

Structural Optimization Using the Principle of Virtual Work and an Analytical Study on Metal Buildings

Christopher Douglas Barrar

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science

In

Civil Engineering

Finley A. Charney

W. Samuel Easterling

Cristopher D. Moen

June 8, 2009

Blacksburg, Virginia

Keywords: Virtual Work, Structural Optimization, Column-Base Connections

Structural Optimization Using the Principle of Virtual Work and an Analytical Study on Metal Buildings

Christopher Douglas Barrar

ABSTRACT

A tool for analyzing and understanding the behavior of structural systems based on the principle of virtual work was developed by Dr. Finley Charney in the early 1990s. The program was called DISPAR, which stands for DISplacement PARticipation factors, and was written to work in accordance with SAP90 and ETABS. This program became outdated once newer versions of SAP90 and ETABS were released. Starting with version 11 of SAP2000, Computers and Structures released an Open Application Programming Interface (OAPI) which allowed programmers efficient access to the information in SAP2000. With this release came the motivation to update the program DISPAR to work with SAP2000 version 11 and other versions to follow. This thesis provides an overview of how the new version of DISPAR was programmed using VB.Net and OpenGL.

This thesis starts off with an in depth discussion and literature review on the development of the principle of virtual work. The literature review covers how virtual work can be used as a tool to understand structural behavior as well as optimize structural performance.

The updated version of DISPAR (DISPAR for SAP2000) was then used to analyze the behavior of metal building frames under various loadings. The focus of this study was to determine the effect modeling the column base connection as partially rigid has on wind drift in metal building frames. Before beginning the study, a literature review was conducted on the rotation stiffness provided by typical column base connections. The information obtain in the literature review was then used to create a finite element model of a typical column base connection in a metal building. Once the finite element model was completed, DISPAR for SAP2000 was used to conduct a study on the sensitivity of the rotational stiffness of the column base connection.

ACKNOWLEDGMENTS

I would like to thank Dr. Finley A. Charney for serving as my primary advisor and Committee Chair. The work I completed on the update version of DISPAR would not have been possible without his guidance along the way. I am very thankful for all that I have learned from him at my time here at Virginia Tech both as an Undergraduate and a Graduate student. I would also like to thank Dr. Samuel Easterling and Dr. Cristopher Moen for serving on my committee and for their valuable input into my thesis. I would also like to thank The Center for Extreme Load Effects on Structures (CELES) at Virginia Tech and MBMA for supporting me financially through my Graduate studies. In addition, I would like to thank SDL Structural Engineers for providing me with the opportunity to work as an intern while allowing me to spend much of my time working on research. For their financial support and the experienced I gained while working there I am very grateful.

Also, without the support of my family and friends along the way I would not have been able to accomplish the goals that I have. I would like that thank them for always being there for me and supporting me when I needed a helping hand. The friendships I have developed here at Virginia Tech will stay with me throughout the rest of my life.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGMENTS.....	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES.....	viii
LIST OF TABLES.....	xii
CHAPTER 1: INTRODUCTION.....	1
1.1 MOTIVATION FOR RESEARCH.....	1
1.2 VIRTUAL WORK.....	2
1.3 DISPAR.....	3
1.4 BASE COLUMN CONNECTION STIFFNESS.....	3
1.5 PROJECT SCOPE.....	4
1.6 OVERVIEW OF CHAPTERS.....	6
CHAPTER 2: LITERATURE REVIEW: THE PRINCIPLE OF VIRTUAL WORK	7
2.1 INTRODUCTION	7
2.2 DEVELOPMENT OF VIRTUAL WORK	8
2.2.1 <i>Another Look At Virtual Work (Truss Applications)</i>	11
2.2.2 <i>Frame Applications</i>	16
2.3 PRACTICAL USES FOR VIRTUAL WORK.....	20
2.3.1 <i>Calculating Displacement Participation (DISPAR) Factors</i>	20
2.3.2 <i>DISPAR Factors For Shell, Link, and Spring Elements</i>	21
2.3.3 <i>Calculating Sensitivity Indices (SIs)</i>	22
2.3.4 <i>Joint Components using Virtual Work</i>	25
2.3.5 <i>Frame Example Including Joint Components</i>	34
2.3.6 <i>Limiting Periods of Vibration using Virtual Work</i>	36
2.4 OPTIMIZATION OF STRUCTURAL STIFFNESS AND WEIGHT WITH VIRTUAL WORK.....	40
2.4.1 <i>Using DISPAR Factors and SIs</i>	40
2.4.2 <i>Automatic Resizing Technique</i>	43
2.5 CONCLUSION.....	48
CHAPTER 3: LITERATURE REVIEW – MODELING COLUMN BASE FIXITY AS PARTIALLY RIGID	49

3.1	INTRODUCTION	49
3.2	TYPES OF CONNECTIONS CONSIDERED	50
3.3	ANCHOR BOLT DETAILS	51
3.4	FACTORS INVOLVED IN THE ROTATION STIFFNESS OF THE CONNECTION.....	53
3.5	TESTING AND MODELING PROCEDURES FOR ROTATIONAL STIFFNESS	56
3.5.1	<i>Testing Procedure</i>	57
3.5.2	<i>Finite Element Model</i>	59
3.6	RESULTS – DETERMINING ROTATION STIFFNESS	61
3.6.1	<i>Analytical Procedure</i>	61
3.6.2	<i>Experimental Results</i>	63
3.6.3	<i>Finite Element Results</i>	66
3.6.4	<i>Effect of Axial load on Rotational Stiffness</i>	68
3.7	EFFECT ON BUCKLING STRENGTH.....	69
3.8	CONCLUSIONS.....	73
CHAPTER 4: DISPAR		74
4.1	DISPAR VERSION 5.4.....	74
4.1.1	<i>Joint Options</i>	75
4.1.2	<i>DPFs for Spring Elements</i>	79
4.1.3	<i>Method of Reanalysis</i>	80
4.1.4	<i>DISPAR for Optimization</i>	80
4.1.5	<i>Application of Virtual Loads</i>	81
4.2	DISPAR FOR SAP2000.....	82
4.2.1	<i>Overview of DISPAR for SAP2000</i>	83
4.2.2	<i>Shell Element DPF's</i>	83
4.2.3	<i>Spring and Link Elements</i>	85
4.2.4	<i>Use of SAP OAPI with VB.Net</i>	85
4.2.4.1	Steps Taken Using OAPI	86
4.2.4.2	Example of SAP2000 OAPI.....	86
4.2.5	<i>OpenGL in DISPAR</i>	89
4.3	LINKING DISPAR WITH EXCEL	90
CHAPTER 5: ANALYTICAL STUDY ON METAL BUILDINGS		91
5.1	METAL BUILDING USED FOR ANALYSIS	91
5.2	MODELING PROCEDURE.....	93
5.2.1	<i>Modeling the Frame</i>	94
5.2.2	<i>Applying the Loads</i>	94
5.2.3	<i>Development of Column Base Stiffness for Model</i>	95

5.2.3.1	Modeling Assumptions	96
5.2.3.2	Development of Model.....	100
5.2.3.3	Comparison with Literature.....	102
5.2.3.4	Stiffness of Typical Metal Building Connection	103
5.2.4	<i>Simplified Modeling Approaches</i>	105
5.2.4.1	Rotation Spring Approach.....	106
5.2.4.2	Springs At Bolt Locations	107
5.2.4.3	Pinned Support and Vertical Spring	108
5.3	BEHAVIOR OF CONNECTIONS WITH VARYING LOAD COMBINATIONS	110
5.3.1	<i>Frame Under Horizontal Load</i>	110
5.3.2	<i>Frame Under Varying Live Load</i>	111
5.4	SENSITIVITY ANALYSIS OF COLUMN BASE CONNECTION	114
5.4.1	<i>Sensitivity of Base Connections with Regard to Overall Drift</i>	115
5.4.2	<i>Base Connection Sensitivity vs. Column and Rafter Sensitivity</i>	118
5.5	CONCLUSION.....	121
CHAPTER 6: SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS		124
6.1	DISPAR AS AN ANALYSIS TOOL – PRACTICAL USES.....	124
6.2	COLUMN BASE CONNECTIONS	125
6.3	FINAL RECOMMENDATIONS	126
6.4	FUTURE RESEARCH.....	127
6.4.1	<i>DISPAR</i>	127
6.4.2	<i>Metal Building Frames</i>	128
REFERENCES		129
APPENDIX A: USER’S MANUAL FOR DISPAR FOR SAP2000		132
A1.	BEFORE GETTING STARTED.....	132
A.2	PROGRAM FEATURES	133
A.3	GETTING STARTED	134
A.3.1	<i>DISPAR Start-Up and File Open Screens</i>	134
A.3.2	<i>Load Case Selection Screen</i>	136
A.3.3	<i>In-Progress Screen</i>	137
A.3.4	<i>Results Screen</i>	138
A.4	EXAMPLES.....	140
A.4.1	<i>Example 1 – 3D Frame</i>	140
A.4.2	<i>Example 2 – Using Shell Elements</i>	149
APPENDIX B: CREATING DISPAR FOR SAP2000 – GUIDE TO GETTING STARTED IN OPENGL WITH VB.NET		152

B.1	INTRODUCTION.....	152
B.2	THE TAO FRAMEWORK	152
B.3	INITIALIZING A WINDOW	153
<i>B.3.1</i>	<i>Using the GLUT Library.....</i>	154
<i>B.3.2</i>	<i>Using VB.Net Window</i>	155
B3.2.1	Pixel Format descriptor.....	155
B.3.2.2	Windows Graphics Layer.....	156
<i>B.3.3</i>	<i>Using Multiple Windows.....</i>	157
B.4	WORKING WITH THE GLUT LIBRARY	158
B.5	DRAWING IN OPENGL.....	158
<i>B.5.1</i>	<i>Drawing Geometric Shapes.....</i>	161
<i>B.5.2</i>	<i>Drawing Text</i>	163
B.6	SETTING VIEW CONTROLS.....	165
<i>B.6.1</i>	<i>Two Dimensional Viewing</i>	165
<i>B.6.2</i>	<i>Three Dimensional Viewing.....</i>	166
B.7	BRIEF OVERVIEW OF ADDITIONAL OPENGL FEATURES	169
B.8	PUTTING IT ALL TOGETHER - EXAMPLE: DRAWING A SIMPLE 3D FRAME AND TEXT USING TWO VIEWING WINDOWS.....	169
<i>B.8.1</i>	<i>Creating the Form</i>	170
<i>B.8.2</i>	<i>Program Results</i>	172
<i>B.8.3</i>	<i>Source Code for Example</i>	173
APPENDIX C: WIND LOADS USED FOR ANALYSIS		178

LIST OF FIGURES

Figure 2-1: Example Structure.....	8
Figure 2- 2: Total Potential Energy Plot.....	9
Figure 2- 3: Truss Member Under Loading.....	11
Figure 2- 4: External and Internal Work Plots.....	12
Figure 2- 5: Truss Structure Under Loading.....	13
Figure 2- 6: Truss Structure Under Load, Q.....	14
Figure 2- 7: Force Deformation Relationship of System.....	15
Figure 2- 8: Stress Strain Relationship of System	15
Figure 2- 9: Displacement vs. SI ratio	25
Figure 2-10: Frame for Joint Region (Charney and Pathak 2007(a))	27
Figure 2-11: Forces in Beam-Column Joint Region (Charney and Pathak 2007(a)).....	30
Figure 2-12: Force Diagrams of Beam-Column Region (Charney and Pathak 2007(a))	31
Figure 2- 13: 3 Story 3 Bay Steel Frame	34
Figure 3- 1: Type of Column Base Connections (Picard and Beaulieu (1985)).....	51
Figure 3- 2: Anchor Bolt Detail (Hon and Melchers (1988))	52
Figure 3- 3: Anchor Bolt Detail (Picard and Beaulieu (1985))	53
Figure 3- 4: Testing Apparatus (Hon and Melchers (1988))	54
Figure 3- 5: Comparison of Moment Rotation Curves (Hon and Melchers 1988).....	56
Figure 3- 6: Test Configuration (Picard and Beaulieu 1985)	58
Figure 3- 7: Finite Element Model (Hamizi and Hannachi 2007)	60
Figure 3- 8: Column-Base Connection Diagram	62
Figure 3- 9: Moment Rotation Plots (Picard and Beaulieu 1985)	65
Figure 3- 10: Moment Rotation Plots (Hamizi and Hannachi 2007).....	67
Figure 3- 11: Moment Rotation of 4 Bolted Connection with Axial Load (Hamizi and Hannachi 2007)	69
Figure 3- 12: Buckling Strength of Frames	70
Figure 3- 13: Effective Length in Columns of a Single – Story Single – Bay Frame (Galambos 1960).....	71

Figure 4- 1: Column Joint Options for Shear (Charney 1995)	76
Figure 4- 2: Column Joint Options for Flexure (Charney 1995)	77
Figure 4- 3: Beam Joint Options for Shear (Charney 1995).....	78
Figure 4- 4: Beam Joint Options for Flexure (Charney 1995).....	79
Figure 4- 5: Application of Virtual Loads	82
Figure 4- 6: SAP Model.....	87
Figure 4- 7: Model with Point Loads.....	88
Figure 4- 8: Tables Available in DISPAR	90
Figure 5- 1: Column Dimensions.....	91
Figure 5- 2: Rafter Dimensions.....	92
Figure 5- 3: Overall Frame Dimensions	92
Figure 5- 4: Building Dimensions.....	93
Figure 5- 5: Frame Finite Element Model	94
Figure 5- 6: Frame Under Wind Loading	95
Figure 5- 7: Plate Bending Behavior	96
Figure 5- 8: Anchor Bolt Stress Distribution.....	97
Figure 5- 9: Nonlinear Spring Force Deformation Relationships.....	99
Figure 5- 10: Finite Element Model of Type 2 Connection.....	101
Figure 5- 11: Spring Grid Layout for 4 Bolt Connection	102
Figure 5- 12: Metal Building Column Base Connection	104
Figure 5- 13: Finite Element Model of Metal Building Base Connection.....	105
Figure 5- 14: Pinned Connection with Rotational Spring (Type 2).....	107
Figure 5- 15: Vertical spring base connection	108
Figure 5- 16: Location of Compression Force for Connection Type 2.....	109
Figure 5- 17: Support Conditions of Plate	109
Figure 5- 18: Frame Under Gravity Live Load.....	112
Figure 5- 19: Drift vs. Percentage of Live Load.....	113
Figure 5- 20: Drift vs. Spring Stiffness.....	116

Figure 5- 21: Derivative of Drift vs. Stiffness Plot.....	117
Figure 5- 22: Spring Contribution vs. Rotational Stiffness	118
Figure 5- 23: Member Labels for SAP Groups.....	119
Figure 5- 24: Member SI vs. Spring Stiffness	120
Figure A. 1: DISPAR Start-Up Screen	134
Figure A. 2: File Open Screen	135
Figure A. 3: Load Case Selection Screen	136
Figure A. 4: DISPAR In-Progress Screen.....	137
Figure A. 5: DISPAR Results Screen	138
Figure A. 6: Typical Floor Plan	141
Figure A. 7: Load Cases Window.....	142
Figure A. 8: Menu Strip	142
Figure A. 9: Display Menu Options.....	143
Figure A. 10: DISPAR Viewing Screen with Display Options	143
Figure A. 11: SI Viewing.....	144
Figure A. 12: 2D Viewing Selection.....	145
Figure A. 13: Edit Group Sections.....	147
Figure A. 14: Model with Shell Elements	150
Figure B. 1: Reference Files in Program	153
Figure B. 2: Files Located in Release Folder.....	153
Figure B. 3: Color Diagram	161
Figure B. 4: Orthographic Projection.....	166
Figure B. 5: Coordinate System.....	167
Figure B. 6: Perspective View	167
Figure B. 7: Rotation Figure	168
Figure B. 8: Form Design for Example	171
Figure B. 9: Completed Example Program Form	172

Figure C. 1: Design Wind Loads for Enclosed Buildings – Method 1 (ASCE 7-05 Fig. 6-2) ... 180
Figure C. 2: Wind Pressures on Frame 182

LIST OF TABLES

Table 2- 1: Equations for Internal Energy in Structural Members (Charney 1995)	19
Table 2-2: Truss DISPAR Factors	21
Table 2- 3: Member DISPAR Factors for Joint Region	35
Table 2- 4: Updated DISPAR Factors with Equal SIs	41
Table 4- 1: ETABS NRGD Variable	75
Table 4- 2: Frame Input	87
Table 4- 3: Displacements from Model	89
Table 5- 1: Initial Connection Stiffness Comparison	103
Table 5- 2: Connection Comparison with Horizontal Load.....	111
Table 5- 3: Drift at the Left Corner of the Frame Based on Percentage of Live Load	113
Table A. 1: Member DISPAR Percentages.....	146
Table A. 2: Member DISPAR Totals.....	146
Table A. 3: Group DISPAR Totals	147
Table A. 4: Shell Member DISPAR Totals	150
Table A. 5: Shell Member Group Totals	151
Table B. 1: Pixel Format Controls	156
Table B. 2: Drawing Commands	163
Table C. 1: Design Wind Loads for Enclosed Buildings – Method 1 (ASCE 7-05 Fig. 6-2)	181

CHAPTER 1: INTRODUCTION

In today's world, engineers are always looking for ways to optimize their designs by having the most efficient system, and structural engineering is no exception to this idea. One of the major tasks faced by structural engineers is to design a building which has enough stiffness to control the drift the building experiences under wind loading. There are many different systems structural engineers use to control the drift in a building; some of which are moment frames, braced frames, and concrete shear walls. Through experience, designers gain an insight into how each of these systems behave and which is the best to use under different conditions. What if designers had a tool that would tell them exactly how each of these systems behave, not only by themselves, but also working together? This would give them a more accurate insight into the systems as well as how to efficiently and economically design them. In the early 1990's a program was developed by Dr. Finley Charney, known as DISPAR (Charney 1995), which was a tool for doing just that. The name DISPAR stands for DISplacement PARTicipation factor, and is based on utilizing the principal of virtual work to calculate displacement participation factors for each member. Each member's DISPAR factor is that member's contribution to the overall drift of a structure. The original code and graphics for DISPAR were written in FORTRAN and worked only under the DOS operating system. This method of writing software is primitive by today's standards, leading to the need for an updated version of DISPAR incorporating object oriented programming and modern graphics utilities.

1.1 MOTIVATION FOR RESEARCH

The original version of DISPAR was written to work with ETABS version 4 and SAP90 version 5 (Computers and Structures 1995), using output files from the software to run the calculations. This method of transferring data between the two programs was effective, but had limitations on how the data had to be formatted. Not only were there restrictions on data formatting, there were limitations on what could be exported back into SAP90 or ETABS. Starting with version 11 of SAP2000 (Computers and Structures 2008b), a license was included for an Open Application Programming Interface (OAPI) which allows users access to information in SAP2000 using programming (Computers and Structures 2008a). The OAPI is a library of functions giving the

user the ability to perform virtually any task in SAP2000 that is available using the graphical user interface. These functions range from opening SAP2000 to building a model to retrieving forces and displacements in that frame under loading.

With the development of SAP2000's OAPI came the idea to update DISPAR, having it link directly with SAP2000 allowing for more efficient access to model information. Not only would this allow for quick access to frame information, it also opened the door for the possibility of real time changes made to a model with DISPAR. In addition to SAP2000's OAPI, there was motivation to investigate a graphics package for DISPAR instead of using the graphics primitives developed in FORTRAN. The graphics package under investigation was OpenGL, which has been used for graphics in many computer and video games. With the use of a modern graphics package, more advanced viewing capabilities will be available in DISPAR.

In the later stages of development for the new version of DISPAR (DISPAR for SAP2000), a practical application presented itself for using the program. The project involved obtaining a better understanding of the behavior of metal buildings under wind loading. The focus of the research in this thesis centered on investigating the contribution of the column base connection if its stiffness is explicitly modeled. With the use of DISPAR for SAP2000, the contribution from the column base connection was isolated. Once its contribution was isolated an analytical study could be conducted on the connection's stiffness with regard to overall drift as well as other members in the frame. The metal building investigation provided a way to show how powerful a tool that DISPAR really is and many of its capabilities.

1.2 VIRTUAL WORK

The principle of virtual work is the cornerstone of DISPAR. Virtual work is based on the energy in a system, with the idea that in equilibrium that there is no change in energy, or the total potential energy is at a minimum. If there is no change in total energy, the external energy put into the system equals the energy in the system. Each member in the system has its own energy,

this energy can be isolated and from there that member's contribution to displacement can be determined. Once the member's contributions have been calculated, these can be used to understand structural behavior. This behavior can be examined from a macro scale where the structure as a whole is examined or a micro scale where individual components such as frame elements can be explored. With this insight into the behavior of the structure, the information can be used for optimization. This optimization can be in relation to a number of parameters. One of these parameters is to optimize the stiffness of the structure under static loads, while limiting its overall weight. Another form of optimization is to limit the natural period of vibration of the structure, which affects its behavior under dynamic loads, while optimizing the placement and weight of material. Given the versatility of calculating member contributions in this manner, it is easy to see why virtual work was used in DISPAR. Formal optimization techniques have not been explicitly included in DISPAR. It is left up to the designer to use the information provided by DISPAR to perform their own optimization.

1.3 DISPAR

As mentioned above, DISPAR is a program which implements the principle of virtual work to calculate displacement participation factors for structural members. DISPAR allows the user to view structural behavior in two different ways, in a graphical and tabular manner. The graphics viewing gives the user the option to select different frames lines, view member contributions based on deformational component (axial, flexural shear), member contributions by color, and member groups. The tabular form provides a list of tables where the user can view all of the member and group contributions. With this information the user has all of the tools necessary to understand and optimize the performance of the structure.

1.4 BASE COLUMN CONNECTION STIFFNESS

Metal buildings are designed to provide large open spaces inside of the building without the use of interior columns. The frames typically consist of built-up members from structural steel plates, where the columns are usually tapered. The rafters may be a combination of tapered and

prismatic members. When analyzing the behavior of metal building frames, designers typically assume the column base connection as pinned. This assumption that the column base connection is pinned is conservative with respect to calculating drift. Therefore, the efficiency of the frame can be improved by explicitly modeling the actual behavior of the column base connection. Before modeling the connection, a study must be conducted to determine if it has sufficient stiffness to include in analysis. First, a literature review will be conducted on previous work that has been completed on column base connection stiffness. The previous work studied will include finite element models and lab testing on various connection types to predict their behavior. Using this information, a model will be developed to predict the rotational stiffness of a typical connection used in metal building frames. With this connection model, a sensitivity analysis will be conducted to determine how explicitly modeling this connection affects the overall drift of the frame. After doing so, a conclusion will be reached on whether there is a significant benefit in explicitly modeling this connection. If it is concluded that there is a significant benefit, recommendations will be made on the approach to modeling the connection.

1.5 PROJECT SCOPE

As with the DISPAR program, the principle of virtual work is the foundation of the research conducted in this thesis. The optimization discussed throughout, is based on optimizing a structure for stiffness, while minimizing parameters such as volume of material or period of vibration. This is not necessarily saying that after using the DISPAR program a designer will have the most efficient structure in every regard, such as cost. There are many other factors that go into the overall cost of a structure, such as construction costs, fabrication costs, detailing and so on. It is left up to the designer to use engineering judgment when using a program such as DISPAR, which is meant to be a tool for an engineer, not to replace an engineer. When using DISPAR, like any program, it is also the user's responsibility to verify the results of the program to make sure they understand where they are coming from and their meaning.

Two of the main focuses in updating DISPAR were to investigate the use of SAP2000's OAPI functions and implementing OpenGL. There are areas in the updated version of DISPAR which

can be improved upon and features that have not been included to date which were in the original version, such as handling concrete cracking and composite action in steel structures. Based on the research conducted on SAP2000's OAPI functions and OpenGL, the modeling and viewing capabilities of DISPAR are only limited by how far the developer wants to take it.

With regard to the metal building frames investigated for this thesis, the primary focus was the column base connection. The frames were modeled using finite elements and the wind loads applied were applied based on ASCE 7-05 (ASCE 2005). However, this is not a study on the behavior of the frame as a whole or the proper loads to apply for design. In addition, this study only applies to drift limit states, the effect on strength limit states was not considered. A later study will be carried out by another graduate student on the appropriate wind loading for design and the effects of diaphragm and connector action. The model developed to represent the behavior of the column base connection was based on a typical connection provided by American Buildings Company (Walsh, 2009) and is not meant to represent the connection used in all metal buildings. The model was verified after comparing the results with multiple lab tests and other finite models developed by researchers (Hon and Melchers 1988; Picard and Beaulieu 1985; Hamizi and Hannachi 2007). This research was conducted to further understand the behavior of the connections and the benefit in design by explicitly modeling their behavior. When taking into account the behavior of a base connection that does not conform to any of the details under consideration in this thesis, the designer should understand how their particular connection behaves before counting on its stiffness.

In conclusion, the goal of this research was to give practicing engineers more tools to assist them in the design process. These tools were not meant to make the decisions for the designer, such as optimizing the structure and recommending sections. It was meant to allow for the "human loop" where the designer is given information into the behavior of the structure and uses engineering judgment and experience to make a decision.

1.6 OVERVIEW OF CHAPTERS

The second chapter in this thesis focuses on a literature review covering the principle of virtual work and many of its applications. Next, Chapter 3 consists of a literature review on the testing and modeling done on various column base connections types to determine a rotational stiffness for each connection type. The fourth chapter discusses both the previous version of DISPAR and the new version of DISPAR for SAP2000 and the features of each version. The next chapter, Chapter 5, is a study on how explicitly modeling the column base connection in a metal building frame will affect the overall drift. The results obtained in Chapter 5 were used to determine an effective way to model the stiffness in the column base connection as well as techniques for optimizing the stiffness of the connection. Conclusions and recommendations were made based on the results in Chapter 5, and discussed in Chapter 6 of the thesis. Appendix A consists of a guide to getting started with DISPAR for SAP2000 and includes examples to help with this. The information in Appendix B is a discussion on the basics of OpenGL and how to begin drawing basic shapes. This chapter also includes an example program with the source code to assist the user with getting started. Lastly, Appendix C shows how the wind loads were determined for the metal building frame analysis in Chapter 5.

CHAPTER 2: LITERATURE REVIEW: THE PRINCIPLE OF VIRTUAL WORK

2.1 INTRODUCTION

The principle of virtual work is the basis for a method used for calculating displacements in a structure when it is loaded under real loading conditions. To calculate the displacement of the structure under real loading conditions, the structure is first loaded with a fictitious force at the location of and in the direction of the real displacement of concern. This force is fictitious because it does not actually exist in the structure's real loading conditions and only serves the purpose of providing virtual energy in the system. This chapter will show how the virtual energy provided by the fictitious or virtual load may be used to retrieve the displacement of the structure under the real loading conditions for both truss and frame structures.

The virtual work method relies on the internal and external energy in a system so it is very versatile in its applications. Not only can the total structural displacement be calculated, but each member's contribution to the total displacement may be determined. The member's contribution can also be broken down into axial, flexural, shear and torsional components. These member contributions can be used to understand the behavior of a structure and optimizing the placement of material for stiffness. Virtual work may also be extended to account for material nonlinearities, loads applied with accelerations or even structural damping, simply by accounting for the energy. Tabulating virtual work quantities for material nonlinearities and dynamic loads is beyond the scope of this thesis, but can be done if the energies associated are included in the calculations.

2.2 DEVELOPMENT OF VIRTUAL WORK

To understand the development of the principle of virtual work, the structure in Fig. 2-1 will be used for reference. In Fig. 2-1(a), the structure experiences a real load of 20 kips applied at each story, resulting in a displacement of d_x at the roof level. A virtual load Q of 1.0 kips is applied at the same location and in the direction of displacement d_x in Fig. 2-1(b). When a structure is loaded with a set of real loads it begins to deflect. Each member in the structure experiences deformations, storing strain energy in the system. If it is assumed that the real loading is applied slowly such that all of the external energy is converted in to strain energy, the external energy is equal to the strain energy. To calculate the displacement, the work done by the stresses moving through internal strains in each member can be compared to the external work done on the system. A work quantity is defined as a force multiplied by a displacement in the direction of the force.

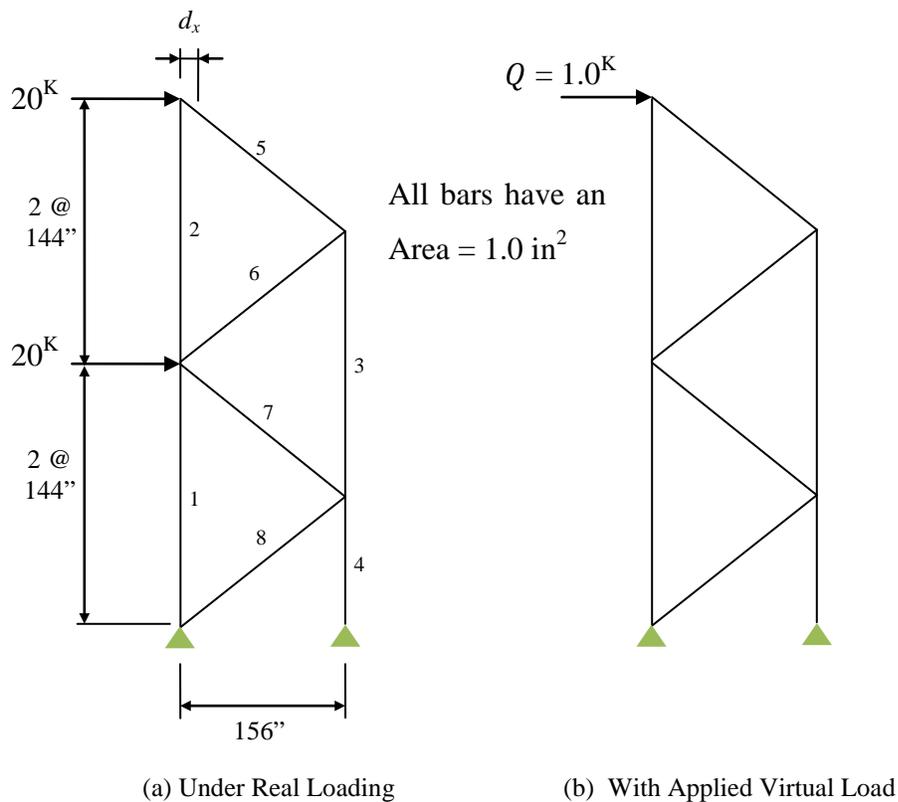


Figure 2-1: Example Structure

To understand how the virtual work method produces accurate displacements the theorem of Minimum Potential Energy must first be explored. The total potential energy in the system is equal to the internal strain energy minus the external work done on the system:

$$\Pi = U - W \tag{2.1}$$

where Π is equal to the total potential energy in the system, U is equal to the strain energy, and W is equal to the work done on the system. According to the theorem of Minimum Potential Energy the system is at equilibrium when the variation in potential energy due to a virtual force is equal to zero.

$$\delta\Pi = \delta U - \delta W = 0 \tag{2.2}$$

where $\delta\Pi$, δU , and δW equal the variation in total potential energy, strain energy, and work respectively. The meaning of Eq. (2.2) is described graphically in Fig. 2-2, where a small “virtual” change in the deformed configuration of the system is shown not to change Π .

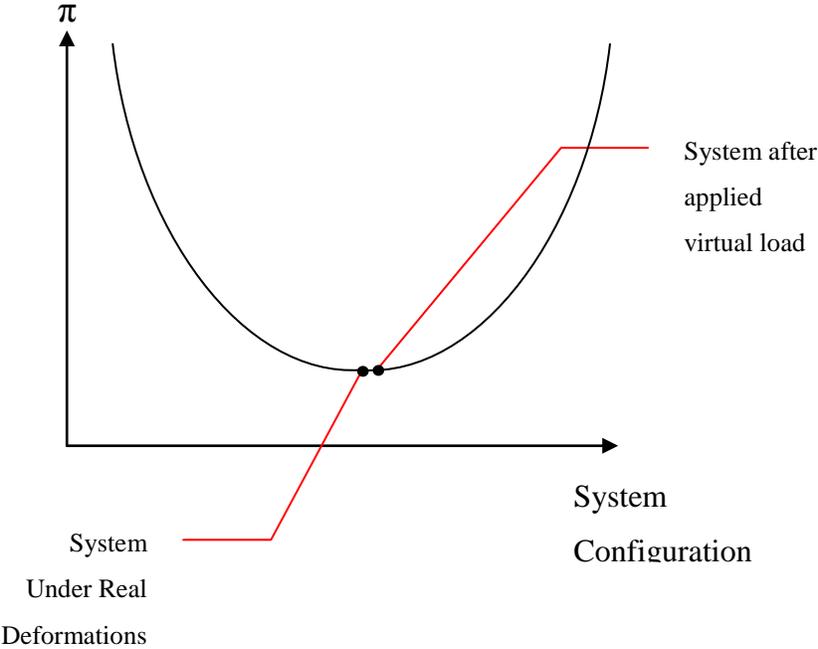


Figure 2- 2: Total Potential Energy Plot

When looking at the system under a real displacement, each member experiences individual strains. If this system is then loaded with a virtual load, each member in the system experiences a stress due to this load. Because the new system including the virtual load is still in equilibrium, the point remains at the bottom on the total potential energy plot. The change in energy of both the systems (under real displacements and with the applied virtual load) may be related to each other in order to determine the real displacement. The change in strain energy and the change in the external work of the systems may be written as follows:

$$\delta U = \int_v \sigma_{\bar{v}} \cdot \varepsilon_r dV \quad (2.3)$$

$$\delta W = Q d_x \quad (2.4)$$

where $\sigma_{\bar{v}}$ is the stress in a given member due to the virtual load, ε_r is the strain in the member due to the real displacements, Q is the virtual force applied to the structure at the point of and in the direction of the displacement of concern, and d_x is the displacement of concern, see Fig. 2-1(b). Because the change in potential energy must equal zero, the change in strain energy and external work must be equal.

$$\delta W = \delta U \quad (2.5)$$

This may also be described as the principle of conservation of energy where the external work must equal the internal work. After substituting in Eq. (2.3) and Eq. (2.4), Eq. (2.5) becomes:

$$Q d_x = \int_v \sigma_{\bar{v}} \cdot \varepsilon_r dV \quad (2.6)$$

If the structure remains elastic the strain energy in the system is sufficient to capture the internal energy in the system, otherwise some energy will be lost in the form of heat due to the hysteretic behavior of the material. In addition, if accelerated loads are applied some of the external energy will take the form of kinetic energy.

2.2.1 ANOTHER LOOK AT VIRTUAL WORK (TRUSS APPLICATIONS)

This section will examine the principle of virtual work from a work energy balance approach, as applied to truss members. This method has a different approach than that of the Minimum Potential energy derivation, but has the same end result. The derivation for this approach was taken from the class notes of a structural analysis class taught at Virginia Tech by Charney (2009). When work and energy equations are used to calculate displacements in a truss member, it is simply a matter of setting the external work equal to the internal work, or:

$$W_E = W_I \quad (2.7)$$

For the truss member show in Fig. 2-3, the member has an original length L , and is loaded with a force of P . Under the load of P , the truss member deforms a distance d .

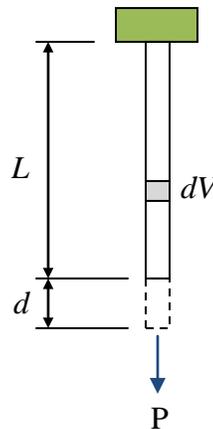


Figure 2- 3: Truss Member Under Loading

As the truss member begins to deform under the load P , the member has a force deformation relationship shown in Fig. 2-4(a). Also, as the member begins to elongate, the stress strain relationship for the differential volume, dV , is shown in Fig. 2-4(b).

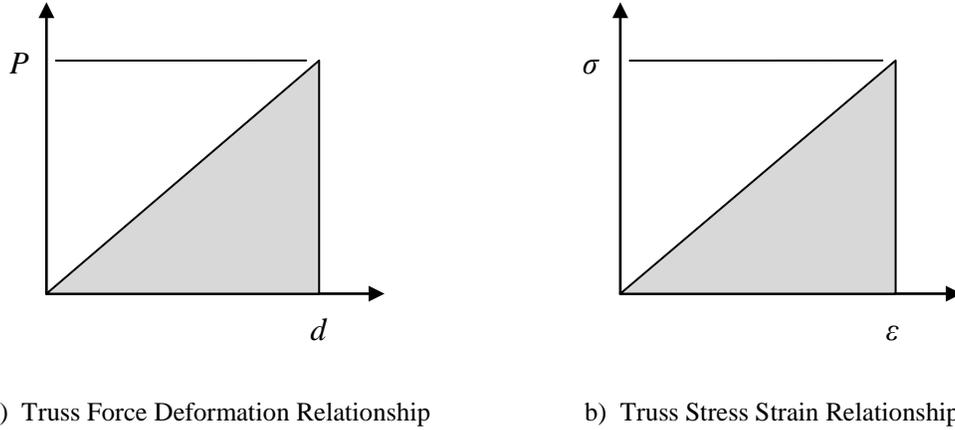


Figure 2- 4: External and Internal Work Plots

The shaded area under the curve in Fig. 2-4(a) is equal to the “External Work” done on the system. This may be written as follows:

$$W_E = \frac{1}{2} P d \quad (2.8)$$

The shaded under the curve in Fig. 2-4(b) is equal to the “Internal Work” done on the differential volume. The internal work on the entire member may be calculated as follows:

$$W_I = \frac{1}{2} \int_v \sigma \cdot \varepsilon dv \quad (2.9)$$

To determine the internal virtual work calculate σ and ε from Eq. (2.3).

$$\sigma = \frac{P}{A} \quad (2.10)$$

$$\varepsilon = \frac{P}{AE} \quad (2.11)$$

where A is the cross-section area of the member, and E is equal to the modulus of elasticity of the member. For a prismatic member the integral over the volume may be written as follows:

$$\int_v dV = \int_0^L A \cdot dx = AL \quad (2.12)$$

Therefore, the equation for internal work becomes:

$$W_I = \frac{1}{2} \cdot \frac{P}{A} \cdot \frac{P}{AE} \cdot AL = \frac{1}{2} \frac{P^2 L}{AE} \quad (2.13)$$

And remembering that the external work is equal to the internal work:

$$\frac{1}{2} P d = \frac{1}{2} \frac{P^2 L}{AE} \quad (2.14)$$

Then solving for the displacement, d :

$$d = \frac{PL}{AE} \quad (2.15)$$

which is only applicable for a truss member. For truss structures, there are multiple elements in the system, therefore the internal work for each member must be accounted for. It is simply a matter of summing the internal strain energies for all of the i members in the system. Refer to Fig. 2-5 for a truss structure and its deflected shape under a load.

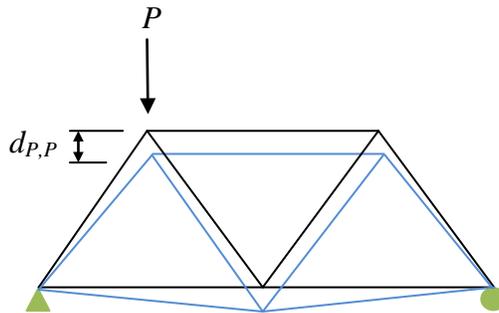


Figure 2- 5: Truss Structure Under Loading

The structure in Fig. 2-5 is loaded with a concentrated force P and has a deflection of $d_{P,P}$ at the location of P , due to the load P . Similar to Eq. (2.8), the external work for the truss structure is equal:

$$W_E = \frac{1}{2} P d_{P,P} \quad (2.16)$$

The internal virtual work for each member, i , is as follows:

$$w_i = \frac{1}{2} \frac{P_i^2 L_i}{A_i E_i} \quad (2.17)$$

Thus the internal work for the entire system is:

$$W_I = \frac{1}{2} \sum_i^n \frac{P_i^2 L_i}{A_i E_i} \quad (2.18)$$

Recalling that external work equals internal work and solving for the displacement, $d_{P,P}$ is equal to:

$$d_{P,P} = \frac{1}{P} \sum_i^n \frac{P_i^2 L_i}{A_i E_i} \quad (2.19)$$

To determine the displacement at a point in the structure other than at the location of P , a virtual load of Q must be applied, see Fig. 2-6. In the absence of Q , an equation for the displacement at the location of Q , or any location other than P , cannot be written.

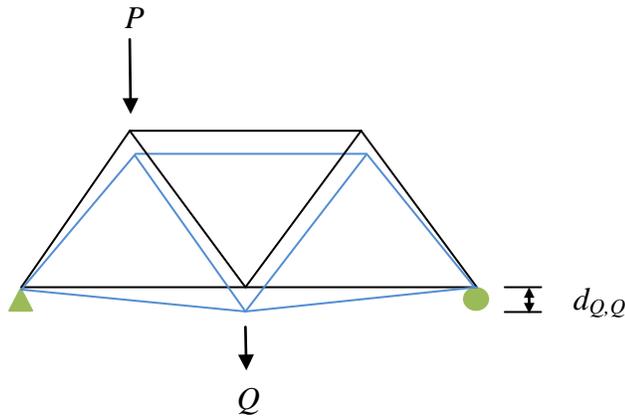


Figure 2- 6: Truss Structure Under Load, Q

To calculate the displacement at the location of Q , begin by first loading the structure with Q and then later add P . The diagram in Fig. 2-7 shows the force deformation relationship of the truss structure as it is loaded with the virtual load Q and then with the load P .

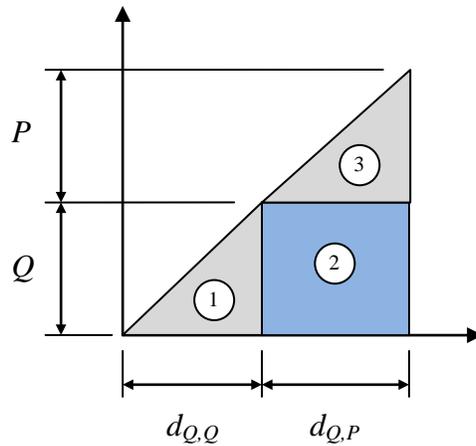


Figure 2- 7: Force Deformation Relationship of System

The shaded area found in region “1” is the external work performed on the truss due to the virtual load Q , where the structure displaces a distance of $d_{Q,Q}$ (displacement at Q due to the load Q). As a load of P is applied to the truss, the truss begins to displace an additional amount of $d_{Q,P}$ (displacement at Q due to P). The area found in region “2” is the external work on the system done by the virtual load, Q , moving through the displacement $d_{Q,P}$. Lastly, the area found in region “3” is the external work on the truss done by the load P moving through the displacement $d_{Q,P}$. The internal work may be calculated in a similar fashion by using the stress strain relationship for member i , found in Fig. 2-8.

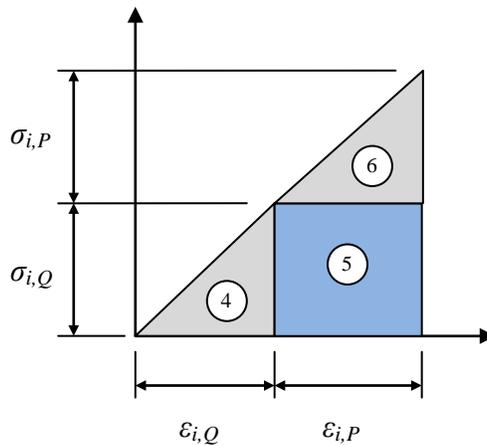


Figure 2- 8: Stress Strain Relationship of System

From conservation of energy it can be stated that region “1” and region “4” are equal, region “2” and region “5” are equal, and region “3” and region “6” are equal. Since the goal is to calculate

the displacement at the location of Q due to P , region “2” and region “5” will be used. The external work in region “2” is equal to:

$$W_E = Qd_{Q,P} \quad (2.20)$$

and the internal work in region “5” for member i is equal to:

$$W_{I,i} = \frac{P_i L_i}{A_i E_i} Q_i \quad (2.21)$$

where P_i is the force in member i due to the load P and Q_i is the force in the member due to load Q . Notice the absence of the 1/2 term in both the external and internal work equations. Now setting the external work equal to the summation of the internal work for all of the members in the truss and solving for $d_{Q,P}$:

$$d_{Q,P} = \sum_i^n \frac{P_i L_i}{A_i E_i} \left(\frac{Q_i}{Q} \right) \quad (2.22)$$

where the term Q_i/Q is referred to as a weighting factor and the rest of the right side of the equal is the bar deformation (Charney 2009). So the displacement $d_{Q,P}$ is equal to the weighted sum of the individual bar deformations. As mentioned previously, this approach to the derivation of virtual work produces the same end result as the Minimum Potential energy approach.

2.2.2 FRAME APPLICATIONS

A similar approach can be taken when using virtual work for frames. When dealing with beams or columns, all of the components of deformations can be taken into account, which include axial, bending, shear and torsion. The torsional component of deformation is typically small, but should be included for completeness. This section will cover the derivation of the work component due to bending and includes the equations for all components, found in Table 2-1. From Eq. (2.6) the virtual stress, $\sigma_{\bar{v}}$, and the real strain, ϵ_r , for frame members must be derived. The stress in a member due to the virtual load is equal to:

$$\sigma_{\bar{v}} = \frac{my}{I} \quad (2.23)$$

where m is equal to the moment in the beam due to the virtual load, y is the distance from the centroid of the cross-section to the extreme fiber and I is the moment of inertia of the section about the bending axis. The strain in a member due to the real load is equal to:

$$\varepsilon_r = \frac{My}{EI} \quad (2.24)$$

where M is equal to the moment in the beam due to the real loading. Then substituting Eq. (2.23) and Eq. (2.24) into Eq. (2.3):

$$W_i = \int_0^L \int_A \frac{my}{I} \cdot \frac{My}{EI} dA \cdot dx \quad (2.25)$$

From basic beam mechanics it can be said that:

$$I = \int_A y^2 dA \quad (2.26)$$

Therefore Eq. (2.25) simplifies to:

$$W_i = \int_0^L \frac{Mm}{EI} dx \quad (2.27)$$

Finally writing the equation in terms of conservation of energy:

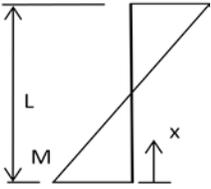
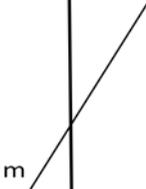
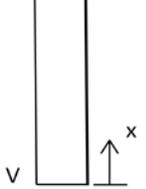
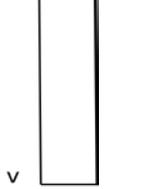
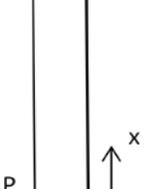
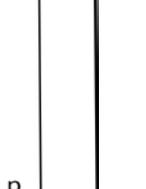
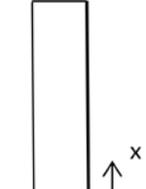
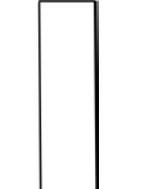
$$Qd_x = \int_0^L \frac{Mm}{EI} dx \quad (2.28)$$

Table 2-1 includes the internal work calculations for each component of deformation experienced by a beam or column. To accurately calculate the displacement of a structure all of the components must be included. Including all of the components of deformation for all members in a structure Eq. (2.28) becomes:

$$d_x = \frac{1}{Q} \sum_{i=1}^n \int_0^L \left[\frac{M(x)_i m(x)_i}{EI_i} + \frac{P(x)_i p(x)_i}{A_i E} + \frac{V(x)_i v_i}{A_{v,i} E} + \frac{T(x)_i t(x)_i}{GJ_i} \right] dx \quad (2.29)$$

where V and v are the member's real and virtual shear forces, T and t are the member's real and virtual torsional forces. With the member having a shear area of $A_{v,i}$, shear modulus of G , and a torsional constant J_i . This equation may be summed for the total number of members, n , in a structure. There are also minor axis components of flexure and shear (not shown in Eq. (2.29)) which must be accounted for. These components are very similar to the major axis components, except they are calculated using the minor axis forces and properties.

Table 2- 1: Equations for Internal Energy in Structural Members (Charney 1995)

Real Force	Virtual Force	DPF
		$d_{x,F} = \int_0^L \frac{M(x)m(x)}{EI_{maj}} dx$
		$d_{x,V} = \int_0^L \frac{V(x)v(x)}{EA} dx$
		$d_{x,A} = \int_0^L \frac{P(x)p(x)}{EA} dx$
		$d_{x,T} = \int_0^L \frac{T(x)t(x)}{GJ} dx$

2.3 PRACTICAL USES FOR VIRTUAL WORK

There are many practical uses for the principle of virtual work to assist a designer in indentifying the behavior a structure as a whole, as well as the individual components that make up structure. The methods using virtual work that will be discussed in this section include member displacement participation (DISPAR) factors, member sensitivity indices, beam – column joint components, and limiting periods of vibration. The ability to calculate member DISPAR factors is not only limited to truss and frame elements, they may also be calculated for shell, link, and spring elements. A brief discussion on the calculation of these elements is also included in this section. Refer to Sections 4.2.2 and 4.2.3 for a more in depth look at the calculation of DISPAR factors for shell, link, and spring elements.

2.3.1 CALCULATING DISPLACEMENT PARTICIPATION (DISPAR) FACTORS

A member's DISPAR factor is that particular member's contribution to the overall drift of the structure in the direction of the virtual load, Q , and has units of length. This factor is derived from the internal virtual work contribution of a member, i.e. the virtual strain energy of a member. Recall from Eq. (2.17) that the virtual strain energy for truss member, i , is equal to:

$$w_i = p_i \frac{P_i L_i}{A_i E}$$

In order to extend the DISPAR factor to a frame element, it must include the contributions of all components of deformation, refer to Eq. (2.29). Therefore, the DISPAR factor for a frame element, not including minor axis forces, is equal to:

$$DPF_i = \frac{1}{Q} \int_0^L \left[\frac{M(x)_i m(x)_i}{EI_i} + \frac{P(x)_i p(x)_i}{A_i E} + \frac{V(x)_i v_i}{A_{v,i} E} + \frac{T(x)_i t(x)_i}{GJ_i} \right] dx \quad (2.30)$$

This equation applies for both frame elements and truss elements. In the event of a truss element, the flexure, shear and torsional forces will be equal to zero and the equation simplifies to Eq. (2.17).

To illustrate this application, the DPF's were calculated for the structure found in Fig. 2-1 where only axial deformations were included (i.e. truss elements). The results of this example may be found in Table 2-2, where the total displacement is equal to 6.53 in. The same method may be extended for a frame structure utilizing the flexural, shear and torsional components found in Eq. (2.30).

Table 2-2: Truss DISPAR Factors

Bar Number	Length (in)	Area (in ²)	Volume (in ³)	Real Force (kips)	Virtual Force (kips)	DPF (in.)	SI*1000 (in. ⁻²)
1	288	1.0	288.0	73.8	2.8	2.03	7.1
2	288	1.0	288.0	18.5	0.9	0.17	0.6
3	288	1.0	288.0	-36.9	-1.9	0.68	2.3
4	144	1.0	144.0	-110.8	-3.7	2.03	14.1
5	212	1.0	212.3	-27.2	-1.4	0.27	1.3
6	212	1.0	212.3	27.2	1.4	0.27	1.3
7	212	1.0	212.3	-54.4	-1.4	0.54	2.5
8	212	1.0	212.3	54.4	1.4	0.54	2.5
Summation			1857.3			6.53	

Notice that the DISPAR factor for bar no. 1 is equal to 2.03 in. This means that bar no. 1 accounts for 2.03 in. of the total drift (6.53 in.) in the direction of and at the location of the virtual load, Q. In addition, the summation of all the DISPAR factors for each bar is equal to the displacement returned (6.53 in.) when using a structural analysis program.

2.3.2 DISPAR FACTORS FOR SHELL, LINK, AND SPRING ELEMENTS

The ability to calculate DISPAR factors for other types of elements can be very useful. Many times, elements outside of a standard frame or truss element are used to model structural systems. An example of this might be the use of shell elements to represent a shear wall, or spring elements to model the behavior of a non-rigid connection. To calculate the DISPAR factors for these elements, the same idea applies as with the frame and truss elements. An internal virtual work quantity must be calculated based on the work done by a virtual force on the element times a real displacement. Dealing with shell, link, and spring elements is carried out

on a nodal basis. There are six components of deformation at each node, so a virtual work quantity for each of these components must be taken into account. For a given component of deformation, multiplying the virtual force experienced at that node by the real displacement results in a virtual work quantity. To calculate the total virtual work quantity of one of these elements, all six of the components at each node must be added together, as well as all of the nodes included in the element. This procedure may also be extended for other types of element, as long as nodal forces and displacements may be retrieved for the analysis program. As mentioned before, a more detailed discussion on the calculation of the DISPAR factor for these elements may be found in Sections 4.2.2 and 4.2.3.

2.3.3 CALCULATING SENSITIVITY INDICES (SIs)

If a structure does not meet the drift limitations set, then the structure must be stiffened to reduce the drift. Many times this leads to increasing the volume of material in one or more members to meet the drift limit. By using the member's DPF a designer can gain insight into how much each member contributes to the overall drift. However the member with highest DPF is not always the member to which material should be added. The values in this section are based on the calculations found in Table 2-2. In this example, both members 1 and 4 have a DPF of 2.03 in., so how do you decide which member's volume to increase? This section will discuss an effective way to determine which members are more heavily influenced by increasing or decreasing their size. In other words, adding a given volume of material to some members will decrease the drift more than adding the same volume of material to others. Because the truss is statically determinate, changing the volume in members does not change the forces experience by the members. Notice the effect of adding 20 cubic in. to members 1 and 4.

$$A_{1,new} = \frac{V_{old} + \Delta V}{L} = \frac{288 + 20}{288} = 1.069 \text{ in}^2 \quad (2.31)$$

$$A_{4,new} = \frac{V_{old} + \Delta V}{L} = \frac{144 + 20}{144} = 1.14 \text{ in}^2$$

Now the updated DPFs:

$$DPF_1 = DPF_{1,old} \left(\frac{A_{1,old}}{A_{1,new}} \right) = 2.03 \left(\frac{1.0}{1.069} \right) = 1.90 \text{ in} \quad (2.32)$$

$$DPF_4 = DPF_{4,old} \left(\frac{A_{4,old}}{A_{4,new}} \right) = 2.03 \left(\frac{1.0}{1.14} \right) = 1.78 \text{ in}$$

The change in DPFs:

$$\Delta DPF_1 = DPF_{1,old} - DPF_{1,new} = 2.03 - 1.9 = 0.13 \text{ in} \quad (2.33)$$

$$\Delta DPF_4 = DPF_{4,old} - DPF_{4,new} = 2.03 - 1.78 = 0.25 \text{ in}$$

Finally, the change in DPF per change in volume:

$$\frac{\Delta DPF_1}{\Delta V} = \frac{0.13}{20} = 0.007 \text{ in/in}^3 \quad (2.34)$$

$$\frac{\Delta DPF_4}{\Delta V} = \frac{0.25}{20} = 0.013 \text{ in/in}^3$$

This means that for every cubic in. of volume added to bar 1, the drift in direction of the virtual load will decrease by approximately 0.007 in. Similarly for bar 4, the drift will decrease by approximately 0.013 in. for every cubic in. of volume added. The change in a member's displacement participation factor per change in volume is known as the Sensitivity Index (*SI*) (Charney 1993). A member's *SI* is a measure of how sensitive it is to change. Hence adding a given volume of material to members with a higher *SI* will reduce the drift more than adding volume to a member with a smaller *SI*. If the members change in volume is taken as zero, *SI* can be stated for each member *i* as follows:

$$SI_i = \lim_{\Delta V \rightarrow 0} \left(\frac{\Delta DPF_i}{\Delta V_i} \right) = \left(\frac{dDPF_i}{dV_i} \right) = \frac{DPF_i}{V_i} \quad (2.35)$$

Using Eq. (2.35) the *SI*s for members 1 and 4 are:

$$SI_1 = \frac{2.03}{288} = 7.1$$

$$SI_4 = \frac{2.03}{144} = 14.1$$

Now, using the Sensitivity Indices for members 1 and 4 it is simply a matter of noticing that member 4 is approximately twice as sensitive to change as member 1. Therefore, it can be stated that adding material to members with higher *SIs* is more beneficial. Conversely, when removing material from members, it is more beneficial to remove material from members with lower *SIs*. To illustrate this point the volume of material in member's 1 and 4 were varied, with the total volume of material in the structure remaining constant. Notice in Fig. 2-9, that as material is added to member 1 and taken away from member 4 the displacement approaches a minimum value. Once the SI_1/SI_4 ratio reaches 1.0, the displacement reaches its lowest point possible when considering only these two members. When the SI_1/SI_4 ratio increases above 1.0 the displacement then begins to increase again. With this in mind, it can be stated that a structure has reached its optimal design for stiffness given a certain volume of material when the *SI* value for each member is equal. At this point, there would be no benefit in increasing or decreasing the volume in any member. The values in Fig. 2-9 assume that there is no change in overall volume of material in the structure, the exact volume removed from one member is added to the next.

In theory, a structure with all of the members having an equal *SI* is optimal, but this is not a practical design (Charney 1995). There are many factors which limit the placement of material in a structure. For example, when reducing the size of members, the section cannot be reduced such that the allowable stresses in the member are exceeded. Also, there are architectural limitations, such as depth restrictions on members based on floor to floor spacing. Moreover, there are limitations on sections sizes readily available from fabricators, so it is not always practical to select the next size up or down for a member. In addition, it is likely the case that using the same section size for multiple members in a structure is more economical. In which case, selecting a different section size for a more efficient design is not an economically wise decision. These are only a few of the limiting factors which do not allow for design where all of the members have

the same SI value. Therefore, practically speaking, this design is not possible but the designer should use the constraints to come up with the best design possible.

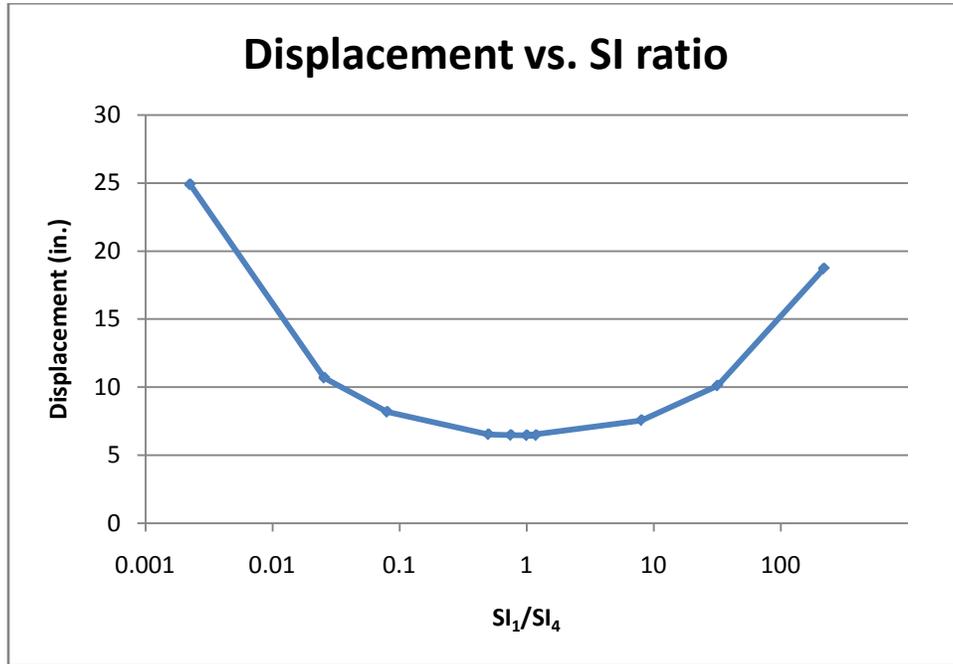


Figure 2- 9: Displacement vs. SI ratio

A member's SI is a very useful tool in determining whether it is more effective to increase or decrease the size of a member. Refer to Table 2-2 for the SIs for all members in the truss example. This method for calculating member SIs may also be extended for use in frame structures. To do so, all components of deformation in the DISPAR factor need be accounted for, Eq. (2.30). An effective resizing method using Sensitivity Indices will be discussed in a later section.

2.3.4 JOINT COMPONENTS USING VIRTUAL WORK

In structural analysis it is essential to properly account for deformations in the beam-column region, which may add up to nearly 30 percent of the overall drift of a structure. Refer to Charney (1990) for an in depth study of the beam-column joint contributions. There are two different methods used to model the beam-column joint region. The first is to model the joint

explicitly with rigid links, rotational springs and shear panels, which are referred to as mechanical models. Mechanical models can be very accurate but the average engineer is either “unaware or unconcerned” with the capabilities in modern software to implement them (Berding 2006), and therefore they are not typically used. Two examples of mechanical models are the Krawinkler and the Scissor models. Refer to Charney and Marshall (2006), Charney and Pathak (2007a & 2007b), and Berding (2006) for an in depth study of mechanical joint models. The other approach to handling the joint region is a modified force method that makes use of equilibrium in the joint. This approach is known as the fictitious joint method and may be applied when the joints are not explicitly modeled. This section will focus on the later of the two methods and show the development of a virtual work based approach to calculate displacement participation factors for the joint regions.

The fictitious joint method is developed on the basis of a subassemblage, therefore a few assumptions must be made when extending to a full frame assembly. These assumptions are, inflections points are located at midspan of the beams and columns, and the beams and columns on either side of the joint are the same size (not necessarily that the beams and columns are the same size). Refer to Fig. 2-10 for an example of the beam-column joint subassemblage. Both of the assumptions are not always the case, so the method is approximate but has been shown to produce reasonably accurate results when compared to a detailed finite element analysis of the region (Charney and Pathak 2007a).

The subassemblage found in Fig. 2-10(b) will be used in the discussion on calculating the joint components of virtual work. Recall from Eq. (2.30) that the DISPAR factor for a column including axial, flexural, and shear components can be calculated as follows:

$$DPF_{col} = \frac{1}{Q} \left[\int_0^H \frac{P(x)p(x)}{AE} + \int_0^H \frac{M(x)m(x)}{EI} + \int_0^H \frac{V(x)v(x)}{GA} \right] \quad (2.36)$$

where Q represents the virtual load applied to the structure, H is the height of the column, followed by the axial, flexural and shear component of deformation in the column (see Section 2.2 for description of each component). The total displacement at the location of load Q , not including the joint component, is equal to the summation of all the displacement participation factors of all the columns and beams in the structure. When including the deformations in the joint region, the displacement at the location of Q is calculated as follows:

$$\Delta = DPF_{C,F} + DPF_{C,S} + DPF_{B,F} + DPF_{B,S} + DPF_{J,F} + DPF_{J,S} \quad (2.37)$$

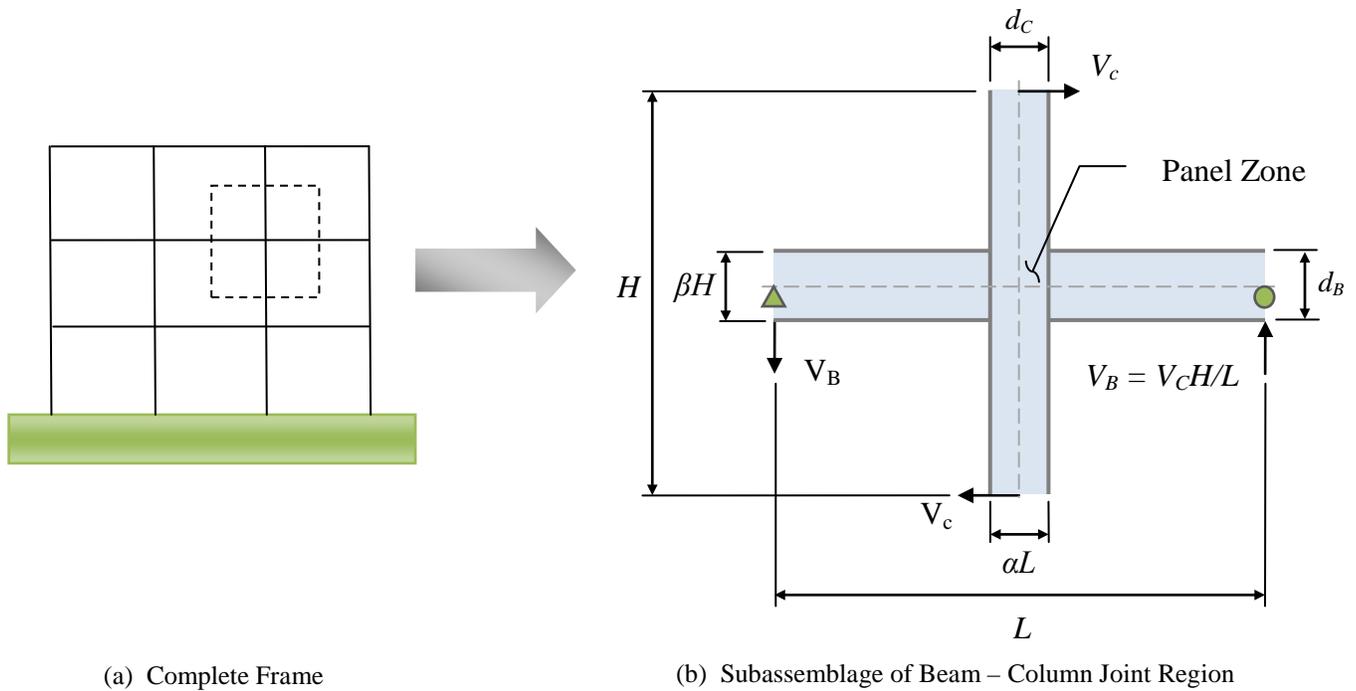


Figure 2-10: Frame for Joint Region (Charney and Pathak 2007(a))

Before developing the method including the joint region, it will be helpful to define two dimensionless parameters which will simplify the equations.

$$\beta = \frac{d_B}{L} \quad (2.38)$$

$$\alpha = \frac{d_C}{H} \quad (2.39)$$

where d_B and d_C are the beam and column depths respectively and H and L are the center to center distance of the columns and beams respectively. Now working with the subassembly shown in Fig. 2-10 to develop the virtual work contributions of the beam-column joint, start by applying equal and opposite shear forces, V_c , at the top and bottom of the column in the subassembly. These forces are applied to represent the actual behavior of this subassembly when it is a part of a full scale structure under real lateral loading conditions. The subassembly is simply supported at the midspan of both beams, which represents the inflection points discussed earlier. Only flexural and shear deformations will be considered in the subassembly because there are no axial forces acting. Because the forces in the joint are indeterminate, it must be assumed the moments in the beam-column joint are completely resolved into flange force couples in the column and beam. Notice the breakdown of forces in Fig. 2-11. This assumption produces minimal errors (Charney and Pathak 2007a).

$$F_{BF} = \frac{0.5L(1 - \alpha)(V_c H/L)}{\beta H} = \frac{0.5V_c(1 - \alpha)}{\beta} \quad (2.40)$$

$$F_{CF} = \frac{0.5V_c H(1 - \beta)}{\alpha L} \quad (2.41)$$

where F_{BF} and F_{CF} are the flange force couples in the beam and column respectively, resulting from the moment in the joint created by the force V_c .

Shear in the joint region of the column is equal to $V_{CJ} = V_c - 2F_{BF}$ which simplifies to:

$$V_{CJ} = \frac{V_c(1 - \alpha - \beta)}{\beta} \quad (2.42)$$

Shear in the joint region of the beam is equal to $V_{BJ} = V_c - 2F_{CF}$ which simplifies to:

$$V_{BJ} = \frac{V_c H(1 - \alpha - \beta)}{\alpha L} \quad (2.43)$$

Now solving for the shear stress along the joint region, which is the same whether solving with the horizontal or vertical shear.

$$\tau_J = \frac{V_{CJ}}{\alpha L t_p} = \frac{V_{BJ}}{\beta H t_p} = \frac{V_c(1 - \alpha - \beta)}{\alpha \beta L t_p} \quad (2.44)$$

where t_p is the thickness of the panel zone, which may include the thickness of doubler plates. If both the numerator and denominator are multiplied by the height H , the effective volume of the panel zone $v_p = \alpha \beta L H t_p$ and the shear stress simplifies to:

$$\tau_J = \frac{V_c H (1 - \alpha - \beta)}{v_p} \quad (2.45)$$

The moments at the center of the column and beam may be found in Fig. 2-12. From the moment diagrams in Fig. 2-12, notice that M_{BC} and M_{CC} are not exactly equal to zero. Therefore, equilibrium is not satisfied in the joint region with this method since M_{BC} does not equal M_{CC} which does not equal zero. This is due to the assumption that all moment is resisted by the flanges of the beam and column, which produces only a minor error. The moment at the center of the column is equal to:

$$M_{CC} = 0.5 H V_c - 0.5 \beta H (2 F_{BF}) = 0.5 V_c H \alpha \quad (2.46)$$

Similarly the moment at the center of the beam is equal to:

$$M_{BC} = 0.5 L V_c H / L - 0.5 \alpha L (2 F_{CF}) = 0.5 V_c H \beta \quad (2.47)$$

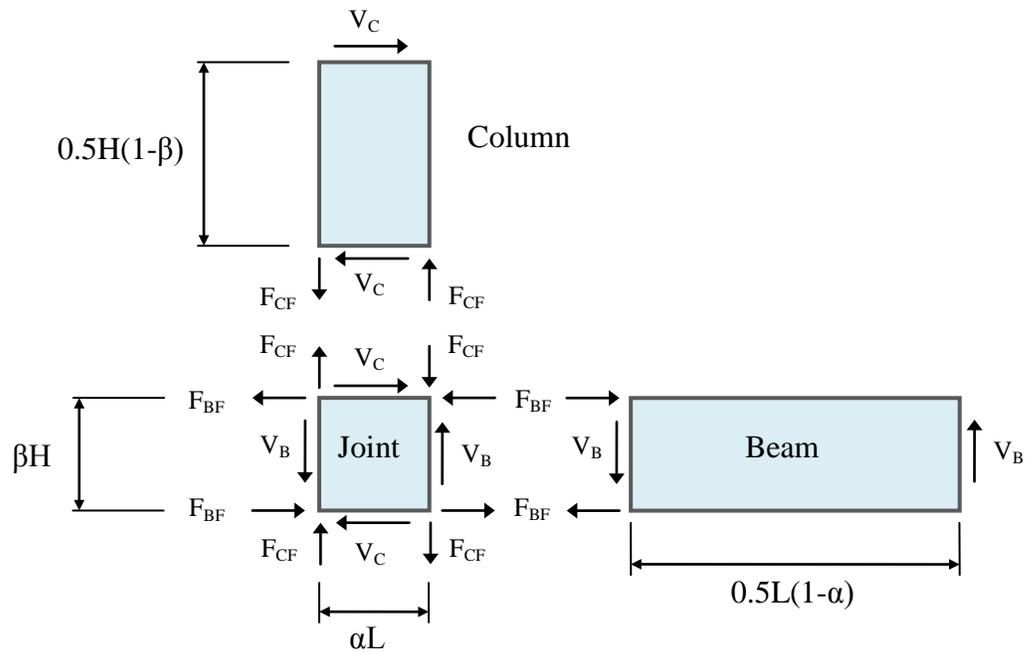
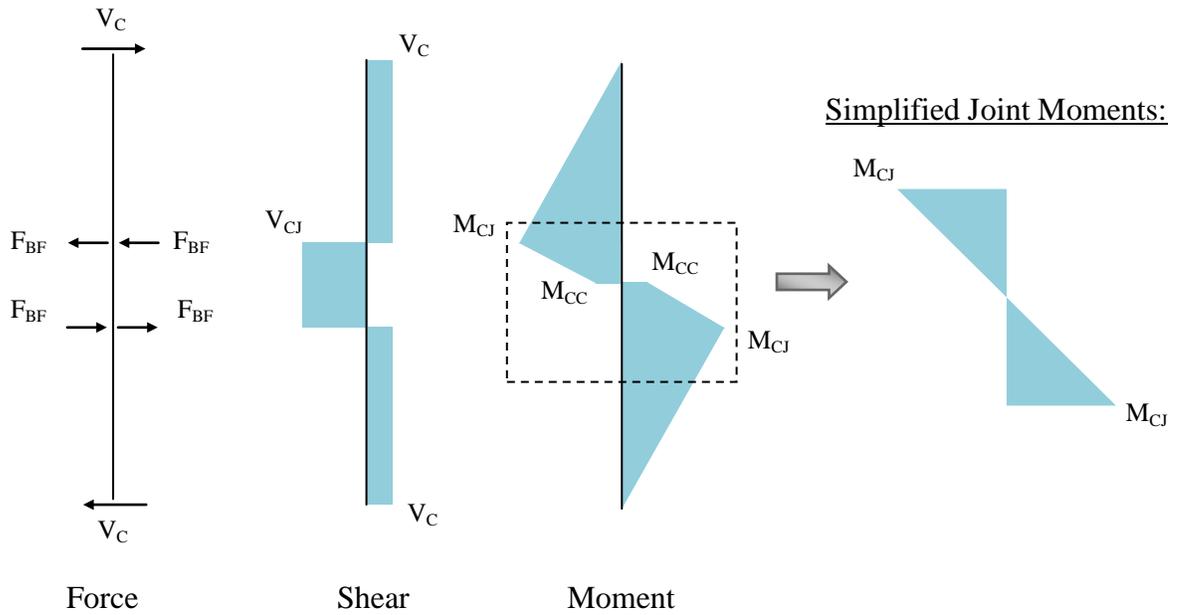
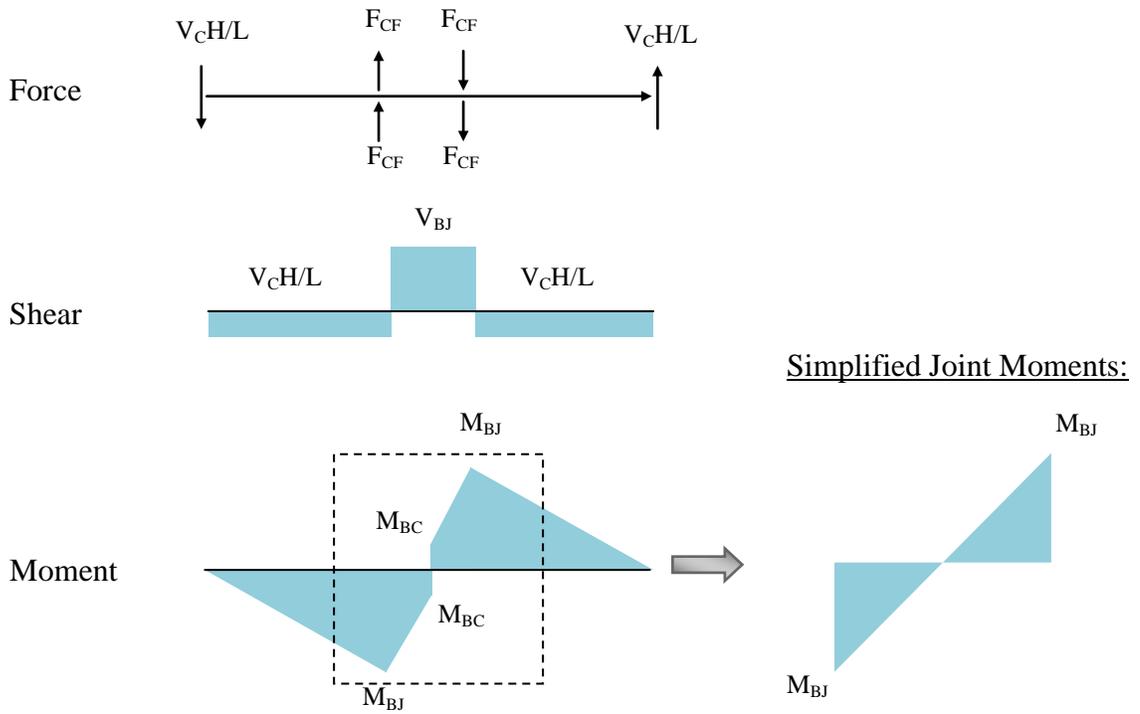


Figure 2-11: Forces in Beam-Column Joint Region (Charney and Pathak 2007(a))



a) Column Diagrams



b) Beam Diagrams

Figure 2-12: Force Diagrams of Beam-Column Region (Charney and Pathak 2007(a))

Now that the shear and moment diagrams, including the joint region, have been developed the displacement participation factors for the beams, columns, and the beam-column joint region can be written. Keep in mind that the only difference between the integrals including the beam-column joint region and the integrals not including the joint region are the bounds of the integral. In other words, if α and β are taken as zero, Eq. (2.48) through Eq. (2.51) will produce the same solutions as Eq. (2.36) for the beam and column displacement participation factors. The equations for the displacement participation factors for the beams and columns outside the joint region are as follows:

Column and Beam shear components, respectively

$$DPF_{C,S} = \frac{2}{Q} \int_0^{0.5H(1-\beta)} \frac{V_c v_c}{GA_{sc}} = \frac{v_c V_c H(1-\beta)}{Q GA_{sc}} \quad (2.48)$$

$$DPF_{B,S} = \frac{2}{Q} \int_0^{0.5L(1-\alpha)} \frac{\frac{V_c H}{L} \frac{v_c H}{L}}{GA_{SB}} = \frac{v_c V_c H^2(1-\alpha)}{Q LGA_{SB}} \quad (2.49)$$

Column and Beam flexural components, respectively:

$$DPF_{C,F} = \frac{2}{Q} \int_0^{0.5H(1-\beta)} \frac{V_c x v_c x}{EI_c} = \frac{v_c V_c H^3(1-\beta)^3}{Q 12EI_c} \quad (2.50)$$

$$DPF_{B,F} = \frac{2}{Q} \int_0^{0.5L(1-\alpha)} \frac{\frac{V_c H x}{L} \frac{v_c H x}{L}}{EI_B} = \frac{v_c V_c H^2(1-\alpha)^2 L}{Q 12EI_B} \quad (2.51)$$

where the 2 represents both sides of the member in the subassembly. The displacement factors for the joint region are based on three components, the shear in the panel, column flexure in the panel and beam flexure in the panel. The shear component is developed using the shear in the column only, this is to avoid double counting the shear stresses in the joint region.

$$DPF_{J,S} = \frac{1}{Q} \int_0^{H\beta} \frac{V_c(1-\alpha-\beta) v_c(1-\alpha-\beta)}{G\alpha L t_p} = \frac{v_c V_c H(1-\alpha-\beta)^2}{Q \alpha \beta G L t_p} \quad (2.52)$$

By multiplying the numerator and denominator by H and remembering from above that $v_p = \alpha \beta L H t_p$, the above equation may be simplified to:

$$DPF_{J,S} = \frac{v_c V_c H^2 (1-\alpha-\beta)^2}{Q G v_p} \quad (2.53)$$

There is some uncertainty as to the proper method for calculating the moment of inertia for the flexure portions of the joint contribution. Refer to Charney and Pathak (2007a) for an approach of this calculation. Using the simplified diagrams for moment in the joint region the flexural displacement participation factors are as follows:

$$DPF_{J,CF} = \frac{v_c V_c H^3 \beta (1-\beta)^2}{Q 12 E I_{CJ}} \quad (2.54)$$

$$DPF_{J,BF} = \frac{v_c V_c H^2 L \alpha (1-\alpha)^2}{Q 12 E I_{BJ}} \quad (2.55)$$

As can be seen, the virtual work method is a very powerful tool when determining the contribution of the beam-column joint region to the overall drift.

2.3.5 FRAME EXAMPLE INCLUDING JOINT COMPONENTS

To illustrate the virtual work method for calculating joint contributions a three story, three bay steel frame was analyzed with DISPAR for SAP2000. The frame used for analysis is shown in Fig. 2-13. For this example, the analysis was run using the centerline assumption and the beam – column joint region was handled using the method described in Section 2.3.4. The member DPFs may be found in Table 2-3.

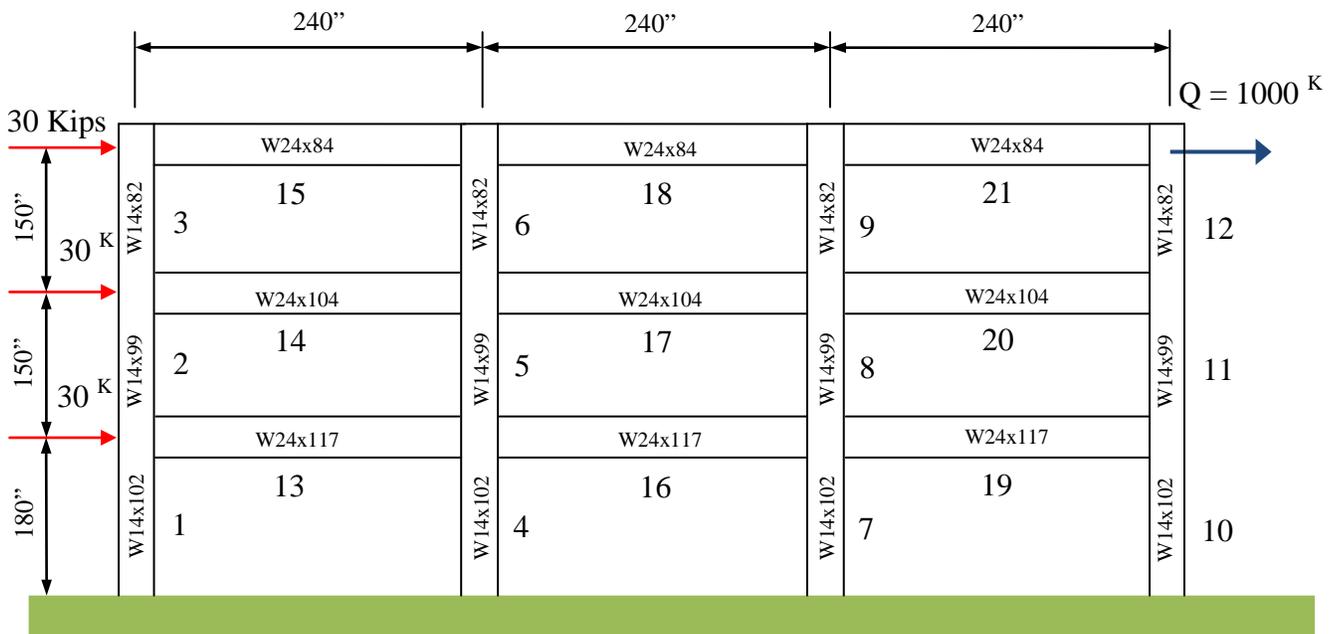


Figure 2- 13: 3 Story 3 Bay Steel Frame

Table 2- 3: Member DISPAR Factors for Joint Region

Number	Axial (in.*1000)	Flexural (in.*1000)	Shear (in.*1000)	Joint (in.*1000)	Total (in.*1000)
Columns					
1	3.2	55.2	8.3	11.0	77.7
2	0.9	12.1	3.7	8.9	25.5
3	0.1	8.5	2.0	6.3	16.9
4	0.2	70.9	11.8	15.7	98.7
5	0.0	31.5	9.1	22.0	62.6
6	0.0	18.7	4.2	13.4	36.3
7	0.2	70.3	11.7	15.6	97.8
8	0.0	31.5	9.1	22.0	62.6
9	0.0	17.9	4.0	12.8	34.8
10	3.1	53.7	8.0	10.7	75.5
11	0.8	12.5	3.8	9.2	26.2
12	0.1	7.4	1.8	5.6	14.9
Subtotal	8.6	390.2	77.5	153.2	629.5

Beams					
13	-0.2	42.7	6.2	8.7	57.5
14	0.2	26.5	3.8	5.3	35.8
15	6.3	7.3	0.8	1.4	15.8
16	0.0	21.0	3.4	4.7	29.0
17	0.1	15.1	2.3	3.3	20.8
18	2.4	3.7	0.5	0.8	7.3
19	0.1	38.5	6.2	8.6	53.3
20	0.0	23.3	3.6	5.1	32.0
21	0.4	5.9	0.7	1.3	8.3
Subtotal	9.2	183.8	27.5	39.2	259.8
Total	17.8	574.0	105.3	192.3	889.3

The member DPFs in Table 2-3 are divided up into axial, flexural, shear, and joint components of deformation. The summation of the member DPFs, found in the lower right corner of Table 2-3, is equal to 1000 times the displacement of the frame. This is because the virtual load applied to the frame was equal to 1000 kips, not 1 kip. Therefore, the total displacement at the location of the applied virtual load, Q , is equal to 0.889 in.

The contributions from the column joint and beam joint regions are equal to 0.153 in. and 0.039 in. respectively, giving a total joint contribution of 0.192 in. Therefore, the joint deformations account for 21.6 percent of the total deformation at the roof level of the frame. If the frame was modeled using totally rigid joints in analysis, or clearspan analysis, the total displacement would be equal to approximately $0.889 - 0.192$ or 0.697 in. This shows how important it is to account for the deformations in the beam – column joint region, otherwise the analysis could report unconservative displacements. Also, notice that the shear deformations contribute 0.105 in. to the overall drift. This accounts for 11.8 percent of the overall drift and should not be ignored.

2.3.6 LIMITING PERIODS OF VIBRATION USING VIRTUAL WORK

In the previous two sections it was shown that virtual work can be extended a variety of different ways to determine the behavior of a structure and how each component contributes. The principle of virtual work can also be extended to optimizing the placement of material in a structure to limit its natural period of vibration. When dealing with high rise structures, it is no longer the drift that controls the design, it is the perception of motion felt by the occupants (motion sickness). The main factor in perception of motion is the accelerations felt by the occupants of the structure. Occupants are typically more sensitive to motion when they are in a space for an extended period of time. Therefore, the place to limit accelerations felt by the structure are the floors deemed to be residences. In high – rise construction, the layout of mixed – use facilities usually consists of retail on the first few floors, then office space, and condos or living space take up the remaining upper floors. In other words, the area most susceptible to perception of motion by the occupants experiences the highest accelerations. Something must be done to control the accelerations of the structure and one of the best ways to do so is limit the natural period of vibration of the structure, which can be carried out using virtual work. The method that will be discussed in this section is based on the work done by Henige (1991).

This method works by setting the internal virtual work densities (same thing as member SIs discussed in section 2.3.3) for all members equal, ensuring optimal placing of the material. In other words, if the work density for each member is equal, then each member is doing the same

amount of work per volume of material. If the work densities are not equal for all of the members, there is excess material in some places and not enough in others, therefore not as efficient. The work density for this method is essentially the same thing as the sensitivity index discussed in Section 2.3.2, and may be calculated as follows:

$$wd_i = \frac{w_i}{v_i} \quad (2.56)$$

where wd_i is the work density for member i , w_i is the work quantity for member i (internal virtual strain energy, see eq. (2.14)), and v_i is the volume of member i .

To relate the work densities for each member back to the external work for the structure we must first look at the undamped free vibration equation for a structure with a circular frequency, ω , and fundamental mode shape, Φ .

$$D(t) = \Phi \cos(\omega t) \quad (2.57)$$

where $D(t)$ is the displacement history of the structure and the velocity vector of the structure is equal to:

$$V(t) = D'(t) = -\omega \Phi \sin(\omega t) \quad (2.58)$$

The Kinetic Energy of the system:

$$KE = \frac{1}{2} V^T M V = \frac{1}{2} \Phi^T M \Phi \omega^2 \sin^2(\omega t) \quad (2.59)$$

The internal Strain Energy of the system:

$$SE = \frac{1}{2} D^T K D = \frac{1}{2} \Phi^T K \Phi \cos^2(\omega t) \quad (2.60)$$

where K and M are the structural stiffness and mass matrices of the system, respectively. By conservation of energy of the undamped system we can say the maximum kinetic energy must be equal to the maximum strain energy:

$$KE_{max} = \frac{1}{2} \Phi^T M \Phi \omega^2 = SE_{max} = \frac{1}{2} \Phi^T K \Phi \quad (2.61)$$

By solving Eq. (2.61) for the circular frequency of the structure, ω :

$$\omega^2 = \frac{\Phi^T K \Phi}{\Phi^T M \Phi} = \frac{K^*}{M^*} \quad (2.62)$$

where K^* and M^* are the generalized stiffness and mass of the structure, respectively.

Since the mass of the structural components is a small portion of the total mass of the structure, optimizing the placement of mass in the structure is not necessarily beneficial nor is it practical. It is helpful to minimize the volume of the material needed to produce a certain stiffness, K , of the structure. Therefore, this will be the goal of the optimization process. First, we must determine a relationship between structural stiffness, circular frequency and the overall energy in the system. If it is assumed that the virtual loads produce the fundamental mode shape, Φ , then we can say:

$$P = Q = K \Phi \quad (2.63)$$

In other words, applying a virtual load of $Q = K \Phi$ forces the structure to deflect into its fundamental mode shape. This will allow us to find the work associated with a structure having a stiffness of K , which is a function of the structures fundamental period $T = 2\pi/\omega$. From there we can say the external work is equal to:

$$K^* = \Phi^T K \Phi = \Phi^T Q = Q^T \Phi = P^T \Phi \quad (2.64)$$

Which is the work required to be performed on a structure in order for that structure to have a stiffness based on a fundamental period of T . Relying on the principle of conservation of energy (work external must equal work internal):

$$K^* = P^T \Phi = \sum w_i \quad (2.65)$$

Now that a relationship between the generalized stiffness of the structure and the internal virtual strain energy has been established, the material must be distributed such that the work density for each member is equal. This is done by using virtual work constraints and Lagrange multipliers to determine an optimal virtual work density for each member. The virtual work constraint is used to limit the overall displacement of the structure based on the fundamental period. The Lagrange multipliers allow for the optimization of the virtual work density experienced by each member. Once the optimization is complete, the virtual work density for each member will be equal. This is the same as saying the Sensitivity Index for each member is equal. From here, the volume of each member may be calculated and a section selected based on this volume.

This method assumes member forces do not vary as the volume changes. It also assumes that the shear area and moment of inertia change in direct proportion with the volume. These assumptions must hold true, otherwise changing a member's volume will not necessarily correlate to a linear change in the virtual work density of a member. The method can be proven for statically determinate truss structure, but does not hold for indeterminate structures. Though the method cannot be proven for indeterminate structures, it is still effective with a small number of iterations (Henige 1991).

2.4 OPTIMIZATION OF STRUCTURAL STIFFNESS AND WEIGHT WITH VIRTUAL WORK

The previous sections have shown the development of the virtual work method along with examples of how it may be applied to examine the behavior of a structure. This section will discuss various methods that have been utilized to optimize the design of structures, with virtual work at the foundation of the method. The first method makes use of the Sensitivity Indices (SIs) of individual members to optimally place material throughout the structure. Using the SI allows for designer interaction, also referred to as “the human loop”, when updating the member sizes. The second method is more mathematically based, known as a discrete optimization problem. In the discrete optimization problem, Lagrange Multipliers are used with regression constants representing the various section types in the AISC steel manual, to determine optimal sizes for members. This method is iterative in nature, but is done with the click of a button rather than using designer’s intuition. Both methods have been shown to produce effective designs, see Charney (1993) and Chan et al.(1995), for examples of each.

2.4.1 USING DISPAR FACTORS AND SIS

Recall from Section 2.3.2 using the SIs of members 1 and 4 to determine which is more sensitive to change in volume. Also, recall that it is beneficial to add material to the members with higher SIs and remove material from members with lower SIs. Therefore, if all members had the same SI the design of the truss would be optimal and there would be no benefit in redistributing the material. To perform this task, select a member n and for all members i set the SI equal to that of member n .

$$SI_n = \frac{P_n p_n L_n}{A_n E V_n} = \frac{P_n p_n}{A_n^2 E} \quad (2.66)$$

Now setting SI_i equal to SI_n :

$$\frac{P_i p_i}{A_i^2 E} = \frac{P_n p_n}{A_n^2 E} \quad (2.67)$$

The updated area for member i becomes:

$$A'_i = A_n \sqrt{\frac{P_i p_i}{P_n p_n}} \quad (2.68)$$

Once the volume has been distributed such that the SIs for all of the members is equal, the displacement may be written as:

$$d' = \sum_{i=1}^m DPF_i = \sum_{i=1}^m V_i SI_i = V'_{Tot} SI' \quad (2.69)$$

This is not to say that the displacement is proportional to the volume with the SI remaining constant. As the total volume increases, the SI will in turn decrease, decreasing the overall displacement. The total volume achieved by following the above procedure generally gives a volume not equal to the original volume of the structure. In order to achieve the original volume, multiply the calculated area of member i from Eq. (2.68) by the ratio of the original volume over the new volume. See Table 2-4 for an example on how to follow the procedure.

$$A''_i = A'_i \frac{V_{Orig}}{V_{Tot}} \quad (2.70)$$

Table 2- 4: Updated DISPAR Factors with Equal SIs

Bar Number	Length (in)	Area (in ²)	Volume (in ³)	A' (in ²)	V' (in ³)	A'' (in ²)	DPF _{new} (in.)	SI _{new} *1000 (in. ⁻²)
1	288	1.0	288.0	1.66	479	1.58	1.29	2.83
2	288	1.0	288.0	0.48	138	0.45	0.37	2.83
3	288	1.0	288.0	0.96	277	0.91	0.74	2.83
4	144	1.0	144.0	2.35	338	2.23	0.91	2.83
5	212	1.0	212.3	0.71	150	0.67	0.40	2.83
6	212	1.0	212.3	0.71	150	0.67	0.40	2.83
7	212	1.0	212.3	1.00	212	0.95	0.57	2.83
8	212	1.0	212.3	1.00	212	0.95	0.57	2.83
Summation			1857.3		1957		5.26	

Note that the sections listed Table 2-4 in are not realistic sizes, it is meant to show the process and how it is carried out. It is up to the designer to select a section that has properties similar to the required properties calculated. Additionally if a larger section is required for strength than was calculated using the *SI* procedure, the designer should select a section based on the controlling limit state. Instead of following the process above, if a target displacement is desired, Eq. (2.66) and Eq. (2.69) may be solved to obtain the following equation (see Charney (1995) for a more detailed derivation):

$$A''_n = \frac{\sqrt{P_n p_n}}{\delta_{Target} E} \sum_{j=1}^m L_j \sqrt{P_j p_j} \quad (2.71)$$

This method is only applicable in determinate truss structures where the forces in the members do not vary with change in area. With indeterminate frame structures, this method is an iterative procedure where the member sizes are updated, the analysis is re-run and the analytical displacement is compared with the target displacement. For frame members, an estimate for determining the new cross-section area is as follows:

$$A_{New} = \sqrt{\frac{A_{Old} DPF_{Old}}{SI_{Target} L}} \quad (2.72)$$

This estimating is accurate when a member is within small changes in sections sizes (i.e. with W21s). This is due to the fact that there exists a near linear relationship between cross-sectional area and the radius of gyration squared ($r = \sqrt{I/A}$). The derivation of Eq. (2.72) can be found in Charney (1993). Once the new area is determined, calculate the updated DISPAR factor with the follow equation:

$$DPF_{New} = DF_{Old} \left(\frac{I_{Old}}{I_{New}} \right) + DA_{Old} \left(\frac{A_{Old}}{A_{New}} \right) + DS_{Old} \left(\frac{A_{S,Old}}{A_{S,New}} \right) \quad (2.73)$$

Where *DF*, *DA*, and *DS* are the flexural, axial, and shear components of deformation respectively. Then calculate the member's new SI factor using Eq. (2.35) and compare with the

target SI value. If the new SI and the calculated value are close (within 15%), move on to the next member, otherwise select a new section and try again.

This method can be a very powerful tool in designing an optimal volume of material for a given frame, both determinate and indeterminate. As mentioned previously, for indeterminate structures it is an iterative procedure, which should converge within two to three iterations. These iterations are what allow for the “human loop”, giving the designer a chance to make decisions on sections sizes based on judgment and experience.

2.4.2 AUTOMATIC RESIZING TECHNIQUE

There are a number of resizing techniques which have been developed. The technique this section will cover was developed by Chan and his colleagues (Chan and Park 1996; Chan et al. 1993, 1995). The method uses an optimization technique based on an Optimality Criterion (OC) developed by Chan et al. (Chan 1995), which minimizes the weight of the structure given various drift and strength constraints. It makes use of the virtual work quantities along with Lagrange multipliers to create a recursive resizing technique. The Lagrange multipliers serve as sensitivity factor in order to determine whether a member should be resized or not. Another method presented by William Baker (Baker 1991), is very similar to the method presented by Chan. It also used the virtual work quantities along with Lagrange multipliers for the members to determine an optimal arrangement of material. Bakers approach will not be presented in this section, but refer to Baker (1991) for the details of this method.

The rest of this section covers the development and procedure for applying Chan’s method for structural optimization. The goal of this method is to minimize the weight of the structure given a number of constraints, which may include drift constraints, strength constraints or period constraints. This could also be referred to as maximizing the stiffness of a structure while limiting the volume of material. The optimization process for a structure with $i = 1, 2, \dots N$

number of members (or member groups), $j = 1, 2, \dots, N_j$ storys, $k = 1, 2, \dots, N_k$ column lines, and $l = 1, 2, \dots, N_l$ loading cases, may be stated as:

Minimize Weight:

$$W(A_i) = \sum_{i=1}^N w_i A_i \quad (2.74a)$$

Subject to (constraint equations):

$$d_{kjl}(A_i) = (\delta_{kjl} - \delta_{kj-1l})/h_j \leq d_j^U \quad (2.74b)$$

$$\sigma_p \leq \sigma_p^U \quad (2.74c)$$

$$A_i^L \leq A_i \leq A_i^U \quad (2.74d)$$

where W is the overall volume of the structure, w_i and A_i are the weight density (per cross sectional area) and the cross sectional area for member i , respectively. Eq. (2.74b) describes the interstory drift between levels j and $j - 1$, where δ_{kjl} is the drift along column line k , at story j (or $j - 1$), due to loading condition l . The difference in drift is then divided by the height between story j and $j - 1$, h_j , then compared with the drift limit d_j^U . Eq. (2.74c) compares the stress state in member P , σ_p , with the stress limit, σ_p^U . Eq. (2.74d) describes the cross sectional area limits for a member where A_i^L and A_i^U are the lower and upper bound limits, respectively. In order to calculate the displacement at level j , employ the principle of virtual work:

$$d_{kjl} = \sum_{i=1}^N \int_0^{L_i} \left(\frac{F_x f_x}{EA} + \frac{F_y f_y}{GA_Y} + \frac{F_z f_z}{GA_Z} + \frac{M_x m_x}{GI_x} + \frac{M_y m_y}{EI_y} + \frac{M_z m_z}{EI_z} \right)_{ijkl} \quad (2.75)$$

where L_i is the length of member i , E and G are the modulus of elasticity and shear modulus; A , A_y , A_z are the area and shear areas for the cross section; I_x , I_y , I_z are the moment of inertias for the cross section; F_x , F_y , F_z are the real internal forces experienced by the member; M_x , M_y , M_z are the real internal moments experienced by the member; f_x , f_y , f_z are the virtual internal forces experienced by the member; m_x , m_y , m_z are the virtual internal moments experienced by the member. For standard steel sections A_y , A_z , I_x , I_y , and I_z can be related to the cross sectional area, A , by a regression analysis. This is helpful, because the virtual work equation can now be expressed only in terms of the cross sectional area, instead of multiple cross sectional properties. This allows for the optimization process to be in terms of only one member property. For example, when relating the moment of inertia about the z-axis with the cross sectional area:

$$\frac{1}{I_z} = \frac{C}{A} + C' \quad (2.76)$$

where C and C' are regression constants. This holds true as long as the section remains within the same shape group (i.e. W21 sections). As mentioned, using these relationships for member properties the drift can now be expressed in terms of the A , E , and member forces by plugging Eq. (2.76) into Eq. (2.75):

$$d_{kjl}(A_i) = \sum_{i=1}^N \left(\frac{e_{ikjl}}{A_i} + e'_{ikjl} \right) \quad (2.77)$$

where e_{ikjl} is the virtual strain energy coefficient for member i and e'_{ikjl} is a correction factor for frames, which is equal to zero for trusses. Both e_{ikjl} and e'_{ikjl} were derived on the basis that the floor acts as a rigid diaphragm, having a lateral and torsional component. The derivation of e_{ikjl} and e'_{ikjl} involves the relationships found in Eq. (2.76) being substituted into Eq. (2.75), then applying a virtual force in the lateral direction and a virtual torque about the vertical axis. The derivation may be found in Chan (1995). Now using the formulation of Eq. (2.77), the optimization problem found in Eq. (2.74a) can be written as (excluding the strength constraint):

Minimize Weight:

$$W(A_i) = \sum_{i=1}^N w_i A_i \quad (2.78a)$$

Subject to (constraint equations):

$$d_s(A_i) = \sum_{i=1}^N \left(\frac{e_{is}}{A_i} + e'_{is} \right) \leq d_s^U \quad (2.78b)$$

$$A_i^L \leq A_i \leq A_i^U \quad (2.78c)$$

where s is equal to the number of stiffness constraints which is equal to the number of drift constraints. The optimization technique for solving the Eq. (2.78), was derived by Chan based on a set of an Optimality Criteria. This method has been shown to be computationally efficient for large scale structures. For derivation and supporting examples refer to Chan et al. (1995). The set of optimality criteria are based on the minimization of an unconstrained Lagrange function:

$$L(A_i, \lambda_i) = \sum_{i=1}^N w_i A_i + \sum_{s=1}^N \lambda_s \left[\sum_{i=1}^N \left(\frac{e_{is}}{A_i} + e'_{is} \right) - d_s^U \right] \quad (2.79)$$

where the first term relates to the weight of the structure and the second term relates to the stiffness of the structure with respect to the drift constraints d_s^U , subject to a Lagrange Multiplier, λ_s . Now differentiate Eq. (2.79) with respect to A_i and setting it equal to zero:

$$\sum_{i=1}^N \lambda_s \frac{e_{is}}{w_i A_i^2} = 1 \quad (2.80)$$

where λ_s is a sensitivity weighting factor for constraint s . Members with a higher λ_s are more influential to the stiffness of the structure. Conversely, members with a zero λ_s have no influence on the stiffness of the structure. For an optimal design all λ_s are equal, this would mean that all

members have the same influence on the stiffness ($\lambda_s = 1/\lambda_i$). Recall the discussion in Section 2.3.2 on a member's Sensitivity Index. This factor and the λ_i used in Chan's method are one in the same. Eq. (2.80) can be used to derive a linear recursive relation for the cross sectional area of member i :

$$A_i^{v+1} = A_i^v \left[1 + \frac{1}{\eta} \left(\sum_{i=1}^N \frac{\lambda_s e_{is}}{w_i A_i^2} - 1 \right) \right] \quad (2.81)$$

where η represents a step-size parameter controlling convergence, and $v+1$ and v indicate successive iterations. In order to determine the current λ_s values solve the set of simultaneous equations:

$$\sum_{i=1}^N \lambda_s \sum_{i=1}^{N-\xi} \frac{e_{is} e_{it}}{w_i A_i^3} = \sum_{i=1}^{N-\xi} \frac{e_{is}}{A_i^v} - \eta (d_t^u - d_t^v) \quad (2.82)$$

where t represents a given constraint and ξ represents the number of inactive members, or members not currently being resized.

The above method presented by Chan is an iterative procedure and relies upon the convergence of λ_s . It starts by applying the optimality criteria then using the resizing algorithm to determine an optimal section size for a given member. Using this size the current λ_s is determined then compared with the λ_s for the other members. Once the procedure has reached convergence, the designer takes the suggested optimal member size and selects a discrete section. Selecting the actual section is based on the engineer's judgment and the design should be run through another analysis to confirm the sections selected.

2.5 CONCLUSION

Virtual work is a very powerful tool in determining the behavior of a structure not only on the global scale, but also individual components. This literature review has shown the development of the virtual work procedure, methods for recognizing structural behavior, and how the knowledge of this behavior can be applied to optimize the structural design. There are many methods which optimize the stiffness given a set volume of material, and it has been shown that at the center of most of the methods, lies the principle of virtual work. From this literature review, it can be shown that creating a program which performs the virtual work calculations could be very beneficial in the design process. A later section will discuss a program that does just that, followed by an updated version of this program which has been the primary focus of the research conducted in this thesis.

CHAPTER 3: LITERATURE REVIEW – MODELING COLUMN BASE FIXITY AS PARTIALLY RIGID

3.1 INTRODUCTION

As mentioned in the introduction, a goal of this thesis is to identify the behavior of various components used in metal buildings. In addition, it is desired to determine areas where the design for drift may be conservative and ways to decrease the material required to meet drift limits. The literature review in this chapter will take a look at what is currently being done or has been done with subject matters relating to the modeling of metal buildings, specifically column base connections. Typically in analysis, a connection is either modeled as pinned (allowing rotation) or fixed (allowing no rotation), yet in reality the actual behavior lies somewhere in between. The cost associated with using a connection having enough rotational stiffness to be assumed fixed usually outweighs the benefit. In most cases, the column to base connection is taken as pinned allowing for rotation at the base and not taking any moment. This assumption is conservative with regard to calculating drift and research has shown that modeling the connection as partially rigid is not only closer to the true behavior of the connection, but it also decreases the drift of the structure in analysis. In addition to a decrease in drift of the overall structure, modeling the base as partially rigid increases the buckling strength of the columns (Galambos 1960), allowing for a smaller column size. A decrease in column size would increase the drift, so careful consideration should be taken before reducing the column size if the frame is controlled by drift. The question still remains, if the actual behavior of the connection lies somewhere in between pinned and fixed, what is the behavior? There are many factors which contribute to the stiffness of a connection. In order to model the connection as partially restrained a proper understanding of how each of the contributing factors affects the stiffness of the connection is required. This section will cover the types of connections which have been researched, testing and modeling procedures used in the research, factors involved in the rotation stiffness of the connection, how to determine the stiffness, the effect of axial load on stiffness, and the increase in buckling strength by assuming a partially rigid connection. At the conclusion of this literature review,

there will be sufficient information to allow for modeling the base connections for metal buildings as partially restrained.

The first section in this chapter examines the different connection types that were tested or modeled by the researchers whose work was included in this literature review. The next section includes the different anchor bolt details used in various investigations. Each of the researchers that conducted tests on the connections (Hon and Melchers 1988; Picard and Beaulieu 1985) used a different anchor bolt detail, both details are included in this section. The next section summarizes the work completed by Hon and Melchers (1988), where different aspects of one connection type were varied to determine the most influential parts with regard to rotational stiffness. To determine the rotational stiffness of the different connection types experimental testing and finite element modeling was conducted. Experimental testing was carried out by Picard and Beaulieu (1985) and finite element modeling was completed by Hamizi and Hannachi (2007). The procedure for both the testing and modeling are discussed in Section 3.5. The results from the testing and modeling are discussed in Section 3.6. In addition to the results from Picard and Beaulieu (1985) and Hamizi and Hannachi (2007), Section 3.6 includes an analytical approach developed by Galambos (1960) for calculating the rotation stiffness of the connection. Section 3.7 discusses the work done by Galambos (1960) on the effects of partially restrained base connections on the buckling strength of columns. Lastly, Section 3.8 includes conclusions made from the literature review carried out column base connections.

3.2 TYPES OF CONNECTIONS CONSIDERED

There were three types of connections considered in the various references where research was presented on testing and modeling to determine connection rotational behavior. Refer to Fig. 3-1 for a diagram of each of the connection types. In the work carried out by Hon and Melchers (1988), only connection Type 1 was considered. All of the connection types found in Fig. 3-1 were examined in Picard and Beaulieu (1985) and Hamizi and Hannachi (2007). The connections consisted of a steel column welded to a steel plate with steel anchor bolts connecting the plate to a concrete foundation, with all steel having a modulus of elasticity equal to 29,000 ksi. The steel

used had yield strengths as follows, 36 ksi for the plate, 43 ksi for I-sections and anchors, and 50 ksi for HSS sections. The first connection was a two bolt connection with bolts located on either side of the web of an I-section. The second connection consisted of four anchor bolts, with two located on the outside end of each flange of an I-section. The third connection was another four bolt connection with two bolts on either end of a rectangular HSS section. Because metal buildings are usually made up of up tapered I-shaped sections, the rest of this section will focus on the first two types of connections.

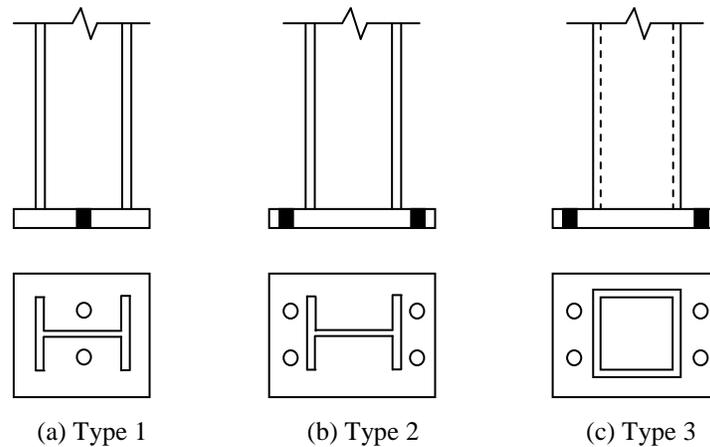


Figure 3- 1: Type of Column Base Connections (Picard and Beaulieu (1985))

3.3 ANCHOR BOLT DETAILS

This section includes the anchor bolt details for both the testing done by Hon and Melchers (1988) and Picard and Beaulieu (1985). All of the anchor bolts were “snug tightened”, therefore the effects of pretensioning on the bolts was not considered. This was done to replicate actual field conditions. The anchor bolt detail used in the testing performed by Hon and Melchers may be found in Fig. 3-2. The anchor bolts were embedded into the concrete block by using a hooked bar and the concrete block was anchored into the floor. A 0.79 in. thick layer of dry packed grout mortar was used in between the base plate and the concrete block.

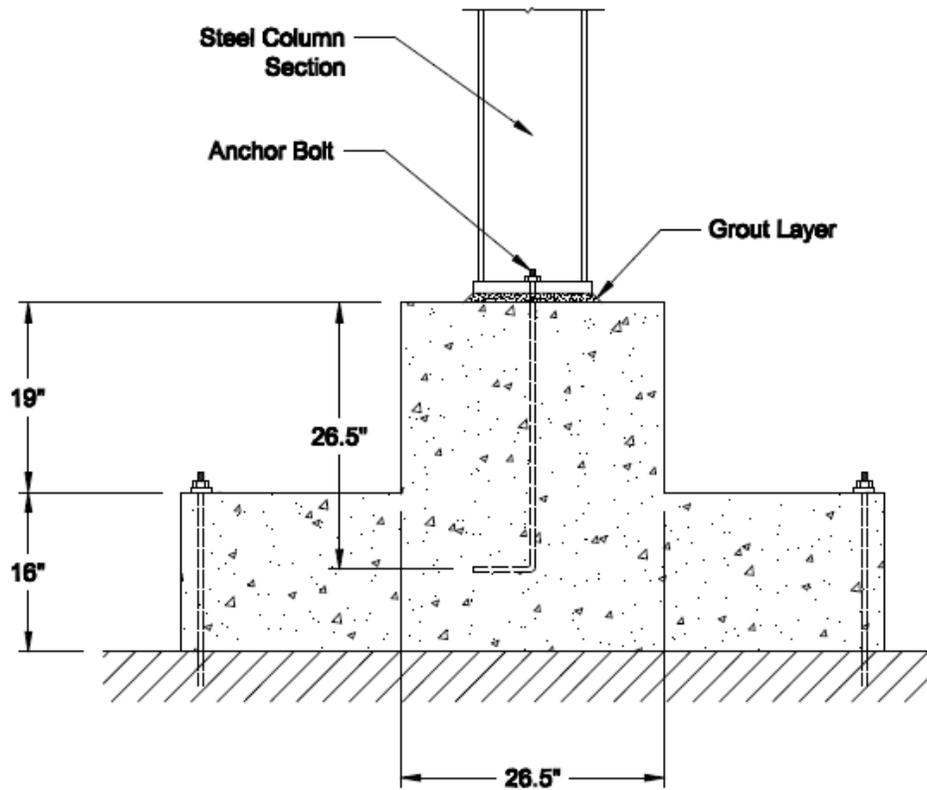


Figure 3- 2: Anchor Bolt Detail (Hon and Melchers (1988))

The anchor bolt detail found in Fig. 3-3 was used for the testing carried out by Picard and Beaulieu (1985). The anchor bolts were embedded down into a concrete block with the use of a steel bearing plate. The concrete block was then tied down into the floor. When connection Type 1 was tested, a similar set-up was used, except the two bolts were located between the flanges of the steel column. The grout layer was approximately 0.79 in. thick and the grout itself had a mean compressive strength equal to 3.6 ksi at 1-day.

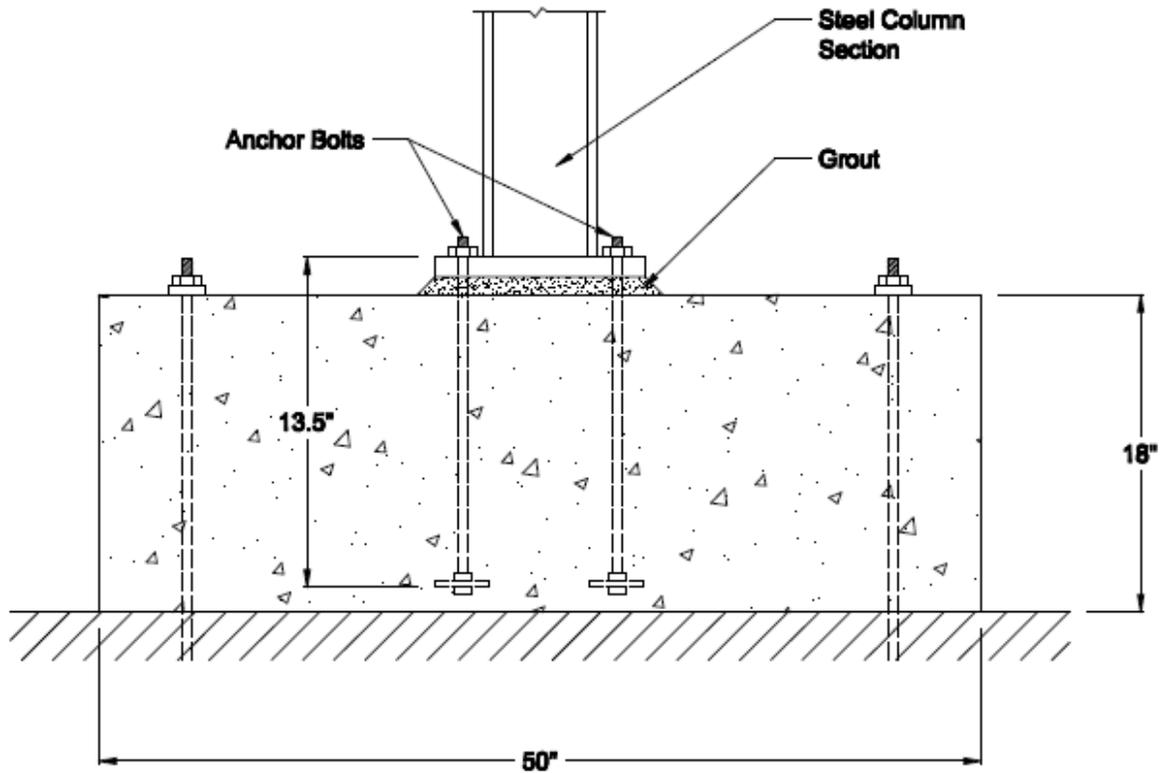


Figure 3- 3: Anchor Bolt Detail (Picard and Beaulieu (1985))

3.4 FACTORS INVOLVED IN THE ROTATION STIFFNESS OF THE CONNECTION

There are multiple factors that affect the rotational stiffness of a column base connection. These include, base plate thickness and size, anchor bolt size, column size, concrete strength, and grout strength. A series of tests were performed by Hon and Melchers (1988) and Melchers (1988), where the column size, plate thickness, and anchor size were varied to determine which was the most influential to the rotation restraint. The test was done on three different column sizes (W18x50, W12x106, W12x30), two bolt sizes (0.79 in. diameter, 0.95 in. diameter), and five base plate thicknesses (0.47 in., 0.63 in., 0.79 in., 1.0 in., 1.18 in.). The connection tested was similar to that of Type 1 found in Fig. 3-1(a), with two anchor bolts on either side of the web of an I-section. A grout layer was provided between the base plate and the concrete foundation, which is typical in actual field conditions. The detail of the testing apparatus used by Hon and Melchers (1988) can be found in Fig. 3-4. The moment was applied by the loading jack with an

eccentricity and the rotations were measured by an inclinometer, see Fig. 3-4. A moment rotation plot was developed by plotted the measured rotations against the applied moment, see Fig. 3-5. The plots of moment vs. rotation in Fig. 3-5 are nonlinear in nature due to the connection yielding. Yielding in the connection can come from a number of places. The bolts can yield under the applied stresses, the plate can yield in bending, or the concrete or grout could even begin to crush if the forces are high enough.

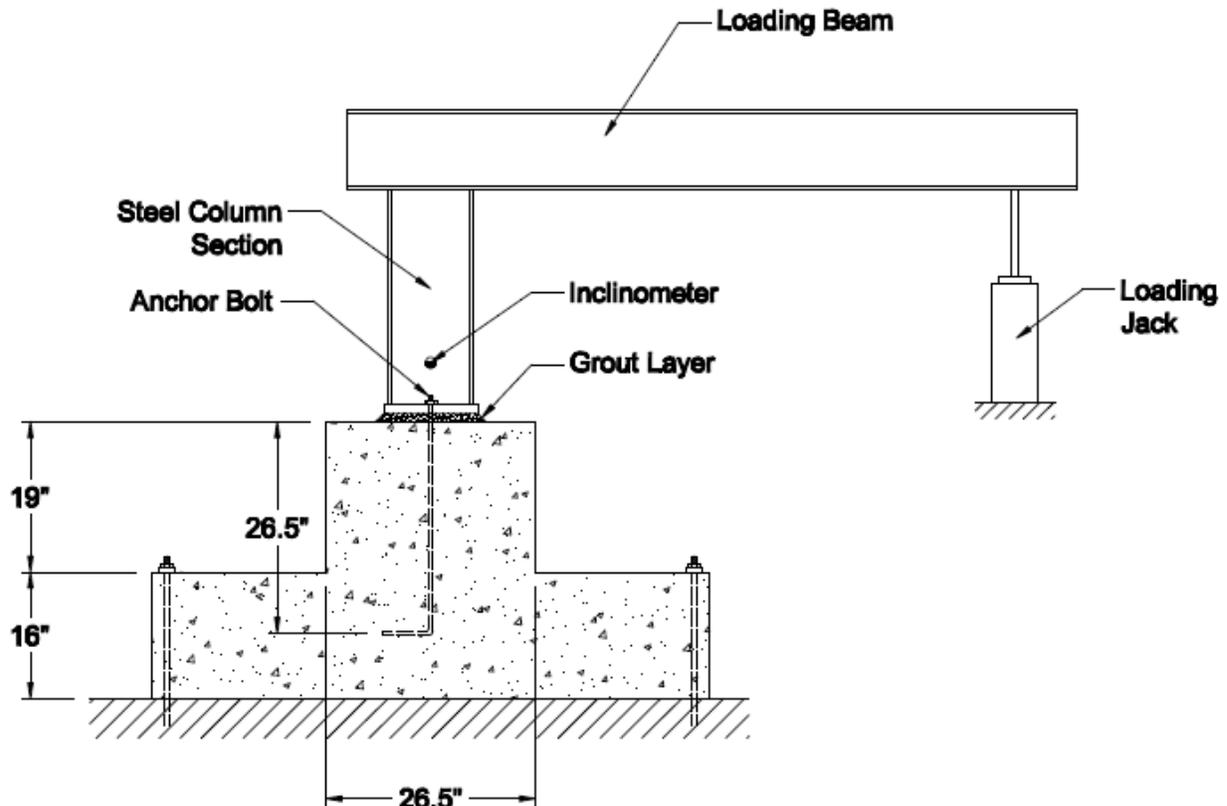


Figure 3- 4: Testing Apparatus (Hon and Melchers (1988))

After testing, it was determined that the connection behavior was most dependent upon the plate thickness. Intuitively, this makes sense because most of the bending is taken by the plate, given that the bolts are designed to handle the prying action. Connections with thicker plates were dominated by bolt failure, and tended to be somewhat brittle (Hon and Melchers 1988). Therefore, it can be stated that the thicker the plate, the stiffer the connection, notice from Fig. 3-5(a) the increase in stiffness between the 0.63 in. and 1.0 in. thick plate. Additionally, increasing the plate plan dimensions with a given thickness also showed a slight increase in the stiffness of

the connection. This was due to the increase in the moment arm from the larger compression block in the concrete.

For connections where the bolt diameter was varied, no significant increase in stiffness can be accounted for. The slope of the moment rotation plots for both the 0.95 in. diameter and 0.79 in. diameter bolts is virtually identical, see Fig. 3-5(b). When looking at column sizes, there is a small benefit in increasing the size or weight of the column if plate bending controls. If the column weight is increased while keeping the depth of the column constant, there is a greater stiffness. If the column depth is increased but the weight of the column is kept approximately equal, the stiffness increases due to a larger lever arm from the bolt line in tension and column flange in compression. Notice the increase in stiffness from the W12x30 to the W18x50 I-section in Fig. 3-5(c). It was also noticed that the grout layer did not exhibit any crushing during testing and therefore not as critical an element with regard to stiffness as expected (Hon and Melchers (1988)). As mentioned previously, the plate thickness has the greatest effect on the rotation stiffness of the connection, while the column and bolt size remain the same. Refer to Hon and Melchers (1988) for a more in depth study of the effects of the various components.

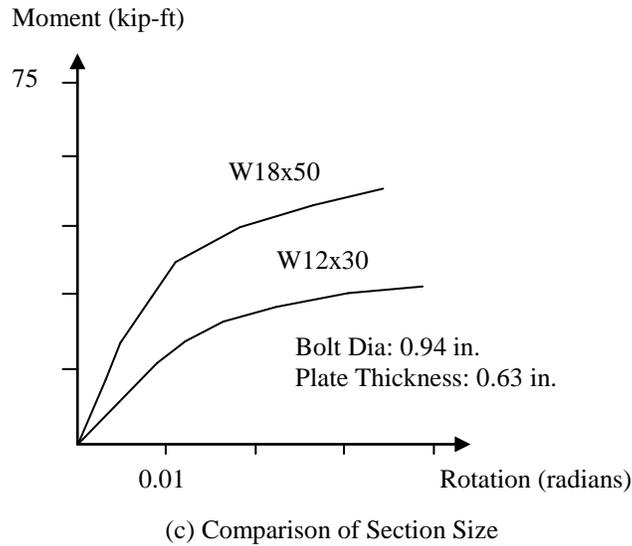
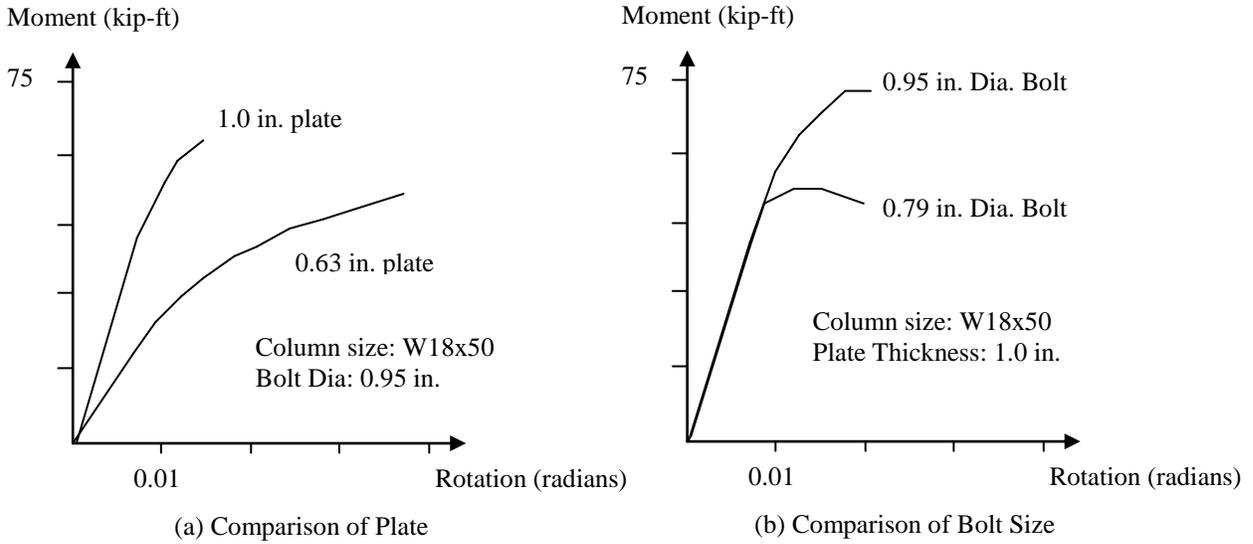


Figure 3- 5: Comparison of Moment Rotation Curves (Hon and Melchers 1988)

3.5 TESTING AND MODELING PROCEDURES FOR ROTATIONAL STIFFNESS

This section discusses the testing and modeling done by Picard and Beaulieu (1985) and Hamizi and Hannachi (2007) on the connections shown in Fig. 3-1. The experimental testing on the connections was done in 1985 by Picard and Beaulieu (Picard and Beaulieu 1985). They conducted a series of tests loading the column in flexure, axial and shear with a variety of

different loading and eccentricities to develop the moment rotation plot, further developing the factors which influence the rotational stiffness of the column base connection. By testing connections varying the bolt layout and loading applied to the connection, they determined the rotational stiffness is drastically increased with a four bolt layout as well as the presence of increased axial loading. Hamizi and Hannachi (2007) developed another finite element model including the interaction between the steel plate and the concrete foundation. This section will discuss the testing procedures for the above experimental tests as well as the finite element model used by Hamizi and Hannachi (2007).

3.5.1 TESTING PROCEDURE

There were two different setups for the tests, one for the connection experiencing bending and shear, then another for bending and axial load, see Fig. 3-6. Each test apparatus had a steel column file welded to a steel plate which was bolted into a concrete foundation by means of an embedded steel bearing plate, with a layer of cement grout in between. The concrete foundation was then anchored to the ground. This was done to ensure that the anchor bolts would develop full tensile capacity, preventing the connection failing in concrete crushing. The anchor bolt and concrete foundation details can be found in Fig. 3-3.

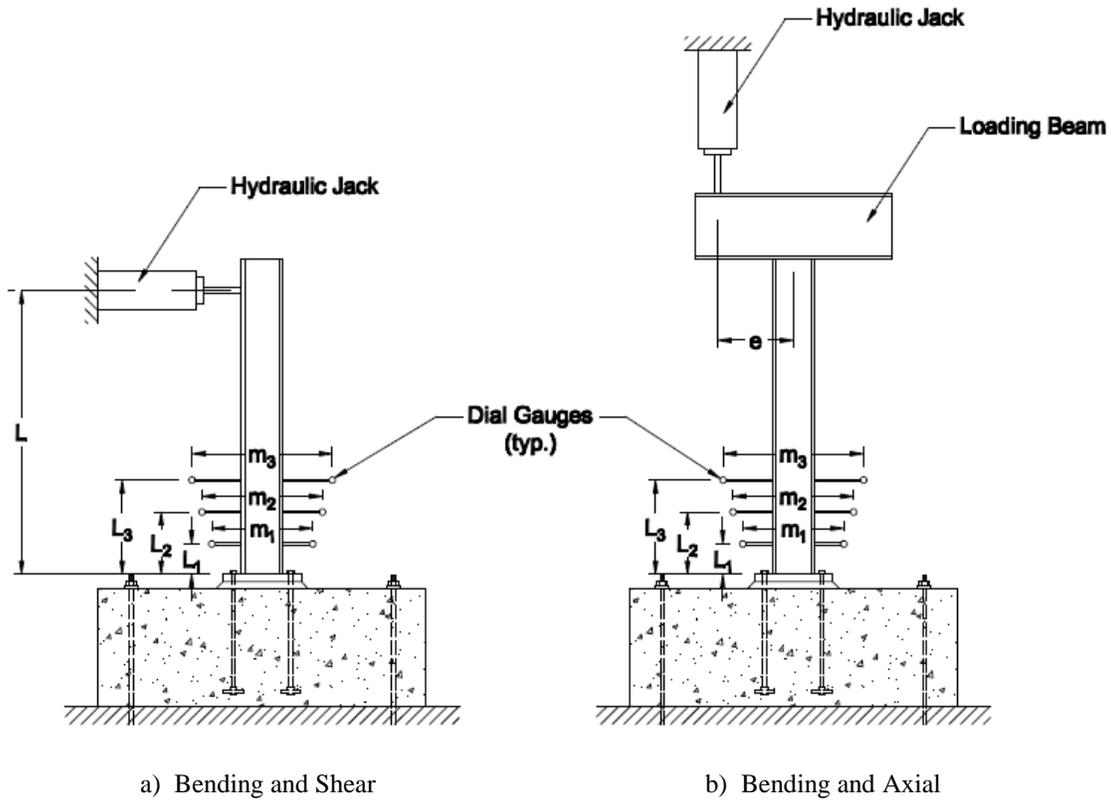


Figure 3- 6: Test Configuration (Picard and Beaulieu 1985)

The testing apparatus for the connection in bending and shear consisted of a cantilever column, connected as described above, and loaded with a hydraulic jack. The jack loaded the column with a concentrated force transverse to its major axis, creating a moment at the base of $P \times L$, with the absence of axial loading, see Fig. 3-6(a). The force P was varied in order to produce the different moments needed for the moment rotation curve. The other apparatus (Fig. 3-6(b)) was set up with a similar cantilever column, but had a loading beam bolted to the top of the column with stiffeners to avoid web crippling in the column. In this case, the hydraulic jack loaded the loading beam with a concentrated force applied transverse to its major axis, see Fig. 3-6(b). This created a moment at the base of the column of $P \times e$, where e is the eccentricity from the applied load to the centerline of the column, with an axial load of P experienced by the column. The load, P , was varied to produce the different moments for the moment rotation curve including

axial load. The article by Picard and Beaulieu (1985) included tests on both of the apparatuses listed above.

The rotations were determined with a series of dial gauges located near the base of the column, each gauge had another one located directly across from it, see Fig. 3-6. The rotation was calculated by dividing the absolute vertical distance between two adjacent dial gauges by the horizontal distance between the two:

$$\theta = \frac{D_{L,i} + D_{R,i}}{m_i} \quad (3.1)$$

where $D_{L,i}$ and $D_{R,i}$ are the absolute value of the difference between the initial dial gauge reading and after rotation at location i for the left and right gauges respectively; m_i is the horizontal distance between the two dial gauges at location i . The rotations were averaged from the three dial gauges before being plotted. The averaged rotations were then plotted against their respective applied moments to obtain the moment rotation diagram.

3.5.2 FINITE ELEMENT MODEL

The finite element model developed by Hamizi and Hannachi (2007) was designed to mimic the behavior of the experiment tests as explained in Section 3.5.1. The model created by Hamizi and Hannachi (2007) took into account the nonlinear interaction between the base plate and grout layer. This was done by using the frictional coefficient between the two surfaces and Coulomb's law for contact between a deformable body and a rigid body, stating that the contact force may be broken into a normal component and a tangential component. In other words, the contact surface between the base plate and concrete foundation was modeled with a normal and a tangential nonlinear stress strain relationship. This way the nodes connecting the base plate with the concrete foundation would behave in a nonlinear fashion instead of being rigidly connected together. For development of the nonlinear stress strain relationships for contact surface between the base plate and the concrete foundation used in their model refer to Hamizi and Hannachi

(2007). A diagram of the model built in Hamizi and Hannachi (2007) may be found in Fig. 3-7. Additionally, in each case the concrete soil interaction was neglected.

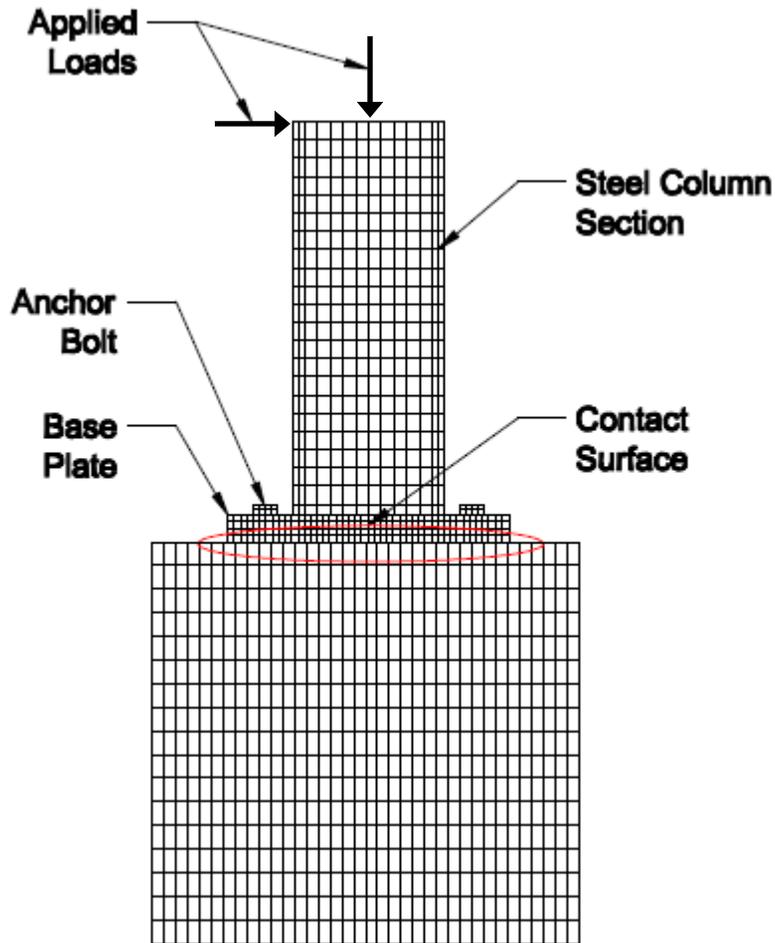


Figure 3- 7: Finite Element Model (Hamizi and Hannachi 2007)

The model was set up with a stub column connected to the base plate, which was bolted to the concrete foundation. All nodes coincided where surfaces were to be in contact with each other. Anchor bolts were modeled as square sections for simplicity, having the same effective cross-sectional area as its respective circular bolt. The anchor bolts were modeled in direct contact with the base plate, not extending down into the concrete. Modeling the bolts in this manner may not

properly account for their behavior since they are not in direct contact with the concrete. In order to produce bending and shear at the connection, a horizontal load was applied at given height from the base. The horizontal loading was applied in increments to produce a moment rotation plot for the connection. In addition to the horizontal loading, for models where an axial load was present, a point load was applied in the vertical down direction at the top of the column. The stiffness associated with the soil was ignored. The rotations were determined at the center of the base of the column and plotted against the applied moment. After the moment rotation plots were developed for all connections, the results were then compared with work done in 1981 at Laval University in Quebec, Canada (Picard and Dion 1981).

3.6 RESULTS – DETERMINING ROTATION STIFFNESS

This section includes the results from the previous sections as well as methods for calculating the rotational stiffness of the connection from the moment rotation plots and other analytical procedures. As mentioned previously, the results will focus on the testing done on I-sections since this more closely represents the connection in a metal building.

3.6.1 ANALYTICAL PROCEDURE

To determine the stiffness of the column base connection, Galambos (1960) developed two analytical methods. The first method is based on the footing not rotating in the soil, thus only taking into account the deformation occurring in the plate, anchor bolts, and the concrete in the footing. For a diagram of the column-base connection see Fig. 3-8. Analytical approximations based on work done by Salmon et al. (1957) and referenced in Galambos (1960) have shown that the stiffness based on this assumption is equal to:

$$K_{con} = \frac{M}{\theta} = \frac{bd^2E_C}{12} \quad (3.2)$$

where M is the moment at the base of the column, θ is the rotation at the base of the column, b is the width of the base plate, d is the length of the base plate, and E_C is the modulus of Elasticity

of the concrete. This equal does not take into account either of the main contributing factors to the rotational stiffness of the connection as discussed in Section 3.4 (plate thickness and column depth). This equation also does not take into account the size or layout of the anchor bolts, which will be discussed in a later section, and is also a large contributing factor to stiffness of the connection. Therefore, careful consideration should be taken when using this equation.

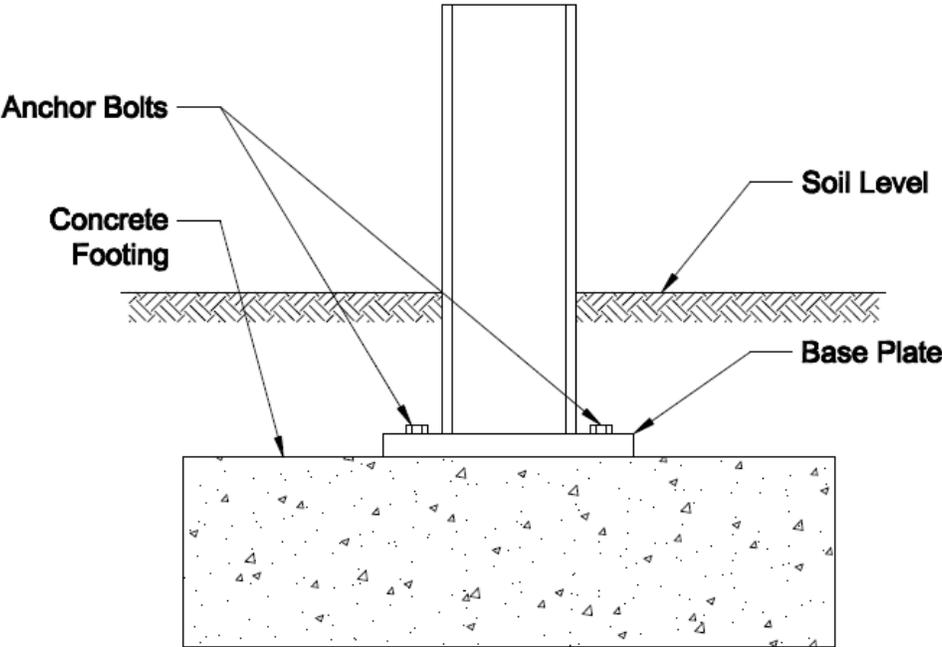


Figure 3- 8: Column-Base Connection Diagram

The other method is based on the footing acting as a rigid beam and the soil behaving as a series of elastic springs, where the rotational stiffness is equal to:

$$K_{soil} = \frac{M}{\theta} = \frac{qgf^3}{12} \tag{3.3}$$

where q is the modulus of subgrade reaction, g is the width of the concrete footing, and f is the length of the concrete footing. This equation assumes that the column base connection is completely rigid and therefore not contributing to the stiffness.

The two methods for calculating the column-base connection stiffness are then used to calculate a coefficient, λ , for determining the buckling strength of a frame using that particular connection. The buckling strength of frames due to partially restrained base connections will be discussed in further detail in Section 3.6. The stiffness of the connection used for calculating λ should be taken as the smaller of Eq. (3.2) and Eq. (3.3) (Galambos 1960). In reality, the stiffness of the column-base should be a combination of the soil stiffness (K_{soil}) and the connection stiffness (K_{con}) added together in series. The method described in Galambos (1960) relies on taking the minimum of the two contributions since the equations are based on two separate assumptions, the footing not rotating in the soil and the connection behaving as a rigid connection.

3.6.2 EXPERIMENTAL RESULTS

The results from the testing performed on connections Type 1 and Type 2 from

Fig. 3-1, in Picard and Beaulieu (1985) will be discussed in this section. For connection Type 1, with plate dimensions 5.12 x 5.5 x 0.43 in., M4 x 12.5 column the average rotational stiffness after multiple tests was equal to 449 kip-ft/radian, see Fig. 3-9(a). For connection Type 2, with plate dimensions 7.5 x 11.8 x 1.14 in., W6 x 25 column, the average rotational stiffness after multiple tests was equal to 4853 kip-ft/radian, see Fig. 3-9(b).

The plots in Fig. 3-9 were copied from Picard and Beaulieu (1985) and no data was presented for the points on the curve. The stiffness for the connections found in Fig. 3-9(a) and Fig. 3-9(b) were developed on the basis that the theoretical stiffness is constant in the elastic region. To calculate the stiffness, all of the data points up to 70 percent of the maximum moment were used. A linear regression analysis was performed on those data points to determine a stiffness for a given test. An average was then taken for the series of tests on each connection to determine the

stiffness drawn on the plots. This method of calculating the rotational stiffness is conservative since most structures do not experience loads that close to their ultimate strength. In addition, for serviceability limit states such as wind design, the loads experienced by a structure are far below 70 percent of the ultimate loads. If the stiffness was developed based on the region where the connection experiences serviceability loading, the resulting stiffness would be much higher. The reader should note that higher stiffness values could be used for serviceability limit states if proper data is present.

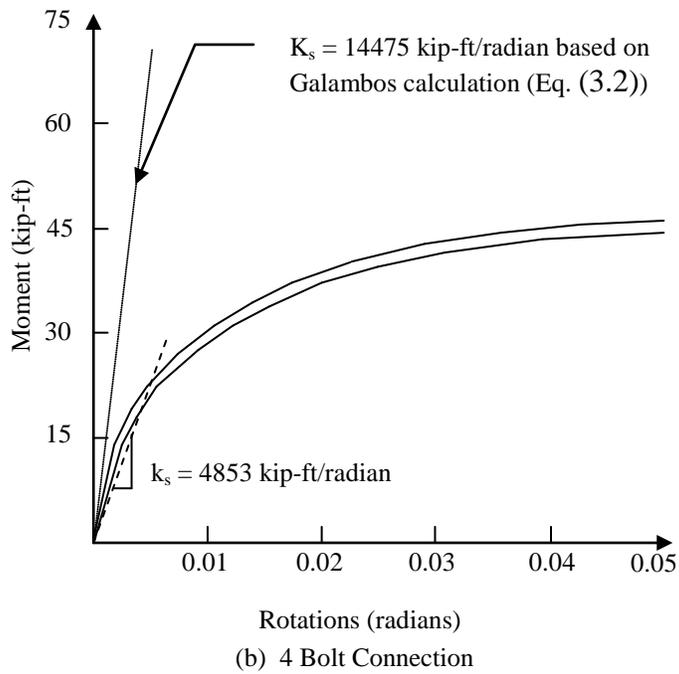
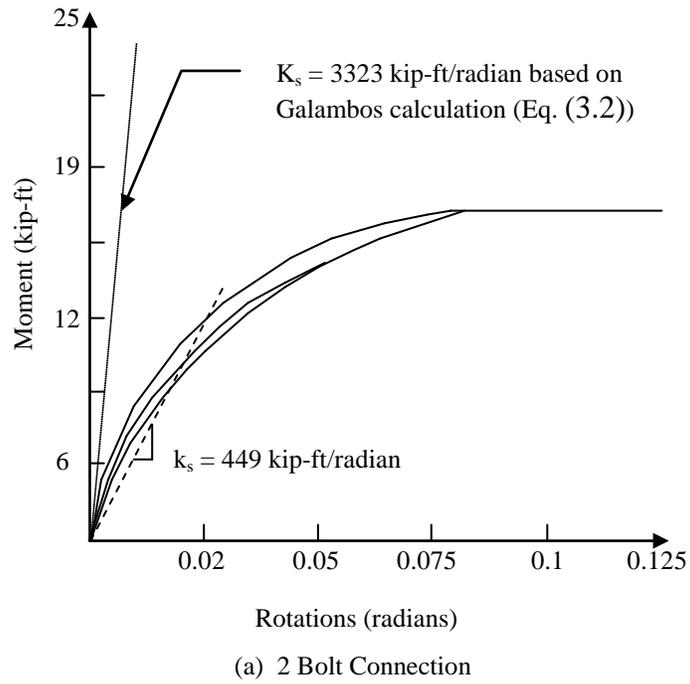
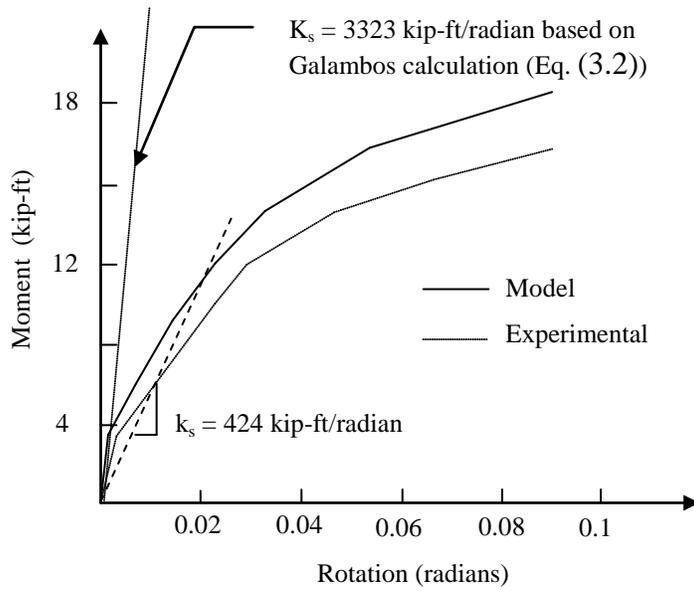


Figure 3- 9: Moment Rotation Plots (Picard and Beaulieu 1985)

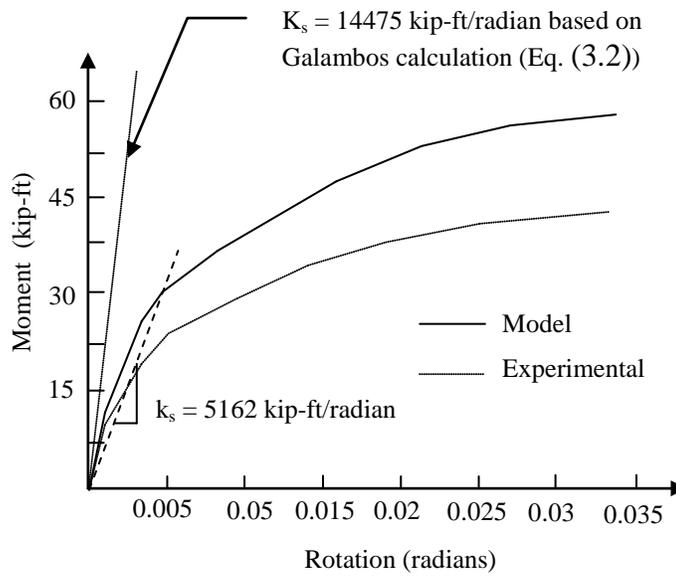
Because the initial stiffness is the primary stiffness of concern for wind drift design, the initial stiffness for each connection was calculated by the author based on the initial slope of the line. The stiffness for the 2 bolt connection would equal approximately 1600 kip-ft/radian and the stiffness for the 4 bolt connection would equal approximately 14000 kip-ft/radian. Notice in Fig. 3-9(b) how the stiffness based on the method present by Galambos (1960) much more closely represents the initial stiffness of the connections. The stiffness calculated using the Galambos (1960) approach does not represent the stiffness of the 2 bolt connection as closely.

3.6.3 FINITE ELEMENT RESULTS

The results from the finite element analysis performed by Hamizi and Hannachi (2007) was plotted against tests done by Picard in 1981 (Picard and Beaulieu 1981). For connection Type 1, with similar plate and section properties as in Picard and Beaulieu (1985), the approximate rotational stiffness was equal to 424 kip-ft/radian, see Fig. 3-10(a). For connection Type 2, with similar plate and section properties as in Picard and Beaulieu (1985), the approximate rotational stiffness was equal to 5162 kip-ft/radian, see Fig. 3-10(b). The stiffness for the connections analyzed in Hamizi and Hannachi (2007) were not calculated, therefore an approach similar to that used in Picard and Beaulieu (1985) was replicated to the best of the authors ability using the plots.



(a) 2 Bolt Connection



(b) 4 Bolted Connection

Figure 3- 10: Moment Rotation Plots (Hamizi and Hannachi 2007)

If a similar approach is used for calculating the initial stiffness of the Hamizi and Hannachi plots, the stiffness for the two bolt connection would equal 1800 kip-ft/radian and the stiffness for the four bolt connection would equal 12000 kip-ft/radian. The initial rotational stiffness developed for both connection types were similar based on the work done by Picard and Beaulieu (1985) and Hamizi and Hannachi (2007), see Table 3-1. As mentioned previously, the stiffness calculation based on the method presented in Galambos (1960) was close to the initial stiffness on the moment rotation plots for the 4 bolt connection, but not for the 2 bolt connection. This shows that Eq. (3.2) is not consistent when predicting the rotational stiffness for base column connections. This is likely because the method presented in Galambos (1960) does not take into account the thickness of the plate, which was shown to be the most influential in Section 3.4.

Table 3- 1: Analytical Stiffness Comparison

Connection Type	Initial Stiffness (kip-ft/radian)		
	Picard and Beaulieu (1985)	Hamizi and Hannachi (2007)	Galambos (1960) eq. (3.2)
Type 1 (2 Bolt)	1600	1800	3323
Type 2 (4 Bolt)	14000	12000	14475

3.6.4 EFFECT OF AXIAL LOAD ON ROTATIONAL STIFFNESS

There was a large increase in rotational stiffness with the presence of an axial load on the column. The results in this section refer to the work done by Hamizi and Hannachi (2007) on the 4 bolted connection. Notice in Fig. 3-11 that the rotational stiffness at the base of the column nearly doubles in the presence of a 23 kip load as compared to the connection with no axial load. Once a load of 90 kips is applied to the column, the connection has a stiffness more than double that of the connection with no axial load applied. It is not likely that a column in a metal building will see a load on the order of magnitude of 90 kips, but it is reasonable to think it could see a load of 23 kips. This is an important behavioral trait that should be considered when modeling these connections. It is also likely that if uplift occurs in the connection due to wind loading (which is likely the case in metal buildings) the stiffness will decrease. This is something that will need to be explored while determining the base connection behavior.

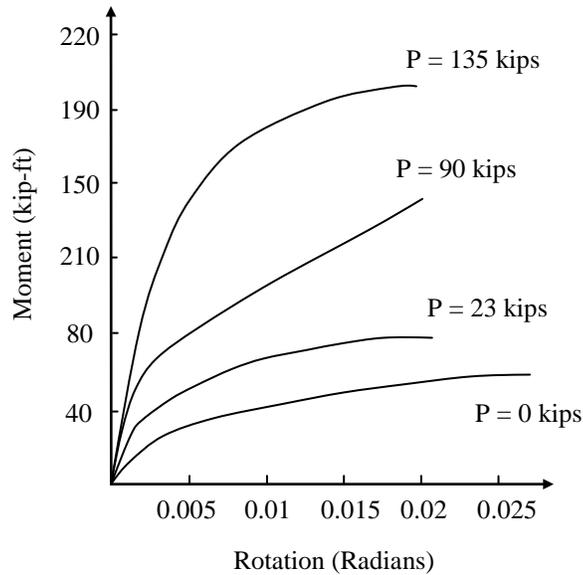


Figure 3- 11: Moment Rotation of 4 Bolted Connection with Axial Load (Hamizi and Hannachi 2007)

3.7 EFFECT ON BUCKLING STRENGTH

The buckling strength of the columns in metal buildings is beyond the scope of this research, but a brief overview of the research done by Galambos in 1960 (Galambos 1960) on the effect of partial base fixity on the buckling strength of frames, is included to be thorough. For more information on this subject see Galambos (1960).

The buckling strength of columns in a fixed based frame can be four to five times that of those in a pinned base frame, see Fig. 3-12. It can be shown that by including a small amount of rotational restraint (base fixity, λ , equal to 1.0, discussed later in this section) in the base of a frame produces nearly the same effect on the buckling strength of the columns as having a fixed base.

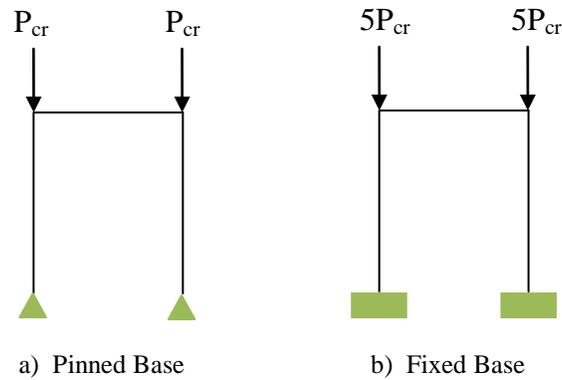


Figure 3- 12: Buckling Strength of Frames

A columns critical elastic buckling load, P_{cr} , is calculated as follows:

$$P_{cr} = \frac{\pi^2 EI}{(kL_c)^2} \quad (3.4)$$

where E is the modulus of elasticity for the column, I is the moment of inertia for the column, L_c is the length of the column, and k is the effective length factor for the column. Therefore, the lower the effective length factor for the column, the higher the critical elastic buckling load of the column. This is how increasing the restraint at the column base connection increases the buckling strength of columns. As the restraint provided by the column base connection increases, the effective length decreases, in turn increasing the buckling load for the column. For an illustration of this, refer to Fig. 3-13.

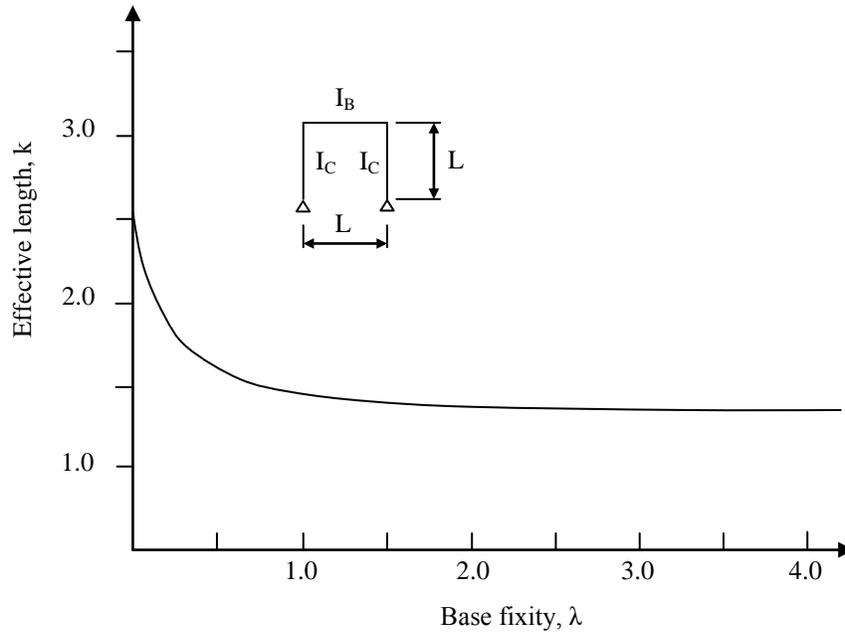


Figure 3- 13: Effective Length in Columns of a Single – Story Single – Bay Frame (Galambos 1960)

where λ is a ratio of the stiffness of the base connection compared to the stiffness in the column. The ratio, λ , may be calculated as follows:

$$\lambda = \frac{I_S/L_B}{I_C/L_C} \quad (3.5)$$

where L_B and L_C are the lengths of the column and beam in the frame respectively; I_C is the moment of inertia of the column; and I_S is the moment of inertia of a fictitious beam spanning from the base of each column. The fictitious beam is used to represent the rotation stiffness of the base connections. The ratio of I_S/L_B may be related to the moment rotation relationship of the connection determined from Eq. (3.2) or Eq. (3.3). This relationship is equal to:

$$\frac{I_S}{L_B} = \frac{bd^2}{72n} \quad (3.6)$$

if Eq. (3.2) controls, where b is the width of the base plate, d is the length of the base plate, and n is the ratio of steel modulus of elasticity to concrete modulus of elasticity. Or I_S/L_B is equal to:

$$\frac{I_S}{L_B} = \frac{qgf^3}{2160 \times 10^6} \quad (3.7)$$

if Eq. (3.3) controls, where q is the modulus of subgrade reaction, g is the width of the concrete footing, and f is the length of the concrete footing.

Notice in Fig. 3-13, once λ equals 1.0, the effective length factor for the columns are nearly the same as that for a frame with a fixed base. The research in this chapter has shown that including partial base fixity is a more accurate method of modeling the behavior of the base restraint and should be taken into account when calculating the buckling load of the columns.

3.8 CONCLUSIONS

Based on the information collected in the literature review contained in this section, it can be concluded that there is sufficient initial stiffness in column base connection to be included in analysis. At this point, this statement can only be applied to the connection types discussed in this chapter. Because the researchers did not calculate the initial stiffness of the connection, there is room for more research to be conducted on the connection's behavior. Also, the researchers did not explicitly explore the affect of the grout layer on the connection stiffness. This is an area that should be explored further to better understand its behavior with regard to stiffness. In addition, the analytical method presented in Galambos (1960) does not take into account the factors discussed in Section 3.4, which have the most affect on the connection stiffness. This method may be used to obtain an estimate of the initial stiffness of a given connection type, but should not be used when modeling the connection stiffness. In conclusion, there is adequate initial stiffness provided by the column base connection types analyzed to be included in modeling. With this in mind, further research should be performed to validate the stiffness value to use when modeling.

CHAPTER 4: DISPAR

The program known as DISPAR is a program designed to assist practicing engineers when designing the lateral resisting systems in buildings. When designing the lateral system in a building, it can be difficult to determine which members have the greatest impact on the stiffness of the system. Valuable time may be wasted by resizing members and noticing that there has been little improvement on the stiffness of the system. DISPAR helps with this very problem, giving an engineer the tools and information to optimize the stiffness of a system while limiting the volume of material. It uses virtual work to develop DISplacement PARTICipation factors (DISPAR Factors) for each member. The DISPAR factors for each member are then divided by its respective volume, the result of which is referred to as a sensitivity index (*SI*) for that member. A member's *SI* value is a weighted measure of how sensitive that member is to change with relation to overall drift (see Section 2.3.3 for a discussion on *SI* values). In other words, increasing the size of a member with a higher *SI* will decrease the drift more than increasing the size of a member with a lower *SI*. By using the DISPAR program, a design engineer is able to effectively size the members in a structure without arbitrarily selecting a member and seeing how it affects the system.

4.1 DISPAR VERSION 5.4

The original DISPAR program was created and developed by Dr. Finley A. Charney of Advanced Structural Concepts of Golden, Colorado (Charney 1995). It was created to work alongside SAP90 and ETABS and included a variety of features allowing an engineer to effectively resize and reanalyze a given structure to optimize stiffness per volume. These features included a graphical representation of each member's contribution to the total drift, which could be viewed in terms of axial, flexural, and shear components of deformation. Member contributions could also be viewed in terms of their total DISPAR factor, *SI*'s, and group information. In addition, tables could be viewed which included the DISPAR information for various groups. These groups included, beams, columns, diagonals, panels, and user defined. There were also four options for handling the beam-column joint region, including the effect of doubler plates. Also, it had the ability to evaluate the effects of cracking in concrete and

composite action in steel structures. Another useful feature was an approximate, yet very accurate, method of reanalysis which was much faster than re-analyzing using SAP or ETABS. In order to view the structure and member contributions a graphics package called DISPLOT™ was provided, which allowed for 3D color plots. For the calculation of drift contribution from the panel zones, a separate utility called PANELS™ was included, which was very accurate when compared with detailed finite element analyses (Charney 1995). These features combined into a program which gave an engineer the information necessary to understand the behavior of the members in a given structure.

4.1.1 JOINT OPTIONS

The joint options in DISPAR take into account the flexibility in the beam column joint region. This is the region where the beams and columns frame into one another, forming a joint where the width is the depth of the column and the height is the depth of the beam. There are multiple ways to handle the effect in this region, which include assuming full flexibility of the beams and columns, full rigidity of beams and columns in the region, or somewhere between the two extremes. DISPAR's methods are based on the ETABS variable NRGD, which allows the user to set the rigidity of beams and columns in the joint region. See Table 4-1 for a description of the NRGD variable.

Table 4- 1: ETABS NRGD Variable

NRGD	z	Description
0	0	No offset, referred to as clearspan analysis
1	0.25	25% offset of the rigid zone
2	0.5	50% offset of the rigid zone
3	0.75	75% offset of the rigid zone
4	1.0	100% offset of the rigid zone, referred to as centerline analysis

There are four different options in DISPAR on handling this region. The first is the ETABS joint option, where deformations are calculated based on a force distribution used in ETABS. The next is the rigid joint option, where deformations in the joint region are assumed to be zero, no matter what the NRGD value is. The third method is to use a centerline analysis, which is the equivalent

of having a NRGD value of 4. The last method is the flexible joint option, which uses the method discussed in Section 2.3.4 to calculate the deformations in the region. All of the methods are based on member forces computed using the modeling approach employed by ETABS. For diagrams of the force distributions see Fig. 4-1 through Fig. 4-4. Typical design practice is to assume a centerline analysis in which the flexural deformations are typically overestimated and shear deformations underestimated. The errors produced by a centerline analysis tend to offset one another (Berding 2006). The ETABS joint option and flexible joint option may also be used to incorporate the behavior in the beam-column joint region. Also, it is important to note that assuming rigid joint behavior typically produces unconservative results.

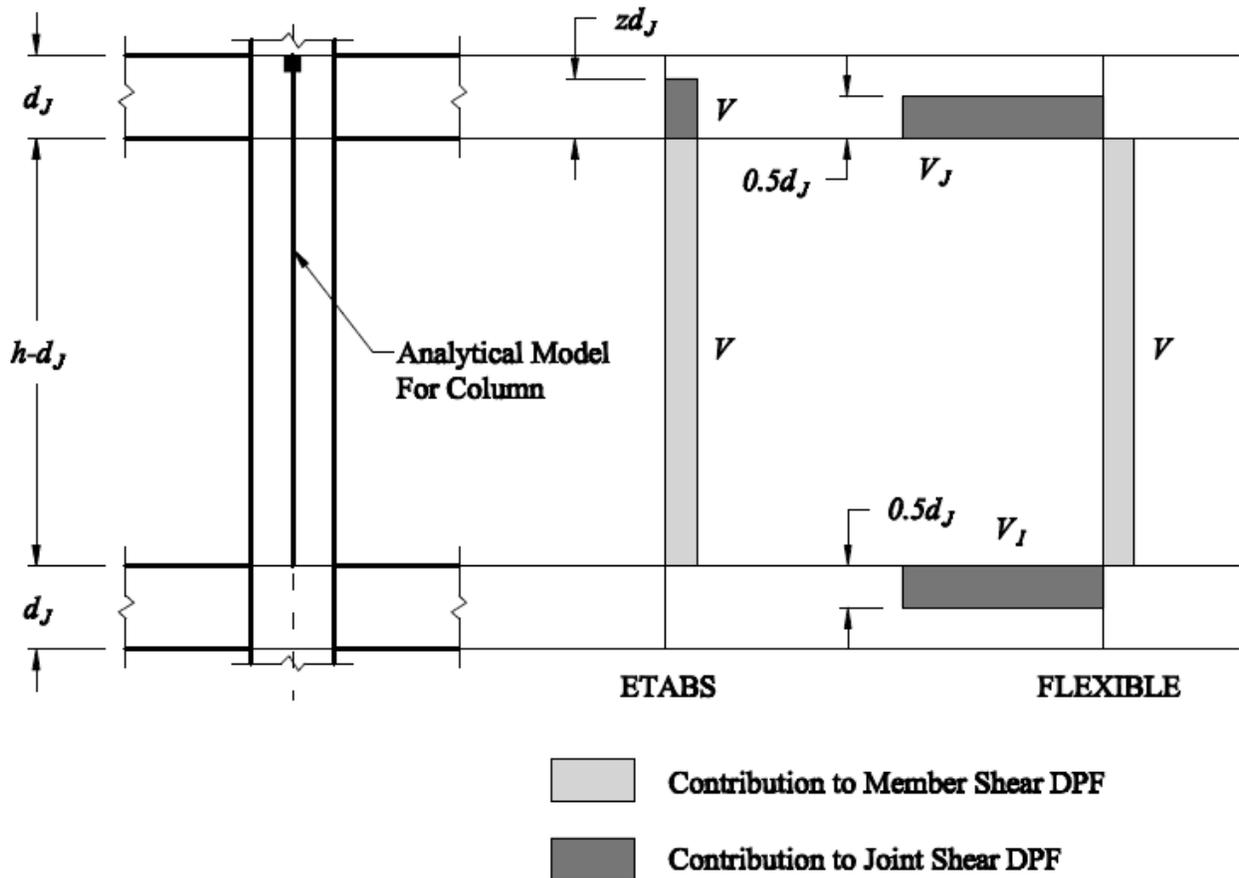


Figure 4- 1: Column Joint Options for Shear (Charney 1995)

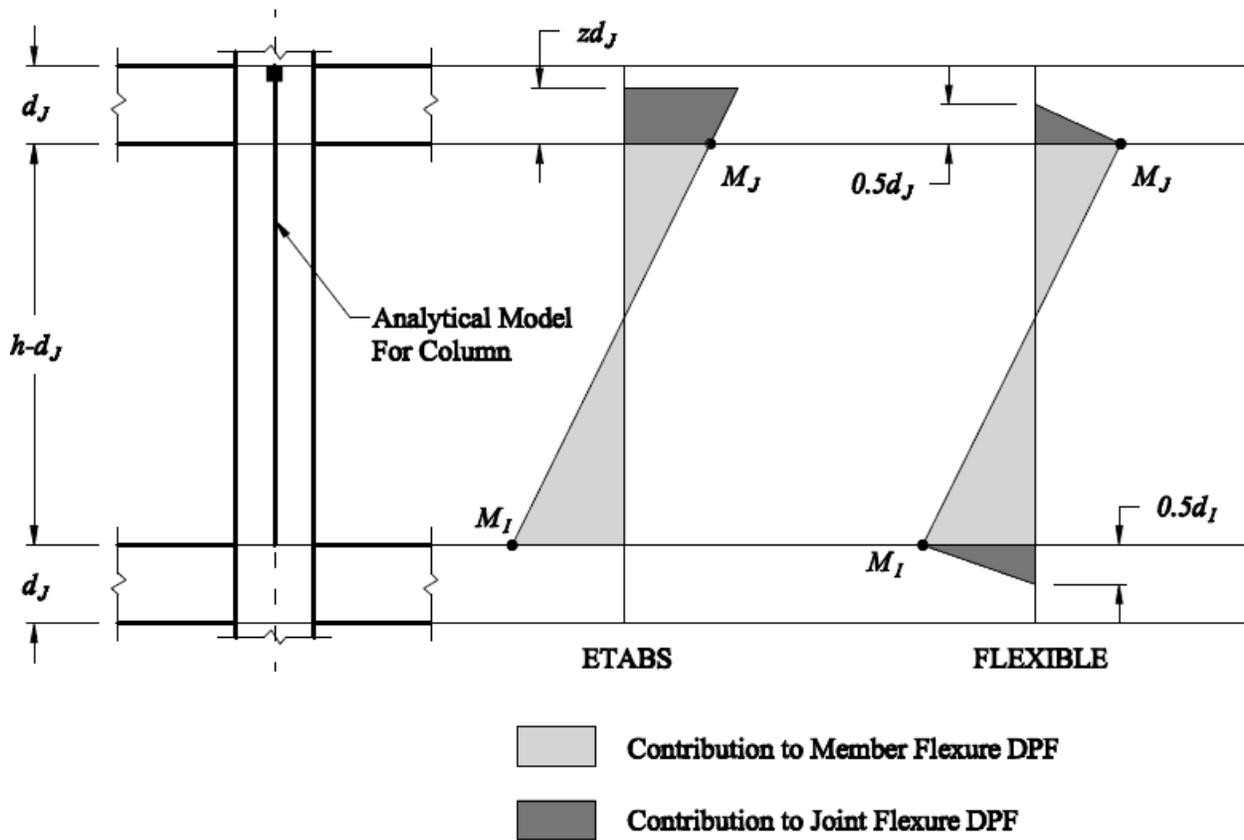


Figure 4- 2: Column Joint Options for Flexure (Charney 1995)

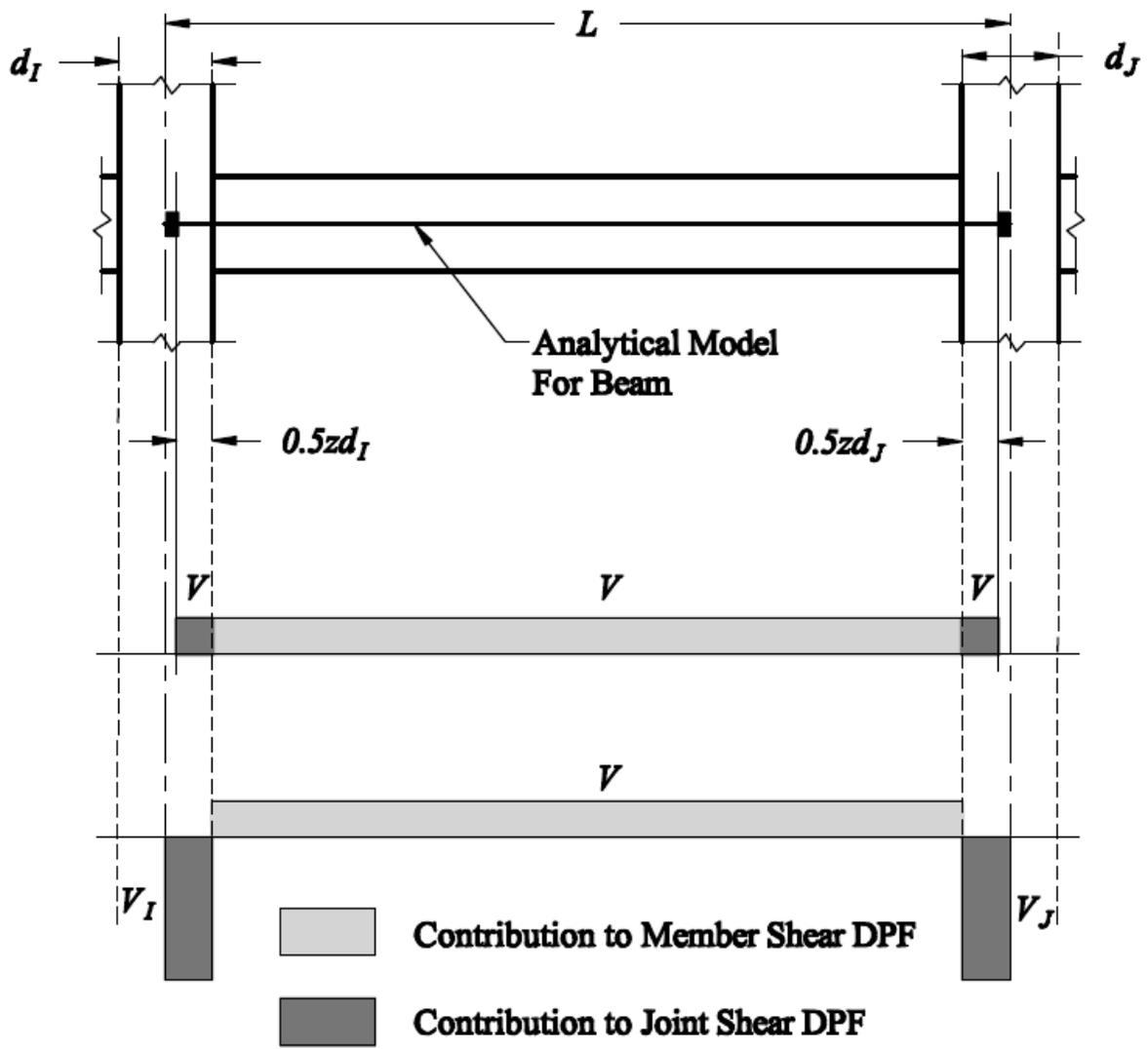


Figure 4- 3: Beam Joint Options for Shear (Charney 1995)

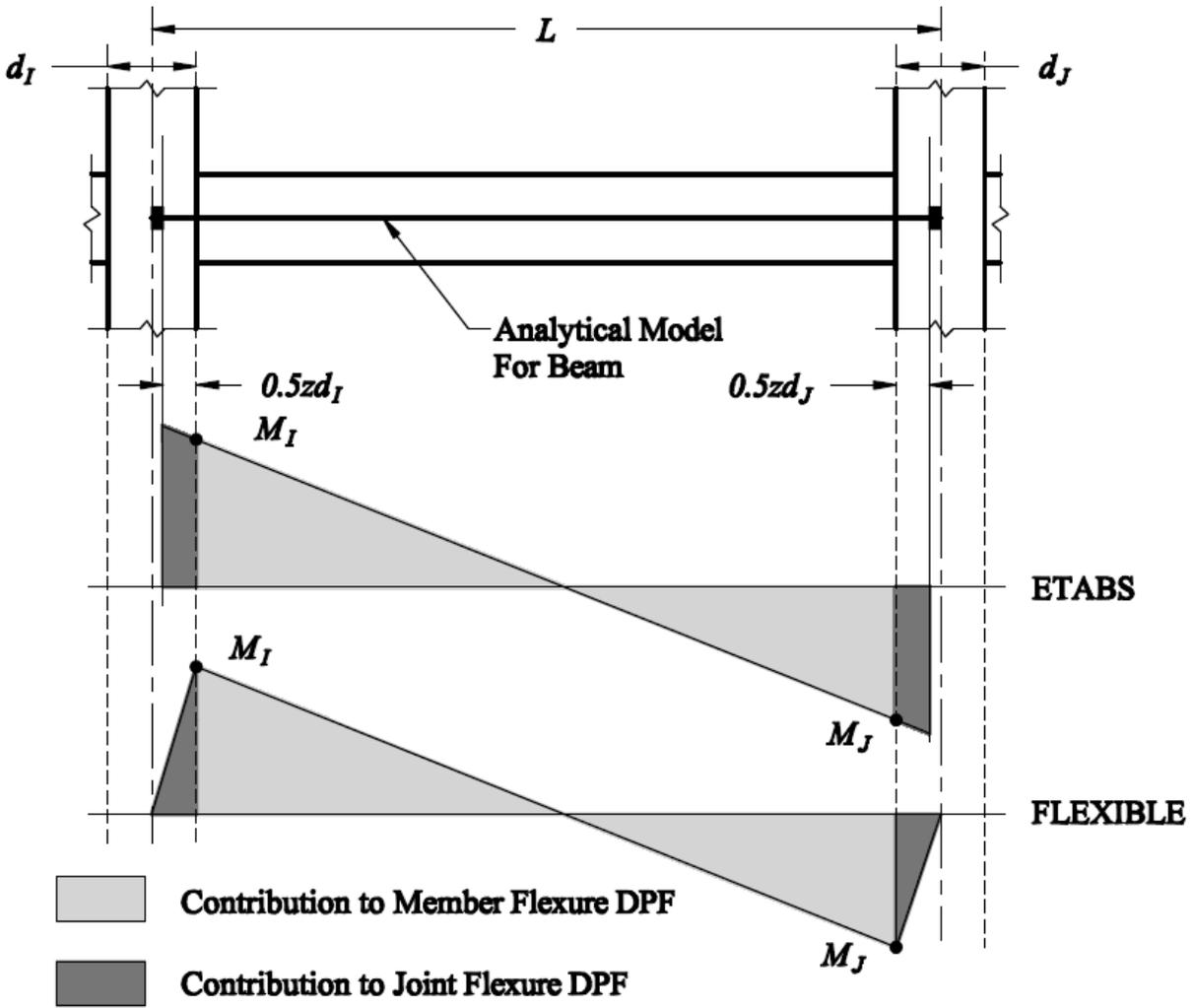


Figure 4- 4: Beam Joint Options for Flexure (Charney 1995)

4.1.2 DPFs FOR SPRING ELEMENTS

Many times joints are not simply fixed or free to rotate and therefore should be modeled using spring elements to capture their stiffness. For this reason, DISPAR was made to handle spring elements by using their stiffness, along with real and virtual displacements, to calculate their displacement participation factor.

$$DPF_{s,i} = drx_i \times dvx_i \times Kx_i + dry_i \times dvy_i \times Ky_i + Qrz_i \times Qvz_i \times Rz_i \quad (4.1)$$

where $DPF_{s,i}$ is the displacement participation factor for spring i , d represents the real or virtual displacement in the x or y direction, and K is the spring stiffness in the respective direction. The Q and R represent the rotation displacement and stiffness in the z direction.

4.1.3 METHOD OF REANALYSIS

The method used by DISPAR for reanalysis is much faster than re-running the structure in ETABS or SAP90. It is based on the idea that the forces in the members in the lateral resisting system are relatively insensitive to change in size. This can also be stated that the system is more heavily controlled by equilibrium rather than compatibility. Given this assumption, the DISPAR factors are recalculated with the original forces in the member, but the updated member properties. The DISPAR factors for all of the members are then summed up to give the total displacement. This method is only able to calculate the updated factors based on the original virtual loading. If the user would like updated factors for a different virtual loading condition the structure will have to be reanalyzed with ETABS. In a study conducted by Charney (1995) on a 20 story frame, 50% of the member sizes were changed by not more than 20%. The displacement calculated using the DISPAR reanalysis technique was essentially equal to that from an ETABS run. This method of reanalysis gives the designer quick and very accurate results on how changes they made affect the stiffness of the structure. The changes made to the stiffness of the structure may also include the effects of concrete cracking.

4.1.4 DISPAR FOR OPTIMIZATION

As mentioned previously, DISPAR has all of the tools to assist an engineer in not only understanding the behavior of their structure, but optimizing its stiffness relative to weight. The components of deformation are broken up into axial, shear, and flexure, informing the designer which property needs to be changed. For instance, if a member's contribution is almost entirely axial, it would do little good to only focus on increasing that member's moment of inertia. Additionally, having a visual representation of the factors and sensitivity indices is very helpful in understanding how the structure behaves as a whole and where member sizes should be increased or conversely where members can be reduced in size. Also, having the ability to create

groups of members can be very beneficial when looking at structures where similar member are used throughout a story. Instead of only viewing how members behave individually, this option helps a designer understand how groups of elements behave. With all of this information, a designer may implement the optimization method described in Section 2.4.1 to optimize the stiffness of their structure. DISPAR does not have an automatic optimization process built in because the author (Charney 1995) wanted to allow for the “human loop”, keeping the designer making the decisions based on experience and engineering judgment.

4.1.5 APPLICATION OF VIRTUAL LOADS

There are a variety of different ways to apply the virtual loads in analysis to understand the behavior of a structure. Each method of applying the virtual loads gives the engineer insight into different modes of behavior. Examples of possible loading conditions may be found in Fig. 4-5, where Q is the applied virtual load. If the loading is applied as found in Fig. 4-5(a), the results will give the designer information on how to optimize the stiffness for the overall drift of the structure. If applied as found in Fig. 4-5(b), it will produce information on the optimizing the stiffness for the interstory drift between the two respective levels. Virtual loads may also be applied at various storys to understand the stiffness of the structure up to that particular story. In addition, if a virtual torque is applied at the center of the structure, as shown in Fig. 4-5(c), an understanding may be gained as to the torsional rigidity of the structure. This can also assist the designer in limiting inherent torsion induced by plan irregularities, which becomes very beneficial in the design in high seismic regions. The possibilities for the virtual load cases are essentially limitless, which gives a designer a very powerful tool in understanding the behavior of all aspects of the structure under consideration.

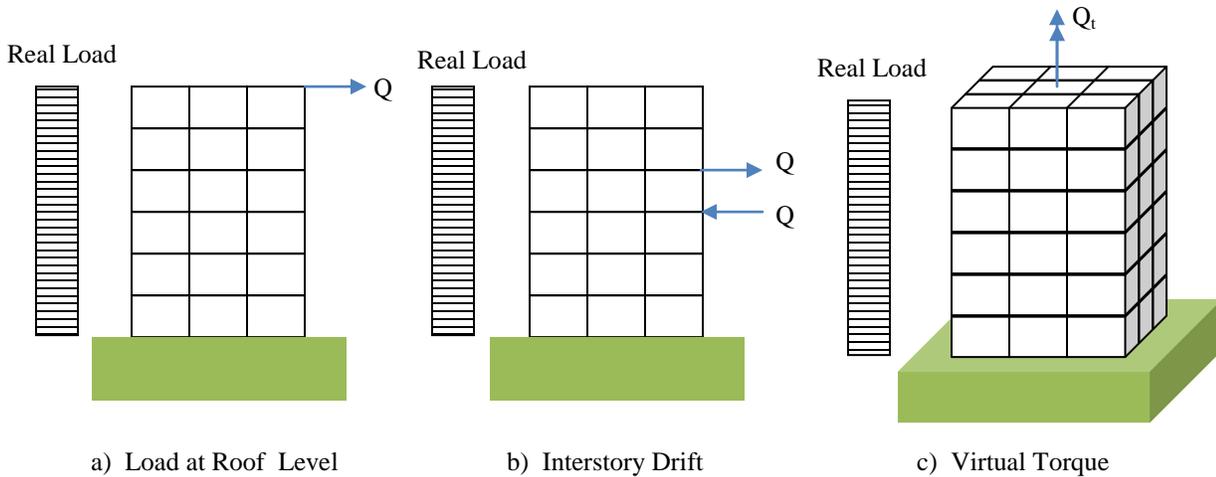


Figure 4- 5: Application of Virtual Loads

4.2 DISPAR FOR SAP2000

The limitations of the previous version of DISPAR discussed in the sections above along with new developments with SAP2000 spurred the development of an updated version. Beginning with version 11 of SAP2000, Computers and Structures included a license to an Open Application Programming Interface (OAPI) for SAP. This allows for a direct link between information in SAP2000 with an external program, eliminating the vast majority of the limitations of the previous version of DISPAR. In addition, a graphics package known as OpenGL was used for viewing capabilities in the new version of the program. OpenGL is a graphics package used for graphics in software games, so its capabilities for this application are virtually limitless. In OpenGL images are drawn using elementary geometric shapes (such as points, lines, and polygons) which combine to make more complex images in a three dimensional world. These images can be colored, shaded, and textured in almost any fashion. This section will discuss how SAP's OAPI and OpenGL, along with other features, such as the linking of tables and graphs with Microsoft Excel (Microsoft Corporation 2007), have improved the capabilities of DISPAR.

4.2.1 OVERVIEW OF DISPAR FOR SAP2000

In the development of DISPAR for SAP2000, more time and concentration was spent on using SAP's OAPI and OpenGL for data input and graphics rather than continuing the theoretical development. Therefore, features such as the PANELS program and the ability to evaluate the effects of cracked concrete have not been included in this version of DISPAR. The new version does include the ability to handle shell elements in the analysis. This becomes a large benefit when understanding the behavior of elements in a detailed finite element analysis of frames or connections. Additionally, the shell elements can be used for shear walls, outrigger beams, and other elements not typically found in a SAP shapes database. A group can be made for the shell elements making up a member and the DISPAR results for that member will show up in the groups table. The new version still includes the ability to handle the joint deformations based on the flexible joint method described in Section 2.3.4 and is handled in a similar manner to the previous version. The DISPAR factors for individual elements and groups are calculated and presented in a similar manner as done in the original DISPAR. The information is presented with three different options, which are graphical representation, in tables, and using bar plots, see Section 4.2.3 for tables included. The new version also includes the ability to handle uncoupled or couple spring elements, in a slightly different approach from the one used in the previous version. Also, DISPAR for SAP2000 has only been developed to work alongside SAP2000 and not ETABS, this is due to the OAPI only being available for SAP2000. According to the Computers and Structures website (Computers and Structures 2008), an OAPI will be available for ETABS soon. Once an OAPI is available for ETABS, DISPAR can be extended to work alongside both ETABS and SAP2000.

4.2.2 SHELL ELEMENT DPF'S

The DPF's for shell elements are handled with the same principle of a virtual force times a real displacement in a member. In the case of shell elements there are six components of deformation located at each node present for a given shell element. The six components of deformation are a force and rotation in the x, y, and z directions. If looking at a typical 4 – node shell element there are 24 directions that the deformation must be considered. Once the analysis has been run on both the real and virtual load cases, the virtual work quantity for node i in the x direction can be

found as follows. By taking the virtual force experience by the shell element in that direction and multiplying that by the real displacement in that direction.

$$Dfx_{n,i} = fx_{v,i} \times dx_{r,i} \quad (4.2)$$

where Dfx is the virtual work quantity for shell element n ; $fx_{v,i}$ and $dx_{r,i}$ represent the virtual force and the real displacement; x represents the lateral translation quantities in the x direction and i represents the node location. In order to find the total virtual work quantity in the x direction for shell element n , the Dfx components from Eq. (4.2) must be added together for each node.

$$Dfx_n = \sum_{i=1}^N fx_{v,i} \times dx_{r,i} \quad (4.3)$$

where N equals to number for nodes in shell element n . Extending Eq. (4.3) for all components of deformation the total DPF for shell element n can be expressed as follows:

$$DPF_{shl,n} = Dfx + Dfy + Dfz + Dmx + Dmy + Dmz \quad (4.4)$$

where the f and the m represent the force and flexural virtual work quantities. To find the total displacement contributed by the shell elements, it is simply a matter of adding the individual quantities for each shell element together.

$$\delta_{shell} = \sum_{n=1}^{Number\ Elements} DPF_{shl,n} \quad (4.5)$$

This method of calculating the virtual work quantities tends to lump the total member contribution into either one of the six components of deformation. Given this, the shell contributions are only accurate when looking at the total contribution for a given element or group of elements. When using shell elements to form a beam, shear wall, or other member, it is best to group all elements together to understand the behavior of the total member. This will give

the user an idea of how a shear wall, for example, behaves when coupled with other shear walls or lateral load resisting system.

4.2.3 SPRING AND LINK ELEMENTS

Spring elements are handled in a similar manner as the shell elements in DISPAR for SAP2000. The virtual force in the spring element for a given direction is multiplied by the real displacement in that direction to determine the virtual work quantity. The spring elements only occupy one node, therefore only six degrees of freedom must be considered (force and rotation about each axis). This method works for both uncoupled and couple springs because SAP returns forces and displacements for each type of spring in the same way. In addition, this returns the contributions of the spring elements in each of the various degrees freedom. The totals for all of the spring elements in each degree of freedom and the total contribution from spring elements are then tabulated. As mentioned previously, allowing for the use of spring elements can be very beneficial when looking at varying support conditions. Expanding this idea to uncoupled and coupled springs in all six degrees of freedom further enhances this capability. The link element work quantities are calculated in a very similar manner, a virtual force times a real displacement.

4.2.4 USE OF SAP OAPI WITH VB.NET

The main motivation for upgrading DISPAR to SAP2000 was SAP's release of their Open Application Programming Interface (OAPI), which allows for quick access to information in SAP2000. The OAPI provided by SAP2000 consists of a library of functions allowing the design engineer to perform virtually all of the tasks inside of SAP2000 without ever physically opening the program. With the use of the OAPI library, the limitations faced in the original version of DISPAR with regard to data input are no longer present. Since SAP2000 can be run in accordance with DISPAR there is no reason to have a limit on the number of load cases which DISPAR can run. Additionally, member sizes can be changed in real time, making the reanalysis in DISPAR that much faster. The only limitation faced with the new version of DISPAR and interacting with SAP2000 is that all load combinations, member groups, and virtual loads must

be applied to the model before running DISPAR. In other words, the user must have a fully functional and ready to analyze model before running it in DISPAR.

4.2.4.1 STEPS TAKEN USING OAPI

In order to get started, the user must select a model to open. After this is done, DISPAR opens the model selected and begins to gather frame, area, coordinate, group, and load combination information. The status bar located at the bottom left corner of the screen tells the user what model is being opened, followed by what the program is currently doing. By using the OAPI, all of the load combinations found in the model opened by the user are available for analysis. This gives the user the power to run any real and virtual load combination set up in the model without ever closing DISPAR. Once the real and virtual load combinations have been selected for analysis, the OAPI functions run the analysis in SAP. Next, the functions allow for the retrieval of the real and virtual forces and displacements experienced by the frame, shell, and spring elements. These values are then placed in arrays and used to calculate all of the virtual work quantities. The quantities are then made available through viewing or in tables. After the analysis has been run, the user has the option to update the member sizes in groups of elements. Using the OAPI functions, any of the section or area sizes predetermined in SAP, groups of elements can be updated in real time. As soon as a member size has been updated, DISPAR recalculates the virtual work quantities based on its reanalysis method.

With the help SAP's OAPI, the capabilities of DISPAR are now only limited by the extents of what SAP2000 can handle and the developers imagination. At this point, DISPAR allows for real time updates to member sizes and load cases under consideration, but with the use of OAPI this could be extended to almost anything with the model.

4.2.4.2 EXAMPLE OF SAP2000 OAPI

The following is an example of how to create a 3D frame model, apply point loads, and after analysis retrieve the displacements at a given location using SAP's OAPI. This example was

done using VBA in Excel and also includes other OAPI functions such as opening SAP2000 and closing SAP2000, before and after the example is done. To create a 3D space frame given the specifications found in Table 4-2:

Table 4- 2: Frame Input

Type	Value
Number Stories	3
Story Height (ft.)	13
Number Bays - X	3
Bay width (ft.)	30
Number Bays - Y	2
Bay width (ft.)	25

Using the SAP OAPI function:

$$New3dFrame(OpenFrame, 3, 13, 3, 30, 2, 25) \tag{4.6}$$

Will generate the frame:

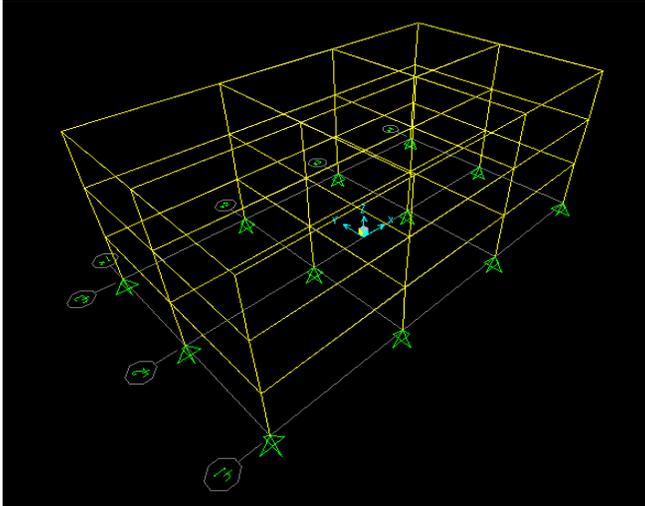


Figure 4- 6: SAP Model

The model generated assumes the default section type in SAP2000 unless specified otherwise. To apply a load of 20 kips in the x direction at a given node, use the SAP OAPI function:

$$\text{SetLoadForce}(\text{node}, \text{"Wind"}, \text{Load}) \quad (4.7)$$

where “Wind” is the load case to apply the load under, and load is a 6 x 1 array where the first term is 20. Each term in the load array represents a force or moment in the x, y and z direction respectively. Apply this function to the nodes at the top level at the frame furthest to the left will results in the model below:

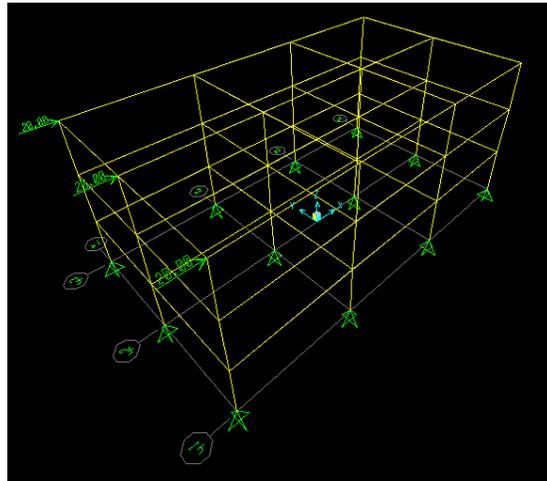


Figure 4- 7: Model with Point Loads

After the model has been analyzed with the above loading conditions, the joint displacements may be returned using the SAP OAPI function:

$$\text{JointDisp}(44, \dots, U1, U2, U3, R1, R2, R3) \quad (4.8)$$

where 44 is the number of the joint located at top level, at the center point of the furthest right frame; U1 through R3 are empty variables which are filled by SAP2000 with their respective translation or rotation; the 1, 2, and 3 terms correlate with x, y, and z. Using the function (4.8)

for the above model given the above loading conditions will return the following results found in Table 4-3:

Table 4- 3: Displacements from Model

Joint	44	
Trans. X	5.938	in.
Trans. Y	0.00	in.
Trans. Z	-0.013	in.
Rot. X	0.00	rad.
Rot. Y	0.005	rad.
Rot. Z	0.00	rad.

4.2.5 OPENGL IN DISPAR

With the use of OpenGL as the graphics package for the new version of DISPAR, the viewing capabilities have been expanded. The model can now be viewed from any angle of rotation about any of the three axes simply by clicking the rotation buttons, with ability to zoom in and out while doing so. Frame lines may be selected for two dimensional viewing. Text features allow for element information to be displayed directly on top of an element, such as group information and member name. In addition, different viewing options may be selected, such as viewing shell elements in the model and a shrunken view of frame elements. Moreover, when a new viewing option or angle of rotation has been selected, the model is automatically regenerated with the current selections.

At this point, the new version of DISPAR is not able to handle creating viewing planes based on point selection, but is certainly possible with OpenGL. This feature would allow the user to select a viewing plane based on three points on the structure. To implement this in DISPAR, the clipping plane and mouse selection features in OpenGL would need to be incorporated. The clipping planes would allow the viewing of only members along the plane selected by the three points. Another feature available in OpenGL which can be implemented in DISPAR viewing is

extruded viewing of elements. The extruded view would allow for the members to be viewed as they are modeled, including angles of rotation. When looking at individual members it could be very helpful to have the ability to select them with the mouse. Doing so would bring up information related to that member as well as allow for that individual member to be resized. Mouse selection has been explored but not implemented in this version of DISPAR. As with the OAPI provided by SAP, OpenGL is a very powerful tool which is virtually limitless when drawing the structure. Refer to Appendix C for a guide on how to use OpenGL with VB.Net.

4.3 LINKING DISPAR WITH EXCEL

Another beneficial feature with the new version of DISPAR is that all of the tables found in the new DISPAR for SAP2000 program can be exported to excel. Also, many of the tables which are exported include bar charts, presenting group totals in a helpful manner. The bar charts include factors for member types, factors for groups, and number of members in a group. This feature can be useful when a user wants to manipulate or arrange the data in another way than is presented in DISPAR. All of the tables that can be found in DISPAR for SAP2000 are listed in Fig. 4-8.

Element Properties	Real Forces	Virtual Forces	DISPAR
<ul style="list-style-type: none"> •Element Type •Section name •Length •Section properties 	<ul style="list-style-type: none"> •Flexural major/minor •Axial •Shear major/minor 	<ul style="list-style-type: none"> •Flexural major/minor •Axial •Shear major/minor 	<ul style="list-style-type: none"> •Element Factors •Element Type totals •Element Type percentages •Group Totals •Group Percentages

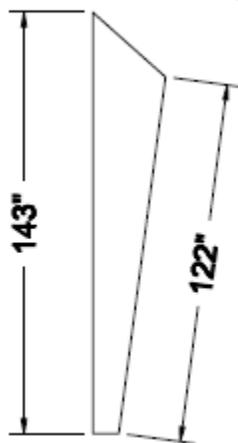
Figure 4- 8: Tables Available in DISPAR

CHAPTER 5: ANALYTICAL STUDY ON METAL BUILDINGS

This chapter is an in-depth study using the principle of virtual work to understand the behavior of metal building frames, focusing on the effect of modeling the column base connection as partially rigid. The metal building used for analysis comes from a study by Davis (1996). The study focused on the strength of members and comparison with typical design procedures. The following study is based on the drift design of metal buildings and how modeling the base connections as partially rigid will improve the economy of metal building frames. In addition to this, a sensitivity analysis will be conducted on the connection stiffness with regard to drift. This study will be used to gain insight into how to optimize the stiffness of a column base connection and steps that can be taken to obtain that stiffness.

5.1 METAL BUILDING USED FOR ANALYSIS

As mentioned previously, the metal building under consideration was the basis of research conducted by Davis in 1996. One frame in particular was used to conduct the investigations on the column base connections. The dimensions and sections sizes for the frame can be found in Fig. 5-1, Fig. 5-2, and Fig. 5-3. The overall building dimensions were not included in Davis 1996, therefore it was assumed that the building included four frames spaced at 25 ft (see Fig. 5-4).



Section Dimensions:

Web Thickness = 0.158"

Flange Width = 6"

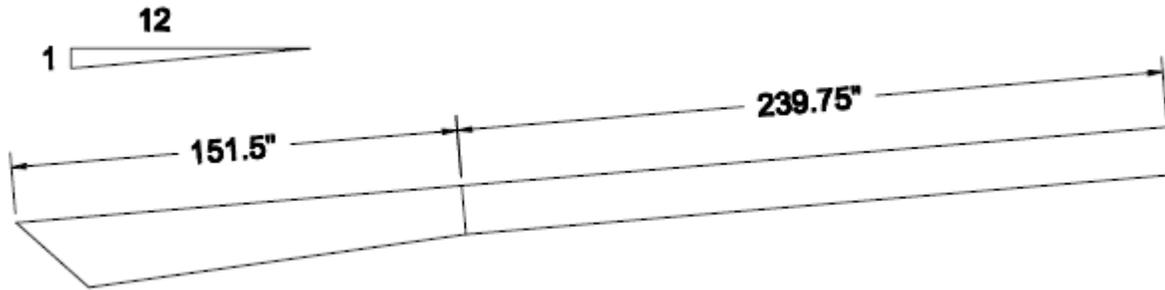
Inside Flange Thickness = 0.378"

Outside Flange Thickness = 0.381"

Smaller End Depth (out-to-out) = 8.75"

Larger End Depth (out-to-out) = 24.625"

Figure 5- 1: Column Dimensions



Tapered Section Dimensions:

Web Thickness = 0.165"
 Flange Width = 6"
 Top Flange Thickness = 0.311"
 Bottom Flange Thickness = 0.313"
 Smaller End Depth (out-to-out) = 16.625"
 Larger End Depth (out-to-out) = 24.625"

Prismatic Section Dimensions:

Web Thickness = 0.128"
 Flange Width = 6"
 Top Flange Thickness = 0.188"
 Bottom Flange Thickness = 0.192"
 Depth (out-to-out) = 16.375"

Figure 5- 2: Rafter Dimensions

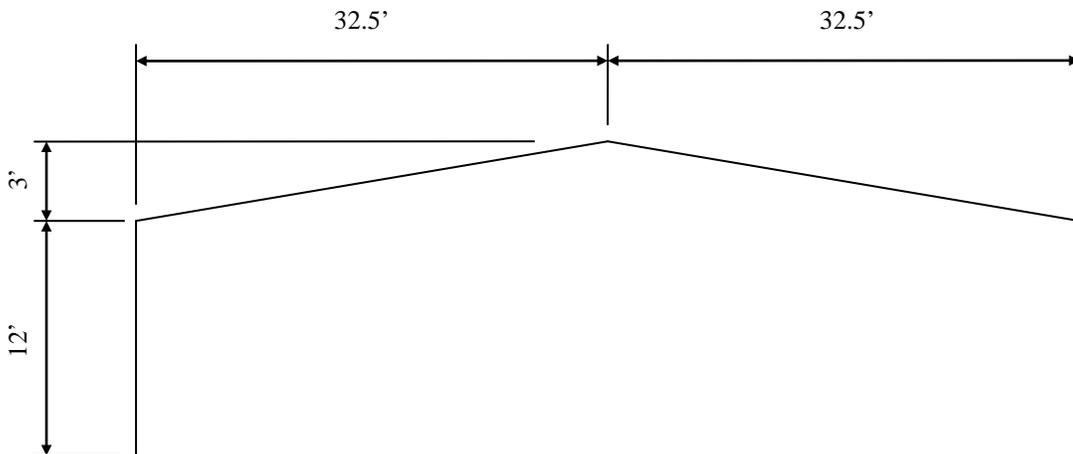


Figure 5- 3: Overall Frame Dimensions

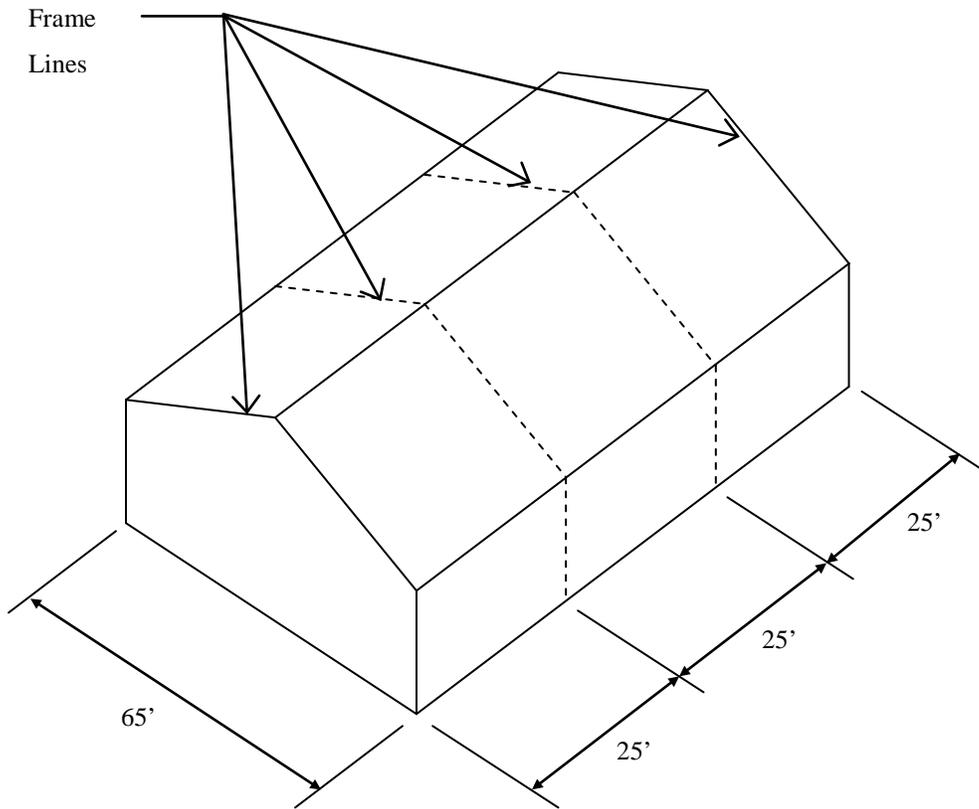


Figure 5- 4: Building Dimensions

5.2 MODELING PROCEDURE

The modeling of the metal building frame was completed using SAP2000 version 12 (Computers and Structures 2008b), with thin shell elements and nonlinear spring elements. Frame elements were modeled as described in Section 5.2.1 and the base column supports were modeled as described in Sections 5.2.3 through 5.2.4. Careful consideration had to be taken while determining the proper way to account for the column base behavior. The method had to take into account the behavior of the connection under compressive load, tensile load, bolts in tension, bolts in compression, and the plate foundation interaction. The end result is a model which behaves very similar to that of the connection in real life.

5.2.1 MODELING THE FRAME

The frame was modeled using thin shell elements with the properties of 50 ksi steel. The shell's thickness was representative of the element it was modeling. For instance, the shells used for the web of the column had a thickness of 0.158 in. Modeling the frame in this manner captures the panel zone behavior, because the panel zone region is explicitly modeled. Below are figures showing the finite element model in elevation as well as a three dimensional view of one of the columns, Fig. 5-5(a) and (b). The columns and beams were modeled based on the dimensions given in Fig. 5-1 and Fig. 5-2.

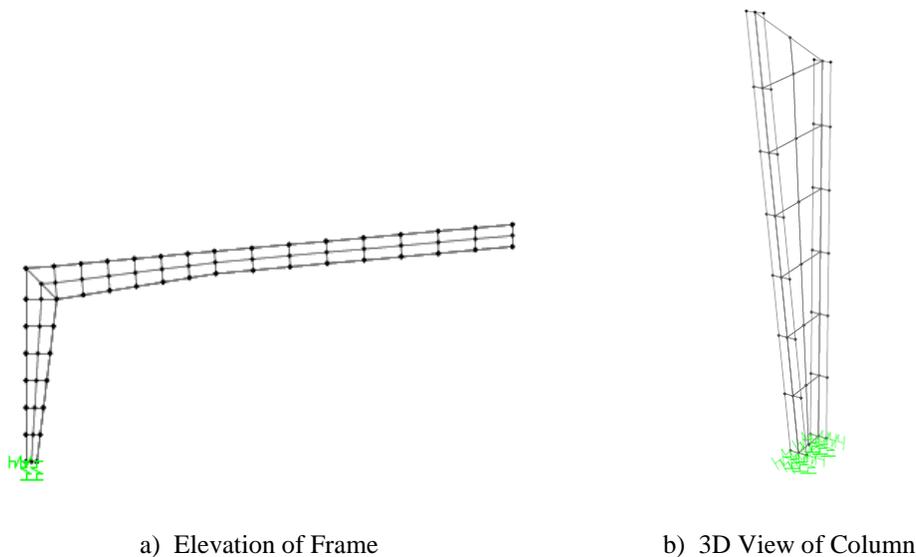


Figure 5- 5: Frame Finite Element Model

5.2.2 APPLYING THE LOADS

The frame was subject to wind loading determined in accordance with ASCE 7-05 Chapter 6 Wind Provisions (ASCE 2005). The building under consideration met the criteria listed in ASCE 7-05 Section 6.4.1.1 for the use of the Simplified Method for calculating the design wind loads. The design wind loads calculated based on this procedure represent the summation of the internal and external pressures on the system. Refer to Appendix C for an outline of how the wind loads used for analysis were determined. The frame was loaded by resolving the wind pressures along the frame into evenly distributed point loads, see Fig. C.2 for the wind loads applied to the frame. The point loads were applied along the centerline of the frame at nodes located on the exterior flange. A diagram of the metal building frame under wind loading may be found in Fig.

5-6. Notice in Fig. 5-6 that the loading applied to the rafters creates uplift on the frame. Also, the horizontal loading along the left rafter acts to the left, this is due to suction of the wind pressures.

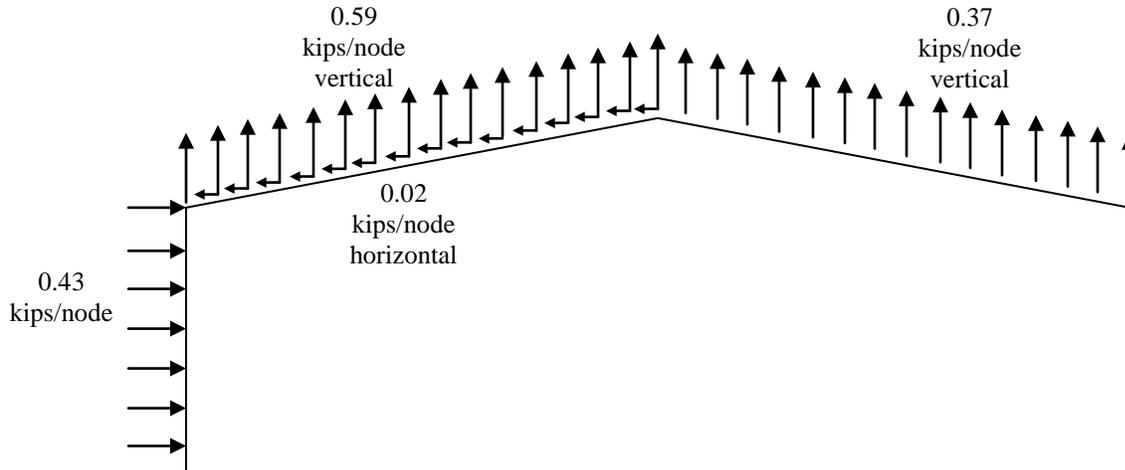


Figure 5- 6: Frame Under Wind Loading

5.2.3 DEVELOPMENT OF COLUMN BASE STIFFNESS FOR MODEL

The typical design procedure for modeling metal buildings is to assume the base connections as pinned, yet this is not true to the actual behavior. As shown in Chapter 3, even a two bolted connection, which is assumed to have no rotational stiffness, has non-negligible rotation stiffness. This section will show the steps taken to determine the rotational stiffness of the column base connection in the metal building under investigation. The first step is to make assumptions on how the column base connection will behave under applied axial and bending loads. Once these assumptions have been made, the connection Types 1 and 2 discussed in Chapter 3 will be modeled based on these assumptions. The models for connection Types 1 and 2 will then be compared with the results from the literature review found in Section 3.6. After these models have been compared with the experimental and analytical data obtained in the literature review, the rotational stiffness for a typical metal building column base connection will be determined.

5.2.3.1 MODELING ASSUMPTIONS

As a moment is applied to the base of the column, one side of the plate goes into compression and the other side goes into tension. The side of the plate in compression is pushing against the concrete foundation, not using any stiffness from the bolts. The side in tension is pulling up on the bolts, where all of the stiffness (flexibility) is coming from the bolts, see Fig. 5-7. Given that the bolts in tension act differently than the concrete in compression, it makes sense that the stiffness provided at each side of the plate will be very different.

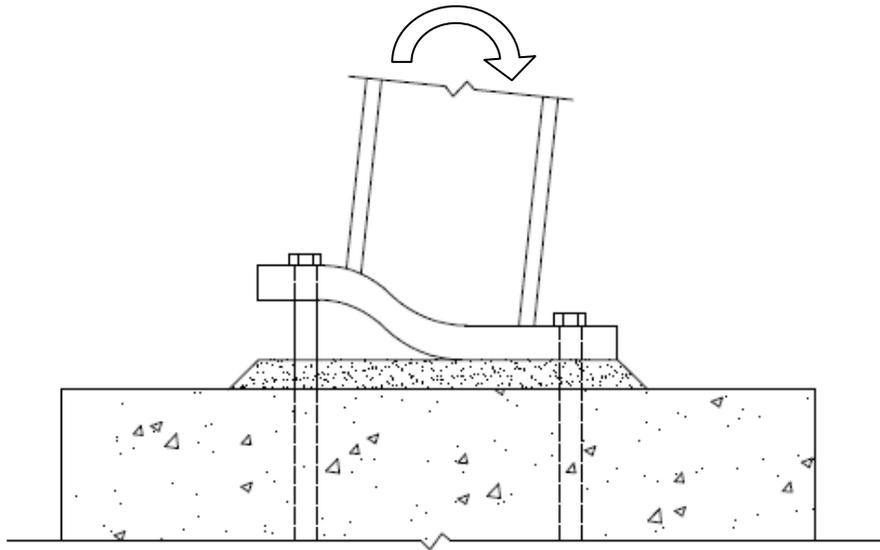


Figure 5- 7: Plate Bending Behavior

To find the stiffness associated with the bolts, an effective length must be determined. This effective length is the length of the bolt anchored into the concrete which contributes to the tensile stiffness. The stiffness can be calculated as follows:

$$k_{bolt} = \frac{AE}{L_{eff}} \quad (5.1)$$

where A is the cross sectional area of the bolt, E is the modulus of elasticity of the bolt, and L_{eff} is the effective length of the bolt. The theoretical effective length of the anchor bolt is based on the stress distribution throughout the bolt. The stress distribution of an anchor bolt embedded in concrete with a steel bearing plate is shown in Fig. 5-8, assuming no loss of bond.

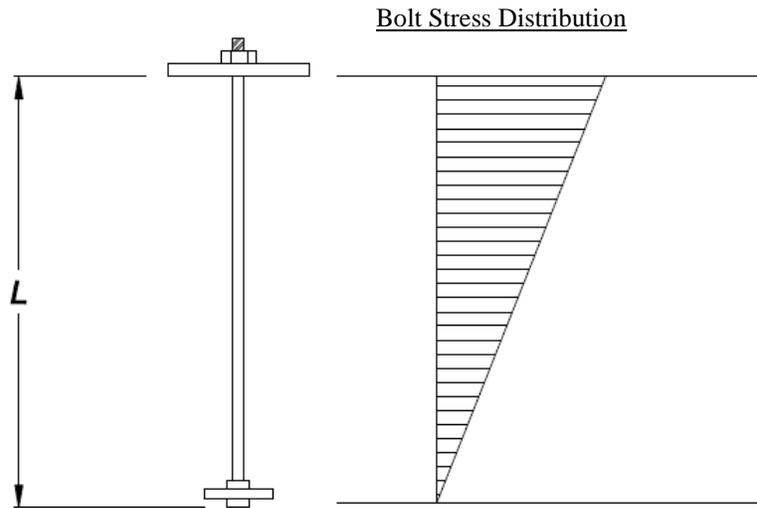


Figure 5- 8: Anchor Bolt Stress Distribution

The stiffness of an anchor bolt with the stress distribution found in Fig. 5-8 is equal to:

$$k_{bolt} = \frac{2AE}{L} \quad (5.2)$$

Therefore, the effective length of an anchor bolt embedded in concrete with a steel bearing plate is equal to:

$$L_{eff} = 0.5L \quad (5.3)$$

So for an anchor bolt length equal to 13.5 in., see Fig. 3-3, the theoretical effective length is equal to 6.75 in. The effect of pretensioning the anchor bolts on the effective length was not investigated since the research (Picard and Beaulieu (1985), Hon and Melchers (1988), Hamizi and Hannachi (2007)) did not explore this area with experimental data. It is the author's opinion that pretensioning the anchor bolts would have an effect on the effective length of the anchor bolts. The uplift on the plate would have to overcome the clamping force in the plate before it begins to lift off. This would increase the stiffness of the anchor bolt by decreasing its effective length.

The next step is to determine the stiffness associated with the concrete in compression. For modeling purposes, it was assumed that the effect of the grout layer on the rotational stiffness of the connection is negligible. This assumption is supported by the testing done by Hon and Melchers (1988), where they noticed the grout layer was not a critical element in the rotational stiffness of the connection. If it is assumed that all of the compression is taken by the concrete foundation, the stiffness of the concrete block relative to the plate, anchor bolts, and drift loads applied, can be said to be infinite. In other words, the deformations associated with the compression side of the plate are negligible. There are multiple ways that this behavior can be modeled. The first of which is to model the concrete in compression using pin supports located along the shell elements representing the plate. The pin supports will be located at the same points as the vertical springs in the width of the plate, as not to induce any off axis bending. Locating the position of the pin supports along the plate requires making some assumptions about the behavior of the plate in bending and also the amount of concrete in compression. These assumptions are very important because they determine the lever arm associated with the moment couple created by the pin support and the vertical springs representing the anchor bolts. The larger the moment arm the greater the rotational stiffness provided. So it can be unconservative to assume a small compression block at the far end of the plate because this will create a larger moment arm than what is likely provided. The problem with this modeling approach is, the actual size compression block will change based on the magnitude of axial load and moment applied to the column and this modeling approach does not account for that.

Another, and more accurate, approach to modeling the column base connection is to use a grid of nonlinear springs to represent the concrete in compression. These springs will have a near infinite stiffness in compression and a near zero stiffness in tension. If the entire plate is supported on a grid on nonlinear springs with a near infinite compression stiffness and a near zero tensile stiffness, the size of the compression block will be dependent on the magnitude of the loads. The bolts will be modeled similar as discussed above, using springs at the location of the bolts, with the exception that a nonlinear spring will be used. The nonlinear springs for the anchor bolts will have a tensile stiffness equal the effective bolt stiffness and a compressive stiffness equal to nearly zero. The effective length used for calculating the tensile stiffness of the

bolts is equal to 6.75 in., as discussed previously. Refer to Fig. 5-9(a) and Fig. 5-9(b) for the force deformation relationship of the concrete springs and the bolt springs.

Using the grid of nonlinear springs allows the model to account for different magnitudes and ratios of vertical load and moment applied to the connection. This is because the nonlinear springs only act in compression for the concrete and tension for the bolts. Also, the model does not have a fixed compression block size like the method using the pinned supports with vertical springs. The compression block size varies depending on the magnitude of the loads. It is assumed that this modeling approach is the most accurate of the ones discussed in this section since it accounts for the critical elements outlined in Section 3.4, as well as the layout of the bolts. Therefore, it will be used to determine the rotational stiffness for the connection type in Fig. 5-12.

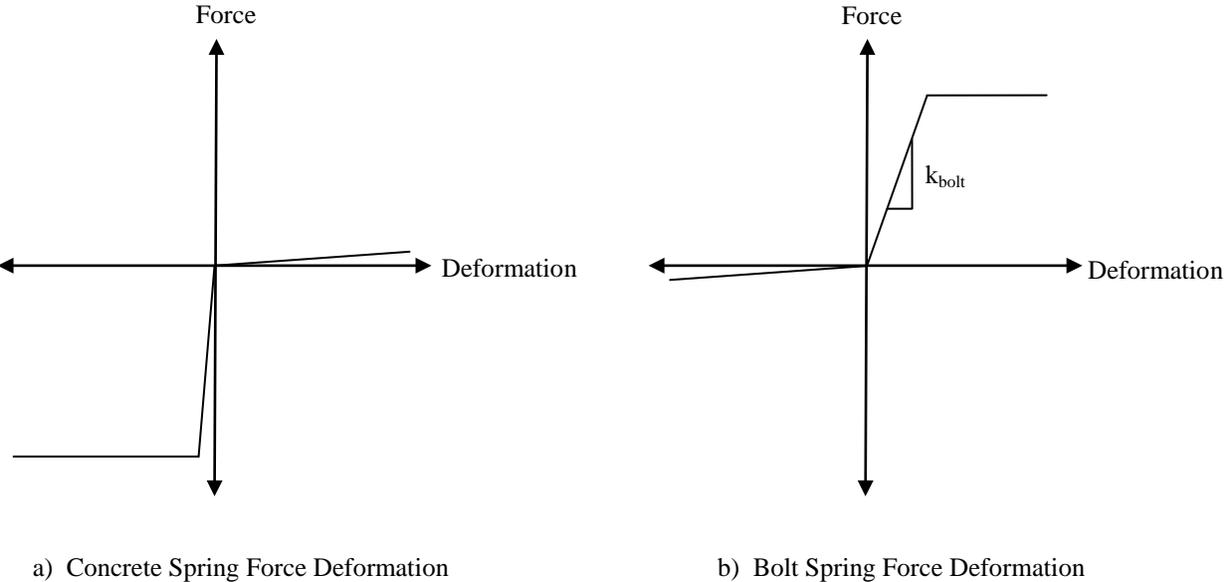


Figure 5- 9: Nonlinear Spring Force Deformation Relationships

The force deformation relationship for the connection for this modeling approach will be linear in nature. This is due to the concrete springs having a linear compressive stiffness and the anchor bolt springs having a linear tensile stiffness. If yielding of the bolts or crushing of the concrete was taken into account, the force deformation relationship for the connection would be nonlinear. Since the force deformation relationship for the connection is linear, the resulting stiffness from model will be the initial stiffness. If the relationship was nonlinear, a tangent or secant stiffness would need to be developed based on the relationship.

5.2.3.2 DEVELOPMENT OF MODEL

To capture the rotational stiffness of connection Types 1 and 2, a stub column model was built for each connection type. The height of the stub column used in the model was taken 12 in. The column and connection was modeled using thin shell elements in SAP2000 with a thickness associated to its member properties. In other words, shell elements used for the flange of the column were assigned its respective thickness. An image of the finite element mode for connection Type 2 is shown in Fig. 5-10. The dimensions and material properties of members used in the model were representative of the actual connections tested in Picard and Beaulieu (1985) and modeled in Hamizi and Hannachi (2007). Further description of these connections may be found in Section 3.5.2.

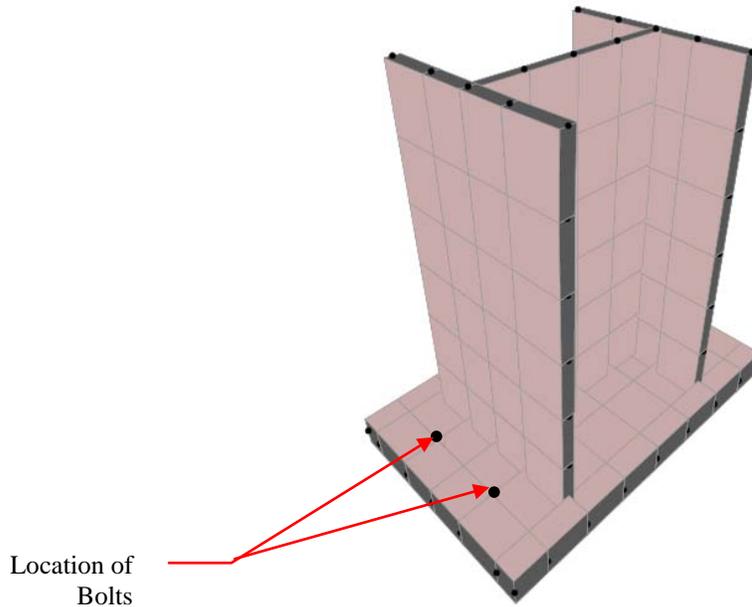


Figure 5- 10: Finite Element Model of Type 2 Connection

As mentioned above, this method involves the use of nonlinear springs to represent both the concrete stiffness and the bolt stiffness in the connection. The concrete springs were connected to the shell elements used to model the plate and were laid out on a grid spacing of approximately one in. The grid spacing was based on plate dimensions, column dimensions, and bolt spacing, and therefore is not a constant one in. throughout but is symmetric, see Fig. 5-11.

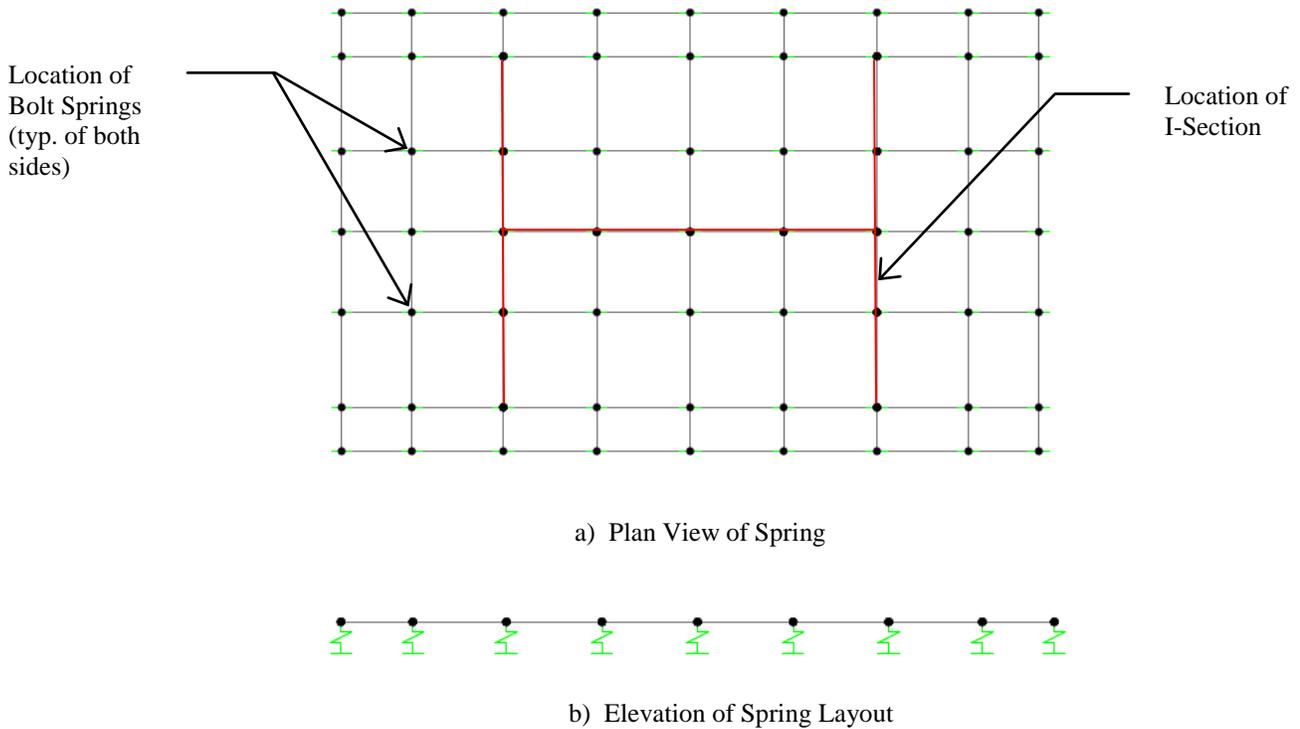


Figure 5- 11: Spring Grid Layout for 4 Bolt Connection

To determine the compressive stiffness assigned to the concrete springs an iterative process was used to build a model for connection Type 2. The iterative process was begun using a stiffness of 1000 k/in for the concrete springs and a stiffness for the bolts of 1800 kip-ft/radian (stiffness using the theoretical effective length of 6.75 in.). The stiffness for the concrete springs was increased until relatively little change was seen in the displacements experienced. The resulting stiffness was equal to 10000 k/in for the concrete springs. Another model was then built for connection Type 1 using the same stiffness values determined for connection Type 2, 10000 kip/in for the concrete spring and 1800 kip/in for the anchor bolt springs.

5.2.3.3 COMPARISON WITH LITERATURE

This modeling technique was then compared with the initial rotational stiffness values determined by research done by Picard and Beaulieu (1985) and Hamizi and Hannachi (2007). The rotational stiffness of the stub column model was determined by applying a 10 kip force at the top of the stub column, creating a moment of 10 kip-ft, and taking the rotation at the base of

the column. The rotation was taken halfway between the flanges of the column, on the top of the base plate. The rotation stiffness of the connection is equal to the moment applied divided by the rotation at the center of the base of the column, see Table 5-1 for the results.

Table 5- 1: Initial Connection Stiffness Comparison

Connection	Initial Stiffness (k-ft/radian)			
	Testing (Picard)	Modeling (Hamizi)	Average	SAP Model
2 Bolted	1600	1800	1700	800
4 Bolted	14000	12000	13000	10000

When compared to the initial stiffness of each connection type from the plots in Section 3.6, the modeling approach developed in this section underestimates the initial stiffness. Therefore, the modeling assumption presented in this thesis is conservative when compared with experimental and analytical test results. This could mean that the effective length used to determine the stiffness of the anchor bolts is a conservative estimate of its actual effective length. This would also mean that the stress distribution in the anchor bolt is slightly different than shown in Fig. 5-8. Since the initial stiffness values determined through modeling are on the same order of magnitude when compared with test results, the assumptions used to model the 2 bolted and 4 bolted connections can be extended to the typical connection used in metal buildings.

5.2.3.4 STIFFNESS OF TYPICAL METAL BUILDING CONNECTION

The column base connection under investigation is based on a typical column base connection detail found in metal buildings designed by American Buildings Company (Walsh 2009), see Fig. 5-12. It was assumed for modeling purposes that the thickness of the plate is 0.5 in. and $\frac{3}{4}$ in. bolts were used. In addition, the anchor bolt embedment detail was not shown in typical column base connection detail in Walsh (2009). Therefore, it was assumed that the anchor bolts will be embedded similar to that of the testing procedure in Picard and Beaulieu (1985). It was also assumed that all anchor bolts are snug tightened, therefore no pretension force is present.

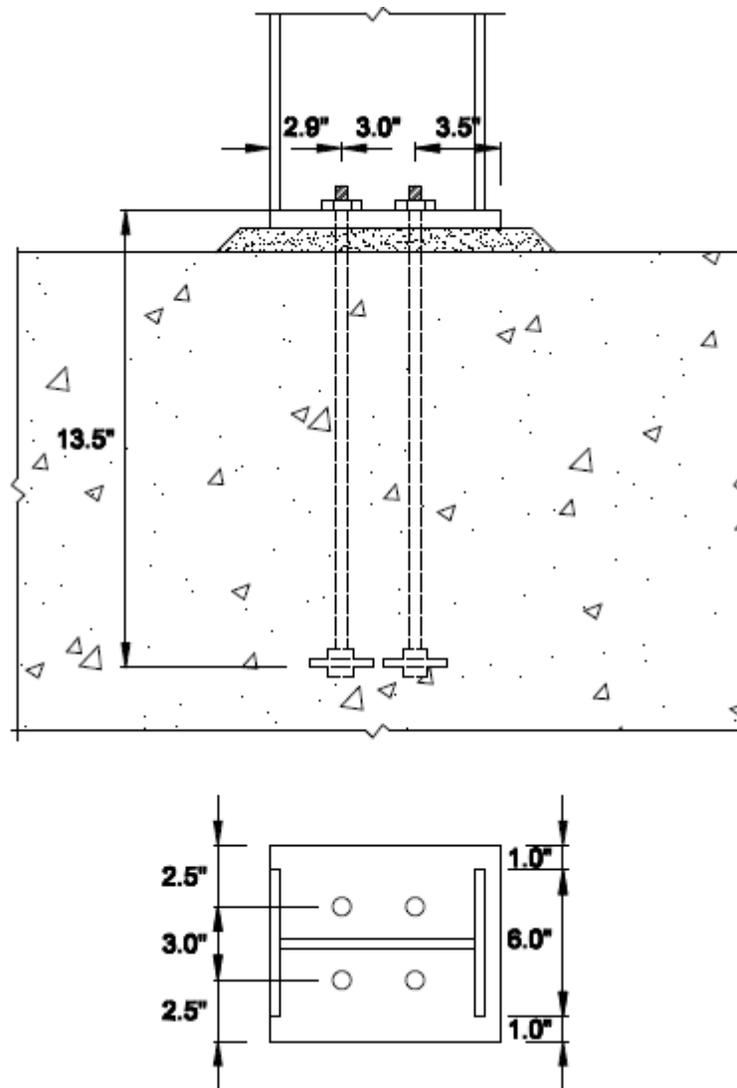


Figure 5- 12: Metal Building Column Base Connection

After verifying the results of the modeling assumptions discussed in Section 5.2.3.2, they were extended to model the behavior of the connection detail in Fig. 5-12. A 3D view of the finite element model used for the column base connection in the metal building is shown in Fig. 5-13.

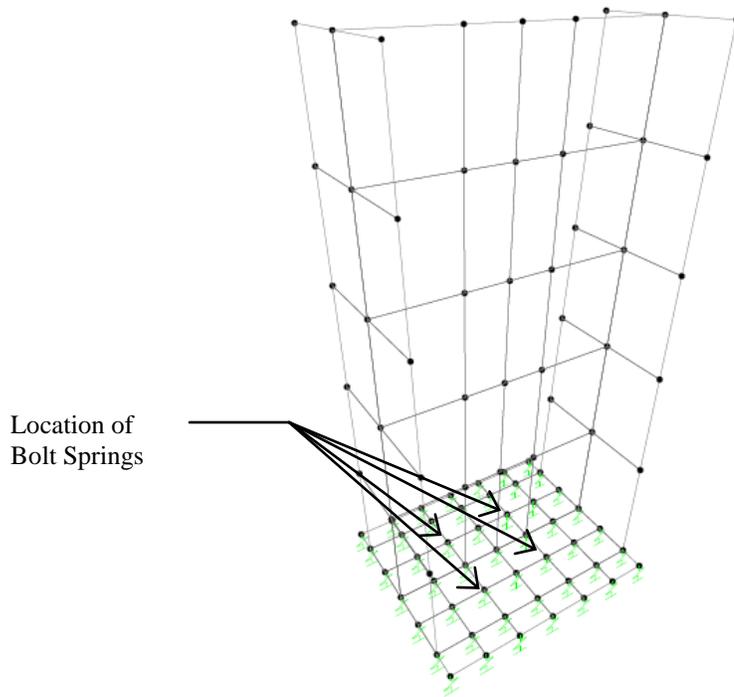


Figure 5- 13: Finite Element Model of Metal Building Base Connection

The rotation stiffness for the column base connection found in Fig. 5-12 was determined to be 4200 kip-ft/radian, using the modeling assumptions previously stated. As one would expect, the connection detailed in Fig. 5-12 with 4 bolts located inside the flanges of the column, has a rotational stiffness in between connection Type 1 (1700 kip-ft/radian) and Type 2 (13000 kip-ft/radian), stiffness taken as average from Table 5-1. It would also make sense that the metal building connection would behave closer to that of the two bolted connection, because the two bolt rows are only 3 in. apart and both are located inside the flanges of the column. Therefore, it is accepted for modeling purposes that the 4 bolted connection found in Fig. 5-12 has an initial stiffness of approximately 4200 kip-ft/radian.

5.2.4 SIMPLIFIED MODELING APPROACHES

The purpose of this section is to discuss simplified modeling approaches to capture the behavior of the column base connection. This will allow designers other options for modeling the

connection that do not require as much time to model as the nonlinear spring method. The first and simplest approach is to model a pinned support at the midpoint of the base of the column. To capture the rotational stiffness, a rotational spring with stiffness equal to that of the column base connection would be used at the pin. Another approach is to assume that each bolt has the same stiffness and use vertical springs at the location of each bolt. The rotational stiffness provided by the connection is then the coupled vertical stiffness from the rows of bolts. There are multiple flaws in assuming that the connection acts in both of the above manners. The first approach does not allow for any uplift of the connection, which is likely the case in metal buildings. It also does not take into account the bending that will occur in the plate as the connection deforms, shown earlier to be critical. With the second approach, it assumes that the connection will behave the same on the side that bolts are in compression and the side that bolts are in tension, which is not the case. Also, if there is only one row of bolts on the connection, the model will have no rotational stiffness. Therefore, physically neither approach captures how the connection actually behaves. The other approach that will be discussed in this section is the pinned support and vertical spring model mentioned in section 5.2.3.1.

5.2.4.1 ROTATION SPRING APPROACH

As mentioned previously, the modeling approach involving a pinned support with a rotational spring is the most simplistic of the four approaches studied. This method requires determining the rotational stiffness of the connection either through testing or a detailed finite element analysis. Once the stiffness of the desired connection type is known, all that is left is to model a pinned support with a rotational spring with the specified stiffness. The stiffness used for the rotational spring for the metal building connection was the stiffness determined in Section 5.2.3.4 of 4200 kip-ft/radian. This modeling approach behaves well as long as there is not a lot of uplift imposed on the connection and overall frame behavior is the primary concern. Refer to Fig. 5-14 for an image of the finite element stub column used for this approach.

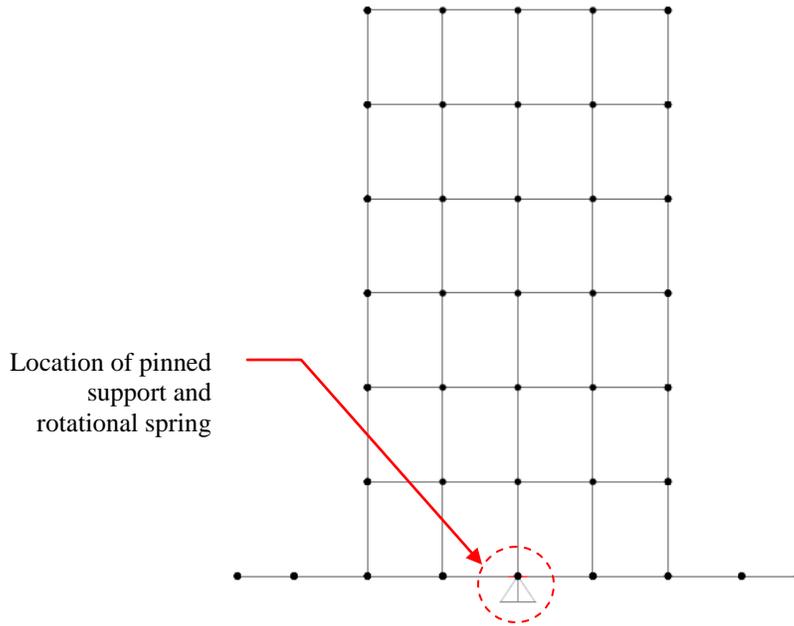


Figure 5- 14: Pinned Connection with Rotational Spring (Type 2)

5.2.4.2 SPRINGS AT BOLT LOCATIONS

The next approach was to model the support condition with vertical spring's location where anchor bolts are located. The spring stiffness associated with each anchor bolt for the metal buildings connection was 1800 kips/in. This was based on the effective length of 6.75 in. discussed previously. Using a vertical spring stiffness based on the effective length of the bolts resulted in a rotational stiffness equal to 950 kip-ft/radian. This modeling approach behaves well in most cases, but produces conservative results. Refer to Fig. 5-15 for the finite element model of this stub column.

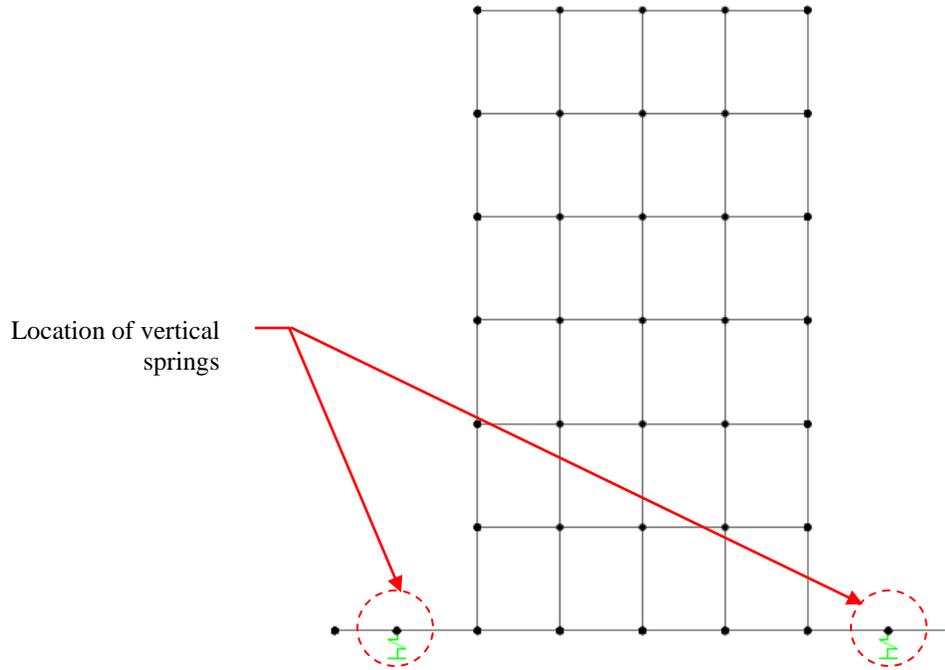


Figure 5- 15: Vertical spring base connection

5.2.4.3 PINNED SUPPORT AND VERTICAL SPRING

The method discussed in this section is based on the pinned support with vertical springs model, assuming a fixed compression block size. For modeling purposes it is assumed that the plate begins to compress the concrete halfway between the bolts in tension and the furthest edge of the concrete in compression. If it is assumed that the concrete compression block is rectangular, the location of the reaction or the location of the pin supports will be length of the plate in compression divided by two, see Fig. 5-16. Therefore, it can be stated based on the above assumptions that the location of the pin support from the edge of the plate in compression is equal to:

$$d_c = L - \frac{a}{2} \quad (5.4)$$

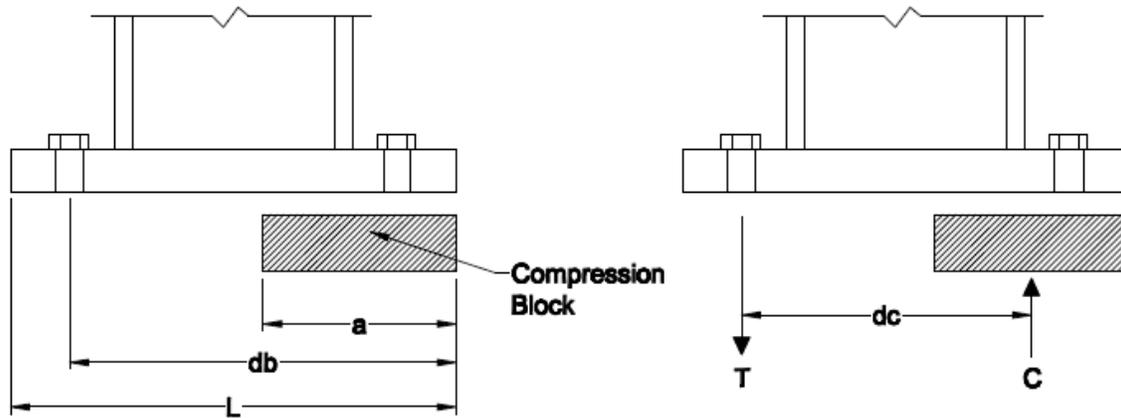


Figure 5- 16: Location of Compression Force for Connection Type 2

where L is equal to the total length of the plate, d_b is equal to distance from the edge of the plate to the bolts in tension, and a is d_b divided by two. The distance a is equal to half of d_b because it is assumed that the reaction force acts at the centroid of the compression block. The next step was to determine the spring stiffness associated with each bolt in tension given the assumptions above. This was based on the effective length of the anchor bolt. The vertical stiffness associated with the bolts in tension was determined to be 1800 kips/in, which correlates to the effective length of 6.75 in. This resulted in a rotational stiffness of the connection equal to 2200 kip-ft/radian. As mentioned before, vertical springs were used to represent the bolts in tension and pinned supports were used in line with the bolts a distance d_c away from the tension side of the plate. A simplistic model representing the support conditions of the plate may be seen in Fig. 5-17.

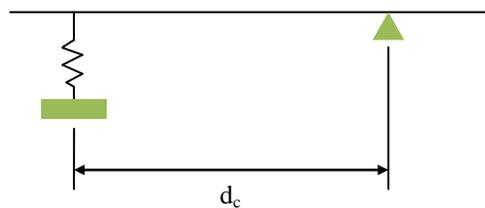


Figure 5- 17: Support Conditions of Plate

5.3 BEHAVIOR OF CONNECTIONS WITH VARYING LOAD COMBINATIONS

Since the nonlinear spring model requires most time and effort, it is desired to determine which of the more simplified approaches produces accurate results when compared to the nonlinear model. This section will explore how each of the modeling approaches behaves when used in a full frame and varying loading conditions. Each of the four methods was modeled based on the rotational stiffness discussed in the previous sections (4200 kip-ft/radian for nonlinear and rotation spring, 950 kip-ft/radian for vertical springs, and 2200 kip-ft/radian for pinned support and vertical springs). In order to carry out this study, multiple load combinations were applied to the metal building frame. The first loading condition was the frame subject to only a horizontal load. A force of 5 kips was applied to the frame at the left and right joints with each individual connection type. The next loading condition was the frame subject to the ASCE 7-05 (ASCE 2005) design wind loads, as described in Section 5.2.2. In addition to the wind loading, this section will look at the effects of varying the gravity live load on the roof of the frame. Once this study is complete, recommendations can be made to designers on which of the simplified approaches to use in design.

5.3.1 FRAME UNDER HORIZONTAL LOAD

As mentioned above, each frame was subject to a 5 kip horizontal load at both the left and right column and rafter joints. The drift reported was the horizontal displacement at the middle of the right column rafter joint. Since the model using the nonlinear springs is considered to be the most accurate, each of the drifts from the other modeling approaches were compared with this. The drift due to the horizontal on the nonlinear spring model was 0.82 in.

Table 5- 2: Connection Comparison with Horizontal Load

Modeling Approach	Drift (in.)	Percent Difference
Pinned	1.29	57.3%
Pin/Rotational Spring	1.22	48.8%
Vertical Springs	1.13	37.8%
Pin/Vertical Spring	1.11	35.4%

Notice in Table 5-2 the more simplistic the model, the less accurate and more conservative the results when the frame is only subjected to a horizontal load. By assuming the connection is pinned, the drift obtained is conservative by 57.3 percent. The model closest to the nonlinear spring approach is the pin and vertical spring model, which is still conservative by 35.4 percent.

5.3.2 FRAME UNDER VARYING LIVE LOAD

Since each of the connection types behaves differently when subject to a vertical load, it was important to look into how they behave when the full frame is subject to vertical load conditions. To do this, the frame was subjected to design wind loads with the gravity live load varying from 0 to 100 percent. The live load was determined from ASCE 7-05 Chapter 4 for live loads on structures for a roof live load for an ordinary pitched roof. The gravity load was equal to 20 pounds per square foot applied over the frame's 25 foot tributary width. To apply the gravity live load, evenly distributed point loads were applied on nodes along the centerline of the exterior flange on the rafters. This is the same approach used for applying the wind loading to the frame, see Fig. 5-18 for a diagram of the applied gravity live loads.

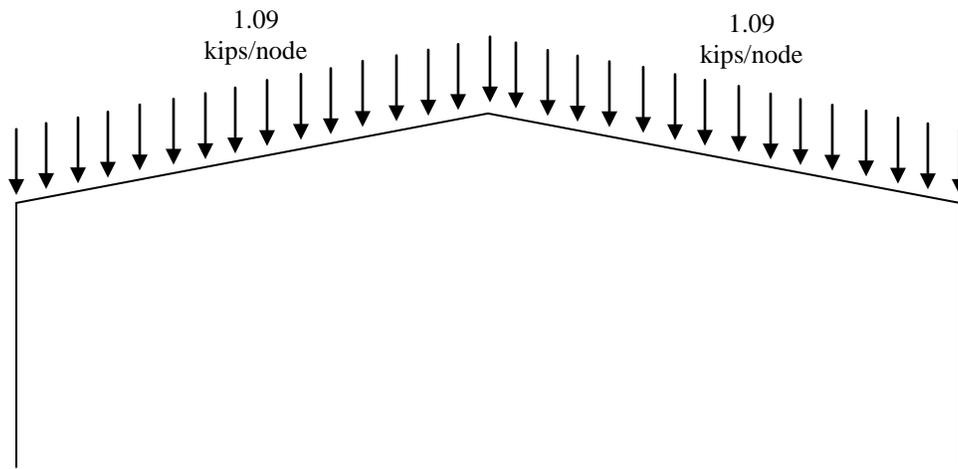


Figure 5- 18: Frame Under Gravity Live Load

Each connection modeling approach was analyzed with 100 percent of the wind load while varying the live load from 0 to 100 percent in intervals of 20 percent. Varying the live load in this manner examines how the connection behaves when there is uplift on the frame (only wind load) as well as when the live load overcomes uplift on the frame. It is highly unlikely that the frame will experience the higher portions of live load while the wind load is at 100 percent. This was done only to show how the frame behaves under varying loading conditions, not actual loading conditions. Refer to Table 5-3 and Fig. 5-19 for the results from the varying live load study.

Table 5- 3: Drift at the Left Corner of the Frame Based on Percentage of Live Load

Modeling Approach	Percentage of Live Load Considered					
	0 (Wind only)	20	40	60	80	100
Pinned	0.32	0.27	0.22	0.17	0.12	0.064
Pin/Rotational Spring	0.31	0.26	0.21	0.16	0.11	0.056
Vertical Springs	0.3	0.25	0.2	0.15	0.1	0.05
Pin/Vertical Spring	0.36	0.28	0.21	0.14	0.06	-0.01
Nonlinear Springs	0.25	0.2	0.16	0.11	0.06	0.01

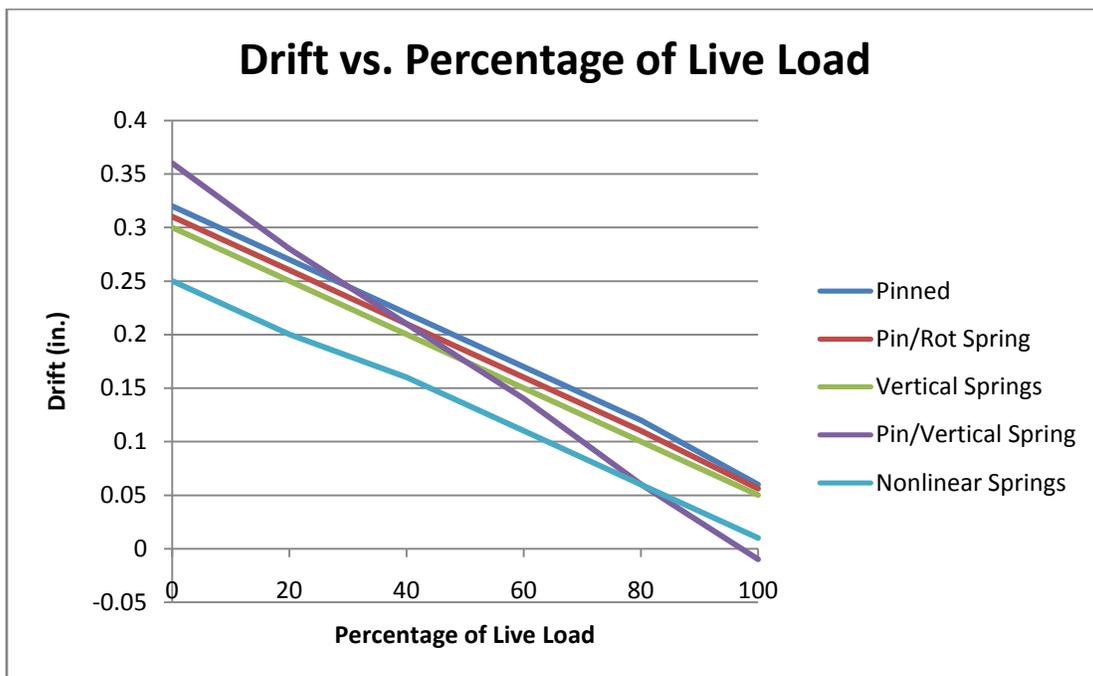


Figure 5- 19: Drift vs. Percentage of Live Load

Notice in Fig. 5-19 that the majority of the modeling approaches follow a similar trend. As the gravity live load is increased on the frame the displacement at the left column rafter joint decreases. This is due to the live load overcoming the uplift of the frame under wind loading and pushing the columns out. The only one that deviates is the model using pinned supports with vertical springs (pin/vertical springs) representing the bolts in tension. All of the approaches except this one follow roughly the same linear path as the gravity live load is increased. This modeling approach behaves in this manner because it only accounts for bending on one direction and once the live load forces the frame to bend in the opposite direction, the connection loses stiffness, increasing the drift in the negative direction. It is apparent that the modeling approach using a pinned support with vertical springs is not an accurate way to model the column base connection when the loading forces the frame to bend in the opposite direction.

Since the majority of the models behave with the same trend, it can be stated that any of the above approaches, other than the pinned support with vertical springs, may be used when modeling the behavior of the column base connection. The approaches are conservative when compared with the nonlinear spring model, but produce less conservative results than using a pinned support.

5.4 SENSITIVITY ANALYSIS OF COLUMN BASE CONNECTION

The sensitivity analysis was conducted on the metal building frame from Section 5.1 with a 10 kip load applied at the column rafter joints on both the left side and the right side of the metal building frame. DISPAR for SAP2000 was utilized to determine the contribution factors for the connection and the frame members. The frame was modeled with a base column connection stiffness that varied from 0 kip-ft/radian (pinned connection) all the way to 1×10^7 kip-ft/radian (fixed connection). Various comparisons were made to determine how sensitive the metal building frame is to the stiffness of the base connection. The connection modeling approach used for analysis was based on the technique discussed in Section 5.2.3.4, as this was the connection determined to behave most accurately. In order to vary the rotational stiffness of the connection the nonlinear spring stiffness for the concrete and bolts was varied. It was determined that within

a given connection stiffness range the frame is sensitive to the connection stiffness. Once outside of that range, the connection behaves as either a fully flexible or fully rigid connection.

5.4.1 SENSITIVITY OF BASE CONNECTIONS WITH REGARD TO OVERALL DRIFT

This section examines the behavior of the column base connection in relation to the drift of the frame as well as the connections contribution to that drift with relation to its stiffness. The connection was modeled using the nonlinear spring approach. For the first graph, Fig. 5-20, the overall drift of the frame was plotted against the rotational stiffness of the base column connection. So for a given rotational stiffness, the drift was recorded to understand how explicitly modeling the stiffness of the base connection will affect its behavior when related to drift. Next, a plot was developed which was the derivative of the first plot to show how much changing the stiffness at a given point will affect the drift, shown in Fig. 5-21. A similar plot was developed with the connection contribution to the overall drift as a percentage versus its respective rotational stiffness, Fig. 5-22. This plot was created using DISPAR for SAP2000 to calculate the displacement participation factor of the springs used in the connection, effectively giving the DISPAR factor for the connection. The connections DISPAR factor was then normalized with the overall drift for that particular run, giving the connections contribution with regard to drift as a percentage. This plot gives insight into how the magnitude of a connection contribution changes depending on its rotational stiffness.

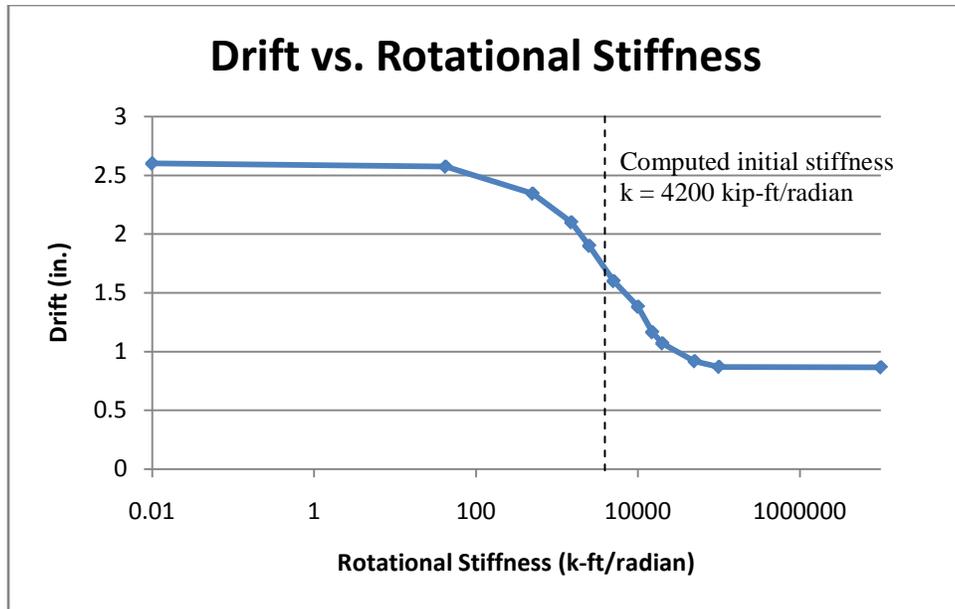


Figure 5- 20: Drift vs. Spring Stiffness

There is a region in which changing the rotational stiffness of the column base connection has almost a linear affect on the overall drift of the frame. Once outside of this region, there is essentially no change in the drift. For the metal building frame and connection type considered, the region where changing the rotational stiffness has an effect on the drift is located between 100 to 10,000 (kip-ft/radian), see Fig. 5-20. If it is determined through analysis or testing that the actual connection stiffness is above or below this region, the connection is behaving as pinned or fully fixed. A connection with stiffness below 100 kip-ft/radian behaves as pinned and a connection with stiffness above 10,000 kip-ft/radian behaves as a fully fixed connection. The metal building connection analyzed in this thesis (4200 kip-ft/radian), Fig. 5-12, is located in between 100 and 10,000 kip-ft/radian so it should be analyzed as partially rigid.

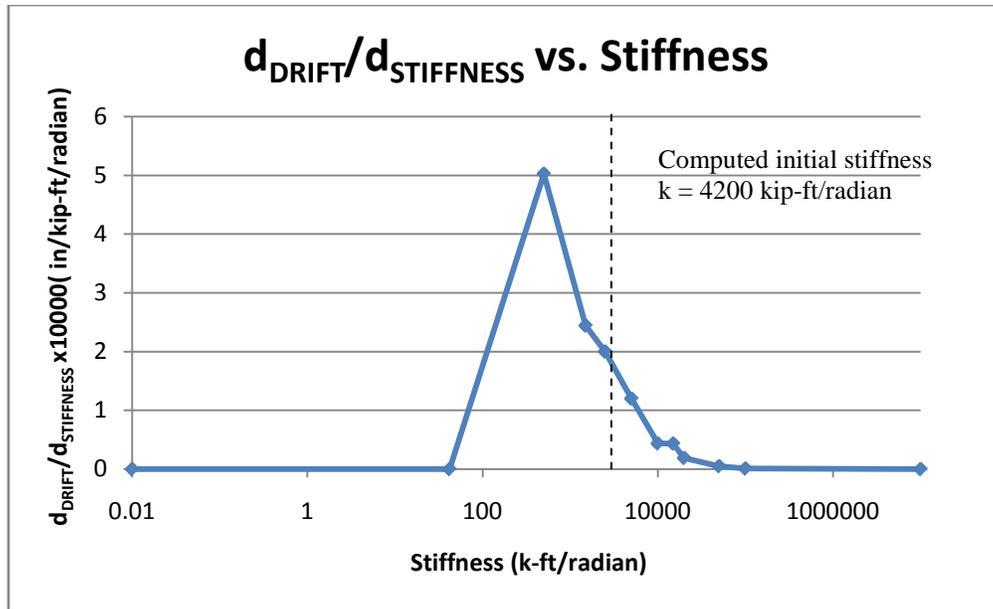


Figure 5- 21: Derivative of Drift vs. Stiffness Plot

The plot in Fig. 5-21 shows how sensitive the rotational stiffness of the connection is to change. In other words, this plot shows for a given stiffness how much change in drift to expect per change in rotational stiffness. The point most sensitive to change based on Fig. 5-21 is located where the rotational stiffness in the connection is equal to 500 k-ft/radian. Moving to the left or right at this point on the plot will affect the drift per change in stiffness more than at other points on the plot. Similar to Fig. 5-20, once the stiffness is below roughly 100 kip-ft/radian there is essentially not change in the drift by decreasing the stiffness. Also, once the stiffness is above 10,000 kip-ft/radian there is little to no change in drift when increasing the stiffness.

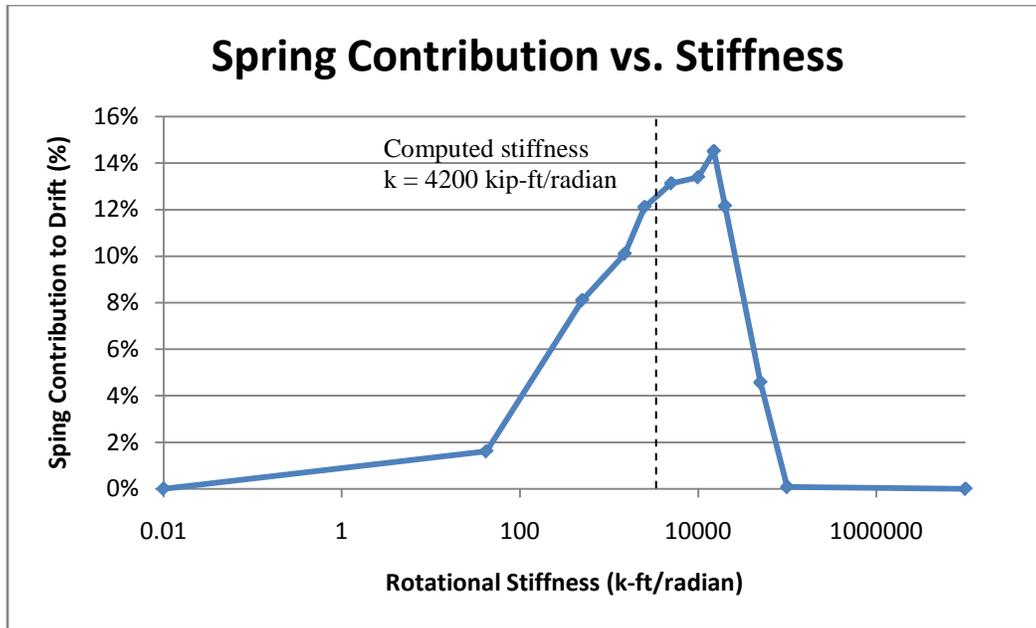


Figure 5- 22: Spring Contribution vs. Rotational Stiffness

Notice in Fig. 5-22 that the stiffness that contributes the most to the overall drift (in terms of percent of total) occurs at approximately 10,000 kip-ft/radian. As the stiffness increases above 10,000 kip-ft/radian its contribution to the overall drift quickly decreases to zero. Similarly, as the connection stiffness decreases below 10,000 kip-ft/radian the connections contribution decreases to zero. The rate at which the connections contribution decreases between 10,000 to 100 kip-ft/radian is less than the decrease above 10,000 kip-ft/radian. Notice that the metal building connection’s contribution is located in this region at 13% of the overall drift.

5.4.2 BASE CONNECTION SENSITIVITY VS. COLUMN AND RAFTER SENSITIVITY

This section compares the behavior of the connection with relation to the behavior of different components of the metal building frame, with varying connection stiffness. The member sensitivity index (SI) was used when comparing the components. Recall from Section 2.3.2 that a member’s SI is a measure of how sensitive it is to change, the higher the SI the more impact changing that member’s size will have on the drift. To calculate the SI for the connection an effective volume was used. This volume was based on the stiffness used for the nonlinear

vertical springs representing the anchor bolts. Since the anchor bolt stiffness was varied to increase the rotational stiffness of the connection, this stiffness is a good indication to have the effective volume of the connection varies. In addition, the anchor bolt stiffness is a function of the area divided by the length of the bolt. If it is assumed that the length of the bolt is kept constant, the area is the property that would change when the stiffness is increased. Therefore, the stiffness of the anchor provides an effective volume that relates well to the connection. The diagram in Fig. 5-23 shows how the metal building frame was broken into groups for analysis. Sensitivity Indices were calculated for each component of the frame, including the connection, for a connection stiffness varying from zero (pinned) to infinite (fixed) and plotted in Fig. 5-24.

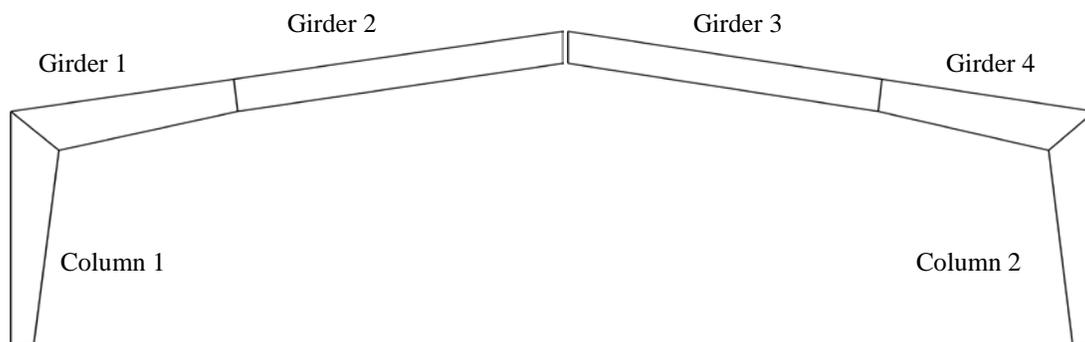


Figure 5- 23: Member Labels for SAP Groups

The columns were considered as individual members (column 1 and column 2) and the rafters were broken up into four different members (Girder 1 – 4). Girders 1 and 2 are the tapered sections framing into the columns and Girders 2 and 3 are the prismatic sections of the rafter. Fig. 5-24 only includes the SI's for the Column 1, and Girders 1 and 2 because the other side of the frame behaves in the same manner.

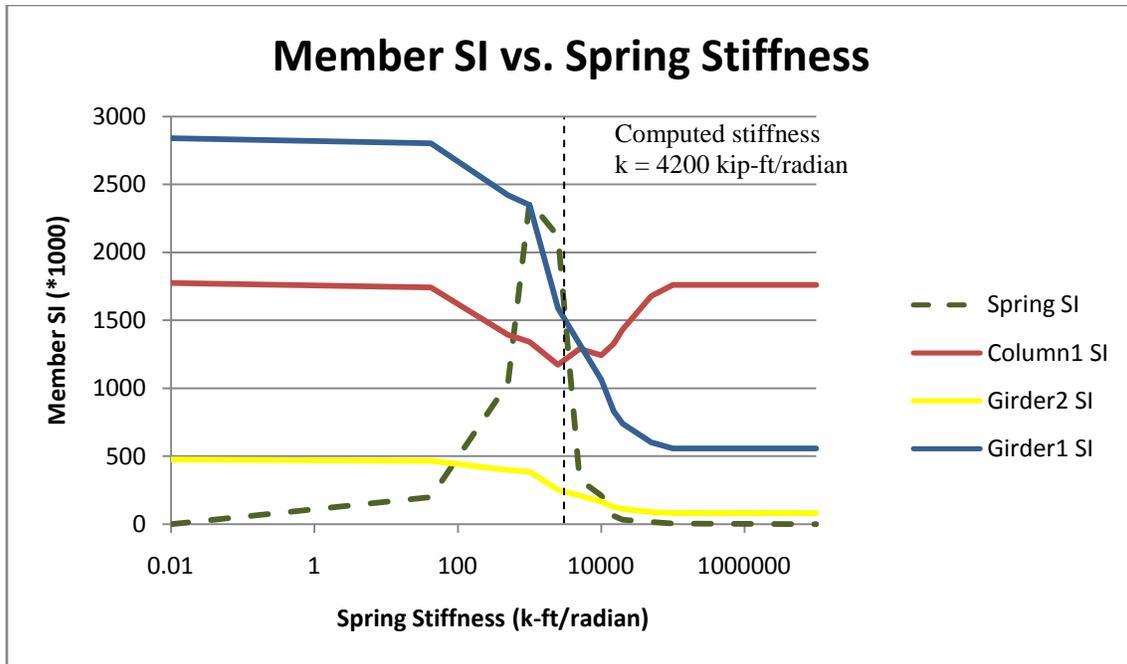


Figure 5- 24: Member SI vs. Spring Stiffness

As the connection stiffness becomes more influential to the overall drift of the frame, the sensitivity of the frame members decreases. It is interesting to note that the SI for all of the frame members decreases as the SI for the connection increases, yet when the connection sensitivity begins to decrease only Column 1’s sensitivity begins to increase. Notice how similar the “Spring SI” plot is to the derivative plot shown in Fig. 5-21. Both plots show that at a connection stiffness of 500 kip-ft/radian, the connection is most sensitive to change. This is not necessarily showing that this is the optimal stiffness for the connection, only that the drift will be impacted the most per change in stiffness at this point.

Also, it is important to note that it might be more economical to reduce the size of Girder2 and increase the size of Girder1. This is especially the case in the region where the connection is behaving as a “pinned connection”, where the SI for Girder1 is nearly six times that of the SI for Girder2. Even as the connection begins to behave as a “fixed” connection, the SI ratio for Girder1 to Girder2 is about the same. At this point, it may be more economical to increase the

size of Column1 and reduce the size of Girder2. In the region where the connection type found in Fig. 5-12 acts (4200 kip-ft/radian), Girder1 and Column1 have approximately the same SI value. With this in mind, it would likely be economical to increase the size of Girder1 and Column1 and reduce the size of Girder2 until the SI values for all are approximately equal. This would be the most economical design with regard to volume of material for this metal building frame and connection type.

5.5 CONCLUSION

For the analysis conducted in this section, it can be stated that explicitly modeling the column base connection stiffness is important. The overall drift of the metal building frame under consideration can vary as much as 57.3 percent (Table 5-2) if the column base connection is assumed to be pinned. This is compared to explicitly modeling the initial stiffness of the column base connection. When modeling the stiffness of the connection, using simplistic models such as a pinned support with a rotation spring can be conservative when compared with more realistic models, but it is closer to real life behavior than a solely pinned connection. In the presence of vertical loading on the full frame most of the modeling procedures behaved in a similar manner, but again the more simplistic models produced more conservative results. Based on the analysis conducted in Section 5.3, a benefit can be seen from including the effects of the column base connection. Therefore, modeling the actual behavior of the column base connection is important because it can reduce the drift and increase the overall economy of a frame.

In general, to optimize the stiffness of a column base connection, it is best to design the connection for a stiffness that contributes the maximum amount to overall drift for a given frame. This stiffness can be determined from the plot shown in Fig. 5-22, which plotted the connections contribution to overall drift versus connection stiffness. Notice in Fig. 5-22 how quickly the connections contribution to overall drift decreases for a stiffness above 10,000 kip-ft/radian. This point also correlates with the stiffness at which the connection begins to behave similar to a “fixed” connection. The plots in Fig. 5-20, Fig. 5-21, and Fig. 5-24 show that once the connection stiffness is increased above 10,000 kip-ft/radian, there is little change in drift or

sensitivity to change in drift per change in stiffness seen. For the metal building frame explored in this chapter, the connection stiffness that meets these criteria is 10,000 kip-ft/radian. This initial stiffness is obtainable based on research done on connection Type 2 (Fig. 3-1(b)), where the initial connection stiffness was equal to 10,000 kip-ft/radian (determined with nonlinear modeling technique Table 5-1). Therefore, it is desired to have a connection stiffness which contributes the maximum amount possible for two reasons. The first reason is because this stiffness correlates with the stiffness where the connection begins to behave similar to a “fixed” connection. The second reason is that it is more economical to increase connection stiffness than increase column or rafter sizes, so it is ideal to get the most benefit possible from the connection.

There are multiple ways to increase the column base connection stiffness with adding little or no material to the connection. Several connection types were explored in both literature review in Chapter 3 and the study conducted in Chapter 5. Various parameters were changed in order to determine what factors most heavily influence the stiffness of the connection. There were two main parameters which had the most effect on the stiffness of the connection. The first parameter was the layout of the bolts. In the literature review a connection with one row of bolts and a connection with two rows of bolts were studied. The connection with two rows of bolts located outside of the column flanges had a stiffness ten times that of the connection with one row of bolts. The connection studied in Chapter 5 also had two rows of bolts, but the bolts were located inside of the column flanges. The stiffness provided by the connection where the bolts were outside of the column flanges had a stiffness more than two times that of the connection where the bolts were inside of the column flanges. Therefore, a large increase in stiffness may be accomplished by using more than one row of bolts and locating those bolts outside of the column flanges. The second parameter was plate thickness and was investigated by Hon and Melchers (1988) but was not explored in further detail in this thesis. From Fig. 3-5(a) when the plate thickness was increased from 0.63 in. to 0.95 the stiffness of the connection doubles, so increasing the plate thickness by 0.32 in. doubled the stiffness of the connection. This is not necessarily a linear correlation, but it is important to note that plate thickness has a large impact on the stiffness of a column base connection. More research should be conducted to further

understand the effect of plate thickness on connection stiffness. These are two methods to increase the stiffness of the column base connection with adding little to no material.

If the goal is to optimize the overall performance of the frame, the plot of the Member SI values vs. Springs Stiffness (Fig. 5-24) can be very beneficial. The connection should be first be optimized based on the recommendations above. Then based on the SI values for each member at the location of the optimized connection stiffness, reduce and increase the size of the members accordingly. For the frame and connection under consideration, it was most economical to increase the size of the column and the rafter framing into the column and reduce the size of the middle rafter. Further investigation should be done with different size frames and different column base connection layouts to better understand optimizing the frame as a whole.

In conclusion, taking into account the actual stiffness of the column base connection can reduce the drift of a frame and increase its economy. Once the connection stiffness has been optimized, the member sizes should be adjusted to improve the overall economy of the frame based on the member sensitivity indices. The stiffness of this connection can be optimized and designed for this optimized stiffness with adding little to no material to the connection.

CHAPTER 6: SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

The purpose of the research conducted for this thesis was first to create an updated version of DISPAR that worked with SAP2000 and second to use DISPAR to understand the behavior of column base connections in metal building frames. The first task required an in depth investigation into both SAP2000's OAPI and OpenGL as a graphics package to determine how these two features could enhance the capabilities of DISPAR. Once this was completed, the next step was to begin programming the updated version using VB.NET. The OAPI included in SAP2000 was designed to work with VBA, so writing the OAPI code using VB.NET was a straightforward process. On the other hand, OpenGL was designed with C# programmers in mind, making that side of the programming a bit more of a challenge. Using SAP2000's OAPI feature, OpenGL for the graphics package, and other updates increased the capabilities and the potential for further updates with DISPAR.

A literature review was completed that covered both analytical and testing models on column base connections, before the affect of modeling the column base stiffness was evaluated. Taking the information from the literature review, a working model which accurately depicted the behavior of the column base connection was developed. Multiple modeling approaches were looked at before one was developed that closely followed the behavior of the connections from the literature review under both moment and axial loading. This modeling approach was then extended to the full metal building frame to understand how explicitly modeling this connection would affect the overall behavior of the frame, with regard to drift. The effect on strength limit states was not considered in this thesis. After using DISPAR to help with understanding the behavior, it was determined that there is a benefit in including the column base connection when modeling the metal building frames.

6.1 DISPAR AS AN ANALYSIS TOOL – PRACTICAL USES

There are many practical uses of DISPAR, it is much more than an education tool to identify the behavior of a given structure. Practicing structural engineers could find many applications where

using DISPAR to analyze their structure would give them an insight into its behavior that they might not have otherwise had. Listed below are a few examples of how DISPAR could be used in a design office:

- Optimization of a structural system using DISPAR output and one of the optimization techniques described in Section 2.4
- Identify how groups of elements, such as the beams on the particular story of a moment frame, behave with the rest of the lateral system
- Identify how combined systems work together, such as a moment frame working with a shear wall
- Understand the behavior of finite elements included in a model, examples could include outrigger beams or shear walls
- Visualize the contribution graphically or in a tabular manner of each element, or groups of elements based on flexural, axial, shear or their sensitivity indices

6.2 COLUMN BASE CONNECTIONS

The standard practice of designers when modeling the metal building frames is to assume pinned connections for the column bases. After investigating the contribution of the column base connection with relation to overall drift of a metal building frame, it was determined that it is in fact conservative to assume a pinned connection. When the frame was subjected to only a lateral load, there was as much as 57.3 percent (Table 5-2) increase in the drift, when comparing the pinned connection to the nonlinear spring approach. Four different modeling techniques were examined to incorporate the behavior of this connection in the frame model. The first technique was using a pinned connection with a rotational spring, where the spring stiffness was equal to that of the rotation stiffness of the connection. This connection was less conservative than assuming a pinned connection but was still conservative when compared with actual behavior. The next approach involved the use of vertical springs in the location of the anchor bolts, which produces similar results to the pinned connection with a rotational spring. The third approach involved assuming a compression block and using vertical springs in the location of the bolts to handle tension. This connection behaved reasonably accurate when compared with actual

behavior as long as bending was inducing compression on the compression block and tension on the vertical springs. The last modeling approach considered used a grid of nonlinear springs to represent the concrete in compression and bolts in tension. This model was determined to be the most accurate when compared to test results and was considered to be close to the actual behavior. It is the author's recommendation that at the very least, a designer should use the pinned support with a rotation spring to account for the stiffness provided by the column base connection

When designing the column base connection, it can be beneficial to design a connection to provide a certain stiffness. This stiffness can be optimized by plotting the connection contribution to overall drift versus the connection stiffness. The optimal stiffness is the stiffness which contributes most to the overall drift of the frame. This stiffness can be reached in a connection by proper bolt layout and plate thickness. After the connection stiffness is optimized, the frame members should be optimized based on the sensitivity indices.

6.3 FINAL RECOMMENDATIONS

When modeling a metal building frame, there is a noticeable decrease in overall drift if the behavior of the column base connection is explicitly modeled. This allows the designer to use less steel to achieve the same drift requirements, resulting in a more economical frame. This benefit should not be ignored by designers. The research conducted in this thesis showed that even a simplistic model for the connection will produce less conservative results than assuming a pinned connection. In order to model this behavior, a designer should first identify the behavior of their particular connection through either testing or the nonlinear modeling technique discussed in chapter 5 of this thesis. From there, include the column base connection in the model with one of the approaches discussed in chapter 5. Even if the designer chooses to use the pinned connection with a rotation spring, a drop in overall drift of approximately 10 percent (Table 5-2, difference in drift with pinned connection and pinned connection with rotational spring) may be seen. It is the author's recommendation to use one of the modeling techniques to explicitly include the column base behavior when modeling metal building frames.

With regard to connection design for stiffness, it is the author's recommendation to use a two row bolt layout where the bolts are located outside of the column flanges. This is a very effective way to increase connection stiffness without having to add any additional material. Also, increasing the thickness of the plate is another effective way to increase the stiffness of the connection. In conclusion, designers should determine an optimal connection stiffness, design the connection for this stiffness, and then model this behavior in design.

6.4 FUTURE RESEARCH

There is a need for future research in both the areas of the development of DISPAR for SAP2000 as well as the modeling of metal building frames. Both of these topics have been researched thoroughly in this thesis, but there are still topics that need to be considered for a complete investigation.

6.4.1 DISPAR

With the use of SAP2000's OAPI functions and OpenGL as the graphics package for the new version of DISPAR, there are further areas of development that can be achieved:

- Mouse clicking – this will enable member selection in DISPAR, which can be linked to information about the member, such as section type, group, and member SI.
- Graphical User Interface (GUI) – could be used to create new members in a system to see how it affects the behavior.
- 2D and 3D plane selection - this would involve using cutting planes and mouse selection.
- Extruded views of members.
- ETABS version – when the OAPI is released for ETABS, DISPAR can be extended to work with both SAP2000 and ETABS.
- Formal Optimization.

6.4.2 METAL BUILDING FRAMES

There are still aspects that need to be researched to develop a more true to life model of metal building frames. Research on the majority of these topics has already begun by another student at Virginia Tech.

- The proper wind load to apply to the frame – the wind load calculations in ASCE 7-05 may be conservative. A computational fluid dynamics (CFD) model should be developed to determine less conservative wind loads to apply to the frame.
- Stiffness associated with the connectors that attach the frame to the diaphragm.
- Additional stiffness to the frame provided by the diaphragm.
- Develop a program to automatically build an accurate model of a metal building frame based on input parameters.
- Further explore the effects of plate thickness on the stiffness of the column base connection.
- Further investigation on different frame sizes and connection types to better understand improving the overall economy of the metal building frames.
- Study different bolt layouts for column base connections to determine an “Optimal Bolt Layout”

REFERENCES

- ASCE. (2005). *Minimum Design Loads For Buildings and Other Structures*. Reston, VA: American Society of Civil Engineers.
- Baker, W. F. (1991). Stiffness Optimization Methods for Lateral Systems of Buildings: A Theoretical Basis. *Electronic Computation*. Indianapolis, Indiana: American Society of Civil Engineers, 269-278.
- Berding, D. C. (2006). Wind Drift Design of Steel Framed Buildings: An Analytical Study and a Survey of the Practice. *Virginia Polytechnic Institute and State University. Department of Civil Engineering. MS Thesis*. Blacksburg, VA.
- Chan, C.-M., and Park, J. (1996). Application Of Structural Optimization to Practical Tall Steel Building Design. *Analysis and Computation* (pp. 123-134). Chicago, Illinois: American Society of Civil Engineers.
- Chan, C.-M., Grierson, D. E., and Sherbourne, A. (1995). Automatic Optimal Design of Tall Steel Building Frameworks. *Journal of Structural Engineering*, 838-847.
- Chan, C.-M., Sherbourne, A. N., and Grierson, D. E. (1993). Stiffness Optimization technique for 3D Tall Steel Building Frameworks under Multiple Lateral Loadings. *Engng Struct., Vol. 16*, 570-576.
- Charney, F. A. (1995). *DISPAR for ETABS version 5.4*. Denver, Colorado: Advanced Structural Concepts.
- Charney, F. A. (1993). Economy of Steel Framed Buildings Through Identification of Structural Behavior. *National Steel Construction Conference*. Orlando, FL: American Institute of Steel Construction, 1-33.
- Charney, F. A. (1990). Sources of Elastic Deformation in Laterally Loaded Steel Framed and Tube Structures. *Design Methods Based on Stiffness*, 893-915.

Charney, F. A., and Marshall, J. (2006). Comparisson of the Krawinkler and Scissors Models for Including Beam-Column Joint Deformations in the Analysis of Moment-Resisting Steel Frames. *Engineering Journal* , 31-48.

Charney, F. A., and Pathak, R. (2007b). Sources of Elastic Deformation in Steel Framed and Framed-Tube Structures: Part 2: Detailed Subassemblage Models. *Journal of Constructional Steel Research* , 1-17.

Charney, Finley A., and Pathak, R. (2007a). Sources of Elastic Deformation in Steel Framed and Framed-Tube Structures: Part 1: Simplified Subassemblage Models. *Journal of Constructional Steel Research* , 1-14.

Computers and Structures. (1995). *ETABS v. 4, SAP90 v. 5* . Computers and Structures Inc. Berkeley, CA.

Computers and Structures. (2008b). *SAP2000 v. 11* . Computers and Structures Inc. Berkeley, CA.

Computers and Structures. (2008a). *SAP2000 API Documentation*. Computers and Structures Inc. Berkeley, CA.

Davis, B. (1996). LRFD Evaluation of Full-Scale Metal Building Rigid Frame Tests. *Virginia Polytechnic Institute and State University. Department of Civil Engineering. MS Thesis* . Blacksburg , Virginia.

Galambos, T. V. (1960). Influence of partial base fixity on frame stability. *J. Struct. Div.*, 85(5) , 85-117.

Hamizi, M., and Hannachi, N. E. (2007). Evaluation by a finite element method of the flexibility factor and fixity degree for the base plate connections commonly used. *Strength of Materials*, 39(6) , 588-599.

Henige, R. A. (1991). Structural Optimization to Limit Natural Periods. *Electronic Computation*. Indianapolis, Indiana: American Society of Civil Engineers, 253-260.

Hon, K. K., and Melchers, R. E. (1988). Experimental behavior of steel column bases. *J. Constr. Eng, 9* , 35-50.

Melchers, R. E. (1988). Modeling of column base behavior. *Connections in Steel Structures: Behavior, Strength, and design* , 151-157.

Microsoft Corporation. (2007). Microsoft Office Excel 2007.

Picard, A., and Dion, J. (1981). Etude Experimental des Assemblages Poteau - Foundation dans les Charpentes d'Acier. *Rapport GCT - 81 - 04, Departement de Genie Civil, Universite Laval, Quebec* .

Picard, B., and Beaulieu, D. (1985). Behavior of a simple column base connection. *Can. J. Civ. Eng., 12* , 126-136.

Salmon, G. C., Schlenker, L., and Johnson, B. G. (1957). Moment Rotation Characteristics of Column Anchorages. *Transactions, ASCE, Vol. 122* , 132-154.

Walsh, J. (2009). Rigid Frame Anchor Bolt Setting. *American Buildings Company* .

Woo, M., Neider, J., Davis, T., and OpenGL Architecture Review Board. (1997). *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.1*. Addison Wesley Publishing Company.

APPENDIX A: USER'S MANUAL FOR DISPAR FOR SAP2000

This appendix is a guide for a user getting started with DISPAR for SAP2000. Included is a list of the features in the program, figures for all of the menu options in the program, and two examples to assist the user on operating the program.

A1. BEFORE GETTING STARTED

It is assumed that the user has a fully developed model of their structure in SAP2000 version 12. Also, that the model includes all necessary load cases for the real portion of the analysis. Before getting started, it is important to apply a 1000 kip load in the direction and location of the displacements of concern, creating a virtual load case pertaining to each applied load. This will be important when selecting load cases for analysis in DISPAR. It is also important to create any element groups in SAP2000 before running the program, DISPAR only recognizes groups which are pre-defined in SAP2000. Additionally, the user must define any frame sections that may be of use when updating groups, DISPAR only recognizes sections which are pre-defined in SAP2000. At this point, DISPAR for SAP2000 is ready for analysis. Make sure to close SAP2000 before opening DISPAR.

A.2 PROGRAM FEATURES

DISPAR for SAP2000 is a post-processor created to optimize the design of both 2-dimensional and 3-dimensional truss and frame structures. The program uses the method of Virtual Work to determine each member's contribution to the total drift of the structure. Each member's contributing factor is known as its **DIS**placement **PART**icipation factor (DISPAR). The element's DISPAR factor is comprised of a major and minor flexural component, a major and minor shear component, an axial component and a torsional component. The DISPAR factor is then divided by the element's volume to obtain its **Sensitivity Index (SI)**, which is a weighted measure of the element's contribution to the total drift. Features provided by DISPAR include:

- The ability to distinguish between different element types, beam, column, or diagonal elements. Along with tabular documentation of the DISPAR totals and percentages for each element type.
- DISPAR can also handle shell elements, by finding a group of shells total contribution to the overall drift.
- Ability to develop DISPAR factors for spring and link elements.
- Tabular and visual representation of each element's DISPAR factors and Sensitivity Indices.
- Ability to handle groups of elements. Tabular documentation of the group's DISPAR totals and percentages. Ability to edit group section types and re-run analysis within the program.
- Accurate method of re-analysis once updates are made without having to re-run the analysis in SAP2000.
- 3-Dimensional viewing of the structure.
- Ability to handle centerline analysis, along with flexible and rigid joint regions.

A.3 GETTING STARTED

The purpose of this section is to show the user how to get started using DISPAR for SAP2000 as well as introduce the various menu options. Included are; the screen the user can expect to see once DISPAR is opened, the file open screen, the load case selection screen, the DISPAR in-progress screen, and the DISPAR results screen. Each screen includes a description of its importance and a short description of menu options on that screen.

A.3.1 DISPAR START-UP AND FILE OPEN SCREENS

When DISPAR for SAP2000 is opened, the screen show in Fig. A.1 will be displayed. Once this screen is displayed, the user should select “*Open*” under the “*File*” dropdown menu to open the desired file for analysis. Next the file open screen will be displayed where the user should select the SAP2000 file to be analyzed, see Fig. A.2.

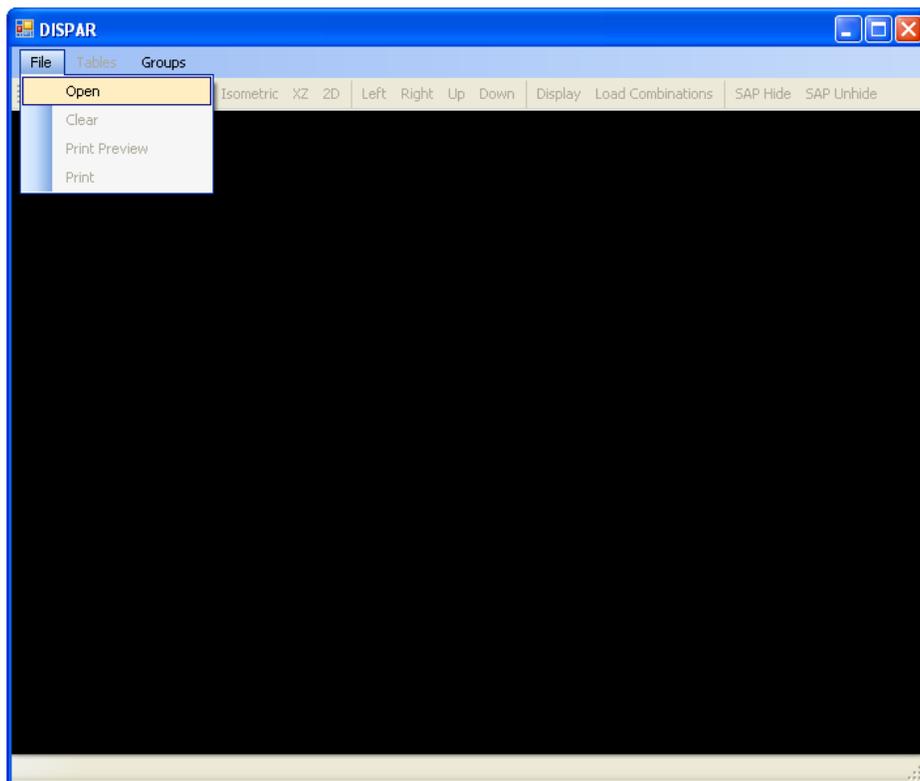


Figure A. 1: DISPAR Start-Up Screen

Menu Options for Fig. A.1 Located Under “File” dropdown menu:

- *Open* - Allows the user to open the desired file for analysis
- *Clear* - Clears the SAP2000 file from DISPAR and allows the user to select another SAP2000 file for analysis
- *Print Preview* - Displays a print preview on the screen of the image to be printed
- *Print* - Allows the user to print the image on the screen

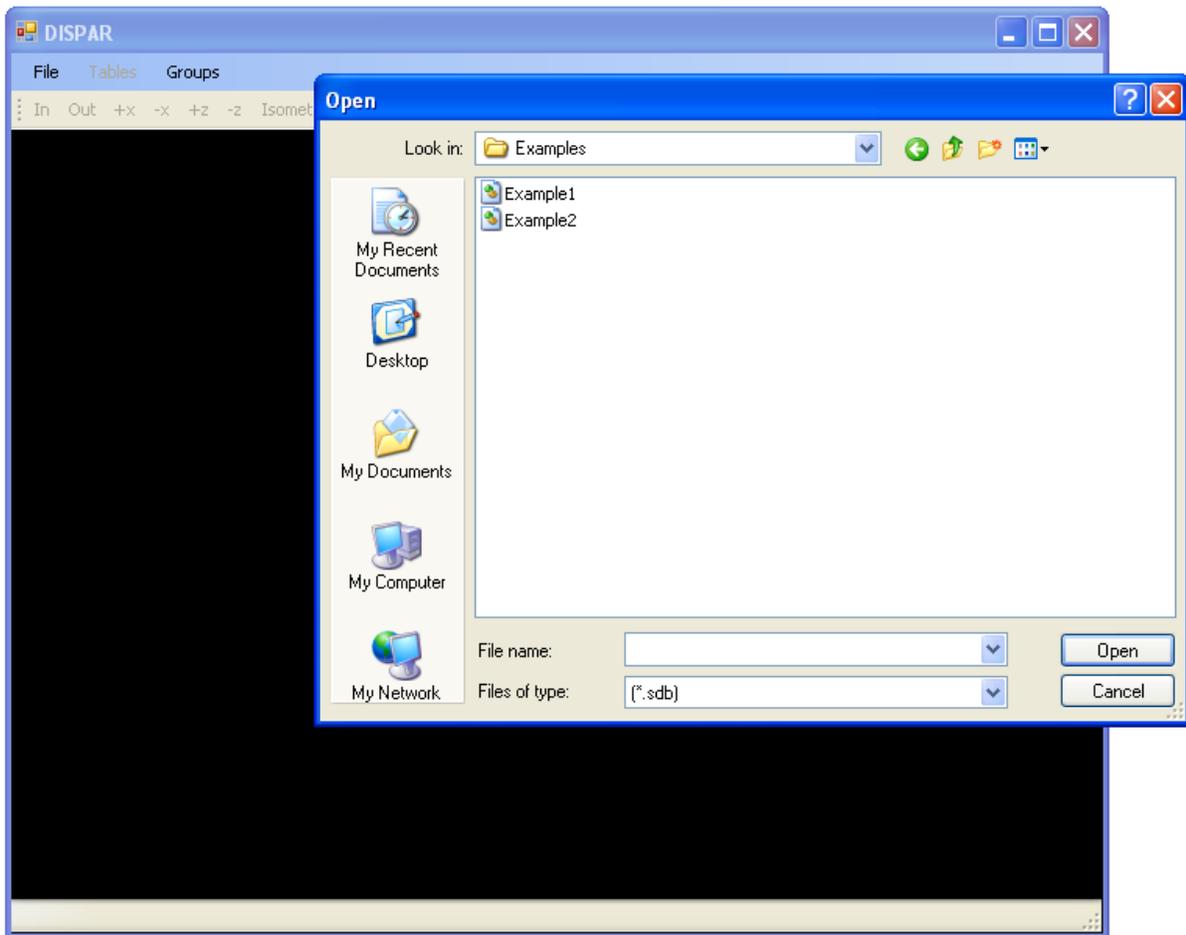


Figure A. 2: File Open Screen

A.3.2 LOAD CASE SELECTION SCREEN

The screen show in Fig. A.3 is what the user should expect to see once a SAP2000 file has been opened in DISPAR. The “*Load Cases*” window is where the user selects the real and virtual load cases which will be used by DISPAR for analysis.

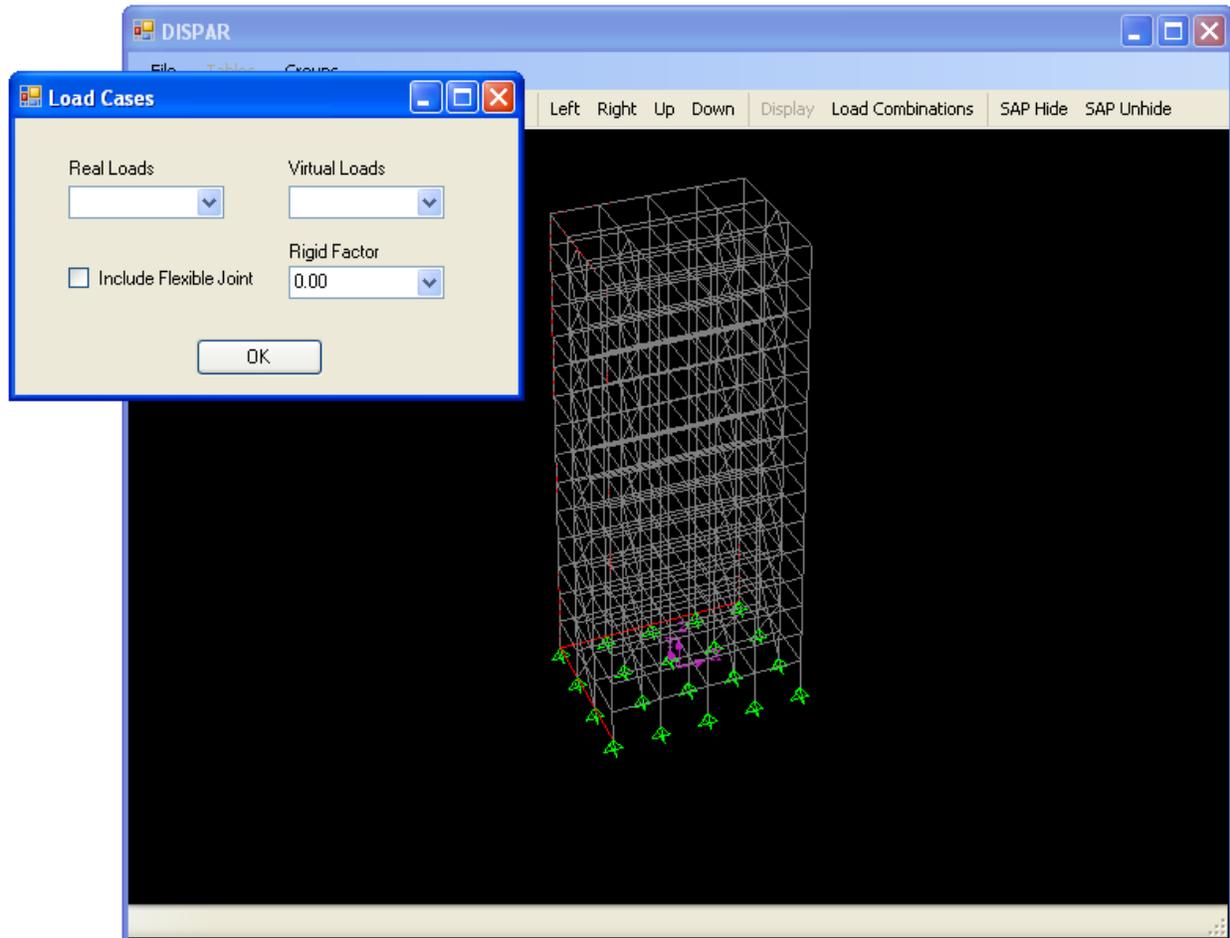


Figure A. 3: Load Case Selection Screen

Menu Options for “*Load Cases*” window in Fig. A.3:

- *Real Loads* - The drop menu where the user selects the real load case for analysis
- *Virtual Loads* - The drop down menu where the user selects the virtual loads for analysis

- *Include Flexible Joint* - By selecting this option, DISPAR will run the analysis assuming the joint region as flexible
- *Rigid Factor* - A dropdown menu where the user can select the portion of the beam-column joint region to be assumed rigid

A.3.3 IN-PROGRESS SCREEN

The screen shown in Fig. A.4 is the screen the user should expect after the SAP2000 file has been selected and after the load cases for analysis have been selected. Notice the “*Status Bar*” and “*Progress Bar*” at the bottom of the screen.

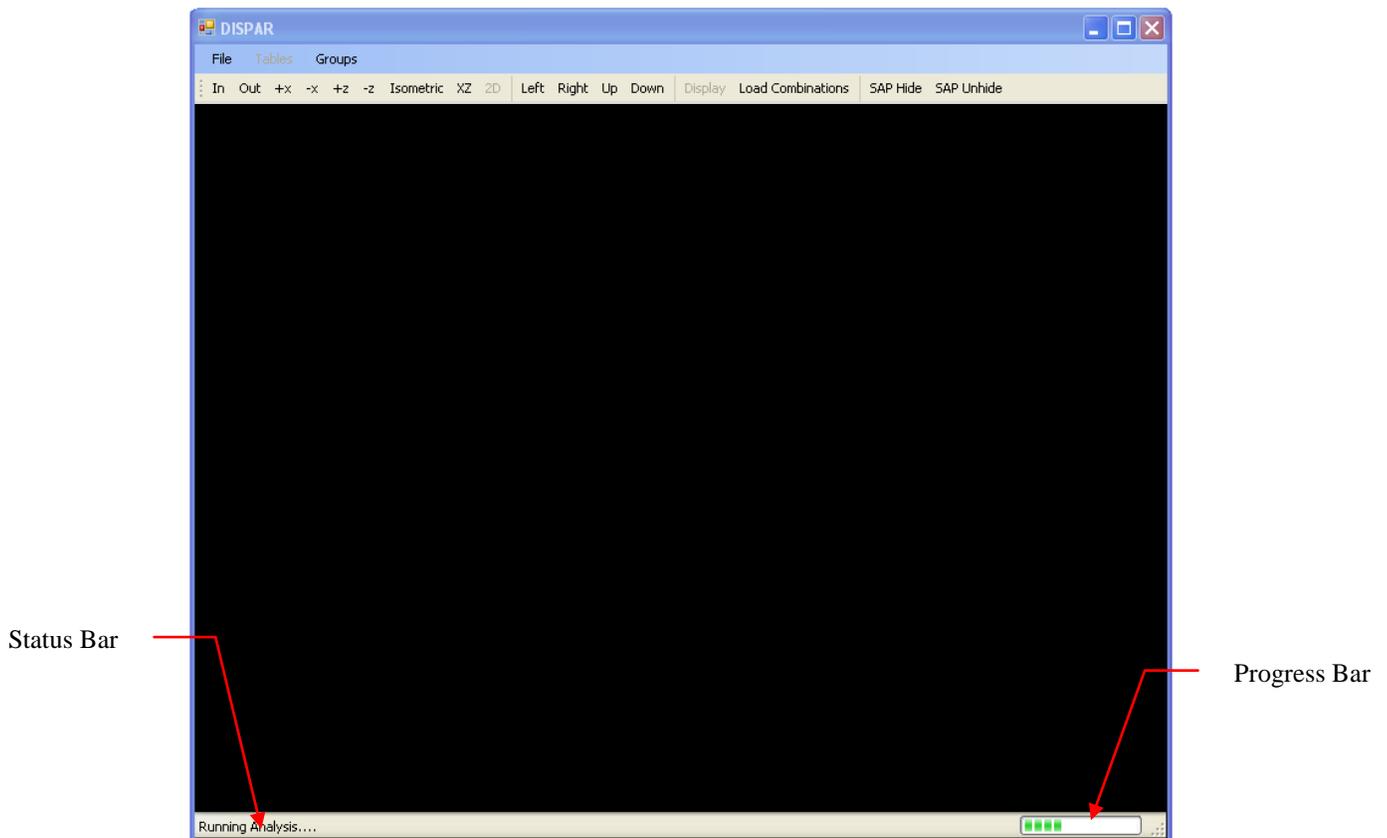


Figure A. 4: DISPAR In-Progress Screen

Menu Options for Fig. A.4:

- *Status Bar* - Informs the user of the current task being completed or the current viewing option selected

- *Progress Bar* - Informs the user of how close DISPAR is to completing its current task

A.3.4 RESULTS SCREEN

The screen show in Fig. A.5 is what the user can expect to see once the load cases have been selected and DISPAR finishes analysis, with the exception that the structure in the viewing screen may be different. This is where the user can view the DISPAR factors for all the elements in tabular or graphical format, change display options, and change member group sizes.

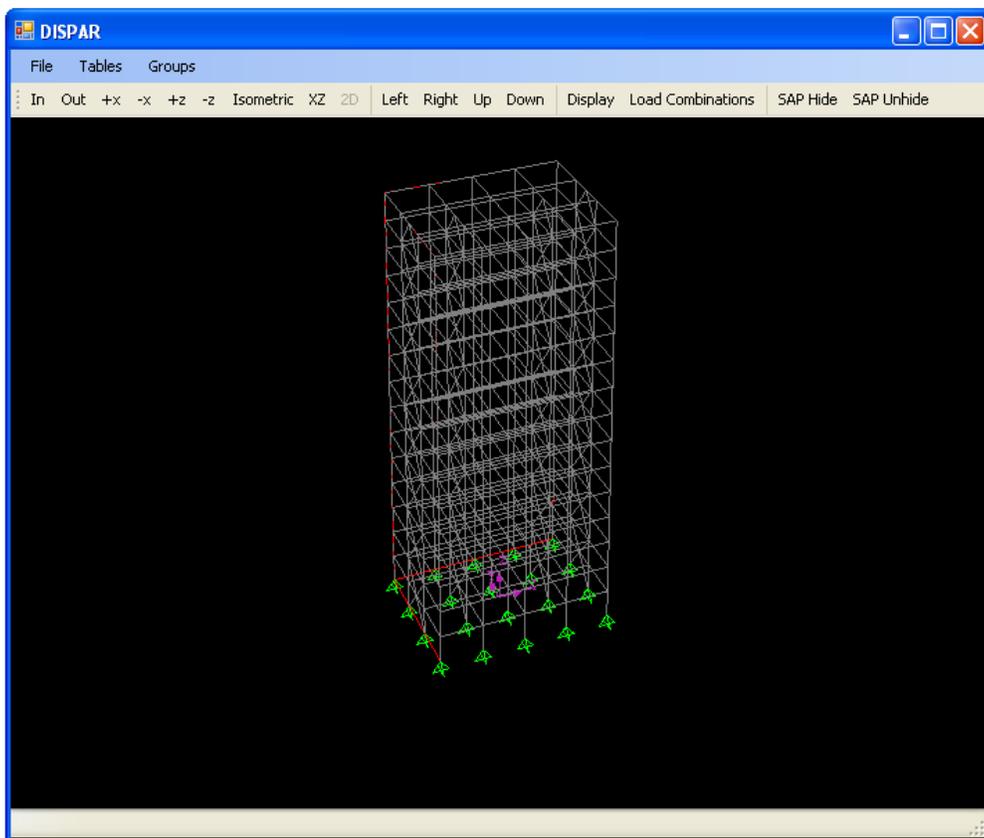


Figure A. 5: DISPAR Results Screen

Menu options for Fig. A.5:

- *Tables* - Dropdown menu allowing the user to select one of the tables available in DISPAR. The tables available are as follows: *Element Properties*, *Real Forces*, *Virtual Forces*, and various *DISPAR* tables. The DISPAR tables include *Factors*,

Totals, Percentages, Group Totals, and Group Percentages. Each DISPAR table allows the user to view the member DISPAR factors in a different way.

- *Groups* - Allows the user to update member group sizes (only available for frame sections)
- *In/Out* - Allows the user to zoom *In* and *Out* of the image on screen
- *+x, +y, +z, -z* - Allows the user to rotate around the x, y, and z axes respectively
- *Isometric* - Returns the image to an isometric view
- *XZ* - Shows the image in the xz plane
- *2D* - Allows the user to select a frame line to view in 2D
- *Left/Right/Up/Down* - Allows the user to pan the image on screen left, right, up, and down respectively
- *Display* - Allows the user to select the display options
- *Load Combinations* - Brings up the “*Load Case*” selection window
- *SAP Hide/SAP Unhide* - Hides or unhides the SAP2000 window

A.4 EXAMPLES

The purpose of this section is to guide the user step by step through two example problems to familiarize them with the program. The first example is of a multi-story frame, showing the use of DISPAR for frame elements. In the second example, shell elements are used with link elements to represent the column base connection. After working through the two examples, the user should have all of the tools necessary to begin analysis using DISPAR for SAP2000. The user should note that the examples are not meant to show how to optimize the stiffness of the structure, they are meant to show the user how to use and access the tools in DISPAR.

A.4.1 EXAMPLE 1 – 3D FRAME

This example will show the user how group elements can be used to update member sizes based on their contribution to overall drift. Groups were only created for elements on the first two stories to show how they can be utilized. If this structure was actually being analyzed, it would be beneficial to create groups for elements on all stories of the structure. In addition, Example 1 goes through the steps to set view controls as well as displaying tables. Group member sizes are then updated based on the information provided by the different views and the information from the tables. Once this is complete, the example shows the accuracy of DISPAR's re-analysis tool. Now begin Example 1.

Model Description

- 15 story office building, floor to floor height of 13 ft, 25 ft bays in each direction.
- Moment frames in the x-direction, 2 story X-brace in the y-direction, see Fig. A.6 for a plan view.
- Groups have been established in SAP2000 for the moment frame beams on the first 2 stories (MF_BM_1), moment frame columns on the first 2 stories (MF_COL_1), braced frame braces on the first 2 stories (BF_BR_1) and the braced frame columns on the first 2 stories (BF_COL_1). These groups will come in handy when upsizing the members in the example below.
- 30 psf wind load in both directions.

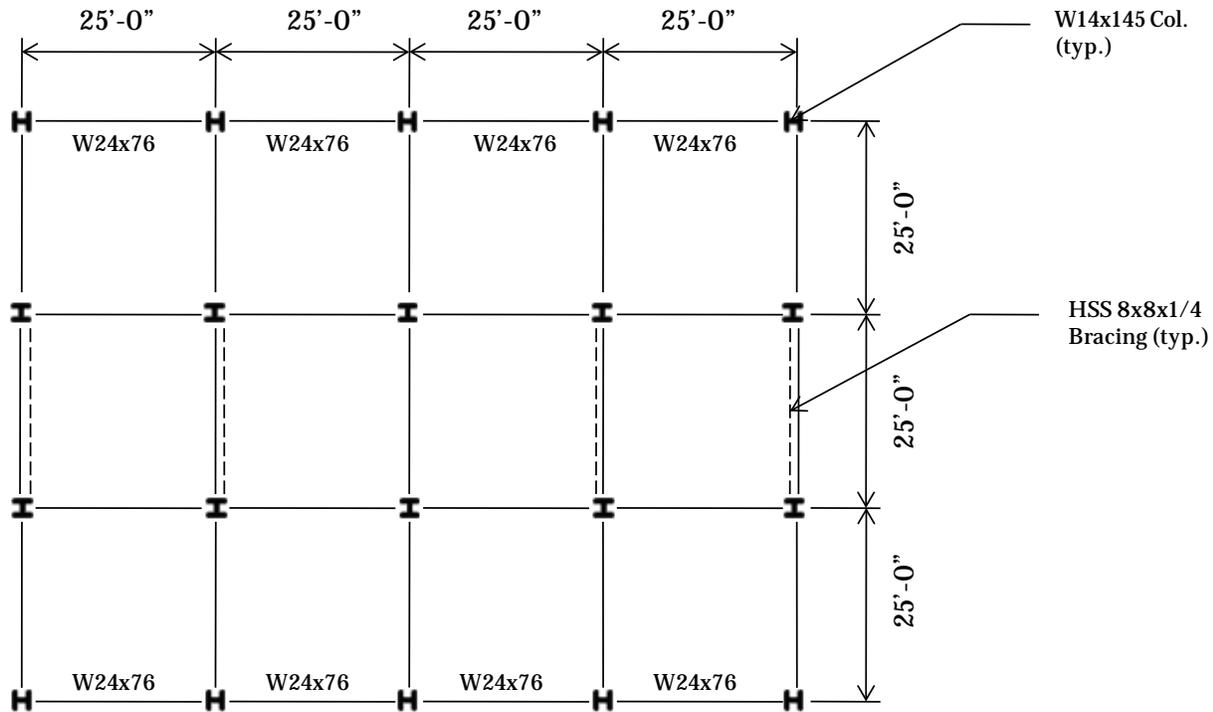


Figure A. 6: Typical Floor Plan

Running DISPAR and viewing results (NOTE: if DISPAR unexpectedly closes, make sure to re-open DISPAR and close it before proceeding, otherwise an error will result in SAP)

1. Open DISPAR for SAP2000.
2. Using the “Open” button under the “File” dropdown menu, open the file *Program Files\VirginiaTech\DISPAR for SAP2000\Examples\Example 1*.
 - a. Notice that SAP2000 automatically opened in the background, which is gathering the necessary information to run DISPAR.
 - b. Once SAP2000 has opened the model, DISPAR will draw the structure, which may be viewed in 3D.
3. Notice that the “Cases” window has opened, Fig. A.7. Select “Real-X” under the “Real Loads” section and select “Virtual-X” under the “Virtual Loads” section.

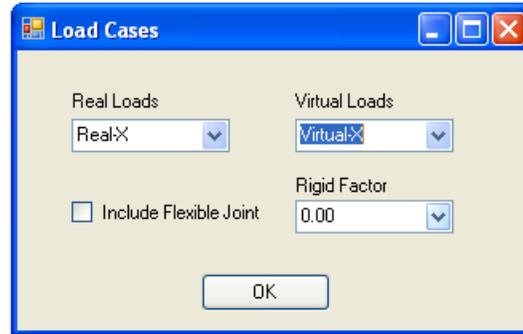


Figure A. 7: Load Cases Window

- a. The real and virtual load cases for analysis are the real and virtual cases in the direction of the drift in concern.
 - b. Click the “*Include Flexible Joint*” option if you want to include the joint region along with selecting the rigid end zone factor. If this option is not selected the program will run a centerline analysis. This is the assumption made by DISPAR, not SAP2000.
4. Click “*OK*”.
 5. Once the “*Load Case*” window has disappeared, SAP2000 has finished running the analysis and all the DISPAR factors and SI’s have been calculated and stored in tables.
 6. To view the model in 3-D, zoom in or out, rotate about the x or y-axis, select the appropriate button located on the menu strip, see Fig. A.8. Various other viewing controls are located under “*Display*” on the menu bar, such as a shrunken view of each element or displaying member names. The “*Display*” options may be found in Fig. A.9.



Figure A. 8: Menu Strip

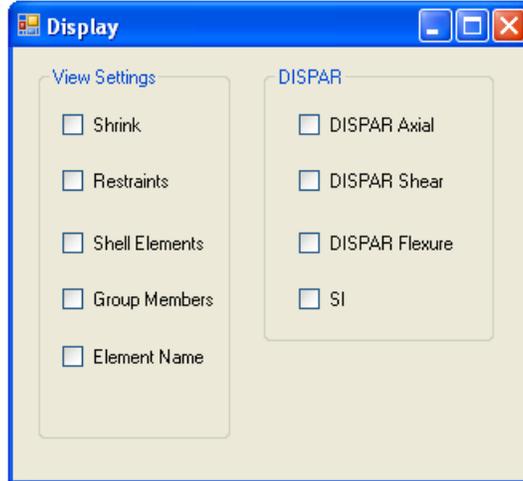


Figure A. 9: Display Menu Options

7. While the “*Display*” options table open, select “*Shrink*” and “*Group Members*”. This will show the names of group members and the shrunken view of the frame elements. Also, with the “*In*” and the “*Up*” button, zoom into the first two stories to get a better view of the group members. See Fig. A.10 for an image of the viewing area after these steps have been followed.

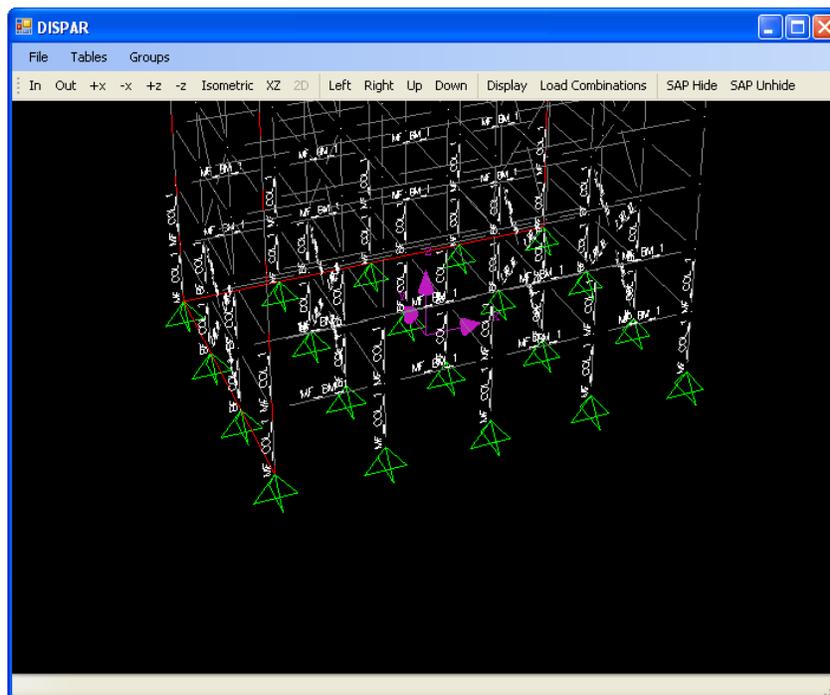


Figure A. 10: DISPAR Viewing Screen with Display Options

- To view the member's DISPAR factors or SI's, click on the "Display" button on the menu strip. Select the "SI" option. Notice in Fig. A.11, that the moment frame beams on the first level are contributing most to the drift in the x-direction.

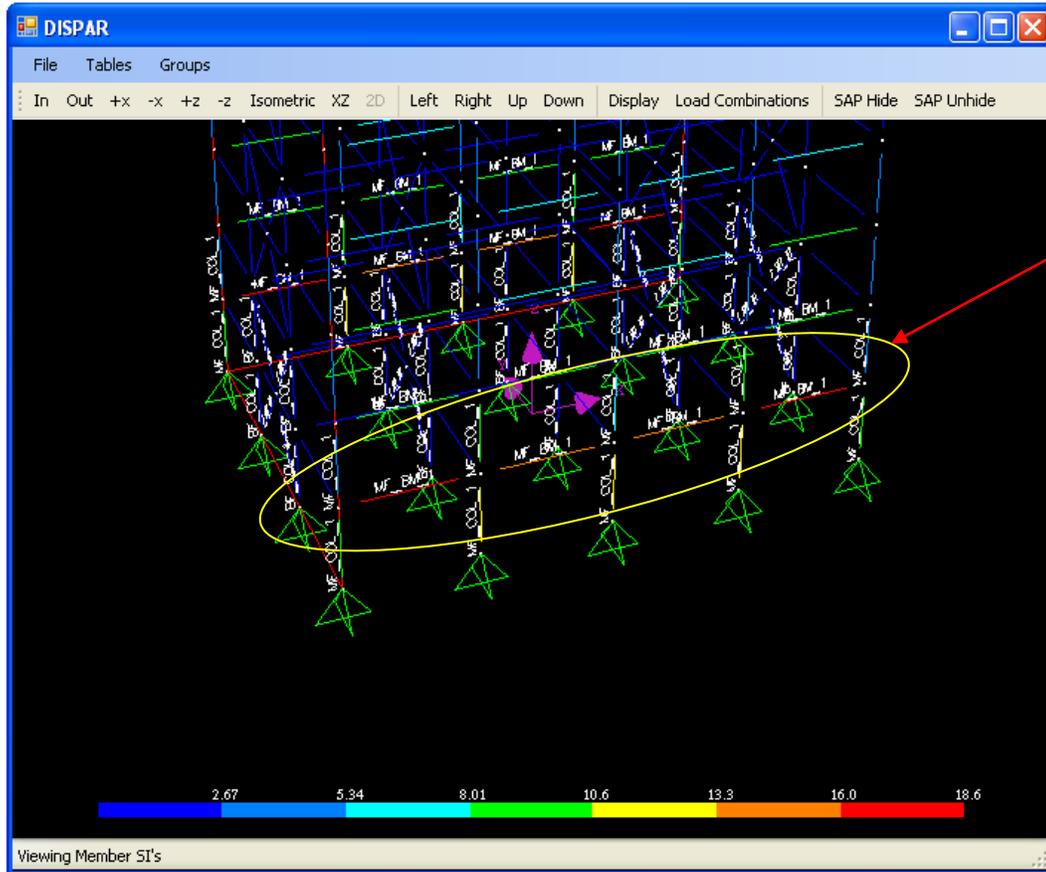


Figure A. 11: SI Viewing

- All of the images on the DISPAR viewing screen may be printed by selecting "Print" under the "File" dropdown menu. The user may select the printer to print with once "Print" has been clicked. In order to view of print preview of the image, click "Print Preview" under the "File" dropdown menu.
- In order to view an individual frame in either the XZ plane or the YZ select the "XY" button on the menu tab, then select the "2D" button on the menu tab. The window in Fig. A.12 will appear. Simply select the plane and the frame line you wish to view and click "Set Viewing". (Note: the frame lines in the 2D Viewing window are perpendicular to the plane you select.)

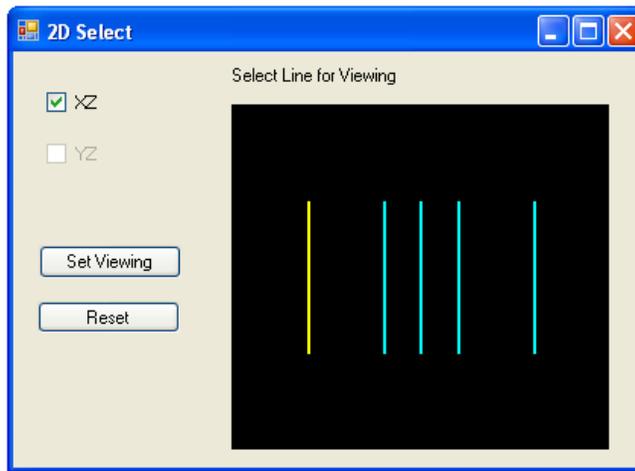


Figure A. 12: 2D Viewing Selection

11. To view the output tables, click on the “*Tables*” drop-down menu and select the desired table.
 - a. For example, select the “*Percentages*” table under “*DISPAR*”. Notice in Table A.1 that the Beams account for 55% of the drift and the columns account for 45%. This indicates that the beams contribute more to 10% more to the drift than do the columns.
 - b. The group DISPAR factors may also be viewed by selecting “*group totals*” or “*group percentages*” under “*DISPAR*”. This is helpful when determining how a preselected group is affecting the overall drift. Notice the bar charts for number of members in each group when exporting the “*group totals table*” to Excel.
12. In order to view this table in Excel format, go to “*File*” dropdown menu on the table and click “*Export Table to Excel*”.

Table A. 1: Member DISPAR Percentages

Member Type	Axial	Flexural Maj	Flexural Min	Shear Maj	Shear Min	Joint	Total
Columns	4.27	33.73	1.14	5.70	0.00	0.00	44.87
Beams	0.00	51.60	0.00	3.50	0.00	0.00	55.13
Diagonals	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shell	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Totals	4.27	85.30	1.10	9.30	0.00	0.00	100.00

13. Now, let’s assume that the drift in the x-direction is of interest, and that it must be limited to $L/400 = 5.85$ in.

- a. In order to view the overall drift, go to “*Tables/DISPAR/Totals*”. The total of all of the DISPAR factors, found in the lower right corner of the table, is the total displacement at the location of the load times 1000.
- b. For this example, the location of the virtual load is at the roof level. The total displacement is the total of the DISPAR factors divided by 1000. Therefore, the total displacement at the roof level in the x-direction is equal to 7.13 in., found at the bottom right corner of Table A.2.

Table A. 2: Member DISPAR Totals

Member Type	Axial	Flexural Maj	Flexural Min	Shear Maj	Shear Min	Joint	Total
Columns	304.83	2405.39	81.08	408	0.7	0	3200.03
Beams	0	3678.1	0	253.2	0	0	3931.26
Diagonals	0.02	0	0	0	0	0	0.02
Shell	0	0	0	0	0	0	0
Totals	304.85	6083.5	81.1	661.2	0.7	0	7131.31

14. Since the moment frame beams located at the bottom of the structure have the highest sensitivity indices, we will upsize those beams first. This can be determined by viewing the SI’s in Fig. A.11 or looking up the group SI’s in under “*Tables/DISPAR/Group DISPAR Totals*”, seen in Table A.3.

- a. Go to “*Edit Group Sections*” under the Groups button on the menu strip, see Fig. A.13.

- b. Select “MF_BM_1” from the dropdown list, notice that the group’s current section is displayed in the adjacent textbox.
- c. Select “W30x116” from the list of sections, then click “Update”.
- d. The beams in the group “MF_BM_1” have been updated to a “W30x116”. To view this, repeat steps a and b, see the current section is a “W30x116”.

Table A. 3: Group DISPAR Totals

Group Name	Number in Group	Axial (in.*1000)	Flexural Maj (in.*1000)	Flexural Min (in.*1000)	Shear Maj (in.*1000)	Shear Min (in.*1000)	Joint (in.*1000)	Total (in.*1000)	Group SI*1000
MF_COL_1	20	144.41	970.9	0	102.7	0	0	1218.06	182.86
BF_COL_1	20	0.04	32.2	55.5	1.6	0.4	0	89.77	13.48
BF_BR_1	16	0.01	0	0	0	0	0	0.01	0
MF_BM_1	16	0	1305.7	0	89.9	0	0	1395.57	207.67
Totals		144.459991	2308.8	55.5	194.2	0.4	0	2703.41016	404.01

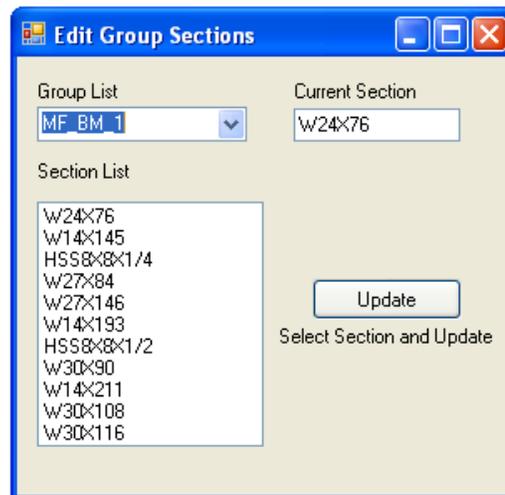


Figure A. 13: Edit Group Sections

15. DISPAR will recalculate the drift based on the new properties of the elements. This is an approximate method, but since the drift is controlled primarily by equilibrium the accuracy of the calculation is within reason. After a couple trial runs we will re-run the analysis to retrieve the updated factors.
16. Notice that the total drift is 6.35”, still above the target value of 5.85” (recall from step 10 how to find the total drift).

- a. View the member SIs (step 7) or the “*Group DISPAR Totals*” table (step 13) and notice that the lower level columns have the highest SI.
 - b. Repeat step 11, but this time selecting group “*MF_COL_1*” from the dropdown list and “*W14x211*” from the section list.
17. The total drift is now 5.92”, which is approximately equal to the limit of $L/400 = 5.85$ ”.
18. Now re-run the analysis and compare the results with the calculated values. After re-running the analysis DISPAR returns a drift of 5.87”, which is within 1% of the calculated value.
19. Close DISPAR.

Note: This example is not meant to show the most efficient design for the above structure. It is meant to show the overall process and how it works.

A.4.2 EXAMPLE 2 – USING SHELL ELEMENTS

This example will look at DISPAR's basic ability to handle shell and link elements using the virtual work technique to determine a displacement participation factor. At this point in time, DISPAR only has the ability to calculate a total DISPAR factor for a given group of shell elements. This can be helpful in determining the contribution of each lateral system when using a combined system of steel framing and a shear wall. This example will also show how creating groups for the shell elements can be helpful to understand the behavior of members in a system. It will also show the link element DISPAR factors and how those add into the total drift. The example will go through the analysis of the metal building used for the sensitivity analysis discussed in Chapter 5.

Model Description:

- 12 ft tall by 65 ft wide metal building frame
- Peak height of 16 ft
- 10 kip load applied at the left and right column rafter joints
- Groups created based on Fig. 5-23
- Nonlinear link elements used for the column base connection with a rotational stiffness of 2100 kip-ft/radian

Running DISPAR

1. Open DISPAR for SAP2000.
2. Using the "Open" button under the "File" dropdown menu, open the file "*Program Files\VirginiaTech\DISPAR for SAP2000\Examples\Example 2*".
3. Follow steps 3 through 6 in the previous example, making sure to select the "Real" and "Virtual" load cases. Fig. A.14 shows the viewing of the shell elements in the example.

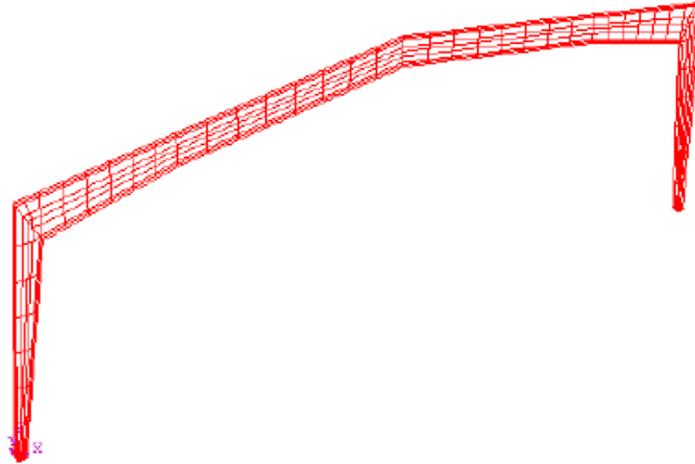


Figure A. 14: Model with Shell Elements

4. To view the frame and link member's contribution to the Drift, select the "Totals" table under "Tables/DISPAR", see Table A.5. Notice the contribution for all of the other members are 0. This is because only shell and link elements are included in this model.

Table A. 4: Shell Member DISPAR Totals

Member Type	Axial (in.*1000) /Fx	Flexural Maj (in.*1000) /Mz	Flexural Min (in.*1000) /My + Mx	Shear Maj (in.*1000) /Fy	Shear Min (in.*1000) /Fz	Joint (in.*1000)	Total (in.*1000)
Columns	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Beams	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Diagonals	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Shell	0.0	0.0	0.0	1904.0	-257.7	0.0	1646.3
Spring	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Link	257.7	0.0	0.0	0.0	0.0	0.0	257.7
Totals	257.7	0.0	0.0	1904.0	-257.7	0.0	1904.0

5. To view the contribution of the shell elements grouped together to represent members in the frame select the “*Group Totals*” table under “*Tables/DISPAR*”, see Table A.5. At this time the group member SI’s are only available for frame elements, this feature has not been added for shell elements.
6. Close DISPAR.

Table A. 5: Shell Member Group Totals

Group Name	Number in Group	Axial (in.*1000)	Flexural Maj (in.*1000)	Flexural Min (in.*1000)	Shear Maj (in.*1000)	Shear Min (in.*1000)	Joint (in.*1000)	Total (in.*1000)
Column1	89.00	-10.26	758.00	0.00	-470.20	0.00	0.10	278
Girder1	94.00	9.24	148.40	0.30	223.90	0.10	-0.10	382
Girder2	106.00	0.50	120.00	-0.30	170.80	0.00	0.00	291
Girder3	113.00	1.41	122.00	-0.30	169.30	0.00	0.00	292
Grider4	91.00	5.27	184.00	0.30	89.50	0.00	-0.10	279
Column2	89.00	-5.98	571.60	0.00	-431.10	-0.10	0.10	135
Totals	0.00	0.18	1904.00	0.00	-257.7	0.00	0.00	1646

APPENDIX B: CREATING DISPAR FOR SAP2000 – GUIDE TO GETTING STARTED IN OPENGL WITH VB.NET

B.1 INTRODUCTION

Appendix B is a guide which explains how the graphics package OpenGL was used for graphics in DISPAR for SAP2000, and which explains how to use OpenGL with Microsoft Windows and VB.Net. It includes information on how to link OpenGL with the .Net framework and how to perform simple tasks once the two are linked together. These tasks include; initialize drawing windows, working with multiple windows, using the glut library, drawing figures and text, setting viewing controls, and other features in OpenGL. OpenGL is a very powerful graphics package and can be used to create almost any scene imaginable, but without an understanding of how to get started it would be very difficult to use. This section will focus on how to get started using OpenGL. The more difficult topics such as shading and texturing (not used in DISPAR) are covered in The Official Guide to Learning OpenGL (Woo et al. 1997). This guide can be found online at www.glprogramming.com/red, it is also referred to as “The Red Book”. The Red Book is a very useful guide when trying to understand how to use the commands in OpenGL, as well as information on what is going on in the background of OpenGL.

B.2 THE TAO FRAMEWORK

The Tao Framework is a set of dynamically linked libraries (dll) which link OpenGL to the .Net framework, which is the network under which VB.Net operates. Without the Tao Framework the user will not have access to commands to perform tasks in OpenGL. The libraries which are important when using OpenGL with VB.Net include:

- *Tao.Opengl*
- *Tao.FreeGlut*
- *Tao.FreeType*
- *Tao.Platform.Windows*

All of these libraries can be downloaded from the Tao Framework website, which is www.taoframework.com. It is very important to reference these files at the start of the program and make sure they are imported. Also, make sure all of the files are found in the “*debug*” and “*release*” folders located in the “*bin*” folder in the program, see Fig. B.1 and Fig. B.2.

```
Imports Tao.OpenGl
Imports Tao.FreeGlut
Imports Tao.FreeType
Imports Tao.Platform.Windows
```

Figure B. 1: Reference Files in Program



Figure B. 2: Files Located in Release Folder

If the files are not referenced properly VB.Net will not recognize any of the functions in OpenGL or the OpenGL Utility Toolkit (Glut) library. Notice that *freeglut.dll* is not present in Fig. B.1, this is because it is not necessary to use `Imports` for both *Tao.FreeGlut* and *freeglut.dll*. The files listed in Fig. B.2 are a set of dynamically linked libraries (dll) which allow for the commands of each library to be used in VB.Net.

B.3 INITIALIZING A WINDOW

There are multiple ways to initialize a drawing window in OpenGL. When a drawing window is created in VB.Net, this window is not linked to OpenGL. Therefore, any images that are drawn in OpenGL will not show up on this drawing window. To have the images draw onto the drawing window in VB.Net, this window must be initialized in OpenGL. This section will cover two

ways of doing so. The first and simplest way is to use the Glut library, which is also the more limited of the two ways. When using the Glut (OpenGL Utility Toolkit) library to create a window, it creates a window which is separate from the forms used in VB.Net. A form in VB.Net is the main window that all of the menu bars, drawing windows, and other features are attached to. If the goal of the program is simply to draw images or scenes, this is the way to go. On the other hand, if the goal of the program is to include dropdown menus and other user functions, this is not the way to go about creating a drawing window. The other way to create a windows drawing area is to use an existing window in the user form and enable that window as an OpenGL viewing area. This way is slightly more complicated, but allows the user to create multiple viewing windows in the same form as well as use the other object oriented functions in VB.Net. This was the method used in creating DISPAR for SAP2000, but both methods will be covered in this section.

B.3.1 USING THE GLUT LIBRARY

When creating a window using the Glut library, it is only a matter of setting the display mode of the window, window size, window position, and the name of the window. This is done using the commands:

glutInitDisplayMode (Glut_Single | Glut_RGB)

glutInitWindowSize (width, height)

glutInitWindowPosition (x , y)

glutCreateWindow ("Window")

where *Glut_Single* creates a single buffered window; *Glut_RGB* sets the color field; width and height are just that; x and y are the coordinates of the top left corner of the window on the computer screen; and "Window" is the assigned name to the window. Once the window has been created, it is time to begin drawing shapes and text in the window.

B.3.2 USING VB.NET WINDOW

This method of initializing a window requires that the programmer knows how to link a windows drawing area to an OpenGL drawing area. The two important points that the programmer must understand are how to set the pixel format of a window and the Windows Graphics Layer. Each of these are vital to linking the OpenGL drawing area to a drawing area created in the .Net framework. Setting both parameters properly allows the program to run multiple windows at the same time along with having use of the object oriented functions which can be used in a form.

B3.2.1 PIXEL FORMAT DESCRIPTOR

The Pixel Format Descriptor, or *pfid*, is a list of parameters which sets the pixel format of the window. In other words, the pfd sets the number of pixels used for coloring along with layer properties of the window. This is the first step in linking the initialized OpenGL drawing window to a VB.Net window. Not doing so will result in a blank window. The pixel format of a window is set using the following commands:

```
pfid.nSize = Len (pfid)  
  
pfid.nVersion = 1  
  
pfid.dwFlags = Gdi.PFD_SUPPORT_OpenGL Or  
Gdi.PFD_DRAW_TO_WINDOW Or  
Gdi.PFD_DOUBLEBUFFER  
  
pfid.iPixelFormat = Gdi.PFD_TYPE_RGBA  
  
pfid.cColorBits = 32  
  
pfid.cDepthBits = 32  
  
pfid.iLayerType = Gdi.PFD_MAIN_PLANE
```

where *pdf* is the *PIXELFORMATDESCRIPTOR*, *Len(pdf)* is the size of the pixels used in drawing, and *Gdi* stands for Graphics device interface (described in the next section). The rest of the commands are outlined and described in Table B.1.

Table B. 1: Pixel Format Controls

Command	Description
PFD_SUPPRORT_OPENGL	Sets OpenGL as the viewing window being set
PFD_DRAW_TO_WINDOW	Tells the program to draw to a window and not a bitmap
PFD_DOUBLEBUFFER	Tells OpenGL to DoubleBuffer, OpenGL renders of screen, the buffers are the swapped to on screen
pdf.iPixelFormat	Specifies the colors being used
PFD_TYPE_RGBA	Sets colors being specified to red, green, and blue
pdf.cColorBits	Sets the number of color bits per pixel
pdf.cDepthBits	Sets the number of depth bits, this sets up the depth buffer for OpenGL, which is used to determine how far an object is from the screen
pdf.iLayerType	Currently the only layer type supported in PFD_MAIN_PLANE

B.3.2.2 WINDOWS GRAPHICS LAYER

In order to link the OpenGL drawing window with the VB.Net window, the rendering context of the drawing window must be set to the viewing context of the viewing window. The rendering context is the drawing information of the given window. In Microsoft Windows, the Graphics Device Interface, or *Gdi*, uses a device context to remember settings for a given window about drawing modes and commands. The *Gdi* is only used to remember the drawing settings and OpenGL is used to render or draw. This process is done using the following commands:

- *wgl.createcontext (hDC)*
- *wgl.deletecontext (hRC)*
- *wgl.makecurrent (hDC, hRC)*

where *hDC* is the window's device context and *hRC* is the windows rendering context. The *createcontext* command creates a handle in OpenGL context while being passed like it is a *Gdi* device context. The delete context does just that, it deletes the rendering. This must be done once the rendering context is no longer in use. The *makecurrent* command makes a rendering context the current rendering context, where the device context must have the same pixel format as the rendering context. This step is important whenever initializing a window for a first time or when using multiple windows.

Now all of the tools have been presented which are necessary to create a window using both the Glut library as well as using a window in a VB.Net form. As can be seen, the first methods is much quicker and easier, requiring less understand of the settings involved in linking windows. The second method is more time consuming and requires a deeper understanding, but allows the programmer far more freedom with the graphics presentation. It is recommended that all OpenGL graphics windows be initialized using the second method.

B.3.3 USING MULTIPLE WINDOWS

In order to work with multiple windows it is simply a matter of applying the method described in Section B.3.2 to all of the windows desired. When drawing to a specific window, that window needs to be made current using the *wgl.makecurrent* command. It is important to retrieve the device context settings of a window only when initially creating the window or if the window size changes. If this is done too often, the program will produce an error. Once finished with a window, always delete the rendering context using *wgl.deletecontext*. Using multiple windows at the same time allows for multiple different drawings or scenes to occur simultaneously, enhancing the effects of the graphics.

B.4 WORKING WITH THE GLUT LIBRARY

The Glut library can be very helpful when trying to perform tasks in OpenGL. Many tasks which may take a number of OpenGL commands to execute, such as creating a sphere, take only one command with the Glut library. To create a sphere using OpenGL drawing commands it is necessary to create a large number of small three dimensional cubes together to form the sphere. The more cubes used, the better the sphere. In order to create the same sphere using the Glut library, the command *glutSolidSphere()* is used. Recall from the previous section the steps for creating a drawing window using Glut commands compared with using the rendering context settings. A lot of time and programming can be saved using the Glut commands, but the limitations of each command must be understood before using them. The Glut library can also be used for things like mouse - clicking and keyboard functions, where OpenGL does not have the capability.

To use the Glut library, the command *glutinit (argc, argv)* must be called, where *argc* and *argv* are system variables. The Glut commands will not be recognized if the library is not initialized properly. Additionally, if the library is initialized more than once, the commands will not work properly. Once the Glut library has been initialized, the commands can be called. It is not necessary to disable the Glut library once it is no longer needed.

B.5 DRAWING IN OPENGL

Before drawing an image in OpenGL, it is important to understand the process of drawing and viewing in OpenGL. With OpenGL, two different processes are occurring at the same time, an image is being displayed on the screen while another is being created. An animation is created if an OpenGL window is continuously updated multiple times every second with the next image in a sequence of images. Think of this as a flipbook where the images being updated on the OpenGL window are the different pictures scene when flipping through the book. The typical movie projector is updated 24 times per second, or at 24 frames per second. This type of

animation was not necessary when creating the images for DISPAR, but could be beneficial if it is desired to animate the modes shapes or deflected shape of a structure.

Each time the image is updated, it is important to have the correct viewing settings and to clear the screen back to the default color. If this not done correctly, the images presented on the screen may not look the way it was designed to. Once the settings have been established, the image may be drawn in OpenGL. After the image has been drawn, the initialized OpenGL window should be updated. At this point, the image drawn in OpenGL will appear on the drawing window located on the VB.Net form.

Listed below are the steps required to draw a line in OpenGL:

Initial Screen Settings:

1. Before an image may be drawn it is important to set the screen clear. The initial screen color is white, but if another background screen color is desired it is important to set it. To set the screen clear color use the *glClearColor(red as single, green as single, blue as single, alpha as single)*, see Fig. B.3 for color selection. *Alpha* is a parameter only used when setting the clear color and should be set to 0.
2. Set the *Matrix Mode* to operate in. There are two different options to choose from:
 - a. *GL_PROJECTION* – This should be called before working with viewing transformations.
 - b. *GL_MODELVIEW* – This should be called before working with drawing transformations, such as shape translations and rotations.
3. Use the *glLoadIdentity* command to make sure the current *Matrix Mode* is loaded.

Viewing Settings:

4. Next, set how the image will be viewed on the screen. This can be done using two dimensional viewing or three dimensional viewing options. How to actually set the viewing options will be covered in section B.6.

Display Settings:

5. It is important to initialize drawing settings and clear the previous settings. Otherwise the image will not appear as intended. A few things to consider are:
 - a. *glEnable (GL_DEPTH_TEST)* – When viewing an drawing in three dimensions, this tests whether the object being drawn has anything else in front of it. If so, only the portions visible are drawn.
 - b. *glClear (GL_COLOR_BUFFER_BIT)* – Clears the window to the set clear color.
 - c. *glClear (GL_DEPTH_BUFFER_BIT)* – Clears the previous depth buffer. This is important when working in three dimensions, otherwise OpenGL gets confused on the depth to draw objects.

Drawing:

6. Set the color of the object to be drawn using *glColor3f (red as single, green as single, blue as single)*. Each blank is filled in with a number between 0 and 1, based on the amount of each color to include, see Fig. B.3 on how to choose color.

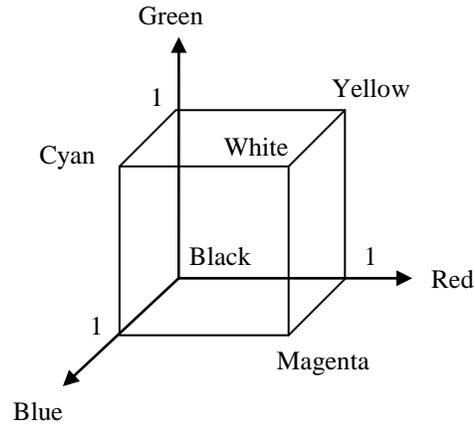


Figure B. 3: Color Diagram

7. Begin drawing the object. The specifics of drawing a shape will be covered in Section B.5.1.
8. Use the *glflush* command to ensure that the drawing commands are initialized.
9. Swap the buffers. In other words, update the image on the VB.Net drawing window with the OpenGL image using the *Gdi.SwapBuffers (current hDC)* command.

B.5.1 DRAWING GEOMETRIC SHAPES

Before getting started with the commands to draw objects in OpenGL it is important to note that any of the OpenGL commands used in VB.Net must be called with the prefix (*Gl.*), similarly Glut commands must be called with the prefix (*Glut.*). So for instance, if the OpenGL command for draw was *glDraw(GL_SHAPE)*, what will actually be programmed is *Gl.glDraw(Gl.GL_SHAPE)*. Doing this lets VB.Net know which library is being referenced for a given command.

There are a few steps required to draw points, lines, or polygons in OpenGL. First, the type of object to be drawn must be selected and then initialized using the *glBegin (mode)* command. Where the mode is the object being drawn, which is selected from a list available in the OpenGL command library. For instance, the command for drawing a line is *GL.GL_LINE*. Once the *glBegin* command has been used, it is time to assign the vertices of the given object. Until the *gl.end* command is used, OpenGL will continue to draw the object specified with the assigned properties (i.e. color and size/width) for all of the vertices listed, as long as they coincide with the number required to draw the object (2 for a line, 3 for a triangle, and 4 for a rectangle). In other words, if the line command is used and more than two vertices are listed, OpenGL will treat each two vertices after the first two as a new line. Similarly for triangles and rectangles, OpenGL will draw another shape for the more than three and four vertices listed, respectively. For points the size of the point may be set using the *glPointSize* command, and for lines the width may be set using the *glLineWidth* command. These settings should be called before the *glBegin* command is called.

Below is a code sample for drawing two red lines, with a line width of 2, from coordinates (0, 0, 0) to (1, 0, 0) and (1, 0, 0) to (1, 1, 0):

```

Gl.glColor3f(1, 0, 0)
Gl.glLineWidth(2)
Gl.glBegin(Gl.GL_LINES)
    Gl.glVertex3f(0, 0, 0)
    Gl.glVertex3f(1, 0, 0)
    Gl.glVertex3f(1, 0, 0)
    Gl.glVertex3f(1, 1, 0)
Gl.glEnd()

```

A similar approach should be taken when drawing points and polygons. Some useful objects commands may be found in Table B.2.

Table B. 2: Drawing Commands

Command	Description
GL_POINTS	Draws a point at each of the vertices listed
GL_LINES	Draws a line between each set of two vertices listed
GL_TRIANGLES	Draws a triangle between each set of three vertices listed
GL_QUADS	Draws a quadrilateral between each set of four vertices listed
GL_POLYGON	Draws an arbitrary polygon between all of the vertices listed. There must be above three vertices and the lines should not intersect each other.

Note: Before drawing a polygon, the *glPolygonMode(face, mode)* command should be called. The *face* parameter controls which side (the front or back) to color and the *mode* parameter selects whether the polygon is colored in or not. For example to have the front side of the polygon colored in and back side of the polygon outlined the follow should be called:

glPolygonMode (GL_FRONT, GL_FILL)

glPolygonMode (GL_BACK, GL_LINE)

To set the colors of each side, the *glColor3f* command must be called before the *glPolygonMode* is set for that particular side. The parameter *mode* may be set as *GL_FILL*, *GL_LINE*, or *GL_POINT*. The default setting is to fill in both sides of the polygon with the last color called.

B.5.2 DRAWING TEXT

There are multiple ways to draw text in OpenGL, this section will focus on the two methods used in DISPAR. Each of the methods has its benefits and drawbacks. Both methods are limited on the fonts that can be used. The first method uses glut bitmap fonts and is particularly useful when drawing text in two dimensions. The bitmap font is only able to draw text horizontal and facing forward. It is also limited in the number of font sizes available. In addition, the text using bitmap fonts cannot be manipulated, as in scaled, rotated, or translated. This method is useful when the text is meant to always face forward. The second method is using glut stroke characters, which allows for the rotation and scaling of text. This method is helpful when working in three dimensions and it is desired to have the text line up with objects.

The text must be drawn one letter at a time using the integer associated with the given letter (character). The sample code below shows how to draw text using both bitmap fonts and well as stroke characters in VB.Net.

Using Bitmap Fonts:

```
'Text to draw
strng = "Text"

'Location to draw
Gl.glRasterPos3d(x, y, z)

'Length of the text as an integer
length = Len(strng)

'For loop to draw the text on character at a time. AscW is a command
in VB.Net to retrieve the character for the given letter i.
For i = 1 To length

    Glut.glutBitmapCharacter(Glut.GLUT_BITMAP_TIMES_ROMAN_24,
        AscW(Mid$(strng, i, i)))

Next i
```

Using Stroke Characters:

```
'Text to draw
strng = "Text"

'Translating the text to the desired location
Gl.glTranslatef(x, y, z)

'Rotating the text based on the desired angles (angx, angy, angz)
Gl.glRotated(angx, 1, 0, 0)
Gl.glRotated(angz, 0, 0, 1)
Gl.glRotated(angy, 0, 1, 0)

'Scaling the text to the desired screen size
Gl.glScaled(0.2, 0.2, 0.2)

'Length of the text as an integer
length = Len(strng)
```

```

'A for loop similar to that of the Bitmap Font
For i = 1 To length
    Glut.glutStrokeCharacter(Glut.GLUT_STROKE_ROMAN, AscW(Mid$(strng,
        i, i)))
Next i

```

In both cases x , y , and z are the coordinates where it is desired to draw the text. The parameters x , y , and z represent the bottom left corner of the string of text. The variables $angx$, $angy$, and $angz$ are angles that the text is rotated.

B.6 SETTING VIEW CONTROLS

As with many other things in OpenGL, there are multiple ways to set the view controls. The view controls can be related to setting up a camera and the different settings are where the camera is located, where it is pointed, and the limits of viewing. The image drawn on the screen is essentially a snapshot taken by the camera, given the settings assigned. There are two different options that will be covered in this section, Orthographic Projection and Perspective Projection. Orthographic Projection basically sets up a box around a given viewing area and relative distance has no effect on the size an object is drawn. This view control was used for two dimensional viewing in DISPAR, it requires little input and provides helpful views of a given frame. Perspective Projection requires multiple settings but is very useful when working in three dimensions. It is also useful for rotation and zoom features.

B.6.1 TWO DIMENSIONAL VIEWING

In DISPAR, Orthographic Projection was used because an object's distance away from the camera does not affect how it is drawn on the screen. The settings required for Orthographic Projection are the left and right limits, top and bottom limits, and near and far limits. For a given frame, if the viewing area is set to the limits of that frame, the only thing that will show up on the drawing screen is a two dimensional drawing of that frame. See Fig. B.4.

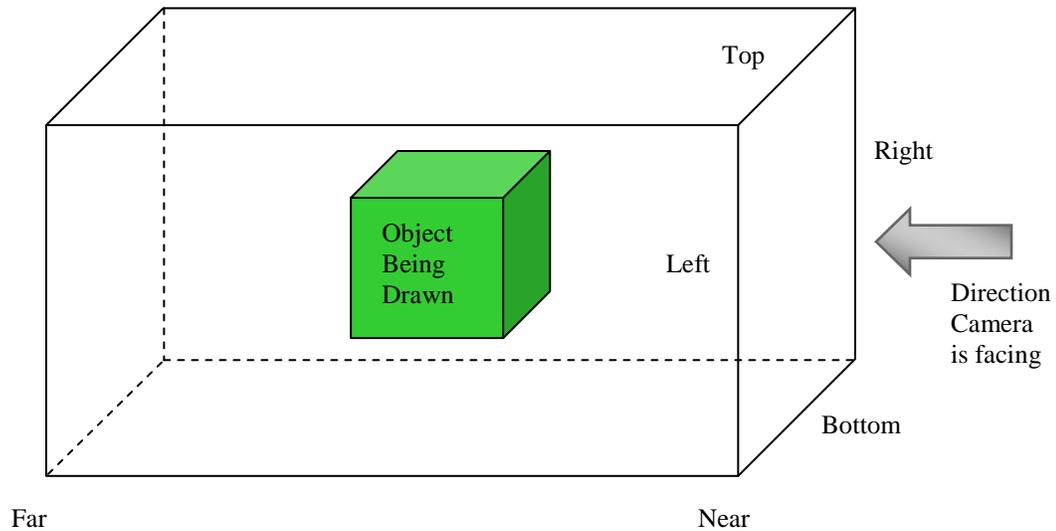


Figure B. 4: Orthographic Projection

The command for setting up this type of viewport is *glOrtho(left, right, bottom, top, near, far)*. If an object does not lie within the bounds of the Orthographic Projection box they will not show up in the drawing. Setting the camera location and viewing direction is the same for both two dimensional and three dimensional viewing and is explained in the next section (B.6.2).

B.6.2 THREE DIMENSIONAL VIEWING

When viewing an object in three dimensions, there are a more parameters that need to be considered other than just the limits. Something to keep in mind is that OpenGL works in a coordinate system where up is in the y direction and z is in and out of the page, see Fig. B.5. Setting this type of viewing is much more like setting up a camera to take a picture. First, the camera settings must be specified, which include how wide the viewing eye is open, also known as the *field of view* (*fovy*, measured in degrees), the near and far distances from the camera location as well as the aspect ratio of the viewing area. This is done using the *glPerspective (field of view, aspect ratio, near distance, far distance)*, see Fig. B.6.

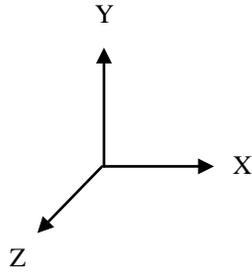


Figure B. 5: Coordinate System

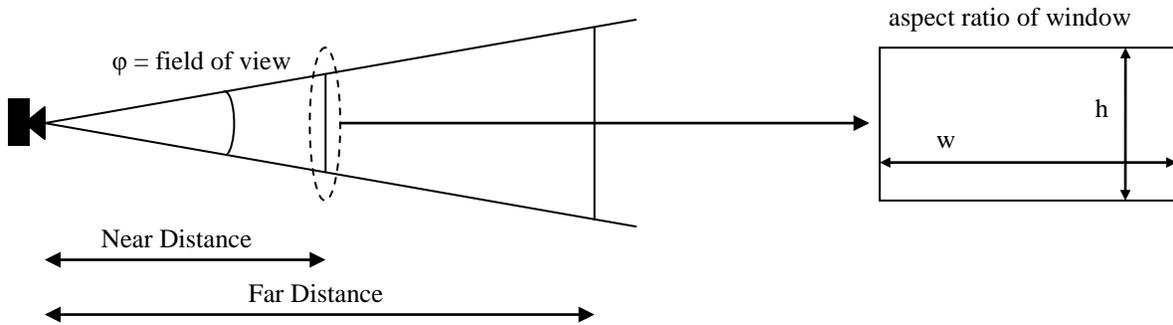


Figure B. 6: Perspective View

Once the camera settings have been specified, it is necessary to establish where the camera is located and where it is looking. This is done using the *gluLookAt()* command. The parameters for this command are the camera location (x, y, z), where the camera is looking (x, y, z), and an up direction vector ($vector_x, vector_y, vector_z$). An up direction vector of $(0, 1, 0)$ would have the camera looking straight up. The commands should be called in this order:

glPerspective (field of view, aspect ratio, near distance, far distance)

gluLookAt (eye x, eye y, eye z, center x, center y, center z, up x, up y, up z)

The eye location consists of the three dimensional coordinate where the eye is located, and the field of view is how wide the eye is open. In order to view the image, the location of where the eye is looking must be set and the up direction is simply which direction is up for the camera. If the viewing is set too wide the image on the screen appears very small, conversely, if the viewing angle is set too small the image on the screen appears very large and portions are cut off. Therefore, the angle must be set correctly, such that the desired amount of the drawing is part of the image in the viewing window. In addition, the up direction becomes important when rotating the image. If the wrong up direction is set, the image will show up as being upside down in the viewing window. Once the eye parameters have been set, the eye takes a snapshot, which is then displayed in the viewing window.

It is important to remember that the view settings should be specified in the order described in Section B.5 after selecting *glMatrixMode (GL_PROJECTION)*. In order to zoom in and out of the image or rotate around in, it is simply a matter of changing the parameters in the *gluLookAt* so that the camera is in the desired location, see Fig. B.7.

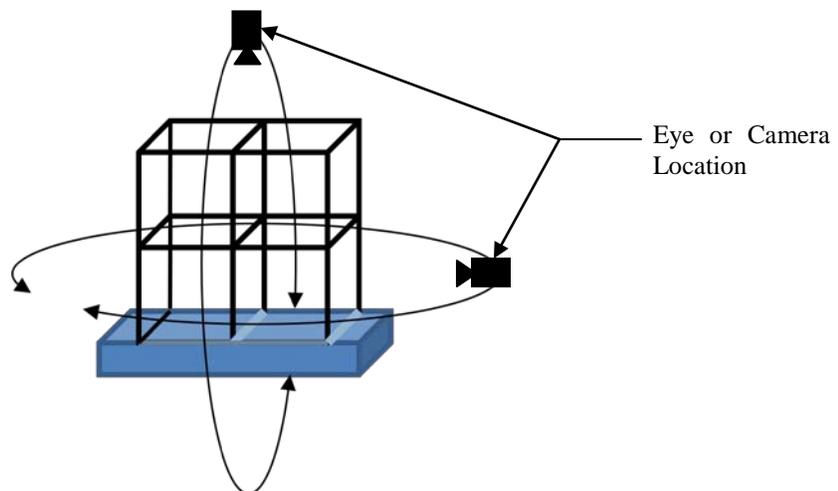


Figure B. 7: Rotation Figure

B.7 BRIEF OVERVIEW OF ADDITIONAL OPENGL FEATURES

Included in this section is a list of additional features in OpenGL that were either not covered in this chapter or were not used in the development of DISPAR. For more information on any of the features see The Official Guide to Learning OpenGL (Woo et al. 1997).

- **Stripped Lines** – these are different line types which may be used, consisting of various dashed and dotted linetypes.
- **Modeling Transformations** – These were mentioned in the chapter but not discussed in any great detail. Modeling transformations are a way to translate and rotate the objects being drawn to any point in space.
- **Clipping Planes** – Allow further refined of the viewing area beyond the left, right, bottom, top, near, and far viewing restrictions. This can be useful for section cuts or limiting the viewing plane to only look at certain frames or members.
- **Lighting** – Allows the user to set a light source in OpenGL, giving the scene a sense of real life. Multiple lighting sources in a given scene may be used
- **Material Properties** – May define a material property which will affect how the object reflects light.
- **Fog** – The use of fog makes the image appear more realistic, having objects further in the background fading out.
- **Texturing** – Gives objects a more realistic look by assigning them a texture. Can be done using an image of the real object, then applying it to an image.

B.8 PUTTING IT ALL TOGETHER - EXAMPLE: DRAWING A SIMPLE 3D FRAME AND TEXT USING TWO VIEWING WINDOWS

This section includes a simple example on how to create an OpenGL project using VB.Net. The example is intended to show how all of the steps covered in this chapter fit together. After understanding this example program, the reader should have all of the tools needed to start using OpenGL with VB.Net. Included in the example is the use of two windows created using the

windows graphics layer, drawing a three dimensional frame with multiple colors, drawing text using Bitmap Fonts, viewing using Perspective Projection and Orthographic project. The example also includes buttons which control the drawing and clearing of each window. The source code for the example is include in Section B.8.3.

B.8.1 CREATING THE FORM

Before getting started with the programming, first create the user form which may be found in Fig. B.8. The labels found in Fig. B.8 refer to the names assigned to the different objects on the VB.Net form. “*Window1*” is the drawing area for the three dimensional frame created using the line commands, along with different colors used for the girders and columns to show depth. The “*btn_DrawFrame*” button, is a button which upon clicking draws the frame, similarly “*btn_DrawText*” draws the Bitmap Font text of “3D Frame”. Both windows may be cleared using the “*btn_ClearWindow1*” button or the “*btn_ClearWindow2*” bottom, associated with their respective windows.

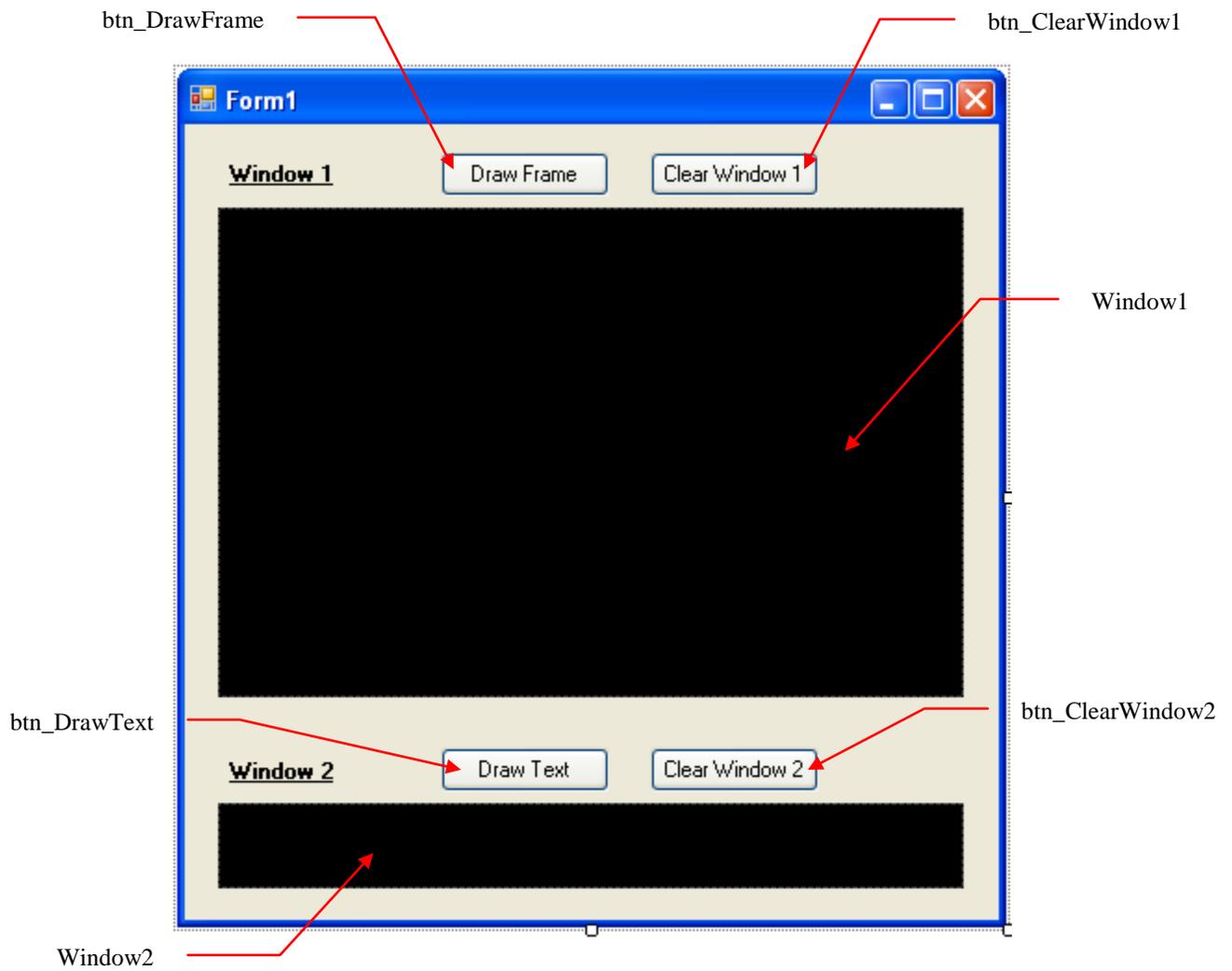


Figure B. 8: Form Design for Example

B.8.2 PROGRAM RESULTS

The form found in Fig. B.9 is what the windows should look like after the program has been run and both of the draw buttons have been clicked. Notice how the front girder, in red, has been drawn overtop of the rear columns, in yellow. This was done by the enabling the depth test, discussed in step 5(a) of Section B.5. The columns were intentionally drawn before the girders in this example, so if the depth test was not enabled the rear columns would be drawn over the girders.

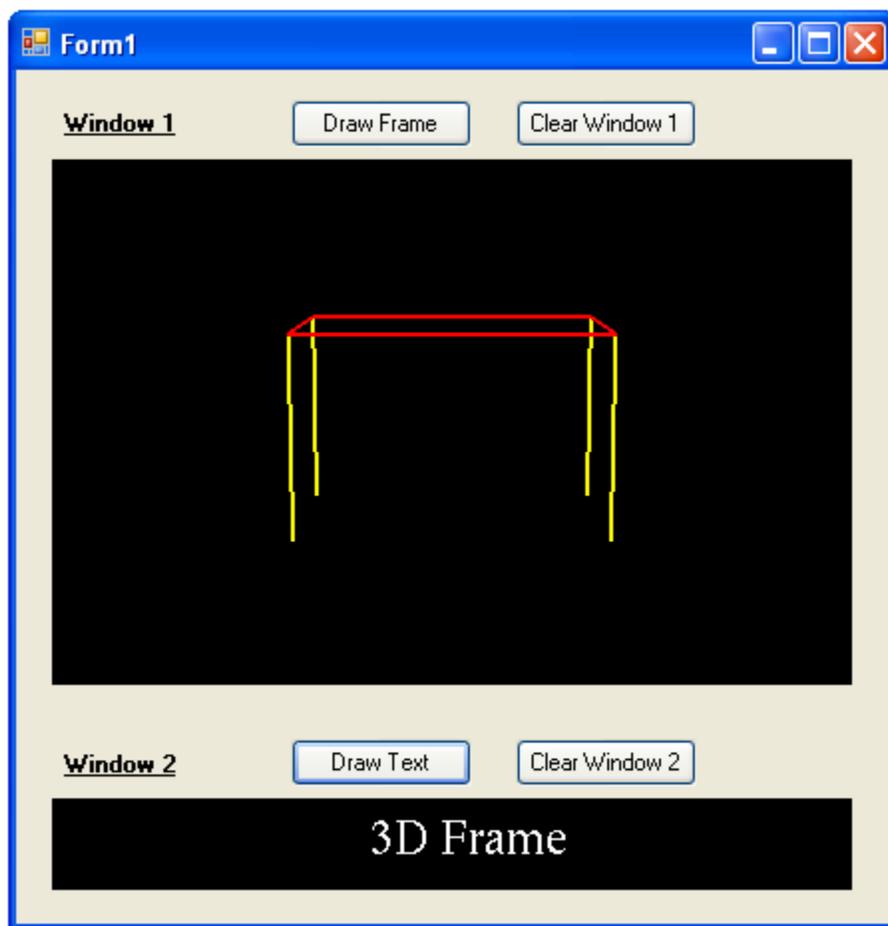


Figure B. 9: Completed Example Program Form

B.8.3 SOURCE CODE FOR EXAMPLE

Included in this section is the source code for the example exactly as it is found in VB.Net. In order for this code to work the files discussed in Section B.2 must be referenced properly under as described in Section B.2. Otherwise, if the form is set up properly, the code should work properly.

Source Code:

```
Imports Tao.OpenGl
Imports Tao.FreeGlut
Imports Tao.Platform.Windows

Public Class Form1
    'Drawing Window Settings Variables
    Dim hDC As System.IntPtr
    Dim hDC2 As System.IntPtr
    Dim hRC As System.IntPtr
    Dim hRC2 As System.IntPtr
    Dim pfd As Gdi.PIXELFORMATDESCRIPTOR
    Dim PixelFormat As Integer

    'GLUT Variables
    Dim argcp As Integer
    Dim argv() As System.Text.StringBuilder

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'Initlialize GLUT Library
        Glut.glutInit(argcp, argv)

        'Get Window Device Context
        hDC = User.GetDC(Window1.Handle)
        hDC2 = User.GetDC(Window2.Handle)

        'Create Pixel Format
        CreatePixel()

        'Create Window1
        CreateWindow1()

        'Create Window2
        CreateWindow2()

    End Sub

    Public Sub CreatePixel()

        'Create Pixel Format
        pfd.nSize = Len(pfd)
        pfd.nVersion = 1
    End Sub
End Class
```

```

        pfd.dwFlags = Gdi.PFD_DRAW_TO_WINDOW Or Gdi.PFD_SUPPORT_OPENGL Or
Gdi.PFD_DOUBLEBUFFER
        pfd.iPixelFormat = Gdi.PFD_TYPE_RGBA
        pfd.cColorBits = 32
        pfd.cDepthBits = 32
        pfd.cStencilBits = 32
        pfd.iLayerType = Gdi.PFD_MAIN_PLANE

        'Fill the variable PixelFormat - Since both windows have the same
pixel format it only
        'needs to be assigned once
        PixelFormat = Gdi.ChoosePixelFormat(hDC, pfd)

End Sub

Public Sub CreateWindow1()

    'Set Pixel Format
    Gdi.SetPixelFormat(hDC, PixelFormat, pfd)

    'Create Window
    hRC = Wgl.wglCreateContext(hDC)

End Sub

Public Sub InitWindow1()

    'Make Current Window1
    Wgl.wglMakeCurrent(hDC, hRC)

End Sub

Public Sub CreateWindow2()

    'Set Pixel Format
    Gdi.SetPixelFormat(hDC2, PixelFormat, pfd)

    'Create Window
    hRC2 = Wgl.wglCreateContext(hDC2)

End Sub

Public Sub InitWindow2()

    'Make Current Window 1
    Wgl.wglMakeCurrent(hDC2, hRC2)

End Sub

Public Sub DrawText()
    'Dim Drawing Variables
    Dim length, i As Integer
    Dim strng As String

    'Initialize OpenGL Viewing
    Gl.glClearColor(0.0, 0.0, 0.0, 0.0) 'Sets clear color as black

```

```

Gl.glMatrixMode(Gl.GL_PROJECTION) 'Sets matrix mode to projection
Gl.glLoadIdentity() 'initializes projection matrix
Gl.glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0) 'Sets 2D viewing window

'Display
Gl.glClear(Gl.GL_COLOR_BUFFER_BIT) 'Clears the screen

'Draw Text
Gl.glColor3f(1, 1, 1) 'set color of text
strng = "3D Frame" 'text to be drawn
Gl.glRasterPos3d(0.4, 0.4, 0) 'position to draw text
length = Len(strng) 'length of text
For i = 1 To length
    'draws one letter at a time using characters (AscW)
    Glut.glutBitmapCharacter(Glut.GLUT_BITMAP_TIMES_ROMAN_24,
AscW(Mid$(strng, i, i)))
Next i

Gl.glFlush() 'Forces drawing commands to begin excution

'Swap Buffers
Gdi.SwapBuffers(hdc2) 'Swaps buffers to actually draw on screen

End Sub

Public Sub DrawFrame()

'Initlialize OpenGL Settings
Gl.glClearColor(0, 0, 0, 0)
Gl.glMatrixMode(Gl.GL_PROJECTION)
Gl.glLoadIdentity()

'Set Viewing Position
Glu.gluPerspective(25, 1, 1, -10) 'Sets viewing angle, window ratio,
near and far viewing distance
Glu.gluLookAt(0, 0.5, 3, 0, 0, 0, 0, 1, 0) 'Sets the location of the
eye, where it is looking, and up direction

'Display
Gl.glEnable(Gl.GL_DEPTH_TEST) 'Enables the Depth Test, Important
for 3D viewing
Gl.glClear(Gl.GL_COLOR_BUFFER_BIT) 'Clears window
Gl.glClear(Gl.GL_DEPTH_BUFFER_BIT) 'Clears previous depth buffer

'Draw Frame
Gl.glColor3f(1, 0, 0) 'Sets the color of the beams as red
Gl.glLineWidth(2) 'Width of the line drawn
Gl.glBegin(Gl.GL_LINES) 'Begins the line command
Gl.glVertex3f(-0.25, 0.25, 0.25) 'Beginning point of first line
Gl.glVertex3f(0.25, 0.25, 0.25) 'End point of first line
Gl.glVertex3f(0.25, 0.25, 0.25)
Gl.glVertex3f(0.25, 0.25, -0.25)
Gl.glVertex3f(0.25, 0.25, -0.25)
Gl.glVertex3f(-0.25, 0.25, -0.25)
Gl.glVertex3f(-0.25, 0.25, -0.25)

```

```

    Gl.glVertex3f(-0.25, 0.25, 0.25)
    Gl.glEnd() 'Ends line command

    Gl.glColor3f(1, 1, 0)
    Gl.glLineWidth(2) 'Sets column color to yellow
    Gl.glBegin(Gl.GL_LINES)
    Gl.glVertex3f(0.25, -0.25, -0.25)
    Gl.glVertex3f(0.25, 0.25, -0.25)
    Gl.glVertex3f(-0.25, -0.25, -0.25)
    Gl.glVertex3f(-0.25, 0.25, -0.25)
    Gl.glVertex3f(-0.25, -0.25, 0.25)
    Gl.glVertex3f(-0.25, 0.25, 0.25)
    Gl.glVertex3f(0.25, -0.25, 0.25)
    Gl.glVertex3f(0.25, 0.25, 0.25)
    Gl.glEnd()

    Gl.glFlush()

    'Swap Buffers
    Gdi.SwapBuffers(hdc)
End Sub

Public Sub ClearWindow1()

    'Set Clear Color and Clear
    Gl.glClearColor(0.0, 0.0, 0.0, 0.0)
    Gl.glClear(Gl.GL_COLOR_BUFFER_BIT)

    'Swap Buffers
    Gdi.SwapBuffers(hdc)

End Sub

Public Sub ClearWindow2()

    'Set Clear Color and Clear
    Gl.glClearColor(0.0, 0.0, 0.0, 0.0)
    Gl.glClear(Gl.GL_COLOR_BUFFER_BIT)

    'Swap Buffers
    Gdi.SwapBuffers(hdc2)

End Sub

Private Sub btn_DrawText_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_DrawText.Click

    'Initialize Window2
    InitWindow2()

    DrawText()

End Sub

Private Sub btn_ClearWindow2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btn_ClearWindow2.Click

```

```

        'Initialize Window2
        InitWindow2()

        ClearWindow2()

    End Sub

    Private Sub btn_DrawFrame_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_DrawFrame.Click
        'Initialize Window1
        InitWindow1()

        DrawFrame()
    End Sub

    Private Sub btn_ClearWindow1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btn_ClearWindow1.Click
        'Initialize Window2
        InitWindow1()

        ClearWindow1()
    End Sub
End Class

```

APPENDIX C: WIND LOADS USED FOR ANALYSIS

This section outlines the calculation of the wind loads used in Section 5.3, which compared the overall behavior of the metal building frame based on different modeling approaches of the base connection. The wind loads were calculated based on the wind provisions found in Chapter 6 of ASCE 7-05 (ASCE 2005). There are three different methods allowed by ASCE 7-05 to determine the wind loading on a structure. The first method is known as the simplified procedure where a structure must meet a set of criteria in order to be used. The second method is an analytical procedure where the designer determines windward, leeward, external and internal pressures on the building. The third method allows for the use of wind tunnel testing to determine the design loads, where the testing method has to meet a certain set of criteria as outlined in Section 6.6 of ASCE 7-05. This method may be used in lieu of methods 1 and 2 or where required by section 6.5.2. This method would rarely, if ever, be used for a metal building.

Method for loading: If the Main Wind Force Resisting System (MWFRS) meets the following criteria (from ASCE 7-05 Section 6.4.1.1) then the simplified procedure may be used.

- The building is a simple diaphragm building. As per ASCE Section 6.2 where a simple diaphragm building is: “A building in which both windward and leeward wind loads are transmitted through floor and roof diaphragms to the same vertical MWFRS” (ASCE 2005)
- The building is a low rise building
- The building is enclosed and conforms to the wind borne debris provisions of Section 6.5.9.3
- The building is a regular-shaped building or structure
- The building is not classified as flexible
- The building does not have response characteristics making it subject to across

wind loading, vortex shedding, instability due to galloping or flutter; and does not have a site location for which channeling or buffeting in the wake of upwind obstructions warrant special consideration

- The building has an approximately symmetrical cross section in each direction with either a flat roof or a gable or hip roof with $\theta \leq 45^\circ$
- The building is exempted from torsional load cases

The metal building frame described in Section 5.1 meets all of the above criteria therefore use Method 1 to determine the wind loading.

Wind Parameters:

Basic wind speed, V =	90 mph	from Fig. 6-1 ASCE 7-05
Importance Factor =	1.0	from Table 6-1 ACSE 7-05 for building category II
Exposure Category =	B	
Exposure Adjustment Factor, λ =	1.0	

The diagram for design wind loads for Method 1 may be found in Fig. C.1. Only the wind blowing in the direction perpendicular to the frame was considered. The table including the appropriate loads for the above wind parameters may be found in Table C.1.

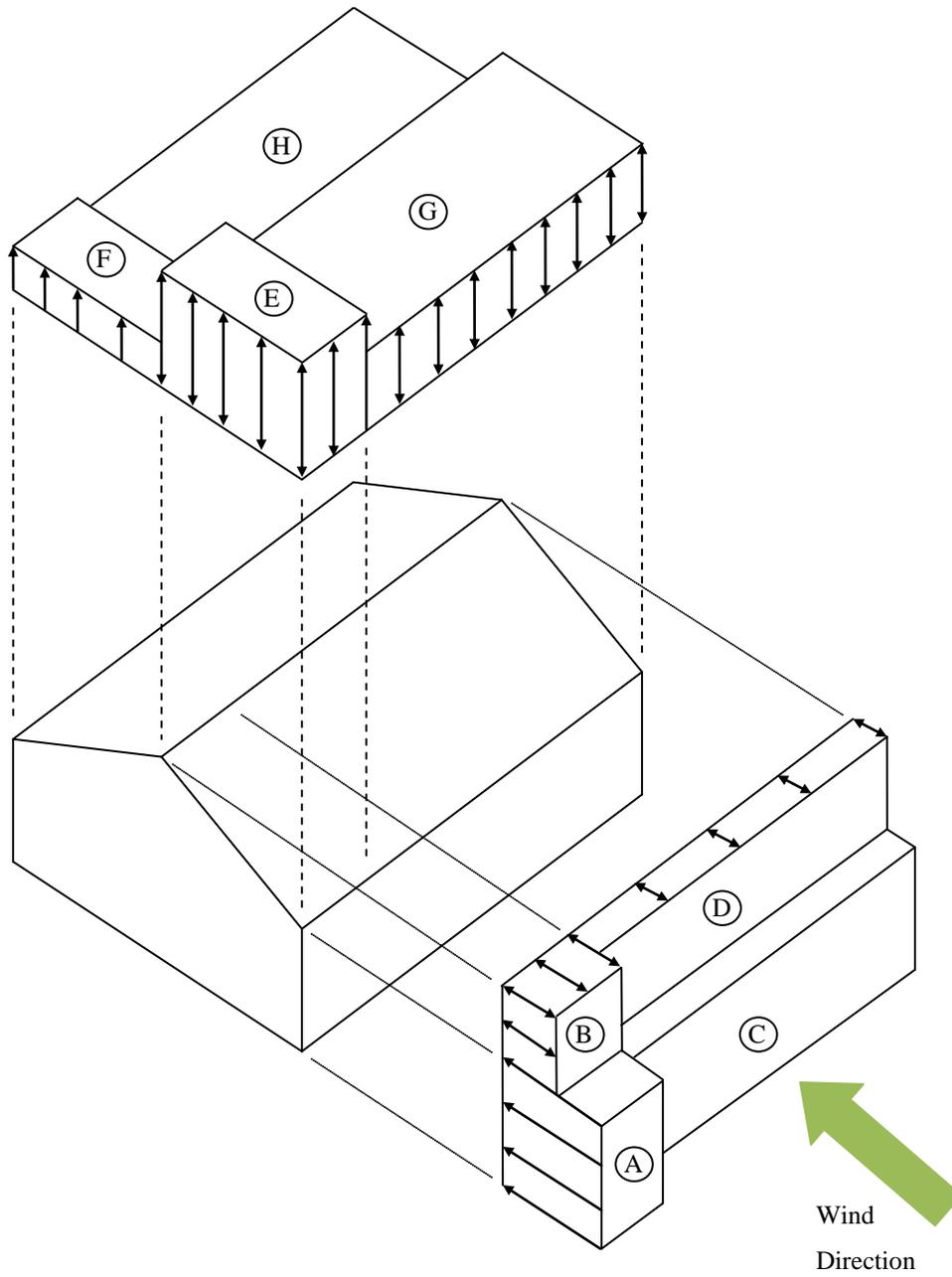


Figure C. 1: Design Wind Loads for Enclosed Buildings – Method 1 (ASCE 7-05 Fig. 6-2)

Table C. 1: Design Wind Loads for Enclosed Buildings – Method 1 (ASCE 7-05 Fig. 6-2)

Basic Wind Speed (mph)	Roof Angle	Horizontal Pressures (psf)				Vertical Pressures (psf)			
		A	B	C	D	E	F	G	H
90	0 - 5°	12.8	-6.7	8.5	-4	-15.4	-8.8	-10.7	-6.8

Determining the Wind Loads on the Frame:

Since the building meets all of the criteria laid out in ASCE 7-06 Section 6.4.1.1, the simplified method for calculating wind loads may be used. This method allows for the use of a table (Fig. 6-2 in ASCE 7-05) which includes the sum of the internal and external pressures on the building. This table represents the net pressures, both internal and external, to be applied to a building based on Method 1 for determining the design wind loads. An excerpt from this may be found in Table C.1. The portion of the table included in Table C.1 is the part which applies to the metal building given the wind parameters above and is based on Exposure B, a mean height equal or less than 30 feet ($\lambda = 1.0$), $K_{zt} = 1.0$ and $I = 1.0$. The pressures found in the table are divided up into different regions, five regions for the horizontal pressures and five regions for the vertical pressures affecting the building. These regions represent the pressure gradient which that particular area of the building experiences. Positive pressures denote a net positive pressure in that region and negative pressures denote a net negative pressure in that region. For example, the exterior side of a building transverse to the wind direction experiences a higher pressure than the interior. This higher pressure is determined based on building dimensions and may be calculated using the notes in Fig. 6-2 in ASCE 7-05. The different pressure regions may be found in Fig. C.1. For the loading used in Chapter 5, only the interior frame is considered for analysis, therefore only the interior pressure regions (C, D, G, and H) are considered. Fig. C.2 shows the design wind pressures on the frame from the example in chapter 5 using ASCE 7-05 Method 1. These pressures were taken from Fig. 6-2 in ASCE 7-05, see Table C.1 for the values that are applicable to the frame.

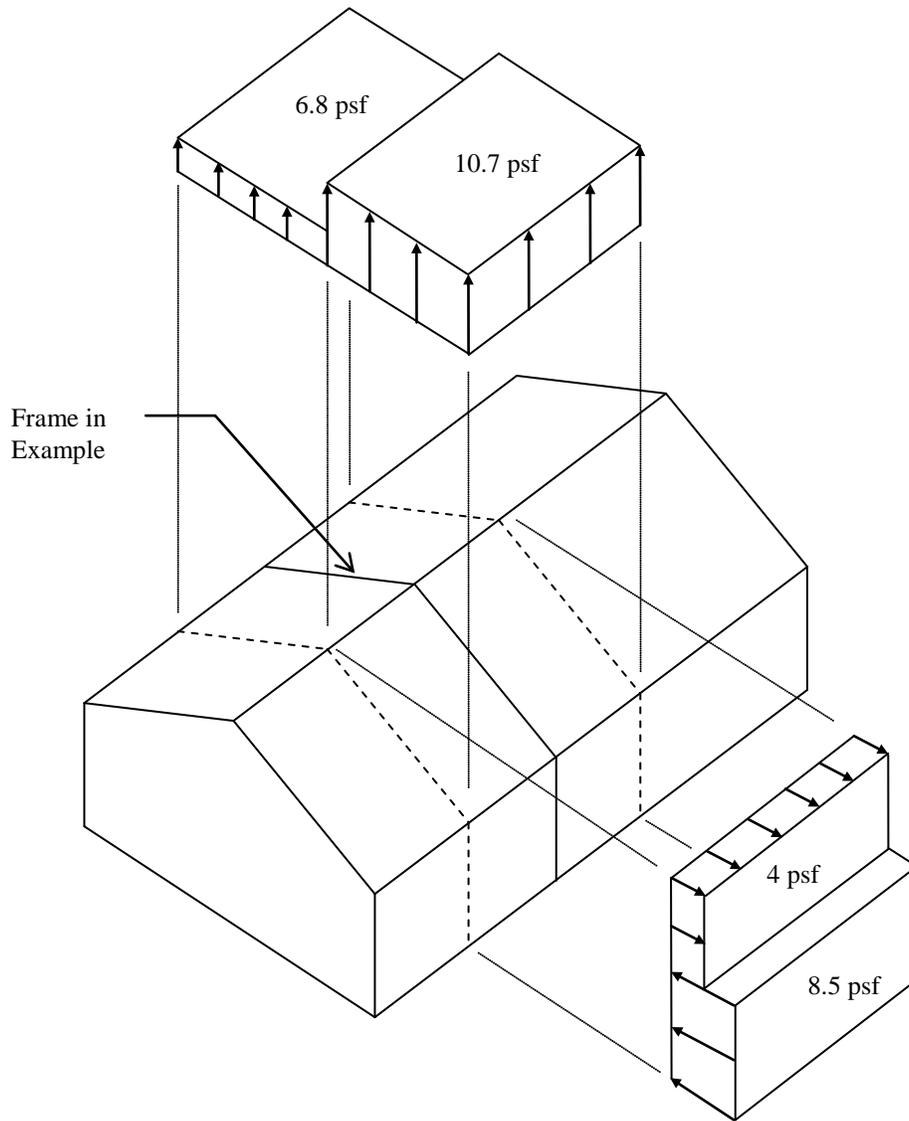


Figure C. 2: Wind Pressures on Frame