

Heuristic, Qualitative, and Quantitative Reasoning About Steel Bridge Fatigue and Fracture

by

W. M. Kim Roddis

B.S. in Civil Engineering, M.I.T. 1977

M.S. in Civil Engineering, M.I.T. 1987

Submitted to the Department of Civil Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy


Massachusetts Institute of Technology

September 1988

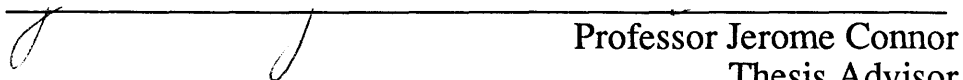
Copyright © W. M. Kim Roddis, 1988

The author grants M.I.T. permission to reproduce and distribute
this thesis in whole or part.

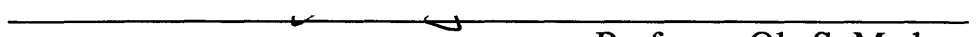
Signature of Author

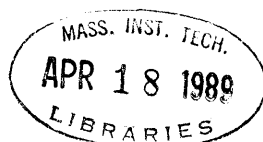

Department of Civil Engineering
September 22, 1988

Certified by


Professor Jerome Connor
Thesis Advisor

Accepted by


Professor Ole S. Madsen
Chairman, Departmental Graduate Committee



Archives

HEURISTIC, QUALITATIVE, AND QUANTITATIVE REASONING
ABOUT STEEL BRIDGE FATIGUE AND FRACTURE

by

W. M. KIM RODDIS

Submitted to the Department of Civil Engineering
on September 22, 1988 in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy in Civil Engineering

ABSTRACT

The growing complexity, sophistication, and bulk of engineering knowledge has lead to situations where incorrect or sub-optimal decisions are made because the appropriate knowledge cannot be accessed and brought to bear in a timely fashion. Just as the number-crunching power of computers several decades ago made possible the routine application of analysis methods too time consuming to use by hand, so Artificial Intelligence (AI) computer techniques promise the power to access and apply specialized knowledge.

Solving an engineering problem requires use of knowledge about: how to gather data to define the problem, how to structure the data into an engineering model, and how to analyze the model to get numeric data. Searching for a solution by reasoning with a simplified model and then verifying, revising, and refining the rough model lies at the heart of engineering problem solving. To effectively capture this behavior requires more than compiled empirical knowledge in the form of antecedent-consequent rules. Analysis and simulation include large chunks of procedural knowledge, not easily accommodated by a rule-based paradigm. The definition, modeling, and interpretation steps are largely symbolic while the analysis step is largely numeric. Qualitative reasoning is one mechanism that can be used to make the transformation from heuristic knowledge to an engineering model suitable for mathematical manipulation.

The strategy of using an intermediate qualitative simulation layer manipulating first order engineering models to connect a predominantly heuristic and symbolic rule-based top layer with a largely procedural and numeric quantitative root layer is applicable to a wide variety of engineering problems. Rules are good for generating hypotheses to define the initial areas to search. Qualitative methods can be used to enumerate possible behaviors and focus on promising models. Quantitative analysis resolves ambiguous behavior and provides quantified answers. Each method is a better tool for a different problem solving phase. This thesis presents such a unified tool implemented with a three reasoning level plus one shared communication level architecture.

To demonstrate the validity of the approach, the specific domain of fatigue and fracture in steel bridges is addressed in CRACK (Consultant Reasoning About Cracking Knowledge). CRACK performance was verified for solving failure analysis, predictive modeling, and design critique tasks for welded plate girder and rolled beam bridges. The results of the three plus one system architecture was satisfactory but this experiment was not an unalloyed success. In particular, the qualitative level was found to be over-engineered and under-utilized. Limitations of the established AI techniques for qualitative reasoning must be overcome for these methods to be useful in solving realistic engineering problems. Recent work in the field of common sense reasoning has promise for overcoming some of these technologic short-comings.

Acknowledgments

The M.I.T. community is a rich educational environment. There are many people who have contributed directly and indirectly to this work. Some offered major amounts of help over extended periods of time. Others offered a few pointed words of advice at a critical point. Still others provided intellectual camaraderie, the most stimulating aspect of graduate studies. I would like to thank:

Lindsey Spratt, my husband and best friend, for being a live-in computer consultant, for being unflaggingly supportive of my studies, and for simply being himself.

Jerry Connor, my thesis supervisor, who pushed at the right time for closure instead of expansion, showing me how to turn a fertile project with many loose ends into a coherent thesis.

Ramesh Patil, Regis Pelloux, Duvvuru Sriram, and Victor Li, the other members of my thesis committee, for constructive criticism and cogent comments on drafts of this document.

Mounir Berrah, Laura Demsetz, Hans Hens, Annette Huber, Shunichi Igarashi, John Ludwigsen, Hisashi Maesaka, Lisa Mulvey, Tommaso Pagnoni, Chee-Kiong Soh, Hayat Tazir, and Thanasis Triantafillou, my friends and colleagues in Structural Engineering.

Stu Chen, a graduate student in Civil Engineering at Lehigh University, who helped me get access to the Fritz Engineering Laboratory Library and who, with his wife Pam (and their Doberman Cleo) provided gracious hospitality for my trip.

Dan Dvorak, a graduate student in Computer Science at the University of Texas in Austin, who provided the draft implementation of QSIM in Prolog.

Lorna Gibson, who was on my general examination committee, for her encouragement at each stage of my graduate work.

Jeanne Richard, Administrative Associate Dean of the Graduate School, for good advice and lifts when Boston winter's and traffic made bicycling an unattractive mode of commuting.

Paul Blair, of A.G.Lichtenstein & Associates, for the opportunity to climb around on bridges for a summer to appreciate the difficulties of bridge assessment under field conditions.

This work was supported by the Fannie and John Hertz Fellowship Foundation.

Contents

Title Page	1
Abstract	2
Acknowledgements	3
Contents	4
List of Figures	5
1. Introduction	10
1.1 Artificial intelligence and engineering	10
1.2 CRACK (Consultant Reasoning About Cracking Knowledge)	12
1.3 Guide to the thesis	13
2. Engineering problem solving	14
2.1 How people solve engineering problems	14
2.1.1 Engineering judgment levels	14
2.1.2 Human problem solving	15
2.1.3 Framework for engineering models	16
2.2 Heuristic knowledge	18
2.3 Quantitative analysis	19
2.4 Interaction between heuristic and quantitative reasoning	19
2.5 Qualitative reasoning: causal knowledge	21
2.6 Architecture for linking rules/qualitative/quantitative	24
3. Problem domain: cracking in highway bridges	32
3.1 Choice of problem domain	32
3.1.1 Need for better solutions in engineering practice	32
3.1.2 Required knowledge is diverse and circumscribed	34
3.1.3 Multiple uses of same knowledge	35
3.2 Envisioned use	38
3.3 Related work	39
3.4 Application of three-level framework to the domain knowledge	40
4. Domain knowledge: fatigue and fracture of welded steel plate girders	43
4.1 Knowledge sources and acquisition	43
4.2 Requirements for structural integrity	43
4.3 Fatigue design philosophies	45
4.4 Parameters governing crack behavior	46

4.5 Elements of linear elastic fracture mechanics	47
4.5.1 Definition of LEFM	47
4.5.2 Stress intensity factors	48
4.5.3 Applicability of LEFM for bridges	49
4.5.4 Determining stress intensity factors	50
4.5.4.1 F_e : elliptical crack shape correction factor	51
4.5.4.2 F_s : free surface correction factor	52
4.5.4.3 F_w : finite width correction factor	53
4.5.4.4 F_g : local stress gradient correction factor	53
4.5.5 Paris power law describing crack growth	55
4.6 Fracture control plan for steel highway bridges	57
4.7 Taxonomy of cracking causes for steel highway bridges	62
4.8 Cracking in welded steel plate girder highway bridges	63
4.8.1 Initial crack size, shape, and orientation	64
4.8.2 Stress history	65
4.8.3 Material properties	68
4.8.4 Structural properties	70
5. Architecture of CRACK	93
5.1 Project history	93
5.2 Implementation choices	93
5.2.1 System requirements	93
5.2.2 Software: Prolog	94
5.2.3 Hardware: Mac II	97
5.3 System modules	97
5.3.1 Rule-based level	97
5.3.1.1 Shell	98
5.3.1.1.1 Role of the shell	98
5.3.1.1.2 Implementation of the shell	99
5.3.1.2 Rule base	102
5.3.2 Qualitative simulation	103
5.3.3 Quantitative analysis	106
5.3.4 Interlevel communication	107
5.4 Flow of the program	107
5.4.1 Task type	108

5.4.2 Problem description	108
5.4.3 Forming hypotheses	109
5.4.4 Qualitative simulation	109
5.4.5 Fracture mechanics calculations	110
5.4.6 Evaluation of hypothesis	110
5.4.7 Statement of conclusions	111
6. Detailed example: failure analysis of actual bridge	119
6.1 Presentation of case study	119
6.1.1 Introduction to case study	119
6.1.2 Description and history of the bridge	119
6.1.3 In-service conditions: loads and temperatures	119
6.1.4 Material testing	120
6.1.5 Fractography	123
6.1.6 Analysis of evidence	124
6.1.7 Repair versus replacement	125
6.2 Application of CRACK to case study	126
6.2.1 Suitability of case study for CRACK	126
6.2.2 CRACK problem solving session	127
6.2.3 Failure analysis mode	127
6.2.4 User supplied information	127
6.2.5 Forming hypotheses	128
6.2.6 Construction of parametric study	129
6.2.7 Fracture mechanics calculations	130
6.2.8 Evaluation of hypothesis	131
6.2.9 Statement of conclusions	131
6.3 Critique of CRACK performance on case study	133
7. System capabilities and limitations	149
7.1 Performance within domain	149
7.1.1 Design critique	150
7.1.2 Predictive modeling	151
7.1.3 Failure analysis	152
7.2 Limitations of domain knowledge	152
7.3 Critique of three level architecture	157
8. Conclusion	165
8.1 Contributions	165

8.2 Bridge engineering practices	166
8.3 Summary	167
References	170
Appendix A: CRACK Test Cases	183
Appendix B: Qualitative Simulation Examples	215

List of Figures

Figure 2-1:	Abstract models	27
Figure 2-2:	Framework for constructing and utilizing engineering models	28
Figure 2-3:	Forward and backward chaining inference procedures	29
Figure 2-4:	Qualitative reasoning	30
Figure 2-5:	Layered system architecture for CRACK	31
Figure 4-1:	Fracture analysis sequence: analogy to yielding	72
Figure 4-2:	Fatigue crack growth mechanism	73
Figure 4-3:	Parameters governing cracking behavior	74
Figure 4-4:	Griffith crack configuration	75
Figure 4-5:	Stress intensity factor K computation	76
Figure 4-6:	F_e : elliptical crack shape correction factor	77
Figure 4-7:	Range of F_w : finite width correction factor	78
Figure 4-8:	Procedure for determining F_g	79
Figure 4-9:	Crack growth rate versus stress intensity range in steel	80
Figure 4-10:	Fatigue design category descriptions	81
Figure 4-11:	Fatigue design category illustrative examples	82
Figure 4-12:	Fatigue design curves	83
Figure 4-13:	Hierarchy of primary causes for cracking in steel bridges	84
Figure 4-14:	Secondary stresses and distortion-induced stress	85
Figure 4-15:	Initial cracks and low fatigue resistant details	86
Figure 4-16:	Pruned hierarchy for cracking in plate girder bridges	87
Figure 4-17:	Initial flaws in welded joints	88
Figure 4-18:	Temperature and load rate effects on toughness	89
Figure 4-19:	Effect of specimen thickness on fracture toughness	90
Figure 4-20:	Some components of a plate girder	91
Figure 4-21:	Example girder cross sections	92
Figure 5-1:	CRACK interface	112
Figure 5-2:	Simple shell structure	113
Figure 5-3:	Rule based shell for CRACK	114
Figure 5-4:	Simple influence network	115
Figure 5-5:	Possible behaviors from QSIM	116
Figure 5-6:	Influence network for crack domain	117
Figure 5-7:	Graphic example: penny shaped crack	118

Figure 6-1:	Elevation: case study bridge	135
Figure 6-2:	Cross section: case study bridge	136
Figure 6-3:	Plate girder failure section: case study bridge	137
Figure 6-4:	Maximum hardness versus carbon content of plain carbon steels	138
Figure 6-5:	Charpy V-Notch impact test results	139
Figure 6-6:	Dynamic mode I fracture toughness K_{I_d}	140
Figure 6-7:	Sketch of fracture surface near origin of crack	141
Figure 6-8:	Detail at intermediate panel points for sway bracing	142
Figure 6-9:	Crack growth simulation: case study: stress range 8.7 ksi	143
Figure 6-10:	Crack growth simulation: case study: stress range 17.4 ksi	144
Figure 6-11:	Qualitative behaviors: case study bridge	145
Photograph 6.1:	Case study bridge: elevation looking north	146
Photograph 6.2:	Case study bridge: typical framing for plate girder spans	146
Photograph 6.3:	Crack in north girder, located at third point of central span	147
Photograph 6.4:	Crack in north girder, severing bottom flange and full web	147
Photograph 6.5:	Interior view of crack at top flange	148
Photograph 6.6:	Interior view of crack at bottom flange	148
Figure 7-1:	Design critique test case results	162
Figure 7-2:	Qualitative behaviors: Appendix B example 2	163
Figure 7-3:	Qualitative behaviors: Appendix B example 3	164

Chapter 1

Introduction

1.1 Artificial intelligence and engineering

The practicing engineer has a voracious need for information and the expertise to apply it. This need has grown due to the increasing sophistication, complication and size of extant engineering knowledge. Although the knowledge exists, there are problems in finding the best source at the right time. In the face of these place and time obstacles, some AI computer techniques in the form of knowledge-based systems can be used to provide better access to the specialized knowledge in a timely manner.

The Civil Engineering Department at MIT has become increasingly involved in the application of AI techniques to engineering problem solving, especially in the area of knowledge-based systems. Ongoing research projects fall into the assorted categories of: communication and coordination in engineering design, design innovation [Mulvey 87], geographic information systems, construction planning [Cherneff 88], applied natural language processing [Chen 88], construction robotics [Slocum 87], infrastructure maintenance [Farach 86], control, simulation, design and analysis advisors [Soh 86], and intelligent computer-aided instruction [Noack 88]. A knowledge-based approach has been used to address diverse problem areas. These domains have included generative tasks, such as design of steel members [Roddis 86] and type studies for bridge preliminary design [Pagnoni 85], as well as diagnostic and prescriptive tasks, such as condition assessment of concrete bridge decks [Seymour 86]. The primary appeal of a knowledge-based approach is the explicit statement of the knowledge used to solve a problem. Explicit knowledge representation: enables automatic generation of explanations and justifications, facilitates incremental development and enhances maintainability by making program intent readily apparent and extendible, and allows multiple uses of the same knowledge.

Engineering encompasses disparate kinds of knowledge (numeric, symbolic, heuristic, algorithmic) which must be brought to bear to solve a problem. Although in theory a knowledge-based approach allows incorporation of heterogeneous knowledge, off the shelf existing computational tools for engineering problem solving employ either numeric or symbolic techniques and thus do not satisfactorily address both. Numeric analytic tools have reached a high level of maturity, but require engineering specialists to hand-craft the analytic models and interpret the results, sometimes spending more time on dealing with the tool than with understanding the problem. First generation knowledge-

based tools generally provide any numeric computations by hooks to the underlying computer language, so that all the advantages of traceability and explainability of the knowledge-based paradigm is lost for the numeric portion of the program. An additional barrier to the dissemination of knowledge-based systems in civil engineering practice has been the perception that special purpose, costly hardware and software is required to solve most kinds of practical engineering problems. Deliverability to practicing engineers demands flexible microcomputer hardware and cost effective software. The recent expanded capabilities of the higher end personal computers make it technically feasible to solve significant engineering problems using a knowledge-based approach with relatively inexpensive hardware and software tools.

1.2 CRACK (Consultant Reasoning About Cracking Knowledge)

The need of the engineering profession for better knowledge dissemination and the promising initial benefits from utilizing a knowledge-based approach provided the motivation for this thesis. The broad objective was development of a computational tool that linked the symbolic and numeric aspects of engineering problem solving within a unified framework, implemented on a hardware platform suitable for wide distribution to practicing engineers. To develop a computational tool with broad applicability to practical engineering problem solving the cognitive tasks involved must be understood and an architecture specified to support execution of those tasks. To test the suitability of the architecture, a specific problem domain, representative of the broad class of engineering problems must be addressed. Implementation of the architectural design and application to the testbed domain is the vehicle for proof of the concept.

CRACK (Consultant Reasoning About Cracking Knowledge) is a computational tool for engineering problem solving in the domain of bridge fracture using multiple levels of knowledge. The system's architecture is capable of supporting engineering reasoning about both the overall problem domain at a relatively high level of abstraction and the narrow sub-domains at a more refined granularity. The approach links heuristic and quantitative simulation levels into an integrated framework by using a middle layer of qualitative abstractions of the underlying quantitative causal models. The ability to direct search by reasoning with a simplified qualitative model and then verifying, revising, and refining the qualitative model by recourse to quantitative analysis lies at the heart of engineering problem solving. Application of the general methodology to the example domain of highway bridge cracking proves the validity of the concept, and illuminates its strengths and shortcomings.

The contributions of this research are as follows:

- 1) A conceptual framework for the construction and utilization of engineering models.
- 2) Improved understanding of the state of the art of fracture mechanics as applied to in-service steel bridges by the explicit compilation of domain heuristics.
- 3) A computational architecture useful for problem solving in a variety of engineering contexts.

The framework for abstract modeling used to solve engineering problems delineated the tasks that the computational model was required to support. In addition to defining system requirements, this conceptual framework was also useful for determining how to structure explanations in a way that would parallel human problem solving behavior, and thus be more readily understandable to the engineer user.

Formalization of domain knowledge about cracking in bridges is valuable in and of itself as is readily apparent when one considers the size of problem. There are about 600,000 highway bridges in the United States alone, many of which are steel. Approximately 40% of these bridges are currently rated as substandard. The risks of fatigue and fracture failures are increasing as the existing bridge population ages, and as older bridges exceed their design life. Any method that adds dissemination of rational control of fatigue and fracture has the potential to be extremely useful since the annual cost to the US economy has been estimated to be 100 billion dollars, half of which could be saved if existing knowledge was optimally applied. This illustrates the enormous potential of constructing tools which can enhance the engineer's access to expertise.

CRACK is a research prototype system representing a first cut on the problem domain. The linkage of heuristic, qualitative, and quantitative reasoning advance this tool beyond the level of first-generation expert system shells. It has been designed to test out the proposed multilevel system architecture for a single sample engineering domain in a research environment.

1.3 Guide to the thesis

This thesis is organized as follows. Chapter 2 investigates the role of analytic models for engineering problem solving. A conceptual framework for the construction and utilization of these models is presented. An architecture to support this conceptual

framework is formulated. Chapter 3 justifies the choice of the sample problem domain of fatigue and fracture in highway bridges and shows how it fits within the proposed framework. Chapter 4 is a presentation of the requisite domain knowledge. Chapter 5 addresses the computer implementation. The system requirements are specified, software and hardware choices are presented, program modules are enumerated, and the program flow is discussed. Chapter 6 is a detailed presentation of an actual bridge cracking failure and the performance of the prototype system on this test case. The capabilities and limitations of CRACK are discussed in Chapter 7. The thesis concludes with Chapter 8 summarizing key findings and pointing toward future extensions. Credible performance in a research setting for a number of test cases gleaned from the literature and example qualitative simulation results are also included in the appendices.

Chapter 2

Engineering problem solving

This chapter examines the engineering problem solving process and looks at the way in which analytic models are utilized within that process. The progression from novice to expert engineer is surveyed. Inspecting the way people solve engineering problems shows the central role of abstract engineering models. The construction and utilization of these models is fit into a conceptual framework. The different types of reasoning (heuristic, qualitative, and quantitative) that are used to construct and manipulate these models are examined. An architecture of three reasoning levels, sharing a common representation of the engineering model is specified.

2.1 How people solve engineering problems

When constructing a computational model to perform cognitive tasks, it is valuable to examine the human cognitive process. When attempting to construct a system that will demonstrate intelligent behavior, people are the only point of comparison that is available to examine how a demonstrably intelligent system works. Another reason for using the human process for guidance is that a system that models human behavior to a certain extent operates in a manner that is more readily comprehensible and will allow a more natural construction of explanations that are satisfying to people. In addition, the user interface is more natural if there are parallels to the approach an engineer would take.

2.1.1 Engineering judgment levels

As engineers gain experience in their discipline, they develop expertise in solving engineering problems. The stages that an engineer passes through developing from a novice to a competent professional may be described with four judgment levels [NCEE 84]. Each of the levels encompasses, to some extent, the cognitive behavior and skills required in the preceding level or levels. These levels may be summarized, from the simple to the more complex, as follows:

Level I: Knowledge/Comprehension: The most novice level of engineering problem solving requires the recall or recognition of specific terminology and facts, knowledge of ways and means of manipulating these specifics for basic and routine tasks, and knowledge of the principles and theories associated with the task. The required basic understanding would be

reflected by the ability to translate, interpret, or extrapolate.

Level II: Application: Early engineering experience is gained through straightforward application of knowledge to specific situations. This level includes the use of abstraction, rules, principles, ideas, and methods in specific and concrete situations

Level III: Analysis/Synthesis: More challenging problems require breaking a task down into components and an understanding of the sequence of the resultant parts and the relationship among them. This level includes assembly of parts to make a whole, forming a new pattern or structure.

Level IV: Evaluation: Professional competence includes the ability to assess the extent to which materials or methods satisfy criteria. This level requires judgment about the value of ideas, solutions, materials and methods for a given purpose.

2.1.2 Human problem solving

Insight into how people solve problems is gained by looking beyond the solution of individual problems to the motives and procedures of the solution. In trying to find the solution, a person may repeatedly change perspectives, focusing on various aspects of the problem. Problem solving breaks down into the four phases of: 1) understanding, 2) planning, 3) carrying out, and 4) looking back [Polya 73]. A problem that is perfectly stated has all necessary and no superfluous data, just sufficient and no contradictory or redundant conditions. Practical problems are usually far from being perfectly stated. In practical problems, some data and conditions *must* be neglected since the data are inexhaustible. Ideally, all data and conditions which significantly affect the solution are included and all others are neglected, but this of course requires prior knowledge of what is relevant. The practical engineering problem leads to a mathematical problem to be solved approximately. In order to solve a problem, a certain amount of previously acquired knowledge is needed. The modern engineer has a highly specialized body of knowledge available including a scientific theory of the strength of materials, personal experience, and the mass of engineering experience stored in special technical literature. Such precise, quantitative, scientific concepts form an essential part of the intellectual equipment of an engineer. Yet an engineer also uses much knowledge which has not yet reached a precise, scientific level, but is rather of an empirical character [Polya 73].

Problem solving uses the skills of discrimination, analysis, and synthesis. Discrimination is the ability to see different parts in observation. Analytic ability is the skill of breaking things into parts. Synthesis is the creative ability of putting parts together into new wholes.

The problems presented by the world are generally ill-structured, lacking definition in some respect. They become well-structured problems in the process of being prepared for solution. Much of the effort of solving a problem is directed at structuring it. When tackling complex problems, analytic ability can be applied to great effect if the problem can be decomposed, partitioning it into sub-problems. Decomposition is effective on most types of engineering problems. For example, an entire design problem begins to acquire structure by being decomposed into various subordinate problems of component design. “The problem is well structured in the small, but ill structured in the large”[Simon 73].

2.1.3 Framework for engineering models

Solving a practical engineering problem leads to a mathematical problem to be solved approximately [Polya 73]. This transformation is the process of engineering modeling. Modeling is a way to handle complexity and solve complex real problems by creating much simpler abstract models. Figure 2-1 [Kuipers&Patil 87] illustrates how modeling simplifies and idealizes the real system to gain tractability. A model is a *description* of a system intended to predict the consequences of actions on the *actual* system. After a model is built to describe the physical system, simulations can be performed by driving the model with suitable inputs and observing the corresponding outputs. If the model was valid within the range of interest, the simulation results generated using the model correctly describe the real world behavior. A useful abstraction must suppress much (or most) of the detail from the actual physical environment. Indeed, a common error in modeling is an excessive amount of detail [Bratley et al 87].

Solving an engineering problem requires use of knowledge about how to gather data to define the problem, how to structure the data into an engineering model, and how to analyze the model to get numeric data. The solution passes through the various stages of problem definition, data gathering and hypothesis generation, construction of analytic models; numeric analysis; processing analytic results; verifying and refining models; and stating conclusions. The information flow is not simply one-way, one-pass. The results of the analytic step are evaluated to find what additional data is required and whether the model needs refinement. In general, multiple models with several refinement versions are postulated, and multiple iterations of the define/model/analyze/evaluate cycle are executed.

This simplified description of engineering modeling is shown in Figure 2-2. The white bordered stages of numeric analysis and evaluation of analytic results involve numeric computation, a discipline with relatively mature computer techniques. The black bordered stages of definition/hypothesis and conclusions are well suited to a rule-based paradigm. Rules work well for generating hypotheses based on the problem definition and expressing conclusions based on abbreviated inference traces at bottom. The gray bordered stages of construction of analytic models and model validation/refinement provide the crucial link connecting heuristic to analytic reasoning.

Many engineering problems have a large diagnostic component. Diagnosis may be viewed as hypothesis directed inquiry . In the cycle depicted in Figure 2-2, after the problem is defined, a hypothesis is formed. It is the testing of this hypothesis which requires the construction of the engineering model. Although the language used to label the stages in Figure 2-2 is slanted toward engineering tasks that are widely recognized as being diagnostic, such as failure analysis, the same general cycle is also descriptive of an important part of design tasks. Engineering design establishes a description of an artifact that will possess specified desired properties (functionality, constructible, quality, cost, operability, maintainability). It fits the definition of a problem that is ill-structured in the large and well-structured in the small after decomposition [Simon 73]. Creation of an engineered artifact frequently proceeds by proposing designs heuristically and verifying designs analytically [Chan 86]. The better the justification in terms of some fundamental causal mechanisms for why a heuristic works, the surer validity of the design proposed by the heuristic. Better heuristics don't eliminate need for testing and verification in design, since correctness remains to be proven. The part of engineering design that involves a critique of a hypothesized artifact to examine whether it satisfies the specification fits within the framework shown in Figure 2-2.

One way of classifying problem solving is to locate the task on a spectrum bounded on one end by derivation and on the other end by formation [Maher 87]. Derivation problem solving, such as diagnosis and classification, start from a known state and infer a path to a known solution. Formation problem solving, such as design and planning, form a solution from stored eligible solution components. Engineering problems fall all along the derivation/formation spectrum. The conceptual framework in Figure 2-2 addresses the part of solving a problem that involves the construction and utilization of an engineering model, regardless of where the entire problem may be placed on the deriva-

tion/formation spectrum.

2.2 Heuristic knowledge

The purpose of heuristic reasoning is to aid discovery by providing direction in the search for a solution.

“Heuristic rules are rules that when used by human beings are said to be based on experience or judgment. Such rules frequently lead to plausible solutions to problems or increase the efficiency of a problem-solving procedure. Whereas algorithms guarantee a correct solution (if there is one) in a finite time, heuristics only increase the likelihood of finding a plausible solution”[Simon&Newell 58].

An explication of heuristic problem solving which captures much of the flavor of engineering modeling makes the distinction between deduction and abduction [Polya 73]. Comparison of demonstrative and heuristic syllogisms shows why heuristic reasoning is inherently provisional. The application of modus tollens in the demonstrative syllogism leads to a logically rigorous conclusion of the falsity of A. The plausible reasoning based on the heuristic syllogism does not have the certainty of a strict demonstration.

DEMONSTRATIVE REASONING

HEURISTIC (PLAUSIBLE) REASONING

Demonstrative syllogism:

IF A THEN B

B FALSE

_____ (therefore)

A FALSE

Heuristic syllogism:

IF A THEN B

B TRUE

_____ (therefore)

A MORE CREDIBLE

Incorporating this kind of reasoning into a computer program was one thread in the seminal work [Buchanan&Shortliffe 84] leading to the development of knowledge-based systems. Knowledge-based systems take a variety of approaches including rule-based, logic-based, frame-based, and task-specific problem solving architectures [Chandrasekaran&Bylander 88]. This thesis focuses on a rule-based expression of heuristic knowledge. (Two of the most frequently cited approaches, rules and frames, have considerable similarity in their expressive power, indicative of the ability to represent knowledge cast in one formalism into the other[Buchanan&Shortliffe 84].) A central premise of knowledge-based systems is the separation of the domain knowledge from the inference mechanism which manipulates that knowledge to solve a problem.

Two formalized inference procedures for reasoning in rule-based systems are forward and backward chaining (see Figure 2-3). Bottom-up, data-driven, and antecedent-driven are terms equivalent to forward chaining. Top-down, goal-driven, and hypothesis-driven are terms equivalent to backward chaining [Maher 87].

2.3 Quantitative analysis

The traditional approach to an engineering analysis of any device's behavior is to formulate a mathematical model that is used to calculate output parameters based on given input parameters. Knowledge of how to simulate the response of an engineered artifact such as a structural detail to driving parameters such as load requires the representation of large chunks of procedural knowledge, normally encoded in a numeric language such as Fortran or C. Input and output parameters are normally not invertible, so if a specific output set is known, a matching input set cannot be directly found.

Numeric analytic tools have reached a high level of maturity, but require engineering specialists to hand craft the analytic models and interpret the results. Correct program use requires large amounts of non-algorithmic knowledge. This requisite knowledge is either represented implicitly in the source code, or not represented in the program at all, except possibly as passive comments. Algorithmic computer applications for engineering cannot explain their reasoning to the user. As programs become bigger and more complex, the user spends an increasing proportion of time and effort on understanding how the *program* works instead of developing and understanding the *problem*.

2.4 Interaction between heuristic and quantitative reasoning

A fundamental characteristic of the the engineering modeling process depicted in Figure 2-2 is the coupling of symbolic and numeric computations. There is a central interaction between the largely symbolic definition/model-construction/evaluation stages and the numeric simulation/analysis stages. Both symbolic and numeric capabilities are needed to solve the problem.

An obvious method of addressing this need for coupling between symbolic and numeric computing is a hybrid system using a rule-based symbolic portion and a traditional numeric portion. Exploration of the use of a rule-based system as an intelligent front end to an existing analysis packages was begun early in the development of expert systems [Bennet et al. 78], and the approach of directly linking rule-based and numeric

computation has continued [Kitzmilller&Kowalik 87]. Most extant coupled systems tend to treat mathematical modeling components as a black box, using production rules to fill in the numeric parameters required by the model. The expert systems serve as a “big switch” for selecting models or algorithms from a pre-enumerated list. Such use of production rules to fill in the parameters of model templates lacks flexibility in structuring the models [Wellman 86]. In addition to not being able to handle problems requiring a model that requires a non-predefined primitive element, such systems frequently cannot flexibly construct suitable models by aggregation of partial-solution components.

One advantage consistently offered for using a knowledge-based approach is the ability of the system to in some limited sense *understand* the basis for taking an action and to provide an *explanation* when queried by the user [Hayes-Roth et al 83]. Early knowledge-based work [Buchanan&Shortliffe 84] found that physicians demanded explanations as part of making the system acceptable. Users require a window into the knowledge base and line of reasoning to be persuaded that the way conclusions are reached is sound. To use a consultant program, a physician must be persuaded that the recommendations are appropriate. Engineers, like physicians, have a moral and legal responsibility for the consequences of their actions. Thus one of the reasons knowledge-based systems have an enormous appeal to the engineering profession is this explanation capability. This feature can be exploited to make use of computational engineering tools more *reliable*, an area of great importance to the practicing engineer. Although hybrid tools that perform numeric computations with hooks to precompiled black boxes in the underlying computer language gain efficiency, they lose all the advantages of traceability and explainability of the knowledge-based paradigm for the numeric portion of the program.

Even when such escapes are not made from the knowledge-based approach, constructing satisfying explanations has been found to be impossible, due to lack of *support knowledge*, the underlying mechanistic or associational links that explain why the action portion of a rule follows logically from the premise. The homogeneity of the knowledge representation scheme may fail to discriminate among change associations, statistical correlations, heuristics based on experience rather than precise statistical studies, causal associations, definitions, knowledge about structure, and taxonomic knowledge. Of particular concern for engineering domains is the lack of a representational distinction between empirical associations and causal relationships. The distinction between rules that arise empirically or causally is made clear by examining the grounds for believing the rule: If A and B then C. For empirical associations, previously when A and B held, C was

also found to be true. This association is only a first step in discovering causality. The association that most construction accidents involve male workers does not indicate that men cause accidents and women don't, but only means there are more male than female workers on construction sites. In contrast, more steel bridges fail in winter than summer because the lower temperature causes a drop in fracture toughness. The difference between association and causality complicates the construction of useful explanations [Charniak&McDermott 85].

Numeric computation should be produced in a way that explicitly represents algorithmic and heuristic knowledge about the programs. This explicit knowledge should be manipulated to make the black box number crunchers transparent to the user. Explicit representation is required for automatic explanation of the system's behavior and justification of conclusions. Even if the algorithmic program was recast in rules (which is not desirable since the representation of large chunks of procedural knowledge is not easily accommodated by a rule-based paradigm), the explanation generated would be in the form of the system's execution of empirical rules. This type of explanation cannot show why a rule works, since the underlying knowledge of causal links and general laws of nature is unavailable [Swartout 83; Swartout&Smoliar 87]. Producing adequate engineering explanations relies on this missing knowledge.

The generation of engineering models is founded on this same underlying domain knowledge of causality and fundamental principles. Knowledge-based work to date has not identified a generally satisfactory strategy for encapsulating this kind of domain knowledge. One approach that has been promising for knowledge transformations between heuristic and mathematical knowledge bases is qualitative reasoning [Cross 83]. It is precisely this transformation which is the key in engineering modeling.

2.5 Qualitative reasoning: causal knowledge

A major research area concerned with automated reasoning about the real (physical) world is qualitative reasoning [Bobrow 85]. Qualitative reasoning can be used to draw inferences about trends in behavior resulting from incomplete, non-numeric knowledge of the parameters which influence that behavior. It can provide an interesting model of the behavior of physical systems [Forbus 81] meeting the dual goals: 1) dealing better with open systems than strictly quantitative reasoning; and 2) being capable of generating explanations that are readily comprehensible to people. People frequently reason about general changes without any specific measure of those changes, only using quantitative rea-

soning as a last resort. For many cases, the simple qualitative information is enough to solve the problem so it is not efficient to perform an exact analytic procedure in a case where a precise answer isn't needed. In addition, exact information may not be available, so some form of qualitative reasoning is essential to get any answer at all. Even if an exact numeric answer is needed, qualitative reasoning is useful to focus attention and figure out what things can be ignored. Qualitative reasoning can be used to direct quantitative reasoning. This approach can also be reversed, quantitative reasoning can be explained qualitatively. An example of a system that uses a qualitative reasoning approach to explain computationally complex, mathematically represented quantitative knowledge is presented in [Cross 83] for the domain of air traffic control.

Qualitative reasoning can be viewed as a level of abstraction from quantitative reasoning as illustrated in Figure 2-4 [Kuipers&Patil 87]. Qualitative models are thus particularly natural in the frequently arising situation where not enough information is available about the physical system to build a quantitative differential equation model without including arbitrary assumptions.

The common goal of qualitative reasoning research is to understand a physical system by deriving a description of the system's behavior from the system's structure [Bobrow 85]. Structure refers to the components of the analysis, the separate behaviors of the components, and the connections among the components. The system's behavior in response to a change is generated by starting at the initial point of disturbance and propagating effects through the connections. The propagation process communicates the effect of the disturbance throughout the system while satisfying the qualitative constraints. These qualitative constraints express dependency relationships among the system's state parameters. The dependency relationships among the system's state parameters. The dependency relationships usually represent causal relations among the parameters. A qualitative notation and calculus are required to computationally support this process. Necessary characteristics for such a calculus within its domain are [deKleer 79] that it is complete, having the capability to simulate every possible behavior, limiting, generating few ambiguities, and articulate, identifying the sources of ambiguities so that other knowledge can be used to deal with them.

Several approaches have been taken to building qualitative models. An approach which focuses on causal analysis, describes the equilibrium state of a system may be described with a set of simultaneous structural equations, each describing a separate mecha-

nism (a law describing a physical process or a local component that can be described as operating according to such laws). Given this set of equations establishing functional relationships among the variables, causal ordering (which variables initiate changes in other variables) may be derived from the dependencies among the equations [Iwasaki&Simon 86]. This causal ordering approach has been extended to cover models consisting of a mixture of dynamic and equilibrium equations [Iwasaki 88]. A reductionist approach is to emphasize the components and connections, defining the behavior of the system by the behaviors of its physical constituents. Each component's behavior is described using confluences (qualitative differential equations) [deKleer 75] which, together with specified topological connections, provide a set of constraints. The overall system behavior is described by the union of the constraints [deKleer&Brown 85]. A process approach also describes individual components but uses influences based on a process vocabulary which specifies how things change within the domain [Forbus 84]. A knowledge engineering approach hand crafts a constraint model of qualitative differential equations for the system constructed from the domain state variables and the constraints among them [Kuipers 85].

Qualitative reasoners have used a variety of notations and signed algebras [Bobrow 85]. For continuous state variables, the continuous domain is quantized into a discrete symbol set called a quantity space. For example, a spring force can be mapped into a quantity space of three values: positive for tension, zero for no force, and negative for compression. The confluence approach uses the $\{+, -, 0\}$ quantity spaces for qualitative values [Iwasaki&Simon 86; deKleer&Brown 85]. Qualitative process theory uses a notation that separates magnitudes of quantities from their signs [Forbus 84]. Qualitative simulation uses a notation that allows an arbitrary number of symbols in the quantity space and an associated direction of change {decreasing, steady, increasing} [Kuipers 85]. The quantity space for the spring might be $\{C_ultimate, C_linear, 0, T_linear, T_ultimate\}$ where $C_ultimate$ and $T_ultimate$ are the maximum compression and tension strengths of the spring, C_linear and T_linear are the limits of linear-elastic spring behavior in compression and tension, and zero is no spring force.

The last approach is used in this thesis because of notational clarity and the usefulness of the particular quantity space representation for engineering domains. The expressive power of this notation is greater than that of confluences due to the expansion of the quantity space beyond $\{-, 0, +\}$, and than qualitative process theory due to the inclusion of direction of change information. This notation also appears to be more readily un-

derstood and quickly grasped by novice users. Additional reasons for selecting qualitative simulation from the various qualitative reasoning approaches is that a standard version of this approach is rigorously defined and readily available [Kuipers 86].

Qualitative simulation, in particular Kuipers' QSIM algorithm [Kuipers 86; [Kuipers 87; Janowski 87], provides a formalized mechanism for deriving device behavior from a qualitative description of its structure that has the desired characteristics of being complete, limiting, and articulate. Qualitative simulation uses a description of the device within its domain in the form of a set of qualitative constraint equations and an initial state to derive a description of the behavior of a mechanism over time in terms of the changing values of its state parameters. QSIM computes and assembles the histories of the individual parameters by constraint propagation, to derive the possible behavior of device over time. This qualitative meta-level can be used to plan and reason about lower level quantitative knowledge of the physical world as illustrated in Figure 2-4.

2.6 Architecture for linking rules/qualitative/quantitative

Early applications of knowledge-based systems to medical diagnosis led to the realization that diagnostic or therapeutic programs must consider a case at various levels of detail in order to integrate overall understanding with detailed knowledge. Expert physicians appear to use several levels of detail, varying from a surface empirical association of symptoms and diseases, to a biochemical and pathophysiological interpretation. An approach using a multilevel causal network has been shown to overcome some of the limitations of the single level of detail approach for dealing with complex clinical situations and organizing large amounts of seemingly unrelated facts into coherent causal descriptions [Patil et al 82].

In a similar way, engineering problem solving utilizes multiple levels of understanding. Engineers attempt a solution at the simplest possible level and then use the ambiguities generated by the simple attempt to guide more sophisticated analysis. The initial problem definition and hypothesis generation steps are approached heuristically in light of accumulated experience. Creation of an abstract model and determination of important parameters, the kinds of physical behavior that dominate, and information that can be ignored are done qualitatively before final numeric methods are used to resolve ambiguity and provide quantified results. Engineers attempt a solution at the simplest possible level and then use the ambiguities generated by the simple attempt to guide more sophisticated analysis. For these reasons, a layered approach is utilized to support engineering problem

solving.

The three layered system architecture of CRACK shown in Figure 2-5 combines the disparate approaches of rule-based, qualitative, and numerical reasoning. The three reasoning layers are linked by one common representation scheme for the engineering models. The border shadings in Figure 2-5 refer back to the shading used in Figure 2-2 for the stages in construction and utilization of engineering models. The black bordered rule-based layer performs the heuristic tasks of problem definition, hypothesis generation, and drawing conclusions. The gray bordered qualitative layer constructs the analytic models and performs model validation/refinement. The white bordered quantitative layer performs numeric analysis and evaluation of analytic results. The arrows between the three reasoning levels show the transitions as the state of problem knowledge is mapped from one abstraction space to another. The arrows between the three layers and the one common representation show how communication occurs between the disparate reasoning levels through the homogeneous language used to describe the engineering models.

The need to limit search is greatest during the stages of definition/modeling/evaluation. The top, predominantly declarative, layer uses a backward chaining rule-based paradigm to process domain knowledge that is predominately symbolic. The domain rules are applied to gather data needed to pose the complete problem, to customize engineering models of the various hypothesized situations, and to evaluate the results of the numerical simulations. The rules are partitioned into subsets to express the structure of the domain knowledge and to keep the size of each rule set small.

The mid-layer uses qualitative descriptions to express the relationships among the physical parameters. This layer reasons about mechanisms based on the relationship between structure and behavior using qualitative simulation [Kuipers 86] to determine the system's physical state over time.

The root layer is made up of quantitative simulators to represent domain knowledge that is largely procedural. This layer contains the procedures needed to mathematically simulate the behavior of the engineering models.

All layers share a case-specific description of engineering models under examination. These models are built up from a library of component templates and hold the

state variables for the multiple hypotheses as the solution progresses.

Each method is a better tool for a different phase of the engineering problem solving process. Rules are good for generating hypotheses and defining initial areas in which to search for a solution. Qualitative reasoning is suited to enumerating possible behaviors and focusing in on the most promising engineering models. Quantitative methods are best used to resolve ambiguities and arrive at precise answers. Using the right tool for the task at hand increases the power of the entire computer application . The ability to direct search by reasoning with a simplified engineering model and then verifying, revising, and refining the rough model by recourse to more exact analysis lies at the heart of engineering problem solving. A computational model that embodies this approach should be applicable to a wide variety of engineering problems.

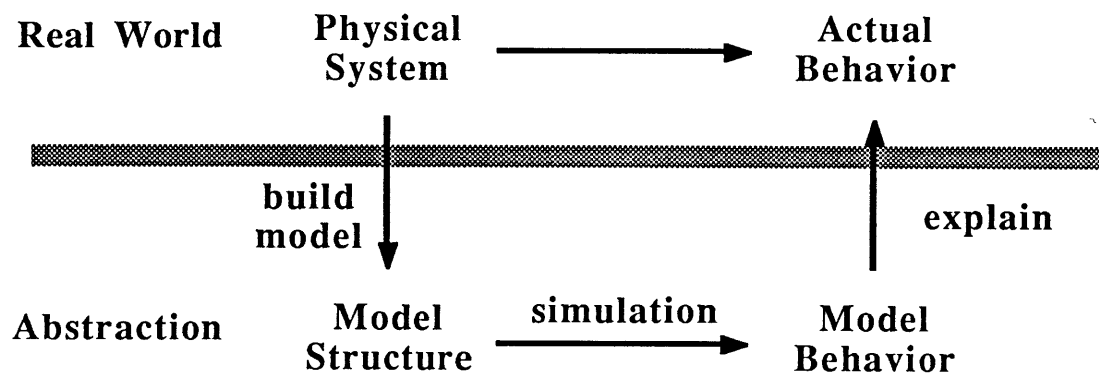


Figure 2-1:
Abstract Models [Kuipers&Patil 87]

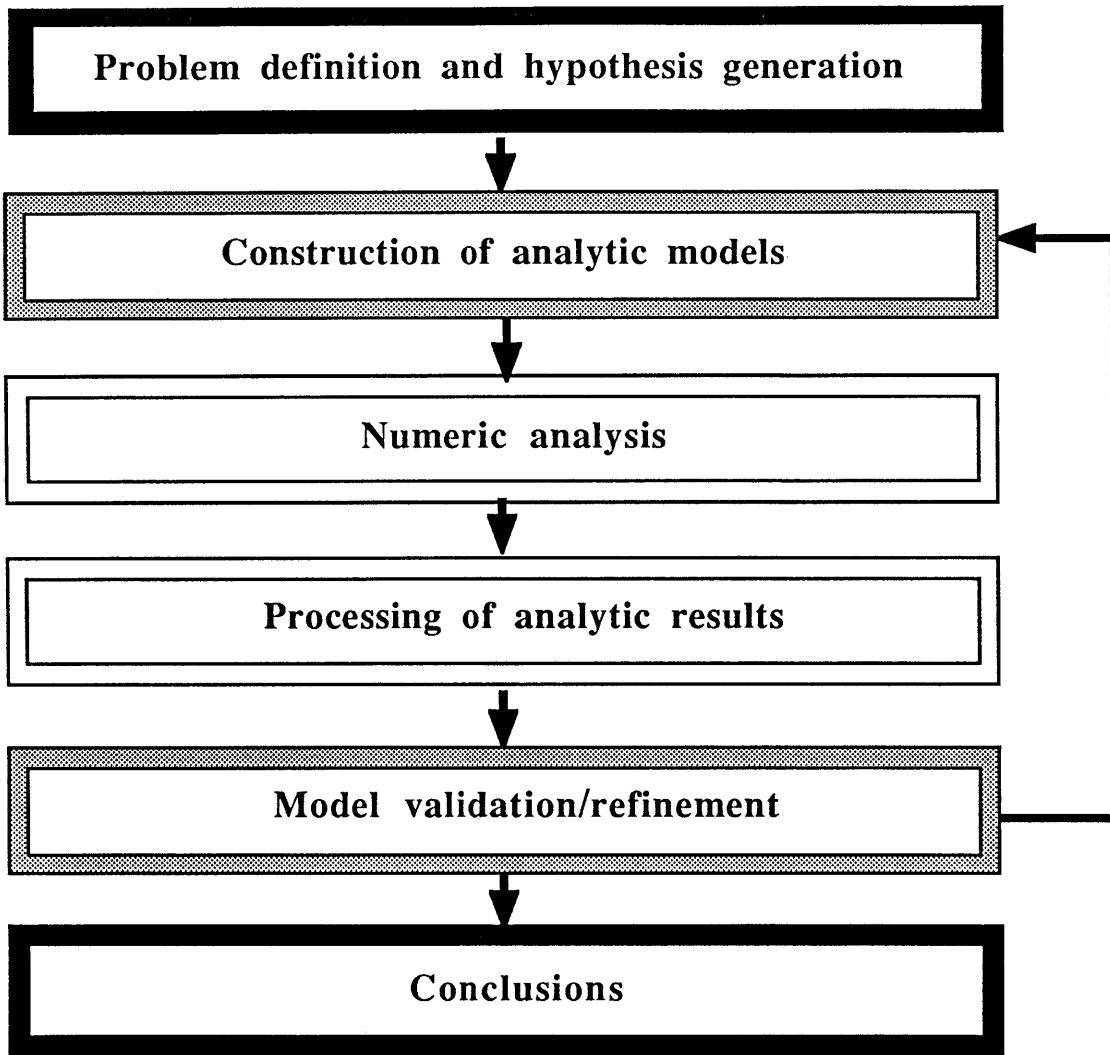


Figure 2-2:
Conceptual framework for constructing
and utilizing engineering models

Forward chaining:

Antecedent-consequent rule statement:

**Rule 1: If A can be established
 and B can be established
 Then C can be concluded**

**Rule 2: If C can be established
 Then D can be concluded**

Forward chaining:

Begin with the observed facts A and B

Use Rule 1 to conclude C

Use Rule 2 to conclude D.

Respond to the current situation (data driven)

Backward chaining:

Consequent-antecedent rule statement:

**Rule 1: Conclude C
 If A can be established
 and B can be established**

**Rule 2: Conclude D
 If C can be established**

Backward chaining:

Begin with the goal to reach D

Use Rule 2 to set subgoal C

Use Rule 1 to set subgoals A and B.

Attempt to achieve a desired objective (goal driven)

**Figure 2-3:
Forward and backward chaining inference procedures**

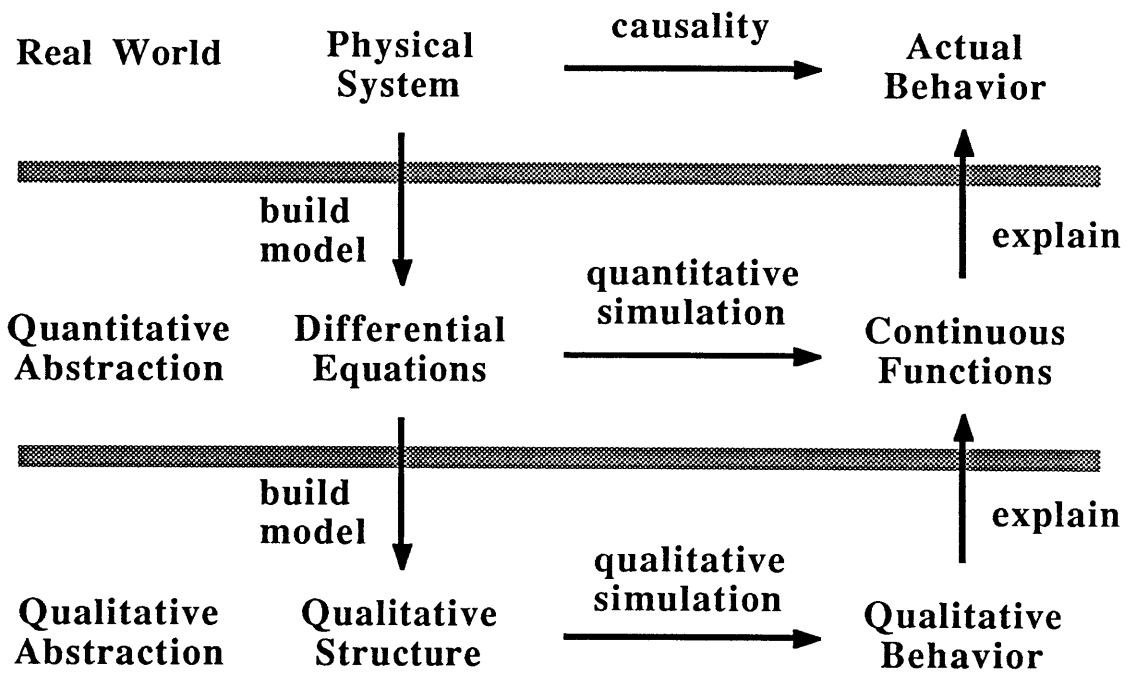


Figure 2-4:
Qualitative Models [Kuipers&Patil 87]

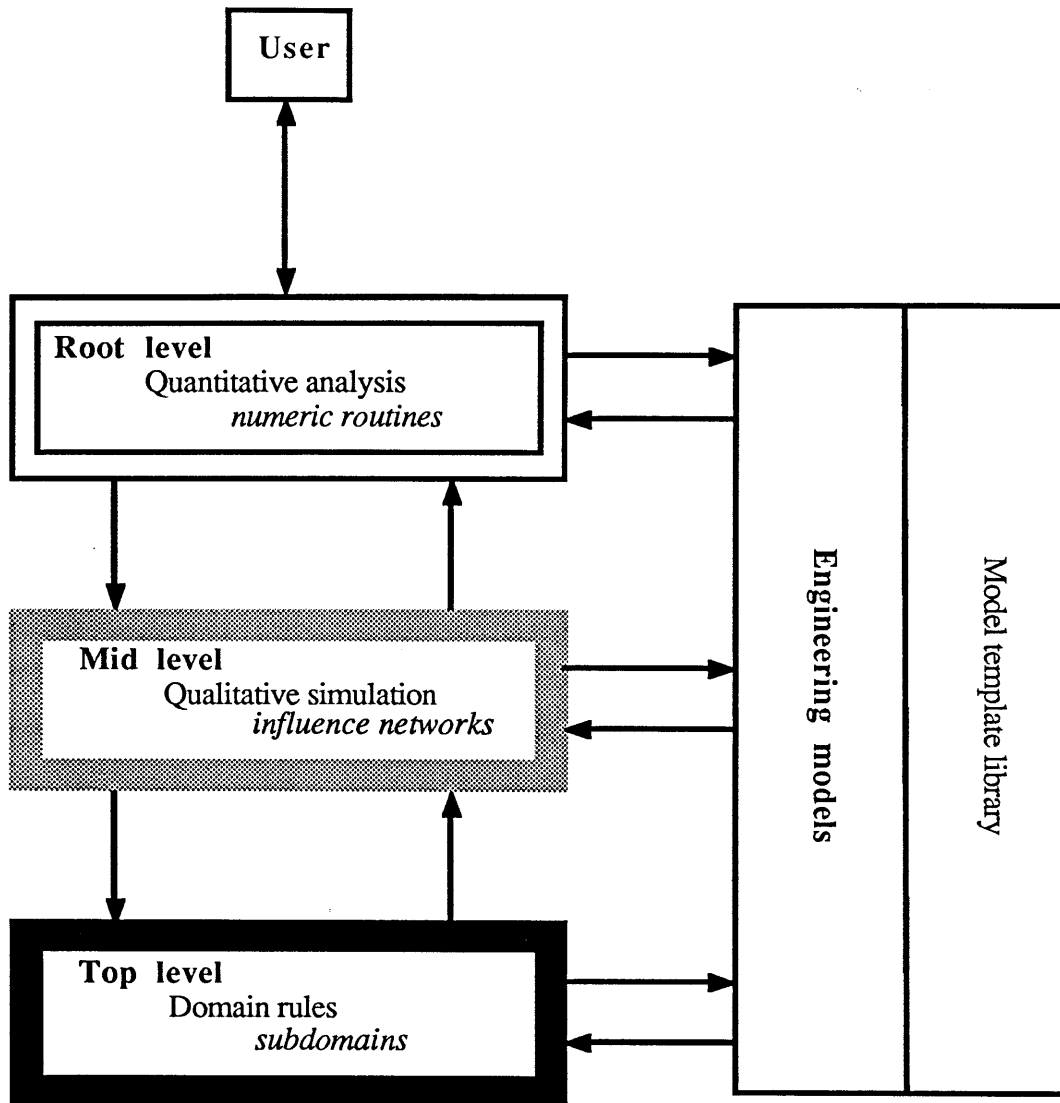


Figure 2-5:
Layered system architecture structure of CRACK

Chapter 3

Problem domain: cracking in highway bridges

3.1 Choice of problem domain

The domain of fatigue and fracture in steel bridge details was chosen as the test domain for this thesis because of the confluence of professional need and technical feasibility. The problem area is a significant and substantive one of inherent engineering interest and usefulness, representative of a broad class of engineering problems. There are many reasons why this problem domain is well suited for a knowledge-based approach, but three deserve special mention. First, this is a practical problem where the knowledge exists but is frequently not utilized. Second, the knowledge is of diverse types (statistical, heuristic, engineering principles, etc.) but circumscribed and well contained so that it is possible to provide a complete coverage of the knowledge needed to solve the problem. Third, there are multiple uses of the same knowledge; for analysis of failures, for determination of causes of distress and prescription of fixes, for prediction of remaining service life, and for verification and optimization of design.

3.1.1 Need for better solutions in engineering practice

Most bridge disasters in last 100 years have been due to either design methods that did not account for structurally significant behavior or poor maintenance. Major changes in policy for bridges is usually spurred by a fatal collapse, as was the implementation of the National Bridge Inventory in the United States. In 1967 only 17 states had federally acceptable bridge inspection programs, while in 1968 all 50 states did. This was in response to the December 15, 1967 collapse of the Silver Bridge between Gallipolis, Ohio and Point Pleasant, West Virginia in which 46 people died and 9 suffered injuries [Ross 84].

Any innovation leading to better utilization of limited resources to maintain highway bridges would be a timely engineering contribution. In 1986, of the 585,059 interstate, state, city/county/township bridges over 20 feet in length, for 50 states and the District of Columbia, 42% were rated substandard. In 1987, of a total of 586,680 bridges, 41.5% were rated substandard. During one year, a minuscule gain was made on the problem of one-half of one percent. Federal money authorized for bridge work for 1988 was \$1.6 billion, a cut of almost 16% from the \$1.8 billion authorized for 1987 [Better Roads 87].

The identification, inspection, evaluation, and repair of fatigue and fracture damage in steel bridges are all complex engineering tasks. The number of qualified experts is far below the level required to deal with the large bridge population. The need for improved inspection, maintenance, and design of bridges was dramatically demonstrated by the recent catastrophic failures of the Mianus River Bridge [ENR 83], the Schoharie Creek Bridge [Thornton et al. 88], and other bridges [Hanson 84]. The risks of fatigue and fracture failures will increase as the existing bridge population ages, and as older bridges exceed their design life.

Since steel highway and railroad bridges carry cyclic loads and exist in adverse environments, the possibility of crack initiation, growth, and fracture must be considered in order to provide a safe structure. A tool that can identify likely cracking problems that may arise in a particular steel bridge could be used to provide inspection guidance and optimize inspection schedules. Better use of the limited resources available for bridge inspection would yield better data to make maintenance designs, which would in turn optimize the use of scarce maintenance funds. Use of a computational tool that provides better dissemination of knowledge about cracking in bridges would lead to more efficient allocation of limited resources and a safer transportation system.

Most practicing engineers responsible for bridge inspection, assessment, and maintenance are not fracture mechanics specialists, and rely on standardized design codes and guidelines [AASHTO 85] to compensate for the lacking specialized knowledge. The underlying fracture mechanics knowledge is basically "compiled" by a group of domain experts into a form that is usable by engineers without specialized training. There are obvious drawbacks to this approach. The standardized provisions are usually tailored for use in design and are not readily adaptable to other tasks needing the same underlying domain knowledge. In an effort to cover the common range of problems in a simple manner, groups of similar problems are lumped together in a way that loses precision. The problem class may be incorrectly identified. Special cases are not covered. The standardized approach may stifle innovation and understanding by the practicing engineer.

Major cracking failures of girders at welded details may occur on bridges of modest age. No indications may be found before a cracking failure, even if an inspection was recently performed. For example, a crack extending the full depth of a 60" deep

girder was discovered in September 1987 on an exit ramp of the Dunn Memorial Bridge over the Hudson River at Albany NY [Morelli 87]. The bridge was only 20 years old and had been inspected in December 1985. The girder cracked along the weld line of a plate attached to the top of the lower flange with a transverse weld. This welded connection plate detail, for attachment of diagonal X bracing between the multiple girders, is no longer used by the DOT for bridge construction.

3.1.2 Required knowledge is diverse and circumscribed

Cracking failures, rendering an entire bridge unsafe due to localized weakness, point out the need for not only thorough inspection, but also better *understanding* on the part of bridge engineers with the factors influencing crack behavior. In general, engineering applications run ahead of the establishment of underlying fundamental principles. This prescientific stage of engineering is fraught with difficulties since governing parameters are not understood, but effective engineering solutions can be reached based on empirical examples instead of theory. However, for many problems involving fracture and fatigue, and for bridge cracking in particular, frequently knowledge about how to avoid the problems *exists* but is not *utilized*. The cross disciplinary nature of the problem as well as technology transfer barriers cause existing expert knowledge to be overlooked, underutilized, and simply be too scarce and costly.

The costs of fatigue and fracture are high. A 1982 National Bureau of Standards study estimated the total annual direct and indirect costs to the US economy of approximately \$119 billion, or 4% of the gross national product. This is staggering even when noting the broad basis used, including: prevention (conservative design, excess material cost, testing and inspection, maintenance, and service life limits); occurrence (loss of equipment, down-time, loss of product, clean-up, injury, and loss of life); and associated costs (research and development, technical support). This study also concluded about half of this cost could be avoided by utilizing existing knowledge about fatigue and fracture.

Four major possible failure modes for structures are [Barsom&Rolfe 87]:

1. General yielding or excessive plastic deformation.
2. Buckling or general instability, either elastic or plastic.
3. Subcritical crack growth leading to loss of section or unstable crack growth.
4. Unstable crack extension, either ductile or brittle.

Of these four modes, engineers tend to concentrate on the first two, assuming proper material selection and design stress levels will prevent the other two, *which is not*

always true. Fracture mechanics elucidates the importance and interaction of crack size, fracture toughness, and applied stress. The area of fracture mechanics as applied to bridges is a relatively self contained and well described engineering domain, characteristics necessary to make the construction of a relatively complete computational model tractable. The discipline is at a level of maturity appropriate for a KBS approach: there are engineering experts who regularly produce results that are generally recognized to be correct, however the application of fracture mechanics to the area of steel bridges is not a purely algorithmic discipline. The engineering process is not fully described by methods derived from first principles, but instead relies on heuristics at critical places.

3.1.3 Multiple uses of same knowledge

Problems concerning fracture in steel bridges arise in various contexts. An engineer may need to perform any of the following tasks depending on the particular situation:

1. Failure analysis: Given a failure, find the cause
2. Diagnosis and prescription: Given an existing bridge with cracking, determine the cause and design a fix
3. Prediction: given an existing bridge, determine the likelihood of problems developing
4. Design verification and optimization: given a proposed design, critique it and suggest improvements to prevent failures from occurring.

All these tasks require specialized engineering knowledge about the field of fracture mechanics as applied to bridges. Fatigue and fracture mechanics concepts are employed to determine why structures failed, to evaluate the integrity of structures containing defects, and to design new structures [Hudson&Rich 86] [Abelkis&Hudson 82]. Representation and reasoning mechanisms that are rich enough to support multiple uses of the knowledge base are clearly desirable. In this way the same knowledge base can be used to solve analysis, diagnosis, prediction and verification tasks. (The areas of prescriptive treatment and design optimization are the least well formulated and most peripheral of the engineering problems postulated and will not be included as implementation goals for this project). Due to the potentially life critical nature of bridge evaluation, the ability to construct an explanation that is acceptable to an engineer is a necessary feature for system acceptance and effectiveness.

The objectives of a *failure analysis* are to: explain the sequence of events that occurred during failure; establish the cause; and prevent the recurrence of failure. After a collapse it can be difficult to determine the cause. In the collapse, many joints and pieces may be broken. The new breaks rust rapidly in rainy or damp weather so it is hard to separate old breaks from new. Under these conditions, identification of the critical element and failure initiation site is challenging. There may be several possible contributing factors such as corrosion, fatigue and overloading. Some hypotheses, such as overloading, can be ruled out by analysis if the conditions at failure are known, but in general physical evidence is needed. For example, conclusive proof of fatigue requires the examination of the critical element [Ross 84]. Surface morphology, including fatigue striations, can be used to estimate the cyclic stress levels, the time to grow the crack, and the magnitude of the local mean stress present. Stress analyses incorporating stress concentration, fatigue, and fracture calculations are an important aspect of failure analyses.

Diagnosis and prescription are required to determine the fitness for future service of a cracked structure. In order to develop a successful fatigue cracking retrofit, the phenomenon inducing the cracking stresses must be identified. To evaluate the significance of a crack, a detailed inspection must establish crack location, length, and width. In-service inspection methods include: visual inspection with 10X magnification, dye penetrant, magnetic particle testing, ultrasonic testing, and radiography. Crack sizes that are detectable by a field crew must be assumed to be at least 1 inch long and about 1/2 inch deep [NCHRP206 79]. Once the cracking cause is correctly diagnosed, a variety of retrofitting techniques may be used to extend the fatigue lives of welded steel bridges [Fisher&Mertz 85].

Prediction of in-service behavior using the principles of linear elastic fracture mechanics with emphasis on subcritical flaw growth provide a rational basis for establishing an inspection program which assures continued operability without threat of catastrophic failure. Durability is the ability of the structure to operate and to be maintained during the planned service life within acceptable economic limits. Damage tolerance is the ability of a damaged (due to fatigue, corrosion, or accidental damage) structure to operate safely between inspections. Durability and damage tolerance of a structure translate specific guidelines on flaw size, location, and criticality that are important in setting nondestructive inspection intervals and sensitivity limits. Since small cracks missed during one or more inspections would eventually grow into large cracks or

failures, setting inspection requirements must account for sizes and types of cracks that are likely to be missed as well as those that are likely to be found. In the vicinity of complex connections, some cracks may become very large before they can be detected [Baldwin et al 81].

Visual exams are the dominate inspection method, since it is not practical or cost effective to use other non-destructive tests on the whole bridge. Knowledge about *where to look* is essential in planning a meaningful inspection. Defect-prone areas may be determined in advance based on knowledge of bridge types, joint, and member details, fabrication, and in-service conditions. An expert in bridge inspection articulates some of the knowledge about where to look for fatigue damage [Better Roads 87]:

"On older, early-vintage welded bridges, there is bad geometry on many member intersections that are places to look. Any place that collects dirt, that collects water, or has high stresses or combined stresses are targets. Complex stress patterns due to connections of diaphragms and cross beams are other points".

Proper test method selection also requires non-destructive testing and evaluation about *how to look*. In addition, the *hands-on* experience of seasoned inspectors is essential to reliably detect cracks under field conditions [Verma&McNamara 88].

The time, effort, and money used for bridge inspection should be in proportion to the problem [Ross 84]. The most important things bridge owners can do, but don't always, are to *clean and paint* the structures [Better Roads 87]:

"When you sandblast the steel to get off the dirt, rust, and old paint, that's when you find the defects you might not otherwise locate. You can't hand clean a bridge during a routine inspection - you haven't got that much money or time."

The author, from personal experience, strongly seconds this expert's opinion. In addition, the corrosion damage which takes place due to lack of timely paint maintenance dwarfs the fatigue damage.

Design verification and optimization must consider the mechanical, material, and environmental variables which influence fatigue and fracture. Experience has shown that in many instances designers have reacted to service fatigue failures in welded structures in the wrong way [Tubby&Wylde 86], with the result that design changes made have not had the desired effect. Fracture mechanics techniques provide the understanding necessary for evaluating the benefit of proposed design changes when service problems are encountered. Fatigue performance and life cycle costs bear on material selection. Assessing defect significance and setting defect acceptance standards for flaws either inherently present or introduced during construction or service is rationally accomplished

with fracture mechanics, highlighting the need for inspection and quality control, not only during maintenance but during fabrication and construction, and provision for access to critical members to facilitate inspection and maintenance. Design of long-life fatigue and fracture resistant structures with a credible safety record requires diligent attention to detail design, manufacturing, maintenance, and inspection procedures.

From a knowledge engineering point of view, the most straight forward task is the failure analysis, since the procedures to follow are fairly well defined [Ryder et al. 85]. In addition, bridge failure analyses are well documented so case studies for knowledge extraction and system verification are readily available. The diagnosis task, excluding retrofit design, is an expanded version of failure analysis where symptoms of structural distress (cracking) are visible, but complete failure hasn't occurred. Detailed case studies, including structural description, summary of cracking and material characteristics, crack analysis using fracture mechanics models, and review of the repair and retrofit scheme, are available.

Performing a failure analysis is a very useful method of assessing the capabilities and limitations of domain knowledge. Human experts are made aware of the limits of their own knowledge when confronted with failures that they cannot satisfactorily explain. Human experts then use the unsatisfactory failure analysis as a case to expand the domain knowledge. Learning new failure modes from unsuccessful failure analysis is beyond the scope of this project. The preferred situation is not just to explain a past failure, but to keep failures from occurring. Failure prevention requires good design and maintenance. A system capable of failure analysis can also be utilized in a predictive manner. A proposed detail can be evaluated to verify that known failure mechanisms have been adequately addressed in its design. The construction documents and service data of an existing bridge can be examined to pinpoint areas for special attention during in-service inspections.

3.2 Envisioned use

The prototype system CRACK is meant to make specialized knowledge accessible to bridge engineers. The intended use is as an *assistant* to a bridge engineer working primarily on the inspection and evaluation of existing steel bridges, with some new design evaluation work. The envisioned user of this tool is a bridge engineer with an academic background equivalent to a bachelor's level and with adequate training in the area of bridge engineering, but who does not necessarily have specialized training in

fracture mechanics. A delivery version tool, usable in the workplace would need to: identify those bridges requiring in-depth inspection for fracture and fatigue damage, recommend inspection strategies for these bridges, perform a structural evaluation based on the inspection results, and suggest guidelines for restoration work. To demonstrate the feasibility of a knowledge-based approach to this problem, a research prototype system was developed for this thesis which only identifies the type of cracking damage to which a particular bridge is susceptible. The system provides information on probable crack cause, location, and danger. This information can be used by a bridge engineer to aid in scanning the bridge population to identify high risk bridges and recommending inspections so that further information acquisition is well-tailored to the specific class of potential cracking problems.

The choice of the role of the knowledge-based system is a choice between philosophies on the appropriate use of the AI technology to aid in the engineering process [James 88]. Ideally, the computational tool is complementary to the human intelligence. The intent is to support and aid the engineer, making knowledge readily available, contributing to the depth of problem understanding and number of engineering alternatives explored. The computer is therefore cast in an assistant's role, providing smooth access and execution of the traditional tedious number crunching tasks, making specialized domain knowledge available, and providing advice in problem solving, but the engineer remains the controlling decision maker.

3.3 Related work

Another project in the area of computer tools to address the problem of highway bridge fracture and fatigue is underway at the NSF sponsored engineering research Center for Advanced Technology For Large Structural Systems at Lehigh University [Chen 87; Chen et al 87; Chen&Wilson 86]. The purpose of the Bridge Fatigue Investigator , whose primary developer is Stuart Chen, is to assist a bridge engineer in inspecting for fatigue damage in steel girder bridges, and evaluating such structures for their susceptibility to fatigue and fracture problems. It is meant to be used for two situations: pre-inspection and post-inspection. The pre-inspection portion identifies those connection details on the given bridge most susceptible to fatigue, and recommends specific features of those details be particularly closely scrutinized. The required inputs are the topology of the bridge components, other design information such as materials and date of construction, and service information such as traffic loads. The output gives guidelines about where to look and what to look for, a customized checklist for the specific bridge to focus

the inspector attention on the most critical areas in limited time available for in-service inspection. The post-inspection portion takes input about an observed crack and provides advice in diagnosing its cause, assessing its seriousness, and suggesting what ought to be done.

The formalization of expert domain knowledge which is being done in this project is an essential contribution toward the goal of increasing the reliability of fatigue damage evaluations and easing access to relevant information to reduce the risks of catastrophic bridge failure and increase productivity of bridge engineering professionals. However there are shortcomings. The implementation is not knowledge-based in the usual meaning of a clean separation of domain knowledge from the inference mechanism used to process that knowledge. Due to the perceived restrictiveness of commercial expert system shells, the decision was made to implement the Bridge Fatigue Investigator in Prolog. There is not a clear separation between the knowledge base and the way the knowledge is processed. The implementation is thus more a bare Prolog program, than a knowledge-based system, a distinction which is discussed in Section 5.2.7.1.1. Explanations are either pre-defined for static queries of the knowledge base or rely on the trace facilities of the Prolog environment for dynamic queries about the line of reasoning. Although some of the pre-defined explanation facilities, like generic pictures of structural details, are useful in this domain, they are still canned, not generated from an explicit knowledge representation. The coupling of symbolic and numeric features is also weak, for example the calculation of remaining fatigue life of a given connection detail is not dependent on a crack size, but on a detail fatigue category. The project personnel recognize the need for a model-based foundation using fundamental laws and theories, and integral role of performing numerical calculations, but have not attempted to address these needs. This project may be summarized as an effort to incorporate expertise about bridge fatigue and fracture into a symbolic program. As such, it may be viewed as orthogonal and complementary to the effort of this thesis. The system architecture developed in CRACK provides a truly knowledge-based framework that can incorporate the expertise formalized in BFI. Both projects are research prototypes. Development of a bridge engineering delivery system incorporating the results of both BFI and CRACK would be much more productive than utilizing only the results of one of the projects.

3.4 Application of three-level framework to domain knowledge

Failure analysis of cracking in a plate girder bridge is a typical engineering problem. There are certain tools available to the engineer that are expressed in

mathematical terms. These tools can be used to simulate the physical response of the actual plate girder. The changes that the girder is subjected to are expressed in terms of physical parameters such as load and temperature, the response is described in terms of stress and strain. To be able to apply available mathematical tools to the problem at hand the engineer constructs an analytic model to capture the relevant physical features in a suitable way for the particular mathematical tool. The analysis is then performed, and the results are evaluated and interpreted to describe a solution or direct model refinement and additional analysis. The model construction is not algorithmic, but is guided by largely heuristic knowledge that usually travels under the rubric "engineering judgment". To apply the three-level architecture shown in Section 2.5 to the test bed problem domain of fatigue and fracture in bridges, the domain knowledge must be structured according to the heuristic/qualitative/quantitative framework and an appropriate representation developed for the engineering models which act as the means of communication between levels.

The heuristic knowledge represented with rules at the top level is of three types: protocol, hypothesis generation, and physical parameters controlling cracking. The high level protocol rules (Section 5.2.1.2) post the initial appropriate sequence of goals to accomplish the different tasks of failure analysis, prediction, and design critique. Knowledge about causes of cracking in bridges (Section 4.7) is structured hierarchically and used to generate hypotheses during the first stage of engineering problem solving. Knowledge about each of the four major parameters affecting the crack growth life and residual strength capacity of structures (initial flaw size, stress history, material properties, and structural geometry) (Section 4.4) is grouped into a separate subdomain rule set.

Causal knowledge is represented by a set of qualitative constraint equations which capture the relationships between the significant physical parameters within the domain. The state of the plate girder is described with a set of parameters such as crack size and stress. Part of a possible qualitative description of a plate girder experiencing fatigue crack growth is: "The crack has entered but not penetrated the bottom flange". An example of a qualitative relationship between state variables is: "An increase in stress range leads to an increase in crack growth rate". The qualitative level represents this domain knowledge (Section 5.2.2.2) with constraint equations which define a network of influences among the parameters.

The central numeric ability required from the system is the computation of crack growth, both for stable and unstable propagation (Section 4.5). Crack behavior over time is computed quantitatively by small functions which take all input from, and post all results to, the data structures of the engineering models. The model library is constructed of plate components and structural connectors from which specific bridge detail configurations can be assembled (Section 4.8.4). These models parallel the physical structure of the actual bridge detail, aiding reasoning about issues such as connectivity and spatial proximity.

Chapter 4

Domain knowledge: fatigue and fracture of welded steel plate girders

4.1 Knowledge sources and acquisition

The major source of domain knowledge for the prototype system is the rich technical literature available on the problem of cracking in steel bridges. There is an extensive literature in both the general realm of fracture mechanics and failure analysis and the specific topic of fracture and fatigue in steel bridges. Textbooks, handbooks, monographs, and journal articles provide generally applicable information on fracture mechanics and failure analysis. Design codes, manuals, and guidelines provide information on fracture and fatigue behavior in steel bridge structures. Failure analyses and laboratory test programs provide information on cracking behavior in particular cases. Of special value are case studies of bridges that have experienced cracking. The case studies which were used in most depth were selected from a collection of 22 studies [Fisher 84], each describing the structure, summarizing cracking and material characteristics, analyzing the crack using fracture mechanics models, and reviewing the repair and retrofit scheme used. The reasons for focusing on this set of cases during the compilation of the prototype knowledge base are first, using a unified presentation by a single widely recognized domain expert gives reasonable assurance of creating a self-consistent knowledge base, and second, using a case study approach gives a set of test cases against which the system can be tested to verify adequate performance. Unfortunately, published literature frequently does not contain sufficiently detailed information to reconstruct complete failure analyses. To fill in the gaps in the published studies, unpublished research reports were consulted where necessary. In addition to knowledge acquisition from literature, a summer was spent accompanying teams actually performing field inspections for in-depth condition assessment of highway bridges. This field experience served to temper and add perspective to the information gleaned from the relevant literature.

4.2 Requirements for structural integrity

An engineered structure, such as a bridge, must satisfy three different measures of structural integrity: strength, stiffness, and longevity. The static and dynamic strength must be sufficient to continue service after carrying the design level of load, and must not fail under the ultimate level of load (the design load times the appropriate factor of safety). The rigidity must limit deflections to a range that ensures serviceability under load.

The life of the structure must be long enough to meet the specified period of use in the face of deleterious effects of the service environment, such as metal fatigue.

A structure with adequate strength to avoid plastic yielding and excessive elastic deformation can still fail catastrophically by fracturing. Fracture is the unstable growth of a crack. Fracture occurs when existing cracks, often flaws resulting from welding, grow to critical size. Crack growth becomes unstable when the amount of energy released by the advancing crack is greater than the amount of energy absorbed by separating the material to form the new crack surface. The initiation of unstable crack growth (fracture) is described by the equation [Kanninen&Popelar 85]:

$$K = K_C$$

where K = stress intensity factor

K_C = fracture toughness

This equation assumes that linear elastic fracture mechanics applies (i.e. where the inherent inelastic deformation surrounding the crack tip is small) as explained in Section 4.5. Both K and K_C are measured in units of stress $\sqrt{\text{length}}$ (e.g. ksi $\sqrt{\text{in.}}$). The stress intensity factor K is directly proportional to the stress at the crack tip. K depends on the crack size, component dimensions, and applied stress. The stress intensity factor K is discussed further in Section 4.5.2. The left-hand side of the equation is a measure of the stress state. The fracture toughness K_C is a measure of resistance of the material to the advance of a crack. K_C is a pseudo-material property that depends on the temperature at the crack tip, the rate of loading, and the thickness of the cracked section. The fracture toughness K_C is discussed further in Section 4.8.3. The right-hand side of the equation is a measure of the material's experimentally determined fracture properties. Fracture occurs when the geometry and loading raise the stress characterizing parameter K to equal the material property K_C . Figure 4-1 [Broek 85] illustrates the failure analysis sequence for fracture, and shows the analogy to plastic yield. The stress intensity represents the mechanical side of the equation, determined by the structural geometry and loading. The fracture toughness represents the material side of the equation, based on the inherent available resistance to fracture.

Cracks may also form and grow stably for values of K well below K_C , if a structure is subjected to fluctuating stresses or to an aggressive environment. Fatigue is the

formation and growth of cracks under cyclic stress. Corrosive environments can also lead to crack formation and growth, but this phenomenon is excluded from this thesis. Cracks grow in a pure metal by the tensile cycle producing a plastic zone which makes the crack open an amount ∂ . The compressive cycle squeezes the crack shut, and the new surface is folded extending the crack forward roughly proportional to ∂ . This crack growth mechanism is illustrated in Figure 4-2 [Ashby&Jones 82]. Real engineering alloys have inclusions, so the plastic zone has voids, and the crack grows a little faster than in a pure metal, due to the holes joining the advancing tip. Different types of structures can be grouped into fatigue of uncracked components and fatigue of cracked components. Fatigue of cracked components may be further divided into high cycle fatigue and low cycle fatigue. Summarizing these fatigue categories are summarized below [Ashby&Jones 82]:

Fatigue of uncracked components: No cracks preexist; initiation-controlled fracture. Examples: almost any small component like ball races, gear teeth, axles, crank shafts, drive shafts.

High cycle fatigue: Fatigue at stresses below general yield; $\sim > 10^4$ cycles to fracture. Examples: all rotating or vibrating systems like wheels, axles, engine components.

Low cycle fatigue: Fatigue at stresses above general yield; $\sim < 10^4$ cycles to fracture. Examples: core components of nuclear reactors, airframes, turbine components, any component subject to occasional overloads.

Fatigue of cracked components: Cracks preexist; propagation controlled fracture. Examples: almost any large structure, particularly those containing welds: BRIDGES, ships, pressure vessels.

Most bridge fractures are preceded by fatigue crack growth [Fisher 84]. As the crack length increases under cyclic loads, the stress intensity factor K also increases. If K increases enough, it will exceed the fracture toughness K_c and fracture will occur.

4.3 Fatigue design philosophies

There are two principle engineering methodologies for dealing with fatigue in structures [Anand&Parks 86; Gallagher et al. 84]: 1) defect free and 2) defect-tolerant design and maintenance. These methodologies for design/analysis parallel the categorization of fatigue presented in Section 4.2. The defect free approach assumes: 1) no crack-like defects are presumed to preexist, and 2) the total fatigue life is taken as the sum of an “initiation life” (the time in service taken for a crack to form) plus a “propagation life” (the time for an existing crack to grow to critical size). In many applications, only

initiation life is considered. Typical examples are most small components which are not safety critical. The defect free methodology, which may be further subdivided into high and low cycle classes, is not appropriate for bridges.

Defect-tolerant design and maintenance assumes: 1) cracks are presumed to pre-exist, 2) the fatigue life consists solely of the fatigue crack propagation of the initial population of cracks (“propagation life”), and 3) periodic in-service inspection and maintenance is possible and usually required. Typical examples are large, fabricated structures such as aircraft, ships, pressure vessels, and BRIDGES, where welds are likely sites for initial defects, and the large size of the components may permit substantial subcritical crack growth, so that the enlarged defect can be detected and repaired or replaced well before it reaches a critical dimension.

There are two sub-approaches for the defect-tolerant methodology: 1) fail-safe design, and 2) safe-life design and maintenance. In fail-safe design the structure should possess a sufficient redundancy of elements or components to provide assurance that, for a specified operating load, the failure or fracture of any single element or component will not lead to catastrophic failure of the structural assembly. This may require a very high degree of conservatism in design. The approach of safe-life design and maintenance is that in a specified operating interval (either total operating life, or more commonly, an operating interval between scheduled shut-down, inspection, and maintenance procedures), no preexisting crack of specified size, location, and orientation should grow to a size at which a specified load would cause the element containing the crack to fail. This requires the integration of a material fatigue crack propagation law relating crack length, a , to number of load cycles, N .

A defect-tolerant methodology safe-life approach is taken for dealing with fatigue in bridges. Although sufficient redundancy to preclude catastrophic failure is indeed a bridge design objective, the design of each element uses the safe-life approach. In order to assure that a crack does not grow to cause an element to fail during a specified service interval, it is necessary to understand what physical parameters influence fatigue and fracture (Section 4.4), and how to quantify those effects (Section 4.5).

4.4 Parameters governing crack behavior

The four major parameters affecting crack growth life and residual strength ca-

capacity of structures [Gallagher et al. 84], shown in Figure 4-3, are:

- 1) Initial crack size, which is a measure of quality.
- 2) Stress history, which is a measure of usage and location.
- 3) Material properties, which are a measure of material resistance to cracking.
- 4) Structural properties, which are a measure of the geometric configuration in the vicinity of the crack.

To prevent a failure due to cracking from occurring, it is necessary that the available strength capacity must be greater than the applied load at all times. The condition for fracture failure was stated to be $K = K_c$. For a fixed configuration and loading, a smaller initial crack size will result in a greater number of stress cycles before failure. For the same pattern of loading, the smaller critical crack size will give a higher residual strength capacity at any time, and thus a longer service life. The stress history depends not only on the loadings to which the bridge is subjected over time, but also to the physical location of the specific piece of material in the overall structure. The actual loading history for an in-service highway bridge is of course not known, but good approximations may be made based on traffic load surveys conducted by agencies such as the Federal Highway Administration. Crack growth rate and fracture toughness are properties that must be experimentally determined for each material. Structural properties are the most complex of the basic parameters. The structural properties of a detail include all the geometric and spatial information necessary to fully describe and dimension the configuration including the crack and method of load application. The structural properties determine the magnitude of stress concentrations. A detail with lower stress concentrations will have a longer life and higher residual strength.

4.5 Elements of linear elastic fracture mechanics

4.5.1 Definition of LEFM

Having identified the major physical parameters affecting fatigue and fracture, means of quantitatively relating them to crack growth and failure conditions are needed. Fracture mechanics is the quantitative analysis of the mechanical process that leads to fracture failure. The process of crack growth by fatigue as well as the final fracture are included. Fracture mechanics models of cracks are used to evaluate and assess the fatigue and fracture behavior of bridge details for both stable and unstable crack growth.

Fracture mechanics based on the theory of elasticity using linear stress-strain ma-

terial behavior is called linear elastic fracture mechanics (LEFM) [Broek 85]. The utility of an elastic approach depends on the extent of the region of inelastic yielding being small compared to characteristic lengths for the problem geometry, i.e. for inelastic deformations which may be classed as Small Scale Yielding (SSY) [Kanninen&Popelar 85]. Under SSY there is a limited amount of plastic deformation occurring at the crack tip. The plastic region is contained within the elastic crack tip stress field. The enveloping stress field must also be close enough to the crack tip to have the stress intensity factor accurately approximate the actual stress field.

4.5.2 Stress intensity factors

LEFM implies the stress at the crack tip is proportional to a single parameter, K , the stress intensity factor. The state of stress at a crack tip in a loaded isotropic linear elastic body can be characterized by the three parameters: $K(I)$, $K(II)$, $K(III)$. These parameters are the stress intensity factors associated with the three respective independent modes of deformation:

- mode I: tensile opening
- mode II: in-plane shearing
- mode III: anti-plane shearing

The magnitude of the stress intensity factors scale the local stress and displacement fields in the vicinity of the sharp crack front. Most engineering applications may be characterized by mode I deformation alone, not only because it is the most common single mode, but also because in general a crack subjected to combined mode loading, unless severely constrained, will reorient itself to experience only mode I deformation [Broek 85]. For this reason, the three possible deformation modes are not distinguished in this thesis and all stress intensity factors are assumed to be mode I deformation only.

Engineering stress analysis in LEFM provides numeric evaluation for the stress intensity factor for a body of specified shape, containing a crack of given size and shape, subject to various loadings [Anand&Parks 86]. The mathematical solutions used are based on an asymptotic approach where only the first term in a series solution is retained. The elastic stress fields may be characterized accurately enough by the asymptotic fields based on K alone if the distance from the crack tip is kept small compared to other in-plane dimensions such as crack length, remaining ligament, distance from tip to load

application point, etc. For example at 10% of the crack length the K stress is within 7% of the exact value. Under small scale yielding the stress, strain, and displacement fields are well described by K. The two conditions for small scale yielding are 1) the crack tip is close enough so the asymptotic elastic solution approximates the complete series and 2) the crack tip is far enough so the crack tip plastic deformation does not excessively perturb the elastic field. When these conditions are met, there is a region, near the crack tip and surrounding the plastic zone, within which the K stress field gives a good approximation to the actual fields.

4.5.3 Applicability of LEFM for bridges

The conditions necessary for LEFM to be applicable are usually satisfied if gross section stresses at failure are below 0.5 yield stress [Anand&Parks 86]. To demonstrate why this is true, a comparison can be made between the plastic zone size and the critical crack length for a Griffith crack configuration at a stress level of 0.5 yield stress. The plastic zone size is given by:

$$r_p = (K / Y)^2 / (2 \pi)$$

where r_p = plastic zone size

K = stress intensity factor

Y = material flow strength

= yield stress (neglecting strain-hardening)

The Griffith crack case, a through crack in an infinitely wide plate under uniaxial tension as shown in Figure 4-4, has a stress intensity factor of:

$$K = \sigma (\pi a)^{1/2}$$

where σ = uniaxial stress

a = half crack length

Setting $\sigma \leq 0.5 Y$ and combining these equations leads to:

$$r_p / a \leq 1/8$$

So for this crack configuration, limiting the gross section stress to less than half

the yield stress limits the plastic zone size to 13% of the half crack length. LEFM is usually applicable for steel bridges [Fisher 84]. Steel bridge details usually have gross section stresses that are a small enough fraction of yield stress so that crack tip plasticity satisfies SSY and LEFM furnishes an adequate description.

4.5.4 Determining stress intensity factors

The structural severity of a crack is measured by the stress intensity factor, characterizing the intensity of the stress field in a small region surrounding the leading edge of crack. Using LEFM, the value of K may be determined given the applied stress, crack size, and geometric configuration of both the crack and the body in which it is embedded. Three possible approaches to calculation of stress intensity factors are analytic solution, numeric methods, and decomposition, as shown in Figure 4.5. Closed-form analytic solutions are known for highly *idealized* geometries. These solutions are tabulated in handbooks [Tada et al. 73], for fundamental crack geometries that are frequently encountered. However, the complexities of actual structural details defeat a straightforward handbook look-up approach. Direct application of numerical methods, primarily finite element methods (FEM) with specialized crack tip elements, is feasible but there are drawbacks. The FEM approach voraciously consumes both computational resources and the time of engineers well-versed in the art of structural modeling. To cite an extreme example, completely characterizing the *uncracked* stress field for a tie girder required almost 1.5 man-years [Kulicki&Mertz 87]. These costs are accentuated when multiple K values are needed for several crack sizes for fatigue calculations. Such resource intensive solutions are unwarranted for routine bridge design and evaluation.

For general use in bridge engineering, what is needed is a rapid method of calculating K values for opening mode cracks at structural details such as transverse stiffeners, coverplate end welds, flange-to-web junctions of welded girders, and eyebar heads. Such a method can indeed be developed by appropriate combination of analytic solutions and numeric methods, based on decomposition. The decomposition approach starts with the idealized case of a central through crack of length 2a in an infinite plate (see Figure 4.4) and utilizes correction factors to fit the case at hand. The stress intensity factor is expressed as [Albrecht&Yamada 77]:

$$\begin{aligned}
 K &= F_{(a)} \sigma \sqrt{\pi a} \\
 &= F_e * F_s * F_w * F_g * \sigma \sqrt{\pi a}
 \end{aligned}$$

The dimensionless factor $F_{(a)}$ is the correction factor required to fit the idealized through crack in an infinite plate under uniform tension to the actual crack and structural geometry. This factor is decomposed into four correction factors which separately account for the shape of the crack front, the amount of unrestrained crack surface, the width of the cracked element, and the nonuniformity of the applied stress field.

$$F_{(a)} = F_e * F_s * F_w * F_g$$

F_e = elliptical crack shape correction factor

F_s = free surface correction factor

F_w = finite width correction factor

F_g = local stress gradient correction factor

This decomposition of $F_{(a)}$ into F_e , F_s , F_w , and F_g assumes no interaction between flaw shape, free surface, finite width, and opening stress distribution. This is reasonably accurate for cases where the state of stress remote from the crack is predominantly one of uniform extension. Most plate girder cracking fits this restriction. The accuracy of the method is determined by the accuracy of the correction factors. It is desirable to use correction factors which lie on the conservative side, but the methods used to determine the correction factors do not always give conservative results. As an example of a possible unconservative approach, for a first cut approximation when little information is available, the correction factors represented by $F_{(a)}$ may be assumed equal to 1 [Ashby&Jones 82], thus using the Griffith crack configuration as a base case. If good estimates can be made for the four F correction factors, this satisfies the need for a rapid method of calculating K values.

4.5.4.1 F_e : elliptical crack shape correction factor

The effect of the shape of the crack front on the value of K is taken into account by F_e , the elliptical crack shape correction factor. For a two-dimensional crack, the crack front is a point and F_e is always 1. For a three-dimensional crack, the line of the crack front may be curved instead of straight. The closed form solution for K for a crack in an

infinite solid subjected to a remote uniform tension normal to the crack plane assuming an elliptical crack front as shown in Figure 4.6 is [Albrecht&Yamada 77]:

$$K_{\max} = F_e \sigma \sqrt{\pi a}$$

where K_{\max} = K at point where the minor axis intersects the crack front

$$F_e = 1 / E_k$$

E_k = complete elliptical integral of the second kind

$$E_k = \int_0^{\pi/2} [1 - k^2 \sin^2 \theta]^{1/2} d\theta$$

$$k = 1 - (a^2/c^2)$$

a = half minor axis of elliptical crack

c = half major axis of elliptical crack

E_k is the complete elliptical integral of the second kind and depends only on the axis ratio a/c. The following series solution for E_k is valid when k^2 is less than 1 [CRC 72]:

$$E_k = (\pi/2) \{ 1 - (1/2)^2 k^2 - (1*3/2*4)^2 k^4 / 3 - (1*3*5/2*4*6)^2 k^6 / 5 - \dots \}$$

The variation of F_e is shown in the graph of Figure 4.6. The range of F_e for the possible three-dimensional crack front shapes is summarized as:

crack shape	circular	elliptical	tunnel or scratch
a/c	1.0	1.0 > a/c > 0	-> 0
F_e	0.637 = 1 / 2π	0.637 < F_e < 1.0	1.0

Therefore, if F_e was always assumed to be its maximum value of 1.0, at worst, the actual value would be over-estimated by about a third.

4.5.4.2 F_s : free surface correction factor

For a configuration where the crack is fully embedded, like the Griffith crack in Figure 4.4, the opening of the crack tip is restrained by tensile forces developed parallel to the crack at its midline. When the crack is an edge crack these midline stresses cannot develop since the midline is a free surface. F_s , the free surface correction factor accounts

for this effect of a free surface released restraint. For two-dimensional cracks, F_s is 1.0 for through cracks and 1.12 for edge cracks [Albrecht&Yamada 77]. For three-dimensional cracks, F_s is a function of the crack shape a/c . The reason for the dependency of F_s on crack shape is that a strongly curved crack front of a semicircle helps to restrain the crack tip more than the less strongly curve front of a flat semiellipse. F_s may be expressed as a function of a/c [Fisher 84]:

$$F_s = 1.211 - 0.186 \sqrt{a/c}$$

Summarizing F_s for the possible three-dimensional crack shapes:

crack shape	circular	elliptical	tunnel or scratch
a/c	1.0	$1.0 > a/c > 0$	$\rightarrow 0$
F_s	1.025	$1.025 < F_s < 1.211$	1.211

4.5.4.3 F_w : finite width correction factor

The uncracked ligament in a finite width plate provides less restraint for crack opening than a plate with infinite width. F_w , the finite width correction factor, depends on the relative size of the remaining ligament and whether the crack configuration is two [Albrecht&Yamada 77] or three dimensional [Fisher 84].

$$F_w = \sqrt{[\sec(\pi a / W)]} \quad \text{for two dimensional cracks}$$

$$= \sqrt{[\{W / (\pi a)\} \tan(\pi a / W)]} \quad \text{for three dimensional cracks}$$

Due to the traditional definition of 'a' as the *half* crack length, W is the plate width for embedded cracks, but is *twice* the plate width for edge and surface cracks. This results in the ratio a/W ranging from 0 for an infinite plate width to 0.5 for a plate completely severed by the crack. Figure 4.7 graphs F_w for two and three dimensional cases. The expressions for F_w lose accuracy as the remaining ligament size becomes small.

4.5.4.4 F_g : local stress gradient correction factor

F_g , the geometric or local stress gradient correction factor, takes into account the effect of nonuniform opening stresses applied over the crack length. The idealized crack configuration shown in Figure 4.4 is subjected to a uniform tensile stress field over the full width and thickness of the plate. For an isolated crack in a web or flange plate far

from geometric discontinuities, elastic beam theory predicts a nearly constant stress distribution through the thickness of the plate. In actual steel structures, cracks generally occur at geometrical discontinuities such as coverplate ends, gusset plates, stiffeners, and bolt holes. These sudden changes in geometry generate severe local stress gradients which in turn have a significant effect on stress intensity factor at the leading edge of crack. The procedure for finding the value of F_g for an arbitrary structural detail geometry is summarized as follows [Albrecht&Yamada 77]:

- 1) Compute stresses along crack length for the uncracked geometry using FEM or another suitable stress analysis technique
- 2) Insert crack of required length
- 3) Compute K by integrating away normal stresses from step 1, and determine F_g
- 4) Repeat steps 2 and 3 for required crack sizes

Figure 4.8 illustrates this procedure with two bolt holes. This geometric discontinuity generates the nonuniform opening stress field. F_g is computed for the following expression based on pairs of discrete stresses. For a more complete presentation of the calculation of F_g refer to [Albrecht&Yamada 77]:

$$F_g = (2 / \pi) \int_{i=1}^n (\sigma_{bi} / \sigma) \{ \arcsin (b_{i+1} / a) - \arcsin (b_i / a) \}$$

where n = number of element pairs

b_i = distance from crack centerline to start of element i

a = half crack length

σ = normal stress in the member uniformly distributed over the thickness of the plate and computed with strength of material formulas

σ_{bi} = normal stresses in the finite element model of the structural detail

where the crack will be inserted

The stress fields generated at some structural details have been extensively investigated with analytic or experimental studies. Values for F_g can thus be directly determined for typical geometries. A parametric approximation applicable to a number of structural details such as stiffeners, attachments, cover plates, and gusset plates is

[Zettlemyer&Fisher 79]:

$$F_g = SCF / (1 + G\alpha^\beta)$$

where SCF = stress concentration factor, for example, at a weld toe

G and β = dimensionless constants

α = a / t = ratio of crack size to plate thickness

Example solutions for typical structural details are [Zettlemyer&Fisher 77]:

Toe of an end-welded coverplated beam:

$$G = 6.689 \quad \alpha = a / t_f \quad \beta = 0.4348$$

$$SCF = - 3.539 \log (Z / t_f) + 1.981 \log (t_{cp} / t_f) + 5.798$$

Z = weld leg size t_{cp} = coverplate thickness t_f = flange thickness

Transverse stiffener at toe of stiffener weld to flange or web

$$G = 2.776 \quad \alpha = a / t \quad \beta = 0.2487$$

$$SCF = 1.621 \log (Z / t) + 3.963$$

Z = weld leg size t = flange or web thickness

4.5.5 Paris power law describing crack growth

To assess fatigue, it is necessary to know how cracks grow. Fatigue crack growth behavior in steel is shown schematically in Figure 4.9 by a log-log plot of the crack growth rate da/dN , versus the stress intensity range ΔK . The sigmoidal shaped curve can be divided into three regions [Fuchs&Stephens 80]. The Region I portion of the curve covers very low stress intensity ranges. Below a threshold value ΔK_{th} , corresponding to a crack growth rate of approximately 10^{-8} inches per cycle, no crack growth is observed. For stress intensity ranges below the threshold, cracks are thus non-propagating. Region II has a nearly linear log-log relationship between crack growth rate and stress intensity range. This is the region in which most fatigue crack growth behavior falls. Region III has a much steeper slope as the growth rates accelerate as the stress intensity range approaches the material fracture toughness, at which point fracture occurs.

Of the various propagation laws which have been developed to describe fatigue

crack growth, the Paris power law is simple and well known. The Paris power law [Paris&Erdogan 63] relates crack growth rate, da/dN , to the stress intensity range in a cycle, ΔK with the following equation:

$$da/dN = A * \Delta K^m$$

where da/dN = crack extension per cycle of load

a = half crack length

N = number of load cycles

A = proportionality factor, material dependent

m = power factor, material dependent

ΔK = stress intensity range = $K_{max} - K_{min}$ = function of $\Delta\sigma$

$\Delta\sigma$ = stress range = $\sigma_{max} - \sigma_{min}$

The subscripts max and min indicate the maximum or minimum value during one load cycle. The Paris law applies to Region II crack growth, where the material dependent constants describe the linear relationship, m being the slope and A being the y intercept. Experimental data to determine values for the material dependent constants for mild (ferrite-pearlite) steels show significant scatter. The upper boundary of the scatter band may be used to give conservative estimates of crack growth rates. Using units of kips and inches these upper bound values applicable for the low carbon steels most commonly used in bridge construction are [Barsom&Rolfe 87]:

$$A = 3.6 * 10^{-10}$$

$$m = 3$$

For a period of crack growth *small enough that the crack geometry does not change appreciably*, the Paris crack growth law can be directly integrated to yield an explicit expression for the number of load cycles spent growing a crack from an initial to a final length [Ashby&Jones 82]:

$$N_f = \int_{a_f}^{a_i} [da / (A * \Delta K^m)]$$

where:

N_f = cycles to failure

a_i = initial crack size

a_f = final crack size

$$\Delta K = F * \Delta\sigma \sqrt{(\pi a)}$$

F = correction factor (assumed constant with respect to a small change in crack length)

integrating:

$$\begin{aligned} N_f &= \int_{a_f}^{a_i} [da / (A * \Delta K^m)] = \int_{a_f}^{a_i} [A^{-1} F^{-3} \Delta\sigma^{-3} \pi^{-3/2}] a^{-3/2} da \\ &= [A^{-1} F^{-3} \Delta\sigma^{-3} \pi^{-3/2}] * [-2 a^{-1/2}]_{a_i}^{a_f} \\ &= [2 A^{-1} F^{-3} \Delta\sigma^{-3} \pi^{-3/2}] * [a_i^{-1/2} - a_f^{-1/2}] \end{aligned}$$

This relationship only applies for an increment of crack growth where the crack geometry does not change significantly. This allows F, which is in general a function of crack length, to be assumed to be constant for the small interval of crack growth.

4.6 Fracture control plan for steel highway bridges

Given an identification of the physical parameters governing fatigue and fracture and the quantitative concepts of LEFM, a rational fracture-control plan can be composed for steel highway bridges. The four elements of a fracture-control plan are [Barsom&Rolfe 87]:

1. Identify the factors that may contribute to the fracture of a structural member or the failure of an entire structure. This includes a description of: probable service conditions such as minimum temperature; loadings and load rates; possible range of quality of fabrication; and inherent notch toughness of candidate materials.
2. Establish the relative contribution of initial crack size, stress history, material properties, and structural configuration to a possible fracture in a member or to the failure of the structure.

3. Determine the relative efficiency and tradeoffs of various design methods to minimize the possibility of either fracture in a member or failure of the structure. The basic options are: decrease design stress; minimize initial discontinuities; and use materials with improved notch toughness.
4. Recommend specific design considerations to ensure the safety and reliability of the structure against fracture, including: material performance and selection; design stress levels; fabrication; and inspection.

Before about 1940, formalized fracture-control plans did not exist. Catastrophic structural fractures were rare because large metal structures were built up of many riveted plates. A crack would only propagate through one plate and then arrest at the riveted joint [Barsom&Rolfe 87]. Ship building during the Second World War saw a major innovation in the way large structures were built [Chiles 88]. The Liberty Ship design was nearly entirely welded, creating a monolithic structure that provided a path for crack growth across many plates, leading to failure of the entire hull. An investigation of hull fractures in the Liberty Ships [USNavy 47] identified certain structural details, primarily the square cutouts at top of shear strake, as the initiation location for most cracking failures. Design improvements to minimize stress concentrations at these initiating details, addition of crack arrestors, and improved notch toughness of materials achieved satisfactory reliability for welded ships. By the early 1950's, ship design specified steels with dynamic load NDT temperatures below their minimum service temperature.

The evolution of fracture-control plans for bridge design has analogies with that for ship design. During the 50's and 60's highway bridges made a transition from riveted and bolted construction to monolithic welded structures, providing a path for cracks to propagate. Like the Liberty ships, a catastrophic and highly publicized fracture failure had a major effect on bridge design and inspection in the US. The collapse of the Point Pleasant Bridge in 1967 was the event which led to a comprehensive inspection program for highway bridges as well as influencing changes in design codes directed at prevention of catastrophic fractures in steel bridges. (This collapse was *not* a weld failure.)

There is a strong empirical component to the codes applying to design of bridges. Specifications are based on extensive service experience for material, design, and fabrication. In addition, substantial research programs have addressed fatigue and fracture in bridges. Coordinated efforts under the National Cooperative Highway Research Program introduce research results into practice. A complete fracture-control plan addresses ma-

terials, design, fabrication (welding), construction, inspection, and maintenance. The current fracture control plan for steel highway bridges is distributed among various documents promulgated by the American Association of State Highway and Transportation Officials. The primary documents are the “Standard Specifications for Highway Bridges” [AASHTO 85] and the “Guide Specifications for Fracture Critical Non-redundant Steel Bridge Members” [AASHTO 86]. In addition, the “Standard Specifications for Transportation Materials and Methods of Sampling and Testing”, “Standard Specifications for Welding of Structural Steel Highway Bridges”, and “Manual for Maintenance Inspection of Bridges” address pertinent issues.

The major parameters of stress history and structural configuration are addressed in the “Standard Specifications for Highway Bridges” [AASHTO 85] which uses a reduced design stress approach to ensure that the fatigue life of structural details is longer than the design life. Structural details are assigned to one of seven fatigue categories based on written guidelines (see Figure 4.10 [AASHTO 85]) and illustrated examples (see Figure 4.11 [AASHTO 85]). These categories are summarized as:

- A plain plate and rolled beams
- B plain welds and welded beams and plate girders
- C stiffeners and short attachments (< 2” long)
- D medium attachments (2” to 4” long)
- E coverplates and long attachments (> 4” long)
- E’ thick flanges and thick coverplates (> 0.8” thick)
- F shear stress on throat of fillet welds

Due to the low weld stresses under normal design conditions, Category F seldom controls [NCHRP206 79]. Each of the remaining six fatigue categories correspond to an empirical curve graphing stress range versus number of cycles to failure. These S-N design curves, shown in Figure 4.12 [NCHRP286 86], are based on the 95% confidence limit for 95% survival data for laboratory test specimens judged to be similar to the detail at hand. These fatigue category design curves are thus used to limit the design stress level to the upper boundary of the experimental scatter band to sustain the number of load cycles for the required design life. The classification of structural details by stress categories in current fatigue design specifications amounts to a classification by severity of local stress

gradients [Albrecht&Yamada 77].

The major parameters of initial crack size and material properties are largely addressed in the “Guide Specifications for Fracture Critical Non-redundant Steel Bridge Members” [AASHTO 86]. This Guide lays out a fracture-control plan for fracture critical members, defined to be those members or member components in tension whose failure would be expected to result in collapse of the bridge. The purpose of this Guide is to provide for additional quality of material and increased care in fabrication to lessen the probability of fracture of critical tension components from crack formation and extension. The goal underlying the material toughness requirements is to prevent plane-strain fracture behavior by having the steel NDT temperature (see Section 4.8.3) be well below the lowest probable service temperature at the maximum in-service loading rate. Since direct testing of K_{Ic} would be prohibitively expensive, Charpy V Notch (CVN) impact tests are used to ensure an adequate level of fracture toughness. The effect of strain rate on the fracture toughness is taken into account as discussed in Section 4.8.3. The requirement for a minimum CVN impact of 15 ft-lb @ 40°F (for primary tension members with yield strength less than 50 ksi) implies a minimum fracture toughness of 50 ksi $\sqrt{\text{in}}$ at an operating temperature of -30°F and strain rate of 10^{-3} sec^{-1} characteristic of bridge load rates. To account for scatter in test results and at increased toughness for fracture critical members, the minimum CVN impact requirement is 25 ft-lb. CVN values are specified in a comparable manner for other service temperatures and yield strength steels.

In addition to specifying minimum acceptable levels of material resistance to cracking, this Guide also addresses welding. Initial crack size is determined by the quality of the fabrication, particularly welding. Specific qualification and certification requirements are laid out for fabricators, welding inspectors, and nondestructive testing personnel. Welding requirements and procedures are intended to insure that: the properties of the weld material are adequate, the possibility of hydrogen cracking is minimized, and initial flaw sizes introduced by such discontinuities as porosity and slag inclusions are within bounds accepted as good welding practice.

There are drawbacks to the current AASHTO approach for addressing fatigue and fracture in bridge design. The presentation of a fracture-control plan is fragmented. The fatigue category approach does not cover important and common classes of cracking. The basis on S-N (applied stress versus endurance) data results in a lack of flexibility to address significant variables other than applied stress and general geometry. The relation-

ship between specified requirements and the underlying fracture mechanics concepts is obscured. Each if these shortcomings are discussed below.

The presentation of the fracture-control plan is diffused over a variety of documents. The “Guide Specifications for Fracture Critical Non-redundant Steel Bridge Members”’s explicit presentation of some elements of a fracture control plan for the important class of fracture critical members is a step in the right direction. Explicit statement of a complete fracture-control plan is needed to ensure good fatigue performance of all bridge details.

The fatigue category approach lacks completeness. Not all types of cracking are covered. In particular, distortion induced cracking which is the most frequently occurring class of bridge cracking, is not addressed. The most recent reassessment of the AASHTO fatigue categories [NCHRP286 86] reviews the entire body of welded bridge detail fatigue test data and recommends slightly revised fatigue design curves that better estimate fatigue resistance for the covered classes of cracking. These recommended changes leave the basic framework of fatigue categories unchanged. Under this system, it is essential to: correctly identify the most severe detail, assign the detail to the appropriate stress category, and design the component for the resulting stress category, while also accounting for the other elements of the implicit fracture control plan dispersed among various specifications.

Welded structures inevitably contain flaws. The quality control philosophy for weld acceptance is based on “fitness for purpose”. Consistent application of this philosophy requires reliable methods of assessing the significance of flaws in the context of fatigue, capable of taking into account variations in stress concentration, initial flaw dimensions, and critical crack size. Fracture mechanics concepts provide greater flexibility than methods based on simple S-N data because significant variables other than applied stress and general geometry can be taken into consideration [Maddox 74].

An approach that does not make the underlying fracture mechanics concepts apparent to the bridge design engineer is unsatisfying. One result of the inherent lack of transparency is the misidentification of stress category. Since the selection of stress category is based on the written and pictorial information presented in Figures 4.10 and 4.11, the front-line engineer may match the detail at hand to a different category that appears pictorially similar but which is not similar based on the hidden fracture mechanics con-

cepts. Such obscurity encourages unequal reliability of various components of a single bridge and of various bridges within the same highway system. This unevenness of reliability is clearly undesirable. Another result of the indirect and diffuse specification of a fracture-control plan is that the procedures developed for design purposes are not usable for inspection and failure investigation purposes. One consequence is that when cracks are encountered during fabrication and construction, inappropriate repairs may be called for which do not correct or may even exacerbate the underlying problem [Tubby&Wylde 86].

This obscurity is not a necessary characteristic of a usable fatigue design code. For example, fatigue has been addressed explicitly and clearly as a major design issue for airplanes for many years [USAirForce 74] [deJonge 76]. Aerospace engineering is a much more sophisticated field than bridge engineering, with very different economic and technical imperatives. However, experience with aircraft could be used to provide direction for more clearly incorporating fracture mechanics principles in codes and specifications and allowing for a more direct accounting of the parameters governing cracking behavior in bridge design[Rudd et al. 82], inspection[Goranson et al. 82], and maintenance[Denyer 82]. The drawbacks of the current AASHTO approach, and the clear desirability of a more transparent approach, are the motivations underlying the development of the multilevel knowledge-based system, CRACK.

4.7 Taxonomy of cracking causes for steel highway bridges

Different kinds of bridges and bridge details are susceptible to cracking from different causes. There are two major classes of bridge fatigue damage: load-induced and distortion-induced. The cyclic stresses driving load-induced fatigue are due to the primary load bearing behavior of the bridge under moving loads. Distortion-induced fatigue is driven by relatively small out-of-plane displacements caused by the mutual presence of an abrupt change in stiffness and a periodic force opposing it [Kulicki&Mertz 87]. Bridge design specifications [AASHTO 85] [ECCS 85] address load-induced fatigue by classifying detail types according to expected cracking behavior. These classifications are used to specify the design provisions intended to prevent crack induced failure. Distortion-induced cracking is not addressed by these design specifications.

To address fatigue and fracture problems, it is necessary to identify the most likely type of cracking which may occur in a particular steel bridge. The task of pinpoint-

ing the feature most likely to cause cracking problems may be viewed as a classification type of problem. A survey [Fisher&Yuceoglu 81] of 142 bridges that had developed cracking gathered information on 149 instances of primary cracking causes (several sites developed more than one type of cracking in different structural details). These 149 cases can be grouped into 28 categories of cracking [Fisher 84]. These categories can be organized into the taxonomy of most prevalent causes of cracking in steel bridges presented in Figures 4-13, 4-14, and 4-15. The ordering of classes within the taxonomy is with the most common problem listed first.

Figure 4.13 shows the upper levels of this taxonomy with the number of cases occurring in each class. Figure 4.14 shows a more detailed picture of the most frequently observed general class of cracking, cracks due to secondary (distortion-induced) stresses. The subclass of out-of-plane distortion in a small gap, usually a segment of a girder web, is the most prevalent cause of cracking. Web-gap cracking is illustrated in Section 4.8.4. Figure 4.15 elaborates the next largest class, initial flaws and low fatigue resistant details. The largest subclass is lack of fusion weld flaws. These flaws most frequently occurred in groove welds for components considered to be secondary components (gussets, longitudinal stiffeners, etc.) with no weld quality and inspection requirements, although some developed in transverse groove welds made in primary members (flange, web, cover-plates) prior to the use of NDT methods for detecting poor quality welds [Fisher&Mertz 84].

To enable construction of a reasonably complete kernel system, the size of the problem domain must be reduced to a manageable level. Care must be taken in selecting the focused subdomain to be sure that generality is not sacrificed. CRACK is restricted to the problem domain of beam and plate girder welded details. The most common type of structural steel highway bridge is the beam or girder bridge. Welded joints experience cracking problems much more often than riveted and bolted components. Figure 4.16 shows the entire taxonomy pruned to represent the sub-problem of cracking in welded steel plate girder bridges. Welded steel plate girder bridges experience an interesting variety of cracking problems. This subset of the entire problem area of cracking in steel highway bridges is the selected domain for the research prototype version of CRACK. Knowledge about the four major parameters governing cracking behavior pertinent to this domain is summarized in the following section.

4.8 Cracking in welded steel plate girder highway bridges

4.8.1 Initial crack size, shape, and orientation

For most welded details, the number of cycles for crack initiation is small. Experimental results indicate that initiation may be neglected and fatigue life can be based on propagation alone [NCHRP286 86]. Even properly fabricated welds are not flaw free. Welding specifications recognize the possible existence of tolerable imperfections that do not adversely affect the structure's performance for its intended use. The definition of good welding practice thus allows for the presence of imperfections, but strives to limit the size of the inherent flaws. The provisions of the American Welding Society Structural Welding Code determine the acceptability of structural welds [AWS 86]. A variety of nondestructive evaluation techniques are used to control the level of workmanship. Test methods include visual, dye penetrant, magnetic particle, radiographic, and ultrasonic [NCHRP242 81]. Figure 4.17 [Barsom&Rolfe 87] illustrates some of the types of initial flaws which occur in welded joints.

These flaws become initiation sites, primarily at weld toes and terminations, where the presence of weld imperfections coincides with regions of high stress concentrations due to geometrical discontinuity. The probable size and frequency of flaws differs for weld configuration and process. For example, discontinuities are more likely at fillet weld roots than at groove welds. This is in *accord* with the better fatigue resistance of groove-welded details compared to fillet-welded flange gussets [NCHRP227 80].

Surface or internal flaws may be origins of fatigue cracks. Surface flaws are the most frequent initiation sites since the stress intensity factor for the same crack size is larger for a surface crack than for an embedded crack. In addition, the regions of highest stress concentration at weld toes and terminations are at the surface. These surface flaws are usually due to slag intrusion. Internal imperfections from porosity, lack of fusion, or trapped slag may also serve as origins of fatigue cracks. The maximum initial size for these flaws, assuming good welding practice, may be taken as [Barsom&Rolfe 87]:

0.016" for surface flaw (weld toes)

0.08" for embedded flaws (gas pockets)

The welding process alters the type and significance of weld flaws. For example, electroslag welds are particularly susceptible to centerline cracking [NCHRP201 79]. Electroslag welds were used during the 1960's and 1970's for splicing heavy steel girders in a single pass. This process, normally used at the steel fabrication shop, requires careful

control of amperage and voltage weld parameters. After failure of the Interstate 79 bridge over the Ohio River near Pittsburgh due to a 2 inch wide, full depth crack in an 11 foot deep girder, electroslag welds were judged unsuitable for use in tension members and hundreds of bridges were repaired beginning in 1977 by bolting plates across the original weld splices .

In addition to initial crack size, initial crack shape is needed to characterize three dimensional cracks. Crack shape for three dimension elliptical cracks may be described using the minor to major axis semidiameter ratio, a/c . Empirical relationships for lower bound a/c ratios, based on experimental observation for different structural detail geometries, are [Fisher 84]:

coverplates	$c = 5.457 a^{1.133}$ (inches)
stiffeners	$c = 1.197 a^{0.951}$ (inches)

The equation for coverplates is influenced by the tendency of cracks growing at weld toes to form at numerous sites along the transverse width perpendicular to the applies stresses. These cracks start to coalesce and form a common crack front very early in the crack growth process, leading a relatively long and shallow crack. Cracks at stiffeners tend to be more semicircular. Cracks at internal flaws are frequently roughly circular. Fro flaws which are elliptical, the shape remains relatively constant as the crack grows.

The orientation of a flaw influences its role in fatigue. Flaws which are parallel to the direction of stress will not originate fatigue cracks. For this reason, cracks such as the lack of fusion flaw between web and flange shown in Figure 6.7 do not influence fatigue behavior. Primary load-induced fatigue cracks grow perpendicular to the two-dimensional calculated stresses, displacement-induced cracks generally grow parallel to this stress field.

4.8.2 Stress history

The stress range, $\Delta\sigma$, is the parameter driving fatigue. Stress fluctuations for bridges are due to randomly variable loadings. Dealing with actual service loadings that have complexly varying stress amplitudes would be greatly simplified if the variable amplitude loading could be related to an equivalent constant amplitude loading. The need to deal with variable stress ranges in real load histories is most commonly addressed by assuming a linear accumulation of damage as proposed by Palmgren and Miner

[Fuchs&Stephens 80]. This empirical method uses a linear summation of the damage accumulated for cycles in each stress range to estimate the total number of cycles to failure for the variable stress range case. The failure criterion, known as Miner's Rule, is given by setting the sum of the ratios of the number of cycles at the i^{th} stress range, n_i , to the number of cycles to failure at stress range i , N_i , equal to unity: $\{ n_i / N_i \} = 1$. This linear cumulative damage rule, combined with the use of the Paris power law for crack growth rate ($da/dN = A F^m \Delta\sigma^m \pi^{m/2} a^{m/2}$) leads to the following expression for a constant amplitude effective stress range for $m = 3$ [Fisher 84]:

$$\Delta\sigma_{\text{eff}} = [\alpha_i (\Delta\sigma_i)^3]^{1/3}$$

where α_i = frequency of occurrence of cyclic stress at stress range i
 $= n_i / N_T$

$\Delta\sigma_i$ = stress range i

N_T = total number of cycles at all stress range levels

Experimental results [NCHRP188 78; NCHRP267 83] support the use of this effective stress range for predicting the fatigue behavior of welded steel bridge details. This method of determining $\Delta\sigma_{\text{eff}}$ requires frequencies for various ranges, which are not always available. When no better data is at hand, the nationwide gross vehicle distribution [Galambos 79] may be used in concert with the design stress range to estimate effective stress range [Fisher 84]:

$$\Delta\sigma_{\text{eff}} = [0.35 \alpha^3]^{1/3} \Delta\sigma_D$$

where α = assumed between 0.6 to 0.7

$\Delta\sigma_D$ = design stress range

The constant amplitude effective stress range is thus between 42 and 49 percent of the design stress range, reflecting the fact that most inservice stress cycles fall well below the design stress range.

For some materials and applications, crack growth rates are dependent on the

stress ratio, $R = P_{\min} / P_{\max} = \sigma_{\min} / \sigma_{\max} = K_{\min} / K_{\max}$ [Barsom&Rolfe 87]. The Paris power law does not account for any effect of R on crack growth rates. This is acceptable for this application, since extensive experimental work has shown that the stress ratio is not a significant load parameter for the fatigue life of welded details in steel beams and girders [NCHRP102 70].

High residual stresses are created by welding. In the vicinity of the weld, cooling and joint restraint usually result in residual stresses at or near the yield point [NCHRP102 70]. Cracking of welded details originates and spends the bulk of the fatigue life within this zone of high residual stress. The full cyclic range is effective for driving crack growth since even a cycle that varies nominally for compression to tension will be felt as a tension-tension cycle within the residual stress field. This is the reason stress ratio is not a significant parameter for fatigue growth at welded details [NCHRP286 86]. The stress level for determining K is usually the yield stress at welded details. The total sum of the residual, dead, and live stresses is the relevant stress level for evaluating the stability of the crack to resist fracture.

The rise time for cyclic live loads is about 1 second for highway bridges [Cudney 68]. This loading rate is intermediate between static and dynamic rates. A conservative measure of the maximum strain rate for bridge loading may be taken as 10^{-3} sec^{-1} [Barsom&Rolfe 87].

Distortion induced cracking is caused by repeated displacements, not traffic induced cyclic stresses. The stresses in a short gap can be approximated from the magnitude of the driving distortions by the relationship [Fisher 84]:

$$M = (4 E I R / L) + (6 E I \Delta / L)$$

- where E = modulus of elasticity
 = 29,000,000 psi for steel
 I = $t^3 / 12$
 t = plate thickness
 L = length of gap
 R = relative rotation of the flange relative to the web
 Δ = out-of-plane movement of the web

For in-service conditions, the second term dominates. For shipping conditions, the first term dominates. These distortion induced stresses may be very high, for example 40

ksi for a 1" web gap at a floorbeam connection [Fisher et al. 87].

4.8.3 Material properties

The fracture toughness of a material, K_{Ic} , is defined as the experimentally determined critical value of K large enough to cause the onset of rapid fracturing. K_{Ic} is a function not only of the material type but also the temperature, strain rate, and specimen thickness.

The resistance of a material to fracture is determined by the micromechanisms of crack propagation [Ashby&Jones 82]. When the mechanism of cracking is ductile tearing, large amounts of energy are absorbed in the process of creating new crack length, resulting in high toughness. A ductile material flows readily (its dislocations are mobile), allowing large plastic deformations at the crack tip, leading to tough behavior since a lot of energy is absorbed by the plastic flow. Metals which exhibit this ductility have low yield strength relative to fracture toughness, allowing development of a large plastic zone and hence a large amount of plastic deformation. The plastic flow around inclusions nucleates voids in plastic zone, blunting the crack and lowering the stress at the tip. When this stress is just sufficient to continue plastic deformation of the work-hardened material at the crack tip, ductile tearing occurs. In contrast to ductile tearing, when the micromechanism of crack propagation is cleavage, little energy is absorbed by plastic flow and the behavior is brittle. Metals which cleave have high yield strength relative to fracture toughness, hence a small plastic zone with a small amount of plastic deformation, little crack blunting, and high crack tip stress. When this stress is greater than the interatomic bond strength, fracture by cleavage results.

In metals with a body-center cubic crystal lattice arrangement, such as ferrite in mild steel, dislocation motion is assisted by thermal agitation of the atoms. At lower temperatures this agitation is less and the dislocation motion is more difficult so the yield strength rises. The plastic zone at crack tip thus shrinks, causing a change from ductile tearing to cleavage as the cracking mechanism. The temperature below which the brittle failure occurs is the nil ductility transition (NDT) temperature. In steels, the ductile-to-brittle transition can be as high as 0°C. This is why bridges are much more likely to fail in winter than in summer [Ashby&Jones 82]. Figure 4.18 [Barsom&Rolfe 87] shows the variation in energy absorption (and corresponding fracture toughness) with temperature.

For bridges, particular attention should be paid to the effect of loading rate. Common fracture toughness tests have a dynamic (1 msec) rise time while the rise times of bridge loadings are in the intermediate (1 sec) or even, for small ratios of live to dead load, in the static (1 min) ranges [Fisher 84]. Fracture toughness decreases with increasing load rate, as shown in Figure 4.18. The yield stress is higher for dynamic loads, leading to a smaller plastic zone size and reduced energy absorption. The static fracture toughness for a material may be estimated by shifting the dynamic fracture toughness curve by a temperature decrease of [Fisher 84]:

$$T = 215 - (3/2) Y$$

where: T = temperature shift, °F
 Y = yield stress, ksi

Transformation of test results from an impact strain rate of sec^{-1} to an intermediate strain rate of 10^{-3} sec^{-1} characteristic of bridge load rates, may be estimated by decreasing by 3/4 of the total temperature shift from dynamic to static conditions [Fisher 84]. Due to the difficulty and expense of direct testing for dynamic K, the normal means of determining toughness for bridge steel is by using Charpy V Notch (CVN) impact data. A well established equation for estimating K_{Id} values for CVN impact data is [Barsom&Rolfe 87]:

$$(K_{Id})^2 / E = 5 * (CVN)$$

where: K_{Id} = fracture toughness,
 dynamic mode I loading, $\text{psi}\sqrt{\text{in}}$
 E = modulus of elasticity
 = 29,000,000 psi for steel
 CVN = Charpy V-notch impact energy, ft-lbs

Specimen thickness influences fracture toughness by affecting the degree of triaxial constraint at the crack tip. The effect of specimen thickness on fracture toughness is illustrated in Figure 4.19 [Fuchs&Stephens 80]. Thin specimens have less triaxial constraint at the crack tip and appreciable “shear lips” occur as shown in Figure 4.19. The crack tip stress conditions approach plane stress, with nonzero strains in the through plate direction. Thinner plates thus have larger plastic zone sizes at fracture and higher fracture toughnesses. Thick specimens have greater triaxial constraint at the crack tip and flat fracture faces occur as shown in Figure 4.19. The crack tip stress conditions approach

plain strain, with nonzero stresses in the through plate direction. The plain strain fracture toughness, K_{Ic} , is a true material property because it is independent of thickness.

Crack growth behavior, like toughness, is strongly dependent on metallurgical, environmental, and mechanical parameters. However, for commonly used bridge steels, in the parametric range of interest, fatigue life as a function of stress range and crack growth rate as a function of stress intensity range have been found to be essentially independent of yield strength, temperature, and cyclic frequency [Fisher 84].

4.8.4 Structural properties

The fatigue life of a structure is governed by localized behavior at the detail with the shortest life for the applied loads. A severe geometric discontinuity subjected to a low stress range may outperform a more moderate geometry with high stress fluctuations. The level of detail to focus on is the cross section, with the configuration of the girder and attachments fully described. A plate girder is built up of component plates that are welded, bolted, or riveted together. If a library of possible components is available, the configuration of the girder is easily specified by identifying the parts and the connections. Figure 4.20 shows some of the standard parts of a plate girder. Once the cross section of a plate girder is specified, the taxonomy of crack causes shown in Figure 4.16 may be consulted to flag structural features known to be susceptible to cracking.

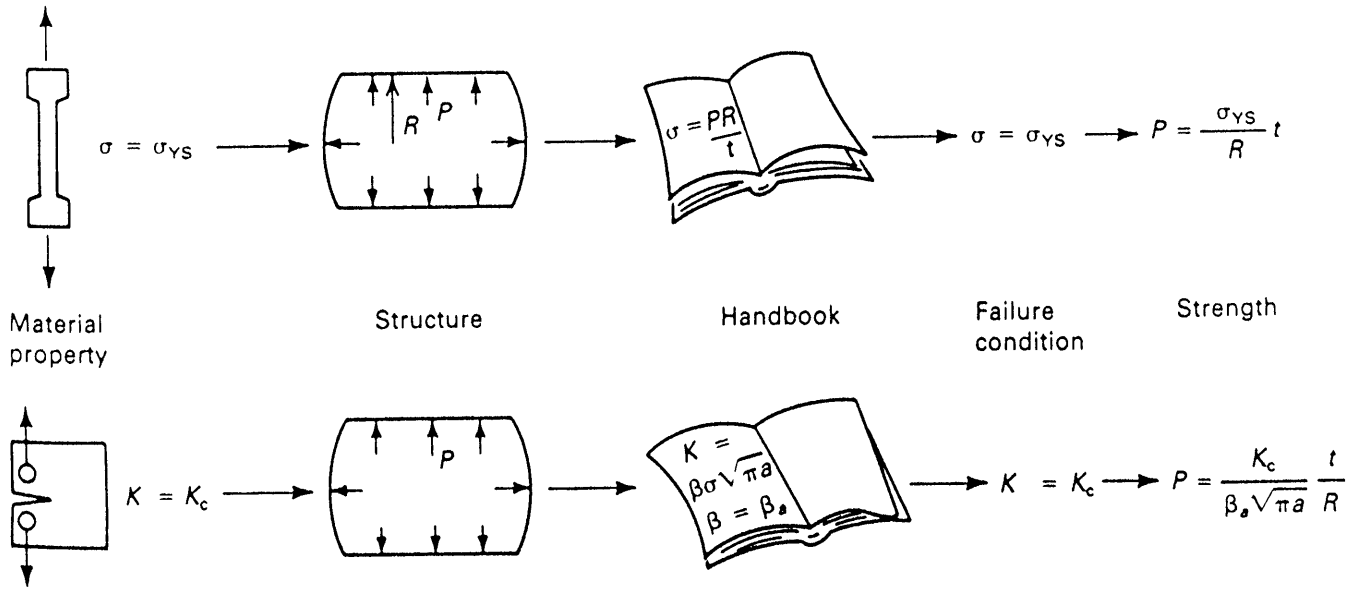
Figure 4.21 shows various cross sections that can be examined for such features. Sections A and B are two examples of the many types of details susceptible to cracking from initial weld flaws. The most common (about 80%) crack origin for Section A, a plain welded girder without attachments, is porosity due to gas entrapment in the continuous fillet welded flange-web connection. The next most frequent type of initiation site for plain girders is another kind of discontinuity in the flange-web weld, such as weld repair, start-stop position, or one of the flaws shown in Figure 4.17. A notch in the flame-cut edge of the flange tip is an occasional origin [Hirt & Fisher 73]. When stiffeners, coverplates, or other attachments are present, the continuous flange-web fillet weld does not act as an initiation source, since the details at the attachments are more critical. Crack initiation and propagation for rolled beams without attachments is similar to welded beams without attachments, but the size of the initial flaw is much smaller and hence the fatigue life is much longer for rolled beams [Hirt et al. 71]. Section B is a coverplated girder. If the coverplate is end-welded, the most likely initiation site is near mid-flange at the toe of the end weld. If the coverplate is not end-welded, the most likely initiation site is at the

termination of the longitudinal coverplate weld [NCHRP102 70].

Section C is susceptible to distortion induced cracking at the small web gap at the top of the floor beam connection plate. As the floorbeam carries traffic loads, the end of the beam rotates, forcing the deflection of the girder web out of its normal longitudinal plane. There is a small (3/4" to 1") gap between the top of the floorbeam connection plate and the top girder flange. This web gap causes an abrupt change in stiffness, concentrating the rotation induced distortion within a short length of the girder web. Repeated pumping of this short gap leads to longitudinal cracking of the web at the top of the connection plate and at the top flange to web weld. This type of web gap cracking at floor beam or floor beam truss connection plates is so prevalent that a survey in one state revealed cracks in half of the bridges with this detail [Fisher 81]. Diaphragm and cross-beam connections are frequently the sites of similar web-gaps, but have less severe imposed rotations than the floorbeam case illustrated. This kind of web-gap detail usually where the top flange is in tension and arises for the ironic reason that the connection plate was not welded to the flange to eliminate an initiation site for load-induced fatigue of the tension flange [Fisher&Mertz 85]. To preclude distortion induced cracking, the connection plate must be either positively attached to the flange or the gap length must be at least 12" long [Fisher et al. 87].

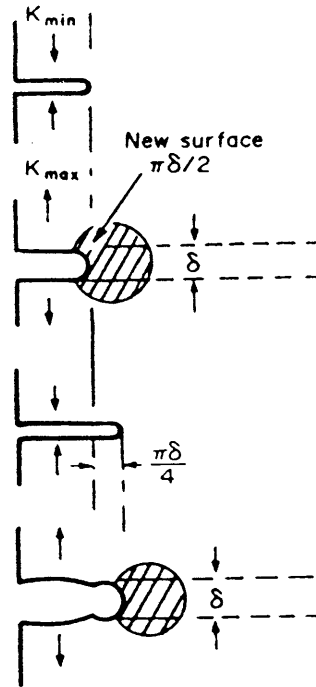
Once the structural features most likely to lead to cracking are identified, the effect of the geometric discontinuities on the stress field may be accounted for using the decomposition method described in Section 4.5.4.4.

Fracture analysis sequence



-72-

Figure 4-1:
Fracture analysis sequence: analogy to yielding
[Broek 85]



cyclic subcritical crack extension

Figure 4-2:
Fatigue crack growth mechanism [Ashby&Jones 82]

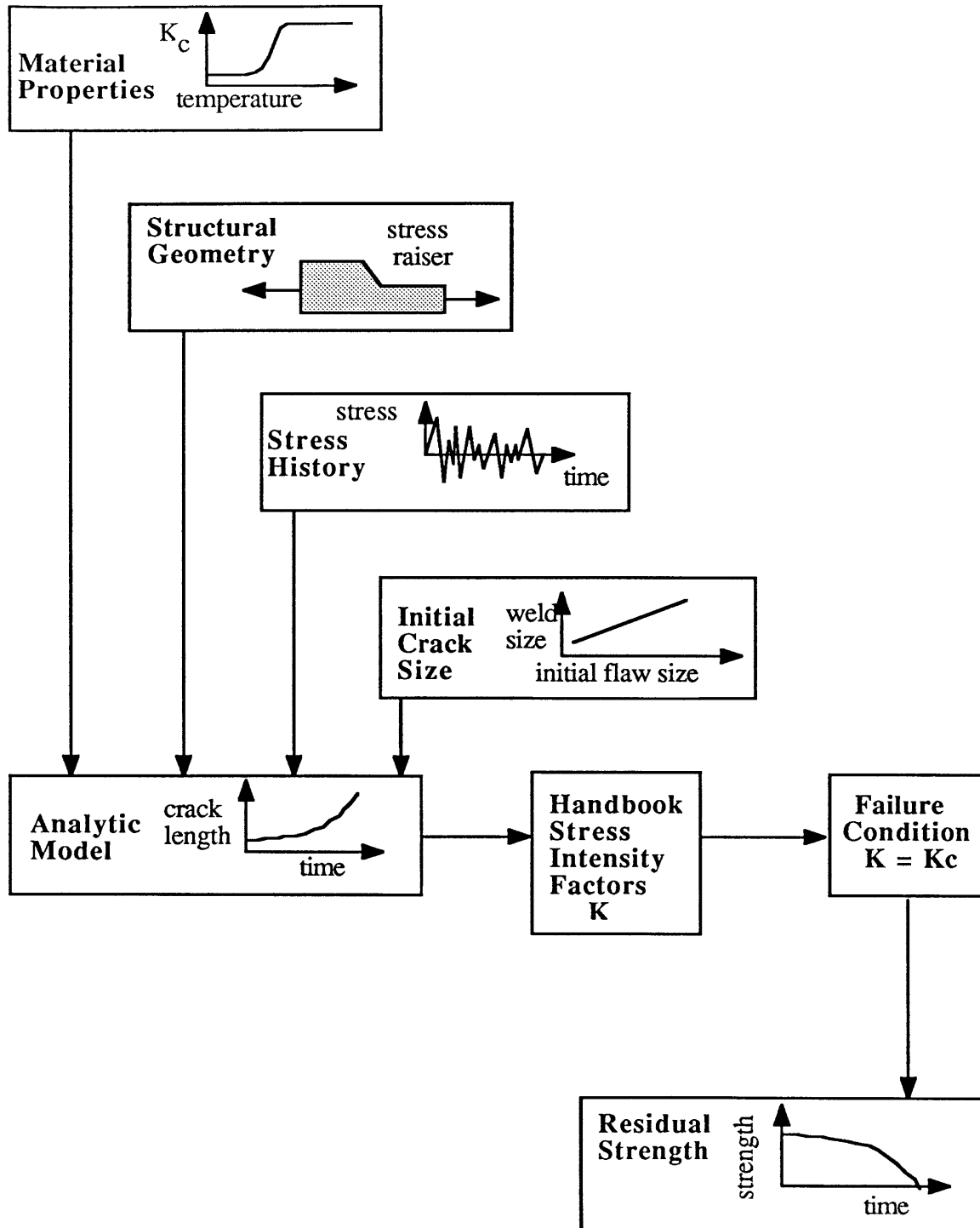


Figure 4-3:
Parameters governing crack behavior

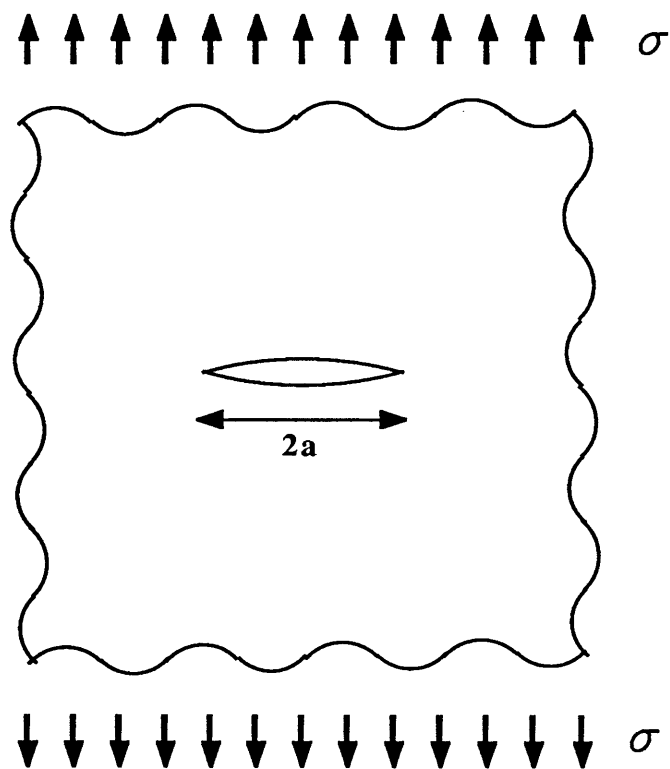


Figure 4-4:
Griffith crack configuration

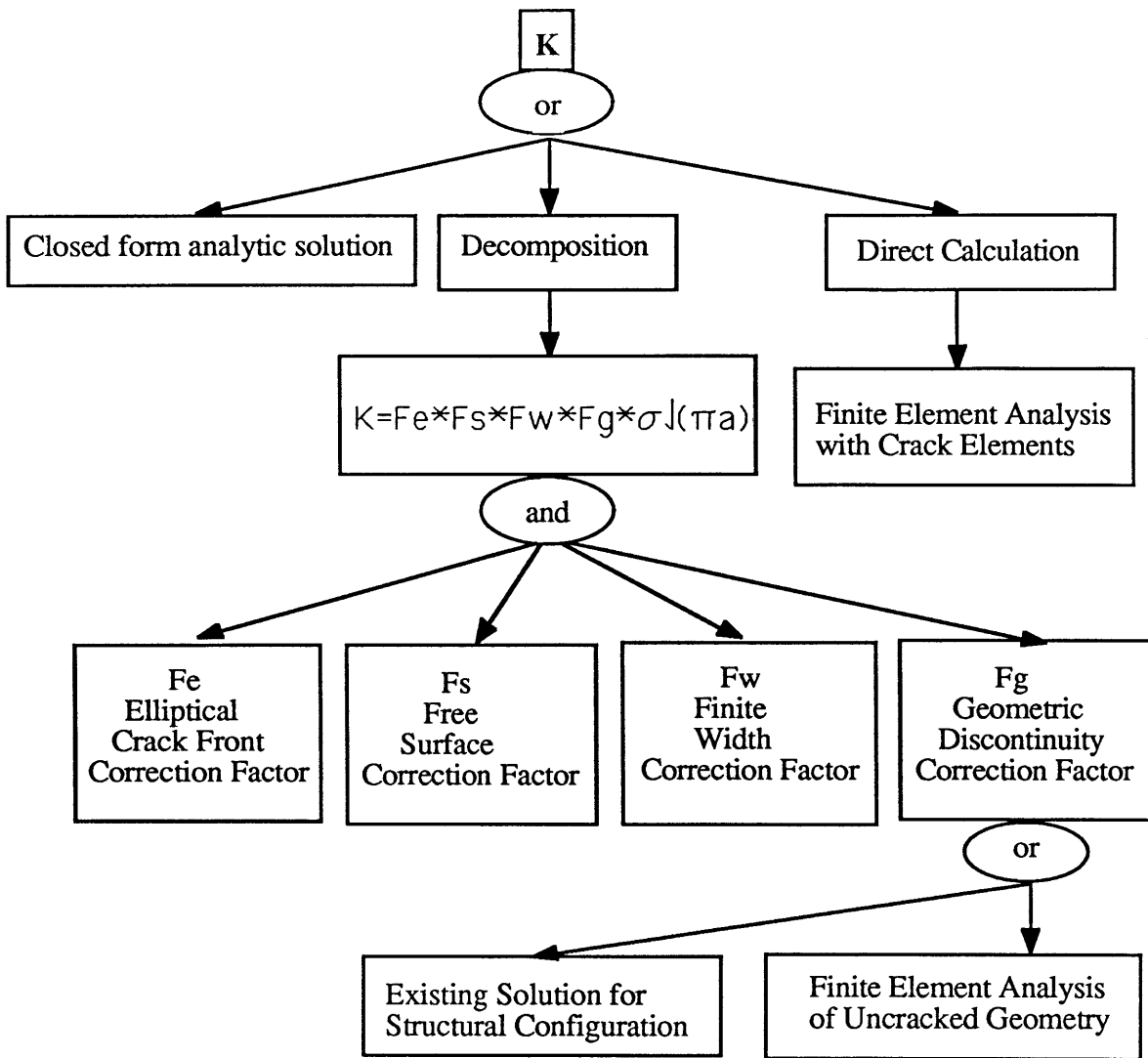


Figure 4-5:
Stress intensity factor K computation

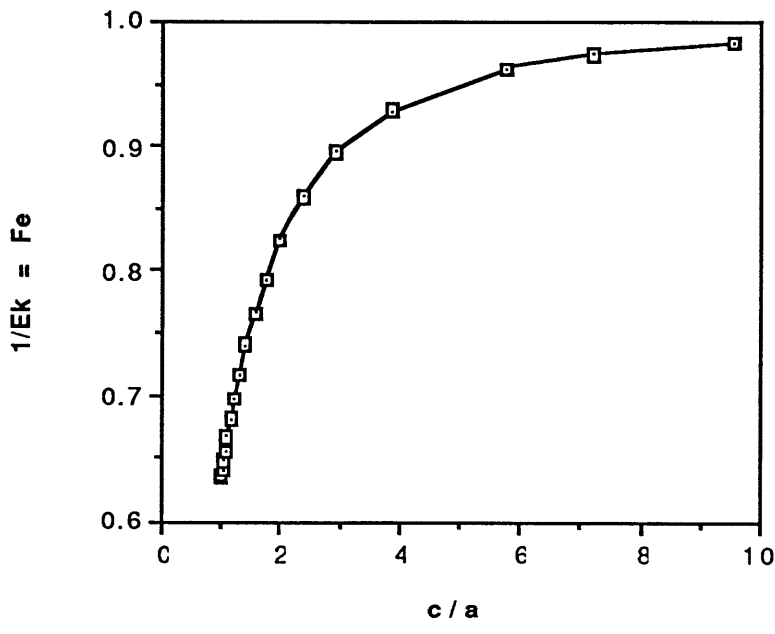
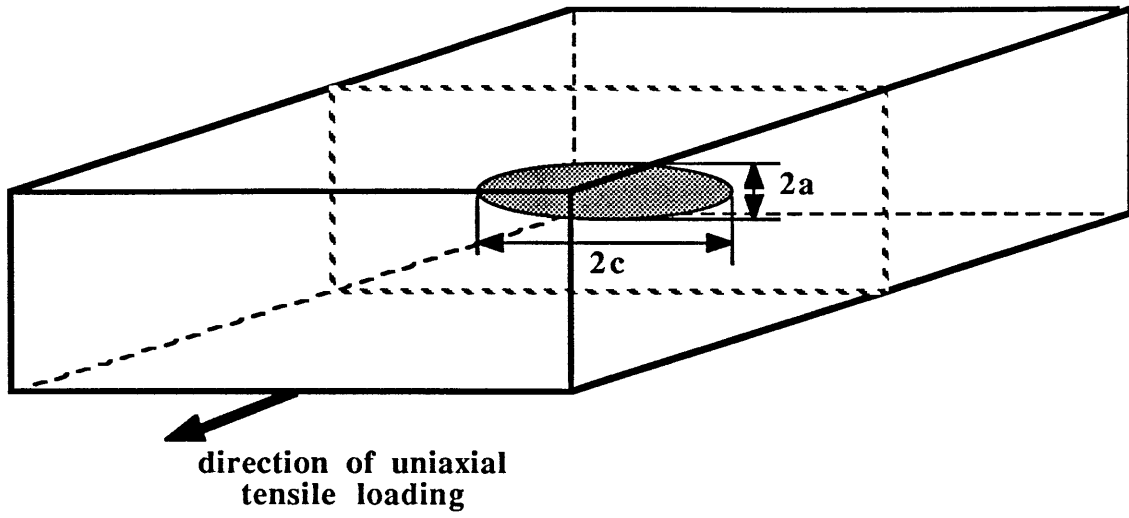


Figure 4-6:
Fe: elliptical crack shape correction factor

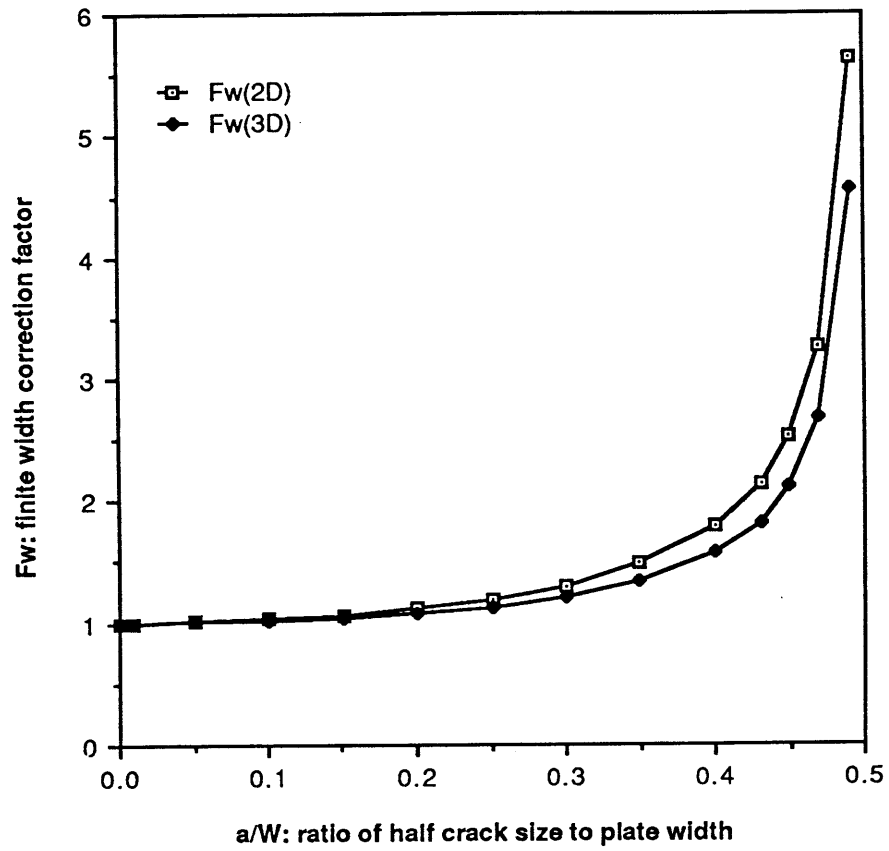
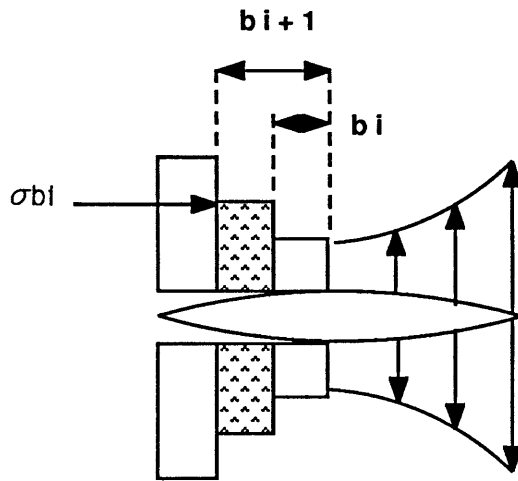
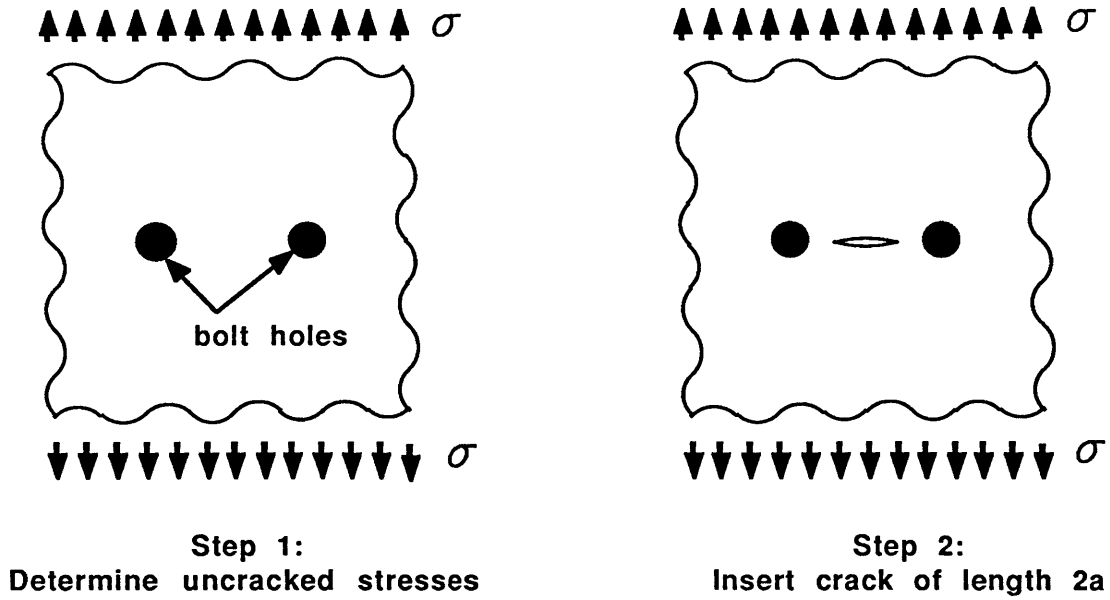


Figure 4-7:
Range of Fw: finite width correction factor



Step 3:
Remove step 1 stresses normal to step 2 crack by
integrating discrete pairs of splitting stresses

Figure 4.8:
Procedure for determining F_g

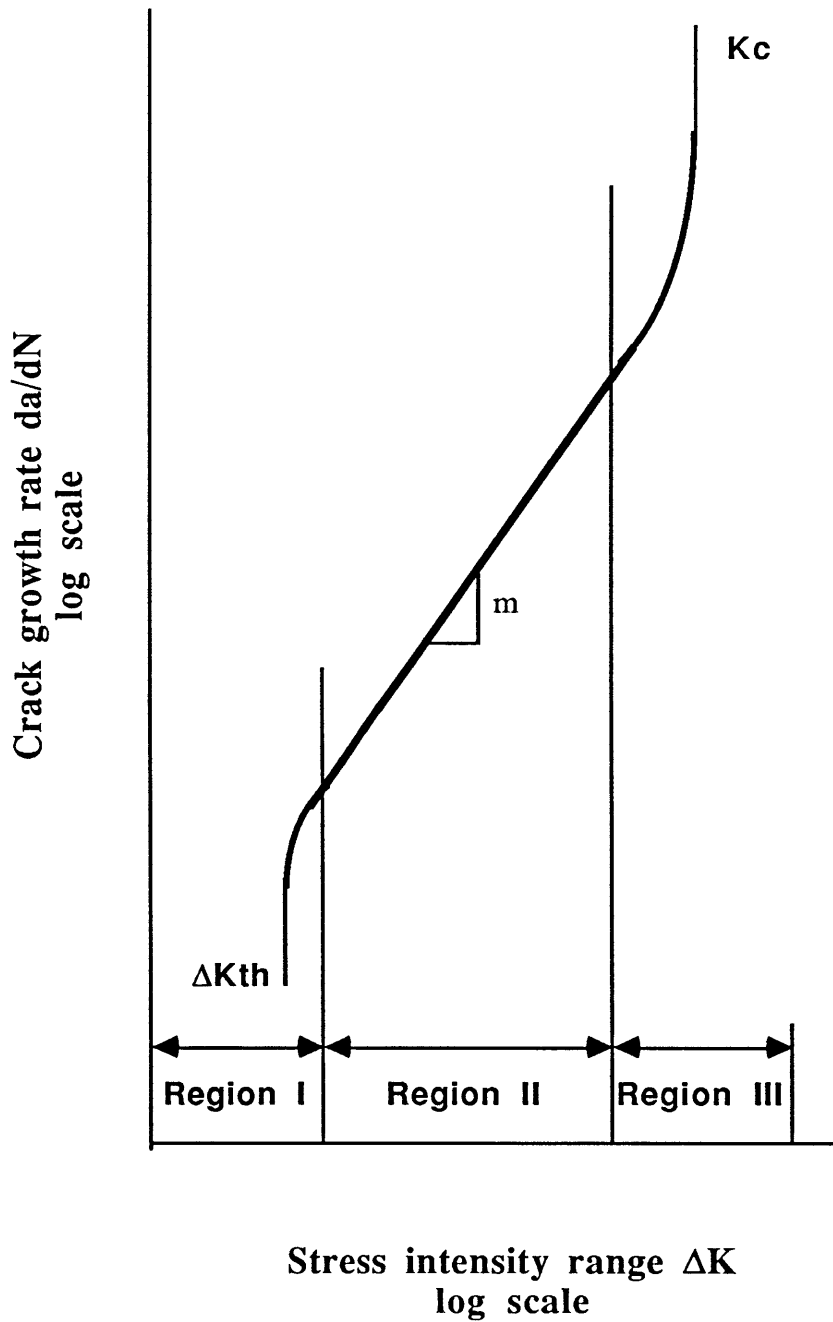


Figure 4.9:
Crack growth rate versus stress intensity range
in steel

General Condition	Situation	Kind of Stress	Stress Category (See Table 10.3.1A)	Illustrative Example (See Figure 10.3.1C)
Plain Material	Base metal with rolled or cleaned surfaces. Flame cut edges with ASA smoothness of 1,000 or less.	T or Rev ^a	A	1, 2
Built-Up Members	Base metal and weld metal in members without attachments, built-up plates, or shapes connected by continuous full or partial penetration groove welds or by continuous fillet welds parallel to the direction of applied stress.	T or Rev	B	3, 4, 5, 7
	Calculated flexural stress at toe of transverse stiffener welds on girder webs or flanges	T or Rev	C	6
Groove Welds	Base metal at end of partial length welded cover plates having square or tapered ends, with or without welds across the ends	T or Rev	E	7
	(a) Flange thickness < 0.8 in.	T or Rev	E'	7
	(b) Flange thickness > 0.8 in.	T or Rev	E'	7
	Base metal and weld metal at full penetration groove welded splices of rolled and welded sections having similar profiles when welds are ground flush and weld soundness established by nondestructive inspection.	T or Rev	B	8, 10, 14
	Base metal and weld metal in or adjacent to full penetration groove welded splices at transitions in width or thickness, with welds ground to provide slopes no steeper than 1 to 2 1/2, with grinding in the direction of applied stress, and weld soundness established by nondestructive inspection.	T or Rev	B	11, 12
	Base metal and weld metal in or adjacent to full penetration groove welded splices, with or without transitions having slopes no greater than 1 to 2 1/2 when reinforcement is not removed and weld soundness is established by nondestructive inspection	T or Rev	C	8, 10, 11, 12, 14
	Base metal at details attached by groove welds subject to longitudinal loading when the detail length, L, parallel to the line of stress is between 2 in. and 12 times the plate thickness but less than 4 in.	T or Rev	D	13
	Base metal at details attached by groove welds subject to longitudinal loading when the detail length, L, is greater than 12 times the plate thickness or greater than 4 inches long.	T or Rev	E	13
Fillet ^b Welded Connections	Base metal at details attached by groove welds subjected to transverse and/or longitudinal loading regardless of detail length when weld soundness transverse to the direction of stress is established by nondestructive inspection.	T or Rev	D	14
	(a) When provided with transition radius equal to or greater than 24 in. and weld end ground smooth	T or Rev	B	14
	(b) When provided with transition radius less than 24 in. but not less than 6 in. and weld end ground smooth	T or Rev	C	14
	(c) When provided with transition radius less than 6 in. but not less than 2 in. and weld end ground smooth	T or Rev	D	14
	(d) When provided with transition radius between 0 in. and 2 in.	T or Rev	E	14
	Base metal at intermittent fillet welds	T or Rev	E	—
	Base metal adjacent to fillet welded attachments with length L, in direction of stress less than 2 in. and stud-type shear connectors	T or Rev	C	13, 15, 16, 17
	Base metal at details attached by fillet welds with detail length, L, in direction of stress between 2 in. and 12 times the plate thickness but less than 4 in.	T or Rev	D	13, 15, 16
Mechanically Fastened Connections	Base metal at attachment details with detail length, L, in direction of stress (length of fillet weld) greater than 12 times the plate thickness or greater than 4 in.	T or Rev	E	7, 9, 13, 16
	Base metal at details attached by fillet welds regardless of length in direction of stress (shear stress on the throat of fillet welds governed by stress category F)	T or Rev	D	14
	(a) When provided with transition radius equal to or greater than 2 in. and weld end ground smooth	T or Rev	D	14
	(b) When provided with transition radius between 0 in. and 2 in.	T or Rev	E	14
Fillet Welds	Base metal at gross section of high-strength bolted slip resistant connections, except axially loaded joints which induce out-of-plane bending in connected material.	T or Rev	B	18
	Base metal at net section of high-strength bolted bearing-type connections	T or Rev	B	18
	Base metal at net section of riveted connections	T or Rev	D	18
	Shear stress on throat of fillet welds	Shear	F	9

Figure 4-10: Fatigue design category descriptions [AASHTO 85]

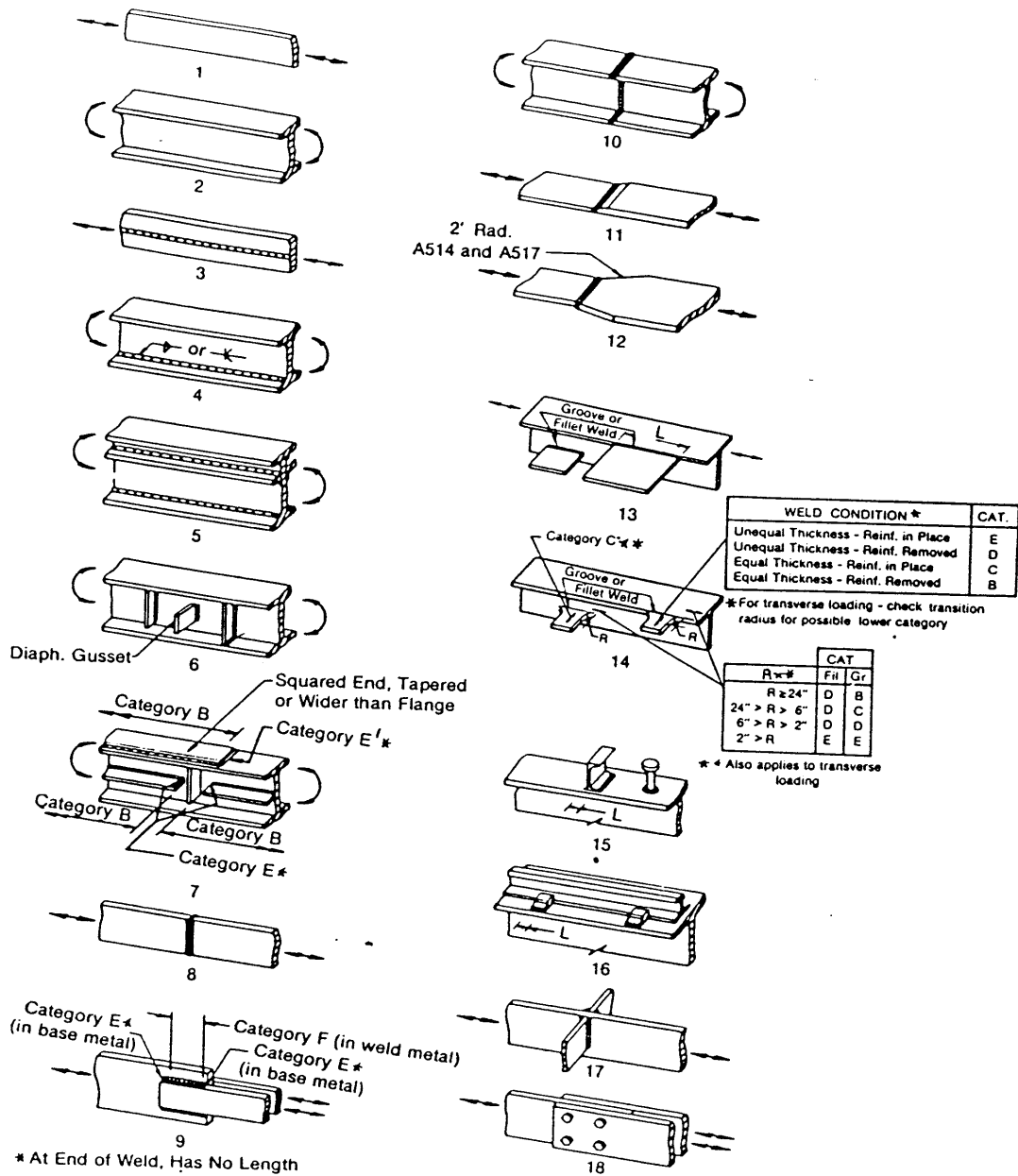


Figure 4-11:
Fatigue design category illustrative examples [AASHTO 85]

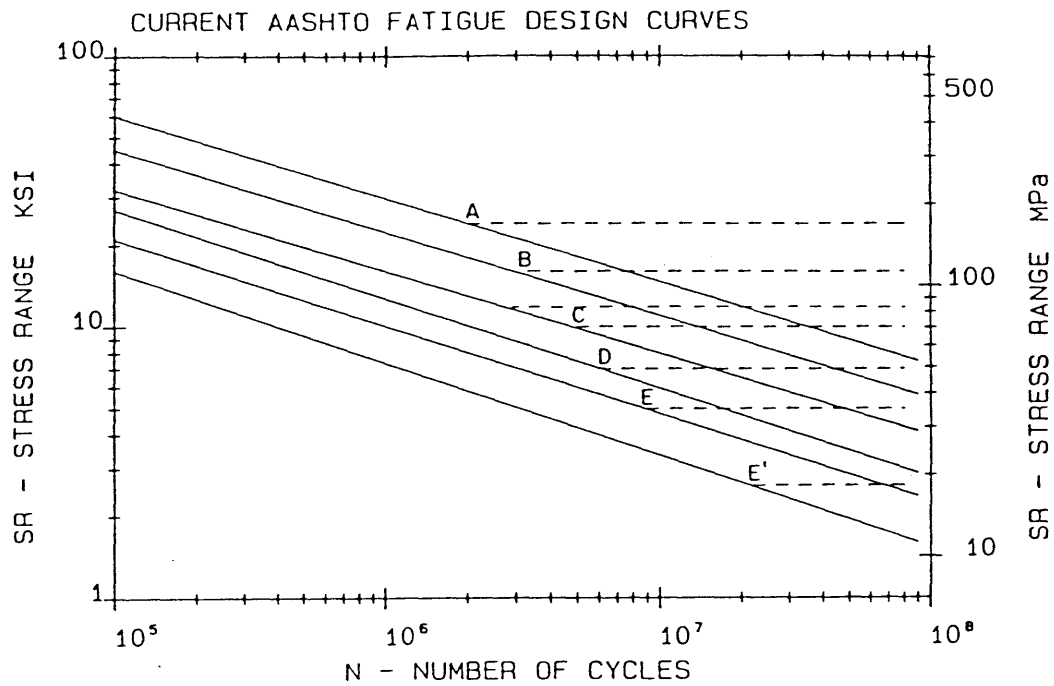


Figure 4-12:
Fatigue design curves [NCHRP 286]

primary cause for cracking (149 total)

secondary stresses and distortion-induced stress

out of plane distortion 64

excessive restraint 19

initial cracks and low fatigue resistant details

initial crack 44

low fatigue resistant detail 17

other causes

aeroelastic instability (hanger wind vibration) 4

stress corrosion cracking (eyebars) 1

low toughness material (not cited as primary cause)

Figure 4-13:

Taxonomy of primary causes for cracking in steel bridges

secondary stresses and distortion-induced stress (83 total)

out of plane distortion

web gaps

plate girder webs

in-service girder web gaps

floor-beam connection plates (+M region) 26

diaphragm connection plates (+M region) 9

tied arch floor beams 8

horizontal connection plates or gussets

lateral bracing vibration 3

gap between stiffener and gusset 2

diaphragm & floor-beam connection plates at piers

(-M region) 4

shipping and handling girder web gaps at transverse stiffeners 4

stringer web gaps at stringer to floor-beam brackets 4

cantilever brackets at cantilever tie plates 3

pin plates at frozen pins 1

excessive restraint

coped members at flame-cut notches 13

**(railroad bridge stringers, suspended span hangers,
or expansion joints at girders)**

stringer end connections at weld termination 3

tied arch floor-beams at weld root 2

lamellar tearing at rigid joints 1

Figure 4-14:

Secondary stresses and distortion-induced stress

initial cracks and low fatigue resistant details 61 total

initial crack

weld flaws

lack of fusion

groove welds **11**

(gussets and transverse in flange, web, coverplates,
or pin plates)

longitudinal stiffeners **4**

welded holes, plug welds, and welded repairs **6**

welded web inserts **1**

electroslag welds **6**

hydrogen cracking

(box girder: corner welds and transverse weld cold cracks) **4**

forge laps, unknown defects (eyebars and pin plates) 12

low fatigue resistant detail

flange gussets 5

intersecting welds 5

coverplated beams 4

flanges & brackets through web 3

**Figure 4-15:
Initial cracks and low fatigue resistant details**

primary cause for cracking in plate girder bridges

secondary stresses and distortion-induced stress

out of plane distortion

web gaps

plate girder webs

in-service girder web gaps

floor-beam connection plates (+M region)

diaphragm connection plates (+M region)

horizontal connection plates or gussets

lateral bracing vibration

gap between stiffener and gusset

diaphragm & floor-beam connection plates at piers
(-M region)

shipping and handling girder web gaps at transverse stiffeners

initial cracks and low fatigue resistant details

initial crack

weld flaws

lack of fusion

groove welds

(gussets and transverse in flange, web, coverplates)

longitudinal stiffeners

welded holes, plug welds, and welded repairs

welded web inserts

electroslag welds

low fatigue resistant detail

flange gussets

intersecting welds

coverplated beams

flanges & brackets through web

Figure 4-16:

Pruned taxonomy for cracking in welded plate girder bridges

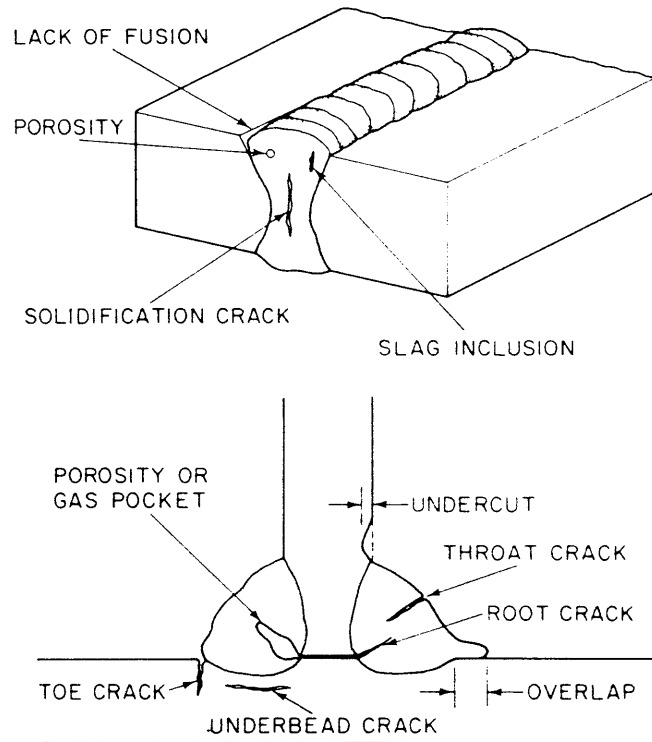


Figure 4.17:
Initial flaws in welded joints [Barsom & Rolfe 87]

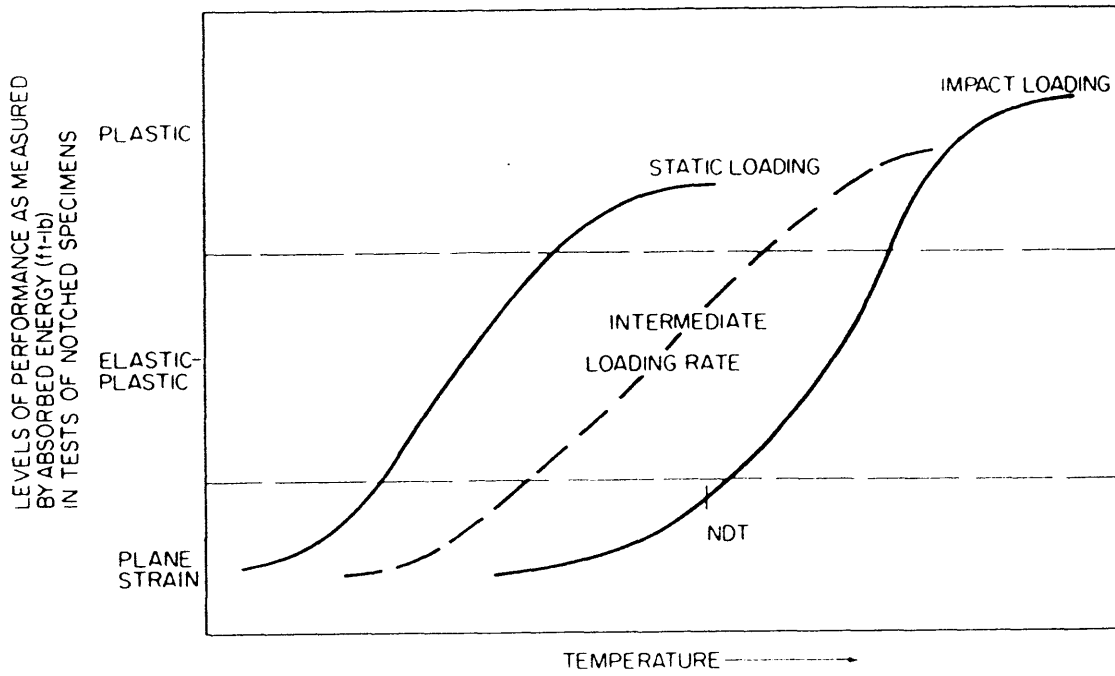


Figure 4.18:
Temperature and load rate effects on toughness
 [Barsom & Rolfe 87]

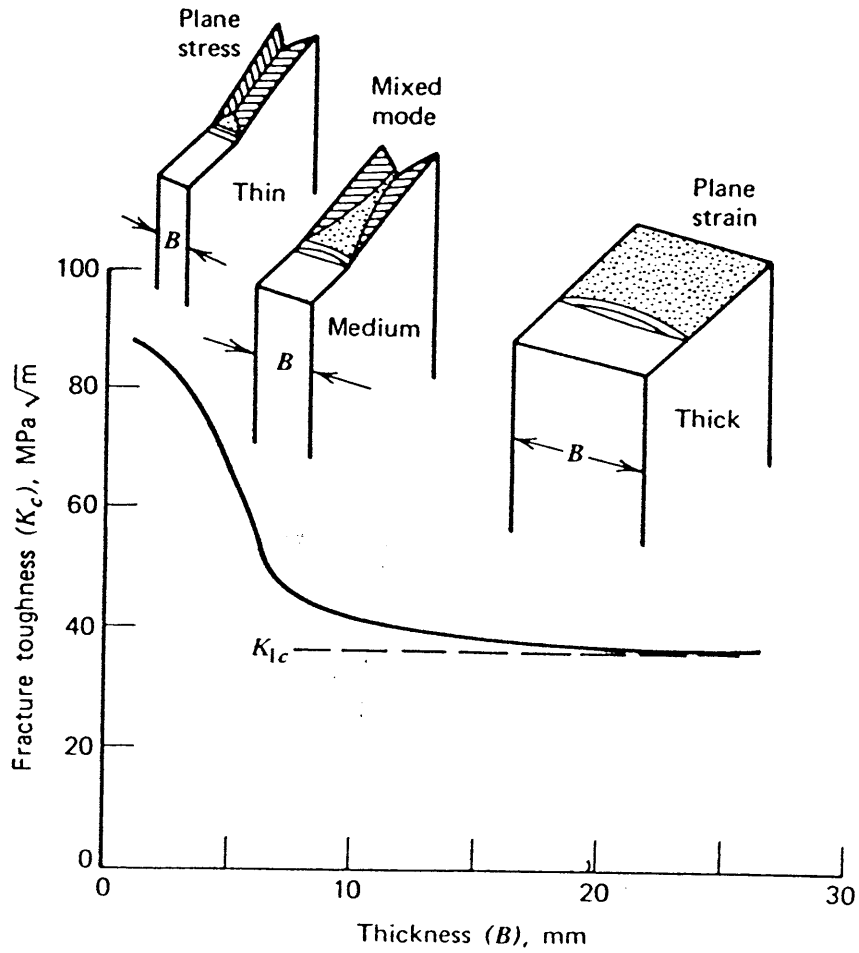


Figure 4.19:
Effect of specimen thickness on fracture toughness
[Fuchs & Stephens 80]

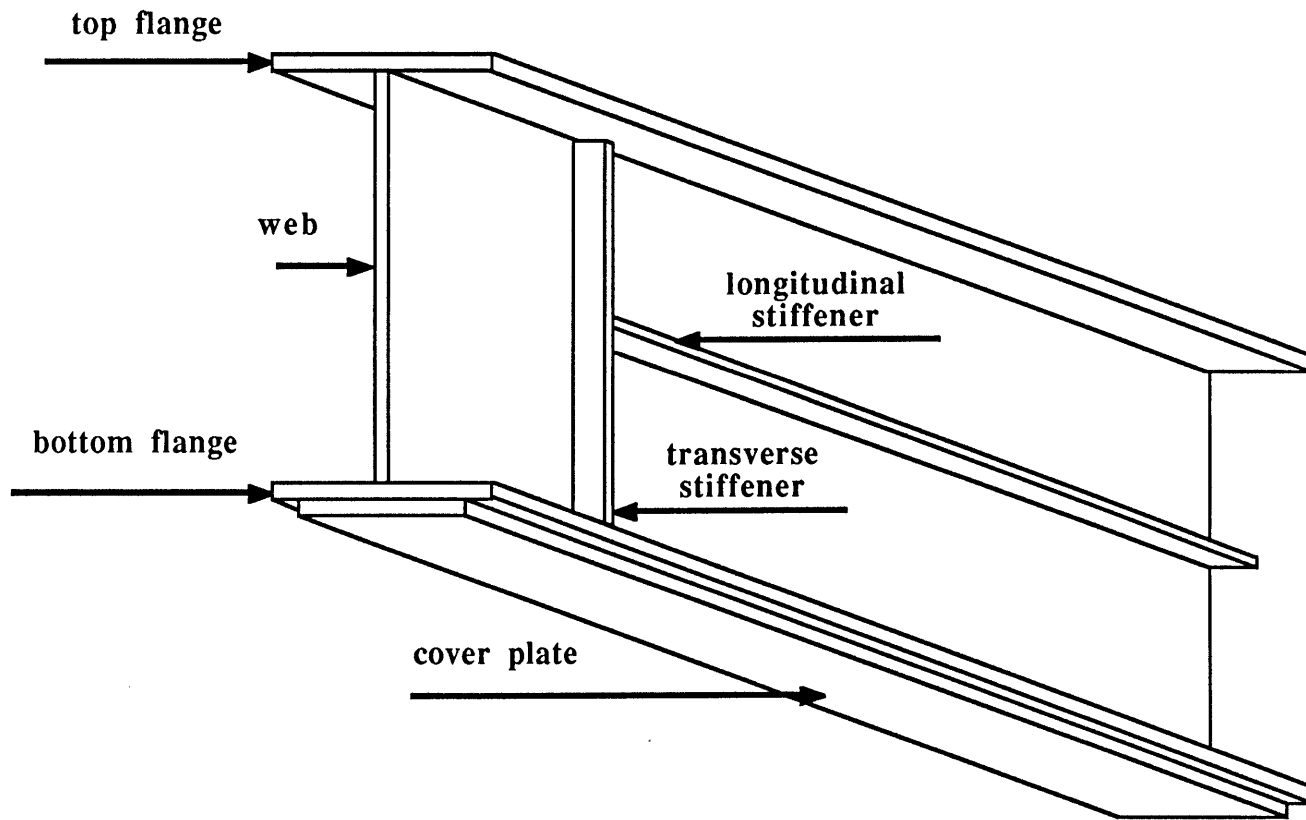
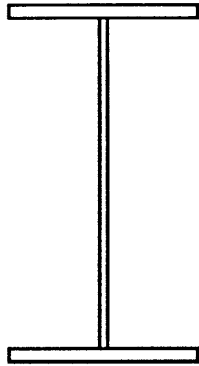
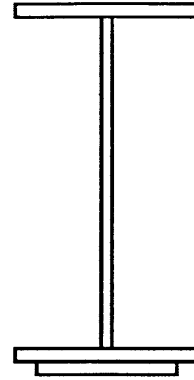


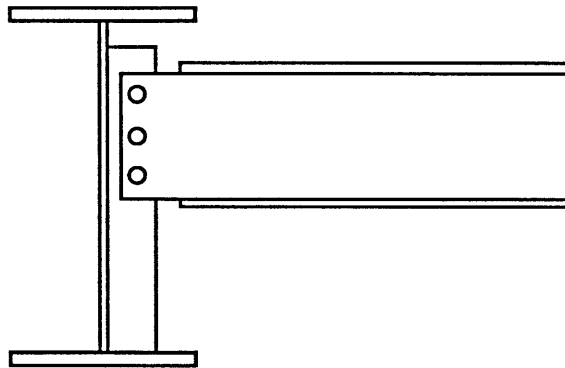
Figure 4.20: Some components of a plate girder



A: plain welded girder



B: coverplated girder



C: floor-beam connection plate web gap

**Figure 4.21:
Example girder cross sections**

Chapter 5

Architecture of CRACK

5.1 Project history

The development of the system proceeded in the evolutionary manner typical of knowledge-based systems, incrementally progressing from simple to hard tasks. Domain knowledge was first gathered from published technical references. To address the practical aspects, the summer of 1986 was spent inspecting bridges in the field as a member of a normal in-depth investigation team. The computer implementation phase began by exploring the problem domain using a simple rule-based tool to determine the types of knowledge representation techniques required and the level of complexity of the problem solving task during the fall of 1986. A first pass implementation of the multilevel architecture was then undertaken during the spring and summer of 1987. Feedback was obtained from other researchers investigating the coupling of symbolic and numeric computation at a workshop held in conjunction with the American Association of Artificial Intelligence in the summer 1987. This led to system revisions. Testing was then undertaken against cases in the literature. The testing phase included a trip to Lehigh University to refer to necessary unpublished reports and discuss the prototype with researchers working on a complementary project. Testing of final system was done in May of 1988 on an actual failure analysis which was not in the literature.

5.2 Implementation choices

5.2.1 System requirements

The requirements analysis stage of a project determines if the planned project will satisfy a real need while being technically and economically feasible. As with any other programming project, the development of an engineering knowledge-based system involves the comparison and selection of an appropriate set of software and hardware tools. To be able to select these tools, the desired characteristics of the assembled system must be defined. For our project, the key goal having the widest effect on system selection was deliverability to practicing engineers. A central motivation for our project was to demonstrate how to realize the promise of advanced computing techniques from Artificial Intelligence in the context of day-to-day work in even a small engineering office. Our list of desired system characteristics is therefore headed by deliverability. The development system does not need to be the same as the target delivery system, but the development process must result in a product able to be ported to the final field delivery

system.

Although the current system is being implemented solely as a research prototype in an academic environment, the design was influenced by consideration of requirements for a fieldable version. For deliverability to practicing engineers, micro-computer hardware of a type with widely applicable usefulness in an engineering office is needed. Accessibility favors personal computer based systems [Maher 87]. A typical engineering office can ill afford to purchase a computer system dedicated only to knowledge-based systems. For the engineer, the hardware should ideally serve as a satisfactory platform for software that addresses the specific area of knowledge-based applications while providing general computing capability. This not only allows the hardware to be used for other tasks in the office, a practical necessity, but also makes it possible to take advantage of rapid advances in software and to maintain flexibility for addressing new problems not foreseen in the original definition of needs. The new generation of personal computers which have become available within the last few years have made it possible to exploit some of the advanced computer methods developed for Artificial Intelligence. To make tools like knowledge-based systems viable for widespread use in civil engineering this transfer from an expensive special purpose hardware platform like a large LISP machine workstation to microcomputers was an essential step.

The system must be user friendly: easy to learn, convenient to use, and have a consistent operating interface. A multi-window interface that supports alphanumerics, menus, and high resolution graphics is needed to obtain a high rate of information communication (Figure 5-1). Graphics capability increases user communication through visualization, easing novice and intermittent use. The software required should be flexible and cost effective. If a high level language package must be purchased, its usefulness is greatly enhanced if it supports interlanguage communication to allow connection to other engineering software.

5.2.2 Software: Prolog

To choose a software tool, available commercial and research shells were first investigated. In general, the multiplicity of required features and possible hardware environments make it impossible for one single shell to satisfy all possible needs without becoming unwieldy. After experimenting with in-house software tools and several commercial products, the conclusion was reached that the shells available in mid 1986 when the project's implementation began did not provide a convenient method of

integrating the disparate reasoning mechanisms of rules, causal models, and numeric analysis which is essential for engineering problem solving.

For this reason, a tool which did have the desired capabilities was built. There are great advantages to using an in-house knowledge engineering tool, especially for a research project. The size of the system is kept small since only necessary features are included. Changes can easily be made as the project development proceeds. A symbolic language and an environment which supports incremental program development is required for system implementation. The strength of many AI techniques relies on exploiting the symmetry between code and data available in a symbolic language. Symbolic processing provides the flexibility needed to be able to try many different approaches to solving the problem. After experimenting with a small rule-based prototype in the problem domain, Prolog (programming in logic) was chosen as the development language. Many examples can be found for using Prolog to implement knowledge engineering tools [Hammond 84; Sterling&Shapiro 86; Bratko 86].

Prolog has many desirable features. Its goal oriented style of programming is generally clear, transparent, concise, and modifiable. There are two basic programming styles. The procedural or imperative approach describes the behavior needed to achieve the desired result. The declarative or descriptive approach gives a descriptive definition of a set of relations or functions to be computed. This procedural/declarative dichotomy can be clarified by making the analogy to material specifications (procedural, describe how to do task by supplying specific items) versus performance specifications (declarative, describe what requirements the end product must meet) in engineering. Prolog facilitates a declarative programming style. The emphasis is on what needs to be done by the program, not on how to do it. This allows concentration on the knowledge rather than the algorithms.

Prolog was an academic mathematical and research tool developed in Europe [Clark&McCabe 84]. Robert Kowalski at Edinburgh contributed crucial theoretical work in the early 1970's. The computer language was first implemented by Alain Colmerauer at Marseilles in 1972. The major step of producing an efficient interpreter/compiler was taken by David Warren at Edinburgh in 1977. This implementation established the Edinburgh syntax (also known as the DEC-10 syntax) which has become the industry standard as the most wide-spread and hence most compatible and portable of the various computer languages in the Prolog family. Clearly, compatibility and portability are

highly desired, since application growth and change in available host machines must be anticipated. The visibility of Prolog was greatly magnified by the public attention given to the 1981 initiation of the Japanese Fifth Generation Computer Project, which gives Prolog a central role.

It was pointed out that the implementation of an efficient compiler was a major step. Those familiar with only traditional programming languages such as FORTRAN and C may not be aware of the difference between an interpreter and a compiler. An interpreter reads a source program and translates it into machine language as it executes it, step by step. Interpreters allow stopping the program at any point, making changes, and continuing the same execution. This flexibility has the price of being time and space intensive. A compiler translates the whole source program to machine code, creating an object code. The object code is then separately executed to make the program run, resulting in much greater efficiency but making changes harder since the source must be edited and recompiled. The space and time requirements of real (non-academic or purely research) applications require the efficient execution of compiled Prolog.

Pattern directed rule-based programming is very natural in Prolog. Built in features of the language (pattern matching, automatic backtracking, and symbolic computation) combined with its well understood deductive formalism make it a good choice for development of an expert system. Prolog, as a full high level language, does have imperative features (assert, delete, cut, file input/output, etc.), so a custom tailored programming environment can be built up. Extensive numeric processing is neither convenient nor efficient in Prolog, so for engineering applications foreign function calls are a valuable extension. One of the attractive features of Prolog is its availability on small machines, allowing the development and delivery of sophisticated knowledge-based systems without large, costly, dedicated hardware.

LPA MacProlog [Clark et al 88], a small but full-featured Prolog, is the specific language utilized. It is compatible with Quintus Prolog, the popular Prolog used on workstations, and it's Edinburgh syntax allows portability. In addition to the interpreter, incremental and optimizing compilers are provided. Trace and debugging facilities are included. LPA MacProlog makes good use of the Apple interface with mice, menus, and icons. Graphic capability is provided with a Graphic Description Language. Numeric processing may be done within Prolog with satisfactory performance for this project. Foreign language calls to C and Pascal are available. This is a flexible software tool making both

symbolic and numeric reasoning available. The environment supports program development and is well integrated and flexible.

5.2.3 Hardware: Mac II

Having selected the software tool, Prolog, the hardware configuration was determined. Apple Computer's Macintosh line of personal computers is not new, but it was not until the introduction of the expandable Macintosh SE and Macintosh II, it included a suitable hardware platform for technical work like engineering. The Mac line's user friendly visual interface, strong graphics capabilities, and connectability into networks supporting multiple kinds of computers, not just Apple products, are all attractive features. The speed and customizability of the Macintosh II, whose standard features include a math co-processor and six expansion slots, make it the most attractive choice for engineering CAD/CAM/CAE and architectural applications [CAD/CAM 87]. The open architecture and commitment of Apple to upward compatibility are attractive features in a world of rapidly changing machines. The use of the Macintosh line of computers in the engineering community is rapidly expanding, leading to a positive feedback of more users supporting an expanded software market, which in turn encourages more users. The availability of high quality engineering software is thus growing quickly.

5.3 System modules

This section describes the main modules that make up CRACK. An overview of each layer is presented with sketches showing the program structure and primary components. Each component is described by giving its purpose and explaining the upper levels of Prolog predicates used to implement the components function.

5.3.1 Rule-based level

The term knowledge-based system applies to programs with explicit knowledge representation and separation of the knowledge base from the inference mechanism. This separation of the knowledge from the inference engine is one of the major architectural principles derived from early experience in building such systems [Davis 82]. Knowledge-based systems have a domain dependent part, called the **knowledge base**, and a domain independent part, called the **shell**. The knowledge base contains the domain information made up of facts, rules describing relations or physical laws, and methods for problem solving in the domain. When the primary method of representing this domain knowledge is rules, the system is categorized as rule-based. Rule-based systems embody another

major architectural principle, using as uniform a representation as possible [Davis 82]. Casting domain knowledge into rules is a uniform representation mechanism of surprising expressiveness. The shell contains the inference engine which uses domain independent methods to draw conclusions by manipulating and using the knowledge base.

5.3.1.1 Shell

5.3.1.1.1 Role of the shell

The shell provides the means for utilizing the knowledge base, as illustrated in Figure 5-2. The basic functions of the shell are a knowledge representation scheme, an inference or search mechanism, a means of describing a problem, and a way to determine the status of a problem while it is being solved [Citrenbaum et al 87]. Figure 5-2 shows the shell divided into two main parts, the interactive component and the inference component. The interactive part handles all the user interface tasks for adequate communication between the system and the user. The inference part contains the actual inference engine for reasoning, the context of the particular consultation, the incomplete solutions as the problem proceeds, and the connections to the knowledge base. Although shells are in principle domain independent, in practice applications must be closely related to satisfactorily use the same shell.

The shell for CRACK is built in Prolog. Prolog is a good tool for implementing a shell because parts of the four basic functions of knowledge representation, inference mechanism, problem description, and status determination are inherent in the programming language. Logic can be used to represent knowledge about objects and actions [Gevarter 87]. The logic formalism of Prolog clauses can be used to express domain object descriptions as facts and actions as rules. The native Prolog inference engine may be used to perform backward chaining on the knowledge base of facts and rules. A limited kind of knowledge-based system can be implemented directly in a subset of Prolog using only logic clauses and the intrinsic theorem prover [Weiss 86], but this approach lacks all but the most rudimentary abilities of problem description and status determination. There is no ability to query the user for facts as they are needed in the course of the problem solution, so all problem specific data must be provided initially. The only mechanism to determine the status of the problem description is to use the debugging trace facility.

Although certain Prolog programs that do not have any shell separate from the Prolog language itself may be considered to be in some sense knowledge-based systems, it is an error to assume that using Prolog as the implementation tool will automatically result in building a knowledge-based system. Prolog programs do not necessarily separate knowledge from the way in which the knowledge is processed. The full Prolog language contains many features, such as meta-logical predicates, cuts, and input/output, that are not part of logic programming. In general, a Prolog program contains a great deal of information about how the program executes. In addition the execution model of Prolog incorporates goal resolution scheduling policy and search and backtracking choices that are not required by the underlying logic programming approach [Sterling&Shapiro 86]. Most Prolog programs have little resemblance to a knowledge base that contains meaningful information independent of the inference mechanism.

The limited knowledge-based capability of bare Prolog can be expanded by providing a shell. Use of a shell can clarify the separation of knowledge from the algorithms that use the knowledge. If the knowledge-based system must be capable of generating explanations and justifications, the operation of the inference engine must be accessible to the program. This necessitates building an inference mechanism on top of Prolog's intrinsic theorem prover, whose execution is not available for introspection. If this is not done, the key advantage discussed in Chapter 2 of knowledge-based systems ability to explain its behavior is lost.

5.3.1.1.2 Implementation of the shell

The rule-based facilities and explanation mechanism are based on the expert system structure outlined in "micro-PROLOG: Programming in Logic" [Clark&McCabe 84]. The shell, affectionately named MESS (My Expert System Shell) has grown from an initial kernel implementation by Lindsey Spratt. The shell fits the two component layout illustrated in Figure 5-2. The interactive component provides the interface between the system and the user, managing input, requesting input data, explaining system behavior, and justifying results. The inference component accesses and manipulates knowledge in the rule base and information from the user to build a chain of reasoning to answer the user's query. The inference component builds a logical proof, managing the working memory and keeping track of partial solutions.

The structure of the rule-based shell for CRACK is illustrated in Figure 5-3. The

interactive component has three modules: the **query manager**; the **input manager**; and the **explanation manager**. The inference component depends on the **proof builder** as the inference engine. The data and control flows between the modules are shown in Figure 5-3. The control flow begins with the user querying the system, starting a consultation. The query manager invokes the proof builder to establish a chain of reasoning which answers the user's question. If the proof builder requires information from the user, the input manager is invoked to handle the dialogue with the user. Data flows to the proof builder from the rule base, for static domain knowledge, and from the user, for consultation specific information. The input manager posts the user's answers to **working memory** where the proof builder may access user supplied facts describing the individual consultation. During the course of building a proof, the partial solution is assembled in a **goal stack**.

Explanation is a crucial feature of a knowledge-based system [Bratko 86], used to validate the system's reasoning during a consultation. A shell should provide mechanisms to explain its behavior and justify its requests for information and its problem solutions. As a minimum it should be able to explain why it asks particular questions and how it reached its conclusions. The explanation manager examines the goal stack to construct how and why explanations. As shown in Figure 5-3, the explanation manager is invoked through the input manager to provide an explanation of why the system is requesting certain information from the user, and through the query manager to provide an explanation of how a system conclusion was reached.

Given this overview of the structure of the rule-based shell, each module is presented in more detail below.

The **query manager** consists of the predicates **query**, **which**, **how**, and **whynot**. **query** is a top level predicate which tries to establish a line of reasoning to support the user specified top goal/s. **which** is the same as query except that the variable bindings are collected for each of the specified goals. **how** shows the chain of reasoning which supports the goal/s. **whynot** is analogous to **how** for a failed goal, showing why the goal/s could not be supported.

The **proof builder** consists of the predicates **confirmed**, **confirmed_single**, and **proved** (which is only used for "how" explanations). The inference engine is housed in the predicates **confirmed** and **confirmed_single**. **confirmed** establishes the veracity of a

goal in one of the following three ways. First, if the goal is a sequence of goals connected by logical “and” operations, the first goal must itself be confirmed singly and the remaining group of goals must be confirmed. Second, if the goal is a sequence of goals connected by a logical “or” operation, the first goal must be confirmed singly or the remaining group of goals must be confirmed. Third, if there is only one goal, it must be confirmed singly. **confirmed_single** establishes the veracity of a single goal. It does this by matching the goal to the head of a clause (fact or rule) in the knowledge base. If the goal matches a fact it is immediately proved. If the goal matches the head of a rule, the veracity of the body of the rule is established.

As an example, a rule in the knowledge base states that any date earlier than 1965 was prior to the routine nondestructive testing of welds for a particular state’s Department of Transportation. As part of a consultation, it is necessary to confirm whether NDT was used on a bridge constructed in 1954. The goal “prior_NDT(1954)” would match the head for the rule (in Prolog format) “prior_NDT(Date) :- Date < 1965.” **confirmed_single** would establish the veracity of the rule body by accessing the Prolog meaning of “Date is 1954, Date < 1965”, which is true.

The inference engine is given access to the Prolog meaning of the knowledge base contents, so the variables and predicates in the knowledge base have meaning in the predicate calculus of Prolog. Any predicate defined in the Prolog language can be used directly in the knowledge base without the need for procedural attachment techniques. The inference mechanism is simple, operating as a resolution theorem prover. A depth first search is performed by backward chaining. When several rules could be tried at any one time, the rule appearing first in the rule base is tried first.

The **goal stack** is built as the inference engine searches for a proof. The path traced so far in generating the partial solution is kept as a context parameter while a goal is being confirmed. This information is used by the explanation manager.

When the inference engine is running, some information which is not in the knowledge base but which is problem specific must be supplied by the user. This problem specific information is kept in **working memory**. The **input_manager** is composed of the predicates **reported**, **user_reported**, **predicate_input**, **get_input**, and a set of predicates ancillary to **get_input**. **reported** scans the working memory to find if the user has already confirmed or denied the desired information and if needed manages the inter-

action for querying the user. User stated facts about the case under examination are asserted into the working memory by `user_reported`. `predicate_input` accesses the rule base's input menus to show the user the appropriate list of valid responses. `get_input` pops up the suitable kind of input window.

The **explanation manager** answers “why” questions by displaying the goal stack, built to preserve the context within which information is requested from the user. Whenever `get_input` pops up a menu, the user can thus ask why a question is being asked. A “help” button is also displayed, which provides access to text describing the piece of problem specific data requested. How questions are answered by showing the immediately preceding link in the chain of reasoning supporting the conclusion. The appropriate rule and the successful variable bindings are shown. Repeated uses of “how” are needed to show the whole reasoning chain. The predicates composing the explanation manager largely address the details of displaying the goal stack and rules and are not discussed individually.

In addition to these major modules, the shell has additional predicates which are gathered into toolboxes. These predicates perform basic database access and clause manipulation tasks and provide other tools to support the work of the major modules.

5.3.1.2 Rule base

There are three main types of rules: domain specific problem solving methods; hierarchy of crack causes; and knowledge about fatigue and fracture. The first type of rule establishes protocols for solving different types of problems within the domain. These **protocol** rules post the initial appropriate sequence of goals to accomplish the different tasks of failure analysis, prediction, and design critique. The knowledge for these rules was acquired from published instructions by experts in how to perform these tasks [Fisher 81; NCHRP 87] and by distilling this information from examining multiple failure analysis cases. This domain specific set of problem solving methods contain knowledge about the conventional order for data gathering and ways of decomposing the overall task into a series of simpler steps. Establishment of such protocols has been found to greatly facilitate the development of the knowledge base, providing a conceptual structure to give direction to the knowledge explication process [Bennett&Englemore 79]. Without such a conceptual structure to organized knowledge acquisition, a web of facts and rules may be assembled that is not sufficient to complete a line of reasoning to solve any of the domain tasks.

To generate hypotheses about the cause of bridge cracking, a classification approach is used. The possible causes of bridge cracking have been organized into a hierarchy. The system requests information on the bridge in question until enough data has been gathered to place the possible crack type in the hierarchy. The most common cracking types are tried first, so that the most likely hypothesis which can be substantiated is presented as the system response. The pruned hierarchy for cracking in plate girder bridges presented in Figure 4-16 is used for the **hypothesis** generation rules.

The domain rules representing knowledge about fatigue and fracture are grouped according to whether they address initial flaw size, stress history, material properties, or structural geometry.

5.3.2 Qualitative simulation

The qualitative level is based on Kuipers' QSIM algorithm [Kuipers 86]. Quantity spaces for QSIM are defined as a linearly ordered set of landmark values. New landmarks may be discovered and inserted. Quantities are described in terms of their ordinal relations with the landmark values. QSIM starts with a set of qualitative constraint equations describing the system and an initial state, and produces a directed graph of the possible future states of the system. The possible behaviors of the system are the paths through the graph. An example clarifies these concepts. The upper right corner of Figure 5-4 shows a plate of thickness T with an initial flaw size a_1 which may be assumed to be subjected to cyclic stress. Figure 5-4 shows a simplified version of the causal influence network for fracture and fatigue, from which the qualitative constraint equations may be drawn. The rectangular boxes are the network nodes which represent the four variables which describe the system's state. These state variables are the crack length, a , the stress intensity factor K , the stress intensity range, ΔK , and the crack growth rate, da/dN . The rounded boxes contain the boundary quantities for each state variable. For example the crack length, a , lies between zero and the plate thickness, T . The arrows between the rectangular boxes represent the qualitative influences between the state variables. For example, the crack growth rate, da/dN , is the derivative of the crack length, a . The crack length, a , has a positive monotonic relationship to the stress intensity factor, K . Figure 5-5 shows the different states in boxes. The paths between the boxes show the three possible ways the system can act over time: 1) the initial flaw grows by fatigue and penetrates the plate; 2) the initial flaw grows by fatigue until the critical crack size is reached and the crack grows unstably (fracture); and 3) the initial flaw grows by fatigue and simulta-

neously penetrates the plate and fractures (which is only theoretically possible).

As shown in Figure 5-5, a qualitative state parameter is either at a landmark value (K begins at the landmark value K_i and ends at the landmark value K_c for fracture) or between landmark values (K is between K_i and K_c for fatigue growth). Similarly, time is represented as an ordered set of distinguished time points and each state is either at a time-point (the initial flaw state is at time 0, the fracture state is at time 1) or between time points (the fatigue crack growth is between times 0 and 1). A qualitative state parameter has various transitions it may undergo to pass from one state to another (K may increase after the initial state, but cannot reach K_c without passing through the region between K_i and K_c). The qualitative differential equations put constraints on which of these transitions are possible simultaneously for related parameters (when the crack length increases, K also increases). Valid next states are described by all possible consistent parameter transitions.

QSIM is an efficient algorithm to predict the next state. The steps in the algorithm are [Kuipers&Patil 87]:

- Propose transitions for each parameter.
 - Filter for consistency with current state.
- Form tuples at each constraint.
 - Filter for consistency with constraint.
 - Filter for consistency with corresponding values.
- Local consistency (Waltz) filtering on tuples.
- Form possible successor states.
 - Filter for global consistency.

A complete specification of the QSIM algorithm is presented in [Kuipers 86]. QSIM is available from Prof. Benjamin Kuipers, Department of Computer Sciences, The University of Texas at Austin, Austin, Texas 78712, ai.kuipers@R20.UTEXAS.EDU , in Common Lisp and C implementations.

The qualitative simulation mechanism used in CRACK is a version of PQSIM [Dvorak 86], a Prolog program that functionally does what the basic QSIM program does, but which is less efficient due to the way the filtering is done (and in particular the absence of Waltz filtering). The set of qualitative constraint equations is expressed as an ex-

ecutable Prolog goal, **consistent**. The generation of candidate states and the filtering for consistency are done by the native Prolog resolution and backtracking mechanism. The major predicates in PQSIM are **find_initial_state**, **envision**, **predict_from**, and **make_new_states**. **find_initial_state** takes the initial state, which may only partially specify each parameter's position in its quantity space and direction of change, fills out the unspecified initial values, and checks that the complete initial state is valid (unique and consistent with the constraints). **envision** is the main recursive predicate which predicts all the new states that can be reached, building the full directed graph of possible states. **predict_from** generates all possible successor states to the selected current state. **make_new_states** detects cycles, creates new landmarks, and posts each new state. The domain causal network is illustrated in Figure 5-6. This is an expanded version of the network shown in Figure 5-4 and discussed earlier in this section. The following basic arithmetic, differential, and functional qualitative constraints are used to show relationships between the state variables: (+, addition), (*, multiplication), (deriv, derivative), and (M+, strictly monotonically increasing). Notice that although crack length, *a*, actually directly influences *F_g* as well as *F_w* and may indirectly influence *F_e* and *F_s* the network only directly accounts for the effect on *F_w*. The state variables represented by the network nodes are:

effective temperature: Temp_eff
 residual toughness: Kr
 fracture toughness: Kc
 stress intensity factor: K
 stress intensity range: ΔK
 crack growth rate: da/dN
 crack length: a
 total correction factor: r F
 elliptical correction factor: Fe
 free surface correction factor: Fs
 stress gradient correction factor: Fg
 width correction factor: Fw
 stress residual: s_{residual}
 stress dead: s_{dead}
 stress live: s_{live}
 stress total: s
 initial flaw size
 flaw growth
 weld size
 load rate
 strain rate
 temperature: Temp

The qualitative constraint model is made up of the following parts:

Temperature effect on K_c :
 $Temp_eff \text{--[M+]-- } K_c$

Enforcing $K < K_c$ for fatigue:
 $K_c = K_r + K$

The main fatigue growth driving loop:
 $F * s_{live} = \Delta K$
 $\Delta K \text{--[M+]-- } da/dN$
 $da/dN < \text{--[deriv]-- } a$
 $a \text{--[M+]-- } F_w$
 $F = F_e + F_s + F_g + F_w$
 $F * s_{live} = K$

Stresses:
 $s = s_{live} + s_{residual} + s_{dead}$
 $F * s = K$

Flaw size:
 $Weld_size \text{--[M+]-- } Initial_flaw$
 $Initial_flaw + Flaw_growth = a$

Temperature and strain rate:
 $Load_rate \text{--[M+]-- } Strain_rate$
 $Strain_rate + Temp = Temp_eff$

5.3.3 Quantitative analysis

The quantitative level numerically simulates the growth of a crack using the LEFM techniques discussed in Section 4.5.4 for determining stress intensity factors and Section 4.5.5 for Paris power law crack growth. The functions that perform the numeric calculations are not large, pre-existing programs. Representing numeric processing elements as a library of atomic processes allows independent access by inferencing elements without requiring significant advance planning. The calculations are done by small procedures implemented in Prolog. Originally, the numeric portion of the program was implemented in Prolog with the intent that this level would be rewritten in C to obtain satisfactory speed for interactive use. The compiled execution speed of the Prolog implementation was found to be satisfactory for demonstration purposes in a research setting, so no reimplementations were done.

Crack growth is simulated by the predicate `grow_crack` using Paris power law direct integration. A limit to crack growth defining the final crack size must be specified before simulation begins. This may be a natural condition of the plate girder geometry, like web or flange thickness, or a more arbitrary limit like 1 inch. The basic algorithm

follows:

1. Find ΔK for the starting crack size
2. Find ΔK for the final crack size
3. If the final ΔK is within 10% of the limiting ΔK , then grow the crack using a single integration step
4. If not, reduce the final crack size for the increment of crack growth to half the remaining ligament
5. Continue crack growth steps, cutting the size of the increment of crack growth in half as often as required to satisfy the condition that each incremental step size is small enough that the crack geometry does not change appreciably as discussed in Section 4.5.5. The concentric rings in Figure 5-7 indicate the location of the crack front for succeeding crack growth incremental steps.

As discussed in Section 4.5.4, there are several methods available to calculate stress intensity factors, each suitable for different degrees of accuracy, with steeply increasing computational resources required for improved accuracy. The decomposition method illustrated in Figure 4-5 is used in CRACK. The quantitative level graphically displays crack growth as it is calculated as shown in Figure 5-7.

5.3.4 Interlevel communication

Since the same language is used to implement all levels of the program for both symbolic and numeric reasoning, interlevel communication was straightforward. The data structures of all levels are mutually accessible. The description of the problem plate girder cross-section is stored in a girder data structure. The girder data structure is built up of plate and connection components, so the data structure in some sense is analogous to the way the actual plate girder is constructed. Major data structures are also provided for cracks and hypotheses. In general, communication is through parameter passing rather than assert/retract. Using parameter passing eases truth maintenance since the native Prolog variable binding mechanism keeps the passed data consistent with the current set of assumptions expressed as other variable bindings.

5.4 Flow of the program

To clarify the flow of the program, the sequence of a CRACK problem solving session is discussed in this section. To see the details of program interaction for a particu-

lar failure analysis, refer to Section 6.2.

The steps in a CRACK problem solving session are:

- 1) Establishing the type of problem to be solved, either design critique, predictive modeling, or failure analysis.
- 2) Describing the problem by gathering information on the girder's geometry, service history, material properties, and observed cracking symptoms.
- 3) Hypothesizing a crack cause.
- 4) Qualitatively simulating possible crack progression sequences to guide quantitative analysis.
- 5) Performing fracture mechanics calculations to determine critical crack sizes and fatigue lives.
- 6) Evaluating the hypothesis to confirm it as most probable crack cause (or to reject it, in which case the sequence loops back to step 3).
- 7) Stating the conclusions.

Phases 1, 2, and 3 are performed by the rule-based shell. Qualitative simulation is used for phase 4. Numeric analysis routines perform phase 5. The final phases 6 and 7 are again done using the rule-based shell.

5.4.1 Task type

A consultation with CRACK begins with a question as to what type of problem is to be solved. The problem solving mode sets the kind of solution to be sought as well as the appropriate problem solving protocol. For a design critique case, the weak point of the cross-section is identified, the fatigue life is calculated, and the probable crack progression is predicted. For a predictive modeling case, the likely crack symptoms for the cross-section are projected based on the specified age and service conditions. For a failure analysis case, the observed cracking failure is explained.

5.4.2 Problem description

To generate an explanation satisfactorily supported by the available evidence, the pertinent information must be gathered. The order that the system requests facts about the case is structured to be sensible and natural to the engineer user. Related facts are grouped together. Sometimes this results in more questions being asked than the minimal set necessary to reach the final conclusions, but the increased ease and coherence of the user-system dialogue is worth the additional questions. The queries are ordered in the following groups: structural configuration, service history, and material properties, and

crack description.

Since the research prototype version of CRACK focuses exclusively on plate girder bridges, the user must pick a plate girder as the type of structural member that experienced the failure. The structural geometry of the member is then provided by the user in response to menu prompts and a scaled cross-section is drawn.

Service history facts are requested next. The start and end dates of bridge use establish the number of years of service. Stress range histograms or traffic counts and design stresses are used to derive the number and range of effective constant amplitude stress cycles. If material properties of yield and fracture toughness are not user supplied they are looked up in the knowledge base from the specified steel ASTM designation. The final user supplied information for the problem description step is observed cracking symptoms. This information would only be available for a failure analysis and that is indeed the only context in which it is requested. Suspected initial flow location, size, and shape are requested as are fractographic features such as chevron markings, beach marks, and shear lips.

5.4.3 Forming hypotheses

The user supplied problem description information and the crack cause taxonomy shown in Figure 4-16 are used to generate a probable crack cause hypothesis. The hypothesis generation follows the evidence-gathering paradigm [Buchanan&Shortliffe 84]. The hypothesis is found not by construction from primitive elements, but by looking at items from a pre-defined list (Figure 4-16). The possible causes in the taxonomy are linked to the user supplied evidence in the problem description by rules. The ordering of crack causes within the taxonomy is from most to least common. The hypothesis which is selected is the first crack cause in the taxonomy that is supported by the evidence. For example, if the problem description structural configuration has floor beam connection plates, CRACK looks for evidence of a web gap to support the hypothesis of web gap distortion at that location. If there is no web gap, evidence will be sought to support one of the weld flaw hypotheses. The evidence gathered in this way supports a hypothesis which is then tested by the following steps 4, 5, and 6.

5.4.4 Qualitative simulation

Having identified a suspected crack origin, the next phase is to envision the

possible ways the crack could grow. This task consists of constructing a tree of the different states the crack may pass through from origination to failure. A crack can grow in different ways through the particular configuration of a beam or plate girder. At a particular time point, the crack growth may be stable fatigue or unstable fracture. This transition from stable to unstable growth depends on conditions such as temperature as well as current crack size. When the geometric limits of a component are reached, the crack may change shape (say from a thumbnail to a through crack), and continue growing in the same plate or it may begin to grow in a connected plate. All possible crack progression sequences for the particular structural configuration and service condition variations are generated by the qualitative simulation.

5.4.5 Fracture mechanics calculations

Given the set of possible cracking sequences, analytic routines are used to simulate crack growth. The number of cycles for crack growth and the stress intensity factors are computed for each fatigue stage. Once these values are quantified, a single cracking sequence can be determined. A choice can be made between multiple possible fatigue stages by taking the one reached with the lowest number of cycles. A choice can be made between fatigue and fracture modes of crack growth based on whether the stress intensity factor reaches the material's fracture toughness.

5.4.6 Evaluation of hypothesis

When the quantitative analysis has been completed, execution returns to the rule-based level. The numeric results such as final crack size and fatigue life are compared to the corresponding facts stated in the problem description. "Poor", "reasonable", or "good" agreement is found between predicted values calculated by the quantitative level and observed values stated by the user. These linguistic terms are generated in a simple manner. If the predicted and observed values differ by a factor of 2.5 or more, the agreement is poor. If they differ by a factor greater than 1.5 but less than 2.5, the agreement is reasonable. If they differ by a factor of 1.5 or less, the agreement is good. This linguistic mapping is rough and arbitrary, but serves as a first cut for prototype purposes. If there is poor agreement for fatigue life, the hypothesis is *rejected* and a new possible cause of cracking must be explored. Definition of poor prediction of fatigue life as sole failure criterion for causing rejection of the hypothesis and backtracking to next crack cause in the taxonomy is again an opportunistic simplification for prototype use.

5.4.7 Statement of conclusions

The report is generated by the rule-based level. CRACK has no natural language generation capability. A text summary of the consultation session is generated by using template sentences and filling in the blanks with the appropriate user supplied or inferred values. The format for the conclusion consists of a summary of the problem description (geometry, materials, and service conditions), a description of the probable crack cause, the probable crack progression sequence to failure, and a comparison between observed and derived fatigue life and crack size. If crack symptoms were reported by the user, the conclusions state whether or not the symptoms substantiate the selected crack cause and progression sequence.

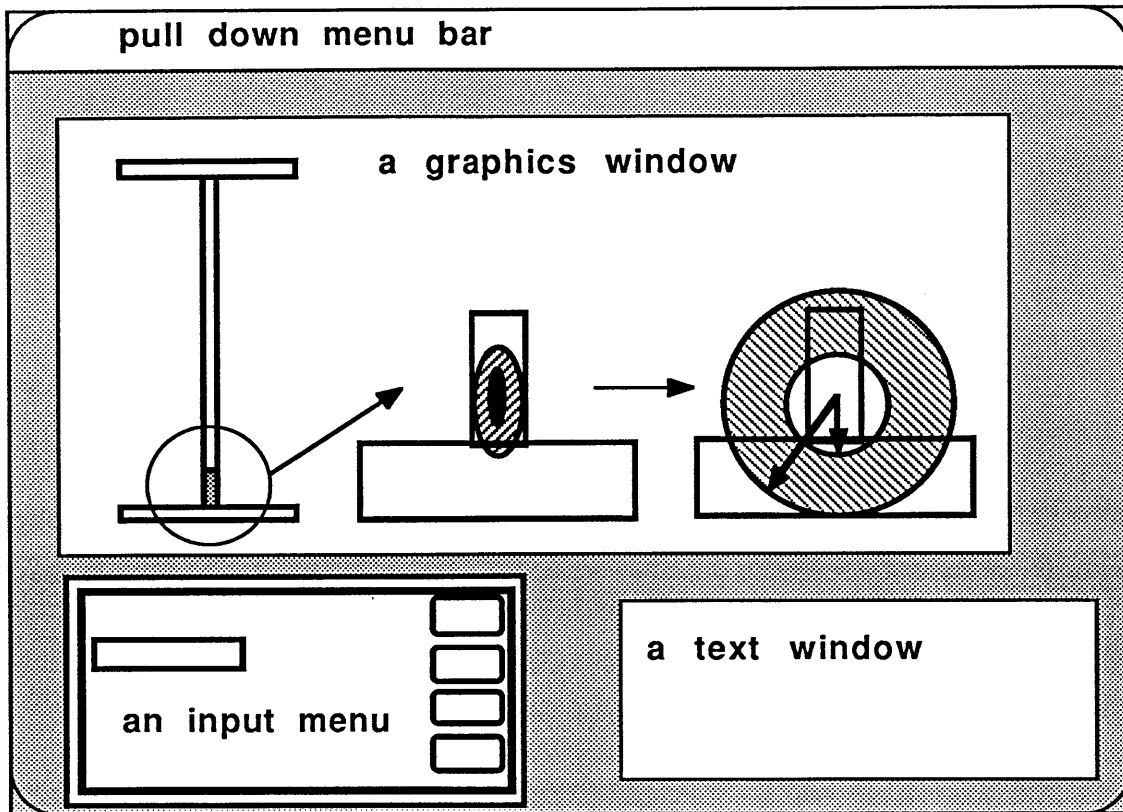


Figure 5-1:
CRACK interface

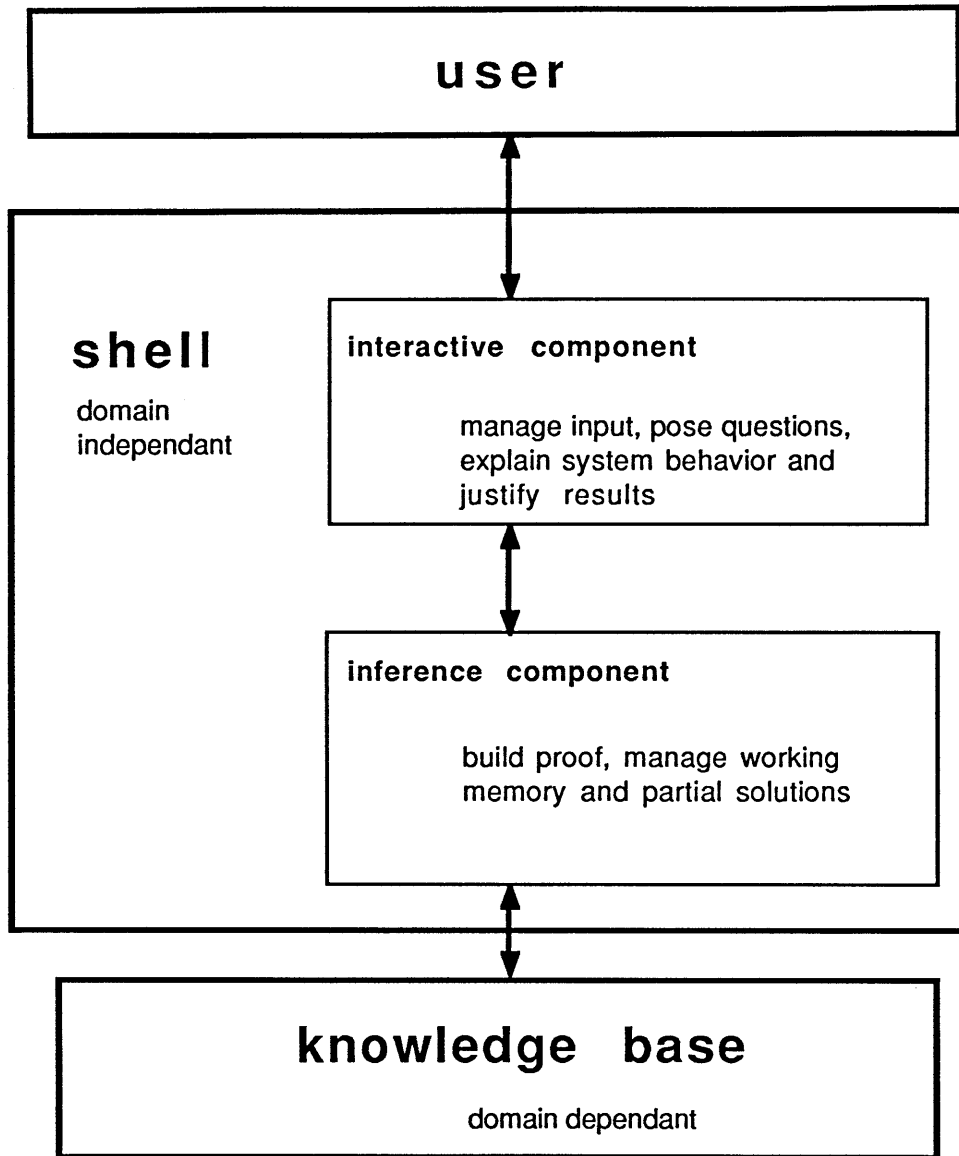


Figure 5-2:
Simple shell structure

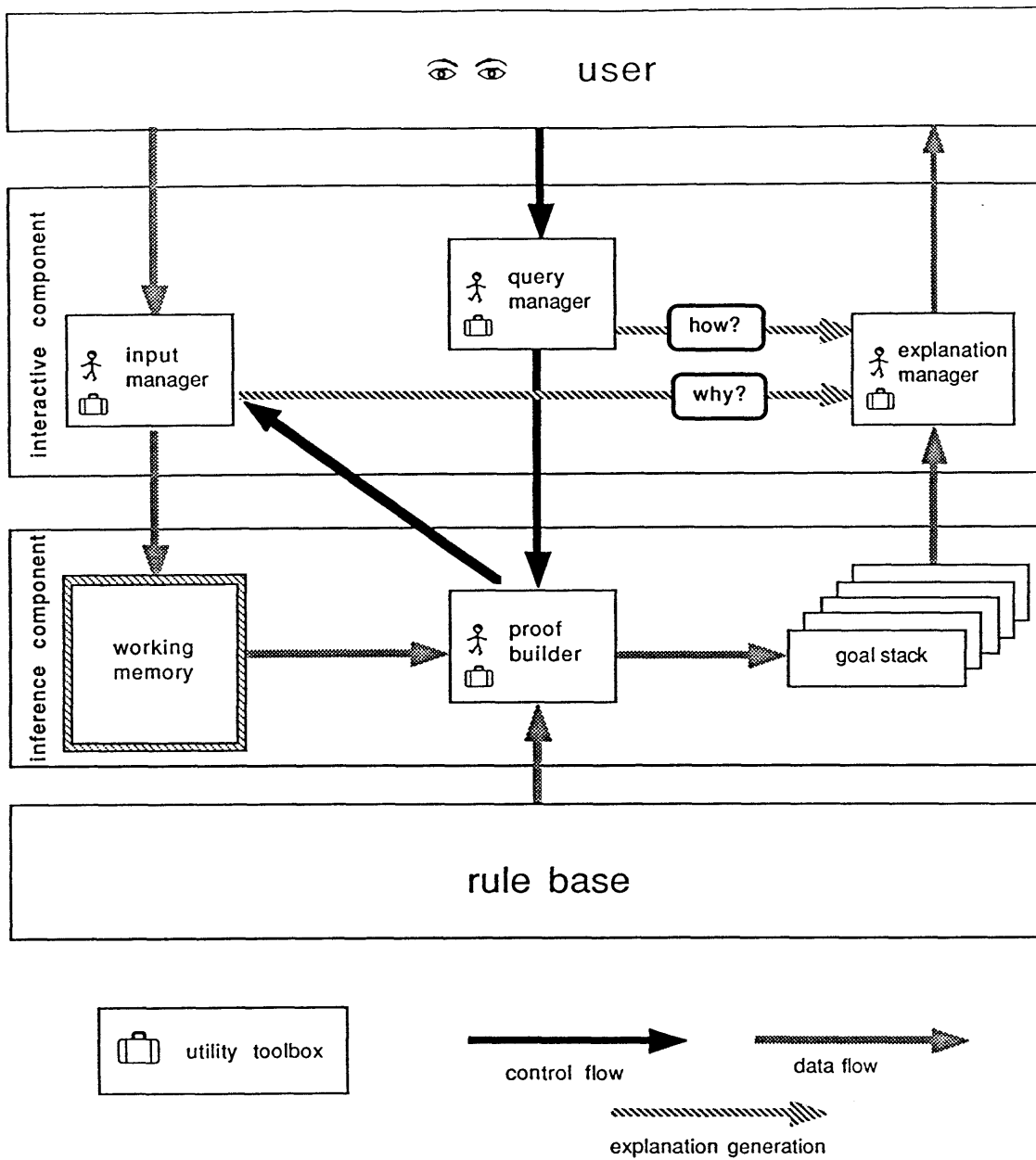



Figure 5-3:
Rule based shell for CRACK

nodes = variables
 links = influences
 = quantity spaces
 M+ = monotonically increasing
 deriv = derivative

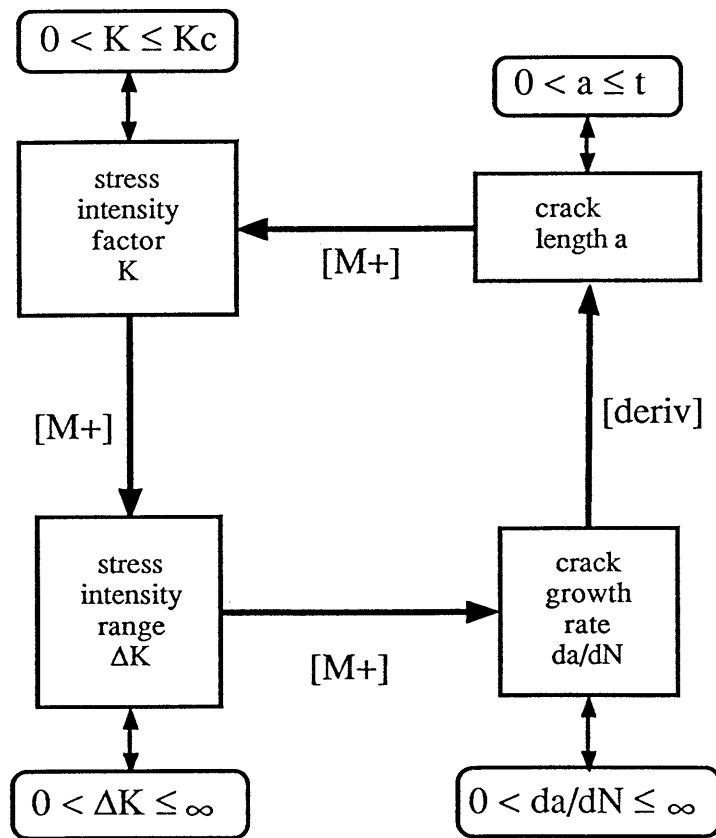
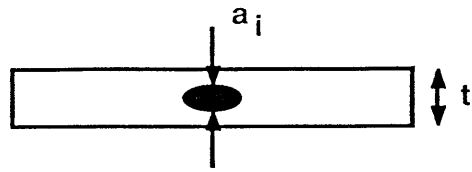


Figure 5-4:
Simple influence network

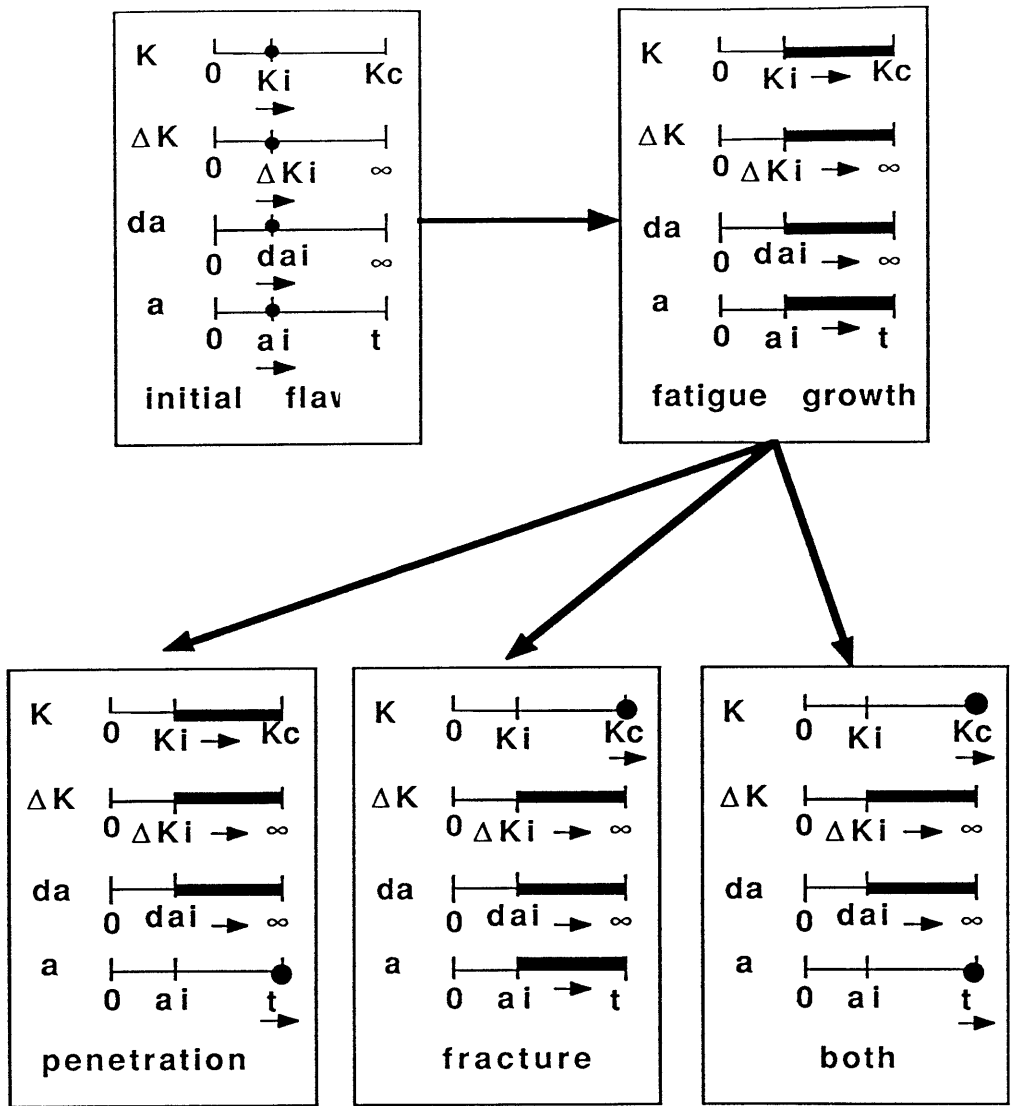


Figure 5-5:
Possible behaviors from QSIM

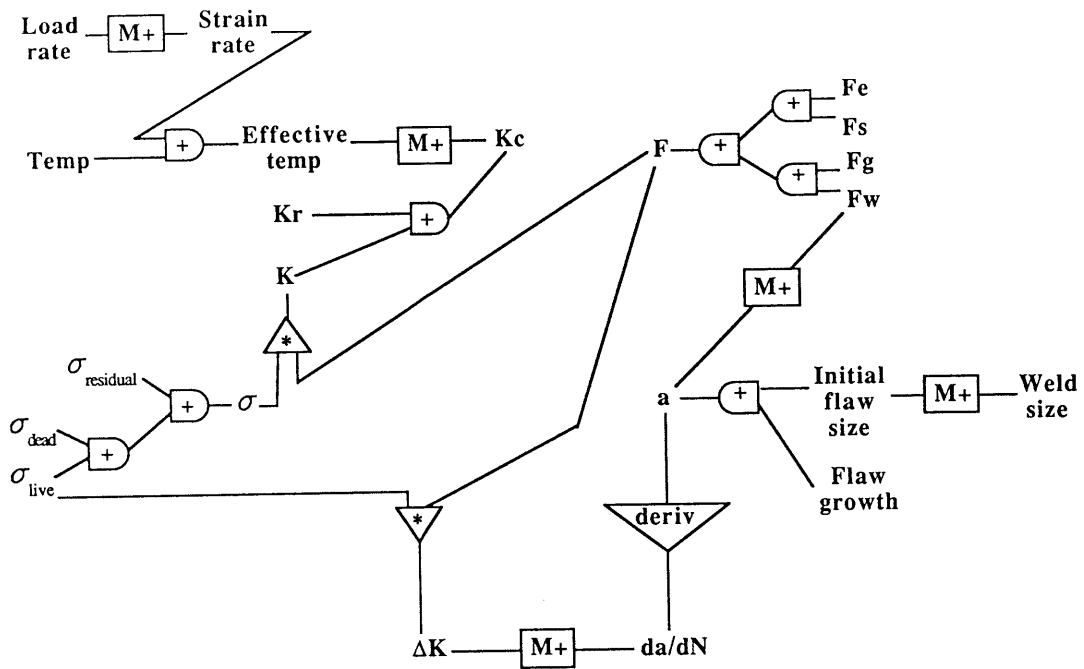


Figure 5-6:
Qualitative influence network for crack domain

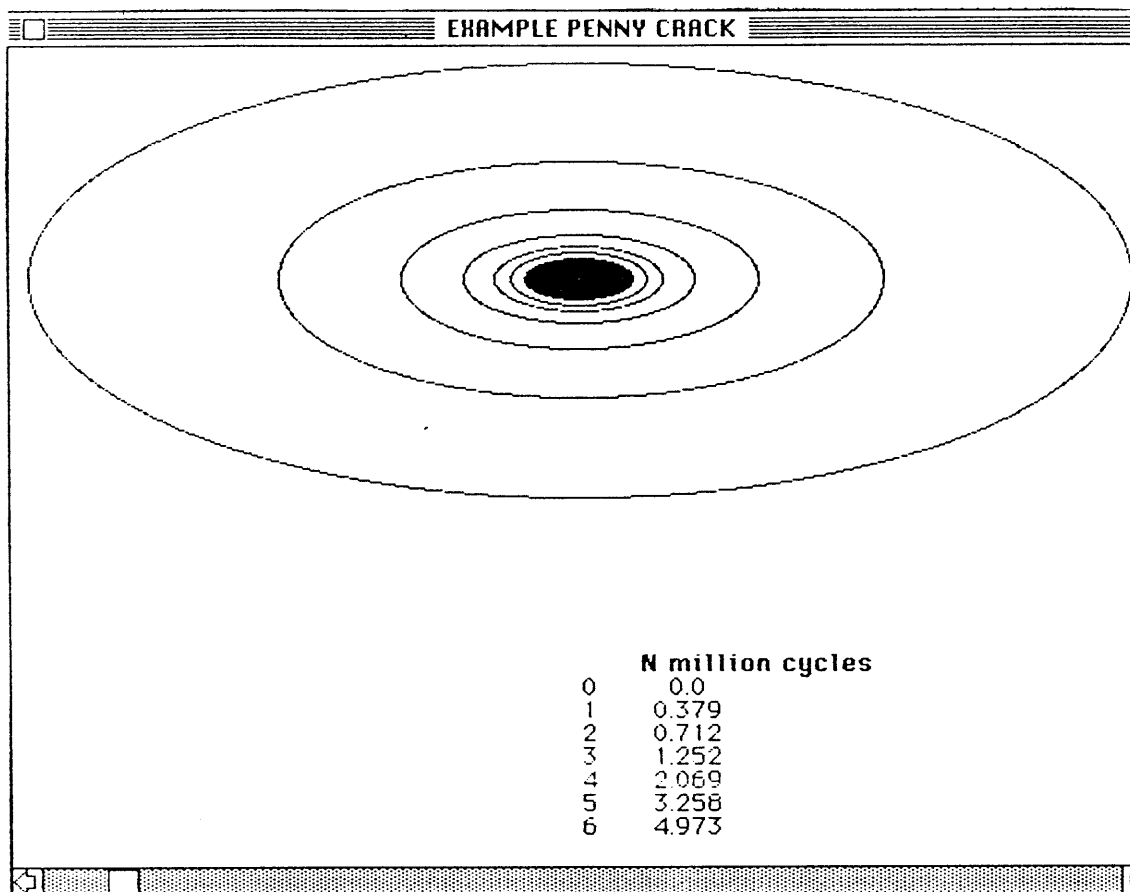


Figure 5-7:
Graphic example: penny shaped crack

Chapter 6

Detailed example: failure analysis of an actual bridge

6.1 Presentation of case study

6.1.1 Introduction to case study

Having described the computational tool CRACK in the preceding chapters, use of the program will be demonstrated by examining an actual bridge cracking failure. The case study, used here as an example to test the performance of the research prototype implementation of CRACK, is an actual case. All data are accurate, but at the request of the engineers who have generously provided the case study description, the specific bridge is not identified.

6.1.2 Description and history of the bridge

The case study bridge carries a two lane highway over two secondary roads and a river. The bridge, constructed in the late 1950's, has a 3 span continuous central structure which is flanked by rolled beam simple spans with welded coverplates. Figure 6.1 shows an elevation of the bridge. The simple approach spans are 55 and 51 feet long. The major bridge structure consists of 3 continuous spans of 150, 179.5, and 150 feet. The continuous structure is made up of two haunched welded plate girders which supports trussed floorbeams, rolled stringers, and a concrete slab. Sway bracing for the main girders is provided by rolled beam K bracing in the lower horizontal plane. The lower chord of the floorbeam truss also functions as the straight branch of the sway bracing K. Figure 6.2 shows the transverse framing of the plate girder spans. The elevation and cross framing for the bridge are shown in Photographs 6.1 and 6.2. Welded connections are used for all the floor system connections as well as for all the main girder splices, both shop and field. During the winter of 1987, a local police patrol noticed a crack in the north plate girder at the location indicated in Figure 6.1. The structure was taken out of service and shored. A detailed inspection of the superstructure and an analysis of the failure were performed. The crack was located at approximately the third point of the center span of the plate girder, as shown in Figure 6.1 and Photograph 6.3. The crack completely severed the bottom flange, ran up the full web depth, and arrested only after entering the top flange, as shown in Photographs 6.4, 6.5, and 6.6. The cross section of the plate girder at the crack location is shown in Figure 6.3.

6.1.3 In-service conditions: loads and temperatures

The most direct method of determining cyclic stresses is through field measurements. Stress histograms can be obtained from instrumentation at critical details and collection of data while the bridge is under normal traffic. Since the bridge was closed to traffic as soon as crack found, it was not possible in this case to get actual traffic data or stress range histograms. Using traffic survey data, an Average Daily Traffic (ADT) count of 7750 with truck traffic (T_{ad}) of 2% is estimated for this bridge. The T_{ad} value is low compared to national averages of 5% to 8% [AASHTO 65] for the service period of the bridge. The derived Average Daily Truck Traffic (ADTT) of 155 is reasonable as a measure of heavy trucks. Therefore the typical truck live load selected is a 36 ton gross weight Type HS AASHTO rating vehicle. Using this truck load and assuming a 20 percent increase for impact on a rough pavement surface, extreme fiber stresses were calculated. At the crack location, the stress range is 8.70 ksi. The maximum stress range of 11.1 ksi was calculated to occur in the side spans, 45 feet from the interior pier. Due to the relatively simple primary system, long spans, and distance of critical points from the interior support points, a single stress cycle is associated with each truck passage [NCHRP 87]. A weather recording station is located near the bridge. Daily low temperatures that were subzero occurred frequently in the month prior to discovery of the crack. The lowest recorded temperature was -10°F. Based on the light oxidation of the fracture surfaces initially observed, the crack probably occurred within a month of discovery. These temperatures classify the bridge as in AASHTO Zone 2 with minimum service temperatures from -1°F to -30°F [AASHTO 86].

6.1.4 Material testing

The east side of the fracture was removed for detailed examination of the failed girder. The flange, web, and weld material were tested to determine their chemical composition, microstructure, and mechanical properties. Bulk chemical composition of the flange and web was determined by laboratory analysis of coupons cut in the vicinity of the crack. The steels meet the specifications of ASTM A373 [ASTM-A373 58], a mild low carbon steel suitable for use in the construction of welded bridges. The following table summarizes the test results. The flange values are the average of two tests. The web values are from a single test. The A373 values are for ladle analysis of plate material.

	carbon C %	manganese Mn %	silicon Si%	phosphorus P %	sulfur S %
flange	.16	.65	.20	.015	.022
web	.24	.84	.05	.08	.028
A373	.26 max	.50 to .90	.15 to.30	.04 max	.05 max

In addition to the bulk chemical analyses on the flange and web plates, microchemical analyses were performed. The test was done on a 2" x 2" polished section cut 0.5" behind the fracture surface at the bottom flange to web junction. An energy dispersive attachment to a scanning electron microscope (SEM) was used for these microchemical analyses.

	manganese Mn%	silicon Si %
north weld	2.36	1.32
south weld	1.85	1.28
web	0.79	0.42
flange	0.62	0.29

The welds especially the north weld, contain high amounts of manganese and silicon. This weld chemistry indicates either the wrong shielding gas, or the wrong welding wire, or both, were used in producing the weld.

Metallographic examination of an etched and polished specimen of the flange-web junction cut 0.5" behind the fracture plane showed the microstructure of the welds, heat affected zones, and base metal flange and web plates. The flange and web exhibited an acceptable and typical as-hot-rolled microstructure of pearlite-ferrite grains. The body centered cubic structure of the ferrite and the concentration of the carbon within the carbide phase of the layered ferrite carbide structure of the pearlite result in the generally desirable characteristics of rolled steels: reasonable strength coupled with ductility and toughness at room temperature. The north weld showed a martensitic microstructure, with the characteristic needlelike appearance of the small platelike crystals. Martensite's noncubic structure with carbon present in the lattice impede slip, making it hard, strong, and brittle [VanVlack 64].

Microhardness measurements were taken on the same specimen used for metallography. The south weld has a Rockwell C hardness of 29. The north weld has an unusually high Rockwell C hardness of 36. Since martensite is much harder than ferrite-pearlite, these high hardnesses in a low carbon steel are indicative of a largely martensitic microstructure. Figure 6.4 [VanVlack 64] shows maximum hardnesses arising from martensite compared with hardnesses developed by pearlitic microstructures. The microhardness tests are thus compatible with the metallographic observations.

Tensile tests were performed on coupons cut from the flange and web in the vicinity of the crack. Tensile properties of the flange and web steels meet the requirements of ASTM A373 [ASTM-A373 58] as shown below.

	yield	ultimate tensile strength	elongation in 2"
	psi	psi	%
flange	33450	63670	37.5
web	33990	67820	32.8
ASTM A373	32000 min	58000 to 75000	24

Charpy V-Notch (CVN) bars were cut from the web and flange. The specimen orientation aligned the Charpy bar cracking parallel to the plate girder crack. Fracture critical members are defined to be those members or member components in tension whose failure would be expected to result in collapse of the bridge. For the two plate girder structure under consideration, the portion of the girder in tension is such a critical tension component. At the crack location the bottom flange and web should meet the AASHTO requirements for fracture critical members. The base metal CVN impact requirement for fracture critical members less than or equal to 1-1/2" thick in temperature Zone 2 is 25 ft-lbs at 40°F [AASHTO 86]. As shown in Figure 6.5, the flange meets this requirement but the web does not, having an estimated CVN of 20 ft-lb @ 40°F. Figure 6.6 shows the dynamic mode I fracture toughness (K_{I_d}) versus temperature data for the flange and web. The correlation equation cited in Section 4.8.3 was used to transform CVN data to K_{I_d} values. A temperature shift to account for an intermediate strain rate was not made for the following reasons. No direct data are available to estimate the fracture toughness of the weld, which is the actual physical parameter of interest. The high weld hardness does indicate that the fracture toughness would be expected to be lower in the weld than in the plates. If the crack began in the low toughness weld, a dynamic strain rate would be felt by the flange. Although using a dynamic fracture toughness is not strictly correct for a running crack [Broek 84], the small kinetic considerations due to the short run to flange entry, the non-arrest of the crack in the tougher flange, the lack of shear lips, and the absence of any better data all support the use of the dynamic fracture toughness.

The fracture toughness in both flange and web steels decreases markedly as the testing temperature falls from 75°F to 32°F to -10°F. The data display the typical

behavior of structural steels having yield strengths less than 140 ksi. There is a fairly flat region of low toughness at low temperatures, a similarly flat region of high toughness at high temperatures, and a steep transition region between these upper and lower shelves. Failure in the low toughness region is brittle, plain strain, with little plastic deformation, and a flat fracture surface. Failure in the high toughness region is ductile, preceded by large plastic deformations, and has the fracture surface inclined at the edges (shear lips). The highest temperature at which a standard specimen breaks in a brittle manner is called the nil-ductility transition (NDT) temperature of the material. For structural steels, the NDT temperature is usually between -10°F and $+10^{\circ}\text{F}$ [Barsom&Rolfe 87]. Figure 6.6 shows that the steels of the flange and web plates were both below their NDT temperatures during the subzero weather at the time of the failure.

6.1.5 Fractography

The crack surface was examined macroscopically and microscopically for fracture features. Macroscopic examination of the fracture surfaces identified clear chevron markings on both the web and lower flange, pointing to the lower web-flange junction. For brittle flat fractures, the direction of crack growth is almost always away from the tips of the chevrons [Ryder et al. 85]. The chevron markings thus point to a failure origin at the web-flange intersection. Figure 6.7 sketches the fracture features in this area. The converging chevron markings indicate an origin in the web to flange fillet weld area. The north weld has a visible discontinuity corresponding to a weld fold. This weld flaw contains some residual red primer paint. The presence of the paint indicates that this transverse weld notch of about 0.1" depth was part of the initially fabricated weld, existing prior to crack initiation. The immediate region of the north weld is heavily oxidized. Low magnification of the surrounding area shows some beach marks. These marks, also called progression or clamshell marks, indicate successive positions of the advancing crack front [Ryder et al. 85]. The beach marks indicate fatigue growth of the initial flaw to a depth of about 0.5" followed by brittle fracture. The beach marks end within the heat affected zone of the north weld. High magnification SEM examination after oxide removal confirmed this mixed mode of fatigue and fast fracture propagation through the weld. Detailed fracture features were obliterated by corrosion of the fracture surfaces.

The fracture surface outside of the beach marked area displays the rough texture and chevron markings characteristic of brittle fracture. No shear lips appear on the edges

of web and flange plates. The absence of shear lips is indicative of plane strain fracture [Fuchs&Stephens 80]. Low strength ductile materials, such as mild structural steels, are subject to plain strain fracture at temperatures below the material's nil-ductility transition (NDT) temperature [Barsom&Rolfe 87].

6.1.6 Analysis of evidence

An explanation of the cause of cracking and the sequence leading to failure can be derived from the information gathered on service conditions, materials, and fractographic features. The fractographic evidence identifies the small weld defect transverse to the north fillet weld between the lower flange and web. The weld defect, marked with paint probably from the time of fabrication, is the crack initiation site. The beach marks delineate the extent of crack propagation by fatigue. The rough texture, chevron markings, and absence of shear lips indicate the crack grew by fast fracture. The very light oxidation on the fracture surfaces at the time of discovery show that the crack occurred during the month of subzero temperatures. The absence of shear lips indicates that the steel was below its NDT temperature, thus exhibiting low toughness resulting in brittle fracture. The sequence of cracking stages is thus an initial flaw of 0.1", fatigue growth to a depth of 0.5", followed by cleavage of the entire lower flange and web plates.

The high residual tensile stresses in the weld area made the full stress variation effective for driving crack growth during the fatigue stage. The reason for the small (0.5") critical crack size is due to the low toughness of the plate girder materials. The north weld had an unusually high hardness due to the high manganese and silicon contents because Mn and Si increase the hardenability of steel. During rapid cooling of the weld material, presence of these elements encourage transformation of the austenitic microstructure to the hard and low toughness microstructure of martensite [VanVlack 64]. The as fabricated plate girder had hard welds with low toughness due to improper welding procedures resulting in unsatisfactory weld chemistry. The CVN data supports the fractographic observations showing brittle cleavage behavior in web and flange. Assuming a failure temperature slightly below 0°F, the flange fracture toughness would be approximately 55 ksi $\sqrt{\text{in}}$ for CVN in the range of 20 ft-lbs.

Thus on a cold day the critical crack size under live loads could reasonably be expected to be in the half inch size range. The bulk of the evidence supports this explanation of an initial flaw growing modestly by fatigue until the simultaneous occur-

rence of low temperature and live load lead to a brittle fracture typical of low carbon steels below the NDT temperature.

6.1.7 Repair versus replacement

The failure investigation is not complete after the determination of the cause and progress of the observed crack. It is necessary to evaluate the salvageability of the existing structure. Given the occurrence of one major cracking failure of the plate girder, is it technically and economically feasible to repair the existing superstructure and return it to service, or is replacement necessary?

The detailed inspection of the plate girder spans revealed numerous weld defects and fatigue susceptible details in the girders and primary connections. The bridge, like many built during the 50's and 60's at the transition from bolted to welded construction has many details at which cracking can initiate. The field welds are of generally poor quality, exhibiting rough surface texture, undercut, improper returns where welds intersect at corners, pinholes, and inclusions. The shop welds, although apparently machine welded, have areas with pinholes and undercutting. Other weld flaws similar to initiating flaw were found during detailed inspection of the superstructure. Dye penetrant testing of north girder weld revealing transverse hairline cracks along the exterior fillet weld of the bottom flange to web connection.

Fatigue life calculations were made using the common method [AASHTO 85] based on empirical curves for structural detail categories. Of the many fatigue susceptible details in the superstructure, the one identified to have the most severe stress categorization is the panel point connection detail of the horizontal K sway bracing shown in Figure 6.8. The diagonal bracing beams were fillet welded to the upper surface of the bottom plate girder flange. This long ($> 4"$) surface attachment to the flange surface creates a highly stressed weld toe which, like the condition at the end weld of a coverplate, is very susceptible to fatigue. Fatigue category E is thus appropriate for evaluation of this detail.

Based on the in-service conditions given in Section 6.1.3 and current AASHTO Stress range versus Number of cycle curves [AASHTO 85], the fatigue life calculated for the maximum stressed K-brace detail is less than 3 years. For the lower stressed K-brace detail adjacent to the cracked section, the fatigue life is 5 years. Using the proposed new

Stress range versus Number of cycle curves [NCHRP 87] the corresponding fatigue lives are 11 and 21 years. These calculated fatigue lives are all less than the bridge's service life of 28 years.

Given the small critical crack size of the observed failure, the discovery of numerous other weld flaws, the demonstrated low fracture toughness of the bridge materials, repair of the superstructure was not judged to be feasible. Since it was not possible to assure adequate residual strength in the superstructure, no attempt was made to return it to service. A single hidden flaw could lead to a recurrence of major cracking. Therefore, the plate girder bridge was removed and replaced with a new superstructure on the existing piers.

6.2 Application of CRACK to case study

6.2.1 Suitability of case study for CRACK

The case study delineated gives a complete set of information about a cracking failure in a highway bridge. The failure analysis presented in Section 6.1 did not include a fracture mechanics evaluation to analytically relate the physical parameters governing cracking behavior: initial flaw, service history, material properties, and structural configuration. The limited fatigue calculations performed were based on an inherently empirical approach and did not address the actual failure cross section, but a different welded detail not observed to have cracking distress. Fracture mechanics can quantify the relationships between service life and fatigue crack growth and between observed critical crack size and material fracture toughness. Application of CRACK to this case study thus adds a fracture mechanics component to the previous failure study.

The availability of the case study data also offers the opportunity to test the ability of CRACK to generate an explanation for the cracking behavior that is compatible with Section 6.1.6. The performance of CRACK thus serves to validate the rule base against a realistic example not gleaned from the literature.

Finally, the case is a fairly simple one, involving only a few stages of crack propagation at a plate girder section with no attachments. An examination of the operation of CRACK on the is case is thus an uncluttered pedagogic example.

6.2.2 CRACK problem solving session

To illustrate the operation of CRACK, the steps that the system executes in the case study session are described in sequence below. The phases in the solution process, as discussed in Section 5.4, are:

1. Establishing type of problem to be solved
2. Describing the problem by gathering information about:
 - the girder geometry
 - service history
 - material properties
 - and observed crack
3. Hypothesizing crack cause
4. Qualitatively simulating possible crack progression sequences
5. Computing critical crack sizes and number of load cycles
6. Evaluation of the hypothesis as most probable crack cause and development
7. Stating conclusions

6.2.3 Failure analysis mode

The first question asked in any consultation with CRACK is what type of problem is to be solved. The case study at hand is a failure analysis type of problem. The solution sought is an explanation of the failure which matches a crack progression sequence to the observed facts.

6.2.4 User supplied information

For the case study, the structural configuration is specified to be:
top & bottom flanges: 24" x 1-1/8" A373 plates
web: 92" x 5/8" A373 plate
top & bottom flange-web connections: 7/16" fillet welds, both sides
no attachments

Facts concerning the bridge's service history are requested next. To find the number of years of service, the start and end dates of bridge use are needed. The bridge was put into service in 1958 and removed from service in 1987 for a service age of 29 years. Stress range histograms are not available, so the number and range of stress cycles

must be derived. The average daily truck traffic is computed from the user specified values of 7750 Average Daily Traffic and Truck Amount at 2% sets the Average Daily Truck Traffic at 155. The number of cycles per truck is founded heuristically, based on the user supplied facts (the cross-section is in a continuous span plate girder more than 1/10 of the span from an interior support). The stress range (8.70 ksi) on the extreme fibers of the cross section and the minimum service temperature (-10°F), which is zone 2, are provided by the user.

Material properties are requested next. If no tests had been performed, CRACK would look up material property values for A373 steel. The values requested are yield strength (33.5 ksi flange, 34.0 ksi web) and average CVN impact energy at the minimum service temperature (6.78 ft-lbs flange, 5.48 ft-lbs web, both at -10°F). The fracture toughness is then calculated (31.4 ksi $\sqrt{\text{in.}}$ flange, 28.2 ksi $\sqrt{\text{in.}}$ web).

The final group of information requested is the crack extent and surface features. The crack completely severs the bottom flange and web, but not the top flange. CRACK knowledge of fractographic features is limited to chevron markings, shear lips, beach marks, and fabrication flaws. The presence of chevrons on right and left portions of the bottom flange and the web, all pointing toward the lower weld, and lack of shear lips are noted. A 0.5" radius, thumbnail shaped, beach marked region is centered at 1.40" above the bottom flange and 0.93" to the left of the web center line. A 0.1" radius , thumbnail shaped flaw has the same center. This completes the information gathering phase.

6.2.5 Forming hypotheses

At this point, a model can be postulated. The knowledge base contains a rule set to hypothesize the crack cause based on the user supplied facts describing the plate girder geometry, service, materials, and cracking. Using this rule set, the likely origins of transverse cracking in a welded plate girder without attachments are [NCHRP102 70]:

1. a weld flaw in the tension flange-web connection
2. a weld flaw in the compression flange-web connection
3. a notch in the flame-cut edge of the flange tip

These hypotheses are ordered from most to least likely. CRACK operates by trying to explain the observed failure facts by assuming the most likely hypothesis. Only if the first hypothesis predicts behavior which does not match the facts will it be discarded. The next hypothesis would then be tried in turn.

From laboratory observations of fatigue cracks in welded plate girder without attachments, most cracks originate at the welded flange-web junction [Roberts et al. 77]. The most common fatigue crack growth from internal discontinuity. Crack origin in web-flange fillet weld at internal discontinuity such as porosity, lack of fusion, or trapped slag. Residual stresses from welding results in tensile residual stresses at or near the yield point. Due to these high residual stresses, the full stress range in a stress cycle is effective in driving crack growth, whether the contribution is tension or compression. For fatigue cracks in welded built-up details which originate at internal discontinuities, the crack front grows in a penny shape until it penetrates an outer flange surface of the welded built-up component. At the fillet welded flange-web joint, the longitudinal lack of fusion discontinuity between the parallel fillet welds is oriented parallel to the stress field and has no effect on the shape or fatigue crack growth behavior.

The facts of the case support the first hypothesized crack origin of a weld flaw in the tension flange-web connection. The crack passes through this weld, the chevron markings point to this area, and a weld flaw was noted in the north fillet. Since fractographic information is available in this case study, the size, shape, and location of the suspected crack origin are all quantified.

6.2.6 Construction of parametric study

Qualitative simulation is used to enumerate the possible ways the crack can grow. This particular case study yields a simple set of possible states, due to the uncluttered geometry and the large number of parameters that may be assigned fixed values based on the extensive user supplied facts.

The initial stage is the thumbnail shaped flaw in the flange-web junction (stage 0). There are three possible crack sequences. First the initial flaw can grow by fatigue (stage 1) to a critical size and fail by fast fracture (stage 2). Second the initial flaw can grow by fatigue until the flange is penetrated (stage 3) at which time the crack changes to a through crack which grows by fatigue (stage 4) to a critical size and fails by fast fracture (stage 5). Third the initial flaw could behave as in the second case except before a critical size is reached the flange fails due to loss of section (stage 6). Figure 6-11 summarizes these three paths generated by the qualitative reasoning. Note that the physically redundant states of simultaneous fracture and penetration (states 2 and 3) and fracture

and loss of section (states 6 and 7) have been eliminated from this summary.

6.2.7 Fracture mechanics calculations

The results of the qualitative simulation and information about the suspected originating flaw are used to set up the analytic model simulating crack growth. The appropriate model for a surface weld flaw at the flange-web junction is an edge crack. (Although the crack front is shaped like a thumbnail crack for a surface flaw or like a penny crack for an embedded flaw, the constraint conditions for cracks in flange-web junction are more closely matched by two dimensional rather than three dimensional models as can be seen in the example of a crack emanating from a spherical weld porosity in [Albrecht&Yamada 77]). The initial flaw size is 0.1", the stress range is 8.7 ksi, and the maximum stress is at yield (33.5 ksi for flange) due to high residual stresses at the weld. For the case study, the first path branch occurs at the choice between stage 2, fracture, and stage 3, fatigue growth through the flange. The flange's fracture toughness is 31.4 ksi√in. Penetration is defined as fatigue growth through 90% of the thickness of the plate, in this case at a crack depth of 1.26". Figure 6-9 shows the simulated crack growth. The first line in the table corresponds to stage 0, the initial flaw. Lines 1 to 4 correspond to stage 1 fatigue growth for path 1, with line 4 corresponding to stage 2 fracture when the stress intensity factor exceeds 31.4 ksi√in. Paths 2 and 3 are eliminated from consideration because stage 2 fracture occurs before line 7, corresponding to stage 3 flange penetration by fatigue growth. Using the expressions for stress intensity factor, stress intensity range, and crack growth rate given in Chapter 4:

$$\begin{aligned}
 F_{(a)} &= F_e * F_s * F_w * F_g \\
 K &= F_{(a)} \beta * \sqrt{(\pi * a)} \\
 \Delta K &= F_{(a)} \Delta \beta * \sqrt{(\pi * a)} \\
 da/dN &= 3.6 * 10^{-10} * \Delta K^3
 \end{aligned}$$

$$\begin{aligned}
 F_{(a)line1} &= 1.0 * 1.12 * 1.003 * 1.0 &= 1.113 \\
 K_{line1} &= 1.113 * 33.5 * \sqrt{(\pi * 0.1)} &= 21.1 \text{ ksi}\sqrt{\text{in}} \\
 \Delta K_{line1} &= 1.113 * 8.7 * \sqrt{(\pi * 0.1)} &= 5.4 \text{ ksi}\sqrt{\text{in}} \\
 (da/dN)_{line1} &= 3.6 * 10^{-10} * 5.479^3 &= 5.92 * 10^{-8} \text{ in/cycle} \\
 F_{(a)line4} &= 1.0 * 1.12 * 1.019 * 1.0 &= 1.141
 \end{aligned}$$

$$\begin{aligned}
K_{\text{line4}} &= 1.113 * 33.5 * \sqrt{(\pi * 0.245)} = 33.6 \text{ ksi}\sqrt{\text{in}} \\
\Delta K_{\text{line4}} &= 1.113 * 8.7 * \sqrt{(\pi * 0.245)} = 8.7 \text{ ksi}\sqrt{\text{in}} \\
(da/dN)_{\text{line4}} &= 3.6 * 10^{-10} * 8.714^3 = 2.38 * 10^{-7} \text{ in/cycle}
\end{aligned}$$

6.2.8 Evaluation of hypothesis

Having simulated cracking behavior based on the chosen hypothesis, the next phase is to compare the predicted and observed facts. If the predictions match the observations, the hypothesis is a good one. If the predictions contradict the evidence, the hypothesis must be discarded and the next one must be evaluated. This cycle continues until a satisfactory match is made. If the hypothesis list is exhausted, CRACK reports its failure to generate a solution.

The numeric results are compared to the observed facts as described in Section 5.3. The simulated crack growth predicts a fatigue life of 1.2 million cycles which closely matches the estimated service life of 1.6 million cycles. The predicted critical crack size of 0.25" is in reasonable correspondence with the 0.5" beach marked area. The prediction of brittle fracture is supported by the presence of chevron markings. The comparison between predicted and observed facts is thus quite good so the hypothesis is judged to be correct.

6.2.9 Statement of conclusions

The only task remaining is to summarize the results and concisely state the conclusions. A text summary of the results is generated by using template sentences and filling in the blanks with the appropriate user supplied or inferred values, as shown below.

Conclusions for CRACK consultation on Case Study bridge.

Task to be done is failure analysis.

The girder top flange is a plate of ASTM a373, with a fracture toughness (ksi√inch) of 31.4 and a yield strength (ksi) of 33.5, 24 inches wide and

1.125 inches thick.

The girder web is a plate of ASTM a373, with a fracture toughness (ksi $\sqrt{\text{inch}}$) of 28.2 and a yield strength (ksi) of 34, 92 inches wide and 0.625 inches thick.

The girder bottom flange is a plate of ASTM a373, with a fracture toughness (ksi $\sqrt{\text{inch}}$) of 31.4 and a yield strength (ksi) of 33.5, 24 inches wide and 1.125 inches thick.

There is a welded 0.438 inches top flange connection.

There is a welded 0.438 inches bottom flange connection.

There are no attachments.

The in-service stress range is 8.7 ksi at a frequency of 155 Average Daily Truck Traffic for a bridge age of 29 years.

The cracking cause is classified as a initial flaw in a welded bottom flange connection type of cracking.

The initial flaw size is 0.1 inches. It is edge shaped in the bottom flange connection.

The probable sequence of crack growth to failure is:

1. fatigue growth as edge crack in bottom flange connection from a crack size of 0.1 inches to 0.228 inches in 1.152 million cycles.
2. fracture as edge crack in bottom flange connection at a crack size of 0.228 inches at a stress intensity factor of 31.765 ksi $\sqrt{\text{in}}$.

The predicted fatigue life of 1.152 million cycles is in good agreement with the estimated service life of 1.641 million cycles.

The predicted final crack size of 0.228 inches is in reasonable agreement with the beach marked size of 0.5 inches.

The presence of chevron markings is in agreement with the prediction of fracture failure.

The absence of shear lips in conjunction with the prediction of fracture failure is indicative of fracture of structural steel below the nil-ductility temperature.

6.3 Critique of CRACK performance on case study

CRACK's explanation for the failure is indeed compatible with Section 6.1.6. This is an important demonstration that CRACK is capable of generating a satisfactory analysis using an actual case different from those used to develop the knowledge base. The fracture mechanics analysis resulted in fatigue life and critical crack size predictions that satisfactorily matched the case study evidence. To understand the relative importance of the major parameters, the effect of changes in fracture toughness, initial crack size, and stress range on the predicted fatigue life and critical crack size are examined.

The effect of changes in fracture toughness and initial crack size may be inferred from Figure 6-9. Increasing the fracture toughness by a factor of two (62.8 ksi $\sqrt{\text{in.}}$) would increase the fatigue life by less than a factor of two (1.9 million cycles) while the critical crack size is more than doubled (0.66"). In contrast, doubling the initial crack size reduces the fatigue life to a very short period (0.1 million cycles) since very little fatigue growth occurs before the critical crack size is reached. In general, the bulk of the fatigue life is spent when the crack is small. Increasing the initial flaw size will strongly reduce the fatigue life. Doubling the stress range would have a profound effect on the fatigue life, as shown in Figure 6-10. Fatigue life predictions are thus heavily dependent on initial crack size and stress range. Since both these parameters are not known with much precision, calculated fatigue lives are highly variable. Changes in fracture toughness increase critical crack sizes but do not have a strong influence on increasing fatigue life.

Given the imprecision of the driving parameters, CRACK's simplified approach to the fracture mechanics analysis is clearly justified. Refining the analytic techniques cannot give great exactness given the uncertainty of the fracture toughness, initial crack size, and stress range. As described in Chapter 4, CRACK assumes: no cycles are spent in crack initiation; the crack growth rate is temperature independent, the stress intensity factors can be approximated by the superposition method; and small scale yielding allows linear elastic fracture mechanics. The final assumption can be directly checked. The critical crack tip plastic zone size is computed to be smaller than 0.14" [Anand&Parks 86]:

$$r_c = (K_{I_d} / Y)^2 / (2 \pi)$$

where:

K_{Id}	= fracture toughness, dynamic mode I loading 31.4 ksi $\sqrt{\text{in}}$ for flange
Y	= yield stress = 33.5 ksi for flange

To assure small scale yielding the plastic zone size should be much smaller than all plate dimensions and the critical crack size. The plastic zone is indeed much smaller than the thicknesses of any plate dimension. The critical crack size is only 1.5 times larger than the plastic zone. However, the actual plastic zone size is smaller than 0.14" in the welded region, since the toughness of the weld material is less than the toughness of the flange. For this reasons, although the small scale yielding conditions are not satisfied to an extent that would give highly accurate results using linear elastic fracture mechanics, LEFM can still be used to give a first cut approximation to predict crack behavior.

CRACK does not contain any knowledge about chemical composition and only limited other knowledge of material properties. For this reason, Section 6.2.9 does not mention the unsatisfactory weld chemistry, high hardnesses, or martensitic microstructure. All this information is only represented in the low fracture toughness computed from CVN data. The low fracture toughness and lack of shear lips are the two very significant facts in this case study. The bridge failure occurred due to inadequate material toughness at low service temperatures. Prevention of this type of failure is complicated. Many bridges built during the 1950's through 1970's have steels with a nil-ductility temperature above their minimum service temperature. There is no convenient field test to measure fracture toughness. Chemistry and hardness data cannot be used to derive fracture toughness. CVN tests cannot be used on in-service bridges except in a vary limited way since the tests are destructive and relatively costly. This case study highlights the difficulty of preventing brittle failures in plate girder bridges with unknown toughness and probable critical crack sizes too small to be reliably found during standard inspections.

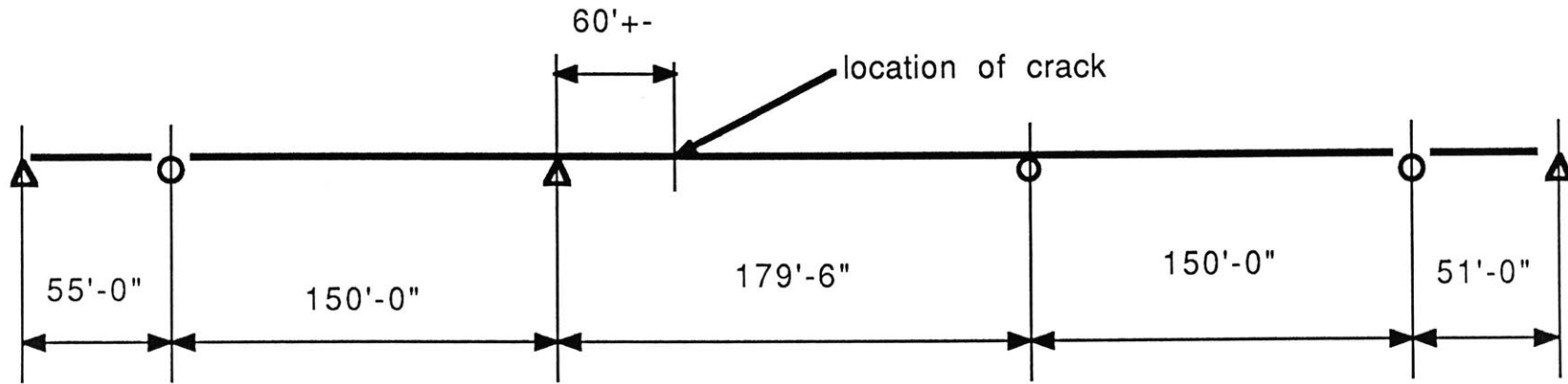
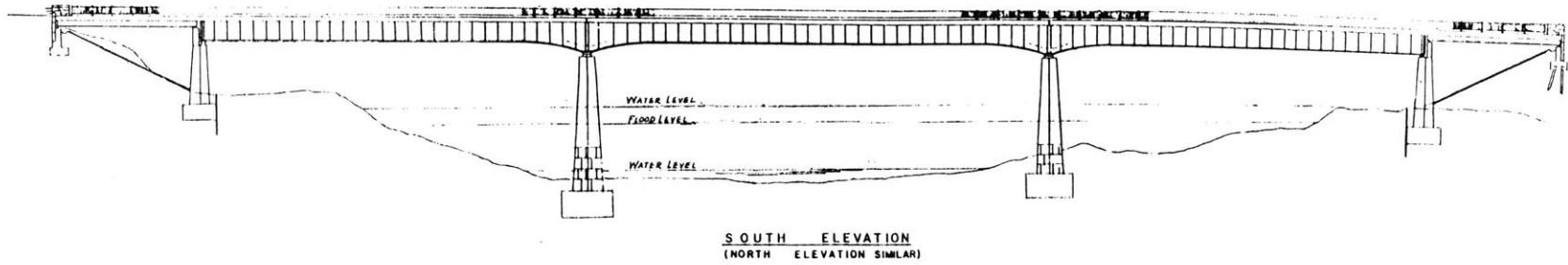


Figure 6.1: Elevation: Case Study Bridge

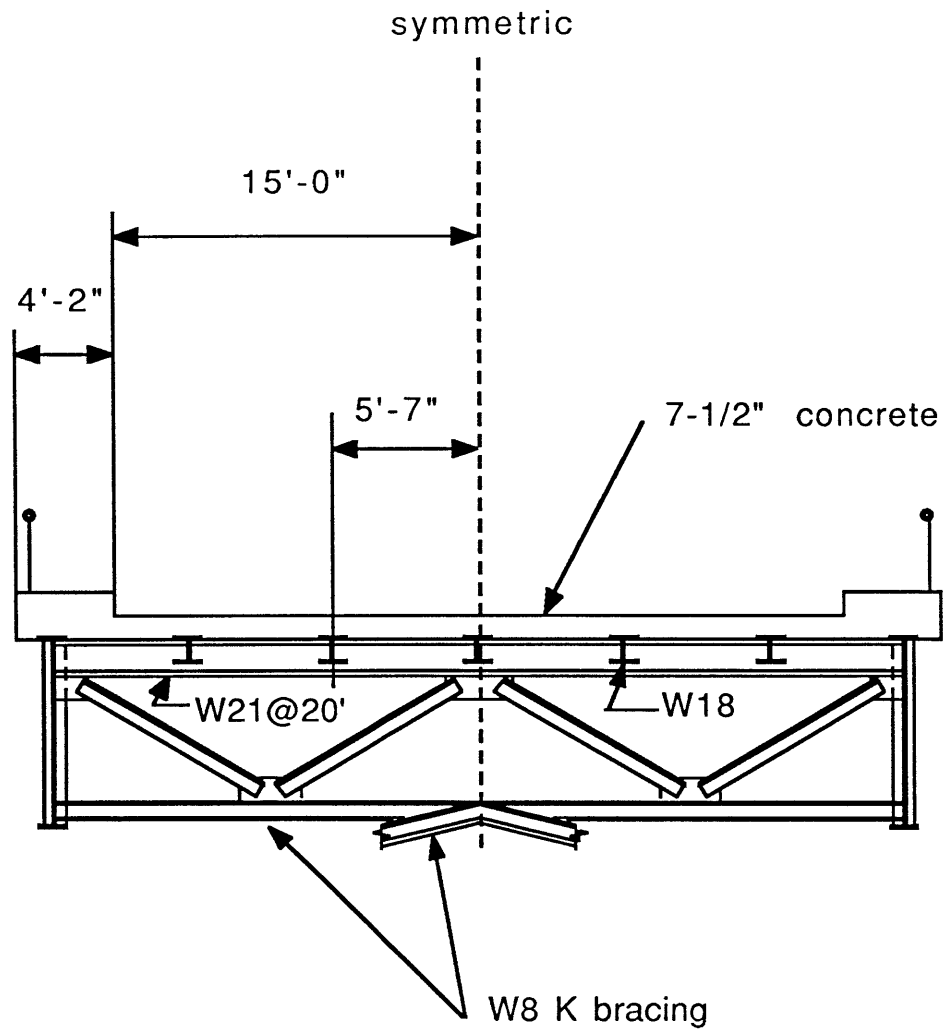
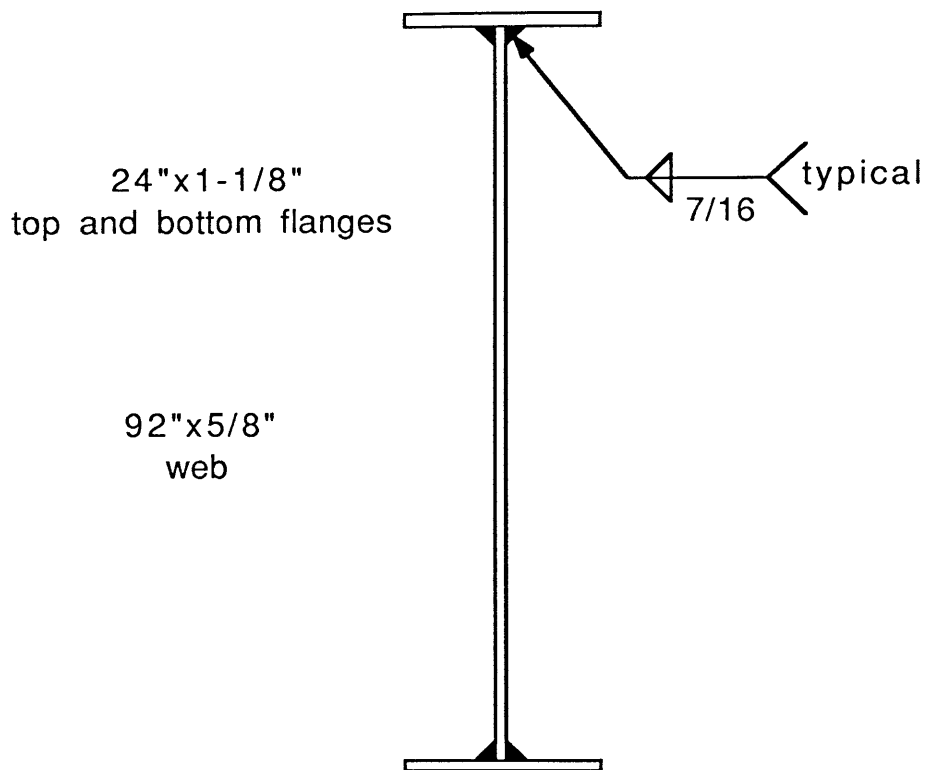


Figure 6-2:
Cross Section: Case Study Bridge



All Welding and Welding
Materials shall Conform to the
Current Specifications for
Welded Highway and Railway Bridges
of the American Welding Society

Figure 6-3:
Plate Girder Failure Section: Case Study Bridge

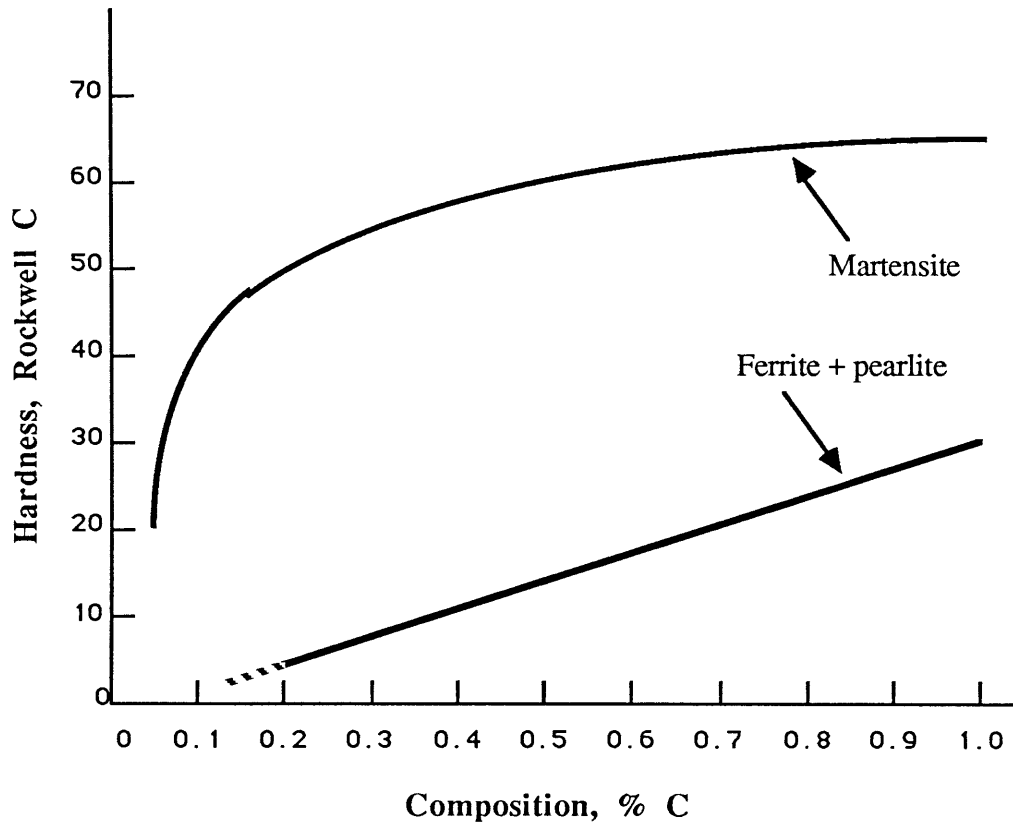


Figure 6-4:
Maximum hardness versus carbon content of plain carbon steels
 [from Van Vlack 64]

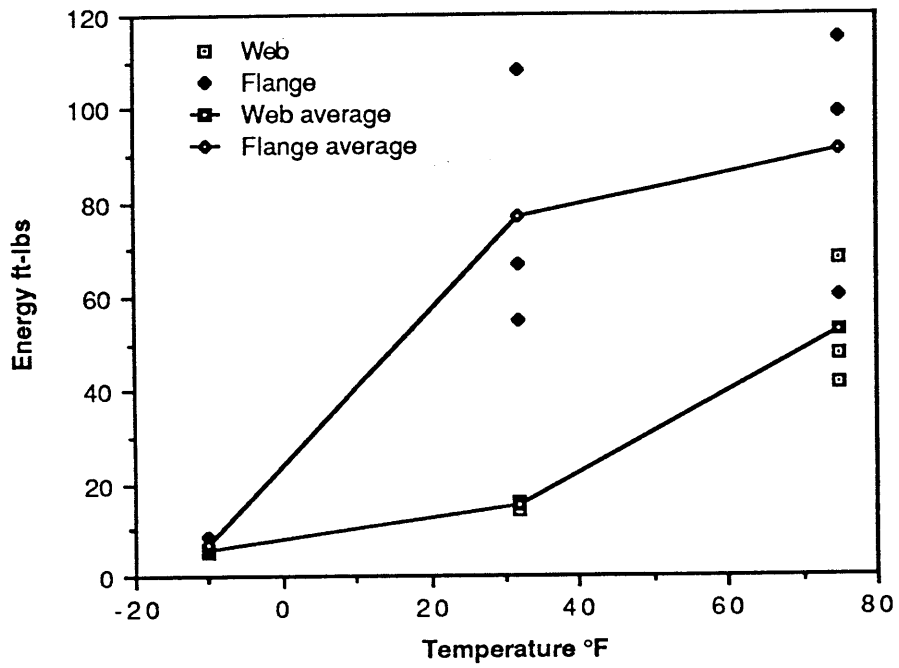


Figure 6-5:
Charpy V-Notch impact test results

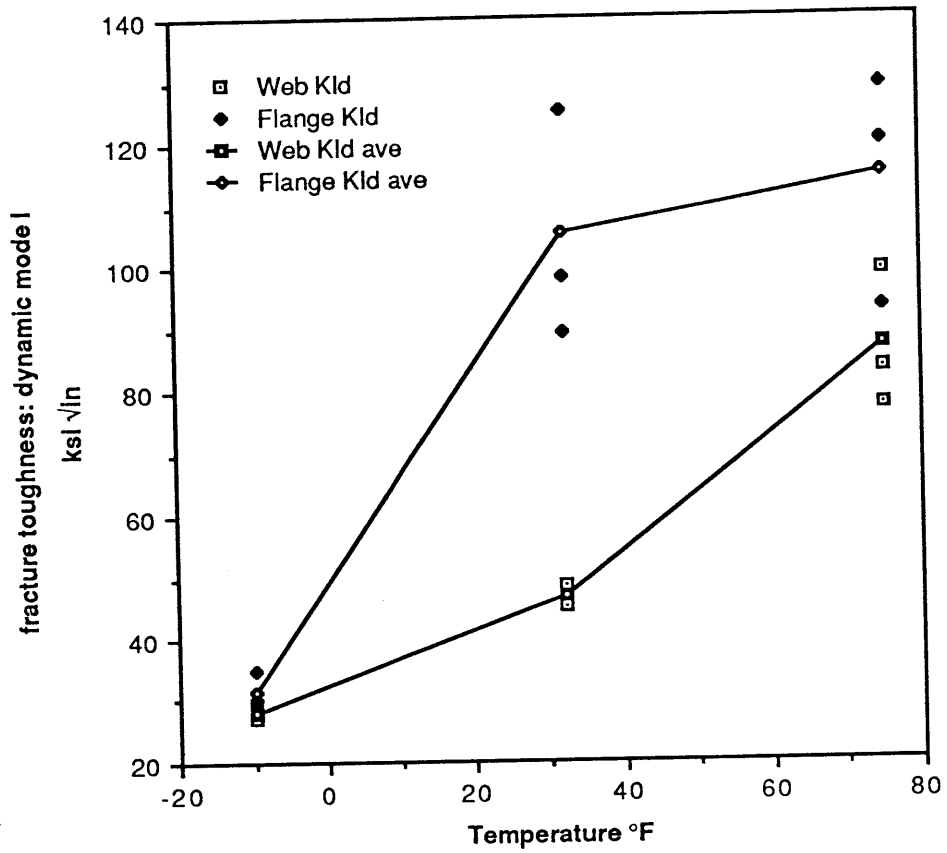


Figure 6-6:
Dynamic mode I fracture toughness KId

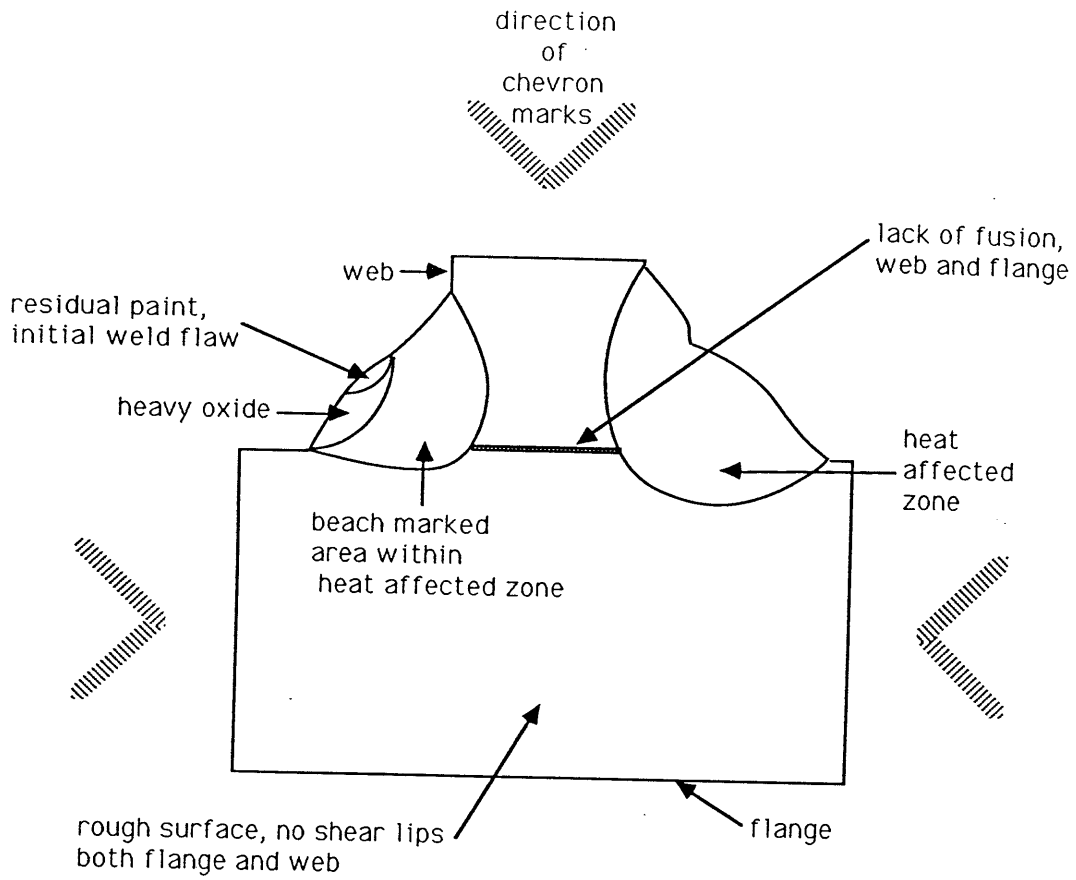
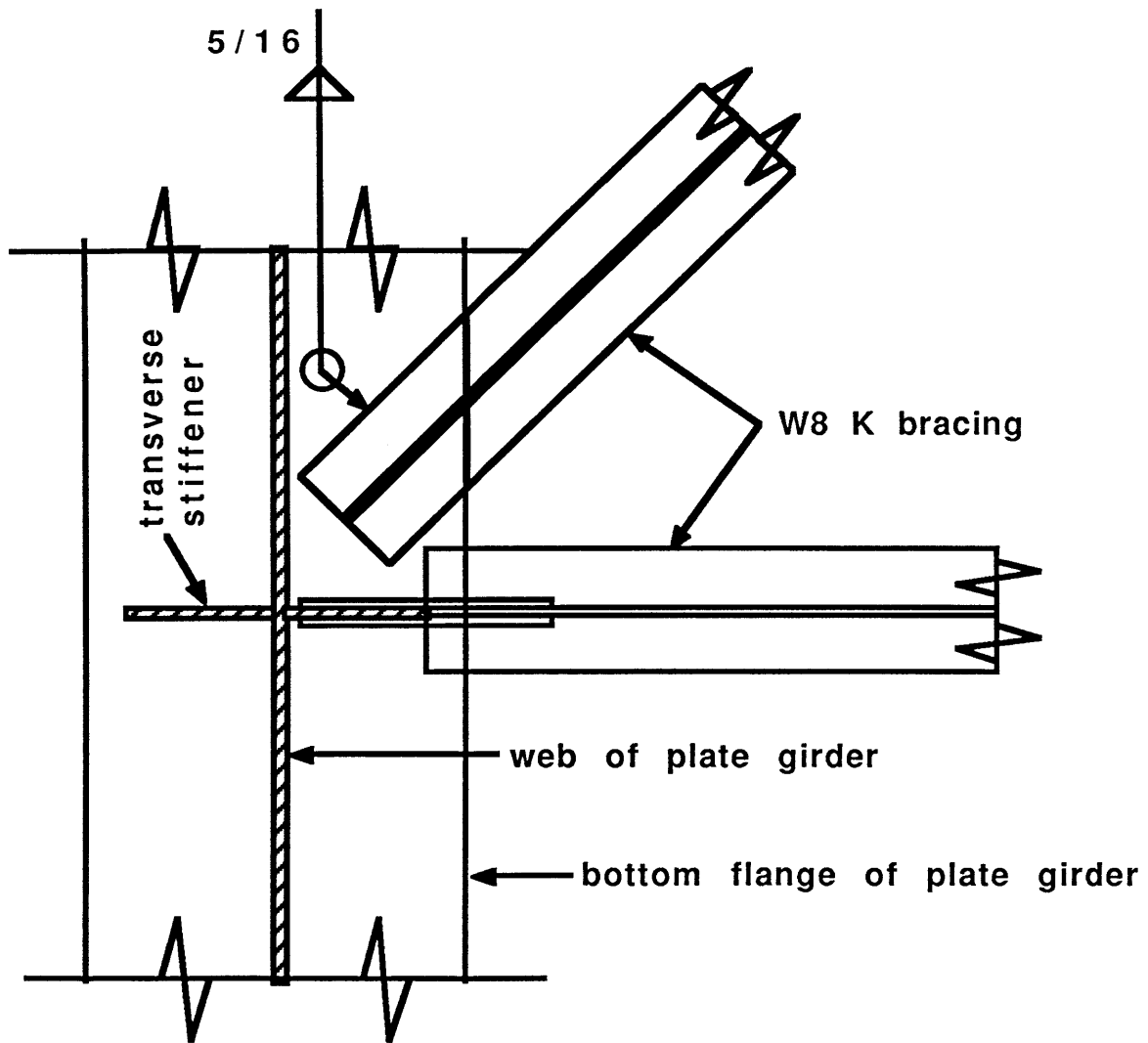


Figure 6-7:
Sketch of fracture surface near origin of crack



View looking down

Figure 6-8:
Detail at intermediate panel points for sway bracing

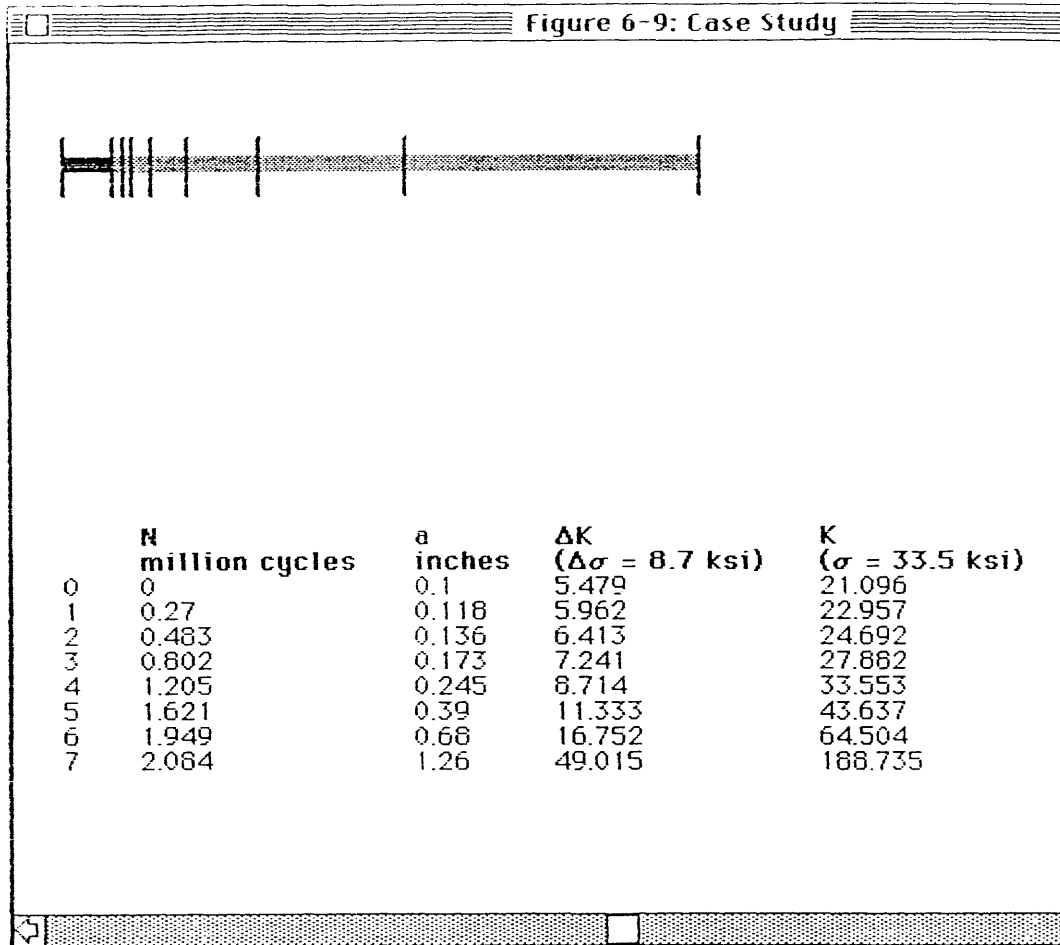


Figure 6-9:
Crack growth simulation: Case Study: stress range 8.7 ksi

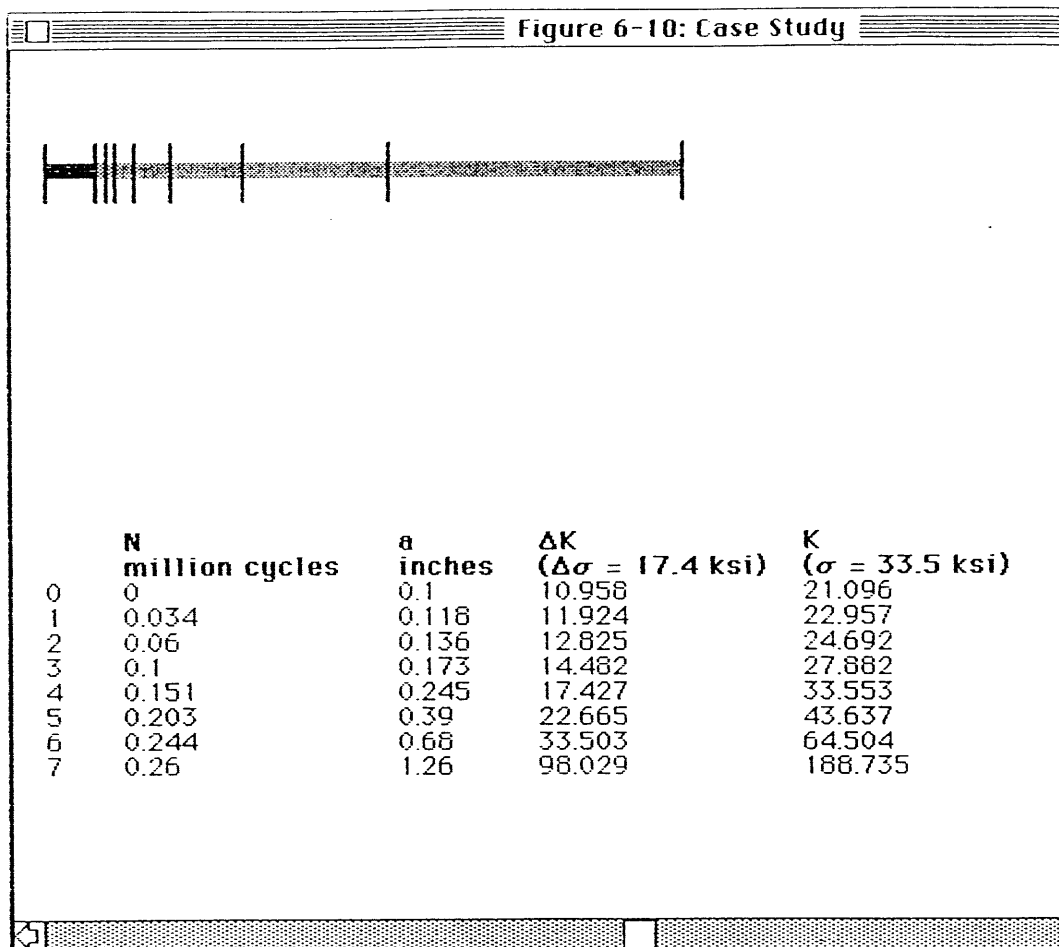


Figure 6-10:
Crack growth simulation: Case Study: stress range 17.4 ksi

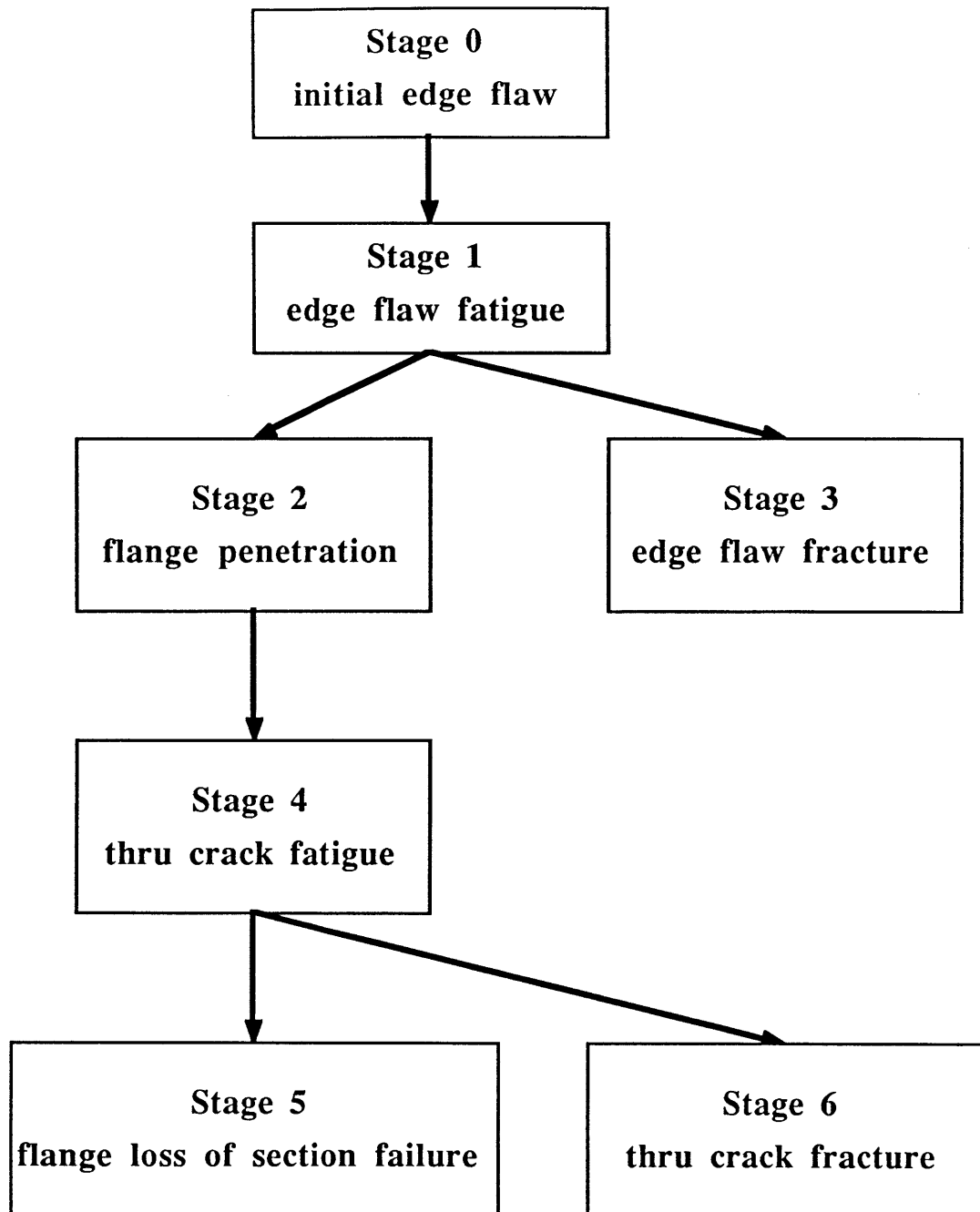
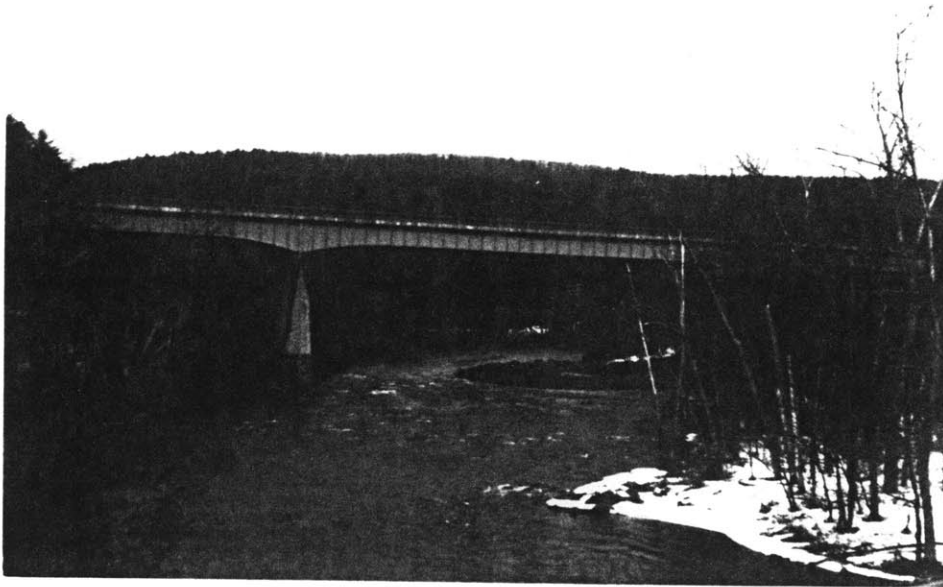
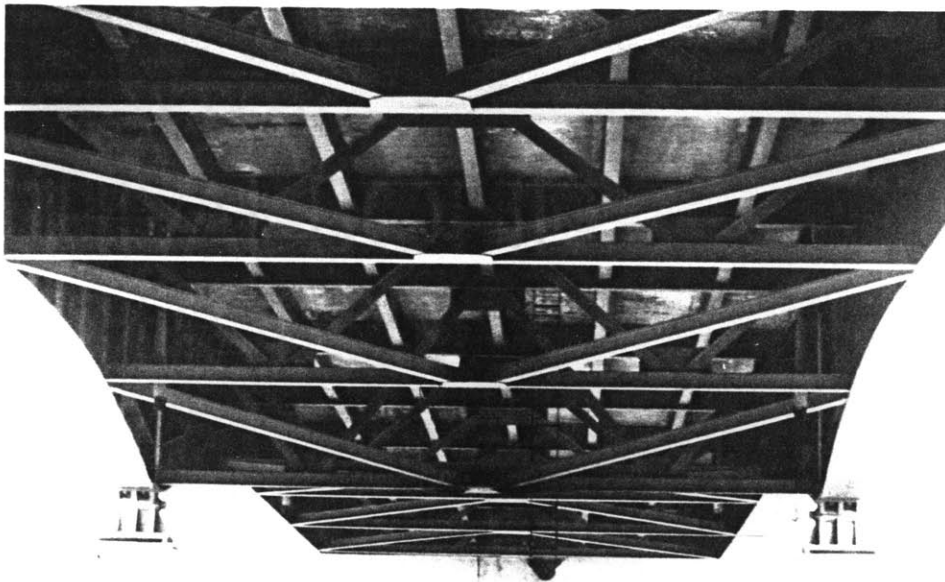


Figure 6-11:
Qualitative behaviors: case study bridge



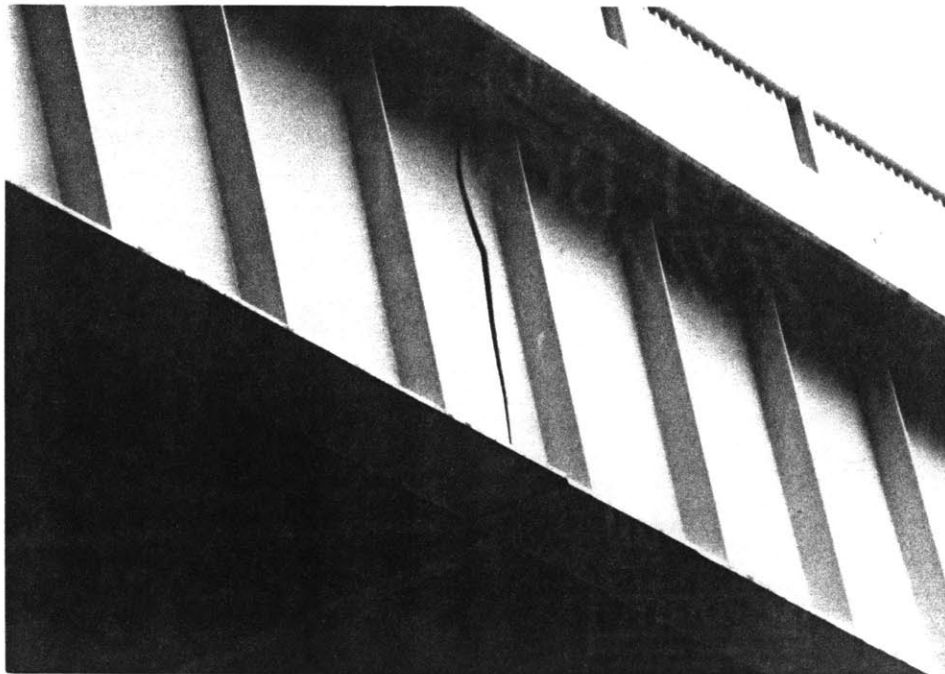
Photograph 6.1
Case study bridge: elevation looking north



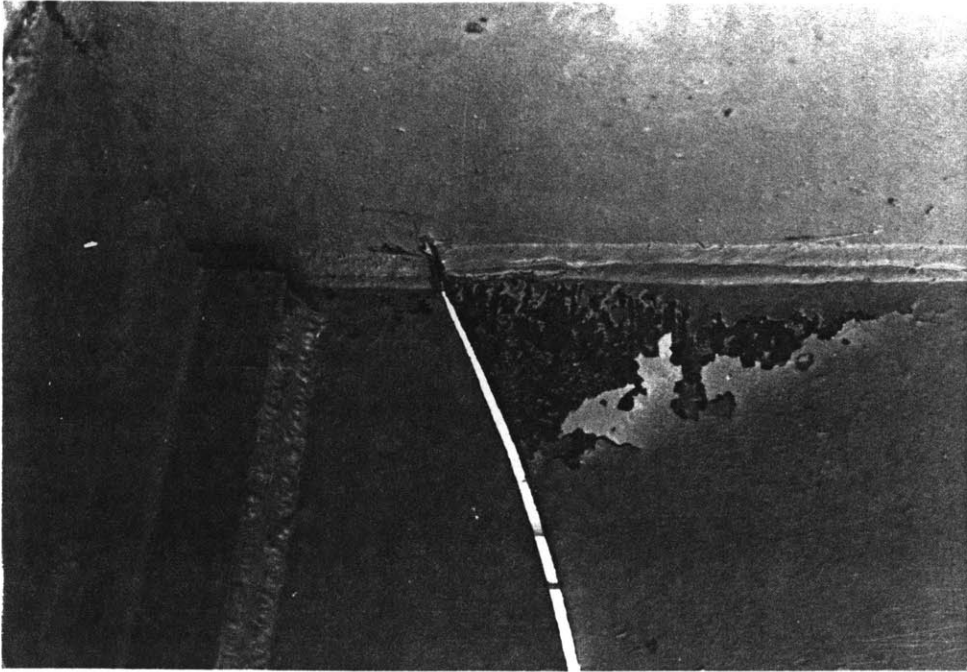
Photograph 6.2
Case study bridge: typical framing for plate girder spans



Photograph 6.3
Crack in north girder, located at third point of central span



Photograph 6.4
Crack in north girder, severing bottom flange and full web



Photograph 6.5
Interior view of crack at top flange



Photograph 6.6
Interior view of crack at bottom flange

Chapter 7

System capabilities and limitations

7.1 Performance within domain

Following the successful construction of the software tool CRACK, the system's performance within the problem domain was investigated. Good software engineering practice involves verification and validation of implementations. Verification determines if the program meets its design specification, that is, being free from defects and consistent with the intended model. Validation checks that the implementation is sufficient for its intended application [Bratley et al 87]. Evaluation of CRACK's performance focuses on verification. Since this implementation is a research prototype, validation for use outside a research setting requires additional development work as discussed in subsequent sections of this chapter. Various approaches may be used for program verification including [Bratley et al 87]:

- 1) manual verification of logic
- 2) modular testing
- 3) checking against known solutions
- 4) sensitivity testing by varying a single parameter
- 5) stress testing with out of range values.

The primary technique used to verify CRACK is testing against known solutions. Test cases of two basic types may be gleaned from the literature: 1) failure case studies and 2) experimental programs. A case study provides extensive information about a specific point in the design space corresponding to the girder cross section which failed. Experimental data forms the basis for the AASHTO fatigue categories. Test results for a particular structural configuration under controlled variation of parameters such as stress range and yield strength gives information about a region in the design space in the context of actual design procedures. A representative set of examples has been assembled from the literature to test the operation of CRACK. The test cases span the three target tasks of failure analysis, predictive modeling, and design critique as well as the different crack cause classes of secondary / distortion induced stress and initial flaw / low fatigue resistant details. The test cases are as follows:

Design critique: experimental program [NCHRP102 70]

Coverplate on welded beam:	6, 10, and 20 ksi stress range
Plain welded beam :	6, 10, and 20 ksi stress range
Plain rolled beam:	6, 10, and 20 ksi stress range
Predictive modeling: case studies [Fisher 84]	
Yellow Mill Pond:	end welded cover plate
Polk County:	floor beam web gap
Failure analysis: case studies [Fisher 84]	
Belle Fourche River:	diaphragm web gap
Cuyahoga County:	stiffener web gap
Cuyahoga County:	stiffener without web gap

Each of these fourteen cases is discussed below. Additional details of the CRACK test runs are provided in Appendix A.

7.1.1 Design critique

The first set of test cases was selected to examine the match between CRACK's conclusions and some of the experiments data underlying the AASHTO fatigue curves. Three detail types were selected to examine three different fatigue categories: end welded cover plates Category E, plain welded beams Category B, and plain rolled beams Category A. The sample cross sections are based on experimental specimens for a testing program to investigate the effect of weldments on the fatigue strength of steel beams, sponsored by the National Cooperative Highway Research Program [NCHRP102 70]. CRACK was run on three stress ranges (6, 20, and 20 ksi) for each detail type in order to plot a line to be compared to the mean regression line representing the least square fit of the experimental data to the linear model:

$$\log N = B_1 + B_2 \log Sr$$

$$\log N = 8.975 - 2.877 \log Sr : \text{welded end of coverplated beams A36 steel}$$

$$\log N = 10.870 - 3.372 \log Sr: \text{plain welded beams}$$

$$\log N = 10.637 - 2.943 \log Sr: \text{plain rolled beams}$$

N = cycles to failure

Sr = stress range (ksi)

Figure 7-1 shows the relationship between the three experimental regression lines and the nine CRACK test cases. The agreement between the experimental and CRACK fatigue lives is good. In most cases, CRACK predicts a slightly shorter fatigue life,

which can be explained by the different definitions of failure. In the experimental program, failure was defined to be an increase in midspan deflection of 0.020", which corresponded to the crack severing approximately 75% of the tension flange area. CRACK uses a definition of failure as flange penetration, which does not include the stage of crack growth as a through crack in the flange. The CRACK fatigue lives would thus be expected to be slightly shorter than the experimental ones, as is generally shown in Figure 7-1.

7.1.2 Predictive modeling

Since the development of CRACK was not done in conjunction with a field testing program, it was not feasible to use the predictive modeling mode to anticipate symptoms and then search for expression of those symptoms in an actual bridge. The predictive modeling mode was instead used on two literature failure analysis examples where cracking distress was identified, but crack growth had not progressed to a failure of the entire cross section.

The first predictive modeling case is the Yellow Mill Pond bridge (Case 3.1 [Fisher 84]). When this bridge reached a modest service age of 12 years, cracking was first found at the toe of the transverse fillet weld connecting the coverplate to the tension flange of the rolled beam. At an age of 18.5 years the cracks had reached a size of approximately 1 inch deep. A total traffic count of 259 million vehicles for 18.5 years with 13.5% trucks gives an average daily truck traffic (ADTT) of 5200. For 1.8 cycles per truck, an effective ADTT of 9360 gives a total of 63.2 million truck cycles for the 18.5 year service life (Note that values and formulae which appear in the first edition of [Fisher 84] must be used with care because of frequent typographic errors). Appendix A shows the conclusions of the crack consultation for the Yellow Mill Pond bridge. The predicted crack size of 0.64 inches is in good agreement with the observed crack size of 1 inch. The prediction of fracture at a critical crack size of 0.64 inches is not in agreement with the observed behavior of complete cracking of the entire beam flange without initiating crack instability. This discrepancy arises because of CRACK's assumption of a residual stress field generating yield stresses at the crack tip. Although this assumption is reasonable when the crack is small and near the weld zone, it becomes increasingly unrealistic as the crack grows deeply into the flange. This causes the incorrect prediction of fracture initiating before the flange is severed by fatigue.

The second predictive modeling case belongs to the category of secondary /distortion induced cracking. After 17 years of service, the Polk County bridge was found to

have cracks in the web plate in the negative moment regions along the web-to-flange fillet welds above the floor-beam connecting plate (Case 12.2 [Fisher 84]). The CRACK consultation, shown in Appendix A, predicted a fatigue life of 5 million cycles for the cracks to grow through the web. This is longer than the observed web penetration at about 3 million cycles. Part of this discrepancy may be attributed to the skewed bridge layout which accelerates web gap cracking. This skew effect is not taken into account in CRACK, leading to an overestimate of the fatigue life.

7.1.3 Failure analysis

Three failure analysis test cases were chosen. Two cases, the Belle Fourche River bridge (Case 13.1 [Fisher 84]) and the Cuyahoga County bridge (Case 11.1 [Fisher 84]) fall into the crack cause category of distortion induced cracking at a web gap. The Cuyahoga County case was also checked without the web gap condition.

The Belle Fourche River bridge experienced web cracking at the gap between the flanges and the transverse stiffener which was used as the attachment plate for X-type cross-frames. The differential deflection of adjacent girders produces out-of-plane deformation in the web gap where the stiffeners are not attached to the flanges. The CRACK consultation in Appendix A predicted a longer fatigue life than was actually observed for web penetration

The Cuyahoga County bridge also experienced web gap cracking at transverse stiffeners. The out-of-plane distortions were not due to service conditions, but were instead produced by sway of the girders as they were shipped by rail. The extremely high shipment stress range resulted in cracking after the relatively short life of about 50,000 cycles. The CRACK conclusions in Appendix A are in good agreement with observed behavior.

As a final test case the Cuyahoga County bridge was reexamined assuming the gap condition was eliminated. This resulted in a CRACK fatigue life of 35 million cycles, which is in reasonable agreement with the 20.6 million cycles ($N = 4.446 \times 10^9 \text{ Sr}^{-3}$) which would be allowed under the proposed AASHTO design curve appropriate for transverse stiffeners [NCHRP286 86].

7.2 Limitations of domain knowledge

When the domain knowledge was presented in Chapter 4, certain simplifying assumptions were made with regard to: crack tip plasticity, applied stress, decomposition method of determining stress intensity factors, methods for determining geometric correction factors, crack growth law, application of fracture mechanics to weld fatigue, and metallurgic and fractographic features. If CRACK is to be refined from the current prototype to a more complete version, these issues must be addressed in more depth. Each of these areas is discussed in turn and methods are suggested for adding capabilities not provided in the present version of CRACK. In addition to these specific knowledge base extensions, development of CRACK into a complete system for use by bridge engineers would require the ability to consider an entire bridge structure. Focusing on a specific cross-section was an appropriate level of granularity for the research scope. Fatigue inspection focuses on localized details, but does not consider them in isolation. To create a practical tool, knowledge must be incorporated to relate the macro description of an entire bridge's overall structural system to the micro description of cross-sections at specific connection details.

Plasticity:

When crack tip plastic deformation is unrestricted by elastic material around the crack, elastoplastic techniques must be used to predict crack size at failure. As discussed in Chapter 4, the plastic zone size is small and hence contained in an elastic field characterized by the stress intensity factor K if the gross stress is kept below half the yield stress. However, at most welds the total stress applied to the crack is the full yield stress due to residual stresses. To improve the accuracy of results in cases with a significantly large plastic zone, the physical crack length may be increased to include the length of the plastic zone:

$$a_{\text{eff}} = a' + r_p$$

where a_{eff} = effective crack length for calculations

a' = physical crack length

r_p = plastic zone size

$$= (K' / Y)^2 / (2 \pi)$$

K' = stress intensity factor for crack of length a'

Y = material flow strength

= yield stress (neglecting strain-hardening)

For example use of this effective crack length approach, refer to cases 5.2 and

7.1 in [Fisher 84]. No such adjustments for plastic zone size at high stress levels is included in CRACK. It is arguable whether such correction is required for welds since the cyclic crack tip plastic zone is usually small, due to reversed yielding [Maddox 74].

Applied Stress:

The concept of stress is simplistically represented in CRACK. The entire description of stresses is collapsed into two numbers, the maximum total stress and the effective stress range. This coarse representation of stress fails to capture aspects of the physical behavior which are significant in some contexts.

Maximum total stress is represented by a single value denoting the sum of dead load, live load, and residual stresses. Use of a single value obviously neglects the spatial variation of the dead and live load moment stresses with distance from the girder's neutral axis and of the residual stresses with distance from the weld. The Yellow Mill Pond bridge discussed in Section 7.1.2 is an example of an error arising from CRACK's lack of knowledge of spatial variation in residual stress. The single value representation also neglects the directionality of the stress, making an implicit assumption that the specified maximum total stress is perpendicular to the crack. Since the orientation between the principle stress direction and the crack is not represented, CRACK cannot differentiate between the immanent threat of a girder crack growing transversely and the less pressing case of a girder crack growing longitudinally.

The effective stress range is represented by a single value denoting the Miner equivalent constant amplitude stress range. The complex stochastic nature of actual highway load histories is thus collapsed into a single number based on the simplistic assumptions of nominal stresses and linear cumulative damage. Use of a single value again neglects variation as the crack grows. Effects caused by the irregularities inherent in the real load history, such as increased fatigue life for periodic overloads [Fuchs&Stephens 80], are not taken into account by using an effective stress range. However, extensive experimental tests have shown that for highway bridges, using such an approach gives good results [NCHRP188 78].

Decomposition determination of stress intensity factors:

The superposition method used in CRACK to compute K factors assumes no interaction between free surface, finite width, flaw shape, and opening stress distribution. As discussed in Chapter 4, this approach is reasonably accurate for cases where the state of stress remote from the crack is predominantly one of uniform extension, as is true for

most plate girder cracking. However, for three dimension cracks with nonuniform and steep gradients between near and far stress states, there is a potential for substantial inaccuracies [Albrecht&Yamada 77]. For example, cases dominated by plate bending or large thermal stress gradients would not be well modeled with this technique. In such cases, more powerful analytic techniques are needed, such as a finite element method (FEM) program with special crack elements.

Numeric methods for geometric correction factor computation

The geometric factor F_g is frequently the dominant correction factor for determining K . As discussed in chapter 4, F_g may be determined from: 1) a known parametric solution based on previous analysis of the generic detail configuration, 2) calculation of opening stress distribution on the cracking line (usually by finite element methods) and elimination of the opening stresses by integration, or 3) directly determination of the stress intensity factor, by-passing the calculation of F_g , by use of numeric methods such as FEM with special crack elements. CRACK does not hook to a FEM package of any kind, so it is limited to details with available parametric solutions.

An obvious extension would be to provide access to an existing FEM package. Choosing an appropriate FEM package depends on computational requirements, compatibility, and availability. Using the decomposition method minimizes the computations since a single stress analysis for the uncracked geometry finds the opening stresses on the crack line. This method circumvents the need for special crack elements, so a basic FEM package could be selected based on wide use and availability of the source code, such as SAPIV [Bathe et al. 74]. Use of a commercial FEM package available on Macintosh hardware, such as MSC/pal [MSC/pal 88] is desirable if porting problems are to be minimized. An existing research program which appears to be particularly well matched for connection to CRACK is FAST-1 Fracture Analysis of Structures [Taheri&Mufti 87]. This Fortran-77 program determines stress intensity factors by FEM and generates an explicit relationship between stress intensity factor and crack length through a discrete table. The initial grid representing the cracked geometry is prescribed by the user. An interface could be implemented between CRACK and FAST-1 in the vein of intelligence front-end interfaces to analysis packages [Nicklaus et al 88]. Automatic mesh generation for FEM is an active area of research.

Crack growth using Paris law:

There are three weaknesses to the crack growth calculation by direct integration of the Paris power law as presented in Chapter 4. First, only the linear portion of the log-log relationship between crack growth rate da/dN and stress intensity range ΔK shown in Figure 4-9 is represented by the Paris law. For cases where the steel has a high fracture toughness relative to yield strength (so the fracture toughness does not fall below Region III) and the applied stress intensity range is high, the crack growth rate will be accelerated due to superposition of a ductile tear mechanism onto the mechanism of cyclic sub-critical crack extension [Barsom&Rolfe 87]. CRACK's use of the Paris law in this case underestimates crack growth rate and overestimates fatigue life. Use of Forman's law [Taheri&Mufti 87] under these circumstances would address the instability of the crack growth when the stress intensity factor K approaches the fracture toughness K_c :

$$da/dN = A \{ \Delta K^n / [(1 - R) K_c - \Delta K] \}$$

$$\text{where } R = K_{\min} / K_{\max}$$

Second, the material constants A and m in CRACK are based on conservative upper-bound values for ferrite-pearlite steels. Use of these fixed values hides the underlying scatter of crack growth rate data, ignores the difference in A and m for martensitic steels, and neglects the environmental effect of temperature and secondary effects of load cycle frequency and wave shape, and plate thickness [Fuchs&Stephens 80].

Third, stress corrosion cracking is not considered. This mechanism is characterized by crack growth under static stresses well below the yield strength for structural elements exposed to an aggressive environment. Most bridge steels are not sensitive to this type of behavior [Fisher 81]. The chemical conditions driving this form of corrosion cracking would need to be added to the knowledge base to account for the possibility of stress corrosion crack growth.

Applying fracture mechanics to weld fatigue:

There are problems in applying fracture mechanics to weld fatigue. The accuracy of the analysis is severely limited by the uncertainty of the input data such as: stress, stress range, size and shape of the initial and the critical cracks, material properties, and stress intensity solutions. There is a need for improved understanding of initiating imperfections and behavior of very small cracks [Barsom&Rolfe 87]. Since the initial flaw size, shape, and orientation have a large influence on fatigue life (the typical experimental scatter factor of four is largely due to these parameters), better characterization of initiating imperfections is needed. The initiation and propagation behavior of very small cracks

points up some of the weaknesses in linear elastic fracture mechanics (LEFM), especially the assumptions that imperfections start as crack-like flaws with no sharpening time, and that the plastic zone size is small. If the underlying assumptions of LEFM are not satisfied, the use of the Paris law and stress intensity factor are not supported. These weaknesses are a partial justification for use of the existing AASHTO fatigue curve approach.

Metallurgy and fractography:

The representation of material properties is rudimentary in CRACK. Metallurgy is an active area of knowledge-based work. The Aladin expert system was developed to assist metallurgists in the design and evaluation of new alloys [Hultage et al. 86]. A large knowledge bank of alloy and metallurgical information is represented including alloy product types, applications, properties, and production methods, phase diagrams, phase transitions, microstructure, and crystal structure. The material property knowledge could be utilized for CRACK, especially since fracture toughness has strong dependence on microstructure and phase. CRACK's representation of fractographic features is limited, only noting presence or absence of chevron marks and shear lips and sized beach marked regions. Work has also been done to build a knowledge-based tool for fractography [Morrill&Wright 86]. In addition to increasing the fractography knowledge base, some kind of sketch display interface would greatly enhance the usefulness of fracture feature information.

7.3 Critique of three level architecture

This research prototype version of CRACK was designed to test out new ideas in combining rule-based, qualitative, and quantitative reasoning. As an experimental piece of software, it is incomplete in many dimensions not needed to test the hypothesized system architecture. The system works in the sense that the ideas work. This prototype is fragile and has just begun to perform in a research setting.

The appropriateness of a knowledge-based approach and the adequacy of the three-level architecture are supported by the successful verification of the system on a variety of test cases. The integration of rules, qualitative, and quantitative methods is thus shown to be a feasible approach to solving realistic engineering problems in the sample domain of fatigue and fracture. The main weakness of the system is in the area of connection and utilization of the qualitative level. This is as would be expected, since qualitative reasoning is the least mature of the methods used.

To illustrate the following discussion of the qualitative level, three examples of qualitative simulations are presented in Appendix B. The examples are: 1) basic fatigue and fracture behavior, 2) growth through multiple girder elements, and 3) effects of a temperature drop. The first case presents the qualitative behaviors shown in Figure 5-5 for the qualitative influence network shown in Figure 5-7. The initial time point state consists of a flaw subject to cyclic stress (state 0). The time interval state which follows represents fatigue growth of the initial flaw (state 1). There are three possible terminating time point states: the initial flaw reaches critical size and the plate fails by fast fracture (state 2), the initial flaw grows by fatigue until the plate is penetrated (state 3), and simultaneous occurrence of fracture and penetration (state 4). Figure 5-5 summarizes these three paths.

Path 1: state 0 ---> state 1 ---> state 2
 Path 2: state 0 ---> state 1 ---> state 3
 Path 3: state 0 ---> state 1 ---> state 4

The model for the second case begins with a flaw in the flange to web connection element. The flaw may grow into the flange after it has grown through the region of the connection. The initial time point state consists of a flaw in the bottom flange connection subject to cyclic stress (state 0). The time interval state which follows represents fatigue growth of the initial flaw within the connection (state 1). The three following time point states are the crack entry into the bottom flange (state 2), fracture in the connection (state 3), and simultaneous fracture and flange entry (state 4). After flange entry, the flaw again grows by fatigue for a time interval, this time in the flange plate (state 5). The three possible terminating time point conditions to flange fatigue growth are: flange penetration (state 6), flange fracture (state 7), and simultaneous penetration and fracture (state 8).

The five paths are summarized in Figure 7-2:

Path 1: state 0 ---> state 1 ---> state 2 ---> state 5 ---> state 6
 Path 2: state 0 ---> state 1 ---> state 2 ---> state 5 ---> state 7
 Path 3: state 0 ---> state 1 ---> state 2 ---> state 5 ---> state 8
 Path 4: state 0 ---> state 1 ---> state 3
 Path 5: state 0 ---> state 1 ---> state 4

The model for the third case begins with a flaw that is not subject to cyclic stresses but instead has a temperature which is about to drop (state 0). The time interval of falling temperature (state 1) has seven possible time point terminating states. The plate may fracture due to the temperature decreased fracture toughness (state 2) while the temperature is falling. The temperature may stabilize with a lower, but positive, residual strength

(state 3). The temperature may stabilize simultaneously with fracture (state 4). The remaining 4 states all include the fracture toughness reaching zero, which is not a physically meaningful condition. The definition of a state variable's quantity space must contain the specific symbol "0", which is used to delineate positive and negative regions of the variable's domain. Since some state variables may always be required to be greater than 0 (or always less than 0), this notational requirement leads to the generation of spurious states where the actually positive value was allowed to fall below its physically significant landmark threshold. This condition could of course be avoided by introducing a steady pseudo-state variable, $K_{c\text{pseudo}}$, which was set equal to $K_{c\text{threshold}}$ and redefining K_c as $K_{c\text{pseudo}} + K_{c\text{threshold}} + K_{c\text{effective}}$. This is representationally undesirable since it introduces state variables and qualitative constraint equations that are not physically meaningful. This change is also inefficient since it expands the network, combinatorially increasing computational resources. A similar mechanistic weakness is the three place predicate definition of the qualitative arithmetic constraints for addition and multiplication. Again, pseudovariables and constraints can overcome the limitation but at a representationally and computational cost.

The seven paths are summarized in Figure 7-3:

Path 1: state 0 ---> state 1 ---> state 2
Path 2: state 0 ---> state 1 ---> state 3
Path 3: state 0 ---> state 1 ---> state 4
Path 4: state 0 ---> state 1 ---> state 5
Path 5: state 0 ---> state 1 ---> state 6
Path 6: state 0 ---> state 1 ---> state 7
Path 7: state 0 ---> state 1 ---> state 8

The qualitative reasoning portion of CRACK as implemented is over-engineered and the results and capabilities of the qualitative simulation are under-utilized. The primary task of the qualitative level is to envision the possible ways a postulated crack could grow, constructing a tree of the different states the crack may pass through from initiation to failure. This qualitative information is used to set up the quantitative analyses of the individual crack growth stages and the connection of the stages into chains of possible behaviors. This to avoids a "big switch" approach to quantitative analysis, which would only allow the use of pre-enumerated solutions. Instead of taking this pre-enumerated approach, CRACK builds up a complete quantitative analysis by assembling partial solution stages under the guidance of the qualitative level. CRACK is over-engineered since the current generation and use of the qualitative level could be functionally replaced by a set of transition rules which could more efficiently predict crack growth

paths without the large underlying mechanism created to support the qualitative simulation approach. This criticism of over-engineering is made clear by examining the first and second examples in Appendix B. The sequence of fatigue growth in a plate followed by penetration, fracture, or simultaneous penetration/fracture is a simple pattern of behaviors that is generated for each succeeding plate that is physically connected to the plate containing the initial flaw. This physical behavior could be captured in a small set of transition rules that required far less computational resources than the qualitative simulation. The disadvantage of replacing the qualitative reasoning level with a sparse set of transition rules is that a great deal of additional domain knowledge would be discarded. Although this knowledge is not easily accessible nor made use in the current version, switching to a domain transition rule mechanism would preclude possible exploitation of this underutilized knowledge.

In addition to this over-engineering criticism, the qualitative information is thus also under-utilized. The ability of the qualitative level to predict valid progressions of physical states based on qualitative assumptions is not used in most of the ways it could be including: revising hypotheses, articulating uncertainty, and generating explanations. The generation of hypotheses is based on assumptions about what parameters are varying and which remain constant. As shown by the third example in Appendix B, if there is no cyclic stress, a crack that was stable may become unstable and fracture due not to an increase in the crack length but to a decrease in temperature. This information is useful in evaluating a hypothesis that seems largely correct but predicts premature fracture under a constant temperature assumption.

A weakness of CRACK is its inability to use the causal domain model represented by the qualitative level in fixing an almost right hypothesis. A way of combining heuristic rules and causal models in a way that does empower hypothesis debugging based on the causal knowledge is the Generate, Test, and Debug paradigm [Simmons&Davis 87; [Simmons 88]. This problem solving approach employs heuristic rules to generate an initial hypothesis, tests the hypothesis using a combined qualitative and quantitative simulation technique, and debugs faulty hypotheses. A bug is defined to be an inconsistency between the desired value of some parameter and its value as predicted by the tester. The tester passes to the debugger a causal explanation for the detected bugs. The debugger traces through the causal dependency structures to determine the underlying assumptions, one or more of which are faulty and causing the bug. The bug is repaired by replacing assumptions. This ability to revise assumptions based on causal dependency information is

completely lacking in CRACK.

The way in which qualitative information may be used to articulate uncertainty is the construction of a parametric study. A parameter study varies one or more parameters over a set range to study how these parameters affect the solution. Engineers use parameter studies extensively [Nicklaus et al 88]. When dealing with input parameters that are characterized by a range rather than a specific value, a set of bracketing cases may be constructed combining the appropriate extreme input values to define the boundaries of the possible physical responses to the driving parameters. The qualitative information contained in the influence network is precisely the knowledge of the physical system that is required to assemble the sets of bounding input values, combining high and low inputs in a way that defines the extremes of the response state variables. In CRACK's domain, such a parametric study could perform an error analysis, giving possible response ranges based on the uncertain values of physical parameters such as fracture toughness, stress range, and crack growth rates. CRACK does not currently make such use of the captured qualitative knowledge. A recent extension to qualitative simulation which seems exactly suited to this problem uses qualitative but incomplete quantitative knowledge to supplement quantitative reasoning [Kuipers&Berleant 88]. The summary judgment on the three-level system architecture is that it demonstrates promise but its potential capabilities are not realized in the current incarnation. Since the particular qualitative notation and calculus was selected based on its expressive power, the difficulties in utilizing qualitative domain knowledge are likely to generalize to the approaches of confluences and qualitative process theory. Further advances in the AI technology seem required before qualitative reasoners will be powerful enough to deal with realistic engineering problem solving beyond a research setting.

The qualitative domain knowledge is not utilized to explain CRACK's behavior nor to justify its conclusions. In the third example in Appendix B, the temperature drop causes a crack to become unstable and fracture with no increase in crack length. This information is potentially useful in explaining why a girder cross-section behaved in the observed fashion, such as fracture at a crack length that was previously known to be sub-critical.

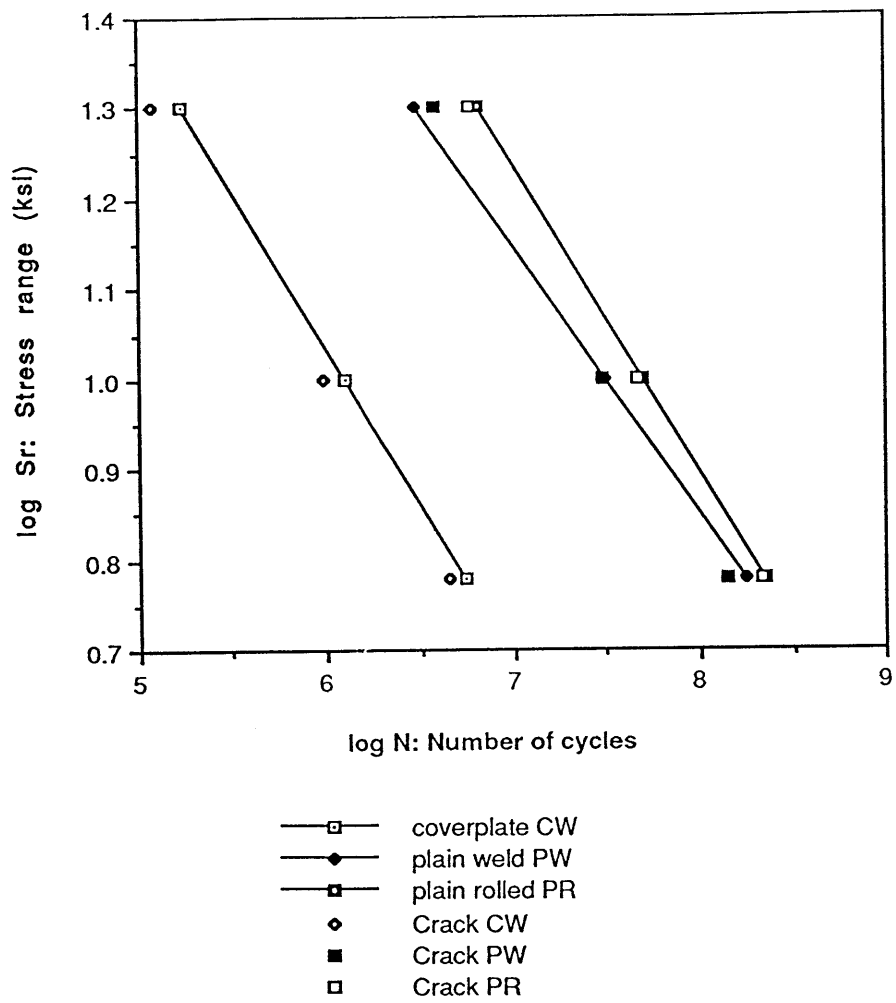


Figure 7-1:
Design critique test case results

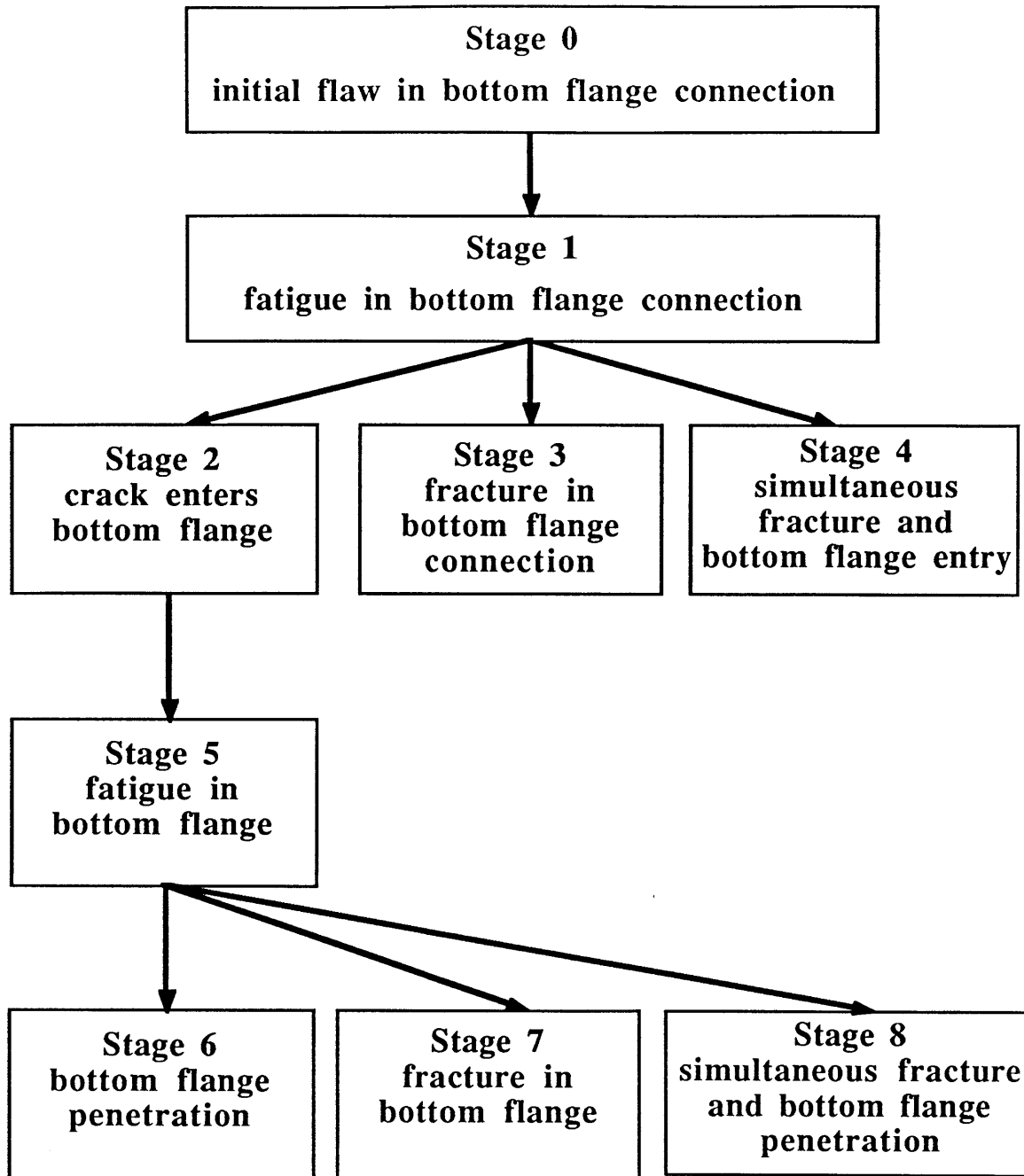


Figure 7-2:
Qualitative behaviors: appendix B example 2

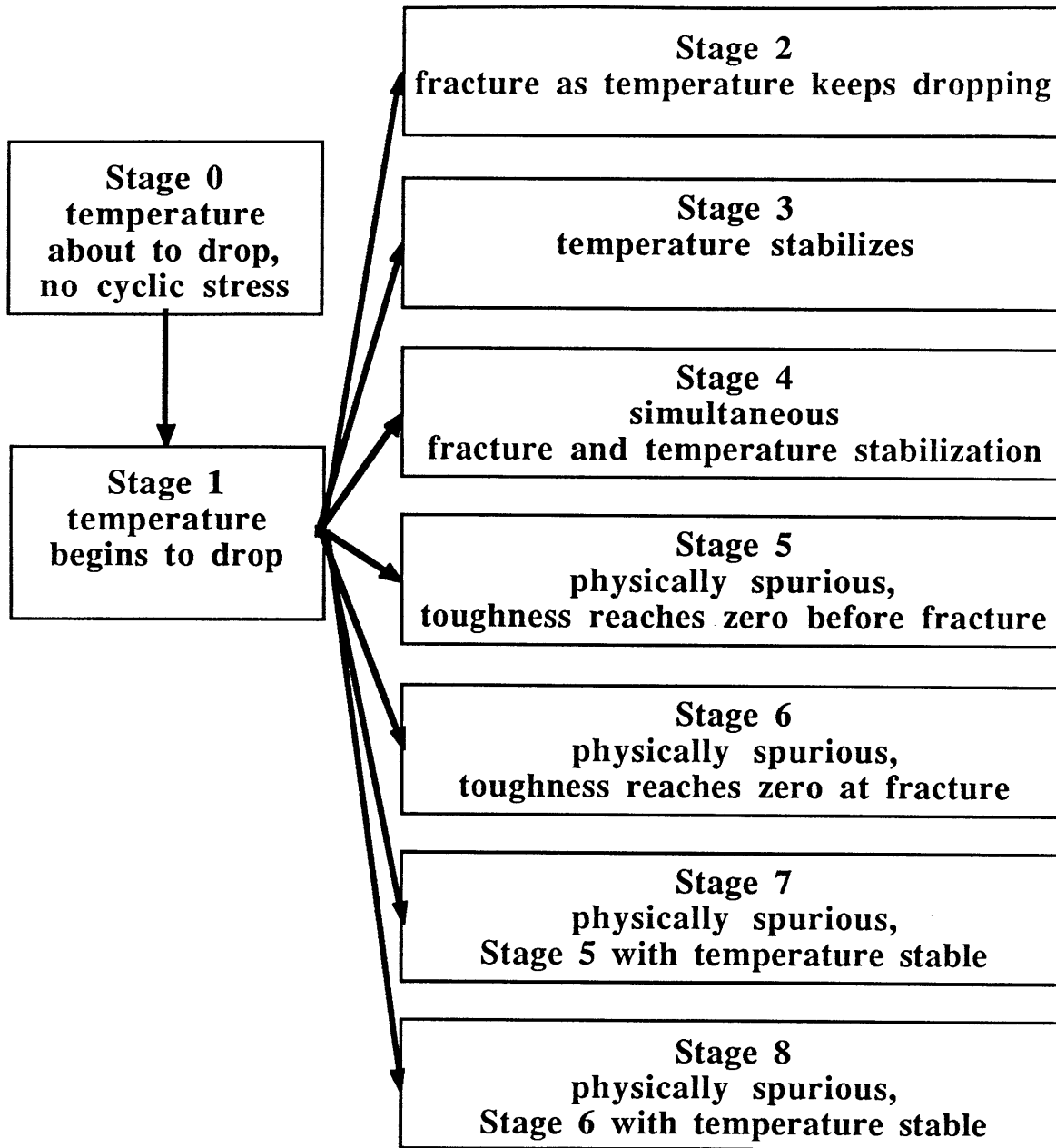


Figure 7-3:
Qualitative behaviors: appendix B example 3

Chapter 8

Conclusion

8.1 Contributions

When building knowledge-based systems in domains which deal with physical behavior, ways of representing and reasoning about the way physical parameters interact are required. One approach that has been suggested by a variety of researchers is qualitative reasoning. The central contribution of the system development for CRACK is the exploration of the use of qualitative simulation as the formalism for representing and manipulating causal influences in the engineering domain of fatigue and fracture. Multiple abstraction levels with different vocabularies has been shown to be a powerful problem solving tool [Patil 81]. The strategy of using an intermediate qualitative simulation layer manipulating first order engineering models to connect a predominantly heuristic and symbolic rule-based top layer with a largely procedural and numeric quantitative root layer is applicable to a wide variety of engineering problems and is indeed being explored by other researchers in different domains. For example, the HTEX [Throop 88] system for diagnosing single faults in idealized heat exchanger trains uses symbolic reasoning for hypothesis formation, order-of-magnitude reasoning for hypothesis selection, and numerical simulation for hypothesis testing. CRACK represents the development in a particular domain of a flexible and robust engineering problem solving tool.

The significance of this research consists of the following contributions. The lessons here are pragmatic ones, based on considered in-depth application of the three-level rule/qualitative/quantitative architecture to one problem domain, carefully selected to be representative of a large class of engineering applications.

Improved understanding of the state of the art of fracture mechanics as applied to in-service steel bridges by the explicit compilation of domain heuristics.

Qualitative reasoning about fracture to direct search for problem solutions and to construct explanations.

Development of a multilevel knowledge representation capable of describing a multifaceted engineering problem solution, using an intermediate qualitative layer and engineering models to connect a heuristic top layer with a numeric root layer.

Creation of a flexible and robust engineering problem solving tool by combined heuristic and causal reasoning. This tool will allow a more appropriate representation, thus providing greater utility and an improved understanding of the engineering problem solving process.

A conceptual framework for constructing and utilizing engineering models in problem solving.

8.2 Bridge engineering practice

The problem area is a significant and substantive one of inherent engineering interest and usefulness. Gathering and structuring the domain is a valuable formalization in and of itself. The identification, inspection, evaluation, and repair of fatigue and fracture damage in steel bridges are all complex engineering tasks. The number of qualified experts is far below the level required to deal with the large bridge population. The problem domain is well suited for use of a knowledge-based approach. The knowledge exists but is frequently not utilized. The knowledge is of diverse types (statistical, heuristic, engineering principles, etc.) but circumscribed and well contained. There are multiple uses of the same knowledge; for analysis of failures, for determining causes of distress and fixes, for predicting remaining service life, and for verification and optimization of design.

This research prototype version of CRACK is a long way from a deliverable tool for use by bridge engineers. Development into a usable tool requires interaction with and knowledge acquisition from state and federal department of transportation officials. The value of a delivery version of CRACK as a practical tool depends not only on the narrow technical performance of the computational tool but also on the ability of the broader decision making environment to effectively utilize the information generated. For example, in the field inspection stage of the knowledge acquisition process, it became clear that steel bridges are frequently not painted in a timely manner, so that preventable corrosion is a significant problem. The appropriateness of a sophisticated fracture and fatigue evaluation tool is questionable for a state department of transportation that does not adequately meet the much simpler need of regular painting.

Although some members of the bridge engineering community do not think further refinement in areas such as fatigue design provisions is warranted due to the great variation in actual bridge loading [Galambos 79] the general professional consensus appears to be that better ways of dealing with fatigue are needed. The vast number of bridges in the United States makes routine testing of all structures impractical. Current standards

and specifications are based on simple design formulations that are generally conservative, tend to be restrictive, and often uneconomical. Research work is underway to better characterize loading, stress, lateral load distribution for multi-girder medium-to-short span deck and steel girder bridges commonly used throughout US highway system, subjected to heavy traffic and extreme loads [Illinois 86]. The ultimate goal of this research is a more accurate view of bridge behavior to be used to update design procedures, specifications, and codes, leading to safer and more economical initial designs, better prediction of remaining operating life, and provision for orderly replacement and repair of existing bridges. Knowledge-based tools similar to CRACK hold great promise to enable practicing design and maintenance bridge engineers to utilize the present and growing knowledge bank about the complex and sophisticated engineering discipline addressing fatigue and fracture in bridges. Knowledge-based tools offer a way around the bottle-neck of expertise in fracture and fatigue being required for solving relatively routine bridge design and evaluation problems. These tools reduce the detailed knowledge that an engineer must have of the numerical analysis routines and provide ready access to the existing knowledge base. The explicit knowledge representation and explanation capability provide mechanisms to record the reasoning on which decisions are made and providing insight into the underlying physical behavior. CRACK is a step towards turning the potential of knowledge-based and expert systems into solid solutions to engineering problems. This approach offers improved use of the limited resources of money and professionals to achieve safer bridges. The transfer of technology by knowledge-based tools leads to more skilled bridge engineers by improved dissemination of existing knowledge, allowing a more rational approach to bridge design and maintenance.

A major design issue for knowledge-based systems for civil engineering is deliverability to practicing engineers. It is highly desirable for a fieldable version to run on micro-computer hardware of a type with wide usefulness in an engineering office to counter the argument that AI requires costly hardware and software to solve non-toy problems. CRACK uses the relatively inexpensive hardware and software tools of the Macintosh personal computer line and the Prolog language.

8.3 Summary

In a narrow sense, the bulk of this thesis involves the synthesis of a software artifact, CRACK. The construction of this artifact was undertaken to provide a model for problem solving in the structural domain. The investigation of this model yields insight into general engineering problem solving. The principles behind the development of

CRACK are more generally applicable than the details surrounding the implementation of the particular artifact.

Engineering problem solving requires a useful, fundamentally based understanding of a system's response to its conditions of use, such as a structure's behavior under service loads. The requisite knowledge is interdisciplinary, spanning such specialties as material science and continuum mechanics. To select and utilize appropriate mathematical tools for design and service evaluation, a qualitative feel for the relative importance of various behaviors is essential. Expertise encompasses knowledge of what to ignore as well as what to include, what cases to investigate and what assumptions to change. The knowledge of how to proceed with partial information, how to focus on promising regions of the solution space, and how to choose and bring to bear applicable solution techniques is largely dependent on a qualitative understanding of the system's behavior. For problem solving in a domain with a well developed theory of causality, the approach used in CRACK provides a path to build a computational tool which allows the solution of complex problems by less experienced engineers.

Multilevel reasoning about fracture in bridges has been used as a test bed to develop a better computer representation of the qualitative to quantitative reasoning transformation which is an abstraction mechanism lying at the heart of engineering and an essential ingredient for explanation and justification of design and analysis methodology. The success of the system was measured by the range of phenomena that can be represented, the correctness and interest of the conclusions that can be supported, and the clarity and precision of the representational description. CRACK's architecture helps the user solve problems that require specialized knowledge or expertise, integrating explanation and problem-solving capabilities of expert systems with precision of traditional numeric computing. This approach is suitable to many engineering problems that involve ambiguous, contradictory, and imprecise data. Such ill-defined problems are not amenable to traditional algorithmic or symbolic techniques but may be successfully attacked by a combination of these methods in the rule/qualitative/quantitative manner.

The difference between pedantry and mastery may be defined as the difference between applying a rule to the letter, rigidly, unquestioningly, in cases where it fits and in cases where it does not fit and applying a rule with natural ease, with judgment, noticing the cases where it fits, and without ever letting the words of the rule obscure the purpose of the action or the opportunities of the situation [Polya 73]. Engineering knowledge-

based systems are restricted to pedantry if they have no knowledge of the fundamental causal mechanisms of their domain. Integrating the various reasoning strategies of rules, qualitative simulation, and quantitative analysis is a step along the way to creating computational tools that can display mastery.

References

- AASHO 65 American Association of State Highway Officials, *A Policy on Geometric Design of Rural Highways*, Washington, DC, 1965.
- AASHTO 85 American Association of State Highway Officials, *Standard Specifications for Highway Bridges*, Thirteenth Edition as amended 1984, and 1985, Washington, D. C., 1985.
- AASHTO 86 American Association of State Highway Officials, *Guide Specifications for Fracture Critical Non-redundant Steel Bridges*, as amended 1981, 1983, 1984, 1985, and 1986, Washington, D. C., 1986.
- Abelkis&Hudson 82 Abelkis, P. R., and C. M. Hudson, Eds., *Design of Fatigue and Fracture Resistant Structures, ASTM STP 761*, American Society for Testing and Materials, Philadelphia, 1982.
- Albrecht&Yamada 77 Albrecht, Pedro, and Kentaro Yamada, "Rapid Calculation of Stress Intensity Factors", *Journal of the Structural Division*, Proceedings of the American Society of Civil Engineers, Vol. 103 No. ST2, February, 1977.
- Anand&Parks 86 Anand, L., and D. M. Parks, *2.32 Mechanical Behavior of Materials II, Course Notes*, Massachusetts Institute of Technology Department of Mechanical Engineering, Cambridge, MA 01239, Spring 1986.
- Ashby&Jones 82 Ashby, Michael F., and David R. H. Jones, *Engineering Materials: An Introduction to their Properties and Applications*, Pergamon Press, Oxford, 1982.
- ASTM-A373 58 American Society for Testing and Materials, *ASTM A373-54T revised 58T, Tentative Specification for Structural Steel for Welding*, Philadelphia, PA, 1954 revised 1958.
- AWS 86 *Structural Welding Code - Steel*, AWS D1.1-86, American Welding Society, 1986.
- Baldwin et al 81 Baldwin, Jr., J. W., H. J. Salane, and R. C. Duffield, "Fatigue Cracks and Their Detection", *Transportation Research Record* 832, Washington, DC, 1981.

- Barsom&Rolfe 87 Barsom, John M., and Stanley T. Rolfe, *Fracture & Fatigue Control in Structures*, Second Edition, Prentice-Hall, Englewood Cliffs, NJ, 1987.
- Bathe et al. 74 Bathe, K.-J., E. L. Wilson, and F. E. Peterson, *SAPIV, A Structural Analysis Program for Static and Dynamic Response of Linear Systems*, Report No. EERC 73-11, University of California, Berkeley, CA, April, 1974.
- Bennet et al. 78 Bennett, James, Lewis Creary, Robert Engelmores, and Robert Melosh, *SACON: A Knowledge-Based Consultant for Structural Analysis*, Heuristic Programming Project, Computer Science Department, Stanford University, September, 1978.
- Bennett&Englemore 79 Bennett, James, and Robert Engelmores, "SACON: A Knowledge-Based Consultant for Structural Analysis", *Proceedings Sixth International Joint Conference on Artificial Intelligence*, 1979.
- Better Roads 87 "Better ways to inspect bridges", *Better Roads*, pages 24-25, November 1987.
- Bobrow 85 Bobrow, Daniel G., editor, *Qualitative Reasoning About Physical Systems*, The MIT Press, Cambridge, MA, 1985.
- Bratko 86 Bratko, Ivan, *PROLOG Programming for Artificial Intelligence*, Addison-Wesley, Reading, Massachusetts, 1986.
- Bratley et al 87 Bratley, Paul, Bennett L. Fox, and Linus E. Schrage, *A Guide to Simulation*, Second Edition, Springer-Verlag, New York, 1987.
- Broek 84 Broek, David, *Elementary engineering fracture mechanics*, Martinus Nijhoff Publishers, Boston, 1982.
- Broek 85 Broek, David, "Failure Analysis and Fracture Mechanics", *Metals Handbook, Ninth Edition, Volume 11, Failure Analysis and Prevention*, American Society for Metals, Metals Park, OH, 1985.
- Buchanan&Shortliffe 84 Buchanan, Bruce G., and Edward H. Shortliffe, *Rule-Base Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley Publishing, 1984.
- CAD/CAM 87 *The CAD/CAM Journal: For The Macintosh Professional*, Concepts Graphic Images Inc, 16 Beaver St., New York, Vol. 1

No. 5, October/November, 1987.

- Chan 86 Chan, Wang Tat, *Logic Programming for Managing Constraint-Based Engineering Design*, PhD Thesis, Department of Civil Engineering, Stanford University, March, 1986.
- Chandrasekaran&Bylander 88 Chandrasekaran, B. and Tom Bylander, "Task-Specific Problem Solving Architectures.", AAAI88 Seventh National Conference on Artificial Intelligence, 1988.
- Charniak&McDermott 85 Charniak, Eugene, and Drew McDermott, *Introduction to Artificial Intelligence*, Addison-Wesley, 1985.
- Chen 87 Chen, Stuart S., Bridge Fatigue Investigator (BFI), Seminar presentation, Lehigh University, November 23, 1987.
- Chen 88 Chen, Frank Hsi-Shang, *Expert-MCA, An Expert System for CAPCES Databases*, Master's Thesis, Civil Engineering Department, Massachusetts Institute of Technology, Cambridge, MA, February, 1988.
- Chen et al 87 Chen, Stuart S., John L. Wilson, and George K. Mikroudis, "Knowledge-Based Expert Systems in Civil Engineering at Lehigh University", *Proceedings of the Fifth National Conference on University Programs in Computer-Aided Engineering, Design and Manufacturing*, Ohio State University, Columbus, Ohio, 1987.
- Chen&Wilson 86 Chen, Stuart S. , and John L. Wilson, *Proposed Usage of BFI (Bridge Fatigue Investigator)*, ATLSS Report No. 86-02, Lehigh University, Bethlehem, PA, October, 1986.
- Cherneff 88 Cherneff, Jonathan, *Automatic Generation of Construction Schedules from Architectural Drawings*, Master's Thesis, Civil Engineering Department, Massachusetts Institute of Technology, Cambridge, MA, June, 1988.
- Chiles 88 Chiles, James R., "The Ships That Broke Hitler's Blockade", *American Heritage of Invention & Technology*, Volume 3, Number 3, Winter 1988.
- Citrenbaum et al 87 Citrenbaum, R., J. R. Geissnam, and R. Schultz, "Selecting a Shell", *A I Expert*, 500 Howard St., San Francisco, CA, Vol. 2 No. 9, September, 1987.

- Clark et al 88 Clark, K. L., F. G. McCabe, N. Jones, and C. Spenser, *LPA MacPROLOG Reference Manual*, Logic Programming Associates, London, Programming Logic Systems, 31 Crescent Drive, Milford, Connecticut, 1988.
- Clark&McCabe 84 Clark, K. L. and F. G. McCabe, *micro-PROLOG: Programming in Logic*, Prentice/Hall International, Englewood Cliffs, New Jersey, 1984.
- CRC 72 *CRC Handbook of Chemistry and Physics*, 53rd Edition, CRC Press, 1972.
- Cross 83 Cross, S., "A Qualitative Reasoning Approach to Mathematical and Heuristic Knowledge Integration.", in *CHI'83 Proceedings*, pages 186-189. Association for Computing Machinery.
- Cudney 68 Cudney, Gene R., "Stress Histories of Highway Bridges", *Journal of the Structural Division*, ST 12, American Society of Civil Engineers, December, 1968.
- Davis 82 Davis, R., *Expert Systems: Where Are We? And Where Do We Go From Here?*, A. I Memo No. 665, MIT Artificial Intelligence Laboratory, June 1982.
- deJonge 76 deJonge, J. B., *The Requirement of Damage Tolerance*, NLR TR 77705 L, 1976.
- deKleer 75 de Kleer, J. "Qualitative and Quantitative Reasoning in Classical Mechanics.", Technical Report MIT-AI-TR-352, Massachusetts Institute of Technology, 1975.
- deKleer 79 de Kleer, J. "Causal and Teleological Reasoning in Circuit Recognition.", Technical Report MIT-AI-TR-529, Massachusetts Institute of Technology, 1979.
- deKleer&Brown 85 de Kleer, Johan, and John Seely Brown, "A Qualitative Physics Based on Confluences", Bobrow, Daniel G., editor, *Qualitative Reasoning About Physical Systems*, The MIT Press, Cambridge, MA, 1985.
- Denyer 82 Denyer, Anthony G., "Aircraft Structural Maintenance Recommendations Based on Fracture Mechanics Analysis", *Case Histories Involving Fatigue and Fracture Mechanics*, ASTM STP 918, C. M. Hudson and T. P. Rich, Eds., American Society for

- Testing and Materials, Philadelphia, 1986.
- Dvorak 86 Dvorak, D., *QSIM a la PROLOG*, unpublished, Department of Computer Sciences, University of Texas at Austin, Austin, Texas, 78712, Dec 1986.
- ECCS 85 European Convention for Constructional Steelwork, Technical Committee 6 - Fatigue, *Recommendations for the Fatigue Design of Steel Structures*, 1985.
- ENR 83 Engineering News Record, *Failed pin assembly dropped span*, ENR, July 7, 1983.
- Farach 86 Farach Zaldivar, Wadi Nicolas, *Preliminary development of a knowledge based system for the condition evaluation of sector gates*, Master's Thesis, Civil Engineering Department, Massachusetts Institute of Technology, Cambridge, MA, 1986.
- Fisher 81 Fisher, John W. *Inspecting Steel Bridges for Fatigue Damage*, Fritz Engineering Laboratory Report 386-15(81), Lehigh University, Bethlehem, PA, 1981.
- Fisher 84 Fisher, J. W., *Fatigue and Fracture in Steel Bridges: Case Studies*, John Wiley & Sons, New York, 1984.
- Fisher et al. 87 Fisher, J. W., E. J. Kaufmann, C. N. Kostem, J. J. Lee, D. Moser, P. G. Papavizas, and B. T. Yen, *Deformation Induced Cracking In Steel-Girder Bridges and Retrofit Guidelines*, FHWA-PA-86-030+83-21, National Technical Information Service, Springfield, VA, July, 1987.
- Fisher&Mertz 84 Fisher, John W., and Dennis R. Mertz, "Fatigue and Fracture in Steel Bridges", *Proceedings of the International Bridge Conference*, Pittsburgh, PA, 1984.
- Fisher&Mertz 85 Fisher, John W., and Dennis R. Mertz, "Hundreds of Bridges -- Thousands of Cracks", *Civil Engineering*, American Society of Civil Engineers, New York, April, 1985.
- Fisher&Yuceoglu 81 Fisher, John W. and Umur Yuceoglu *Fatigue and Fracture in Steel bridges: Case Studies*, Fritz Engineering Laboratory Report 448-2(81), Lehigh University, Bethlehem, PA, 1981.

- Forbus 81 Forbus, K. *A Study of Qualitative and Geometric Knowledge in Reasoning about Motion*. Technical Report MIT-AI-TR-615, Massachusetts Institute of Technology, 1981.
- Forbus 84 Forbus, K. *Qualitative Process Theory*. Technical Report MIT-AI-TR-789, Massachusetts Institute of Technology, 1984.
- Fuchs&Stephens 80 Fuchs, H. O., and R. I. Stephens, *Metal Fatigue in Engineering*, John Wiley & Sons, New York, 1980.
- Galambos 79 Galambos, Charles F., "Highway bridge loadings", *Engineering Structures*, Vol. 1, October, 1979.
- Gallagher et al. 84 Gallagher, J. P., F. J. Giessler, A. P. Berens, and R. M. Engle Jr., *USAF Damage Tolerant Design Handbook*, AFWAL-TR-82-3073, Air Force Wright Aeronautics Laboratories, Wright-Patterson Air Force Base, Ohio, May 1984.
- Gevarter 87 Gevarter, William B., "The Nature and Evaluation of Commercial Expert System Building Tools", *IEEE Computer*, Volume 20, Number 5, May 1987.
- Goranson et al. 82 Goranson, U. G., J. Hall, J. R. Maclin, and R. T. Watanabe, "Long-Life Damage Tolerant Jet Transport Structures", *Design of Fatigue and Fracture Resistant Structures*, ASTM STP 761, P. R. Abelkis and C. M. Hudson, Eds., American Society for Testing and Materials, Philadelphia, 1982.
- Hammond 84 Hammond, P., "micro-PROLOG for Expert Systems", *micro-PROLOG: Programming in Logic*, by K. L. Clark and F. G. McCabe, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
- Hanson 84 Hanson, John M., *Bridge Failures - Aftermath and Investigation*, T R News, Number 112, May-June, 1984.
- Hayes-Roth et al 83 Hayes-Roth, Frederick, Donald A. Waterman, and Douglas B. Lenat, *Building Expert Systems*, Addison-Wesley, Reading, MA, 1983.
- Hirt et al. 71 Hirt, Manfred A., Ben T. Yen, and John W. Fisher, "Fatigue Strength of Rolled and Welded Steel Beams", *Journal of the Structural Division*, American Society of Civil Engineers, ST7, July, 1971.

- Hirt&Fisher 73 Hirt, Manfred A., and John W. Fisher, "Fatigue Crack Growth in Welded Beams", *Engineering Fracture Mechanics*, Vol 5, Pergamon Press, 1973.
- Hudson&Rich 86 Hudson, C. M., and T. P. Rich, Eds., *Case Histories Involving Fatigue and Fracture Mechanics*, ASTM STP 918, American Society for Testing and Materials, Philadelphia, 1986.
- Hultage et al. 86 Hulthage, Ingemar, Michael D. Rychener, Mark S. Fox, and Martha L. Farinacci, "The Use of Quantitative Databases in Aladin, an Alloy Design System", in *Coupling Symbolic and Numerical Computing in Expert Systems*, J. S. Kowalik (Editor), Elsevier Science Publishers B. V. (North-Holland), 1986.
- Illinois 86 "To Build a Better Bridge", *Engineering Outlook*, University of Illinois at Urbana-Champaign, College of Engineering, Vol 28 No 5, December 1986.
- Iwasaki 88 Iwasaki, Yumi, "Causal Ordering in a Mixed Structure", *Proceedings AAAI-88 Seventh National Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, San Mateo, CA, 1988.
- Iwasaki&Simon 86 Iwasaki, Y. and H.A. Simon, "Causality in device behavior.", *Artificial Intelligence*, 29, 1986, pages 3-32.
- James 88 James, John R., "Lessons Learned in Coordinating Symbolic and Numeric Computing in Knowledge-Based Systems for Control Design", *Coupling Symbolic and Numerical Computing in Expert Systems, II*, edited by J. S. Kowalik and C.T.Kitzmilller, Elsevier Science Publishers B. V. , North-Holland, 1988.
- Janowski 87 Janowski, Rick, "An introduction to QSIM and qualitative simulation", *Artificial Intelligence in Engineering*, Computational Mechanics, 1987.
- Kanninen&Popelar 85 Kanninen, M. F., and C. H. Popelar, *Advanced Fracture Mechanics*, Oxford University Press, New York, 1985.
- Kitzmilller&Kowalik 87 Kitzmilller, C. T., and J. S. Kowalik, "Coupling Symbolic and Numeric Computing in Knowledge-Based Systems", *AI Magazine*, Vol 8 No 2, American Association for Artificial Intelligence, Menlo Park, CA, Summer 1987.

- Kuipers 85 Kuipers, Benjamin, "Commonsense Reasoning about Causality: Deriving Behavior from Structure", Bobrow, Daniel G., editor, *Qualitative Reasoning About Physical Systems*, The MIT Press, Cambridge, MA, 1985.
- Kuipers 86 Kuipers, B., "Qualitative Simulation", *Artificial Intelligence*, Vol. 29, No. 2, March 1986.
- Kuipers 87 Kuipers, Benjamin, "Qualitative Simulation as Causal Explanation", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol SMC-17, No. 3 May/June, 1987.
- Kuipers&Berleant 88 "Using Incomplete Quantitative Knowledge in Qualitative Reasoning", *Proceedings AAAI-88 Seventh National Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, San Mateo, CA, 1988.
- Kuipers&Patil 87 Kuipers, Benjamin, and Ramesh Patil, Qualitative Simulation and Causal Models, Tutorial No. HA 4, Sixth National Conference on Artificial Intelligence, July, 1987.
- Kulicki&Mertz 87 Kulicki, John M., and Dennis R. Mertz, *Case Studies of Displacement-Induced Fatigue*, Sixth Annual Structures Congress, 1987.
- Maddox 74 Maddox, S. J., "Assessing the Significance of Flaws in Welds Subject to Fatigue", *Welding Journal*, American Welding Society, September, 1974.
- Maher 87 Maher, Mary Lou, editor, *Expert Systems for Civil Engineers*, American Society of Civil Engineers, New York, 1987.
- Morelli 87 Theresa Morelli, "Ramp girder cracks in two", *Engineering News Record*, September 24, 1987.
- Morrill&Wright 86 Morrill, J. P. and D. Wright, "A knowledge-based program for diagnosing metallurgical failures.", *Computers & Structures*, Vol. 24, No. 2, pp. 305-311, 1986.
- MSC/pal 88 MSC/pal 1.98, MacNeal-Schwendler Corp, 815 Colorado Boulevard, Los Angeles, CA, 1988.
- Mulvey 87 Mulvey, Lisa, *Methodologies in Innovative Structural Design*, Master's Thesis, Civil Engineering Department, Massachusetts Institute of Technology, Cambridge, MA, September, 1987.

- NCEE 84 National Council of Engineering Examiners, *Principles and Practice of Engineering (PE) Item Writer's Guide 9th Revision*, November 1984.
- NCHRP 87 Moses, Fred, Charles G. Schilling, and K. S. Raju, National Cooperative Highway Research Program, *Fatigue Evaluation Procedures for Steel Bridges*, Project 12/28(3) Final Report, Transportation Research Board, Washington, DC, 1987.
- NCHRP102 70 Fisher, John W., Karl H. Frank, Manfred A. Hirt, and Bernard M. McNamee, *Effect of Weldments on the Fatigue Strength of Steel Beams*, National Cooperative Highway Research Program Report 102, Transportation Research Board, Washington, DC, 1970.
- NCHRP188 78 Schilling, C. G., K. H. Klippstein, J. M. Barsom, and G. T. Blake, *Fatigue of Welded Steel Bridge Members Under Variable-Amplitude Loadings*, National Cooperative Highway Research Program Report 188, Transportation Research Board, Washington, DC, 1978.
- NCHRP201 79 Benter, Jr., W. P., and C. G. Schilling, *Acceptance Criteria for Electorslag Weldments in Bridges*, National Cooperative Highway Research Program Report 201, Transportation Research Board, Washington, DC, 1979.
- NCHRP206 79 Fisher, John W., Hans Hausammann, Michael D. Sullivan, and Alan W. Pense, *Detection and Repair of Fatigue Damage in Welded Highway Bridges*, National Cooperative Highway Research Program Report 206, Transportation Research Board, Washington, DC, 1979.
- NCHRP227 80 Fisher, J. W., B. M. Barthelemy, D. R. Mertz, and J. A. Edinger, *Fatigue Behavior of Full-Scale Welded Bridge Attachments*, National Cooperative Highway Research Program Report 102, Transportation Research Board, Washington, DC, 1980.
- NCHRP242 81 Jessop, T. J, P. J. Mudge, and J. D. Harrison, *Ultrasonic Measurement of Weld Flaw Size*, National Cooperative Highway Research Program Report 242, Transportation Research Board, Washington, DC, 1981.
- NCHRP267 83 Fisher, J. W., D. R. Mertz, and A. Zhong, *Steel Bridge Members*

- Under Variable Amplitude Long Life Fatigue Loading*, National Cooperative Highway Research Program Report 267, Transportation Research Board, Washington, DC, 1983.
- NCHRP286 86 Keating, P. B., and J. W. Fisher, *Evaluation of Fatigue Tests and Design Criteria on Welded Details*, National Cooperative Highway Research Program Report 286, Transportation Research Board, Washington, DC, 1986.
- Nicklaus et al 88 Nicklaus, Dennis J., Karen S. Overton, Siu S. Tong, and Carol J. Russo, "Knowledge Representation and Technique for Engineering Design Automation", in *Coupling Symbolic and Numerical Computing in Expert Systems, II*, J. S. Kowalik and C. T. Kitzmiller (Editors), Elsevier Science Publishers B. V. (North-Holland), 1988.
- Noack 88 Noack, Carlos, *Program and Expert System Development for Tutorial Applications in Engineering Geology*, Master's Thesis, Civil Engineering Department, Massachusetts Institute of Technology, Cambridge, MA, June, 1988.
- Pagnoni 85 Pagnoni, Tommaso, "Expert Systems for Bridge Engineering", IBC-85-34, *Proceedings 2nd Annual International Bridge Conference*, Engineer's Society of Western Pennsylvania, Pittsburgh, PA, 1985.
- Paris&Erdogan 63 Paris, P. C., and F. A. Erdogan, "A Critical Analysis of Crack Propagation Laws", *Transactions of the ASME, Journal of Basic Engineering*, Volume 85, Series D, Number 4, American Society of Mechanical Engineers, December 1963.
- Patil 81 Patil, Ramesh "Causal Representation of Patient Illness for Electrolytic and Acid-Base Diagnosis", Massachusetts Institute of Technology, Laboratory for Computer Science, Technical Report 267, 1981.
- Patil et al 82 Patil, R. S., P. Szolivits, and W. B. Schwartz, "Causal Understanding of Patient Illness in Medical Diagnosis", *Readings in Medical Artificial Intelligence: The First Decade*, W. J. Clancey and E. H. Shortliffe editors, Addison-Wesley, 1982.

- Polya 73 Polya, G, *How to Solve It*, Princeton University Press, New Jersey, 1973.
- Roberts et al. 77 Roberts, R., J. M. Barsom, S. T. Rolfe, and J. W. Fisher, *Fracture Mechanics for Bridge Design*, Department of Transportation, Federal Highway Administration, Washington, DC, 1977.
- Roddis 86 Roddis, W. M. Kim, "Putting Windows in the Black Box: Computer-Aided Instruction of Steel Member Design", *National Education Computing Conference 86*, International Council on Computers in Education, San Diego, CA, June 4-6, 1986.
- Ross 84 Ross, Steven S., *Construction Disasters: Design Failures, Causes, and Prevention*, McGraw-Hill Book Company, New York, 1984.
- Rudd et al. 82 Rudd, J. L., J. N. Yang, S. D., Manning, and W. R. Garver, "Durability Design Requirements and Analysis for Metallic Airframes", *Design of Fatigue and Fracture Resistant Structures, ASTM STP 761*, P. R. Abelkis and C. M. Hudson, Eds., American Society for Testing and Materials, Philadelphia, 1982.
- Ryder et al. 85 Ryder, D. A., T. J. Davies, I. Brough, and F. R. Hutchings, "General Practice in Failure Analysis", *Metals Handbook*, Ninth Edition, Volume 11, Failure Analysis and Prevention, American Society for Metals, Metals Park, OH, 1985.
- Seymour 86 Seymour, William Joseph, *Preliminary development of a bridge management system with rule based expert system decision support enhancements*, Master's Thesis, Civil Engineering Department, Massachusetts Institute of Technology, Cambridge, MA, 1986.
- Simmons 88 Simmons, Reid G., "A Theory of Debugging Plans and Interpretations", *Proceedings AAAI-88 Seventh National Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, San Mateo, CA, 1988.
- Simmons&Davis 87 Simmons, Reid, and Randal Davis, *Generate, Test and Debug:*

- Combining Associational Rules and Causal Models*", Artificial Intelligence Laboratory, MIT Industrial Liaison Program Report 11-31-87, MIT, 1987.
- Simon 73 Simon, Herbert A., "The Structure of Ill Structured Problems", *Artificial Intelligence*, North-Holland, Number 4, 1973.
- Simon&Newell 58 Simon, Herbert, and Allen Newell, "Heuristic Problem Solving: The Next Advance in Operations Research", *Operations Research*, Vol 6, January-February, 1958.
- Slocum 87 Slocum, Alexander H., Laura Demsetz, David Levy, Bruce Schena, and Andrew Ziegler, *Construction Automation Research at MIT*, Fourth International Symposium on Robotics & Artificial Intelligence in Building Construction, Haifa, Israel, 1987.
- Soh 86 Soh, Chee-Kiong, *Preliminary development of a knowledge based system for the preliminary design of fixed steel offshore jacket structures*, Master's Thesis, Civil Engineering Department, Massachusetts Institute of Technology, Cambridge, MA, 1986.
- Sterling&Shapiro 86 Sterling, L., and E. Shapiro, *The Art of PROLOG*, MIT Press, Cambridge, Massachusetts, 1986.
- Swartout 83 Swartout, W. R., "XPLAIN: a system for creating and explaining expert consulting programs.", *Artificial Intelligence* 21, 1983.
- Swartout&Smoliar 87 Swartout, W.R. and S.W. Smoliar, "On making expert systems more like experts.", *Expert Systems*, Vol. 4, No. 3, August 1987.
- Tada et al. 73 Tada, H., P. C. Paris, and G. R. Irwin, *The Stress Analysis of Cracks Handbook*, Del Research Corp., Hellertown, PA, 1973.
- Taheri&Mufti 87 Taheri, Farid, and Aftab A. Mufti, *FAST; A Computer Aided-Design Program For Fracture Mechanics*, ASCE Annual Conference, 1987.
- Thornton et al. 88 Thornton, Charles H., Richard L. Tomasetti and Leonard M. Joseph, *Lessons From Schoharie Creek*, Civil Engineering, May, 1988.
- Throop 88 Throop, David R., "Getting Rules to Talk to Models", in *Coupling Symbolic and Numerical Computing in Expert Systems, II*, J. S. Kowalik and C. T. Kitzmiller (Editors), Elsevier Science Publishers B. V. (North-Holland), 1988.

- Tubby&Wylde 86 Tubby, P. J., and J. G. Wylde, "Role of Fracture Mechanics in Assessing the Effect on Fatigue Life of Design Changes in Welded Fabrications", *Case Histories Involving Fatigue and Fracture Mechanics*, ASTM STP 918, C. M. Hudson and T. P. Rich, Eds., American Society for Testing and Materials, Philadelphia, 1986.
- USAirForce 74 United States Air Force, Ross, *Airplane Damage Tolerance Requirements*, Military Specification NIL-A-83444 (USAF), 1974.
- USNavy 47 United States Navy, *The Design and Methods of Construction of Welded Steel Merchant Vessels*, Report of an Investigation, 15 July 1946, Government Printing Office, Washington, 1947.
- VanVlack 64 Van Vlack, Lawrence H., *Elements of Materials Science*, Second Edition, Addison-Wesley, Reading, MA, 1964.
- Verma&McNamara 88 Verma, K Krishna, and Michael P. McNamara, "A Conceptual Approach to Prevent Crack-Related Failure of Steel Bridges", *Engineering Journal*, American Institute of Steel Construction, Vol 25, No 1, 1st Quarter, 1988.
- Weiss 86 Weiss, Volker, "Towards Failure Analysis Expert Systems", ASTM Standardization News, April, 1986.
- Wellman 86 Wellman, Michael P., "Reasoning About Assumptions Underlying Mathematical Models", *Coupling Symbolic and Numerical Computing in Expert Systems*, C.T.Kitzmilller (Editor), Elsevier Science Publishers B. V. (North-Holland), 1986.
- Zettlemoyer&Fisher 77 Zettlemoyer, N., and J. W. Fisher, "Stress Gradient Correction Factor for Stress Intensity at Welded Stiffeners and Cover Plates", *Welding Journal*, American Welding Society, December 1977.
- Zettlemoyer&Fisher 79 Zettlemoyer, N., and J. W. Fisher, *The Prediction of Fatigue Strength of Welded Details*, Fritz Engineering Laboratory Report No. 386-10(79), Lehigh University, Bethlehem, PA, January, 1979.

Appendix A:

CRACK Test Cases:

Design critique: experimental program [NCHRP102 70]

Coverplate on welded beam: 6, 10, and 20 ksi stress range

Plain welded beam : 6, 10, and 20 ksi stress range

Plain rolled beam: 6, 10, and 20 ksi stress range

Predictive modeling: case studies [Fisher 84]

Yellow Mill Pond: end welded cover plate

Polk County: floor beam web gap

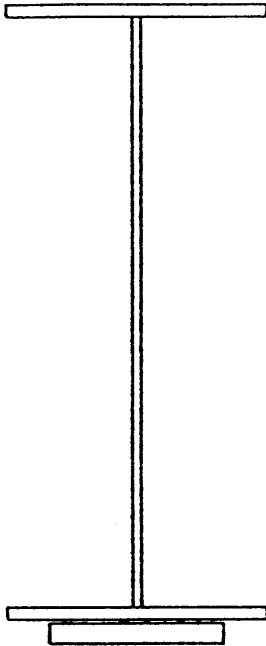
Failure analysis: case studies [Fisher 84]

Belle Fourche River: diaphragm web gap

Cuyahoga County: stiffener web gap

Cuyahoga County: stiffener without web gap

**Coverplate on welded beam:
cross-section and confirmed facts**



was_confirmed(problem_title('NCHRP 102 CW Sr=6')).
was_confirmed(ask_classification('design critique')).
was_confirmed(topflange_width(6.67)).
was_confirmed(topflange_thick(0.37)).
was_confirmed(topflange_matlname(a36)).
was_confirmed(topflange_fracture_toughness(_1)).
was_confirmed(topflange_yield_strength(33.88)).
was_confirmed(web_width(13.82)).
was_confirmed(web_thick(0.267)).
was_confirmed(web_matlname(a36)).
was_confirmed(web_fracture_toughness(_1)).
was_confirmed(web_yield_strength(37.65)).
was_confirmed(botflange_width(6.67)).
was_confirmed(botflange_thick(0.37)).
was_confirmed(botflange_matlname(a36)).
was_confirmed(botflange_fracture_toughness(_1)).
was_confirmed(botflange_yield_strength(33.88)).
was_confirmed(topflange_conn_type(welded)).
was_confirmed(topflange_conn_size(0.188)).
was_confirmed(botflange_conn_type(welded)).
was_confirmed(botflange_conn_size(0.188)).
was_confirmed(attachments(botcoverpl)).
was_confirmed(botcoverpl_width(4.49)).
was_confirmed(botcoverpl_thick(0.54)).
was_confirmed(botcoverpl_matlname(a36)).
was_confirmed(botcoverpl_fracture_toughness(_1)).
was_confirmed(botcoverpl_yield_strength(33.76)).
was_confirmed(coverplate_conn(end_fillet_weld)).
was_confirmed(coverplate_conn_size(0.25)).
was_confirmed(miner_stress_range(6)).
was_confirmed(ask_max_total_stress(unknown)).
was_confirmed(ask_adtt(500)).
was_confirmed(ask_age(20)).
was_confirmed(min_service_temp_zone(zone1)).

**Coverplate on welded beam: 6 ksi stress range
crack growth**



	N million cycles	a inches	ΔK Δσ=6ksi	K σ=33.88ksi
0	0	0.03	3.814	21.534
1	0.423	0.041	4.076	23.017
2	0.788	0.051	4.284	24.19
3	1.365	0.072	4.613	26.046
4	2.29	0.115	5.132	28.977
5	3.54	0.2	6.213	35.083
6	4.471	0.37	12.386	69.94

Coverplate on welded beam: 6 ksi stress range conclusions

Conclusions for CRACK consultation on NCHRP 102 CW Sr=6 bridge

Task to be done is design critique.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 37.65, 13.82 inches wide and 0.267 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. There is a welded 0.188 inches top flange connection. There is a welded 0.188 inches bottom flange connection. Attachments for the girder are: The bottom coverplate is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.76, 4.49 inches wide and 0.54 inches thick.

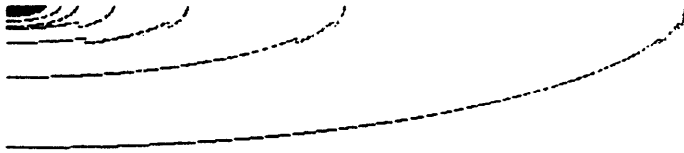
The in-service stress range is 6 ksi at a frequency of 500 Average Daily Truck Traffic for a bridge age of 20 years.

The cracking cause is classified as a low fatigue resistant detail at a end weld of a cover plate type of cracking. The initial flaw size is 0.03 inches. It is thumbnail shaped in the bottom flange.

The probable sequence of crack growth to failure is to grow by fatigue as thumbnail crack in bottom flange from a crack size of 0.03 to 0.37 in 4.471 million cycles.

The predicted fatigue life of 4.471 million cycles is greater than the design life of 3.65 million cycles. The predicted failure mode is by fatigue, not fracture, as is desirable.

**Coverplate on welded beam: 10 ksi stress range
crack growth**



	N million cycles	a inches	ΔK Δσ=10ksi	K σ=33.88ksi
0	0	0.03	6.356	21.534
1	0.091	0.041	6.794	23.017
2	0.17	0.051	7.14	24.19
3	0.295	0.072	7.688	26.046
4	0.495	0.115	8.553	28.977
5	0.766	0.2	10.355	35.083
6	0.966	0.37	20.643	69.94

Coverplate on welded beam: 10 ksi stress range conclusions

Conclusions for CRACK consultation on NCHRP 102 CW Sr=10 bridge

Task to be done is design critique.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 37.65, 13.82 inches wide and 0.267 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. There is a welded 0.188 inches top flange connection. There is a welded 0.188 inches bottom flange connection. Attachments for the girder are: The bottom coverplate is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.76, 4.49 inches wide and 0.54 inches thick.

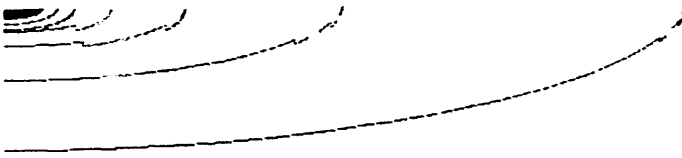
The in-service stress range is 10 ksi at a frequency of 500 Average Daily Truck Traffic for a bridge age of 20 years.

The cracking cause is classified as a low fatigue resistant detail at an end weld of a cover plate type of cracking. The initial flaw size is 0.03 inches. It is thumbnail shaped in the bottom flange.

The probable sequence of crack growth to failure is to grow by fatigue as thumbnail crack in bottom flange from a crack size of 0.03 to 0.37 in 0.966 million cycles.

The predicted fatigue life of 0.966 million cycles is less than the design life of 3.65 million cycles. The design is UNSATISFACTORY. The predicted failure mode is by fatigue, not fracture, as is desirable.

**Coverplate on welded beam: 20 ksi stress range
crack growth**



	N million cycles	a inches	ΔK Δσ=20ksi	K σ=33.88ksi
0	0	0.03	12.712	21.534
1	0.011	0.041	13.587	23.017
2	0.021	0.051	14.28	24.19
3	0.037	0.072	15.376	26.046
4	0.062	0.115	17.106	28.977
5	0.096	0.2	20.71	35.083
6	0.121	0.37	41.287	69.94

Coverplate on welded beam: 20 ksi stress range conclusions

Conclusions for CRACK consultation on NCHRP 102 CW Sr=20 bridge

Task to be done is design critique.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 37.65, 13.82 inches wide and 0.267 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. There is a welded 0.188 inches top flange connection. There is a welded 0.188 inches bottom flange connection. Attachments for the girder are: The bottom coverplate is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.76, 4.49 inches wide and 0.54 inches thick.

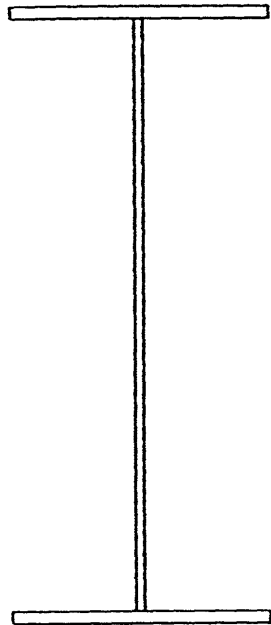
The in-service stress range is 20 ksi at a frequency of 500 Average Daily Truck Traffic for a bridge age of 20 years.

The cracking cause is classified as a low fatigue resistant detail at a end weld of a cover plate type of cracking. The initial flaw size is 0.03 inches. It is thumbnail shaped in the bottom flange.

The probable sequence of crack growth to failure is to grow by fatigue as thumbnail crack in bottom flange from a crack size of 0.03 to 0.37 in 0.121 million cycles.

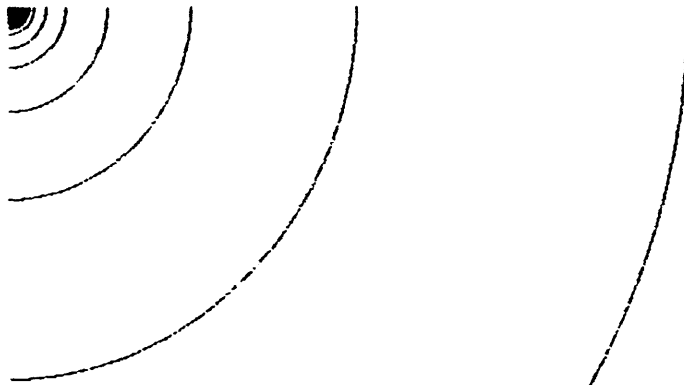
The predicted fatigue life of 0.121 million cycles is less than the design life of 3.65 million cycles. The design is UNSATISFACTORY. The predicted failure mode is by fatigue, not fracture, as is desirable.

Plain welded beam:
cross-section and confirmed facts



was_confirmed(problem_title(NCHRP102 PW Sr=6')).
was_confirmed(ask_classification('design critique')).
was_confirmed(topflange_width(6.67)).
was_confirmed(topflange_thick(0.37)).
was_confirmed(topflange_matlname(a36)).
was_confirmed(topflange_fracture_toughness(_1)).
was_confirmed(topflange_yield_strength(33.8)).
was_confirmed(web_width(13.82)).
was_confirmed(web_thick(0.267)).
was_confirmed(web_matlname(a36)).
was_confirmed(web_fracture_toughness(_1)).
was_confirmed(web_yield_strength(37.65)).
was_confirmed(botflange_width(6.67)).
was_confirmed(botflange_thick(0.37)).
was_confirmed(botflange_matlname(a36)).
was_confirmed(botflange_fracture_toughness(_1)).
was_confirmed(botflange_yield_strength(33.88)).
was_confirmed(topflange_conn_type(welded)).
was_confirmed(topflange_conn_size(0.188)).
was_confirmed(botflange_conn_type(welded)).
was_confirmed(botflange_conn_size(0.188)).
was_confirmed(attachments(none)).
was_confirmed(miner_stress_range(6)).
was_confirmed(ask_max_total_stress(unknown)).
was_confirmed(ask_adtt(10000)).
was_confirmed(ask_age(50)).
was_confirmed(min_service_temp_zone(zone1)).

Plain welded beam: 6 ksi stress range
crack growth



	N million cycles	a inches	ΔK Δσ=6ksi	K σ=33.88ksi
0	0	0.01	0.677	3.823
1	11.399	0.011	0.723	4.083
2	20.865	0.013	0.766	4.327
3	35.804	0.016	0.846	4.779
4	56.214	0.021	0.987	5.573
5	79.718	0.033	1.221	6.892
6	102.686	0.055	1.588	8.966
7	122.406	0.1	2.141	12.09
8	137.94	0.19	2.952	16.669
9	149.56	0.37	4.123	23.284

**Plain welded beam: 6 ksi stress range
conclusions**

Conclusions for CRACK consultation on NCHRP 102 PW Sr=6 bridge

Task to be done is design critique.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 37.65, 13.82 inches wide and 0.267 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. There is a welded 0.188 inches top flange connection. There is a welded 0.188 inches bottom flange connection. There are no attachments.

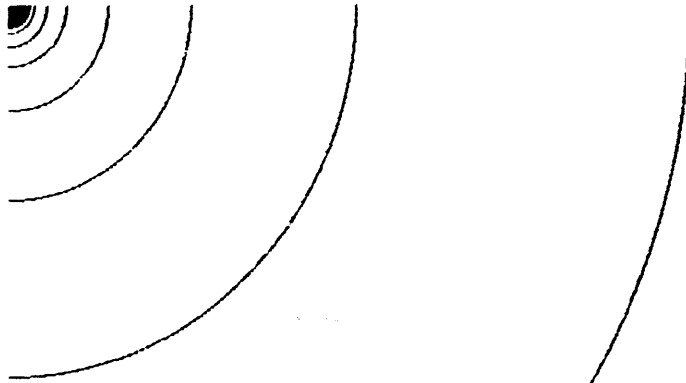
The in-service stress range is 6 ksi at a frequency of 10000 Average Daily Truck Traffic for a bridge age of 50 years.

The cracking cause is classified as a initial flaw in a welded bottom flange connection type of cracking. The initial flaw size is 0.01 inches. It is penny shaped in the bottom flange connection.

The probable sequence of crack growth to failure is to grow by fatigue as penny crack in bottom flange connection from a crack size of 0.01 to 0.37 in 149.56 million cycles.

The predicted fatigue life of 149.56 million cycles is less than the design life of 182.5 million cycles. The design is UNSATISFACTORY. The predicted failure mode is by fatigue, not fracture, as is desirable.

Plain welded beam: 10 ksi stress range
crack growth



	N million cycles	a inches	ΔK Δσ=10ksi	K σ=33.88ksi
0	0	0.01	1.128	3.823
1	2.462	0.011	1.205	4.083
2	4.507	0.013	1.277	4.327
3	7.734	0.016	1.41	4.779
4	12.142	0.021	1.645	5.573
5	17.219	0.032	2.034	6.892
6	22.18	0.055	2.646	8.966
7	26.44	0.1	3.569	12.09
8	29.795	0.19	4.92	16.669
9	32.305	0.37	6.872	23.284

**Plain welded beam: 10 ksi stress range
conclusions**

Conclusions for CRACK consultation on NCHRP 102 PW Sr=10 bridge

Task to be done is design critique.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 37.65, 13.82 inches wide and 0.267 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. There is a welded 0.188 inches top flange connection. There is a welded 0.188 inches bottom flange connection. There are no attachments.

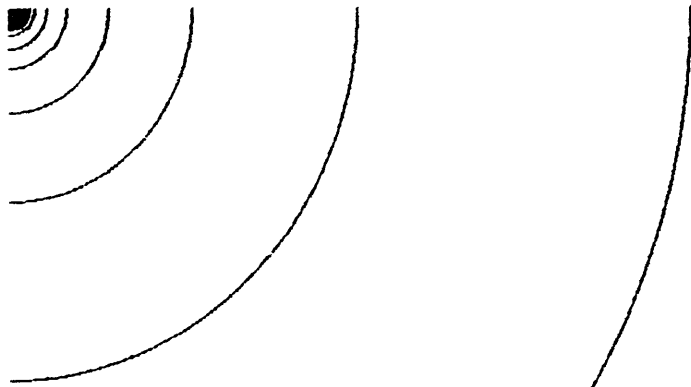
The in-service stress range is 10 ksi at a frequency of 10000 Average Daily Truck Traffic for a bridge age of 50 years.

The cracking cause is classified as a initial flaw in a welded bottom flange connection type of cracking. The initial flaw size is 0.01 inches. It is penny shaped in the bottom flange connection.

The probable sequence of crack growth to failure is to grow by fatigue as penny crack in bottom flange connection from a crack size of 0.01 to 0.37 in 32.305 million cycles.

The predicted fatigue life of 32.305 million cycles is less than the design life of 182.5 million cycles. The design is UNSATISFACTORY. The predicted failure mode is by fatigue, not fracture, as is desirable.

Plain welded beam: 20 ksi stress range
crack growth



	N million cycles	a inches	ΔK Δσ=20ksi	K σ=33.88ksi
0	0	0.01	2.257	3.823
1	0.308	0.011	2.41	4.083
2	0.563	0.013	2.554	4.327
3	0.967	0.016	2.821	4.779
4	1.518	0.021	3.29	5.573
5	2.152	0.032	4.068	6.892
6	2.773	0.055	5.293	8.966
7	3.305	0.1	7.137	12.09
8	3.724	0.19	9.84	16.669
9	4.038	0.37	13.745	23.284

Plain welded beam: 20 ksi stress range conclusions

Conclusions for CRACK consultation on NCHRP 102 PW Sr=20 bridge

Task to be done is design critique.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 37.65, 13.82 inches wide and 0.267 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 33.88, 6.67 inches wide and 0.37 inches thick. There is a welded 0.188 inches top flange connection. There is a welded 0.188 inches bottom flange connection. There are no attachments.

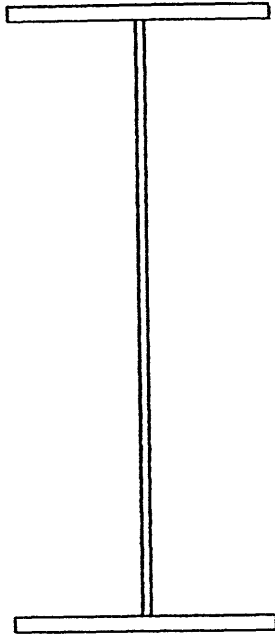
The in-service stress range is 20 ksi at a frequency of 10000 Average Daily Truck Traffic for a bridge age of 50 years.

The cracking cause is classified as a initial flaw in a welded bottom flange connection type of cracking. The initial flaw size is 0.01 inches. It is penny shaped in the bottom flange connection.

The probable sequence of crack growth to failure is to grow by fatigue as penny crack in bottom flange connection from a crack size of 0.01 to 0.37 in 4.038 million cycles.

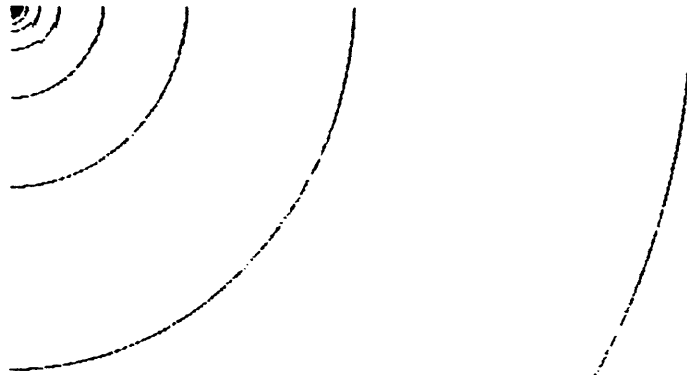
The predicted fatigue life of 4.038 million cycles is less than the design life of 182.5 million cycles. The design is UNSATISFACTORY. The predicted failure mode is by fatigue, not fracture, as is desirable.

Plain rolled beam:
cross-section and confirmed facts



was_confirmed(problem_title(NCHRP102 PR Sr=6')).
was_confirmed(ask_classification('design critique')).
was_confirmed(topflange_width(6.8)).
was_confirmed(topflange_thick(0.394)).
was_confirmed(topflange_matlname(a36)).
was_confirmed(topflange_fracture_toughness(_1)).
was_confirmed(topflange_yield_strength(34.51)).
was_confirmed(web_width(14.04)).
was_confirmed(web_thick(0.286)).
was_confirmed(web_matlname(a36)).
was_confirmed(web_fracture_toughness(_1)).
was_confirmed(web_yield_strength(39.31)).
was_confirmed(botflange_width(6.8)).
was_confirmed(botflange_thick(0.394)).
was_confirmed(botflange_matlname(a36)).
was_confirmed(botflange_fracture_toughness(_1)).
was_confirmed(botflange_yield_strength(34.51)).
was_confirmed(topflange_conn_type(rolled)).
was_confirmed(botflange_conn_type(rolled)).
was_confirmed(attachments(none)).
was_confirmed(miner_stress_range(6)).
was_confirmed(ask_max_total_stress(unknown)).
was_confirmed(ask_adtt(10000)).
was_confirmed(ask_age(50)).
was_confirmed(min_service_temp_zone(zone1)).

Plain rolled beam: 6 ksi stress range
crack growth



	N million cycles	a inches	ΔK Δσ=6ksi	K σ=34.51ksi
0	0	0.005	0.479	2.754
1	17.288	0.006	0.514	2.955
2	31.459	0.007	0.547	3.144
3	53.51	0.008	0.607	3.491
4	83.088	0.011	0.713	4.099
5	116.499	0.017	0.887	5.1
6	148.612	0.029	1.159	6.667
7	175.867	0.054	1.568	9.018
8	197.186	0.102	2.165	12.453
9	213.083	0.199	3.025	17.399
10	224.622	0.394	4.256	24.476

Plain rolled beam: 6 ksi stress range conclusions

Conclusions for CRACK consultation on NCHRP 102 PR Sr=6 bridge

Task to be done is design critique.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 34.51, 6.8 inches wide and 0.394 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 39.31, 14.04 inches wide and 0.286 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 34.51, 6.8 inches wide and 0.394 inches thick. There is a rolled top flange connection. There is a rolled bottom flange connection. There are no attachments.

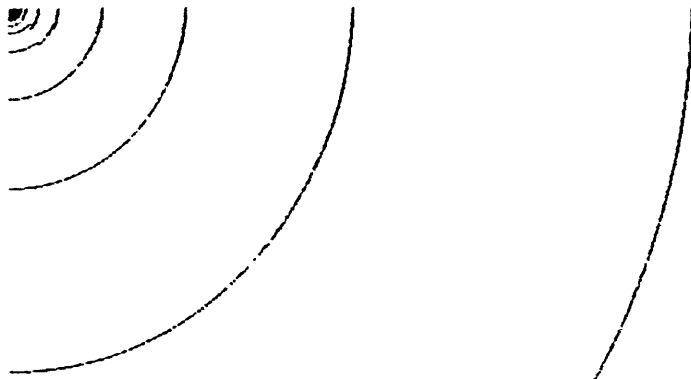
The in-service stress range is 6 ksi at a frequency of 10000 Average Daily Truck Traffic for a bridge age of 50 years.

The cracking cause is classified as a initial flaw in a rolled bottom flange connection type of cracking. The initial flaw size is 0.005 inches. It is penny shaped in the bottom flange connection.

The probable sequence of crack growth to failure is to grow by fatigue as penny crack in bottom flange connection from a crack size of 0.005 to 0.394 in 224.622 million cycles.

The predicted fatigue life of 224.622 million cycles is greater than the design life of 182.5 million cycles. The predicted failure mode is by fatigue, not fracture, as is desirable.

Plain rolled beam: 10 ksi stress range
crack growth



	N million cycles	a inches	ΔK Δσ=10ksi	K σ=34.51ksi
0	0	0.005	0.798	2.754
1	3.734	0.006	0.856	2.955
2	6.795	0.007	0.911	3.144
3	11.558	0.008	1.012	3.491
4	17.947	0.011	1.188	4.099
5	25.164	0.017	1.478	5.1
6	32.1	0.029	1.932	6.667
7	37.987	0.054	2.613	9.018
8	42.592	0.102	3.609	12.453
9	46.026	0.199	5.042	17.399
10	48.518	0.394	7.093	24.476

Plain rolled beam: 10 ksi stress range conclusions

Conclusions for CRACK consultation on NCHRP 102 PR Sr=10 bridge

Task to be done is design critique.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 34.51, 6.8 inches wide and 0.394 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 39.31, 14.04 inches wide and 0.286 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 34.51, 6.8 inches wide and 0.394 inches thick. There is a rolled top flange connection. There is a rolled bottom flange connection. There are no attachments.

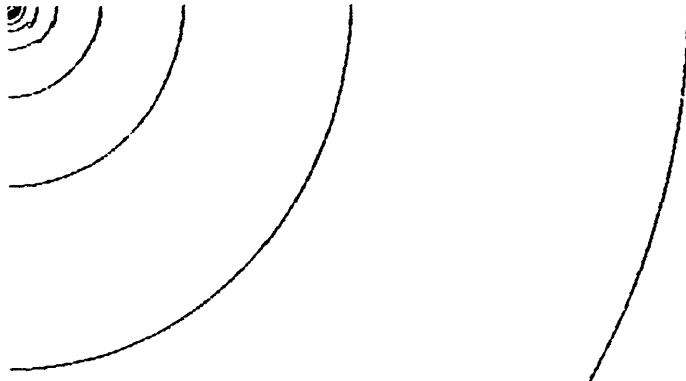
The in-service stress range is 10 ksi at a frequency of 10000 Average Daily Truck Traffic for a bridge age of 50 years.

The cracking cause is classified as a initial flaw in a rolled bottom flange connection type of cracking. The initial flaw size is 0.005 inches. It is penny shaped in the bottom flange connection.

The probable sequence of crack growth to failure is to grow by fatigue as penny crack in bottom flange connection from a crack size of 0.005 to 0.394 in 48.518 million cycles.

The predicted fatigue life of 48.518 million cycles is less than the design life of 182.5 million cycles. The design is UNSATISFACTORY. The predicted failure mode is by fatigue, not fracture, as is desirable.

Plain rolled beam: 20 ksi stress range
crack growth



	N million cycles	a inches	ΔK Δσ=20ksi	K σ=34.51ksi
0	0	0.005	1.596	2.754
1	0.467	0.006	1.713	2.955
2	0.849	0.007	1.822	3.144
3	1.445	0.008	2.023	3.491
4	2.243	0.011	2.375	4.099
5	3.145	0.017	2.956	5.1
6	4.013	0.029	3.864	6.667
7	4.748	0.054	5.226	9.018
8	5.324	0.102	7.217	12.453
9	5.753	0.199	10.083	17.399
10	6.065	0.394	14.185	24.476

Plain rolled beam: 20 ksi stress range conclusions

Conclusions for CRACK consultation on NCHRP 102 PR Sr=20 bridge

Task to be done is design critique.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 34.51, 6.8 inches wide and 0.394 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 39.31, 14.04 inches wide and 0.286 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 34.51, 6.8 inches wide and 0.394 inches thick. There is a rolled top flange connection. There is a rolled bottom flange connection. There are no attachments.

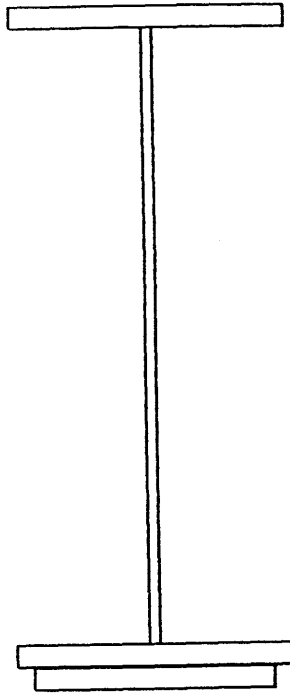
The in-service stress range is 20 ksi at a frequency of 10000 Average Daily Truck Traffic for a bridge age of 50 years.

The cracking cause is classified as a initial flaw in a rolled bottom flange connection type of cracking. The initial flaw size is 0.005 inches. It is penny shaped in the bottom flange connection.

The probable sequence of crack growth to failure is to grow by fatigue as penny crack in bottom flange connection from a crack size of 0.005 to 0.394 in 6.065 million cycles.

The predicted fatigue life of 6.065 million cycles is less than the design life of 182.5 million cycles. The design is UNSATISFACTORY. The predicted failure mode is by fatigue, not fracture, as is desirable.

Yellow Mill Pond: cross-section, confirmed facts, and crack growth



```

was_confirmed(problem_title('Yellow Mill Pond')).
was_confirmed(task_classification('predictive modeling')).
was_confirmed(topflange_width(16.5)).
was_confirmed(topflange_thick(1.25)).
was_confirmed(topflange_matlname(a242)).
was_confirmed(topflange_fracture_toughness(80)).
was_confirmed(topflange_yield_strength(_1)).
was_confirmed(web_width(33.375)).
was_confirmed(web_thick(0.75)).
was_confirmed(web_matlname(a242)).
was_confirmed(web_fracture_toughness(80)).
was_confirmed(web_yield_strength(_1)).
was_confirmed(botflange_width(16.5)).
was_confirmed(botflange_thick(1.25)).
was_confirmed(botflange_matlname(a242)).
was_confirmed(botflange_fracture_toughness(80)).
was_confirmed(botflange_yield_strength(_1)).
was_confirmed(topflange_conn_type(rolled)).
was_confirmed(botflange_conn_type(rolled)).
was_confirmed(attachments(botcoverpl)).
was_confirmed(botcoverpl_width(14.5)).
was_confirmed(botcoverpl_thick(1.25)).
was_confirmed(botcoverpl_matlname(a242)).
was_confirmed(botcoverpl_fracture_toughness(80)).
was_confirmed(botcoverpl_yield_strength(_1)).
was_confirmed(coverplate_conn_end_fillet_weld)).
was_confirmed(coverplate_conn_size(0.63)).
was_confirmed(miner_stress_range(2)).
was_confirmed(task_max_total_stress(unknown)).
was_confirmed(task_adtt(9360)).
was_confirmed(task_age(18.5)).
was_confirmed(min_service_temp_zone(zone2)).
    
```



	N million cycles	a inches	ΔK Δσ=2ksi	K σ=50ksi
0	0	0.03	1.808	45.2
1	3.634	0.04	1.945	48.636
2	6.687	0.049	2.055	51.377
3	11.441	0.068	2.225	55.62
4	18.922	0.106	2.46	61.503
5	29.447	0.183	2.759	68.986
6	43.825	0.335	3.145	78.632
7	63.304	0.64	3.84	95.996
8	77.15	1.25	7.843	196.067

Yellow Mill Pond: conclusions

Conclusions for CRACK consultation on Yellow Mill Pond bridge

Task to be done is predictive modeling.

The girder top flange is a plate of ASTM a242, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 50, 16.5 inches wide and 1.25 inches thick. The girder web is a plate of ASTM a242, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 50, 33.375 inches wide and 0.75 inches thick. The girder bottom flange is a plate of ASTM a242, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 50, 16.5 inches wide and 1.25 inches thick. There is a rolled top flange connection. There is a rolled bottom flange connection. Attachments for the girder are: The bottom coverplate is a plate of ASTM a242, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 50, 14.5 inches wide and 1.25 inches thick.

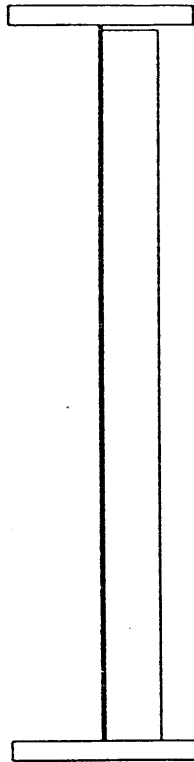
The in-service stress range is 2 ksi at a frequency of 9360 Average Daily Truck Traffic for a bridge age of 18.5 years.

The cracking cause is classified as a low fatigue resistant detail at a end weld of a cover plate type of cracking. The initial flaw size is 0.03 inches. It is thumbnail shaped in the bottom flange.

The probable sequence of crack growth to failure is: 1. fatigue growth as thumbnail crack in bottom flange from a crack size of 0.03 inches to 0.64 inches in 63.304 million cycles. 2. fracture as thumbnail crack in bottom flange at a crack size of 0.64 inches at a stress intensity of 95.996 ksi√inches.

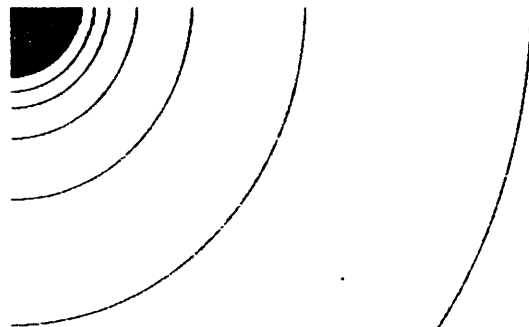
The predicted fatigue life of 63.304 million cycles is in good agreement with the estimated service life of 63.203 million cycles. For a service life of 63.304 million cycles, close to the current life of 63.203 million cycles, a crack size of approximately 0.64 inches is expected.

Polk County: cross-section, confirmed facts, and crack growth



```

was_confirmed(problem_title(Polk County)).
was_confirmed(ask_classification(predictive modeling)).
was_confirmed(topflange_width(21)).
was_confirmed(topflange_thick(2.25)).
was_confirmed(topflange_matlname(a36)).
was_confirmed(topflange_fracture_toughness(_1)).
was_confirmed(topflange_yield_strength(_1)).
was_confirmed(web_width(74)).
was_confirmed(web_thick(0.438)).
was_confirmed(web_matlname(a36)).
was_confirmed(web_fracture_toughness(_1)).
was_confirmed(web_yield_strength(_1)).
was_confirmed(botflange_width(21)).
was_confirmed(botflange_thick(2.25)).
was_confirmed(botflange_matlname(a36)).
was_confirmed(botflange_fracture_toughness(_1)).
was_confirmed(botflange_yield_strength(_1)).
was_confirmed(topflange_conn_type(welded)).
was_confirmed(topflange_conn_size(0.5)).
was_confirmed(botflange_conn_type(welded)).
was_confirmed(botflange_conn_size(0.5)).
was_confirmed(attachments(floorbeam_plate)).
was_confirmed(floorbeam_plate_width(73.5)).
was_confirmed(floorbeam_plate_thick(0.75)).
was_confirmed(floorbeam_plate_matlname(a36)).
was_confirmed(floorbeam_plate_fracture_toughness(_1)).
was_confirmed(floorbeam_plate_yield_strength(_1)).
was_confirmed(floorbeam_conn_size(0.25)).
was_confirmed(positive_attachment(no)).
was_confirmed(miner_stress_ranget(10.4)).
was_confirmed(ask_max_total_stress(unknown)).
was_confirmed(ask_adtt(500)).
was_confirmed(ask_age(17)).
was_confirmed(min_service_temp_zone(zone2)).
    
```



	N million cycles	a inches	ΔK Δσ=10.4ksi	K σ=36ksi
0	0	0.03	3.071	10.632
1	0.494	0.036	3.275	11.337
2	0.911	0.042	3.458	11.969
3	1.568	0.054	3.779	13.081
4	2.524	0.077	4.312	14.925
5	3.728	0.124	5.175	17.913
6	4.998	0.219	6.793	23.514

Polk County: conclusions

Conclusions for CRACK consultation on Polk County bridge

Task to be done is predictive modeling.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 50 and a yield strength (ksi) of 36, 21 inches wide and 2.25 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 50 and a yield strength (ksi) of 36, 74 inches wide and 0.438 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 50 and a yield strength (ksi) of 36, 21 inches wide and 2.25 inches thick. There is a welded 0.5 inches top flange connection. There is a welded 0.5 inches bottom flange connection. Attachments for the girder are: The floorbeam connection plate is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 50 and a yield strength (ksi) of 36, 73.5 inches wide and 0.75 inches thick.

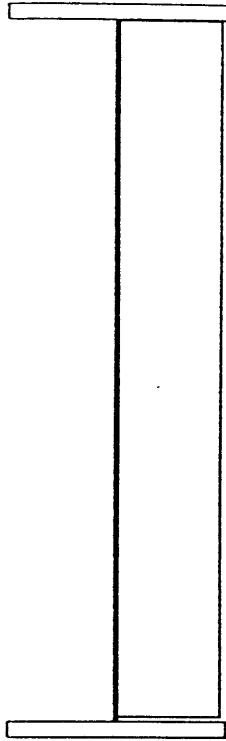
The in-service stress range is 10.4 ksi at a frequency of 500 Average Daily Truck Traffic for a bridge age of 17 years.

The cracking cause is classified as a out of plane distortion at a web gap type of cracking. The initial flaw size is 0.03 inches. It is thumbnail shaped in the web.

The probable sequence of crack growth to failure is to grow by fatigue as thumbnail crack in web from a crack size of 0.03 inches to 0.219 inches in 4.998 million cycles.

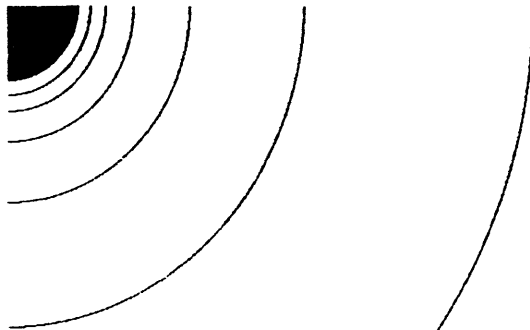
The predicted fatigue life of 4.998 million cycles is in reasonable agreement with the estimated service life of 3.102 million cycles. For a service life of 3.728 million cycles, close to the current life of 4.998 million cycles, a crack size of approximately 0.124 inches is expected.

Belle Fourche River: cross-section, confirmed facts, and crack growth



```

was_confirmed(problem_title('Belle Fourche River')).
was_confirmed(ask_classification('failure analysis')).
was_confirmed(topflange_width(20)).
was_confirmed(topflange_thick(1.5)).
was_confirmed(topflange_matlname(a373)).
was_confirmed(topflange_fracture_toughness(_1)).
was_confirmed(topflange_yield_strength(_1)).
was_confirmed(web_width(58)).
was_confirmed(web_thick(0.438)).
was_confirmed(web_matlname(a373)).
was_confirmed(web_fracture_toughness(_1)).
was_confirmed(web_yield_strength(_1)).
was_confirmed(botflange_width(20)).
was_confirmed(botflange_thick(1.5)).
was_confirmed(botflange_matlname(a373)).
was_confirmed(botflange_fracture_toughness(_1)).
was_confirmed(botflange_yield_strength(_1)).
was_confirmed(topflange_conn_type(welded)).
was_confirmed(topflange_conn_size(0.438)).
was_confirmed(botflange_conn_type(welded)).
was_confirmed(botflange_conn_size(0.438)).
was_confirmed(attachments(diaphragm_plate)).
was_confirmed(diaphragm_plate_width(57.25)).
was_confirmed(diaphragm_plate_thick(0.375)).
was_confirmed(diaphragm_plate_matlname(a373)).
was_confirmed(diaphragm_plate_fracture_toughness(_1)).
was_confirmed(diaphragm_plate_yield_strength(_1)).
was_confirmed(diaphragm_conn_size(0.25)).
was_confirmed(positive_attachment(no)).
was_confirmed(surface_features(no)).
was_confirmed(miner_stress_range(10)).
was_confirmed(ask_max_total_stress(unknown)).
was_confirmed(ask_adtt(280)).
was_confirmed(ask_age(30)).
was_confirmed(min_service_temp_zone(zone1)).
    
```



	N million cycles	a inches	ΔK Δσ=10ksi	K σ=32ksi
0	0	0.03	2.953	9.45
1	0.556	0.036	3.149	10.077
2	1.024	0.042	3.325	10.639
3	1.764	0.054	3.634	11.627
4	2.839	0.077	4.146	13.267
5	4.193	0.125	4.976	15.923
6	5.622	0.219	6.532	20.901

Belle Fourche River: conclusions

Conclusions for CRACK consultation on Belle Fourche River bridge

Task to be done is failure analysis.

The girder top flange is a plate of ASTM a373, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 32, 20 inches wide and 1.5 inches thick. The girder web is a plate of ASTM a373, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 32, 58 inches wide and 0.438 inches thick. The girder bottom flange is a plate of ASTM a373, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 32, 20 inches wide and 1.5 inches thick. There is a welded 0.438 inches top flange connection. There is a welded 0.438 inches bottom flange connection. Attachments for the girder are: The diaphragm connection plate is a plate of ASTM a373, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 32, 57.25 inches wide and 0.375 inches thick.

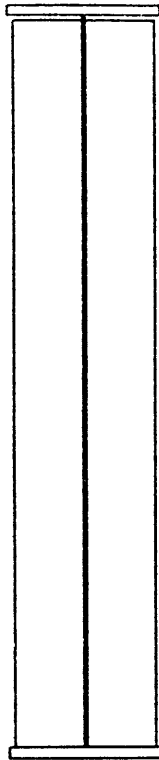
The in-service stress range is 10 ksi at a frequency of 280 Average Daily Truck Traffic for a bridge age of 30 years.

The cracking cause is classified as a out of plane distortion at a web gap type of cracking. The initial flaw size is 0.03 inches. It is thumbnail shaped in the web.

The probable sequence of crack growth to failure is to grow by fatigue as thumbnail crack in web from a crack size of 0.03 inches to 0.219 inches in 5.622 million cycles.

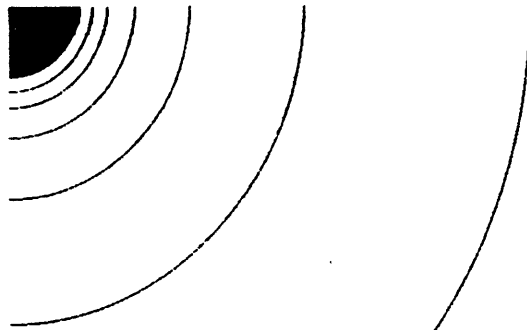
The predicted fatigue life of 5.622 million cycles is in reasonable agreement with the estimated service life of 3.066 million cycles.

Cuyahoga County with web gap: cross-section, confirmed facts, and crack growth



```

was_confirmed(problem_title('Cuyahoga County')).
was_confirmed(task_classification('Failure analysis')).
was_confirmed(topflange_width(16.5)).
was_confirmed(topflange_thick(1.25)).
was_confirmed(topflange_matlname('a36')).
was_confirmed(topflange_fracture_toughness(1)).
was_confirmed(topflange_yield_strength(1)).
was_confirmed(web_width(72)).
was_confirmed(web_thick(0.438)).
was_confirmed(web_matlname('a36')).
was_confirmed(web_fracture_toughness(1)).
was_confirmed(web_yield_strength(46)).
was_confirmed(botflange_width(16.5)).
was_confirmed(botflange_thick(1.25)).
was_confirmed(botflange_matlname('a36')).
was_confirmed(botflange_fracture_toughness(1)).
was_confirmed(botflange_yield_strength(1)).
was_confirmed(topflange_conn_type('welded')).
was_confirmed(topflange_conn_size(0.25)).
was_confirmed(botflange_conn_type('welded')).
was_confirmed(botflange_conn_size(0.25)).
was_confirmed(attachments('stiffeners')).
was_confirmed(stiffeners_width(71.5)).
was_confirmed(stiffeners_thick(0.625)).
was_confirmed(stiffeners_matlname('a36')).
was_confirmed(stiffeners_fracture_toughness(1)).
was_confirmed(stiffeners_yield_strength(1)).
was_confirmed(stiffeners_conn_size(0.25)).
was_confirmed(positive_attachment('no')).
was_confirmed(surface_features('no')).
was_confirmed(minor_stress_range(6)).
was_confirmed(task_max_total_stress('unknown')).
was_confirmed(task_adtl(10000)).
was_confirmed(task_age(20)).
was_confirmed(min_service_temp_zone('zone 1')).
was_confirmed(shipped('rail')).
was_confirmed(miles_shipped(100)).
    
```



	N million cycles	a inches	ΔK Δσ=46ksi	K σ=46ksi
0	0	0.03	13.585	13.585
1	0.006	0.036	14.486	14.486
2	0.011	0.042	15.294	15.294
3	0.018	0.054	16.714	16.714
4	0.029	0.077	19.071	19.071
5	0.043	0.124	22.889	22.889
6	0.058	0.219	30.046	30.046

Cuyahoga County with web gap: conclusions

Conclusions for CRACK consultation on Cuyahoga County bridge

Task to be done is failure analysis.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 36, 16.5 inches wide and 1.25 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 46, 72 inches wide and 0.438 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 36, 16.5 inches wide and 1.25 inches thick. There is a welded 0.25 inches top flange connection. There is a welded 0.25 inches bottom flange connection. Attachments for the girder are: The stiffeners is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 36, 71.5 inches wide and 0.625 inches thick.

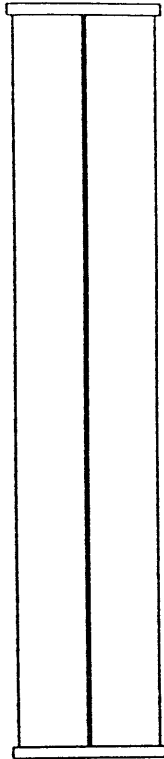
Web gap cracking is due to a rail shipping stress range is 46 ksi at a frequency of 500 cycles per mile for a shipment of 100 miles.

The cracking cause is classified as a out of plane distortion at a web gap type of cracking. The initial flaw size is 0.03 inches. It is thumbnail shaped in the web.

The probable sequence of crack growth to failure is to grow by fatigue as thumbnail crack in web from a crack size of 0.03 inches to 0.219 inches in 0.058 million cycles.

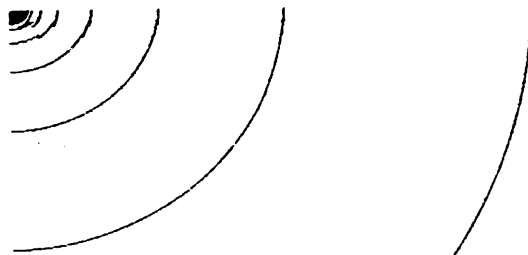
The predicted fatigue life of 0.058 million cycles is in good agreement with the estimated service life of 0.05 million cycles.

Cuyahoga County without web gap: cross-section, confirmed facts, and crack growth



```

was_confirmed(problem_title('Cuyahoga County (no gap)')).
was_confirmed(ask_classification('failure analysis')).
was_confirmed(topflange_width(16.5)).
was_confirmed(topflange_thick(1.25)).
was_confirmed(topflange_matlname(a36)).
was_confirmed(topflange_fracture_toughness(1)).
was_confirmed(topflange_yield_strength(1)).
was_confirmed(web_width(72)).
was_confirmed(web_thick(0.438)).
was_confirmed(web_matlname(a36)).
was_confirmed(web_fracture_toughness(1)).
was_confirmed(web_yield_strength(46)).
was_confirmed(botflange_width(16.5)).
was_confirmed(botflange_thick(1.25)).
was_confirmed(botflange_matlname(a36)).
was_confirmed(botflange_fracture_toughness(1)).
was_confirmed(botflange_yield_strength(1)).
was_confirmed(topflange_conn_type(welded)).
was_confirmed(topflange_conn_size(0.25)).
was_confirmed(botflange_conn_type(welded)).
was_confirmed(botflange_conn_size(0.25)).
was_confirmed(attachments(stiffeners)).
was_confirmed(stiffeners_width(72)).
was_confirmed(stiffeners_thick(0.625)).
was_confirmed(stiffeners_matlname(a36)).
was_confirmed(stiffeners_fracture_toughness(1)).
was_confirmed(stiffeners_yield_strength(1)).
was_confirmed(stiffeners_conn_size(0.25)).
was_confirmed(positive_attachment(yes)).
was_confirmed(surface_features(no)).
was_confirmed(miner_stress_range(6)).
was_confirmed(ask_max_total_stress(unknown)).
was_confirmed(ask_adtl(1130)).
was_confirmed(ask_age(50)).
was_confirmed(min_service_temp_zone(zone1)).
    
```



	N million cycles	a inches	ΔK Δσ=6ksi	K σ=36ksi
0	0	0.03	1.945	11.669
1	1.61	0.035	2.047	12.282
2	3.011	0.04	2.14	12.839
3	5.305	0.049	2.304	13.826
4	8.808	0.068	2.576	15.455
5	13.46	0.106	2.989	17.937
6	19.162	0.183	3.582	21.492
7	25.487	0.335	4.433	26.595
8	31.759	0.64	5.951	35.707
9	35.132	1.25	13.495	80.973

**Cuyahoga County without web gap:
conclusions**

Conclusions for CRACK consultation on Cuyahoga County (no gap) bridge

Task to be done is failure analysis.

The girder top flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 36, 16.5 inches wide and 1.25 inches thick. The girder web is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 46, 72 inches wide and 0.438 inches thick. The girder bottom flange is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 36, 16.5 inches wide and 1.25 inches thick. There is a welded 0.25 inches top flange connection. There is a welded 0.25 inches bottom flange connection. Attachments for the girder are: The stiffeners is a plate of ASTM a36, with a fracture toughness (ksi√inch) of 80 and a yield strength (ksi) of 36, 72 inches wide and 0.625 inches thick.

The in-service stress range is 6 ksi at a frequency of 1130 Average Daily Truck Traffic for a bridge age of 50 years.

The cracking cause is classified as a low fatigue resistant detail at a stiffener weld type of cracking. The initial flaw size is 0.03 inches. It is thumbnail shaped in the bottom flange.

The probable sequence of crack growth to failure is: 1. fatigue growth as thumbnail crack in bottom flange from a crack size of 0.03 inches to 1.25 inches in 35.132 million cycles. 2. fracture as thumbnail crack in bottom flange at a crack size of 1.25 inches at a stress intensity factor of 80.973 ksi√inches.

The predicted fatigue life of 35.132 million cycles is in reasonable agreement with the estimated service life of 20.622 million cycles.

Appendix B:

Qualitative Simulation Example 1

```

/* Qualitative Simulation Example 1: Basic qualitative fatigue and fracture behavior
*
* CRACK MODEL as shown in Figure 5.6
* BEHAVIORS as shown in Figure 5.5
*   parameters:
*     effective temperature Temp_eff
*     residual toughness Kr
*     fracture toughness Kc
*     stress intensity factor K
*     stress intensity range deltaK
*     crack growth rate da/dN
*     crack length a
*     total correction factor F
*     elliptical correction factor Fe
*     free surface correction factor Fs
*     stress gradient correction factor Fg
*     Fesg is used to represent the sum of Fe + Fs + Fg
*     width correction factor Fw
*     stress residual
*     stress dead
*     S_residual_and_dead represents the sum of residual and dead stress
*     stress live S_live
*     stress total S_total
*     initial flaw
*     flaw growth
*     weld size
*     load rate
*     strain rate
*     temperature Temp
*
* The constraint model is made up of the following parts:
*   Temperature effect on Kc:
*     Temp_eff --[M+]-- Kc
*   Enforcing  $K < Kc$  for fatigue:
*      $Kc = Kr + K$ 
*   The main fatigue growth driving loop:
*      $F * S\_live = \text{delta}K$ 
*      $\text{delta}K \text{ --[M+]-- } da/dN$ 
*      $da/dN < \text{--[deriv]-- } a$ 
*      $a \text{ --[M+]-- } Fw$ 
*      $F = Fe + Fs + Fg + Fw$ 
*      $F * S\_live = K$ 
*   Stresses:
*      $S\_total = S\_live + S\_residual\_and\_dead$ 
*      $F * S\_total = K$ 
*
*   Flaw:
*     Weld_size --[M+]-- Initial_flaw

```

```

*           Initial_flaw + Flaw_growth = a
*           Temperature and Strain rate
*           Load_rate --[M+]-- Strain_rate
*           Strain_rate + Temp = Temp_eff
*/

```

```

Envisioning on state 0; remaining states are []
Transitions for temp_eff = [[temp_eff_i,std]]
Transitions for fracture_toughness = [[kci,std]]
Transitions for residual_toughness = [[[0,kri],dec]]
Transitions for stress_intensity_factor = [[[ki,inf],inc]]
Transitions for stress_intensity_range = [[[delta_ki,inf],inc]]
Transitions for crack_growth_rate = [[[rate_i,inf],inc]]
Transitions for crack_length = [[[a_i,thickness],inc]]
Transitions for f = [[[f_i,inf],inc]]
Transitions for fseg = [[fseg_i,std]]
Transitions for fw = [[[fw_i,inf],inc]]
Transitions for s_residual_and_dead = [[s_residual_and_dead_i,std]]
Transitions for s_live = [[s_live_i,std]]
Transitions for s_total = [[s_total_i,std]]
Transitions for initial_flaw = [[initial_flaw_i,std]]
Transitions for flaw_growth = [[[flaw_growth_i,inf],inc]]
Transitions for weld_size = [[weld_size_i,std]]
Transitions for load_rate = [[load_rate_i,std]]
Transitions for strain_rate = [[strain_rate_i,std]]
Transitions for temp = [[temp_i,std]]

```

```

Envisioning on state 1; remaining states are []
Transitions for temp_eff = [[temp_eff_i,std]]
Transitions for fracture_toughness = [[kci,std]]
Transitions for residual_toughness = [[[0,kri],dec],[[0,kri],std],[0,dec],[0,std]]
Transitions for stress_intensity_factor = [[[ki,inf],inc],[[ki,inf],std]]
Transitions for stress_intensity_range = [[[delta_ki,inf],inc],[[delta_ki,inf],std]]
Transitions for crack_growth_rate = [[[rate_i,inf],inc],[[rate_i,inf],std]]
Transitions for crack_length = [[[a_i,thickness],inc],[[a_i,thickness],std],[thickness,inc],[thickness,std]]
Transitions for f = [[[f_i,inf],inc],[[f_i,inf],std]]
Transitions for fseg = [[fseg_i,std]]
Transitions for fw = [[[fw_i,inf],inc],[[fw_i,inf],std]]
Transitions for s_residual_and_dead = [[s_residual_and_dead_i,std]]
Transitions for s_live = [[s_live_i,std]]
Transitions for s_total = [[s_total_i,std]]
Transitions for initial_flaw = [[initial_flaw_i,std]]
Transitions for flaw_growth = [[[flaw_growth_i,inf],inc],[[flaw_growth_i,inf],std]]
Transitions for weld_size = [[weld_size_i,std]]
Transitions for load_rate = [[load_rate_i,std]]
Transitions for strain_rate = [[strain_rate_i,std]]
Transitions for temp = [[temp_i,std]]

```

```

Envisioning on state 2; remaining states are [3,4]
Transitions for temp_eff = [[temp_eff_i,std]]
Transitions for fracture_toughness = [[kci,std]]
Transitions for residual_toughness = [[[0,kri],dec]]
Transitions for stress_intensity_factor = [[[ki,inf],inc]]

```


Transitions for stress_intensity_range = [[[delta_ki,inf],inc]]
 Transitions for crack_growth_rate = [[[rate_i,inf],inc]]
 No successor to thickness in its qspace.
 Transitions for crack_length = []
 Transitions for f = [[[f_i,inf],inc]]
 Transitions for fesg = [[fesg_i,std]]
 Transitions for fw = [[[fw_i,inf],inc]]
 Transitions for s_residual_and_dead = [[s_residual_and_dead_i,std]]
 Transitions for s_live = [[s_live_i,std]]
 Transitions for s_total = [[s_total_i,std]]
 Transitions for initial_flaw = [[initial_flaw_i,std]]
 Transitions for flaw_growth = [[[flaw_growth_i,inf],inc]]
 Transitions for weld_size = [[weld_size_i,std]]
 Transitions for load_rate = [[load_rate_i,std]]
 Transitions for strain_rate = [[strain_rate_i,std]]
 Transitions for temp = [[temp_i,std]]
 Terminal state: no consistent predictions from this state

Envisioning on state 3; remaining states are [4]
 Transitions for temp_eff = [[temp_eff_i,std]]
 Transitions for fracture_toughness = [[kci,std]]
 No predecessor to 0 in its qspace.
 Transitions for residual_toughness = []
 Transitions for stress_intensity_factor = [[[ki,inf],inc]]
 Transitions for stress_intensity_range = [[[delta_ki,inf],inc]]
 Transitions for crack_growth_rate = [[[rate_i,inf],inc]]
 Transitions for crack_length = [[[a_i,thickness],inc]]
 Transitions for f = [[[f_i,inf],inc]]
 Transitions for fesg = [[fesg_i,std]]
 Transitions for fw = [[[fw_i,inf],inc]]
 Transitions for s_residual_and_dead = [[s_residual_and_dead_i,std]]
 Transitions for s_live = [[s_live_i,std]]
 Transitions for s_total = [[s_total_i,std]]
 Transitions for initial_flaw = [[initial_flaw_i,std]]
 Transitions for flaw_growth = [[[flaw_growth_i,inf],inc]]
 Transitions for weld_size = [[weld_size_i,std]]
 Transitions for load_rate = [[load_rate_i,std]]
 Transitions for strain_rate = [[strain_rate_i,std]]
 Transitions for temp = [[temp_i,std]]
 Terminal state: no consistent predictions from this state

Envisioning on state 4; remaining states are []
 Transitions for temp_eff = [[temp_eff_i,std]]
 Transitions for fracture_toughness = [[kci,std]]
 No predecessor to 0 in its qspace.
 Transitions for residual_toughness = []
 Transitions for stress_intensity_factor = [[[ki,inf],inc]]
 Transitions for stress_intensity_range = [[[delta_ki,inf],inc]]
 Transitions for crack_growth_rate = [[[rate_i,inf],inc]]
 No successor to thickness in its qspace.
 Transitions for crack_length = []
 Transitions for f = [[[f_i,inf],inc]]
 Transitions for fesg = [[fesg_i,std]]
 Transitions for fw = [[[fw_i,inf],inc]]

Transitions for s_residual_and_dead = [[s_residual_and_dead_i,std]]
 Transitions for s_live = [[s_live_i,std]]
 Transitions for s_total = [[s_total_i,std]]
 Transitions for initial_flaw = [[initial_flaw_i,std]]
 Transitions for flaw_growth = [[[flaw_growth_i,inf],inc]]
 Transitions for weld_size = [[weld_size_i,std]]
 Transitions for load_rate = [[load_rate_i,std]]
 Transitions for strain_rate = [[strain_rate_i,std]]
 Transitions for temp = [[temp_i,std]]
 Terminal state: no consistent predictions from this state

State 0: Time = 0

/* Initial flaw and cyclic stress */

Predecessors = []

Successors = [1]

temp_eff = temp_eff_i, std
 fracture_toughness = kci, std
 residual_toughness = kri, dec
 stress_intensity_factor = ki, inc
 stress_intensity_range = delta_ki, inc
 crack_growth_rate = rate_i, inc
 crack_length = a_i, inc
 f = f_i, inc
 fseg = fseg_i, std
 fw = fw_i, inc
 s_residual_and_dead = s_residual_and_dead_i, std
 s_live = s_live_i, std
 s_total = s_total_i, std
 initial_flaw = initial_flaw_i, std
 flaw_growth = flaw_growth_i, inc
 weld_size = weld_size_i, std
 load_rate = load_rate_i, std
 strain_rate = strain_rate_i, std
 temp = temp_i, std

Quantity Spaces:

temp_eff : [0,temp_eff_i,inf]
 residual_toughness : [0,kri,inf]
 fracture_toughness : [0,kci,inf]
 stress_intensity_factor : [0,ki,inf]
 stress_intensity_range : [0,delta_ki,inf]
 crack_growth_rate : [0,rate_i,inf]
 crack_length : [0,a_i,thickness]
 f : [0,f_i,inf]
 fseg : [0,fseg_i,inf]
 fw : [0,fw_i,inf]
 s_live : [0,s_live_i,inf]
 s_residual_and_dead : [0,s_residual_and_dead_i,inf]
 s_total : [0,s_total_i,inf]
 initial_flaw : [0,initial_flaw_i,inf]
 flaw_growth : [0,flaw_growth_i,inf]
 weld_size : [0,weld_size_i,inf]
 load_rate : [0,load_rate_i,inf]
 strain_rate : [0,strain_rate_i,inf]
 temp : [0,temp_i,inf]

```

State 1: Time = [0,1]
/* Fatigue crack growth */
Predecessors = [0]
Successors = [4,3,2]
temp_eff = temp_eff_i, std
fracture_toughness = kci, std
residual_toughness = [0,kri], dec
stress_intensity_factor = [ki,inf], inc
stress_intensity_range = [delta_ki,inf], inc
crack_growth_rate = [rate_i,inf], inc
crack_length = [a_i,thickness], inc
f = [f_i,inf], inc
fesg = fesg_i, std
fw = [fw_i,inf], inc
s_residual_and_dead = s_residual_and_dead_i, std
s_live = s_live_i, std
s_total = s_total_i, std
initial_flaw = initial_flaw_i, std
flaw_growth = [flaw_growth_i,inf], inc
weld_size = weld_size_i, std
load_rate = load_rate_i, std
strain_rate = strain_rate_i, std
temp = temp_i, std
Quantity Spaces:
temp_eff : [0,temp_eff_i,inf]
fracture_toughness : [0,kci,inf]
residual_toughness : [0,kri,inf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,rate_i,inf]
crack_length : [0,a_i,thickness]
f : [0,f_i,inf]
fesg : [0,fesg_i,inf]
fw : [0,fw_i,inf]
s_residual_and_dead : [0,s_residual_and_dead_i,inf]
s_live : [0,s_live_i,inf]
s_total : [0,s_total_i,inf]
initial_flaw : [0,initial_flaw_i,inf]
flaw_growth : [0,flaw_growth_i,inf]
weld_size : [0,weld_size_i,inf]
load_rate : [0,load_rate_i,inf]
strain_rate : [0,strain_rate_i,inf]
temp : [0,temp_i,inf]

```

```

State 2: Time = 1
/* Plate penetration by fatigue crack growth */
Predecessors = [1]
Successors = []
temp_eff = temp_eff_i, std
fracture_toughness = kci, std
residual_toughness = [0,kri], dec
stress_intensity_factor = [ki,inf], inc
stress_intensity_range = [delta_ki,inf], inc

```

```

crack_growth_rate = [rate_i,inf], inc
crack_length = thickness, inc
f = [f_i,inf], inc
fesg = fesg_i, std
fw = [fw_i,inf], inc
s_residual_and_dead = s_residual_and_dead_i, std
s_live = s_live_i, std
s_total = s_total_i, std
initial_flaw = initial_flaw_i, std
flaw_growth = [flaw_growth_i,inf], inc
weld_size = weld_size_i, std
load_rate = load_rate_i, std
strain_rate = strain_rate_i, std
temp = temp_i, std

```

Quantity Spaces:

```

temp_eff : [0,temp_eff_i,inf]
fracture_toughness : [0,kci,inf]
residual_toughness : [0,kri,inf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,rate_i,inf]
crack_length : [0,a_i,thickness]
f : [0,f_i,inf]
fesg : [0,fesg_i,inf]
fw : [0,fw_i,inf]
s_residual_and_dead : [0,s_residual_and_dead_i,inf]
s_live : [0,s_live_i,inf]
s_total : [0,s_total_i,inf]
initial_flaw : [0,initial_flaw_i,inf]
flaw_growth : [0,flaw_growth_i,inf]
weld_size : [0,weld_size_i,inf]
load_rate : [0,load_rate_i,inf]
strain_rate : [0,strain_rate_i,inf]
temp : [0,temp_i,inf]

```

State 3: Time = 1

/* Fracture */

```

Predecessors = [1]
Successors = []
temp_eff = temp_eff_i, std
fracture_toughness = kci, std
residual_toughness = 0, dec
stress_intensity_factor = [ki,inf], inc
stress_intensity_range = [delta_ki,inf], inc
crack_growth_rate = [rate_i,inf], inc
crack_length = [a_i,thickness], inc
f = [f_i,inf], inc
fesg = fesg_i, std
fw = [fw_i,inf], inc
s_residual_and_dead = s_residual_and_dead_i, std
s_live = s_live_i, std
s_total = s_total_i, std
initial_flaw = initial_flaw_i, std
flaw_growth = [flaw_growth_i,inf], inc

```

```
weld_size = weld_size_i, std
load_rate = load_rate_i, std
strain_rate = strain_rate_i, std
temp = temp_i, std
```

Quantity Spaces:

```
temp_eff : [0,temp_eff_i,inf]
fracture_toughness : [0,kci,inf]
residual_toughness : [0,kri,inf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,rate_i,inf]
crack_length : [0,a_i,thickness]
f : [0,f_i,inf]
fsg : [0,fsg_i,inf]
fw : [0,fw_i,inf]
s_residual_and_dead : [0,s_residual_and_dead_i,inf]
s_live : [0,s_live_i,inf]
s_total : [0,s_total_i,inf]
initial_flaw : [0,initial_flaw_i,inf]
flaw_growth : [0,flaw_growth_i,inf]
weld_size : [0,weld_size_i,inf]
load_rate : [0,load_rate_i,inf]
strain_rate : [0,strain_rate_i,inf]
temp : [0,temp_i,inf]
```

State 4: Time = 1

/* Simultaneous plate penetration by fatigue crack growth and fracture */

```
Predecessors = [1]
```

```
Successors = []
```

```
temp_eff = temp_eff_i, std
fracture_toughness = kci, std
residual_toughness = 0, dec
stress_intensity_factor = [ki,inf], inc
stress_intensity_range = [delta_ki,inf], inc
crack_growth_rate = [rate_i,inf], inc
crack_length = thickness, inc
f = [f_i,inf], inc
fsg = fsg_i, std
fw = [fw_i,inf], inc
s_residual_and_dead = s_residual_and_dead_i, std
s_live = s_live_i, std
s_total = s_total_i, std
initial_flaw = initial_flaw_i, std
flaw_growth = [flaw_growth_i,inf], inc
weld_size = weld_size_i, std
load_rate = load_rate_i, std
strain_rate = strain_rate_i, std
temp = temp_i, std
```

Quantity Spaces:

```
temp_eff : [0,temp_eff_i,inf]
fracture_toughness : [0,kci,inf]
residual_toughness : [0,kri,inf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
```

crack_growth_rate : [0,rate_i,inf]
crack_length : [0,a_i,thickness]
f : [0,f_i,inf]
fesg : [0,fesg_i,inf]
fw : [0,fw_i,inf]
s_residual_and_dead : [0,s_residual_and_dead_i,inf]
s_live : [0,s_live_i,inf]
s_total : [0,s_total_i,inf]
initial_flaw : [0,initial_flaw_i,inf]
flaw_growth : [0,flaw_growth_i,inf]
weld_size : [0,weld_size_i,inf]
load_rate : [0,load_rate_i,inf]
strain_rate : [0,strain_rate_i,inf]
temp : [0,temp_i,inf]

Qualitative Simulation Example 2

```

/* Qualitative Simulation Example 2: Growth through multiple girder elements
*
* CRACK MODEL sub-network of that shown in Figure 5.6
* BEHAVIORS as shown in Figure 7-2
*   parameters:
*       temperature Temp
*       residual toughness Kr
*       fracture toughness Kc
*       stress intensity factor K
*       stress intensity range deltaK
*       crack growth rate da/dN
*       crack length a
*       total correction factor F
*       elliptical correction factor Fe
*       free surface correction factor Fs
*       stress gradient correction factor Fg
*       Fesg is used to represent the sum of Fe + Fs + Fg
*       width correction factor Fw
*
*   The simplified constraint model made up of the following parts:
*       Temperature effect on Kc:
*           Temp --[M+]-- Kc
*
*       Enforcing  $K < K_c$  for fatigue:
*            $K_c = K_r + K$ 
*
*       The main fatigue growth driving loop:
*           F --[M+]-- deltaK
*           deltaK --[M+]-- da/dN
*           da/dN <--[deriv]-- a
*           a --[M+]-- Fw
*            $F = F_e + F_s + F_g + F_w$ 
*           F *--[M+]-- K
*/

```

```

Envisioning on state 0; remaining states are []
Transitions for temperature = [[temp_i,std]]
Transitions for fracture_toughness = [[kci,std]]
Transitions for residual_toughness = [[[0,kri],dec]]
Transitions for stress_intensity_factor = [[[ki,inf],inc]]
Transitions for stress_intensity_range = [[[delta_ki,inf],inc]]
Transitions for crack_growth_rate = [[[rate_i,inf],inc]]
Transitions for crack_length = [[[botflange_conn,botflange],inc]]
Transitions for f = [[[f_i,inf],inc]]
Transitions for fesg = [[fesg_i,std]]
Transitions for fw = [[[fw_i,inf],inc]]

```

```

Envisioning on state 1; remaining states are []
Transitions for temperature = [[temp_i,std]]
Transitions for fracture_toughness = [[kci,std]]

```

Transitions for residual_toughness = [[[0,kri],dec],[[0,kri],std],[0,dec],[0,std]]
 Transitions for stress_intensity_factor = [[[ki,inf],inc],[[ki,inf],std]]
 Transitions for stress_intensity_range = [[[delta_ki,inf],inc],[[delta_ki,inf],std]]
 Transitions for crack_growth_rate = [[[rate_i,inf],inc],[[rate_i,inf],std]]
 Transitions for crack_length =
 [[[botflange_conn,botflange],inc],[[botflange_conn,botflange],std],[botflange,inc],[botflange,std]]
 Transitions for f = [[[f_i,inf],inc],[[f_i,inf],std]]
 Transitions for fesg = [[fesg_i,std]]
 Transitions for fw = [[[fw_i,inf],inc],[[fw_i,inf],std]]

Envisioning on state 2; remaining states are [3,4]

Transitions for temperature = [[temp_i,std]]
 Transitions for fracture_toughness = [[kci,std]]
 Transitions for residual_toughness = [[[0,kri],dec]]
 Transitions for stress_intensity_factor = [[[ki,inf],inc]]
 Transitions for stress_intensity_range = [[[delta_ki,inf],inc]]
 Transitions for crack_growth_rate = [[[rate_i,inf],inc]]
 Transitions for crack_length = [[[botflange,flange_penetration],inc]]
 Transitions for f = [[[f_i,inf],inc]]
 Transitions for fesg = [[fesg_i,std]]
 Transitions for fw = [[[fw_i,inf],inc]]

Envisioning on state 3; remaining states are [4,5]

Transitions for temperature = [[temp_i,std]]
 Transitions for fracture_toughness = [[kci,std]]
 No predecessor to 0 in its qspace.
 Transitions for residual_toughness = []
 Transitions for stress_intensity_factor = [[[ki,inf],inc]]
 Transitions for stress_intensity_range = [[[delta_ki,inf],inc]]
 Transitions for crack_growth_rate = [[[rate_i,inf],inc]]
 Transitions for crack_length = [[[botflange_conn,botflange],inc]]
 Transitions for f = [[[f_i,inf],inc]]
 Transitions for fesg = [[fesg_i,std]]
 Transitions for fw = [[[fw_i,inf],inc]]
 Terminal state: no consistent predictions from this state

Envisioning on state 4; remaining states are [5]

Transitions for temperature = [[temp_i,std]]
 Transitions for fracture_toughness = [[kci,std]]
 No predecessor to 0 in its qspace.
 Transitions for residual_toughness = []
 Transitions for stress_intensity_factor = [[[ki,inf],inc]]
 Transitions for stress_intensity_range = [[[delta_ki,inf],inc]]
 Transitions for crack_growth_rate = [[[rate_i,inf],inc]]
 Transitions for crack_length = [[[botflange,flange_penetration],inc]]
 Transitions for f = [[[f_i,inf],inc]]
 Transitions for fesg = [[fesg_i,std]]
 Transitions for fw = [[[fw_i,inf],inc]]
 Terminal state: no consistent predictions from this state

Envisioning on state 5; remaining states are []

Transitions for temperature = [[temp_i,std]]
 Transitions for fracture_toughness = [[kci,std]]

Transitions for residual_toughness = [[[0,kri],dec],[[0,kri],std],[0,dec],[0,std]]
 Transitions for stress_intensity_factor = [[[ki,inf],inc],[[ki,inf],std]]
 Transitions for stress_intensity_range = [[[delta_ki,inf],inc],[[delta_ki,inf],std]]
 Transitions for crack_growth_rate = [[[rate_i,inf],inc],[[rate_i,inf],std]]
 Transitions for crack_length =
 [[[botflange,flange_penetration],inc],[[botflange,flange_penetration],std],[flange_penetration,inc],[flange_penetration,std]]
 Transitions for f = [[[f_i,inf],inc],[[f_i,inf],std]]
 Transitions for fesg = [[fesg_i,std]]
 Transitions for fw = [[[fw_i,inf],inc],[[fw_i,inf],std]]

Envisioning on state 6; remaining states are [7,8]
 Transitions for temperature = [[temp_i,std]]
 Transitions for fracture_toughness = [[kci,std]]
 Transitions for residual_toughness = [[[0,kri],dec]]
 Transitions for stress_intensity_factor = [[[ki,inf],inc]]
 Transitions for stress_intensity_range = [[[delta_ki,inf],inc]]
 Transitions for crack_growth_rate = [[[rate_i,inf],inc]]
 No successor to flange_penetration in its qspace.
 Transitions for crack_length = []
 Transitions for f = [[[f_i,inf],inc]]
 Transitions for fesg = [[fesg_i,std]]
 Transitions for fw = [[[fw_i,inf],inc]]
 Terminal state: no consistent predictions from this state

Envisioning on state 7; remaining states are [8]
 Transitions for temperature = [[temp_i,std]]
 Transitions for fracture_toughness = [[kci,std]]
 No predecessor to 0 in its qspace.
 Transitions for residual_toughness = []
 Transitions for stress_intensity_factor = [[[ki,inf],inc]]
 Transitions for stress_intensity_range = [[[delta_ki,inf],inc]]
 Transitions for crack_growth_rate = [[[rate_i,inf],inc]]
 Transitions for crack_length = [[[botflange,flange_penetration],inc]]
 Transitions for f = [[[f_i,inf],inc]]
 Transitions for fesg = [[fesg_i,std]]
 Transitions for fw = [[[fw_i,inf],inc]]
 Terminal state: no consistent predictions from this state

Envisioning on state 8; remaining states are []
 Transitions for temperature = [[temp_i,std]]
 Transitions for fracture_toughness = [[kci,std]]
 No predecessor to 0 in its qspace.
 Transitions for residual_toughness = []
 Transitions for stress_intensity_factor = [[[ki,inf],inc]]
 Transitions for stress_intensity_range = [[[delta_ki,inf],inc]]
 Transitions for crack_growth_rate = [[[rate_i,inf],inc]]
 No successor to flange_penetration in its qspace.
 Transitions for crack_length = []
 Transitions for f = [[[f_i,inf],inc]]
 Transitions for fesg = [[fesg_i,std]]
 Transitions for fw = [[[fw_i,inf],inc]]
 Terminal state: no consistent predictions from this state

```

State 0: Time = 0
/* initial flaw in bottom flange connection */
Predecessors = []
Successors = [1]
temperature = temp_i, std
fracture_toughness = kci, std
residual_toughness = kri, dec
stress_intensity_factor = ki, inc
stress_intensity_range = delta_ki, inc
crack_growth_rate = rate_i, inc
crack_length = botflange_conn, inc
f = f_i, inc
fesg = fesg_i, std
fw = fw_i, inc
Quantity Spaces:
temperature : [0,tempi,tempf]
residual_toughness : [0,kri,inf]
fracture_toughness : [0,kci,kcf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,rate_i,inf]
crack_length : [0,botflange_conn,botflange,flange_penetration]
f : [0,f_i,inf]
fesg : [0,fesg_i,inf]
fw : [0,fw_i,inf]

```

```

State 1: Time = [0,1]
/* fatigue crack growth in bottom flange connection */
Predecessors = [0]
Successors = [4,3,2]
temperature = temp_i, std
fracture_toughness = kci, std
residual_toughness = [0,kri], dec
stress_intensity_factor = [ki,inf], inc
stress_intensity_range = [delta_ki,inf], inc
crack_growth_rate = [rate_i,inf], inc
crack_length = [botflange_conn,botflange], inc
f = [f_i,inf], inc
fesg = fesg_i, std
fw = [fw_i,inf], inc
Quantity Spaces:
temperature : [0,tempi,tempf]
fracture_toughness : [0,kci,kcf]
residual_toughness : [0,kri,inf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,rate_i,inf]
crack_length : [0,botflange_conn,botflange,flange_penetration]
f : [0,f_i,inf]
fesg : [0,fesg_i,inf]
fw : [0,fw_i,inf]

```

State 2: Time = 1

/* crack enters bottom flange */

Predecessors = [1]

Successors = [5]

temperature = temp_i, std
fracture_toughness = kci, std
residual_toughness = [0,kri], dec
stress_intensity_factor = [ki,inf], inc
stress_intensity_range = [delta_ki,inf], inc
crack_growth_rate = [rate_i,inf], inc
crack_length = botflange, inc
f = [f_i,inf], inc
fesg = fesg_i, std
fw = [fw_i,inf], inc

Quantity Spaces:

temperature : [0,tempi,tempf]
fracture_toughness : [0,kci,kcf]
residual_toughness : [0,kri,inf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,rate_i,inf]
crack_length : [0,botflange_conn,botflange,flange_penetration]
f : [0,f_i,inf]
fesg : [0,fesg_i,inf]
fw : [0,fw_i,inf]

State 3: Time = 1

/* fracture in bottom flange connection */

Predecessors = [1]

Successors = []

temperature = temp_i, std
fracture_toughness = kci, std
residual_toughness = 0, dec
stress_intensity_factor = [ki,inf], inc
stress_intensity_range = [delta_ki,inf], inc
crack_growth_rate = [rate_i,inf], inc
crack_length = [botflange_conn,botflange], inc
f = [f_i,inf], inc
fesg = fesg_i, std
fw = [fw_i,inf], inc

Quantity Spaces:

temperature : [0,tempi,tempf]
fracture_toughness : [0,kci,kcf]
residual_toughness : [0,kri,inf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,rate_i,inf]
crack_length : [0,botflange_conn,botflange,flange_penetration]
f : [0,f_i,inf]
fesg : [0,fesg_i,inf]
fw : [0,fw_i,inf]

State 4: Time = 1

```

/* simultaneous fracture and bottom flange entry */
Predecessors = [1]
Successors = []
temperature = temp_i, std
fracture_toughness = kci, std
residual_toughness = 0, dec
stress_intensity_factor = [ki,inf], inc
stress_intensity_range = [delta_ki,inf], inc
crack_growth_rate = [rate_i,inf], inc
crack_length = botflange, inc
f = [f_i,inf], inc
fesg = fesg_i, std
fw = [fw_i,inf], inc
Quantity Spaces:
temperature : [0,tempi,tempf]
fracture_toughness : [0,kci,kcf]
residual_toughness : [0,kri,inf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,rate_i,inf]
crack_length : [0,botflange_conn,botflange,flange_penetration]
f : [0,f_i,inf]
fesg : [0,fesg_i,inf]
fw : [0,fw_i,inf]

```

State 5: Time = [1,2]

```

/* fatigue crack growth in bottom flange */
Predecessors = [2]
Successors = [8,7,6]
temperature = temp_i, std
fracture_toughness = kci, std
residual_toughness = [0,kri], dec
stress_intensity_factor = [ki,inf], inc
stress_intensity_range = [delta_ki,inf], inc
crack_growth_rate = [rate_i,inf], inc
crack_length = [botflange,flange_penetration], inc
f = [f_i,inf], inc
fesg = fesg_i, std
fw = [fw_i,inf], inc
Quantity Spaces:
temperature : [0,tempi,tempf]
fracture_toughness : [0,kci,kcf]
residual_toughness : [0,kri,inf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,rate_i,inf]
crack_length : [0,botflange_conn,botflange,flange_penetration]
f : [0,f_i,inf]
fesg : [0,fesg_i,inf]
fw : [0,fw_i,inf]

```

State 6: Time = 2

```

/* bottom flange penetration */

```

```

Predecessors = [5]
Successors = []
  temperature = temp_i, std
  fracture_toughness = kci, std
  residual_toughness = [0,kri], dec
  stress_intensity_factor = [ki,inf], inc
  stress_intensity_range = [delta_ki,inf], inc
  crack_growth_rate = [rate_i,inf], inc
  crack_length = flange_penetration, inc
  f = [f_i,inf], inc
  fesg = fesg_i, std
  fw = [fw_i,inf], inc
Quantity Spaces:
  temperature : [0,tempi,tempf]
  fracture_toughness : [0,kci,kcf]
  residual_toughness : [0,kri,inf]
  stress_intensity_factor : [0,ki,inf]
  stress_intensity_range : [0,delta_ki,inf]
  crack_growth_rate : [0,rate_i,inf]
  crack_length : [0,botflange_conn,botflange,flange_penetration]
  f : [0,f_i,inf]
  fesg : [0,fesg_i,inf]
  fw : [0,fw_i,inf]

```

State 7: Time = 2

/* fracture in bottom flange */

```

Predecessors = [5]
Successors = []
  temperature = temp_i, std
  fracture_toughness = kci, std
  residual_toughness = 0, dec
  stress_intensity_factor = [ki,inf], inc
  stress_intensity_range = [delta_ki,inf], inc
  crack_growth_rate = [rate_i,inf], inc
  crack_length = [botflange,flange_penetration], inc
  f = [f_i,inf], inc
  fesg = fesg_i, std
  fw = [fw_i,inf], inc
Quantity Spaces:
  temperature : [0,tempi,tempf]
  fracture_toughness : [0,kci,kcf]
  residual_toughness : [0,kri,inf]
  stress_intensity_factor : [0,ki,inf]
  stress_intensity_range : [0,delta_ki,inf]
  crack_growth_rate : [0,rate_i,inf]
  crack_length : [0,botflange_conn,botflange,flange_penetration]
  f : [0,f_i,inf]
  fesg : [0,fesg_i,inf]
  fw : [0,fw_i,inf]

```

State 8: Time = 2

/* simultaneous fracture and bottom flange penetration */

```

Predecessors = [5]

```

```
Successors = []
  temperature = temp_i, std
  fracture_toughness = kci, std
  residual_toughness = 0, dec
  stress_intensity_factor = [ki,inf], inc
  stress_intensity_range = [delta_ki,inf], inc
  crack_growth_rate = [rate_i,inf], inc
  crack_length = flange_penetration, inc
  f = [f_i,inf], inc
  fesg = fesg_i, std
  fw = [fw_i,inf], inc
Quantity Spaces:
  temperature : [0,tempi,tempf]
  fracture_toughness : [0,kci,kcf]
  residual_toughness : [0,kri,inf]
  stress_intensity_factor : [0,ki,inf]
  stress_intensity_range : [0,delta_ki,inf]
  crack_growth_rate : [0,rate_i,inf]
  crack_length : [0,botflange_conn,botflange,flange_penetration]
  f : [0,f_i,inf]
  fesg : [0,fesg_i,inf]
  fw : [0,fw_i,inf]
```

Qualitative Simulation Example 3

```

/* Qualitative Simulation Example 3: Temperature decrease causing fracture
*
* CRACK MODEL sub-network of that shown in Figure 5.6
* BEHAVIORS as shown in Figure 7-3
*   parameters:
*       temperature Temp
*       residual toughness Kr
*       fracture toughness Kc
*       stress intensity factor K
*       stress intensity range deltaK
*       crack growth rate da/dN
*       crack length a
*
*   The simplified constraint model made up of the following parts:
*       Temperature effect on Kc:
*           Temp --[M+]-- Kc
*
*       Enforcing  $K < Kc$  for fatigue:
*            $Kc = Kr + K$ 
*
*       The main fatigue growth driving loop:
*           K --[M+]-- deltaK
*           deltaK --[M+]-- da/dN
*           da/dN <--[deriv]-- a
*           a --[M+]-- K
*/

```

```

Envisioning on state 0; remaining states are []
Transitions for temperature = [[[0,temp_i],dec]]
Transitions for fracture_toughness = [[[0,kc_i],dec]]
Transitions for residual_toughness = [[[0,kr_i],dec]]
Transitions for stress_intensity_factor = [[ki,std]]
Transitions for stress_intensity_range = [[delta_ki,std]]
Transitions for crack_growth_rate = [[0,std]]
Transitions for crack_length = [[a_i,std]]

```

```

Envisioning on state 1; remaining states are []
Transitions for temperature = [[[0,temp_i],dec],[[0,temp_i],std],[0,dec],[0,std]]
Transitions for fracture_toughness = [[[0,kc_i],dec],[[0,kc_i],std],[0,dec],[0,std]]
Transitions for residual_toughness = [[[0,kr_i],dec],[[0,kr_i],std],[0,dec],[0,std]]
Transitions for stress_intensity_factor = [[ki,std]]
Transitions for stress_intensity_range = [[delta_ki,std]]
Transitions for crack_growth_rate = [[0,std]]
Transitions for crack_length = [[a_i,std]]

```

```

Envisioning on state 2; remaining states are [3,4,5,6,7,8]
Transitions for temperature = [[[0,temp_i],dec]]
Transitions for fracture_toughness = [[[0,kc_i],dec]]
No predecessor to 0 in its qspace.
Transitions for residual_toughness = []
Transitions for stress_intensity_factor = [[ki,std]]

```

Transitions for stress_intensity_range = [[delta_ki,std]]
Transitions for crack_growth_rate = [[0,std]]
Transitions for crack_length = [[a_i,std]]
Terminal state: no consistent predictions from this state

Envisioning on state 3; remaining states are [4,5,6,7,8]

Envisioning on state 4; remaining states are [5,6,7,8]

Envisioning on state 5; remaining states are [6,7,8]
No predecessor to 0 in its qspace.
Transitions for temperature = []
No predecessor to 0 in its qspace.
Transitions for fracture_toughness = []
Transitions for residual_toughness = [[[0,kri],dec]]
Transitions for stress_intensity_factor = [[ki,std]]
Transitions for stress_intensity_range = [[delta_ki,std]]
Transitions for crack_growth_rate = [[0,std]]
Transitions for crack_length = [[a_i,std]]
Terminal state: no consistent predictions from this state

Envisioning on state 6; remaining states are [7,8]
No predecessor to 0 in its qspace.
Transitions for temperature = []
No predecessor to 0 in its qspace.
Transitions for fracture_toughness = []
No predecessor to 0 in its qspace.
Transitions for residual_toughness = []
Transitions for stress_intensity_factor = [[ki,std]]
Transitions for stress_intensity_range = [[delta_ki,std]]
Transitions for crack_growth_rate = [[0,std]]
Transitions for crack_length = [[a_i,std]]
Terminal state: no consistent predictions from this state

Envisioning on state 7; remaining states are [8]

Envisioning on state 8; remaining states are []

State 0: Time = 0

/* temperature about to decrease, no cyclic stress */

Predecessors = []

Successors = [1]

temperature = tempi, dec
fracture_toughness = kci, dec
residual_toughness = kri, dec
stress_intensity_factor = ki, std
stress_intensity_range = delta_ki, std
crack_growth_rate = 0, std
crack_length = a_i, std

Quantity Spaces:

temperature : [0,tempi,inf]
residual_toughness : [0,kri,inf]
fracture_toughness : [0,kci,inf]
stress_intensity_factor : [0,ki,inf]


```
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,inf]
crack_length : [0,a_i,thickness]
```

State 1: Time = [0,1]

/* temperature begins to decrease */

```
Predecessors = [0]
Successors = [8,7,6,5,4,3,2]
temperature = [0,tempi], dec
fracture_toughness = [0,kci], dec
residual_toughness = [0,kri], dec
stress_intensity_factor = ki, std
stress_intensity_range = delta_ki, std
crack_growth_rate = 0, std
crack_length = a_i, std
```

Quantity Spaces:

```
temperature : [0,tempi,inf]
fracture_toughness : [0,kci,inf]
residual_toughness : [0,kri,inf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,inf]
crack_length : [0,a_i,thickness]
```

State 2: Time = 1

/* fracture while temperature continues to decrease */

```
Predecessors = [1]
Successors = []
temperature = [0,tempi], dec
fracture_toughness = [0,kci], dec
residual_toughness = 0, dec
stress_intensity_factor = ki, std
stress_intensity_range = delta_ki, std
crack_growth_rate = 0, std
crack_length = a_i, std
```

Quantity Spaces:

```
temperature : [0,tempi,inf]
fracture_toughness : [0,kci,inf]
residual_toughness : [0,kri,inf]
stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,inf]
crack_length : [0,a_i,thickness]
```

State 3: Time = 1

/* temperature stabilizes at a lower temperature with a lower residual strength */

```
Predecessors = [1]
Successors = []
temperature = temperature1, std
fracture_toughness = fracture_toughness1, std
residual_toughness = residual_toughness1, std
stress_intensity_factor = ki, std
stress_intensity_range = delta_ki, std
```

```

    crack_growth_rate = 0, std
    crack_length = a_i, std
Quantity Spaces:
    temperature : [0,temperature1,tempi,inf]
    fracture_toughness : [0,fracture_toughness1,kci,inf]
    residual_toughness : [0,residual_toughness1,kri,inf]
    stress_intensity_factor : [0,ki,inf]
    stress_intensity_range : [0,delta_ki,inf]
    crack_growth_rate : [0,inf]
    crack_length : [0,a_i,thickness]

```

State 4: Time = 1

/* simultaneous fracture and temperature stabilization */

```

Predecessors = [1]
Successors = []
    temperature = temperature2, std
    fracture_toughness = fracture_toughness2, std
    residual_toughness = 0, std
    stress_intensity_factor = ki, std
    stress_intensity_range = delta_ki, std
    crack_growth_rate = 0, std
    crack_length = a_i, std
Quantity Spaces:
    temperature : [0,temperature2,tempi,inf]
    fracture_toughness : [0,fracture_toughness2,kci,inf]
    residual_toughness : [0,kri,inf]
    stress_intensity_factor : [0,ki,inf]
    stress_intensity_range : [0,delta_ki,inf]
    crack_growth_rate : [0,inf]
    crack_length : [0,a_i,thickness]

```

State 5: Time = 1

/* not a physically possible state, fracture toughness reaches zero */

```

Predecessors = [1]
Successors = []
    temperature = 0, dec
    fracture_toughness = 0, dec
    residual_toughness = [0,kri], dec
    stress_intensity_factor = ki, std
    stress_intensity_range = delta_ki, std
    crack_growth_rate = 0, std
    crack_length = a_i, std
Quantity Spaces:
    temperature : [0,tempi,inf]
    fracture_toughness : [0,kci,inf]
    residual_toughness : [0,kri,inf]
    stress_intensity_factor : [0,ki,inf]
    stress_intensity_range : [0,delta_ki,inf]
    crack_growth_rate : [0,inf]
    crack_length : [0,a_i,thickness]

```

State 6: Time = 1

/* not a physically possible state, fracture toughness reaches zero */

```

Predecessors = [1]
Successors = []
  temperature = 0, dec
  fracture_toughness = 0, dec
  residual_toughness = 0, dec
  stress_intensity_factor = ki, std
  stress_intensity_range = delta_ki, std
  crack_growth_rate = 0, std
  crack_length = a_i, std
Quantity Spaces:
  temperature : [0,tempi,inf]
  fracture_toughness : [0,kci,inf]
  residual_toughness : [0,kri,inf]
  stress_intensity_factor : [0,ki,inf]
  stress_intensity_range : [0,delta_ki,inf]
  crack_growth_rate : [0,inf]
  crack_length : [0,a_i,thickness]

```

State 7: Time = 1

/* not a physically possible state, fracture toughness reaches zero */

```

Predecessors = [1]
Successors = []
  temperature = 0, std
  fracture_toughness = 0, std
  residual_toughness = residual_toughness2, std
  stress_intensity_factor = ki, std
  stress_intensity_range = delta_ki, std
  crack_growth_rate = 0, std
  crack_length = a_i, std
Quantity Spaces:
  temperature : [0,tempi,inf]
  fracture_toughness : [0,kci,inf]
  residual_toughness : [0,residual_toughness2,kri,inf]
  stress_intensity_factor : [0,ki,inf]
  stress_intensity_range : [0,delta_ki,inf]
  crack_growth_rate : [0,inf]
  crack_length : [0,a_i,thickness]

```

State 8: Time = 1

/* not a physically possible state, fracture toughness reaches zero */

```

Predecessors = [1]
Successors = []
  temperature = 0, std
  fracture_toughness = 0, std
  residual_toughness = 0, std
  stress_intensity_factor = ki, std
  stress_intensity_range = delta_ki, std
  crack_growth_rate = 0, std
  crack_length = a_i, std
Quantity Spaces:
  temperature : [0,tempi,inf]
  fracture_toughness : [0,kci,inf]
  residual_toughness : [0,kri,inf]

```

stress_intensity_factor : [0,ki,inf]
stress_intensity_range : [0,delta_ki,inf]
crack_growth_rate : [0,inf]
crack_length : [0,a_i,thickness]