L. T. " Shape Design Sensitivity Analysis and Optimization with An Existing Associative CAD System," 5th Symposium on Multidisciplinary Analysis and Optimization, Panama City, Florida, September, 1994, pp.1483-1490.

Ling, Q. "Design of Automotive Joints Using Response Surface Polynomials and Neural Networks," Master Thesis, Department of Aerospace and Ocean Engineering, Virginia Polytechnic Institute and State University, 1998.

Lippmann, R. P. "An Introduction to Computing with Neural Nets," IEEE Acoustics, Speech and Signal Processing Magazine, Vol. 4., No. 2, April, 1987, pp.4-22.

Mallows, C. L. "Some Comments on C_p ," Technometrics, Vol. 15, No. 4, 1973, pp.661-675

Maeder, R. E. "Programming in Mathematica," Addision Wesley Publisher, 1997.

Murphy, S. T. "A Parametric Model for Rapid Analysis of Automotive Joints," Master Thesis, Department of Aerospace and Ocean Engineering, Virginia Polytechnic Institute and State University, 1995.

Nikolaidis, E., and Lee, K. "A 3-D Joint Model for Automotive Structures," 8th International Conference on Vehicle Structural Mechanics and CAE, Traverse City, Michigan, 1992.

Nikolaidis, E., Long, L., and Ling, Q. "Neural Networks and Response Surface Polynomials for Design of Vehicle Joints," AIAA-98-1777, 1998.

Nikolaidis, E., Long, L., and Ling, Q. "Neural Networks and Response Surface Polynomials for Design of Vehicle Joints," for Computer and Structures, manuscript, 1998.

Pries, H., and Wille, H. C. “Some Examples of Modern Vehicle Design Analysis by the Finite Element Method,” International Journal of Vehicle Design, Vol. 5, No.1, 1984, pp.53-66.

Plutowski, M., Sakata, S., and White H. “Cross-Validation Estimates IMSE,” Advances in Neural Information Processing System 6, 1994, pp. 391-398.

Puttre, M. “Gearing Up for Conceptual Design,” Mechanical Engineering, Vol. 115, No. 3, 1993, pp.46-50.

Puttre, M. “Putting Optimization Routines in the Loop,” Mechanical Engineering, Vol. 115, No. 7, 1993, pp.76-80.

Rehak, D. R., and Thewalt, C. R., ASCE, A. M., and Doo, L. B. “Neural Network Approaches in Structural Mechanics Computations, Computer Utilization in Structural Engineering,” Proceeding of the Session Related to Computer Utilization at Structures Congress, 1989, pp.168-176.

Rogers, J. L. “Simulating Structural Analysis with Neural Network,” Journal of Computing in Civil Engineering, Vol. 8, No. 2, 1994, pp. 252-265.

Rosenblatt, F. “The Perceptron: A Probabilistic Model For Information Storage and Organization in the Brain,” Psychological Review, Vol. 65, 1958, pp. 386-408.

Roux, W. J. Stander, N., and Haftka, R. T. “Response Surface Approximations for Structural Optimization,” 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1996, pp. 565-578.

SAS Institute Inc. “JMP Introductory Guide,” 1995.

SAS Institute Inc. “JMP Statistics and Graphics Guide,” 1995.

SAS Institute Inc. “JMP User's Guide,” 1995.

Schramm, U., Pilkey, W. D., Devries, R. I., and Zebrowski, M. P. “ Shape Design for Thin-Walled Beam Cross Sections Using NURBS,” Proceedings of the Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization (MA&O), Panama City, Florida, 1994, pp.1-11.

Schramm, U., and Pilkey, W. D. “The Coupling of Geometric Descriptions and Finite Elements Using NURBs – A Study in Shape Optimization,” Finite Elements in Analysis and Design, Vol. 15, 1993, pp.11-34.

Shi, Q., and Hagiware, I. “Structural Optimization Based on Holographic Neural Network and its Extropolations,” AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Symposium, 1998, pp. 2124-2132, No. AIAA-98-4975.

Shieh, R. C. “Neural Network Assisted Large-Scale Structural Analysis/Reanalysis in a Massively Parallel Environment,” AIAA-94-1512-CP.

Sollich, P., and Krogh, A. “Learning with Ensembles: How over-fitting can be useful,” Advances in Neural Information Processing System 8, Proceedings of the 1995 Conference, 1995, pp.190-196.

Sunami, Y., Yugawa, T., and Yoshida, Y. “Analysis of the Joint Rigidity In-Plane Bending of Plane-Joint Structures,” Japanese Society of Automotive Engineers, Vol. 9, No. 2, 1987, pp. 44-51.

Sunami, Y., Yugawa, T., and Yoshida, Y. “Analysis of the Joint Rigidity of the Automotive Body Structure - Out-of-Plane Bending of Plane-Joint Structures,” Japanese Society of Automotive Engineers, Vol. 11, No. 3, 1990, pp.59-66.

Swift, R. A., and Batill, S. M. “Application of Neural Network to Preliminary Structural Design,” AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural

Dynamics, and Materials Conference, 1991, A Collection of Technical paper, 1991, pp. 335-343.

Swift, R. A., and Batill, S. M. "Simulated Annealing Utilizing Neural Networks for Discrete Variable Optimization Problems in Structural Design," The 33rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference, 1992, A Collection of Technical Paper, 1992, pp. 2536-2544.

Tresp, V., Ahmad, S., and Neuneier, R. "Training Neural Networks with Deficient Data," Advances in Neural Information Processing Systems, Vol. 6, 1994, pp. 128-135.

Vanderplaats Research & Development, Inc. "DOT Users Manual," Version 4.20, 1995.

Venter, G., Haftka, R. T., and Starnes, J. H. "Construction of Response Surfaces for Design Optimization Applications," 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1996, pp.548-564.

Wu, B. "An Introduction to Neural Networks and Their Applications in Manufacturing," Journal of Intelligent Manufacturing, Vol. 3, 1992, pp. 391-403.

Wu, L., and Moody, J. "A Smoothing Regularizer for Recurrent Neural Networks," Advances in Neural Information Processing Systems 8, Proceedings of the 1995 Conference, 1995, pp.458-464.

Zhu, M. "Translators for Design Guidance of Joints in Automotive Structures," Ph.D. Dissertation, Department of Aerospace and Ocean Engineering , Virginia Polytechnic Institute and State University, 1994.

Vita

Luohui Long was born in Luoyang, Henan province, China on January 21, 1969. In 1990 he graduated from National University of Defense Technology with a Bachelor of Science degree in Applied Mechanics. Since 1990, he studied at Beijing Institute of Astronautical Systems Engineering as a graduate student. He earned his Master of Science degree in Structural Design and Structural Optimization from Beijing Institute of Astronautical Systems Engineering. After graduation in 1993, he worked in the same institute as a research engineer for one and a half years. In August 1994 he continued his education at Virginia Polytechnic Institute and State University in Blacksburg, Virginia. While studying at Virginia Polytechnic Institute and State University he worked as a co-operative student at American Bureau of Shipping for six months. In 1998 he graduated from Virginia Polytechnic Institute and State University with a Doctor of Philosophy degree in Aerospace Engineering.