2-2011

Development of an Open Source Bridge Management System

Michael H. Scott
Oregon State University

Recommended Citation

Scott, Michael. Development of an Open Source Bridge Management System. OTREC-RR-11-11. Portland, OR: Transportation Research and Education Center (TREC), 2011. https://doi.org/10.15760/trec.93

OTREC

OREGON
TRANSPORTATION
RESEARCH AND
EDUCATION CONSORTIUM

FINAL REPORT

Development of an Open Source Bridge Management System

OTREC-RR-11-11
February 2011

A National University Transportation Center sponsored by the U.S. Department of Transportation's Research and Innovative Technology Administration

DEVELOPMENT OF AN OPEN-SOURCE BRIDGE MANAGEMENT SYSTEM

Final Report

OTREC-RR-11-11

by

Prof. Michael H. Scott
Oregon State University

for

Oregon Department of Transportation Research Unit
200 Hawthorne Avenue SE, Suite B-240
Salem, OR 97301-5192

and

Oregon Transportation Research
and Education Consortium (OTREC)
P.O. Box 751
Portland, OR 97207

OTREC
OREGON TRANSPORTATION RESEARCH
AND EDUCATION CONSORTIUM

February 2011

Technical Report Documentation Page

1. Report No. OTREC-RR-11-11	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Development of an Open Source Bridge Management System		5. Report Date February 2011
		6. Performing Organization Code
7. Author(s) Prof. Michael H. Scott		8. Performing Organization Report No.
9. Performing Organization Name and Address Oregon State University School of Civil and Construction Engineering 220 Owen Hall Corvallis, OR 97331-3212		10. Work Unit No. (TRAIS)
		11. Contract or Grant No. 2008-156
12. Sponsoring Agency Name and Address Oregon Transportation Research and Education Consortium (OTREC) P.O. Box 751 Portland, Oregon 97207		13. Type of Report and Period Covered Final Report
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

A bridge management system is developed using the Tcl scripting language in conjunction with the OpenSees finite element software framework. Fully programmable and string-based, Tcl is ideal for implementing live load analysis through scripts and experimenting with emergent bridge rating methodologies. Since Tcl is an interpreted language, the application also has the important advantage that new bridge capacity models and rating factor calculations can be implemented on multiple platforms without compiling source code. The network programming features of Tcl give the system access to databases for conducting internet-based bridge rating. The system is demonstrated for rating a conventionally reinforced concrete girder; however, it is readily extensible to other types of bridge components.

17. Key Words Bridge maintenance, Computer-aided simulation, Computer programming, Live loads, Ratings, Structural analysis		18. Distribution Statement No restrictions. Copies available from OTREC: ww.otrec.us	
19. Security Classification (of this report) Unclassified	20. Security Classification (of this page) Unclassified	21. No. of Pages 36	22. Price

ACKNOWLEDGEMENTS

This project was funded by the Oregon Transportation Research and Education Consortium (OTREC) with match funding from the Oregon Department of Transportation (ODOT). The author is thankful for this opportunity provided by OTREC and ODOT.

DISCLAIMER

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

Seismic analysis is a conventional part of structural analysis (*Chopra 2001*). All of the finite-element programs implement it as a basic function, and the effects of moving vehicles are always of special concerns in bridge engineering. Previously, vehicles were often approximated as moving loads, which in many cases allows a finite-element program to implement it without difficulties. But as increasingly larger vehicles are manufactured, bridges are subjected to larger and heavier loads so that the inertia effect of the vehicles can no longer be neglected.

To consider this effect, various vehicle models were developed by researchers. Thus the problem of vehicle-bridge interaction becomes an issue of great concern which has been addressed extensively in recent literature (*Lu et al. 2009; Li et al. 2008; Xia et al. 2006; Ju et al. 2006; Kwasniewski et al. 2006; Broquet et al. 2004; Kim and Kawatani 2006; Yang and Wu 2002; Yang and Wu 2001; Nassif et al. 2003; Lee and Yhim 2005; Yang et al. 1997*). However, the combined effect of theses loads has not been adequately addressed (*Matsumoto et al. 2004; Yau and Fryba 2007; Fryba and Yau 2009; Sogabe et al. 2007*).

In analyzing the VBI (Vehicle Bridge Interaction) system, two sets of equations of motion have to be written for the vehicles and for the bridge. The two equations are coupled together by the interaction forces at contact points. One way to solve the two sets of equations is through an iterative procedure (*Broquet et al. 2004; Xia et al. 2006*). For example, by first assuming the weights of vehicles as interaction forces, one can solve the bridge equations to obtain the displacements at contact points and then proceed to solve the vehicle equations for improved values of interaction forces. This is the first cycle of iteration. For this method, the convergence rate is likely to be low when dealing with large problems.

Another way is to merge two equations by eliminating the contact forces between vehicle wheels and bridge to one equation (*Yang and Wu 2001; Kim and Kawatani 2006; Li et al. 2008; Lu et al. 2009*). This is an efficient way to solve the VBI problem, especially for computing the bridge responses. The only drawback is that one should develop his own program to solve the condensed equations. To do the combined analysis, the function of seismic analysis needs to be added in the program. This could be very time consuming and unnecessary, since the tools for seismic analysis in existing finite-element programs have been proven reliable and stable for years.

The last option would be to incorporate the vehicle model in existing finite-element software by defining the vehicles as structural components. The advantage is that all of the tools for structural analysis, including seismic analysis, are available. But the problem is that most finite-element programs assume the structure is invariant in a same analysis. In other words, vehicles defined as structural components are unable to move. Taking these into account, an idea advanced in this study is to move the vehicle properties instead of the vehicle itself. This aim can be achieved with the help of OpenSees and Tcl scripting language.

OpenSees (Open system for earthquake engineering simulation) is a software framework aimed at simulating the seismic response of structural and geotechnical systems. OpenSees has been developed as the computational platform for research in performance-based earthquake engineering at the Pacific Earthquake Engineering Research Center (PEER) and is also the simulation component for the NEESit since 2004. OpenSees is open source and designed in a modular fashion to support the finite-element method with loose coupling of analysis and model building components. Developers do not need to know everything that is in the framework, allowing them to make improvements or create applications in their areas of expertise.

Most users of OpenSees build models and conduct analysis via the string-based Tcl scripting language. Tcl is fully programmable with the control structures, variable substitutions and procedures that are necessary to automate routine operations using scripts (*Welch 2000*). The aim of Tcl is to serve as a glue language that assembles software building blocks into customized applications. This is accomplished by allowing developers to extend the Tcl interpreter with commands that suit the needs of an application. In the case of OpenSees, the Tcl interpreter is extended with commands to define the nodes, boundary conditions, elements, loads, mass and solution strategies of a finite-element analysis (*Mazzoni et al. 2009*).

The objective of this report is to describe the method for combined seismic and live-load analysis of bridges in the OpenSees, and to outline how Tcl is used as a glue to construct a bridge-analysis application from finite-element analysis building blocks in the OpenSees framework. The presentation begins with modeling approaches of the bridge and vehicles. The subsequent sections present Tcl scripts that define bridge and vehicle models, apply seismic and gravity loads, and perform the analysis. Analytical and experimental examples are shown to demonstrate the method and the application, followed by the conclusion.

1.0 INTRODUCTION

Rating is the primary tool for evaluating the fitness of highway bridges for continued service.

To rate a bridge component, the load-effect demands from truck configurations are computed by structural analysis, then compared with an estimated capacity of the component. Bridge rating requires a significant amount of time and produces a large amount of data; fortunately, it has been automated in several software applications. For a closed software model where only a few developers have access to source code it is often difficult, if not impossible, for a user to incorporate new simulation models for bridge components; to modify the capacity and rating factor calculations to keep up with changing code provisions; to experiment with emerging rating methodologies such as those based on structural reliability [1, 7]; or to extend the software to system-level models of bridge network performance [10]. A notable exception to the closed software model is the Alternate Route Project [4], which uses an open-source model to develop bridge design and analysis applications.

The open-source approach also has been adopted in the development of the software framework OpenSees, which consists of a set of cooperating modules that can be used to construct applications for structural and geotechnical engineering [13]. OpenSees is based on the finite-element method [5, 3, 24], which is the most versatile approach to computing the response of structural systems to general loadings. OpenSees is designed in a modular fashion to support the finite-element method with loose coupling of analysis and model building components [12]. This allows users and developers in different fields, including engineering, computer science, and numerical analysis, to modify or implement specific modules with relatively little dependence on other modules. A developer does not need to know everything that is in the framework, allowing them to concentrate on making improvements or creating applications in their areas of expertise. Furthermore, modules can be optimized to take advantage of computing hardware, communication and visualization without the changes propagating throughout the entire system.

Most users of OpenSees build models and conduct analyses via the string-based Tool Command Language, Tcl [16]. Tcl is fully programmable with the control structures, variable substitutions and procedures that are necessary to automate routine operations using scripts (*Welch 2000*). The aim of Tcl is to serve as a glue language that assembles software building blocks, or primitives, into applications. Tcl uses an interpreter that can be extended with customized commands that suit the needs of an application. In the case of OpenSees, the Tcl interpreter is extended with commands to define the nodes, boundary conditions, elements, loads, and solution strategies of a finite-element analysis [11]. To suit the needs of a particular application, these commands can be composed into customized procedures that are portable and do not require source code compilation. There are many advantages to giving users of OpenSees a fully programmable, interpreted language for defining models and solution methods, including the ability to conduct parameter studies, to provide network access to data and storage, to control hybrid simulations, and to communicate with graphical user interfaces [17, 22, 19, 8].

This report outlines how Tcl is used as a glue to construct bridge-rating applications from “primitive” finite-element analysis modules in the OpenSees framework. As indicated in Figure 1.1, this approach permits loose coupling of building blocks and thus greater flexibility and modularity than applications that tightly bundle all components. The presentation begins with scripts to define bridge models and compute demands by moving-load analysis using the finite-element modules of OpenSees. Procedures to compute the moment and shear capacity of conventionally reinforced concrete girders and to implement specification-based rating equations are presented next. An example bridge rating using the network programming features of Tcl is shown, then the report concludes with future extensions of the bridge-rating application.

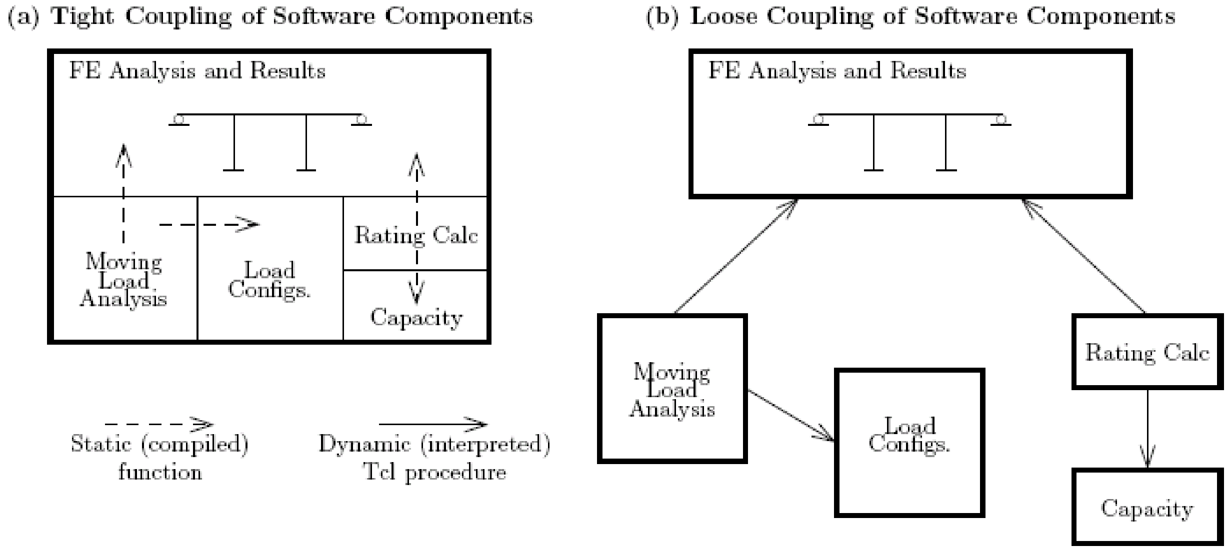


Figure 1.1: Software architectures for a typical bridge-rating application: (a) tight coupling of modules linked by static function calls and (b) loosely coupled modules linked by dynamic procedure calls.

2.0 SCRIPTS FOR LIVE-LOAD BRIDGE ANALYSIS

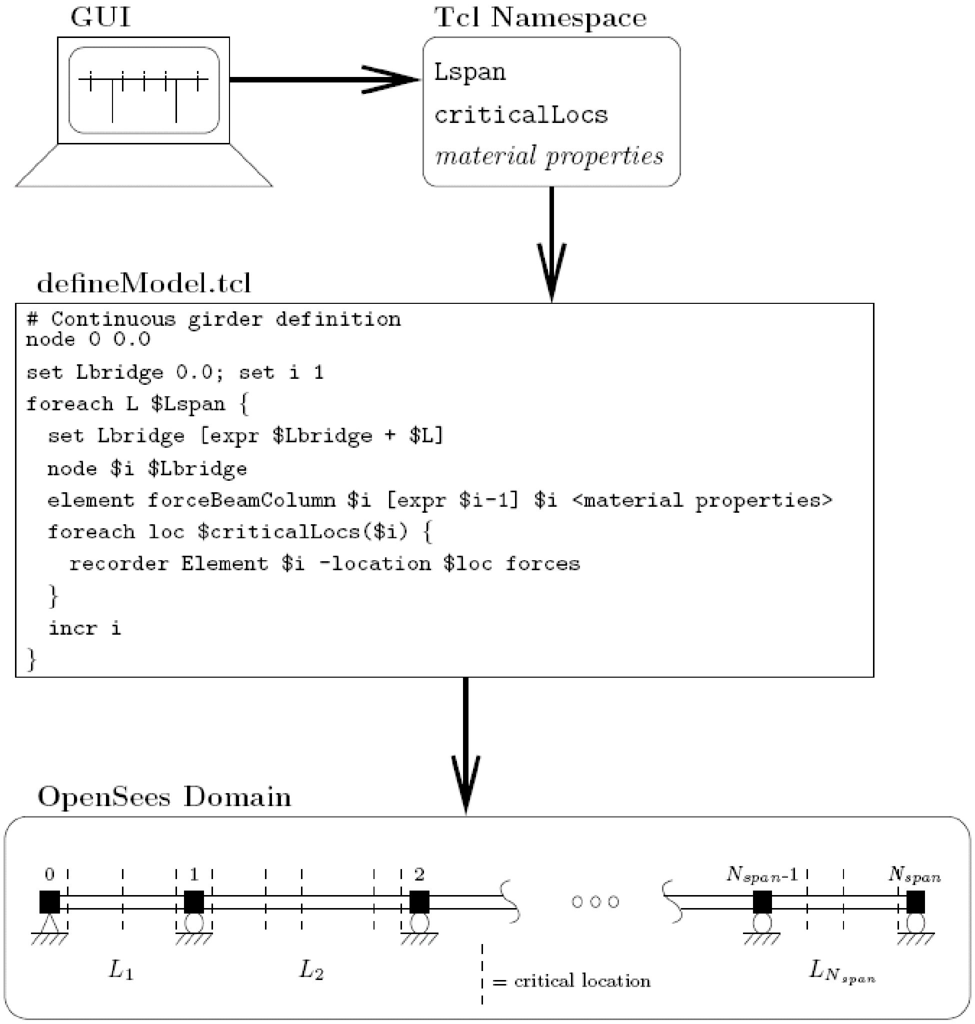
This report considers the rating of continuous bridge girders, for which the typical approach to compute live-load demands is to perform static, two-dimensional analyses. Distribution factors approximate three-dimensional effects of load transfer through the bridge deck and an impact factor accounts for dynamic effects of vehicle loading. Axle weights are represented as point loads and a uniformly distributed load represents the wearing surface and self-weight of the bridge deck. The demands due to dead and live loads are recorded at critical girder locations that coincide with changes in reinforcing details and transitions in built-up sections, as determined from design drawings, as well as other locations of interest (e.g., those observed during field inspection).

In the following sections, Tcl scripts for model building and moving-load analysis in OpenSees are presented. Qualitative analysis of moving loads based on the Muller-Breslau principle and superposition of influence functions is cumbersome to incorporate in a finite-element setting. Tcl is suited to quantitative analysis of moving loads where the position and combination of loads that will produce the maximum demands are determined by repeated structural analyses. The moving load procedures are trivial in two dimensions; however, they are shown here to demonstrate Tcl syntax and lay the foundation for the bridge-rating application presented herein.

2.1 SPECIFICATION OF FINITE-ELEMENT MODEL

To represent a continuous bridge girder in OpenSees, a mesh of nodes and beam elements is generated by the Tcl script defineModel.tcl shown in Figure 2.1. The script invokes the node and element commands, added to the Tcl interpreter by OpenSees, to build the finite-element model using two arrays, Lspan and criticalLocs, defined in the Tcl namespace and dereferenced using a dollar sign ($). The Lspan array contains the length of each span while the criticalLocs array contains the critical locations along each span. To iterate over spans, the foreach loop is preferred over a for loop since the indices of a Tcl array can be any string value, not just integers.

The recorder command is issued at each critical location so that the internal forces will be logged at each step of the live-load analysis. Although omitted from Figure 2.1 for brevity, the section and material properties for each critical location must be defined in the Tcl namespace in order to determine the distribution of forces along the girder during the analysis. One beam element per span is shown in Figure 2.1; however, it is a simple extension to discretize each span (e.g., at tenth points) if desired. The length of the bridge is computed during the model definition and added to the Tcl namespace for use in subsequent scripts.

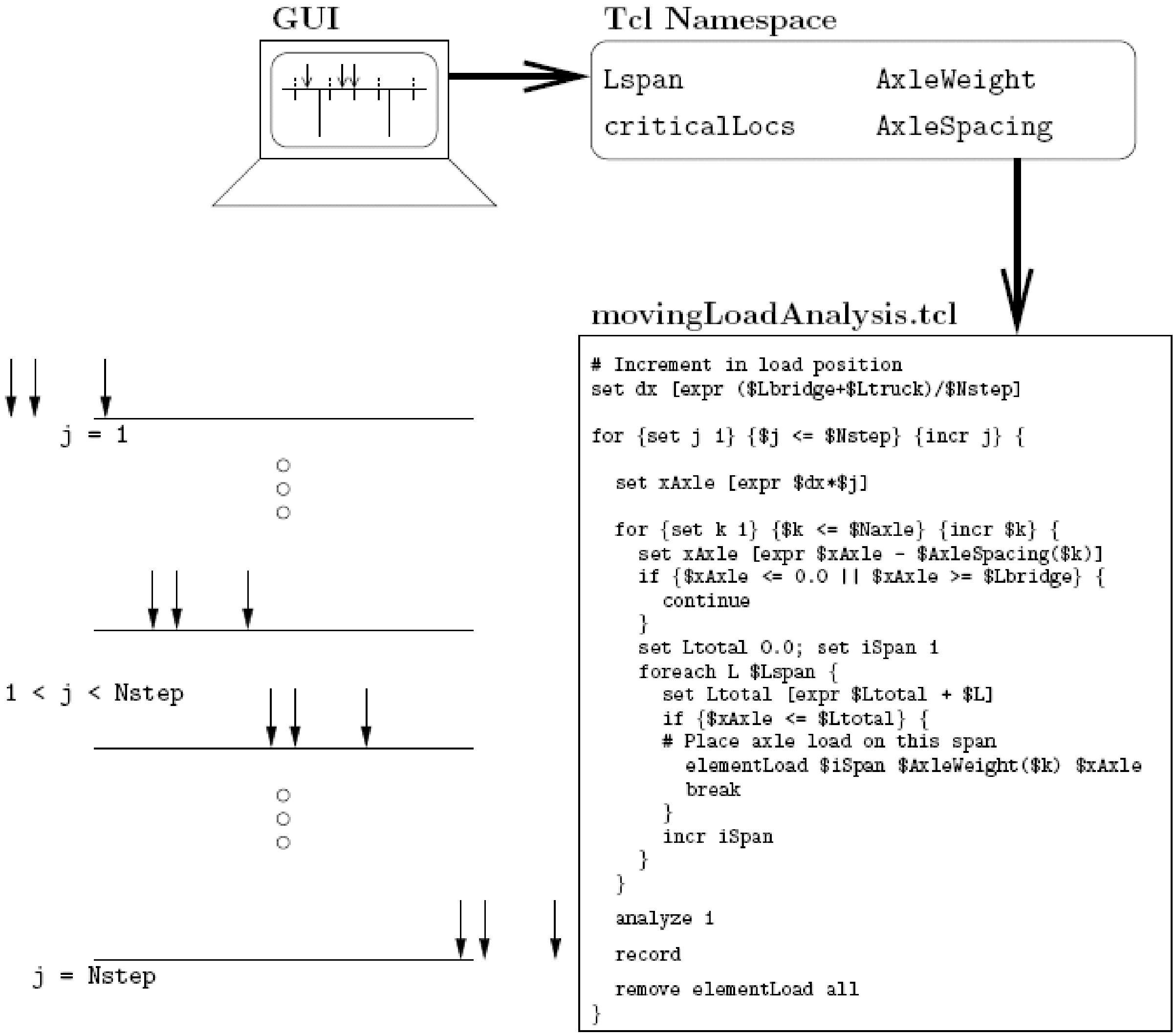


```
# Continuous girder definition
node 0 0.0
set Lbridge 0.0; set i 1
foreach L $Lspan {
  set Lbridge [expr $Lbridge + $L]
  node $i $Lbridge
  element forceBeamColumn $i [expr $i-1] $i <material properties>
  foreach loc $criticalLocs($i) {
    recorder Element $i -location $loc forces
  }
  incr i
}
```

Figure 2.1: Tcl code to create the nodes and finite elements that represent a continuous bridge girder and to record analysis results for each critical location.

2.2 BRIDGE ANALYSIS FOR MOVING LOADS

A Tcl script that moves axle loads across the bridge model is shown in Figure 2.2. In addition to the variables created in the model definition, the arrays AxleWeight and AxleSpacing containing the magnitude and relative spacing of axle weights are required to conduct the moving-load analysis. These arrays can be populated by a user via a script or read from a database. As shown in Figure 2.2, a short control sequence determines where to apply each axle load. After all axle loads are placed on the bridge, the analyze command invokes the structural analysis procedures and finite-element solution methods of the OpenSees framework. Once the demand is recorded at each critical location, all loads are removed from the bridge and then moved to new locations. This process continues until all loads move off the bridge.



```
# Increment in load position
set dx [expr ($Lbridge+$Ltruck)/$Nstep]

for {set j 1} {$j <= $Nstep} {incr j} {

  set xAxle [expr $dx*$j]

  for {set k 1} {$k <= $Naxle} {incr $k} {
    set xAxle [expr $xAxle - $AxleSpacing($k)]
    if {$xAxle <= 0.0 || $xAxle >= $Lbridge} {
      continue
    }
    set Ltotal 0.0; set iSpan 1
    foreach L $Lspan {
      set Ltotal [expr $Ltotal + $L]
      if {$xAxle <= $Ltotal} {
      # Place axle load on this span
        elementLoad $iSpan $AxleWeight($k) $xAxle
        break
      }
      incr iSpan
    }
  }
  analyze 1

  record

  remove elementLoad all
}
```

Figure 2.2: Tcl code to move a multi-axle vehicle loading across a continuous bridge girder.

3.0 TCL PROCEDURES FOR GIRDER CAPACITY SPECIFICATION

This section describes the implementation of a Tcl procedure to compute the moment and shear capacity of reinforced concrete bridge girders. Multiple capacity formats exist, ranging from the use of design equations set out by ACI (2005) to those based on the Modified Compression Field Theory (MCFT) [23]. Regardless of the format, the capacity calculation can be encapsulated by a Tcl procedure. The ACI design equations for moment and shear capacity of reinforced concrete beams demonstrate the approach. The nominal moment capacity is a function of the longitudinal steel reinforcement and the section geometry

$$M_n = A_s f_{yl}(d - \frac{a}{2}) \qquad (3\text{-}1)$$

The depth of the equivalent stress block is

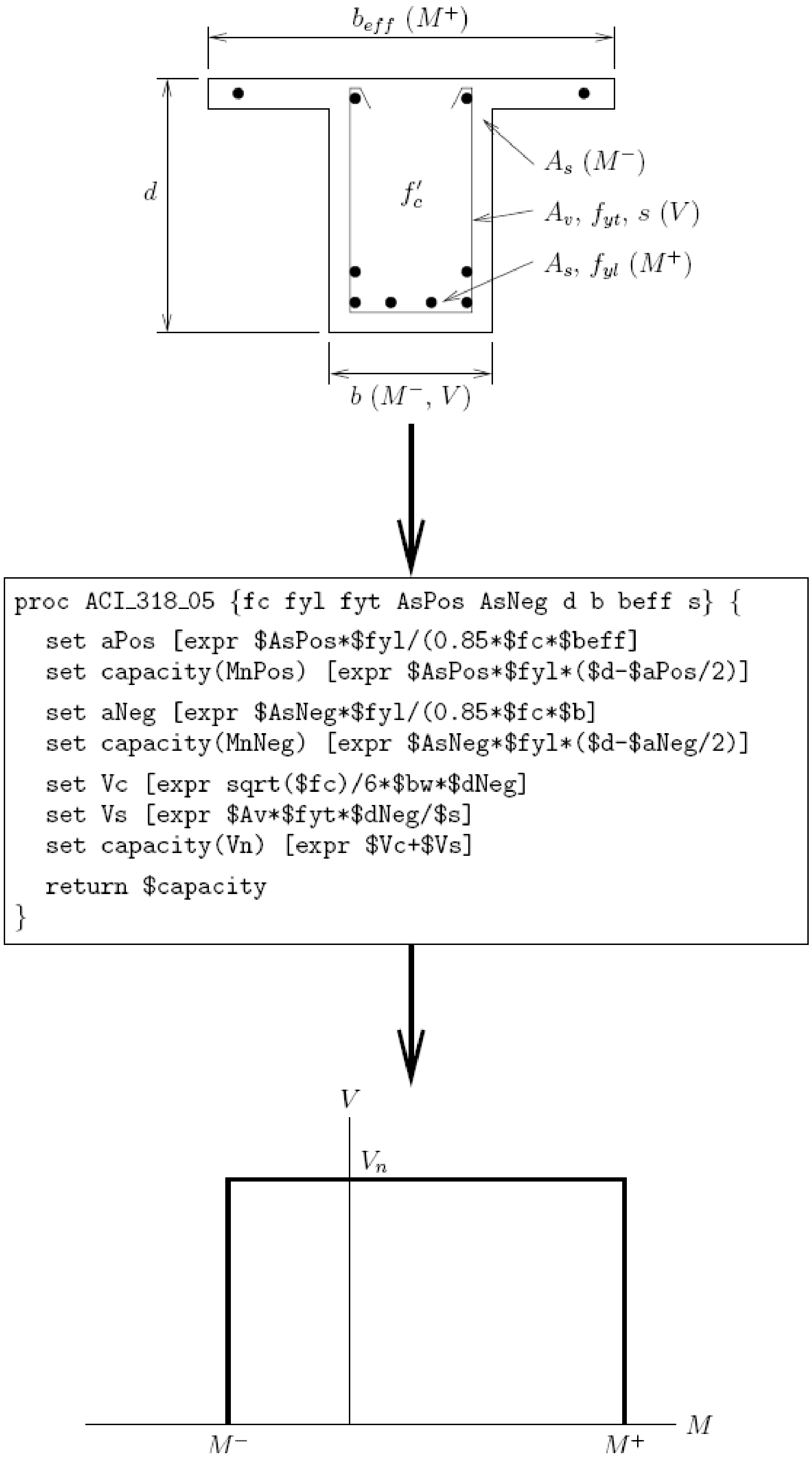
$$a = \frac{A_s f_{yl}}{0.85 f_c^{'} b} \qquad (3\text{-}2)$$

The equation for M_n computes positive and negative moment capacities using the section dimensions shown in Figure 3 for M_+ and M_-. The nominal shear capacity is the sum of contributions from the concrete and transverse steel reinforcement

$$V_n = V_c + V_s = 2\sqrt{f_c^{'}} bd + \frac{A_v f_{yt} d}{s} \qquad (3\text{-}3)$$

Full details of these design equations are found in ACI (2005) and they are implemented in the Tcl procedure ACI 318 05 shown in Figure 3.1. The procedure takes as input the material properties, dimensions and reinforcing details (assuming consistent units) of a reinforced concrete section, and returns a list containing nominal values for shear and positive and negative moment capacity. The use of a Tcl procedure to implement capacity models bypasses the need to compile source code and shortens development time when compared to rating applications written exclusively in a high-level language such as FORTRAN.

The ACI 318 05 procedure is nested in loops in order to compute the moment and shear capacity at every critical girder location, as shown in Figure 3.1. The capacities are stored in the MnPos, MnNeg and Vn arrays for subsequent use in the rating application. The indices of these arrays contain information regarding the span number and critical location, and they can be extended to include the bridge name when creating a database of capacity information for multiple bridges.



```
proc ACI_318_05 {fc fyl fyt AsPos AsNeg d b beff s} {
  set aPos [expr $AsPos*$fyl/(0.85*$fc*$beff]
  set capacity(MnPos) [expr $AsPos*$fyl*($d-$aPos/2)]
  set aNeg [expr $AsNeg*$fyl/(0.85*$fc*$b]
  set capacity(MnNeg) [expr $AsNeg*$fyl*($d-$aNeg/2)]
  set Vc [expr sqrt($fc)/6*$bw*$dNeg]
  set Vs [expr $Av*$fyt*$dNeg/$s]
  set capacity(Vn) [expr $Vc+$Vs]
  return $capacity
}
```

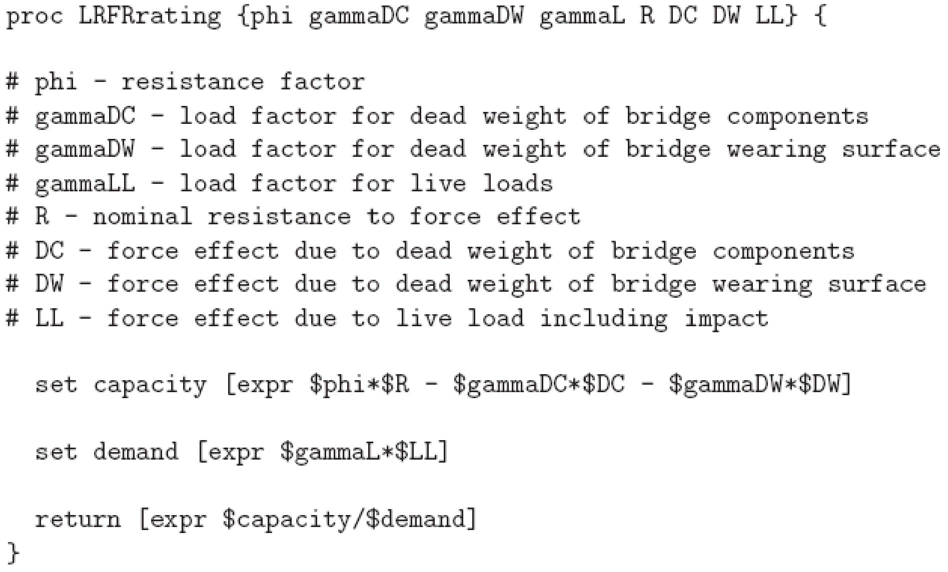
Figure 3.1: Calculation of moment and shear capacities for a girder section using ACI specifications implemented in a Tcl procedure.

4.0 *TCL* COMMANDS FOR RATING FACTOR CALCULATIONS

With the demands and the capacity known at each critical girder location, it is possible to compute a rating factor. Several approaches are available to determine rating factors for bridge girders considering flexure, shear, or some combination of the two. Using procedures and mathematical expressions in Tcl, specification-based rating factors considering individual force effects are straightforward to implement. The most basic rating format is given by Eq. (6-1) in AASHTO-LRFR (2003b),

$$RF = \frac{\phi R - \gamma_{DC} DC - \gamma_{DW} DW}{\gamma_L LL_{IM}} \qquad (4\text{-}1)$$

where R is the nominal resistance to the force effect, as determined from design drawings. Permanent load effects, DC and DW, due to the weight of structural components and wearing surface, respectively, and live-load LLIM accounting for dynamic load effects (via an impact factor) are obtained from the analysis in OpenSees. The resistance factor, φ, and load factors, γ, are tabulated in the LRFR manual for a variety of force effects, bridge types and service states; further details on this rating format are given by [14]. A Tcl procedure to compute rating factors for individual force effects according to Equation 4-1 is shown in Figure 4.1. The treatment of combined force effects (e.g., moment-shear interaction) is not considered; however, the bridge rating application presented herein easily incorporates such effects.



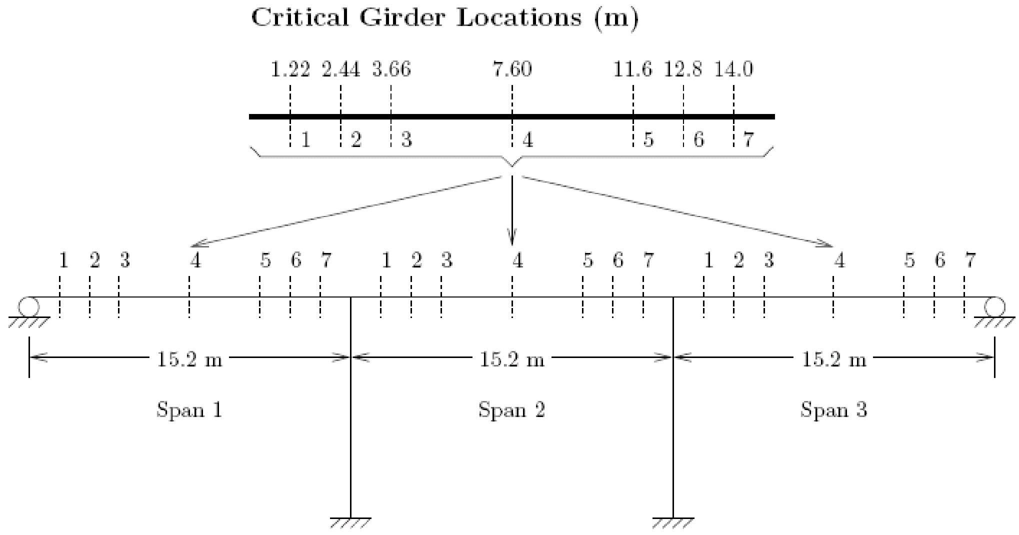
```
proc LRFRrating {phi gammaDC gammaDW gammaL R DC DW LL} {

# phi - resistance factor
# gammaDC - load factor for dead weight of bridge components
# gammaDW - load factor for dead weight of bridge wearing surface
# gammaLL - load factor for live loads
# R - nominal resistance to force effect
# DC - force effect due to dead weight of bridge components
# DW - force effect due to dead weight of bridge wearing surface
# LL - force effect due to live load including impact

  set capacity [expr $phi*$R - $gammaDC*$DC - $gammaDW*$DW]

  set demand [expr $gammaL*$LL]

  return [expr $capacity/$demand]
}
```

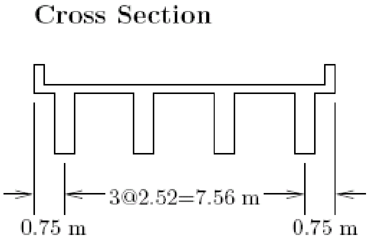
Figure 4.1: Tcl procedure to compute a rating factor based on AASHTO-LRFR specifications.

The use of Tcl makes the rating application an open environment that can incorporate advanced rating methodologies based on risk and reliability [21, 1, 2]. Furthermore, the application allows developers of emergent rating methodologies to utilize the advanced finite-element reliability analysis modules that are available in OpenSees [6] in order to account for a wide range of uncertain input parameters in the bridge rating.

5.0 EXAMPLE BRIDGE-GIRDER RATING APPLICATION

An interior girder of the McKenzie River Bridge, a reinforced concrete deck girder (RCDG) bridge located on Interstate 5 just north of Eugene, Ore., serves as an example for the bridge-rating application using OpenSees and Tcl. An idealized model of the girder is shown in Figure 5.1 as three spans of equal length (50 ft). Critical girder locations are assumed at midspan and d, 2d, and 3d away from the supports, where d = 4ft is the girder depth.





Span	Location	d (cm)	b (cm)	A_s^- (cm^2)	A_s^+ (cm^2)	s (cm)
1	1	122	33	2.58	46.1	30.5
	2	122	33	2.58	60.4	48.3
	3	122	33	2.58	80.5	48.3
	4	122	33	2.58	80.5	48.3
	5	122	33	20.3	47.4	23.0
	6	122	39	40.3	30.2	23.0
	7	122	45	60.4	30.2	23.0
2	1	122	45	60.4	30.2	23.0
	2	122	39	55.0	30.2	23.0
	3	122	33	40.3	30.2	30.5
	4	122	33	2.58	46.6	48.3
Symmetric						

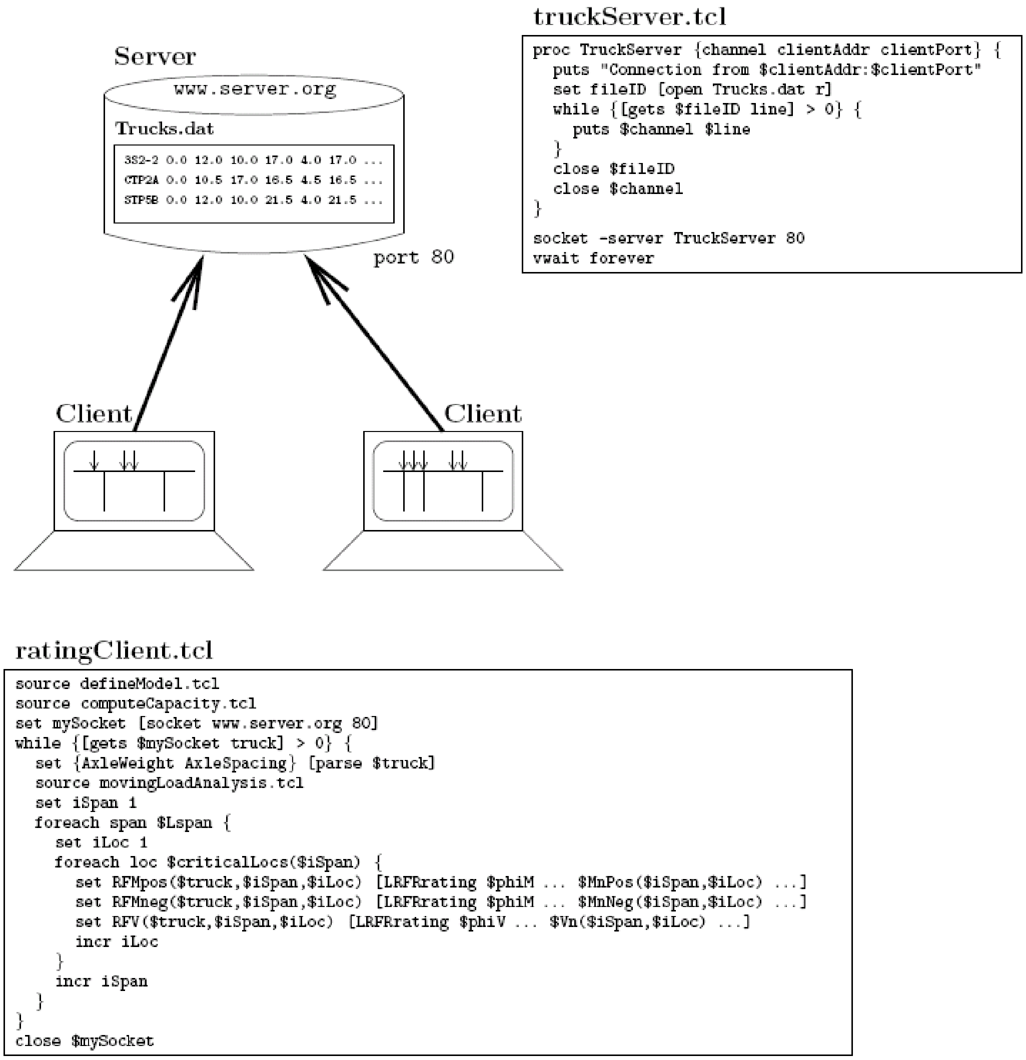
Figure 5.1: Span numbers, lengths, and critical locations for the McKenzie River Bridge.

Distribution factors for moment and shear of the interior girder are calculated as 0.854 and 0.884, respectively, using a Tcl procedure that implements distribution factors for reinforced concrete girder bridges, as prescribed in Section 4.6.2 of AASHTO-LRFD (2003a). An impact factor of 1.33 is used in the analysis. The combined weight of permanent loads (structural components and wearing surface) is estimated as 1.8 kip/ft. [18] present field data and three-dimensional analysis of this bridge for distribution and impact factors.

The girder response remains in the linear-elastic range, as verified by the field tests, and changes in cross-section dimensions due to taper or haunch are accounted for explicitly by numerical integration in force-based finite elements [20, 15]. The integration points coincide with critical girder locations without a significant loss of numerical accuracy [9]. The use of force-based elements to simulate girder response keeps the number of nodes in the finite-element model to a minimum in order to represent the distributed weight of permanent loads. Furthermore, it makes the extension to nonlinear behavior for overload conditions straightforward.

6.0 NETWORK-BASED RATING APPROACH

To demonstrate the network programming features of Tcl, the bridge rating is set up as the client-server application shown in Figure 6.1. The truckServer.tcl script accepts connections from clients who wish to access the load configurations contained in a text file on the server. Clients connect to the server using the socket command in order to retrieve load configurations, then perform analysis and compute rating factors at each critical girder location.



```
3S2-2 0.0 12.0 10.0 17.0 4.0 17.0 ...
CTP2A 0.0 10.5 17.0 16.5 4.5 16.5 ...
STP5B 0.0 12.0 10.0 21.5 4.0 21.5 ...
```

```
proc TruckServer {channel clientAddr clientPort} {
  puts "Connection from $clientAddr:$clientPort"
  set fileID [open Trucks.dat r]
  while {[gets $fileID line] > 0} {
    puts $channel $line
  }
  close $fileID
  close $channel
}

socket -server TruckServer 80
vwait forever
```

```
source defineModel.tcl
source computeCapacity.tcl
set mySocket [socket www.server.org 80]
while {[gets $mySocket truck] > 0} {
  set {AxleWeight AxleSpacing} [parse $truck]
  source movingLoadAnalysis.tcl
  set iSpan 1
  foreach span $Lspan {
    set iLoc 1
    foreach loc $criticalLocs($iSpan) {
      set RFMpos($truck,$iSpan,$iLoc) [LRFRrating $phiM ... $MnPos($iSpan,$iLoc) ...]
      set RFMneg($truck,$iSpan,$iLoc) [LRFRrating $phiM ... $MnNeg($iSpan,$iLoc) ...]
      set RFV($truck,$iSpan,$iLoc) [LRFRrating $phiV ... $Vn($iSpan,$iLoc) ...]
      incr iLoc
    }
    incr iSpan
  }
}
close $mySocket
```

Figure 6.1: Client rating a single bridge for vehicles contained on a server.

After connecting to the server, clients invoke the defineModel.tcl and computeCapacity.tcl scripts in order to define the finite-element model and compute the girder capacity at each critical location. To determine the moment and shear demands, structural analyses are performed for each load configuration by setting the AxleWeight and AxleSpacing arrays, and calling the movingLoadAnalysis.tcl script. Rating factors for moment and shear are then computed at each critical location for each load configuration on the server by invoking the LRFRrating procedure after obtaining the capacity from the appropriate arrays and the demand from the recorder objects of OpenSees. The computed rating factors are stored in arrays using a concatenation of span, critical location and truck information as the indices. Compared to traditional engineering software applications that rely on formatted text files for output, the storage of rating factors in Tcl arrays offers increased flexibility in communicating with databases and graphical user interfaces for storage and visualization.

The rating factors for moment and shear of an interior girder under the OR-STP-5B load configuration (Figure 6.2), representative of single-trip permit (STP) vehicles in Oregon, are shown in Figure 6.3. The finite-element model is symmetric as is the loading, which is moved across the bridge in both directions. Thus only half of the bridge is shown. The example uses resistance factors, _, of 0.9 and 0.75 to rate for moment and shear, respectively, and factors L = 1.5 for live loads and DC = DW = 1.2 for permanent loads. The girder does not rate sufficiently for shear near the bridge supports nor for positive moment at 11.6 m from the approaches, indicating these portions of the bridge require strengthening to support the weight configuration or the bridge may require load posting. The load histories at two critical girder locations in Figure 6.4 compare the demands, including LRFD load factors and distribution and impact factors, imposed for all steps of the moving-load analysis to the nominal and factored capacities for moment and shear. The entire demand history is not required to rate the bridge girder for individual force effects according to Eq. (4); however, it is shown here to demonstrate the application is readily extensible to rating for the interaction of moment and shear.

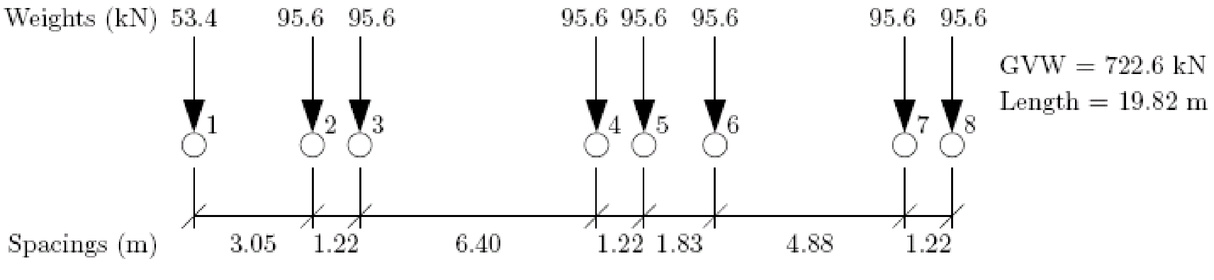
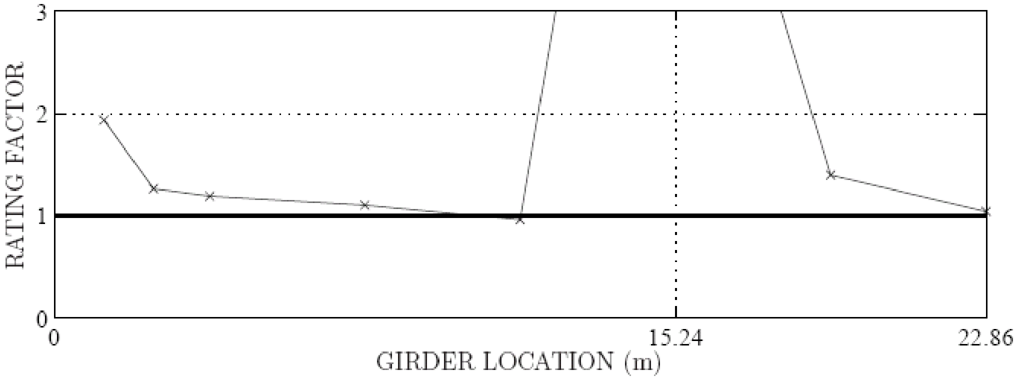
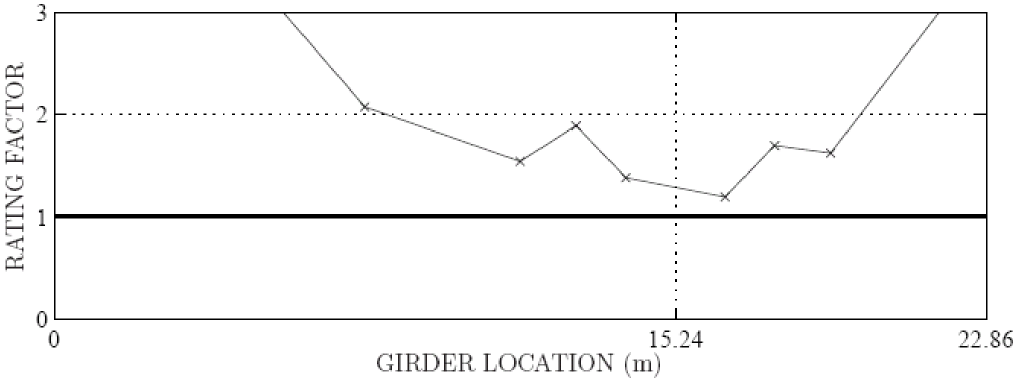


Figure 6.2: Load configuration for the Oregon single-trip permit vehicle, OR-STP-5B (adapted from Oregon Department of Transportation load-rating manuals).

(a) POSITIVE MOMENT



(b) NEGATIVE MOMENT



(c) SHEAR

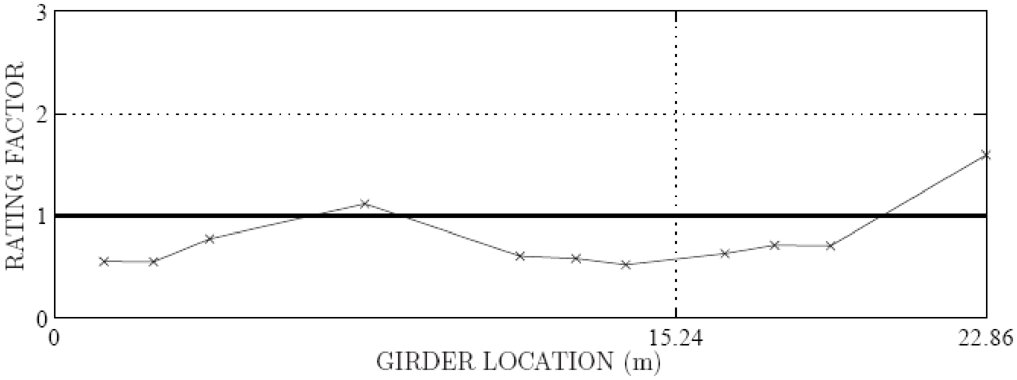
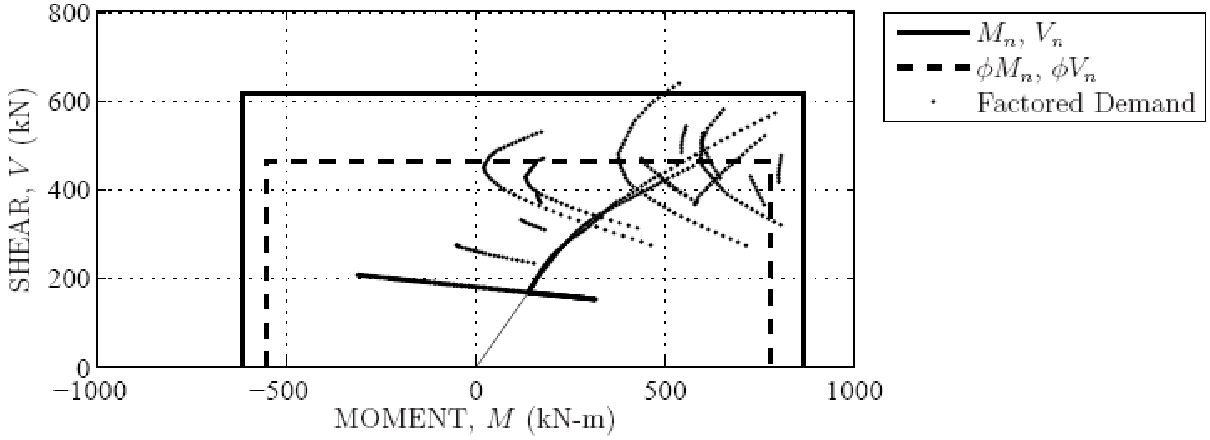


Figure 6.3: Visualization of LRFD rating factors for the OR-STP-5B load configuration on an interior girder of the McKenzie River Bridge considering individual effects: (a) positive moment, (b) negative moment, and (c) shear.

(a) SPAN 1, SECTION 5



(b) SPAN 2, SECTION 1

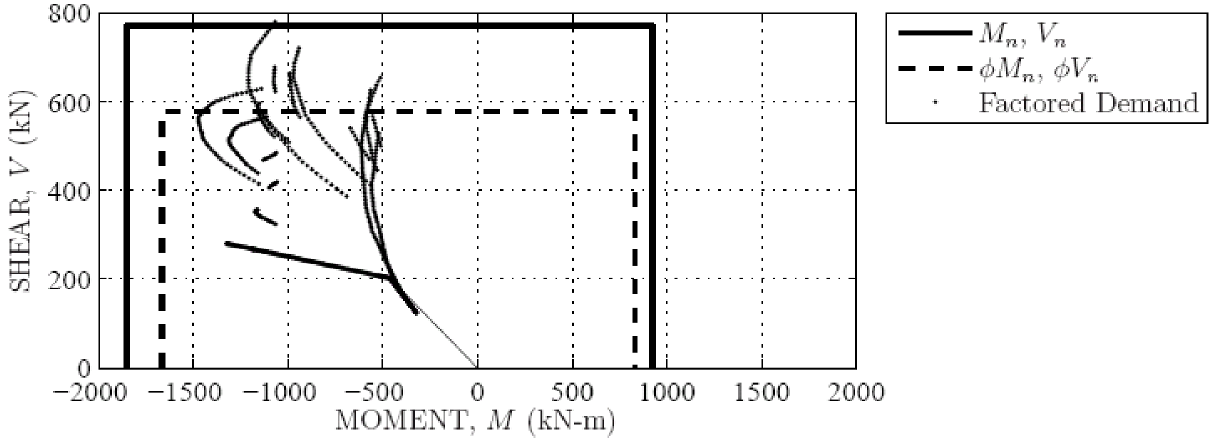


Figure 6.4: Visualization of demand history and capacity envelopes at two critical locations for the OR-STP-5B load configuration applied to an interior girder of the McKenzie River Bridge: (a) 11.6 m along span 1; (b) 1.22 m along span 2.

7.0 CONCLUSIONS

A software application for bridge rating was developed in which the Tcl scripting language glues the finite-element analysis modules of OpenSees to customized procedures for live-load analysis, bridge capacity and rating factor calculations. The application was demonstrated for rating a conventionally reinforced concrete bridge girder for moment and shear effects; however, it is clear that the application is extensible to other structural components and bridge types. Further development will make the rating application a powerful tool to help prioritize highway bridges for targeted repair, rehabilitation and/or replacement.

- As noted in the foregoing discussion, the following benefits are realized by using Tcl to build the bridge-rating application.
- All steps in the application are transparent to a user, who is thus not bound to specific capacity models and rating formats and is free to experiment with existing bridge-rating methodologies (e.g., for site-specific load factors).
- Procedures that implement bridge-capacity models and rating-factor calculations are portable to multiple computing architectures.
- Compilation is not necessary in the interpreted Tcl environment, thereby reducing development times for emergent bridge-rating methodologies.
- The underlying architecture of Tcl allows an application to communicate efficiently with network servers (e.g., to interface with weigh-in-motion data in order to monitor structural health).

In addition to the benefits offered by Tcl, the following extensions of the bridge-rating application are foreseen:

- The simulation of bridge-overload conditions using the nonlinear element and constitutive models of OpenSees;
- Component and system reliability analysis for multi-hazard effects of vehicle, seismic, wave, and blast loads;
- Sensitivity-based applications including structural reliability, optimization, and system identification are possible using the framework for finite-element reliability and sensitivity analysis in OpenSees; and
- Analyzing and rating large systems of bridges in a transportation network while taking advantage of the parallel and distributed computing capabilities of OpenSees.

Given the combined capabilities of Tcl and OpenSees, the bridge-rating application presented in this paper forms the basis for a comprehensive bridge-management system that incorporates the latest advances in simulation models and information technology. Furthermore, Tcl can be used as a glue language in other fields of civil engineering to develop advanced applications from “primitive” building blocks in order to simulate the behavior of transportation, geotechnical, coastal and hydrological systems.

8.0 REFERENCES

[1] F. Akg¨ul and D. M. Frangopol. Rating and reliability of existing bridges in a network. Journal of Bridge Engineering, 8(6):383–393, 2003.

[2] F. Akg¨ul and D. M. Frangopol. Computational platform for predicting lifetime system reliability profiles for different structure types in a network. Journal of Computing in Civil Engineering, 18(2):92–104, 2004.

[3] K. J. Bathe. Finite Element Procedures. Prentice-Hall, Upper Saddle River, NJ, 1996.

[4] R. Brice. Alternate Route Project: Open Source Software for Government. Washington State Department of Transportation, Olympia, WA, 1999.

[5] R. D. Cook, D. S. Malkus, and M. E. Plesha. Concepts and Applications of Finite Element Analysis. John Wiley & Sons, New York, NY, 1989.

[6] T. Haukaas and A. Der Kiureghian. Methods and object-oriented software for FE reliability and sensitivity analysis with application to a bridge structure. Journal of Computing in Civil Engineering, 21(3):151–163, 2007.

[7] C. Higgins, T. K. Daniels, D. V. Rosowsky, T. H. Miller, and S. C. Yim. Assessment and risk ranking of conventionally reinforced concrete bridges for shear. Journal of the Transportation Research Board, 1928:110–117, 2005.

[8] T. C. Hutchinson, F. Kuester, and M. E. Phair. Sketching finite-element models within a unified two-dimensional framework. Journal of Computing in Civil Engineering, 21(3):175–186, 2007.

[9] A. Kidarsa, M. H. Scott, and C. C. Higgins. Analysis of moving loads using force-based finite elements. Finite Elements in Analysis and Design, 44(4):214–224, 2008.

[10] M. Liu and D. M. Frangopol. Probability-based bridge network performance evaluation. Journal of Bridge Engineering, 11(5):633–641, 2006.

[11] S. Mazzoni, F. McKenna, M. H. Scott, and G. L. Fenves. Open System for Earthquake Engineering Simulation User Command-Language Manual – Version 1.7.3. University of California, Berkeley, CA, 2006. http://opensees.berkeley.edu/OpenSees/manuals/usermanual/.

[12] F. McKenna. Object-Oriented Finite Element Programming: Frameworks for Analysis, Algorithms, and Parallel Computing. PhD thesis, University of California, Berkeley, CA, December 1997.

[13] F. McKenna, G. L. Fenves, and M. H. Scott. Open System for Earthquake Engineering Simulation. University of California, Berkeley, CA, 2000. http://opensees.berkeley.edu.

[14] C. Minervino, B. Sivakumar, F. Moses, D. Mertz, and W. Edberg. New AASHTO guide manual for load and resistance factor rating of highway bridges. Journal of Bridge Engineering, 9(1):43–54, 2004.

[15] A. Neuenhofer and F. C. Filippou. Evaluation of nonlinear frame finite-element models. Journal of Structural Engineering, 123(7):958–966, July 1997.

[16] J. K. Ousterhout. Tcl and the Tk Toolkit. Addison-Wesley, Reading, MA, 1994.

[17] J. Peng and K. H. Law. A prototype software framework for internet-enabled collaborative development of a structural analysis program. Engineering With Computers, 18(1):38–49, 2002.

[18] T. Potisuk and C. Higgins. Field testing and analysis of CRC deck girder bridges. Journal of Bridge Engineering, 12(1):53–63, 2007.

[19] A. Schellenberg and S. Mahin. Integration of hybrid simulation within the general purpose

computational framework OpenSees. In Eighth U.S. National Conference on Earthquake Engineering, San Francisco, CA, 2006.

[20] E. Spacone, F. C. Filippou, and F. F. Taucer. Fiber beam-column model for nonlinear analysis of R/C frames: Formulation. Earthquake Engineering and Structural Dynamics, 25:711–725, 1996.

[21] M. G. Stewart, D. V. Rosowsky, and D. Val. Reliability-based bridge assessment using risk-ranking decision analysis. Structural Safety, 23:397–405, 2001.

[22] Y. Takahashi and G. L. Fenves. Software framework for distributed experimental-computational simulation of structural systems. Earthquake Engineering and Structural Dynamics, 35(3):267–291, 2006.

[23] F. J. Vecchio and M. P. Collins. Predicting the response of reinforced concrete beams subjected to shear using the modified compression field theory. ACI Structural Journal, 85(S27):258–268, 1988.

[24] O. C. Zienkiewicz and R. L. Taylor. The Finite Element Method: Volume 1, The Basis. Butterworth-Heinman, Stoneham, MA, 5th edition, 2000.

OTREC
OREGON TRANSPORTATION RESEARCH
AND EDUCATION CONSORTIUM

P.O. Box 751
Portland, OR 97207

OTREC is dedicated to stimulating and conducting collaborative multi-disciplinary research on multi-modal surface transportation issues, educating a diverse array of current practitioners and future leaders in the transportation field, and encouraging implementation of relevant research results.