การออกแบบโปรแกรมเชิงวัตถุสำหรับการประเมินการคงสภาพของท่อที่มีรอยร้าวด้วยระเบียบวิธี R6

นายเนติพันธ์ พุทธรักษ์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2551

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DESIGN OF OBJECT-ORIENTED PROGRAM FOR CRACKED PIPE INTEGRITY ASSESSMENT USING R6 PROCEDURE

Mr. Natipan Puttarak

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Mechanical Engineering
Department of Mechanical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2008
Copyright of Chulalongkorn University

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หัวข้อวิทยานิพนธ์	การออกแบบโปรแกรมเชิงวัตถุสำหรับการประเมินการคงสภาพของท่อที่มีรอยร้าวด้วยระเบียบวิธี R6
โดย	นายเนติพันธ์ พุทธรักษ์
สาขาวิชา	วิศวกรรมเครื่องกล
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.จิรพงศ์ กสิวิทย์อำนวย

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(รองศาสตราจารย์ ดร.ชัยโรจน์ คุณพนิชกิจ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.จิรพงศ์ กสิวิทย์อำนวย)

.....กรรมการ
(อาจารย์ ดร.ชัญญาพันธ์ วิรุฬห์ศรี)

.....กรรมการ
(อาจารย์ ดร. สวัสดิ์ เหลืองเรืองฤทธิ์)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

เนติพันธ์ พุทธรักษ์ : การออกแบบโปรแกรมเชิงวัตถุสำหรับการประเมินการคงสภาพของท่อที่มีรอยร้าวด้วยระเบียบวิธี R6 (DESIGN OF OBJECT-ORIENTED PROGRAM FOR CRACKED PIPE INTEGRITY ASSESSMENT USING R6 PROCEDURE) อ.ที่ปรึกษาวิทยานิพนธ์: ผศ. ดร. จิรพงศ์ กสิวิทย์อำนวย, 341 หน้า.

วิทยานิพนธ์นี้ออกแบบโปรแกรมเชิงวัตถุสำหรับการประเมินการคงสภาพของท่อที่มีรอยร้าวด้วยระเบียบวิธี R6 โปรแกรมสามารถประเมินท่อที่มีรอยร้าวผิวรูปครึ่งวงรี 2 ชนิด คือ ชนิดแรกรอยร้าววางตัวภายในตามแนวเส้นรอบวงและชนิดที่สองวางตัวภายในตามแนวแกน รอยร้าวชนิดแรกรับภาระแรงดึงตามแนวแกน และโมเมนต์ดัด ส่วนรอยร้าวชนิดที่สองรับภาระความดันภายในโดยใช้การประเมินในระดับที่ 1 และระดับที่ 2 กับแผนภาพการประเมิน FAD ประเภทที่ 1 และประเภทที่ 2 ได้ นอกจากนี้ยังมีส่วนของโปรแกรมที่สามารถจัดการกับรอยร้าวกรณีโครงสร้างมีรอยร้าวปรากฏอยู่ใกล้กันสองรอยโดยใช้วิธีการตามข้อแนะนำของมาตรฐาน API 579

การออกแบบโปรแกรมประยุกต์ใช้วิธีของ Rumbaugh et al. มีขั้นตอนคือ 1) เขียนเนื้อความของปัญหา 2) หาคลาสจากเนื้อความของปัญหา 3) สร้างคลาสไดอะแกรม 4) ปรับปรุงคลาสไดอะแกรม และ5) สร้าง sequence diagram

การตรวจสอบโปรแกรมแบ่งเป็นการตรวจสอบผลการระบุลักษณะรอยร้าวและผลการประเมินโดยนำไปเปรียบเทียบกับผลที่ได้จากโปรแกรม MathCAD จากการตรวจสอบพบว่า ผลการระบุลักษณะรอยร้าวและผลการประเมินมีค่าตรงกัน

ในอนาคตหากมีความต้องการปรับปรุงเพิ่มเติมขีดความสามารถของโปรแกรมให้สามารถประเมินโครงสร้างที่หลากหลาย รอยร้าวชนิดอื่น ๆ หรือระดับการประเมินและประเภทของแผนภาพที่นอกเหนือจากนี้ โปรแกรมที่ออกแบบสามารถทำได้ง่ายเนื่องจากโปรแกรมถูกออกแบบด้วยแนวคิดเชิงวัตถุ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมเครื่องกล ลายมือชื่อนิสิต เนติพันธ์ พุทธรักษ์

สาขาวิชา วิศวกรรมเครื่องกล ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก

ปีการศึกษา 2551

4870590221 : MAJOR MECHANICAL ENGINEERING

KEYWORDS : R6 PROCEDURE / OBJECT-ORIENTED CONCEPT / CRACK CHARACTERIZATION / CRACK GROWTH

NATIPAN PUTTARAK : DESIGN OF OBJECT-ORIENTED PROGRAM FOR CRACKED PIPE INTEGRITY ASSESSMENT USING R6 PROCEDURE. ADVISOR : ASST.PROF. JIRAPONG KASIVITAMNUAY, D.ENG. 341 pp.

This thesis develops a cracked pipe's integrity assessment software based on an R6 procedure. The program's structure is designed using an object-oriented concept. The program can analyze a pipe containing a semi-elliptical surface crack locates at an inner wall and orients in circumferential or axial directions. For the case of a circumferential crack, the program can assess the integrity of a pipe subjected to an axial force or a bending moment or an internal pressure. For the case of an axial crack, the program can assess the integrity of a pipe subjected to an internal pressure or an arbitrarily distributed circumferential stress. The program can perform levels 1 and 2 assessment and has two options of a failure assessment diagram. Furthermore, the program has a crack characterization routine for the case of a pipe having 2 cracks and has a crack growth routine for calculating a change of crack size due to fatigue loading.

Steps in the program's structure development start from: writing down a problem statement, identifying classes from the problem statement, creating a class diagram, refining a class diagram and creating a sequence diagram.

Validation of the program is divided into 2 parts: 1) validating the crack characterization and 2) validating the integrity assessment. The validation results show that the program can correctly characterize the cracks and assess the integrity of a pipe.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Department : Mechanical Engineering Student's Signature .....

Field of Study : Mechanical Engineering Advisor's Signature .....

Academic Year : 2008

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงลงได้ด้วยความช่วยเหลือจากบุคคลหลายท่าน โดยเฉพาะการช่วยเหลืออย่างดียิ่งของ ผู้ช่วยศาสตราจารย์ ดร. จิรพงศ์ กสิวิทย์อำนวย อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงที่ท่านได้ให้ความความรู้ คำปรึกษา และแนวคิดที่มีค่ายิ่งในการทำวิทยานิพนธ์และตลอดจนการทำงานในอนาคต

ขอกราบขอบพระคุณคณะกรรมการทุกท่านที่ให้แนวทางและข้อเสนอแนะสำหรับการปรับปรุงวิทยานิพนธ์ จนทำให้วิทยานิพนธ์ฉบับนี้มีความสมบูรณ์ยิ่งขึ้น

ขอบคุณเพื่อน ๆ และคนรอบข้างที่ให้กำลังใจและช่วยเหลือในทุก ๆ เรื่องด้วยความเต็มใจอย่างดียิ่งตลอดเวลา

สุดท้ายขอกราบขอบพระคุณบิดา มารดา และพี่สาวที่สนับสนุนด้านการศึกษา และคอยช่วยเหลือทุก ๆ เรื่อง ขอขอบคุณน้องชายสำหรับกำลังใจ ซึ่งตลอดระยะเวลาที่ผ่านมาทำให้ผู้วิจัยสามารถทำวิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

สารบัญตาราง..... ฑ

สารบัญภาพ..... ฌ

คำอธิบายสัญลักษณ์และคำย่อ..... ท

บทที่ 1 **บทนำ**

1.1 ความสำคัญและที่มาของวิทยานิพนธ์..... 1

1.2 วัตถุประสงค์ของวิทยานิพนธ์..... 6

1.3 ขอบเขตของวิทยานิพนธ์..... 7

1.4 วิธีดำเนินงาน..... 7

1.5 ประโยชน์ที่คาดว่าจะได้รับ..... 7

บทที่ 2 **กลศาสตร์การแตกหักยืดหยุ่นเชิงเส้น**

2.1 โหมดของการเสียรูปที่ปลายรอยร้าว..... 8

2.2 พารามิเตอร์ปลายรอยร้าว..... 9

2.3 พฤติกรรมการแตกหัก..... 11

2.4 เกณฑ์การแตกหัก..... 12

2.5 การเติบโตของรอยร้าวล้า..... 14

2.6 การเติบโตของรอยร้าวเนื่องจากสิ่งแวดล้อม..... 16

บทที่ 3 **ระเบียบวิธี** R6

3.1 ขอบเขตและภาพรวมของขั้นตอนการประเมิน..... 18

3.2 การแบ่งประเภทของภาระและวิเคราะห์ความเค้น..... 19

3.3 การหาสมบัติแรงดึงของวัสดุ..... 20

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

3.4 การเลือกแผนภาพประเมินความเสียหาย..... 20
3.5 การระบุลักษณะของรอยร้าว..... 22
3.5.1 การระบุรูปร่างอุดมคติของรอยร้าว..... 22
3.5.2 การจัดทิศทางรอยร้าว..... 23
3.5.3 การพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว..... 25
3.5.4 การระบุชนิดรอยร้าวซ้ำอีกครั้ง..... 26
3.6 การเลือกระดับการประเมิน..... 29
3.7 การหาค่าความต้านทานการแตกหัก..... 29
3.8 การระบุขนาดของรอยร้าว..... 30
3.9 การคำนวณอัตราส่วนภาระ L_r 30
3.10 การคำนวณอัตราส่วนพารามิเตอร์ K หรือ K_r 30
3.11 พล็อตจุด (L_r, K_r) บน FAD..... 32
3.12 การพิจารณาว่าต้องการวิเคราะห์การเติบโตของรอยร้าวหรือไม่..... 32
3.13 การประเมินนัยสำคัญของผลการประเมิน..... 33
3.14 การประเมินให้ผลลัพธ์ยอมรับได้หรือไม่..... 35
3.14.1 การประเมินด้วยระดับที่สูงขึ้น..... 35
3.14.2 การระบุลักษณะรอยร้าวซ้ำอีกครั้ง..... 36
3.14.3 การเลือก FAD แบบอื่น..... 37
3.14.4 วิเคราะห์ความเค้นให้แม่นยำขึ้น..... 37

บทที่ 4 ความรู้เบื้องต้นเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุ

4.1 ข้อจำกัดของแนวคิดเชิงกระบวนการ..... 38
4.2 ข้อได้เปรียบของแนวคิดเชิงวัตถุ..... 38
4.3 ออบเจ็กต์และคลาส..... 40
4.4 การจัดกลุ่ม..... 42
4.5 ลำดับชั้นของคลาส..... 42
4.6 คุณสมบัติของโปรแกรมเชิงวัตถุ..... 45
4.6.1 การควบคุมการเข้าถึง..... 45
4.6.2 การสืบทอด..... 46

หน้า

4.6.3 การมีหลายรูปแบบ..... 47
4.7 Design Pattern..... 48

บทที่ 5 การออกแบบโครงสร้างของโปรแกรมและผลการออกแบบ

5.1 เนื้อความของปัญหา..... 50
5.2 การหาคลาสจากเนื้อความของปัญหา..... 51
5.3 การสร้างคลาสไดอะแกรม..... 52
5.4 การปรับปรุงคลาสไดอะแกรม..... 54
5.4.1 การปรับปรุงคลาส..... 54
5.4.1.1 การปรับปรุงคลาส pipe..... 55
5.4.1.2 การปรับปรุงคลาส crack..... 55
5.4.1.3 การปรับปรุงคลาส FAD…..... 56
5.4.1.4 การปรับปรุงคลาส SIF..... 56
5.4.1.5 การปรับปรุงคลาส crackGrowth..... 57
5.4.2 การเพิ่มเติมคลาส..... 58
5.4.2.1 การเพิ่มคลาส standard..... 58
5.4.2.2 การเพิ่มคลาส database..... 59
5.4.2.3 การเพิ่มคลาสในส่วนติดต่อกับผู้ใช้งาน..... 61
5.5 sequence diagram..... 61
5.6 รายละเอียดของคลาสและแต่ละคลาส..... 65
5.6.1 รายละเอียดของคลาส pipe…..... 65
5.6.2 รายละเอียดของคลาส matProp…..... 66
5.6.3 รายละเอียดของคลาส crack…..... 69
5.6.4 รายละเอียดของคลาส load…..... 70
5.6.5 รายละเอียดของคลาส assessment..... 72
5.6.6 รายละเอียดของคลาส charCrack…..... 73
5.6.7 รายละเอียดของคลาส charStd…..... 74
5.6.8 รายละเอียดของคลาส limitLoad..... 74
5.6.9 รายละเอียดของคลาส SIF..... 74

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

5.6.10 รายละเอียดของคลาส crackGrowth…..... 75
5.6.11 รายละเอียดของคลาส FAD…..... 75
5.7 ผลการออกแบบโปรแกรม..... 76
5.7.1 ส่วนติดต่อกับผู้ใช้งาน..... 76
5.7.2 ส่วนการประเมิน..... 83

บทที่ 6 การตรวจสอบความถูกต้องโปรแกรม

6.1 การตรวจสอบการระบุลักษณะรอยร้าว..... 84
6.1.1 กรณีระหว่างรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย..... 84
6.1.2 กรณีระหว่างรอยร้าวฝังรูปวงรีทั้งสองรอย..... 90
6.1.3 กรณีระหว่างรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี..... 94
6.2 การตรวจสอบผลการประเมิน..... 99
6.2.1 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวง
รับภาระแรงดึงแนวแกน..... 100
6.2.1.1 ข้อมูล..... 100
6.2.1.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1
และระดับที่ 2..... 101
6.2.1.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1
และระดับที่ 2..... 102
6.2.2 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวง
รับภาระโมเมนต์ดัด..... 102
6.2.2.1 ข้อมูล..... 103
6.2.2.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1
และระดับที่ 2..... 104
6.2.2.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1
และระดับที่ 2..... 105
6.2.3 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระ
ความดันภายใน..... 106
6.2.3.1 ข้อมูล..... 106

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

6.2.3.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และระดับที่ 2..... 107
6.2.3.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และระดับที่ 2..... 109

บทที่ 7 สรุปผล

7.1 สรุปเกี่ยวกับการออกแบบและการตรวจสอบโปรแกรม..... 112
7.2 ข้อจำกัดของโปรแกรม..... 113
7.3 แนะนำงานวิจัยต่อเนื่อง..... 114

รายการอ้างอิง..... 115

ภาคผนวก..... 117
ภาคผนวก ก..... 118
ก.1 คลาสไดอะแกรม..... 118
ก.2 คลาสไดอะแกรมที่รวมส่วนติดต่อกับผู้ใช้งานแล้ว..... 119
ก.3 sequence diagram ของการระบุลักษณะรอยร้าว..... 120
ก.4 sequence diagram ของการประเมิน..... 121
ภาคผนวก ข..... 122
ข.1 การระบุลักษณะรอยร้าว..... 122
ข.1.1 การระบุลักษณะรอยร้าวกรณีรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย..... 122
ข.1.2 การระบุลักษณะรอยร้าวกรณีรอยร้าวฝังรูปวงรีทั้งสองรอย..... 128
ข.1.3 การระบุลักษณะรอยร้าวกรณีรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี..... 134
ข.2 การประเมินการคงสภาพ..... 140
ข.2.1 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระแรงดึงตามแนวแกน..... 140

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

ข.2.1.1 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1..... 140
ข.2.1.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 2..... 145
ข.2.1.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1..... 152
ข.2.1.4 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 2..... 157
ข.2.2 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระโมเมนต์ดัด..... 165
ข.2.2.1 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1..... 165
ข.2.2.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 2..... 170
ข.2.2.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1..... 178
ข.2.2.4 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 2..... 184
ข.2.3 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายใน..... 193
ข.2.3.1 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1..... 193
ข.2.3.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 2..... 197
ข.2.3.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1..... 204
ข.2.3.4 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 2..... 209
ข.3 การประเมินการเติบโตของรอยร้าวล้า..... 216
ภาคผนวก ค..... 218
ค.1 ซอร์สโค้ดของฟอร์ม base..... 218
ค.2 ซอร์สโค้ดของฟอร์ม userInterface..... 219
ค.3 ซอร์สโค้ดของฟอร์ม structure..... 242
ค.4 ซอร์สโค้ดของฟอร์ม pipe..... 243
ค.5 ซอร์สโค้ดของฟอร์ม matProp..... 246
ค.6 ซอร์สโค้ดของฟอร์ม crack..... 253
ค.7 ซอร์สโค้ดของฟอร์ม crackGrowth..... 267
ค.8 ซอร์สโค้ดของคลาส structureFactory..... 271
ค.9 ซอร์สโค้ดของคลาส structure..... 272
ค.10 ซอร์สโค้ดของคลาส crack..... 281
ค.11 ซอร์สโค้ดของคลาส load..... 284

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

ค.12 ซอร์สโค้ดของคลาส matProp..... 286
ค.13 ซอร์สโค้ดของคลาส charCrack..... 289
ค.14 ซอร์สโค้ดของคลาส charStd..... 293
ค.15 ซอร์สโค้ดของคลาส crackTemp..... 305
ค.16 ซอร์สโค้ดของคลาส assessment..... 307
ค.17 ซอร์สโค้ดของคลาส limitLoad…..... 324
ค.18 ซอร์สโค้ดของคลาส SIF…..... 328
ค.19 ซอร์สโค้ดของคลาส FAD…..... 336
ค.20 ซอร์สโค้ดของคลาส crackGrowth…..... 338
ค.21 ซอร์สโค้ดของคลาส constitutive…..... 340

ประวัติผู้เขียนวิทยานิพนธ์..... 341

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตารางที่		หน้า
3.1	การระบุรูปร่างอุดมคติของรอยร้าว	23
3.2	ปฏิสัมพันธ์ของรอยร้าวต่างระนาบ (ระเบียบวิธี API 579)	27
3.3	ปฏิสัมพันธ์ของรอยร้าวบนระนาบเดียวกัน (ระเบียบวิธี API 579)	28
5.1	แอตทริบิวต์ของคลาส pipe และคำอธิบาย	65
5.2	เมทธอดของคลาส pipe และคำอธิบาย	65
5.3	แอตทริบิวต์ของคลาส matProp และคำอธิบาย	67
5.4	เมทธอดของคลาส matProp และคำอธิบาย	68
5.5	แอทตริบิวต์ของคลาส crack และคำอธิบาย	69
5.6	เมทธอดของคลาส crack และคำอธิบาย	70
5.7	แอตทริบิวต์ของคลาส load และคำอธิบาย	71
5.8	เมทธอดของคลาส load และคำอธิบาย	71
5.9	แอตทริบิวต์ของคลาส assessment และคำอธิบาย	73
5.10	เมทธอดของคลาส assessment และคำอธิบาย	73
5.11	เมทธอดของคลาส crackGrowth และคำอธิบาย	75
6.1	รายละเอียดของรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย	86
6.2	ผลการคำนวณความยาวและความลึกรอยร้าวสมมูลของรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย	88
6.3	ผลการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย	89
6.4	รายละเอียดของรอยร้าวฝังรูปวงรีทั้งสองรอย	91
6.5	ผลการคำนวณความยาวและความลึกรอยร้าวสมมูลของรอยร้าวฝังรูปวงรีทั้งสองรอย	92
6.6	ผลการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าวฝังรูปวงรีทั้งสองรอย	94
6.7	รายละเอียดของรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี	96
6.8	ผลการคำนวณความยาวและความลึกรอยร้าวสมมูลของรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี	98
6.9	ผลการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี	99

ตารางที่ หน้า

6.10 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระแรงดึงแนวแกน (FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และ2)..... 101
6.11 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระแรงดึงแนวแกน (FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และ 2)..... 102
6.12 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระโมเมนต์ดัด (FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และ 2)..... 104
6.13 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระโมเมนต์ดัด (FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และ 2)..... 105
6.14 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตาม แนวแกนรับภาระความดันภายใน(FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และ 2)..... 107
6.15 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายในและภาระความดันเปลี่ยนแปลง (FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และ 2)..... 108
6.16 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายใน (FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และ 2)..... 110
6.17 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายในและภาระความดันเปลี่ยนแปลง (FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และ 2)..... 111

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

รูปที่		หน้า
1.1	แผนภาพประเมินความเสียหายของ Dowling และ Townley….....	2
1.2	แผนภาพประเมินความเสียหายของ Dowling และ Townley ที่ปรับปรุงแล้ว.....	3
1.3	การทวนสอบแผนภาพประเมินความเสียหายของ Dowling และ Townley ที่ปรับปรุงแล้ว.....	3
1.4	แผนภาพประเมินความเสียหายของ Harrison et al.....	4
1.5	ข้องอมีรอยร้าวทะลุความหนาตามแนวรัศมีและตามแนวแกน.....	5
1.6	ท่อรูปตัวที.....	6
2.1	โหมดการเสียรูปที่ปลายรอยร้าว.....	9
2.2	องค์ประกอบความเค้นบริเวณปลายรอยร้าว.....	9
2.3	พฤติกรรมการแตกหักที่ปลายรอยร้าว	
	(ก) การแตกหักเปราะ.....	12
	(ข) การแตกหักเหนียว.....	12
3.1	แผนผังขั้นตอนการประเมินการคงสภาพด้วยระเบียบวิธี R6.....	19
3.2	การจัดทิศทางรอยร้าวเฉียงตามระเบียบวิธี API 579 วิธีที่ 2	
	(ก) รอยร้าวและสนามความเค้น.....	24
	(ข) รอยร้าวที่จัดทิศทางแล้ว (กรณี $\alpha \leq 45^{\circ}$).....	24
3.3	การระบุความลึกของรอยร้าว ตามมาตรฐาน API 579	
	(ก) การฉายระนาบรอยร้าว.....	25
	(ข) ความลึกสมมูล.....	25
3.4	ผลของตำแหน่งสัมพัทธ์ของรอยร้าวต่อค่าของพารามิเตอร์ K.....	26
3.5	ผลของ ภาระ ขนาดรอยร้าว ความต้านทายการแตกหัก และความเค้นคราก ต่อตำแหน่ง และทิศทางการเคลื่อนที่ของจุดประเมิน.....	34
3.6	การระบุลักษณะรอยร้าวซ้ำอีกครั้งในกรณีรอยร้าวฝังรูปวงรี.....	36
3.7	การระบุลักษณะรอยร้าวซ้ำอีกครั้งในกรณีรอยร้าวผิวรูปครึ่งวงรี.....	36
4.1	การสื่อสารระหว่าง Customer A และ SalesPerson B.....	40
4.2	คลาส Customers และคลาส SalesPerson.....	41
4.3	สัญลักษณ์แสดงคลาส (กรณีคลาส SalesPerson).....	42

รูปที่		หน้า
4.4	ความสัมพันธ์ของคลาส Employee คลาส SalesPerson และคลาส Manager	43
4.5	โครงร่างลำดับชั้นของคลาส Employee	43
4.6	การสร้างคลาสทั่วไปจากคลาสที่เฉพาะเจาะจงมากกว่า	44
4.7	การสร้างคลาสเฉพาะเจาะจงจากคลาสทั่วไป	44
4.8	ลำดับชั้นของคลาสบุคคล	45
4.9	การสืบทอดของคลาส	46
4.10	คลาสไดอะแกรมความสัมพันธ์การสืบทอดของรูปทรงเรขาคณิต	47
4.11	คลาสไดอะแกรมสำหรับ Factory pattern	49
5.1	คลาสไดอะแกรมกลุ่มโครงสร้าง	52
5.2	คลาสไดอะแกรมกลุ่มการคำนวณ	53
5.3	คลาสไดอะแกรมหลังรวมกลุ่มโครงสร้างและกลุ่มการคำนวณ	54
5.4	คลาส pipe ที่ถูกปรับปรุงด้วยแนวคิดการสร้างลักษณะทั่วไปและการสืบทอด	55
5.5	คลาส crack ที่ถูกเพิ่มเติมด้วยแนวคิดการสร้างลักษณะเฉพาะและการสืบทอด	55
5.6	คลาส FAD ที่ถูกเพิ่มเติมด้วยแนวคิดการสร้างลักษณะเฉพาะและการสืบทอด	56
5.7	คลาส SIF ที่ถูกเพิ่มเติมด้วยแนวคิดการสร้างลักษณะเฉพาะและการสืบทอด	56
5.8	คลาส crackGrowth ที่ถูกเพิ่มเติมด้วยแนวคิดการสร้างลักษณะเฉพาะและการสืบทอด	57
5.9	คลาสไดอะแกรมหลังปรับปรุงคลาสด้วยแนวคิดเชิงวัตถุ	58
5.10	คลาส standard ที่ถูกเพิ่มเติมด้วยแนวคิดการสร้างลักษณะเฉพาะและการสืบทอด	59
5.11	คลาส dataBase ที่ถูกเพิ่มเติม	59
5.12	คลาสไดอะแกรมหลังปรับปรุงด้วยสมบัติของแนวคิดเชิงวัตถุและเพิ่มเติมคลาสที่เกี่ยวข้องแล้ว	60
5.13	sequence diagram ของการระบุลักษณะรอยร้าว	62
5.14	sequence diagram ของการประเมินการคงสภาพ	63
5.15	sequence diagram ของการประเมินการเติบโตของรอยร้าว	64
5.16	แอตทริบิวต์และเมทธอดของคลาส pipe	65
5.17	แอตทริบิวต์และเมทธอดของคลาส matProp…	66
5.18	แอตทริบิวต์และเมทธอดของคลาส crack…	69

รูปที่		หน้า
5.19	แอตทริบิวต์และเมทธอดของคลาส load….	71
5.20	แอตทริบิวต์และเมทธอดของคลาส assessment….	72
5.21	แอตทริบิวต์และเมธอดของคลาส charCrack….	73
5.22	แอตทริบิวต์และเมทธอดของคลาส charStd….	74
5.23	แอตทริบิวต์และเมทธอดของคลาส limitLoad….	74
5.24	แอตทริบิวต์และเมทธอดของคลาส SIF….	74
5.25	แอตทริบิวต์และเมทธอดของคลาส crackGrowth….	75
5.26	แอตทริบิวต์และเมทธอดของคลาส FAD….	76
5.27	ส่วนติดต่อกับผู้ใช้งาน	77
5.28	ส่วนระเบียบวิธีการประเมิน	78
5.29	ส่วนโครงสร้าง	
	(ก) เลือกชนิดของโครงสร้าง	78
	(ข) หน้าต่างสำหรับกำหนดค่าขนาดของท่อ	78
5.30	ส่วนรับค่าสมบัติของวัสดุ	
	(ก) ปุ่มกดเพื่อจะเข้าสู่หน้าต่าง	79
	(ข) หน้าต่างสำหรับกำหนดค่าสมบัติขอวัสดุ	79
5.31	ส่วนภาระ	
	(ก) ภาระปฐมภูมิ	79
	(ข) ภาระทุติยภูมิ	80
5.32	ส่วนรอยร้าว	
	(ก) ปุ่มกดเพื่อจะเข้าสู่หน้าต่าง	80
	(ข) หน้าต่างสำหรับกำหนดค่าข้อมูลของรอยร้าวและระบุลักษณะรอยร้าว	81
5.33	ส่วนการกำหนดตำแหน่งที่จะประเมินบนรอยร้าว	82
5.34	ส่วนการเติบโตของรอยร้าวล้า	82
5.35	ส่วนการแสดงผล	83
6.1	กรณีมองด้านข้างของรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย	85
6.2	กรณีมองด้านบนของรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย	85
6.3	รอยร้าวผิวรูปครึ่งวงรีทั้งสองรอยหลังผ่านการจัดทิศทางการวางตัว	87
6.4	รอยร้าวผิวรูปครึ่งวงรีทั้งสองรอยหลังผ่านการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว	89

รูปที่		หน้า
6.5	กรณีมองด้านข้างของรอยร้าวฝังรูปวงรีทั้งสองรอย	90
6.6	กรณีมองด้านบนของรอยร้าวฝังรูปวงรีทั้งสองรอย	90
6.7	รอยร้าวฝังรูปวงรีทั้งสองรอยหลังผ่านการจัดทิศทางการวางตัว	92
6.8	รอยร้าวฝังรูปวงรีทั้งสองรอยหลังผ่านการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว	93
6.9	กรณีมองด้านข้างของรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี	95
6.10	กรณีมองด้านบนของรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี	95
6.11	รอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรีหลังผ่านการจัดทิศทางการวางตัว	97
6.12	รอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรีหลังผ่านการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว	98
6.13	ลักษณะของท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระแรงดึงตามแนวแกน	100
6.14	ลักษณะของท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระโมเมนต์ดัด	103
6.15	ลักษณะของท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายใน	106

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

คำอธิบายสัญลักษณ์และคำย่อ

สัญลักษณ์	คำอธิบาย
A, k	สัมประสิทธิ์ค่าคงที่ในสมการ (2.1)
C, m	ค่าคงตัวในสมการ (2.12) ซึ่งขึ้นกับชนิดวัสดุ สภาวะของภาระ สภาพแวดล้อม
E	มอดูลัสของความยืดหยุ่น
F_a	ตัวประกอบสำรองของขนาดรอยร้าว
F_K	ตัวประกอบสำรองของความต้านทานการแตกหัก
F_σ	ตัวประกอบสำรองของความเค้นคราก
F_L	ตัวประกอบสำรองของภาระ
K_{max}	ค่าสูงสุดของพารามิเตอร์ K
K_{min}	ค่าต่ำสุดของพารามิเตอร์ K
K	พารามิเตอร์ K
K_I	พารามิเตอร์ K ในโหมดการเสียหายโหมดที่ 1
K_{mat}	ความต้านทานการแตกหักของวัสดุ $(K_C, K_{IC}, K_{0.2}, K_g)$
K_r	อัตราส่วนพารามิเตอร์ K ต่อความต้านทานการแตกหักของวัสดุ
K_C	ความต้านทานการแตกหักของวัสดุ ณ จุดที่รอยร้าวเริ่มต้นการแตกหักแบบเปราะและปลายรอยร้าวมีสถานะความเค้นเป็นความเค้นระนาบ
K_{IC}	ความต้านทานการแตกหักของวัสดุ ณ จุดที่รอยร้าวเริ่มต้นการแตกหักแบบเปราะและปลายรอยร้าวมีสถานะความเค้นเป็นความเครียดระนาบ
$K_{0.2}$	ความต้านทานการแตกหักของวัสดุ ณ จุดที่รอยร้าวเติบอย่างมีเสถียรภาพไปเป็นระยะ 0.2 มิลลิเมตร
K_g	ความต้านทานการแตกหักของวัสดุ ณ จุดที่รอยร้าวเติบอย่างมีเสถียรภาพไปเป็นระยะ Δa_g มิลลิเมตร
K_r^P	อัตราส่วนพารามิเตอร์ K ปฐมภูมิ ต่อความต้านทานการแตกหัก K_{mat} $(K_C, K_{IC}, K_{0.2})$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

K_r^S	อัตราส่วนพารามิเตอร์ K ทุติยภูมิต่อความต้านทานการแตกหัก $K_{mat}\left(K_C, K_{IC}, K_{0.2}\right)$
K_{rg}^P	อัตราส่วนพารามิเตอร์ K ปฐมภูมิต่อความต้านทานการแตกหัก $K_{mat}\left(K_C, K_{IC}, K_{0.2}\right)$
K_{rg}^S	อัตราส่วนพารามิเตอร์ K ทุติยภูมิต่อความต้านทานการแตกหัก $K_{mat}\left(K_C, K_{IC}, K_{0.2}\right)$
K_{mat}^{assess}	ความต้านทานการแตกหักขณะประเมิน
$K_{mat}^{critical}$	ความต้านทานการแตกหักขณะโครงสร้างแตกหัก
L_r	อัตราส่วนภาระ
$L_r^{\max}$	ค่าขีดจำกัดของอัตราส่วนสูงสุดของภาระที่โครงสร้างสามารถรับได้
P	ภาระที่ทำให้เกิดความเสียหาย
P_u	ภาระที่ทำให้เกิดความเสียหายเนื่องจากการครากทั้งหน้าตัดของโครงสร้างที่มีรอยร้าว
P_f	ภาระที่ทำให้เกิดความเสียหายเนื่องจากการเติบโตอย่างไร้เสถียรภาพของรอยร้าวแบบเปราะ
P_L	ภาระขีดจำกัด
$P_{critical}$	ภาระที่ทำให้เกิดความเสียหาย
P_{assess}	ภาระขณะทำการประเมิน
S_r	อัตราส่วนภาระต่อภาระขีดจำกัด
a_0	ความยาวรอยร้าวเริ่มต้น
a_i	ความยาวรอยร้าวรอบที่ i หลังจากรับภาระ
a_{eff}	ความยาวรอยร้าวประสิทธิผล
$a_{critical}$	ขนาดของรอยร้าววิกฤติ
a_{assess}	ขนาดของรอยร้าวขณะประเมิน
r	ระยะแนวรัศมีจากปลายรอยร้าวไปยังจุดใด ๆ
r_y	ครึ่งหนึ่งของบริเวณคราก
σ^u	ความเค้นดึงสูงสุดของวัสดุ
σ^{uc}	ความเค้นดึงสูงสุดของวัสดุในโครงสร้างที่ไม่มีรอยร้าว
σ^f	ความเค้นที่ได้จากการวิเคราะห์ทางทฤษฎีกลศาสตร์การ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

	แตกหักยืดหยุ่นเชิงเส้น
σ_{ij}	องค์ประกอบความเค้น
θ	มุมที่ระยะแนวรัศมี r ทำกับระนาบรอยร้าว
f_{ij}, g_{ij}	ฟังก์ชันไร้หน่วยที่ขึ้นกับ θ
Δa_{blunt}	ความยาวรอยร้าวที่เพิ่มเนื่องจากของปลายรอยร้าวทื่อ
Δa_{stable}	ความยาวรอยร้าวที่เพิ่มจากเดิมอย่างมีเสถียรภาพ
ΔK	พิสัยของพารามิเตอร์ K
$(da)_i$	ความยาวรอยร้าวที่เพิ่มขึ้นต่อรอบ ณ รอบภาระที่ i
ΔK_i	พารามิเตอร์ ΔK ที่ความยาวรอยร้าว a_i
σ_0	normalizing stress ในสมการที่ (3.5)
ε_0	normalizing strain ในสมการที่ (3.5)
α, n	ค่าคงตัวของวัสดุในสมการที่ (3.5)
σ_y	ความเค้นคราก
σ_u	ความเค้นดึงสูงสุด
$\bar{\sigma}$	flow stress
ε_{ref}	ความเครียด ณ ความเค้นอ้างอิง
ε_y	ความเครียดคราก
υ	อัตราส่วนปัวซง
Δa_g	ระยะที่เพิ่มขึ้นของรอยร้าวเนื่องจากการเติบโตอย่างมีเสถียรภาพจากความยาวเดิม
σ^P	ความเค้นปฐมภูมิ
σ^S	ความเค้นทุติยภูมิ
σ_L	ความเค้นขีดจำกัด
ρ	ตัวประกอบปรับแก้พลาสติกซิตี้
σ_y^{assess}	ความเค้นครากขณะประเมิน
$\sigma_y^{critical}$	ความเค้นครากขณะโครงสร้างแตกหัก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของวิทยานิพนธ์

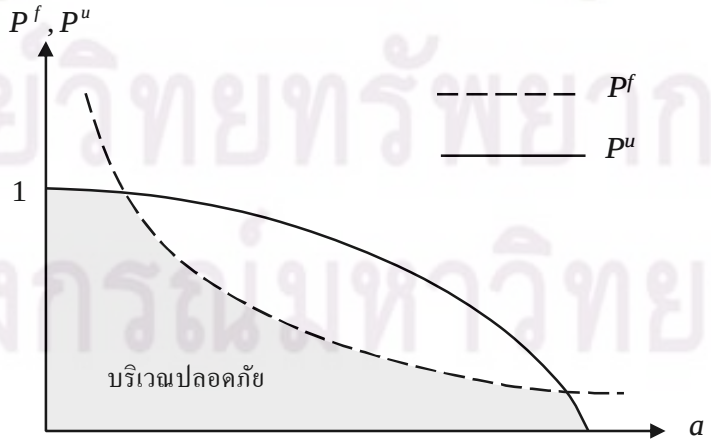
ความเสียหายของโครงสร้างทางวิศวกรรมตั้งแต่อดีตจนถึงปัจจุบันสร้างความเสียหายร้ายแรงต่อชีวิตและทรัพย์สินของมนุษย์ เช่น ท่อก๊าซธรรมชาติ LPG ในประเทศรัสเซียเกิดระเบิดขึ้น แรงระเบิดทำให้รางรถไฟที่อยู่ใกล้ขาดจากกัน เหตุการณ์ดังกล่าวทำให้มีผู้เสียชีวิตจากรถไฟตกรางและแรงระเบิดถึง 575 คน และบาดเจ็บกว่า 600 คน [1] ท่อก๊าซธรรมชาติในรัฐนิวเม็กซิโก ประเทศสหรัฐอเมริกาเกิดรั่ว หลังจากนั้นจึงเกิดการระเบิดขึ้น เปลวไฟและแรงระเบิดทำให้มีผู้เสียชีวิต 12 คน การตรวจสอบพบว่า สาเหตุของการระเบิดเริ่มจากการกัดกร่อน (corrosion) ภายในท่อ และเกิดรอยร้าวขึ้นภายใน [2] เป็นต้น นอกจากนี้ยังมีความเสียหายเนื่องจากภาชนะความดันระเบิดเกิดขึ้นทั่วโลกอีกจำนวนมาก [3]

ความปลอดภัยของท่อหรือภาชนะความดัน (pressure vessels) ขึ้นอยู่กับการออกแบบและการติดตั้งตามมาตรฐานที่ได้รับการยอมรับ เช่น มาตรฐาน ASME (American Society of Mechanical Engineering) เป็นต้น ความปลอดภัยของภาชนะความดันยังขึ้นอยู่กับการใช้งาน เช่น ใช้งานในพิกัดของภาระที่ออกแบบ มีการตรวจสอบสภาพและซ่อมบำรุงตามระยะเวลาที่กำหนด ฯลฯ อย่างไรก็ดี ในการออกแบบหรือการใช้งานภาชนะความดันจะมีปัจจัยอื่น ๆ ที่ไม่สามารถระบุได้ชัดเจน เช่น การกัดกร่อน ความเค้นตกค้าง (residual stress) และความบกพร่อง (defect) ในเนื้อวัสดุ เป็นต้น ปัจจัยเหล่านี้สามารถทำให้ท่อค่อย ๆ เสื่อมสภาพ (degradation) และปรากฏเป็นรอยร้าว (crack) ได้ในที่สุด การประเมินว่าท่อหรือภาชนะความดันที่มีรอยร้าวยังมีความปลอดภัยในการใช้งานเพียงใด และการทำนายว่าท่อหรือภาชนะความดันมีอายุใช้งานเหลืออีกเท่าใดจึงเป็นสิ่งที่สำคัญ ภารกิจดังกล่าวคือ การประเมินการคงสภาพของโครงสร้าง (structural integrity assessment) ในกรณีที่โครงสร้างสามารถใช้งานต่อไปได้ โครงสร้างจะถูกกำหนดระยะเวลาการซ่อมบำรุงโดยอาศัยแนวทางจากผลการทำนายอายุที่เหลือ

การประเมินการคงสภาพของโครงสร้างที่มีรอยร้าวเริ่มต้นในปี ค.ศ. 1975 โดย Dowling และ Townley [4] พวกเขาเสนอว่า โหมดความเสียหายของโครงสร้างที่มีรอยร้าวจะอยู่ภายในขอบเขตของโหมดความเสียหาย 2 ชนิด คือ ความเสียหายเนื่องจากการเติบโตอย่างไร้เสถียรภาพของรอยร้าว (unstable crack growth) และความเสียหายเนื่องจากการครากทั้งหน้าตัด (plastic collapse) แนวคิด

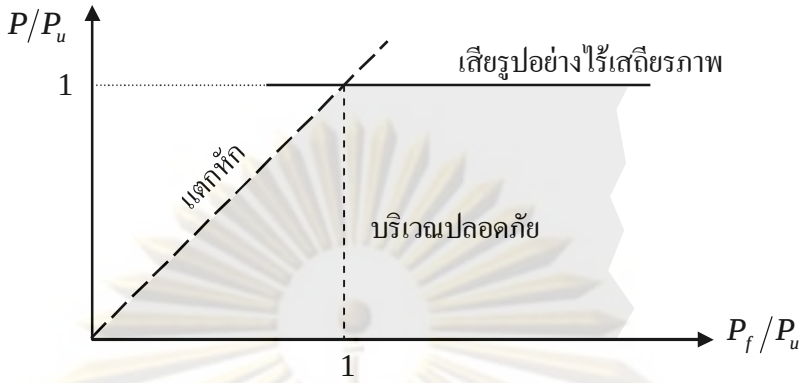
นี้นำไปสู่การพัฒนาแผนภาพประเมินความเสียหาย (failure assessment diagram, FAD) ในรูปที่ 1.1 (ก) แผนภาพในรูปนี้มีแกนตั้ง คือ ตัวแปรไร้หน่วยสำหรับบ่งชี้ความเสียหายแต่ละชนิด และแกนนอนคือ ขนาดรอยร้าว สำหรับความเสียหายเนื่องจากการครากทั้งหน้าตัดจะพิจารณาจากตัวแปร $P_u = \sigma^u / \sigma^{uc}$ โดย σ^u คือ ความเค้นสูงสุดที่โครงสร้างซึ่งมีรอยร้าวสามารถรับได้ และ σ^{uc} คือ ความเค้นสูงสุดที่โครงสร้างซึ่งไม่มีรอยร้าวสามารถรับได้ สำหรับความเสียหายเนื่องจากการเติบโตอย่างไร้เสถียรภาพของรอยร้าวจะพิจารณาจากตัวแปร $P_f = \sigma^f / \sigma^{uc}$ โดย σ^f คือ ความเค้นที่ทำให้รอยร้าวเติบโตอย่างไร้เสถียรภาพ ความเค้นนี้ได้จากการวิเคราะห์ด้วยทฤษฎีกลศาสตร์การแตกหักยืดหยุ่นเชิงเส้น (linear elastic fracture mechanic, LEFM) และ σ^{uc} คือ ความเค้นสูงสุดที่โครงสร้างที่ไม่มีรอยร้าวสามารถรับได้ เส้นประเมินความเสียหายของโครงสร้าง (failure assessment line) เนื่องจากการครากทั้งหน้าตัด และเนื่องจากการเติบโตอย่างไร้เสถียรภาพของรอยร้าว ที่ความยาวรอยร้าวต่าง ๆ แสดงด้วยเส้นทึบและเส้นประ ตามลำดับ จากรูป ความเสียหายของโครงสร้างมีความยาวรอยร้าวต่าง ๆ จะเกิดขึ้นเมื่อภาระที่กระทำกับโครงสร้างมีขนาดเท่ากับค่าต่ำสุดระหว่างภาระที่ทำให้เกิดความเสียหายเนื่องจากการเติบโตอย่างไร้เสถียรภาพของรอยร้าว และภาระที่ทำให้เกิดความเสียหายเนื่องจากการครากทั้งหน้าตัด ซึ่งก็คือบริเวณที่แรเงาในรูป 1.1

เนื่องจากลักษณะและแนวโน้มของเส้นประเมินความเสียหายทั้ง 2 เส้นในรูปที่ 1.1 จะแตกต่างกันขึ้นกับ ชนิดและขนาดของโครงสร้าง ลักษณะรอยร้าว และสมบัติของวัสดุ เช่น ความต้านทานการแตกหัก (fracture toughness) และความเค้นดึงสูงสุด (ultimate strength) เป็นต้น Dowling et al. [4] จึงเสนอตัวแปรไร้หน่วยตัวใหม่ ซึ่งนำไปสู่แผนภาพการประเมินความเสียหายอีกลักษณะหนึ่งดังรูปที่ 1.2 จากรูปแกนตั้งคือ P/P_u โดย P คือ ภาระที่ทำให้เสียหาย และ P_u คือ ภาระที่ทำให้โครงสร้างที่มีรอยร้าวเกิดการครากทั้งหน้าตัด ส่วนแกนนอนใช้ตัวแปร P_f / P_u โดย



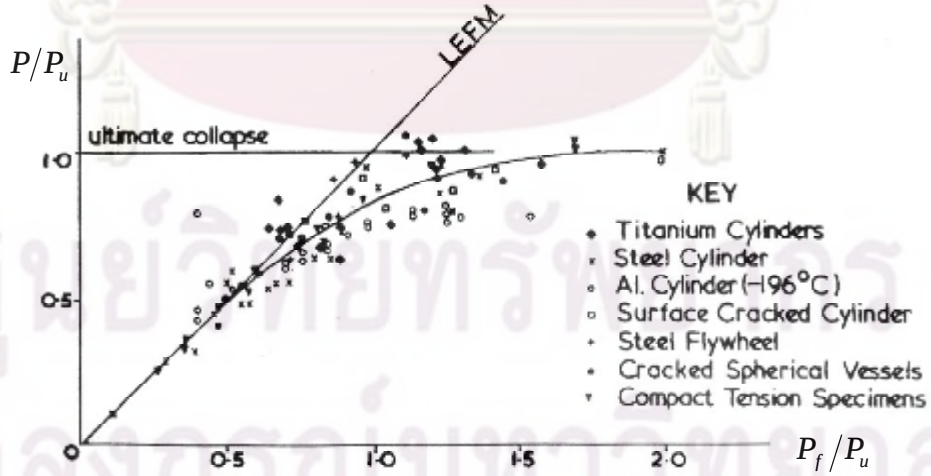
รูปที่ 1.1 แผนภาพประเมินความเสียหายของ Dowling และ Townley

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



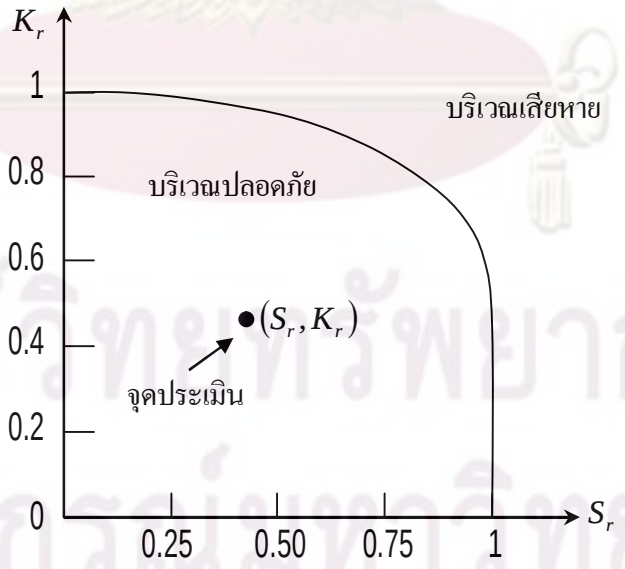
รูปที่ 1.2 แผนภาพประเมินความเสียหายของ Dowling และ Townley ที่ปรับปรุงแล้ว

P_f คือ ภาระที่ทำให้โครงสร้างเกิดความเสียหายเนื่องจากการเติบโตอย่างไร้เสถียรภาพของรอยร้าว บริเวณที่ปลอดภัยของโครงสร้างที่ประเมินคือ บริเวณที่อยู่ใต้เส้นประเมินความเสียหายเส้นที่อยู่ต่ำกว่า หรือบริเวณแรเงาในรูป เส้นตรงขีดจำกัดทั้งสองเส้นในรูปที่ 1.2 สามารถแทนด้วยเส้นโค้งสัมผัสเพียงเส้นเดียวดังรูปที่ 1.3 [4] รูปนี้ยังแสดงผลการทดสอบจนกระทั่งเสียหายของโครงสร้างชนิดต่าง ๆ วัสดุชนิดต่าง ๆ และรับภาระชนิดต่าง ๆ จากรูปจะเห็นว่าจุดประเมิน ณ สภาวะที่โครงสร้างเสียหายส่วนใหญ่อยู่บนเส้นโค้งสัมผัส ซึ่งหมายความว่าเส้นโค้งสามารถทำนายสถานะวิกฤติได้แม่นยำ และเส้นโค้งนี้สามารถประยุกต์ใช้ได้โดยไม่ขึ้นกับชนิดโครงสร้าง ชนิดวัสดุ และชนิดของภาระ



รูปที่ 1.3 การทวนสอบแผนภาพประเมินความเสียหายของ Dowling และ Townley ที่ปรับปรุงแล้ว

Harrison et al. [4] ปรับปรุงแผนภาพการประเมินของ Dowling et al. ให้เข้าใจง่ายขึ้น โดยเปลี่ยนตัวแปร P_f/P_u เป็นอัตราส่วนของตัวประกอบความเข้มของความเค้น (stress intensity factor ratio) หรือเรียกอีกชื่อหนึ่งว่าอัตราส่วนพารามิเตอร์ K และแทนด้วยสัญลักษณ์ K_r และเปลี่ยนตัวแปร P/P_u เป็นอัตราส่วนภาระ (หรืออัตราส่วนความเค้น) S_r แผนภาพประเมินความเสียหายที่ได้จะมีลักษณะดังแสดงในรูปที่ 1.4 ตัวแปร K_r กำหนดให้เท่ากับอัตราส่วนระหว่างพารามิเตอร์ K กับความต้านทานการแตกหักในสภาวะความเครียดระนาบของวัสดุ (plane strain fracture toughness) K_{IC} ส่วนตัวแปร S_r กำหนดให้เท่ากับ อัตราส่วนระหว่างภาระ (หรือความเค้น) ที่กระทำกับโครงสร้างกับภาระคราก (หรือความเค้นคราก) ความสัมพันธ์ของตัวแปรทั้งสองคือเส้นประเมินความเสียหาย บริเวณที่แสดงว่าโครงสร้างยังอยู่ในสถานะที่ปลอดภัยคือ บริเวณใต้เส้นประเมินความเสียหาย สถานะของโครงสร้างที่สภาวะใช้งานจะแสดงด้วยของพิกัด (S_r, K_r) หรือเรียกว่า “จุดประเมิน” ดังนั้นถ้าพล็อตจุดประเมินบนแผนภาพประเมินความเสียหาย ก็จะทราบว่าโครงสร้างอยู่ในสถานะใด หากจุดประเมินอยู่ใต้เส้นประเมินความเสียหายจะถือว่าโครงสร้างปลอดภัย แต่ถ้าอยู่บนเส้นหรืออยู่เหนือเส้นประเมินแล้วจะถือว่าโครงสร้างไม่ปลอดภัย ต่อมาแนวคิดของ Harrison et al. ถูกปรับปรุงให้ครอบคลุมกรณีที่โครงสร้างรับภาระเนื่องจากความร้อน (thermal load) และกรณีโครงสร้างทำด้วยวัสดุซึ่งพฤติกรรมการเสียรูปมี strain hardening เช่น เหล็กกล้าความแข็งแรงปานกลาง (medium strength steel) เป็นต้น

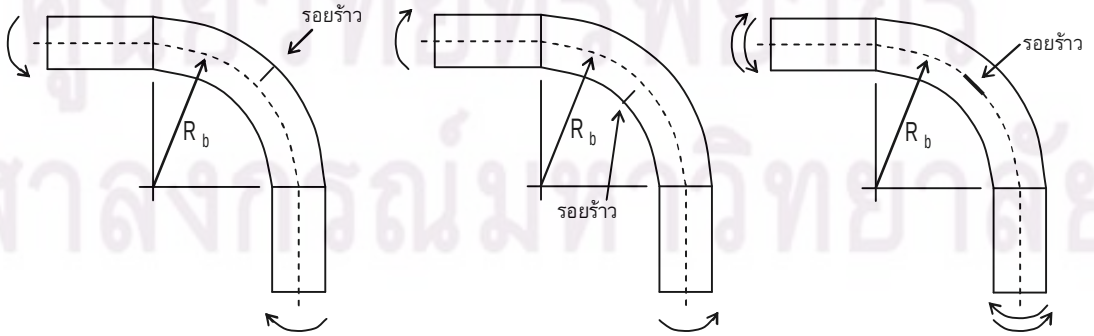


รูปที่ 1.4 แผนภาพประเมินความเสียหายของ Harrison et al.

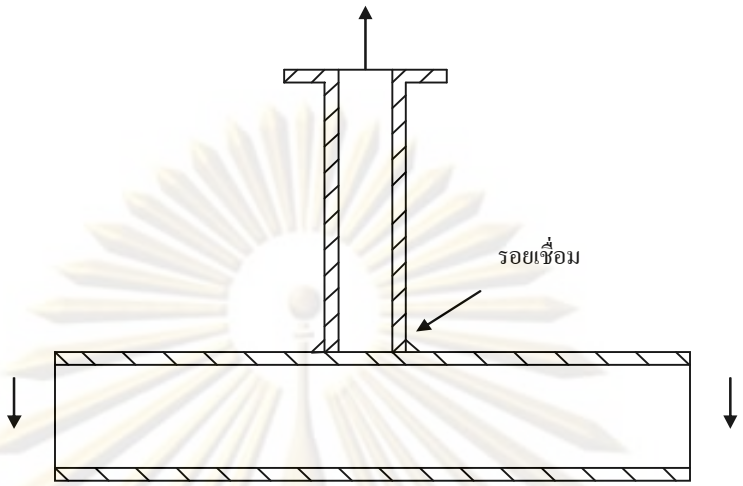
ในปี ค.ศ. 1976 Central Electricity Generating Board, CEGB เสนอระเบียบวิธีสำหรับประเมินการคงสภาพของโครงสร้างที่ตรวจพบรอยร้าว ซึ่งทำงานในสภาวะที่ไม่มีผลของการคืบ ระเบียบวิธีดังกล่าวมีชื่อเรียกว่า “ระเบียบวิธี R6” [4] ระเบียบวิธี R6 ได้รับการพัฒนาอย่างต่อเนื่อง ในปี ค.ศ. 1980 มีการตีพิมพ์ระเบียบวิธี R6 ฉบับแก้ไขครั้งที่สอง ในปี 1988 มีการตีพิมพ์ระเบียบวิธี R6 ฉบับแก้ไขครั้งที่ 3 [5] และในปี ค.ศ. 2001 มีการตีพิมพ์ระเบียบวิธี R6 ฉบับแก้ไขครั้งที่ 4 [6] ระเบียบวิธีนี้ยังเป็นต้นแบบการพัฒนาระเบียบวิธีอื่น ๆ เช่น ระเบียบวิธี API 579 [7] ระเบียบวิธี SINTAP [8] เป็นต้น

ในโรงงานปิโตรเคมีหรือโรงไฟฟ้า จะมีท่อสำหรับขนส่งของเหลวหรือก๊าซเป็นจำนวนมาก หากท่อเหล่านี้เสียหาย ของเหลวหรือก๊าซในท่อที่รั่วออกมามีโอกาสสร้างความสูญเสียทั้งชีวิตและทรัพย์สิน จากการสืบค้นพบว่ามีการนำระเบียบวิธี R6 มาประยุกต์กับท่อ ได้แก่ การทำนายการแตกหักของข้องอ (elbows) ที่มีรอยร้าวทะลุความหนาตามแนวรัศมีและตามแนวแกน ภายใต้โมเมนต์ดัดบนระนาบ (in-plane bending moment) ในรูปที่ 1.5 [9] การทำนายพฤติกรรมการแตกหักบริเวณรอยเชื่อมของท่อรูปตัวที (welded tubular T-joints) [10] ดังรูปที่ 1.3 ทั้งสองกรณีระเบียบวิธี R6 สามารถทำนายภาระที่ทำให้ท่อเสียหายได้ใกล้เคียงกับผลการทดลอง

ประเมินการคงสภาพโครงสร้างด้วยวิธี R6 (ฉบับแก้ไขครั้งที่ 3) มีขั้นตอนทั้งหมด 13 ขั้นตอน แต่ละขั้นตอนก็มีรายละเอียดและเงื่อนไขมาก (บทที่ 3 หัวข้อ 3.1) นอกจากนี้ผู้ประเมินจำเป็นต้องมีความรู้เกี่ยวกับกลศาสตร์การแตกหัก ดังนั้นจึงมีการพัฒนาโปรแกรมคอมพิวเตอร์สำเร็จรูป ได้แก่ R-Code Software [11] , SACC (safety assessment of component with crack) [12] และโปรแกรมที่พัฒนาโดย Kobayashi et al. [13] โปรแกรมเหล่านี้ช่วยให้ผู้ประเมินสามารถประเมินการคงสภาพของโครงสร้างได้โดยไม่จำเป็นต้องมีความรู้ทางทฤษฎีหรือทราบรายละเอียดของระเบียบวิธี R6 อย่างลึกซึ้ง



รูปที่ 1.5 ข้องอมีรอยร้าวทะลุความหนาตามแนวรัศมีและตามแนวแกน



รูปที่ 1.6 ท่อรูปตัวที

อย่างไรก็ดี โปรแกรมที่กล่าวไปนั้น ยังมีข้อจำกัดคือ 1) มีเส้นประเมินความเสียหายเพียงแบบที่ 1 จึงมีแนวโน้มที่จะให้ผลการประเมินออกมาปลอดภัยเกินไป (conservative) กล่าวอีกอย่างหนึ่งคือทำนายว่าโครงสร้างเสียหายทั้ง ๆ ที่ยังสามารถใช้งานต่อไปได้ และ 2) ไม่มีส่วนของโปรแกรมสำหรับทำหน้าที่ระบุลักษณะรอยร้าว (crack characterization) ดังนั้นเพื่อให้การประเมินแม่นยำขึ้น โปรแกรมควรทำการประเมินระดับที่ 2 และระดับที่ 3 ได้ นอกจากนี้ถ้าโปรแกรมสามารถระบุลักษณะของรอยร้าวได้จะทำให้การประเมินในกรณีที่โครงสร้างมีรอยร้าวหลายรอยอยู่ใกล้กันทำได้สะดวกขึ้น

วิทยานิพนธ์นี้พัฒนาโปรแกรมประเมินการคงสภาพของท่อที่ตรวจพบรอยร้าวด้วยระเบียบวิธี R6 โดยโปรแกรมสามารถทำการประเมินระดับที่ 1 และระดับที่ 2 ได้ การพัฒนาโปรแกรมจะใช้แนวคิดเชิงวัตถุ เพราะแนวคิดนี้ช่วยให้การปรับปรุงโปรแกรมในอนาคตทำได้ง่ายกว่าการเขียนโปรแกรมแบบโครงสร้าง (structured programming) [14] นอกจากนี้โปรแกรมที่พัฒนาจะเพิ่มเติมส่วนที่ทำหน้าที่ระบุลักษณะรอยร้าว ซึ่งจำเป็นสำหรับกรณีที่ท่อมีรอยร้าวหลายรอย การระบุลักษณะรอยร้าวจะอ้างอิงกับมาตรฐาน API 579 เพราะมีรายละเอียดครบถ้วนกว่าระเบียบวิธี R6

1.2 วัตถุประสงค์ของวิทยานิพนธ์

การออกแบบโปรแกรมเชิงวัตถุสำหรับการประเมินการคงสภาพของท่อที่มีรอยร้าวด้วยระเบียบวิธี R6

1.3 ขอบเขตของวิทยานิพนธ์

โปรแกรมที่พัฒนามีขอบเขตการใช้งานดังนี้

1. สามารถประเมินการคงสภาพของท่อตรงที่มีรอยร้าว 2 ชนิด คือ partially-circumferential semi-elliptical crack และ axial semi-elliptical crack
2. สามารถประเมินการคงสภาพของท่อตรงที่มีรอยร้าวชนิดแรกโดยชนิดของภาระที่กระทำกับท่อ คือ internal pressure และ axial load, bending moment สำหรับรอยร้าวชนิดที่สองภาระที่กระทำกับท่อ คือ internal pressure และ arbitrary stress distribution over crack face
3. สามารถประเมินโครงสร้างด้วยเส้นประเมินความเสียหายแบบที่ 1 และแบบที่ 2 ได้
4. สามารถระบุลักษณะรอยร้าวได้

1.4 วิธีการดำเนินงาน

1. ศึกษาทฤษฎีที่จำเป็นดังนี้
   - ทฤษฎีกลศาสตร์การแตกหัก
   - ระเบียบวิธี R6
   - การออกแบบโปรแกรมด้วยแนวคิดเชิงวัตถุ และภาษาคอมพิวเตอร์ที่รองรับแนวคิด
2. เขียนโปรแกรมประเมินการคงสภาพของท่อที่มีรอยร้าวด้วยระเบียบวิธี R6 ด้วยแนวคิดเชิงวัตถุ
3. ตรวจสอบความถูกต้องของโปรแกรม
   - ตรวจสอบความถูกต้องของผลการระบุลักษณะรอยร้าว
   - ตรวจสอบความถูกต้องของผลการประเมินการคงสภาพ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ลดความยุ่งยากในการประเมินการคงสภาพของท่อที่มีรอยร้าว
2. ประหยัดค่าใช้จ่ายในการซื้อโปรแกรมสำเร็จรูป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

กลศาสตร์การแตกหักยืดหยุ่นเชิงเส้น

กลศาสตร์การแตกหัก (fracture mechanics) ศึกษาผลของรอยร้าวต่อความแข็งแรงของชิ้นส่วน ชิ้นส่วนที่มีรอยร้าวเมื่อรับภาระ วัสดุบริเวณใกล้กับปลายรอยร้าวจะมีแนวโน้มเกิดความเสียหายมากกว่าบริเวณที่อยู่ห่างออกไปเพราะเสียรูปมากกว่า กลศาสตร์การแตกหักแบ่งออกเป็น 3 แขนง ตามพฤติกรรมการเสียรูปบริเวณปลายรอยร้าวของวัสดุ ได้ดังนี้ 1) กลศาสตร์การแตกหักยืดหยุ่นเชิงเส้น สำหรับการเสียรูปแบบยืดหยุ่นเชิงเส้น LEFM 2) กลศาสตร์การแตกหักอิลาสติก-พลาสติก (elastic-plastic fracture mechanic, EPFM) สำหรับการเสียรูปแบบอิลาสติก-พลาสติก และ 3) กลศาสตร์การแตกหักที่ขึ้นกับเวลา (time-dependent fracture mechanic, TDFM) สำหรับกรณีที่พฤติกรรมการเสียรูปขึ้นกับอัตราเร็วของการเสียรูป (rate of deformation) เช่น การเสียรูปคืบ (creep deformation) เป็นต้น อย่างไรก็ดี ทฤษฎีที่จำเป็นต้องใช้ในวิทยานิพนธ์นี้จะอยู่ในกรอบของ LEFM เท่านั้น

ในบทนี้จะกล่าวถึงโหมดการเสียรูปที่ปลายรอยร้าว พารามิเตอร์ปลายรอยร้าว พฤติกรรมการแตกหัก เกณฑ์การแตกหัก การเติบโตของรอยร้าวล้า และการเติบโตของรอยร้าวเนื่องจากสิ่งแวดล้อม

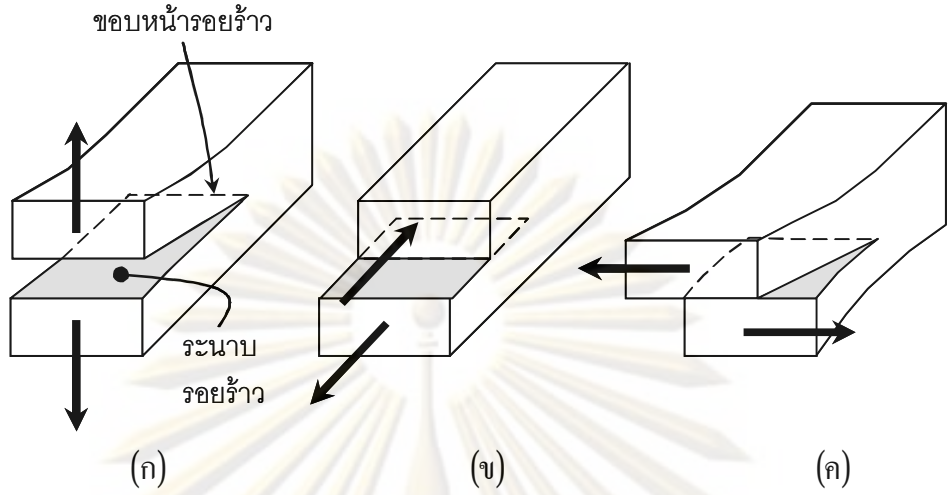
2.1 โหมดของการเสียรูปที่ปลายรอยร้าว

โหมดของการเสียรูปที่ปลายรอยร้าวกำหนดจากทิศทางการเคลื่อนตัวของระนาบรอยร้าว (crack surface) เทียบกับขอบหน้ารอยร้าว (crack front) โดยใช้นิยามนี้สามารถจำแนกโหมดการเสียรูปออกเป็น 3 โหมด คือ

1) โหมดเปิด (opening mode) หรือโหมดที่ 1 ในโหมดนี้ผิวรอยร้าวจะเคลื่อนตัวในทิศตั้งฉากกับระนาบรอยร้าวดังแสดงในรูปที่ 2.1(ก)

2) โหมดเฉือนบนระนาบ (in-plane shear mode) หรือโหมดไถล (sliding mode) หรือโหมดที่ 2 ในโหมดนี้ผิวรอยร้าวจะเคลื่อนตัวในทิศทางตั้งฉากกับขอบหน้ารอยร้าวดังแสดงในรูปที่ 2.1(ข)

3) โหมดเฉือนนอกระนาบ (out-of-plane shear mode) หรือโหมดฉีก (tearing mode) หรือโหมดที่ 3 ในโหมดนี้ผิวรอยร้าวจะเคลื่อนตัวขนานกับขอบหน้ารอยร้าวดังแสดงในรูปที่ 2.1(ค)



รูปที่ 2.1 โหมดการเสียรูปที่ปลายรอยร้าว

2.2 พารามิเตอร์ปลายรอยร้าว

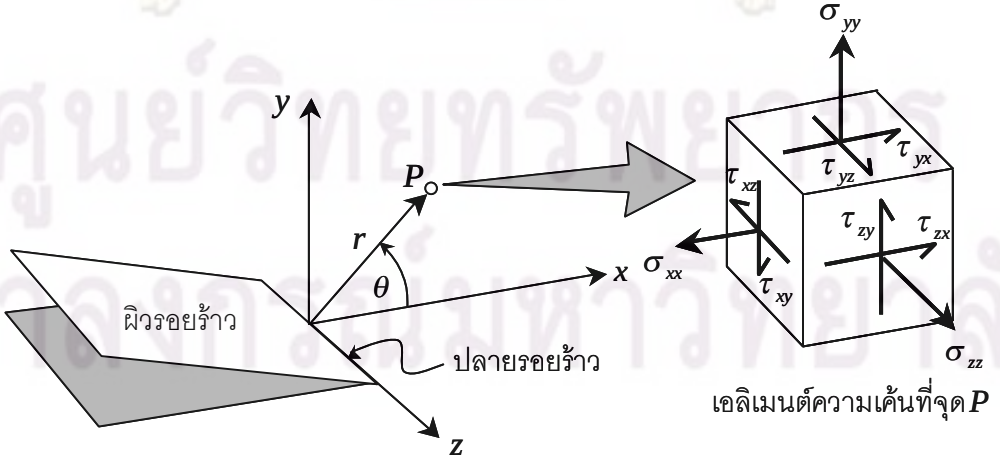
การประยุกต์ทฤษฎียืดหยุ่น (elasticity theory) เพื่อหาผลเฉลยองค์ประกอบความเค้น σ_{ij} ในวัตถุที่มีรอยร้าว (รูปที่ 2.2) พบว่าการกระจายความเค้นสามารถเขียนในรูปทั่วไปได้ดังนี้ [7]

$$\sigma_{ij}(r,\theta)=\left(\frac{k}{\sqrt{r}}\right)f_{ij}(\theta)+A_0 g_{ij,0}(\theta)+\sum_{m=1}^{\infty}A_m r^{\frac{m}{2}} g_{ij,m}(\theta) \qquad (2.1)$$

โดย r และ θ คือ ระยะจากปลายรอยร้าวไปยังจุดใด ๆ และมุมที่ทำกับระนาบรอยร้าว

k และ A คือ สัมประสิทธิ์

f_{ij} และ $g_{ij,0}$ และ $g_{ij,m}$ คือ ฟังก์ชันไร้หน่วยที่ขึ้นกับ θ



รูปที่ 2.2 องค์ประกอบความเค้นบริเวณปลายรอยร้าว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จากสมการที่ (2.1) จะเห็นว่าเทอมด้านขวามือแบ่งออกเป็น 3 ส่วน ส่วนแรกคูณกับ $r^{-1/2}$ ส่วนที่สองคูณกับ r^0 (ไม่ขึ้นกับระยะทางจากปลายรอยร้าว) และส่วนที่สามเป็นผลบวกของเทอมคูณกับ r ซึ่งยกกำลังจำนวนจริงบวก

เทอมที่คูณกับ $r^{-1/2}$ จะเป็นเทอมเด่น (dominant term) เมื่อ r มีค่าเข้าใกล้ศูนย์หรือจุด P เข้าใกล้ปลายรอยร้าว แต่เมื่อ r มีค่าเพิ่มขึ้นหรือจุด P อยู่ห่างจากปลายรอยร้าวมากขึ้นแล้ว เทอมส่วนที่สามจะกลายเป็นเทอมเด่น เพราะเทอมที่ติด $r^{-1/2}$ จะมีค่าเข้าใกล้ศูนย์ อย่างไรก็ตามในปัญหารอยร้าวจะให้ความสำคัญกับการกระจายความเค้นบริเวณใกล้ ๆ กับปลายรอยร้าวเพราะการแตกหักเกิดในบริเวณนี้ ดังนั้นองค์ประกอบความเค้นที่สำคัญ คือ

$$\sigma_{ij}(r,\theta)=\left(\frac{k}{\sqrt{r}}\right)f_{ij}(\theta)+A_0 g_{ij,0}(\theta) \tag{2.2}$$

เทอมที่ไม่ขึ้นกับ r ในสมการที่ (2.2) มีชื่อเรียกว่า "ความเค้น-ที (T-stress)" [7] แม้ว่าเทอมนี้ จะมีผลต่อสถานะความเค้นและการคราก ของวัสดุบริเวณปลายรอยร้าว แต่การวิเคราะห์การแตกหักโดยใช้เพียงเทอมแรกเทอมเดียวนั้นก็ถือว่ายอมรับได้ [7] ดังนั้นสนามความเค้นบริเวณปลายรอยร้าวจึงลดรูปเหลือ

$$\sigma_{ij}=\left(\frac{k}{\sqrt{r}}\right)f_{ij}(\theta) \tag{2.3}$$

สมการที่ (2.3) แสดงให้เห็นว่าการระบุระดับความรุนแรงที่ปลายรอยร้าวในเทอมของความเค้นนั้นไม่สามารถทำได้ เพราะความเค้นที่ปลายรอยร้าวมีค่าเข้าสู่อนันต์ ด้วยเหตุนี้จึงมีการเสนอพารา-มิเตอร์ตัวใหม่เรียกว่า ตัวประกอบความเข้มของความเค้น (stress intensity factor, K) หรือเรียกอีกชื่อหนึ่งว่า พารามิเตอร์ K นิยามของพารามิเตอร์นี้ในแต่ละโหมดคือ

$$K_I=\lim_{r\to 0}\left(\sqrt{2\pi r}\,\sigma_{yy}(r,0^o)\right) \tag{2.4ก}$$

$$K_{II}=\lim_{r\to 0}\left(\sqrt{2\pi r}\,\tau_{xy}(r,0^o)\right) \tag{2.4ข}$$

$$K_{III}=\lim_{r\to 0}\left(\sqrt{2\pi r}\,\tau_{yz}(r,0^o)\right) \tag{2.4ค}$$

โดย ตัวห้อย I, II, และ III แสดงโหมดการเสียรูปที่ปลายรอยร้าวโหมดที่ 1, 2 และ 3 ตามลำดับ

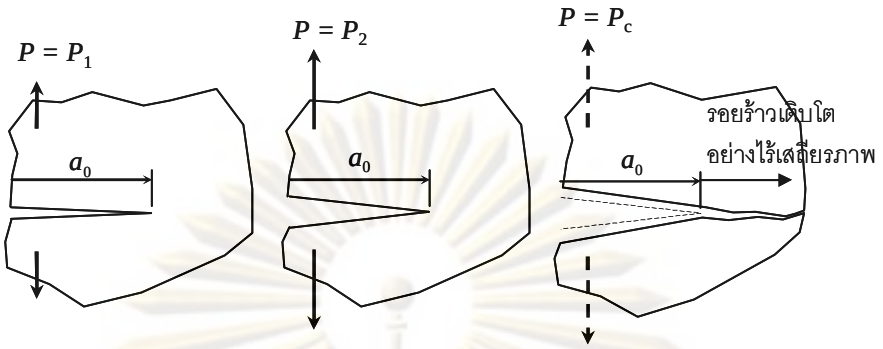
นอกจากพารามิเตอร์ K จะขึ้นอยู่กับโหมดการเสียรูปที่ปลายรอยร้าวแล้ว พารามิเตอร์นี้ยังขึ้นกับ ขนาด รูปร่าง การวางตัวของรอยร้าว ชนิดของภาระที่กระทำ และชนิดของโครงสร้าง (ท่อ เพลา แผ่นแบน ฯลฯ) การหาผลเฉลย K ของกรณีต่าง ๆ เหล่านี้สามารถทำได้ทั้งระเบียบวิธีเชิงวิเคราะห์ ระเบียบวิธีเชิงตัวเลข และวิธีทดลอง [7]

2.3 พฤติกรรมการแตกหัก

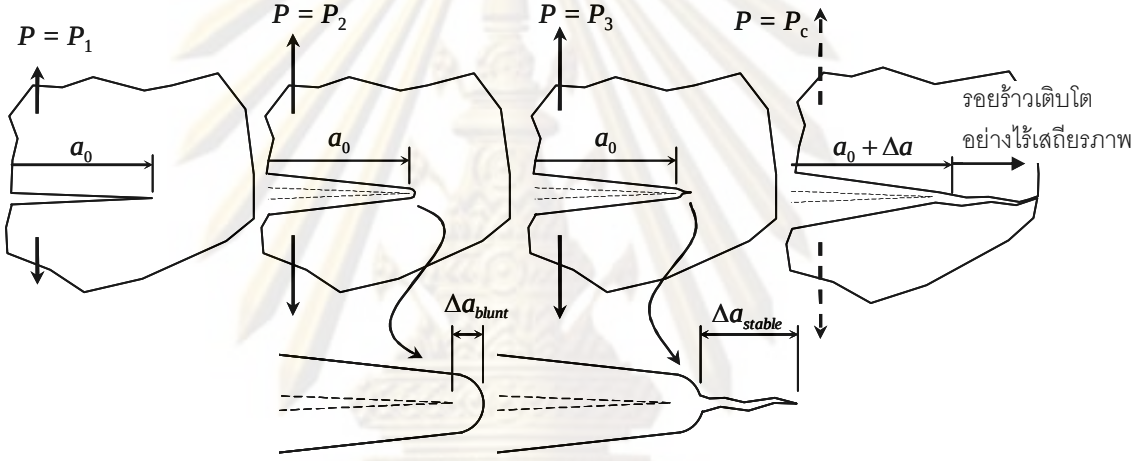
การแตกหักของโครงสร้างที่มีรอยร้าวคือ สภาวะที่รอยร้าวเติบโตด้วยอัตราเร็วสูงมาก หรือเรียกอีกอย่างหนึ่งว่า สภาวะที่รอยร้าวเติบโตอย่างไร้เสถียรภาพ (unstable fracture) การเกิดสภาวะดังกล่าวนี้เป็นไปได้ 2 รูปแบบ คือ 1) รอยร้าวเติบโตอย่างไร้เสถียรภาพจากความยาวเริ่มต้นทันทีที่ขนาดภาระเพิ่มขึ้นถึงค่าวิกฤติ หรือ 2) รอยร้าวเติบโตอย่างมีเสถียรภาพ (stable crack growth) จากความยาวเดิมเป็นระยะทางหนึ่ง ก่อนที่จะเติบโตอย่างไร้เสถียรภาพจากความยาวใหม่ พฤติกรรมการแตกหักแบบแรกเรียกว่า การแตกหักเปราะ ส่วนแบบหลังเรียกว่า การแตกหักเหนียว

กระบวนการแตกหักเปราะแสดงอยู่ในรูปที่ 2.3(ก) จากรูปเมื่อขนาดของภาระเพิ่มขึ้นจาก P_1 เป็น P_2 ปากรอยร้าว (crack mouth) จะเปิดกว้างขึ้นแต่ปลายรอยร้าวยังคงแหลมเหมือนเดิม เมื่อขนาดของภาระถึงค่าวิกฤติ P_c รอยร้าวจะเติบโตจากความยาวเดิม a_0 อย่างไร้เสถียรภาพและทำให้โครงสร้างเสียหาย เกณฑ์การแตกหักที่เหมาะสมกับพฤติกรรมการแตกหักนี้เรียกว่า เกณฑ์การกำเนิดรอยร้าว (crack initiation criteria)

กระบวนการแตกหักเหนียวแสดงอยู่ในรูปที่ 2.3(ข) จากรูปเมื่อขนาดภาระเท่ากับ P_2 จะเกิดบริเวณคราก (yield zone) ที่บริเวณปลายรอยร้าว ซึ่งทำให้ปลายรอยร้าวทื่อและเพิ่มความยาวรอยร้าวจากเดิมเป็นระยะ Δa_{blunt} เมื่อขนาดภาระเพิ่มขึ้นเป็น P_3 ความเครียดบริเวณปลายรอยร้าว (ที่อ) จะเพิ่มถึงค่าวิกฤติ ทำให้กำเนิดรอยร้าวใหม่ที่เติบโตต่อจากเดิมเป็นระยะ Δa_{stable} การเติบโตของรอยร้าวระหว่างนี้ถือว่าเป็นการเติบโตอย่างมีเสถียรภาพ หลังจากนั้นรอยร้าวจะหยุดการเติบโตถ้าหากภาระยังคงอยู่ที่ P_3 ถ้าต้องการให้รอยร้าวเติบโต (อย่างมีเสถียรภาพ) ต่อไปอีกก็จะต้องเพิ่มขนาดภาระอีก ในความเป็นจริงกระบวนการเติบโตอย่างมีเสถียรภาพจะเกิดขึ้นพร้อม ๆ กับการเพิ่มขนาดภาระ อย่างไรก็ดี ในท้ายที่สุดแล้ว เมื่อภาระมีขนาดเท่ากับ P_c รอยร้าวก็จะเติบโตอย่างไร้เสถียรภาพ เกณฑ์การแตกหักที่เหมาะสมกับพฤติกรรมการแตกหักนี้เรียกว่า เกณฑ์การเติบโตอย่างไร้เสถียร-ภาพ (unstable crack growth criteria)



(ก) การแตกหักเปราะ



(ข) การแตกหักเหนียว

รูปที่ 2.3 พฤติกรรมการแตกหักที่ปลายรอยร้าว

2.4 เกณฑ์การแตกหัก

พารามิเตอร์ K บอกถึงสถานะความรุนแรงที่เกิดกับวัสดุบริเวณปลายรอยร้าวได้ แต่บอกไม่ได้ว่าสถานะการรับภาระขณะนั้นใกล้จะเกิดการแตกหักมากเพียงใด การจะทราบระดับความวิกฤติของโครงสร้างจำเป็นต้องทราบสมบัติของวัสดุที่เรียกว่า ความต้านทานการแตกหัก K_C เพื่อใช้เปรียบเทียบกับค่าของพารามิเตอร์ K

ความต้านทานการแตกหักบอกถึง ความสามารถของวัสดุในการต้านทานการเติบโตของรอยร้าว (จากที่มีอยู่เดิม) ความต้านทานการแตกหักจะแตกต่างกันไปขึ้นกับชนิดของวัสดุ และปัจจัยอื่น ๆ เช่น สถานะความเค้น สภาพแวดล้อม อุณหภูมิ กรรมวิธีทางความร้อน ฯลฯ ดังนั้นการ

ทดสอบเพื่อหาความต้านทานการแตกหักจะต้องใช้ชิ้นงานที่มีสภาพคล้ายคลึงกับสภาพวัสดุที่ทำโครงสร้าง และจำลองสภาวะทดสอบให้สอดคล้องกับสภาวะใช้งานของโครงสร้าง

สำหรับเกณฑ์การกำเนิดรอยร้าวจะถือว่าการแตกหักเกิดขึ้นเมื่อ พารามิเตอร์ปลายรอยร้าว ณ ความยาวรอยร้าวเริ่มต้น a_0 มีค่าเท่ากับความต้านทานการแตกหักของวัสดุ ในกรณีนี้ความต้านทานการแตกหักจะเป็นตัวเลขค่าเดียว (สำหรับชนิดวัสดุ อุณหภูมิ สถานะความเค้น ฯลฯ ที่กำหนด) สำหรับการเสียรูปโหมดที่ 1 เกณฑ์การกำเนิดรอยร้าวจะเขียนได้ในรูปต่อไปนี้

$$K_I(a_0) = K_c \tag{2.5}$$

โดย a_0 คือ ความยาวรอยร้าวเริ่มต้น

K_c คือ ความต้านทานการแตกหักของวัสดุ (จากการทดสอบภายใต้สภาวะที่คล้ายคลึงกับสภาวะใช้งาน)

ในกรณีที่ปลายรอยร้าวมีบริเวณครากขนาดเล็ก จะทำให้ขัดแย้งกับเงื่อนไขทางทฤษฎีของ LEFM เพราะการนิยามของพารามิเตอร์ K มาจากการวิเคราะห์ยืดหยุ่นเชิงเส้น อย่างไรก็ดี Irwin [7] เสนอวิธีการปรับแก้เพื่อทำให้พารามิเตอร์ K สามารถใช้งานในขอบเขตดังกล่าวได้ Irwin เสนอให้คำนวณค่าพารามิเตอร์ K โดยใช้ความยาวรอยร้าวประสิทธิผล (effective crack length) a_{eff} ซึ่งเท่ากับผลรวมของความยาวรอยร้าวขณะนั้นกับครึ่งหนึ่งของขนาดเสียรูปพลาสติก ดังนั้นในกรณีนี้เกณฑ์การกำเนิดรอยร้าวจะเขียนได้เป็น

$$K_I(a_{eff}) = K_c \tag{2.6}$$

และ

$$a_{eff} = a + r_y \tag{2.7}$$

โดย r_y คือ ครึ่งหนึ่งของบริเวณคราก ซึ่งคำนวณได้จาก [7]

$$r_y = \frac{1}{2\pi}\left(\frac{K_I}{\sigma_y}\right)^2 \tag{2.8ก}$$

กรณีสถานะความเค้นแบบความเค้นระนาบ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$r_y = \frac{1}{6\pi}\left(\frac{K_I}{\sigma_y}\right)^2 \qquad (2.8ข)$$

กรณีสถานะความเค้นแบบความเครียดระนาบ

สำหรับเกณฑ์การเติบโตอย่างไร้เสถียรภาพ จะถือว่าการแตกหักเกิดขึ้นเมื่อ 1) พารามิเตอร์ปลายรอยร้าว ณ ความยาวรอยร้าวขณะนั้น a (ซึ่งเท่ากับ $a_0+\Delta a$) มีค่าเท่ากับความต้านทานการแตกหักของวัสดุ (ที่ความยาวรอยร้าวขณะนั้น) และ 2) อัตราการเปลี่ยนแปลงของค่าพารามิเตอร์ K เทียบกับการเปลี่ยนแปลงความยาวรอยร้าว มีค่าเท่ากับการเปลี่ยนแปลงของความต้านทานการแตกหักเทียบกับการเปลี่ยนแปลงความยาวรอยร้าว ในกรณีนี้ความต้านทานการแตกหัก K_R จะเป็นฟังก์ชันของความยาวรอยร้าว (หรือความยาวรอยร้าวที่เพิ่มขึ้น Δa) หรือ

$$K_R = f(\Delta a) = f(a - a_0) \qquad (2.9)$$

สำหรับการเสียรูปโหมดที่ 1 เกณฑ์การเติบโตอย่างไร้เสถียรภาพจะเขียนได้ในรูปต่อไปนี้

$$K(a_0 + \Delta a) = K_R(a_0 + \Delta a) \qquad (2.10ก)$$

และ
$$\frac{dK}{da}(a_0 + \Delta a) = \frac{dK_R}{da}(a_0 + \Delta a) \qquad (2.10ข)$$

ในกรณีที่ปลายรอยร้าวมีบริเวณครากขนาดเล็ก เกณฑ์การเติบโตอย่างไร้เสถียรภาพจะเขียนอยู่ในรูปต่อไปนี้

$$K(a_0 + r_y + \Delta a) = K_R(a_0 + r_y + \Delta a) \qquad (2.11ก)$$

และ
$$\frac{dK}{da}(a_0 + r_y + \Delta a) = \frac{dK_R}{da}(a_0 + r_y + \Delta a) \qquad (2.11ข)$$

2.5 การเติบโตของรอยร้าวล้า

เมื่อชิ้นส่วนที่มีรอยร้าวต้องรับภาระที่กระทำเป็นรอบ (cyclic loading) รอยร้าวที่มีอยู่สามารถเติบโตจากความยาวเดิมได้ ในกรณีนี้นิยมเรียกรอยร้าวว่ารอยร้าวล้า (fatigue crack) การประยุกต์กลศาสตร์การแตกหักกับปัญหาการเติบโตของรอยร้าวล้าคือ การประยุกต์พารามิเตอร์ปลายรอยร้าว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ที่เหมาะสมเพื่อสร้างสหสัมพันธ์กับอัตราการเติบโตของรอยร้าว (crack growth rate) สำหรับปัญหาในขอบเขตของ LEFM มีการเสนอสมการต่อไปนี้

$$\frac{da}{dN} = C(\Delta K)^m \tag{2.12}$$

โดย C และ m คือ ค่าคงตัวซึ่งขึ้นกับชนิดวัสดุ สภาวะของภาระ สภาพแวดล้อม ฯลฯ

ΔK คือ พิสัยของพารามิเตอร์ K (stress intensity factor range)

พิสัยของพารามิเตอร์ K หาได้จากสมการต่อไปนี้

$$\Delta K = K_{\max} - K_{\min} \tag{2.13}$$

โดย K_{max} และ K_{min} คือ ค่าสูงสุดและต่ำสุดของพารามิเตอร์ K ในรอบภาระเดียวกัน ตามลำดับ

สมการที่ (2.12) ใช้สำหรับคำนวณอายุการเติบโต (จำนวนรอบภาระ) ของรอยร้าวจากความยาวเริ่มต้นถึงขนาดสิ้นสุด หรือในทางกลับกันคือใช้คำนวณหาความยาวรอยร้าวหลังจากรับภาระเป็นจำนวนรอบที่กำหนด ในกรณีของการประเมินการคงสภาพของโครงสร้าง ความยาวรอยร้าวเริ่มต้นก็คือความยาวที่ตรวจพบ ณ เวลาประเมินโครงสร้าง หากการประเมินต้องการทราบอายุใช้งานที่เหลือของโครงสร้างจะต้องคำนวณความยาวรอยร้าวที่ยอมรับได้ (ความยาวรอยร้าววิกฤติหารด้วยตัวประกอบความปลอดภัย) จากนั้นจึงคำนวณจำนวนรอบที่รอยร้าวใช้เติบโตจากความยาวเริ่มต้นถึงความยาวที่ยอมรับได้ภายใต้สภาวะใช้งานที่โครงสร้างรับหลังการประเมินครั้งปัจจุบัน

การคำนวณความยาวรอยร้าว ณ จำนวนรอบภาระใด ๆ ทำได้หลายวิธีได้แก่ วิธีอินทิเกรตเชิงตัวเลข หรือวิธีแก้สมการเชิงอนุพันธ์ [15] อย่างไรก็ดี วิธีคำนวณที่ง่ายที่สุดคือ การคำนวณทีละรอบ (cycle by cycle) สำหรับอัตราการเติบโตของรอยร้าวในรูปของสมการที่ (2.12) ความยาวรอยร้าว ณ รอบภาระที่ i คือ a_{i+1} ซึ่งคำนวณได้จากสมการต่อไปนี้

$$a_{i+1} = a_i + \Delta a_i \;\; ; \; i = 0, 1, 2 \mathbf{K} \tag{2.14ก}$$

และ

$$\Delta a_i = C(\Delta K_i)^m \tag{2.14ข}$$

โดย a_i คือ ความยาวรอยร้าว ณ รอบภาระที่ i

Δa_i คือ ความยาวรอยร้าวที่เพิ่มขึ้นต่อรอบ ณ รอบภาระที่ i

ΔK_i คือ ค่าพารามิเตอร์ ΔK ที่ความยาวรอยร้าว a_i

การคำนวณจะทำซ้ำตามจำนวนรอบภาระ (หรือรอบการทำซ้ำที่กำหนด) หรือทำซ้ำจนกระทั่งความยาวรอยร้าวถึงค่าที่กำหนดก็ขึ้นกับว่าการวิเคราะห์นั้นต้องการทราบอะไร

2.6 การเติบโตของรอยร้าวเนื่องจากสิ่งแวดล้อม

เงื่อนไขอีกอย่างหนึ่งที่สามารถทำให้รอยร้าวเติบโตได้คือ การที่โครงสร้างรับความเค้นสถิตย์ในสภาพแวดล้อมกัดกร่อน หรือการเติบโตของรอยร้าวเนื่องจากความเค้นร่วมกับการกัดกร่อน (stress-corrosion cracking, SCC) อัตราการเติบโตของรอยร้าวในกรณีนี้สามารถเขียนในรูปสมการต่อไปนี้ได้

$$\frac{da}{dt} = DK^n \tag{2.15}$$

โดย D และ n คือ ค่าคงตัวซึ่งขึ้นกับชนิดวัสดุ สภาวะของภาระ สภาพแวดล้อม ฯลฯ

สมการที่ (2.15) ใช้สำหรับคำนวณหาอายุการเติบโต (ระยะเวลา) ของรอยร้าวจากความยาวเริ่มต้นถึงขนาดสิ้นสุด หรือในทางกลับกันคือใช้คำนวณหาความยาวรอยร้าวหลังจากรับภาระ (สถิตย์) เป็นระยะเวลาที่กำหนด ในกรณีของการประเมินการคงสภาพของโครงสร้าง ความยาวรอยร้าวเริ่มต้นก็คือความยาวที่ตรวจพบ ณ เวลาประเมินโครงสร้าง หากการประเมินต้องการทราบอายุใช้งานที่เหลือของโครงสร้างจะต้องคำนวณความยาวรอยร้าวที่ยอมรับได้ (ความยาวรอยร้าววิกฤติหารด้วยตัวประกอบความปลอดภัย) จากนั้นจึงคำนวณจำนวนรอบที่รอยร้าวใช้เติบโตจากความยาวเริ่มต้นถึงความยาวที่ยอมรับได้ภายใต้สภาวะใช้งานที่โครงสร้างรับหลังการประเมินครั้งปัจจุบัน

วิธีคำนวณความยาวรอยร้าวที่เวลาใด ๆ ทำได้โดยแบ่งช่วงเวลาที่รับภาระ t ออกเป็นช่วงเวลาย่อย (time step) Δt_i สำหรับอัตราการเติบโตของรอยร้าวในรูปของสมการที่ (2.15) ความยาวรอยร้าวเมื่อสิ้นสุดการรับภาระในช่วงเวลาย่อย ช่วงที่ i คือ a_{i+1} ซึ่งคำนวณได้จากสมการต่อไปนี้

$$a_{i+1} = a_i + \Delta a_i \quad ; \; i = 0, 1, 2\mathbf{K} \qquad (2.16\text{ก})$$

และ
$$\Delta a_i = D(K_i)^n \Delta t_i \qquad (2.16\text{ข})$$

โดย a_i คือ ความยาวรอยร้าวตอนเริ่มรับภาระในช่วงย่อยเวลาช่วงที่ i

Δa_i คือ ความยาวรอยร้าวที่เพิ่มขึ้นหลังรับภาระเป็นเวลา Δt_i

K_i คือ ค่าพารามิเตอร์ K ที่ความยาวรอยร้าว a_i

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

ระเบียบวิธี R6

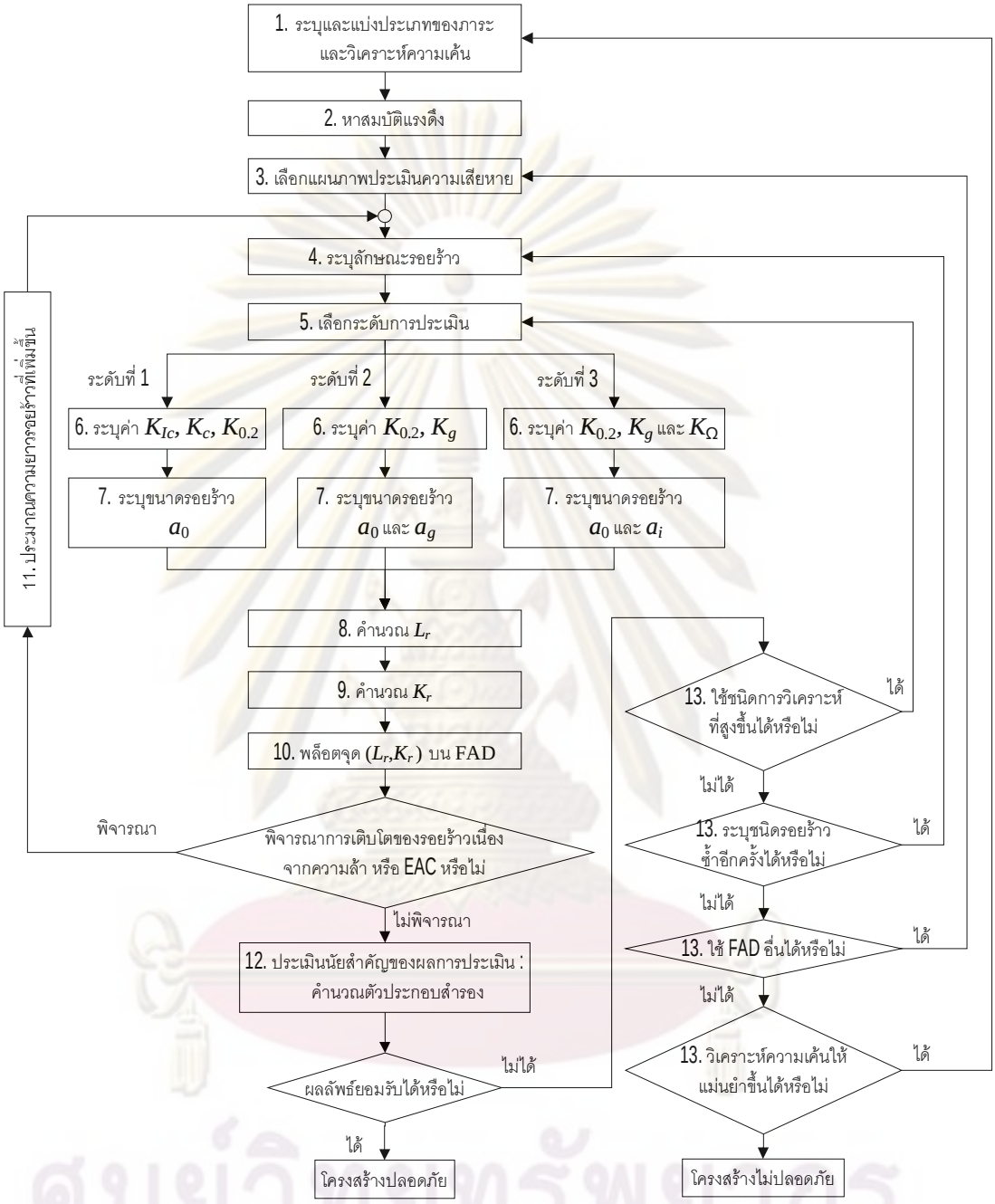
ในบทนี้จะกล่าวถึง การประเมินการคงสภาพของโครงสร้างด้วยระเบียบวิธี R6 หัวข้อที่ 3.1 จะกล่าวถึงขอบเขตและภาพรวมของขั้นตอนการประเมิน หัวข้อที่ 3.2 ถึง 3.14 จะกล่าวถึงรายละเอียดในแต่ละขั้นตอนของการประเมิน ซึ่งมีทั้งหมด 13 ขั้น เรียงตามลำดับ

3.1 ขอบเขตและภาพรวมของขั้นตอนการประเมิน

ขอบเขตของระเบียบวิธี R6 ในการประเมินการคงสภาพของโครงสร้างที่มีรอยร้าว คือ 1) โครงสร้างไม่ได้ทำงานในสภาวะที่การคืบมีผล 2) โครงสร้างไม่ได้รับแรงกระแทก (impact load) และ 3) โครงสร้างไม่เสียหายเนื่องจากการโก่งงออย่างไร้เสถียรภาพ (buckling) ขั้นตอนการประเมินการคงสภาพด้วยระเบียบวิธี R6 มีทั้งหมด 13 ขั้นตอน [5] ดังนี้

1) แบ่งประเภทของภาระและวิเคราะห์ความเค้นจากภาระแต่ละประเภท (หัวข้อที่ 3.2)
2) หาสมบัติแรงดึงของวัสดุที่ทำโครงสร้าง ได้แก่ ความต้านแรงดึงคราก ค่าโมดูลัสของความยืดหยุ่น (หัวข้อที่ 3.3)
3) เลือกชนิดของ FAD (หัวข้อที่ 3.4)
4) ระบุลักษณะรอยร้าว (หัวข้อที่ 3.5)
5) เลือกระดับการวิเคราะห์ (หัวข้อที่ 3.6)
6) ระบุค่าความต้านทานการแตกหักสำหรับระดับการวิเคราะห์ที่เลือก (หัวข้อที่ 3.7)
7) ระบุขนาดรอยร้าว (หัวข้อที่ 3.8)
8) คำนวณ L_r (หัวข้อที่ 3.9)
9) คำนวณ K_r (หัวข้อที่ 3.10)
10) พล็อตจุดประเมิน (L_r, K_r) บน FAD ที่เลือก (หัวข้อที่ 3.11)
11) พิจารณาว่าต้องการวิเคราะห์การเติบโตของรอยร้าวหรือไม่ (หัวข้อที่ 3.12)
12) ประเมินนัยสำคัญของผลการประเมิน (หัวข้อที่ 3.13)
13) แก้ไขรายละเอียดบางขั้นตอนของการประเมิน หากการประเมินที่ผ่านมาใช้วิธีหรือข้อมูลซึ่งจะให้ผลลัพธ์ด้านปลอดภัย แล้วการประเมินบ่งชี้ว่าโครงสร้างไม่ปลอดภัย (หัวข้อที่ 3.14)

ขั้นตอนทั้งหมดสามารถสรุปเป็นแผนผังได้ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 แผนผังขั้นตอนการประเมินการคงสภาพด้วยระเบียบวิธี R6

3.2 การแบ่งประเภทของภาระ และวิเคราะห์ความเค้น

การจำแนกประเภทของภาระมีความสำคัญเพราะว่าภาระแต่ละประเภทมีผลต่อโหมดความเสียหายของโครงสร้างไม่เหมือนกัน ภาระที่กระทำกับโครงสร้างสามารถแบ่งออกเป็น 2 ประเภท

คือ ภาระปฐมภูมิ และภาระทุติยภูมิ ซึ่งภาระทั้งสองจะทำให้เกิดความเค้นปฐมภูมิ (primary stress) และความเค้นทุติยภูมิ (secondary stress) ตามลำดับ

ความเค้นปฐมภูมิ คือ ความเค้นที่เกิดจากแรงภายนอกที่มากระทำกับโครงสร้าง เช่น ความเค้นที่เกิดจากความดัน ความเค้นที่เกิดจากน้ำหนักของโครงสร้าง เป็นต้น หากความเค้นมีค่าสูงพอจะสามารถทำให้โครงสร้างเสียหายจากการครากได้

ความเค้นทุติยภูมิ คือ ความเค้นที่สมดุลในตัวเอง (self-equilibrium) ดังนั้นแรงลัพธ์และโมเมนต์ลัพธ์บนหน้าตัดใด ๆ จะเท่ากับศูนย์ ความเค้นทุติยภูมิจะไม่ทำให้โครงสร้างเสียหายจากการคราก ตัวอย่างของความเค้นทุติยภูมิ ได้แก่ ความเค้นตกค้างบริเวณรอยเชื่อม ความเค้นเนื่องจากอุณหภูมิ เป็นต้น อย่างไรก็ตาม ความเค้นทุติยภูมิมีผลต่อค่าของพารามิเตอร์ K โดยพารามิเตอร์ K สุทธิจะเท่ากับผลบวกของตัวพารามิเตอร์ K เนื่องจากความเค้นปฐมภูมิและความเค้นทุติยภูมิ ตามลำดับ

3.3 การหาสมบัติแรงดึงของวัสดุ

สมบัติแรงดึงของวัสดุที่ต้องใช้ในการประเมินประกอบด้วย ความเค้นคราก σ_y หรือ ความเค้นดึงสูงสุด σ_u และมอดูลัสของความยืดหยุ่น E

สำหรับ σ_y ในกรณีที่กราฟความเค้น-ความเครียดแสดงจุดครากชัดเจนจะให้ความเค้นครากเท่ากับ ความเค้นครากขั้นล่าง (lower yield stress) แต่ถ้าหากเส้นโค้งความเค้น-ความเครียดแสดงจุดครากไม่ชัดเจนจะให้ความเค้นครากเท่ากับ ความเค้นพิสูจน์ (proof stress) สำหรับวัสดุที่แสดงพฤติกรรม strain hardening ระเบียบวิธี R6 แนะนำให้ใช้ ความเค้นไหล (flow stress) $\bar{\sigma}$ สำหรับหาขีดจำกัดของอัตราส่วนภาระ $L_r^{\max}$ (หัวข้อที่ 3.4) โดยความเค้นไหล คือ

$$\bar{\sigma} = \frac{1}{2}\left(\sigma_y + \sigma_u\right) \tag{3.1}$$

3.4 การเลือกแผนภาพประเมินความเสียหาย

ระเบียบวิธี R6 มีแผนภาพประเมินความเสียหาย หรือ FAD ให้เลือก 3 แบบ แต่ละแบบมีเส้นโค้งประเมินความเสียหายต่างกัน ในที่นี้จะกล่าวถึงเส้นโค้งประเภทที่ 1 และประเภทที่ 2 เท่านั้น

แบบที่ 1 คือ เส้นโค้งทั่วไป เส้นโค้งประเมินความเสียหายกรณีนี้จะไม่ขึ้นกับชนิดของวัสดุ แต่ใช้ได้เฉพาะกรณีที่เส้นโค้งความเค้น-ความเครียดเป็นเส้นโค้งต่อเนื่อง สมการเส้นโค้งประเมินความเสียหายคือ

$$K_r = \begin{cases} \left(1-0.14L_r^2\right)\left[0.3+0.7\exp\left(-0.65L_r^6\right)\right] & ; \quad L_r \le L_r^{\max} \\ 0 & ; \quad L_r > L_r^{\max} \end{cases} \tag{3.2}$$

โดย $L_r^{\max}$ คือ ค่าขีดจำกัดของอัตราส่วนภาระ ซึ่งมีนิยามว่า

$$L_r^{\max} = \frac{\bar{\sigma}}{\sigma_y} \tag{3.3}$$

ถ้าใช้เส้นโค้งประเมินความเสียหายแบบนี้แล้ว สมบัติแรงดึงที่ต้องการคือ ค่ากลางของความเค้นครากขั้นล่าง และค่ากลางของความเค้นไหล

แบบที่ 2 คือ เส้นโค้งที่ขึ้นกับชนิดของวัสดุ สมการเส้นโค้งประเมินความเสียหายกรณีนี้คือ

$$K_r = \begin{cases} \left(\dfrac{\left(\varepsilon_{ref}/\varepsilon_y\right)}{L_r} + \dfrac{L_r^3}{2\left(\varepsilon_{ref}/\varepsilon_y\right)}\right)^{-\frac{1}{2}} & ; \quad L_r \le L_r^{\max} \\ 0 & ; \quad L_r > L_r^{\max} \end{cases} \tag{3.4}$$

โดย ε_{ref} คือ ความเครียด ณ ความเค้นอ้างอิง (reference stress) σ_{ref}

ε_y คือ ความเครียดคราก เท่ากับ σ_y/E

ความเค้นอ้างอิงหาได้จากสมการต่อไปนี้

$$\sigma_{ref} = L_r\sigma_y \tag{3.5}$$

ส่วน ε_{ref} อ่านค่าได้โดยตรงจากเส้นโค้งความเค้น-ความเครียดจริง (true stress-strain curve) ถ้าสมมุติว่าเส้นโค้งความเค้น-ความเครียดจริง แทนได้ด้วยสมการของ Ramberg-Osgood (สมการที่ 3.6)

$$\frac{\varepsilon}{\varepsilon_0} = \frac{\sigma}{\sigma_0} + \alpha\left(\frac{\sigma}{\sigma_0}\right)^n \tag{3.6}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

โดย ε คือ ความเครียดจริง

σ คือ ความเค้นจริง

ε_0 คือ ความเครียดนอร์มัลไลซ์ ซึ่งมักจะเป็น ε_y

σ_0 คือ ความเค้นนอร์มัลไลซ์ ซึ่งมักจะเป็น σ_y

α, n คือ ค่าคงตัวของวัสดุ

แล้วจะได้ ε_{ref} คือ

$$\varepsilon_{ref} = \varepsilon_0 \left[\frac{\sigma_{ref}}{\sigma_0} + \alpha \left(\frac{\sigma_{ref}}{\sigma_0} \right)^n \right] \tag{3.7}$$

ถ้าใช้เส้นโค้งประเมินความเสียหายชนิดนี้แล้ว สมบัติแรงดึงที่ต้องการคือ ค่ากลางของความเค้นครากขั้นล่าง ค่ากลางของความเค้นไหล และเส้นโค้งความเค้น-ความเครียดจริง (หรือความสัมพันธ์ทางคณิตศาสตร์ที่แทนเส้นโค้งได้แม่นยำ)

3.5 **การระบุลักษณะรอยร้าว**

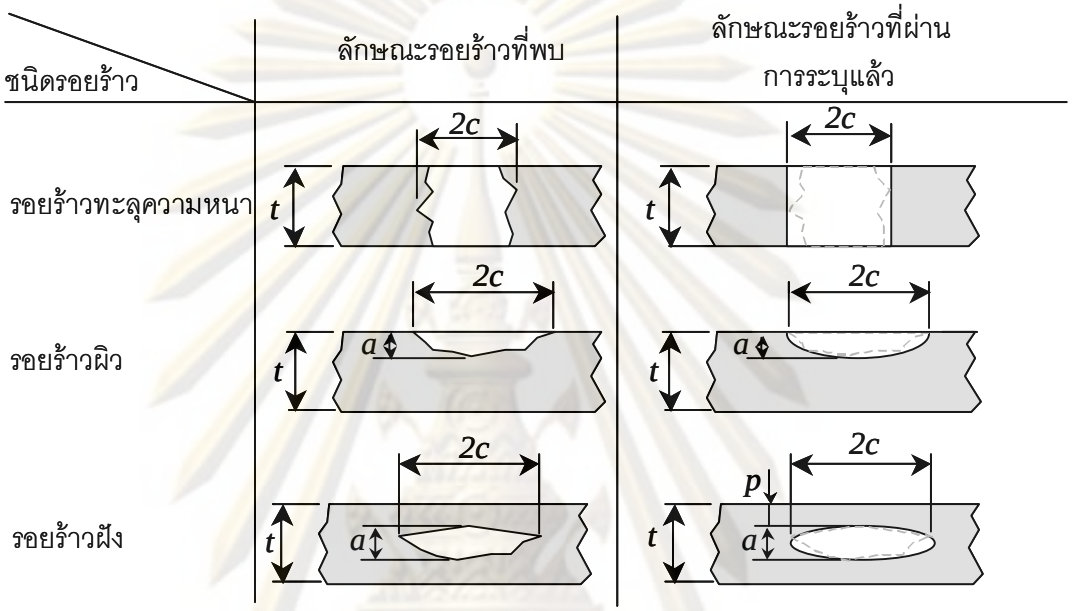
การระบุลักษณะรอยร้าวหรือการให้ลักษณะของรอยร้าวทำเพื่อเปลี่ยนลักษณะของรอยร้าวให้อยู่ในรูปแบบที่มีผลเฉลยของปลายรอยร้าวและภาระขีดจำกัดได้ เนื่องจากระเบียบวิธี R6 ไม่ได้ให้รายละเอียดส่วนนี้ไว้ครบถ้วน วิทยานิพนธ์นี้จึงใช้คำแนะนำวิธีระบุลักษณะของรอยร้าวของระเบียบวิธี API 579 [7] การระบุลักษณะรอยร้าวประกอบด้วย 4 ขั้นตอนย่อย ดังนี้ 1) การระบุรูปร่างอุดมคติของรอยร้าว (flaw idealization) 2) การจัดทิศทางรอยร้าว (crack orientation) 3) การพิจารณาปฏิสัมพันธ์ของรอยร้าว (crack interaction) และ 4) การระบุชนิดรอยร้าวซ้ำอีกครั้ง (crack recategorization) รายละเอียดทั้ง 4 ขั้นตอนจะกล่าวในหัวข้อที่ 3.5.1 ถึง 3.5.4

3.5.1 การระบุรูปร่างอุดมคติของรอยร้าว

การระบุรูปร่างอุดมคติของรอยร้าว หมายถึง การแทนขอบหน้าของรอยร้าวที่พบด้วยเส้นโค้งเรียบหรือเส้นตรง รูปร่างอุดมคติของรอยร้าวต้องรักษาเค้าโครงเดิมของรอยร้าว โดยทั่วไปเส้นโค้งเรียบที่ใช้แทนขอบหน้ารอยร้าวจะล้อมรอบขอบหน้าของรอยร้าวจริง ดังตารางที่ 3.1 การทำเช่นนี้ช่วยให้ผลการประเมินโน้มเอียงไปในด้านความปลอดภัยเพิ่มขึ้น เพราะทำให้ขนาดรอยร้าวเพิ่มขึ้น สำหรับตัวอย่างในตารางที่ 3.1 จะเห็นว่ารูปร่างอุดมคติของรอยร้าวทะลุความหนา (through-thickness crack) คือ รอยร้าวทะลุความหนา ที่มีความยาวรอยร้าวเท่ากันตลอดความหนา

ส่วนรอยร้าวผิว (surface crack) และรอยร้าวฝัง (embedded crack) จะถูกแทนด้วยรูปครึ่งวงรีและวงรี ตามลำดับ

ตารางที่ 3.1 การระบุรูปร่างอุดมคติของรอยร้าว



3.5.2 การจัดทิศทางรอยร้าว

ระเบียบวิธี API 579 แนะนำการจัดทิศทางของรอยร้าวเฉียง ไว้ 2 วิธี วิธีแรกคือ จัดให้ระนาบของรอยร้าวตั้งฉากกับทิศของความเค้นหลักสูงสุด และกำหนดให้ความยาวและความลึกของรอยร้าวดังกล่าวเท่ากับความยาวและความลึกของรอยร้าวที่ตรวจพบ วิธีที่สองคือ จัดให้ระนาบของรอยร้าวตั้งฉากกับทิศของความเค้นหลัก และกำหนดให้ความยาวและความลึกของรอยร้าวดังกล่าวเท่ากับความยาวสมมูล (equivalent crack length) และความลึกสมมูล (equivalent crack depth) ตามลำดับ เนื่องจากวิทยานิพนธ์นี้เลือกใช้วิธีที่สอง ดังนั้นจะกล่าวถึงรายละเอียดเฉพาะวิธีนี้ดังต่อไปนี้

ขั้นแรกคำนวณหาอัตราส่วนของความเค้นหลัก (biaxiality ratio) B จากสมการที่ (3.8)

$$B = \frac{\sigma_2}{\sigma_1} \tag{3.8}$$

โดย $0 \le \sigma_2 \le \sigma_1$ ดังนั้น $0 \le B \le 1$

ขั้นที่สองคำนวณความยาวรอยร้าวสมมูล c_{eq} (รูปที่ 3.2) จากสมการต่อไปนี้

กรณี $\alpha \le 45$ องศา ฉายรอยร้าวไปบนระนาบที่ σ_1 กระทำ และ

$$\frac{c_{eq}}{c} = \cos^2\alpha + \frac{(1-B)\sin\alpha\cos\alpha}{2} + B^2\sin\alpha \tag{3.9}$$

กรณี $\alpha > 45$ องศา ฉายรอยร้าวไปบนระนาบที่ σ_2 กระทำ และ

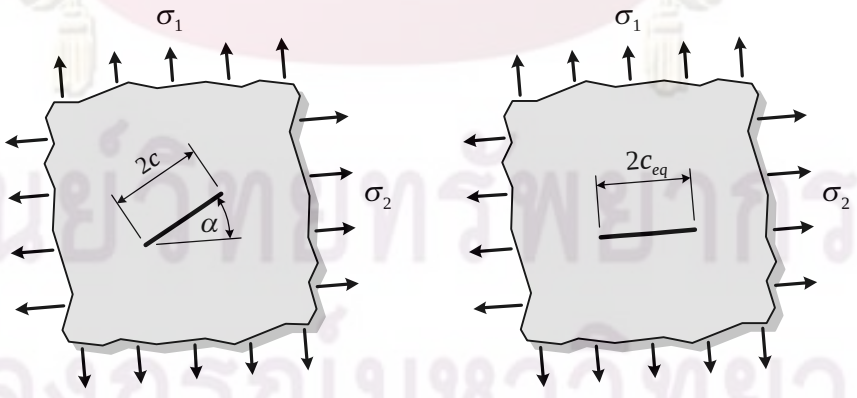
$$\frac{c_{eq}}{c} = \frac{\cos^2\alpha}{B^2} + \frac{(1-B)\sin\alpha\cos\alpha}{2B^2} + \sin^2\alpha \tag{3.10}$$

ขั้นที่สามคำนวณความลึกสมมูล a_{eq} โดยเริ่มจากการฉายรอยร้าวไปบนระนาบที่ใช้หาความยาวสมมูล (รูปที่ 3.3(ก)) จากรูปความลึกของรอยร้าวจากการฉาย คือ $a_0 = a\cos(\theta)$ จากนั้นคำนวณความลึกสมมูล a_{eq} โดยใช้สมการ

$$a_{eq} = a_0 W \tag{3.11}$$

และ W คือ ตัวประกอบปรับแก้ความลึก ซึ่งหาได้จาก

$$W = \text{ค่าสูงสุดระหว่าง } W_\theta \text{ และ } 1 \tag{3.12}$$



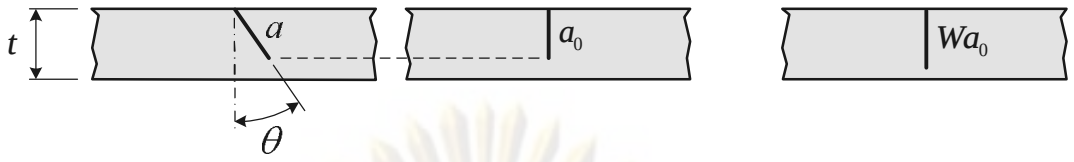
(ก) (ข)

รูปที่ 3.2 การจัดทิศทางรอยร้าวเฉียงตามระเบียบวิธี API 579 วิธีที่ 2

ก) รอยร้าวและสนามความเค้น

ข) รอยร้าวที่จัดทิศทางแล้ว (กรณี $\alpha \le 45^o$)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



(ก) (ข)

รูปที่ 3.3 การระบุความลึกของรอยร้าว ตามมาตรฐาน API 579
(ก) การฉายระนาบรอยร้าว (ข) ความลึกสมมูล

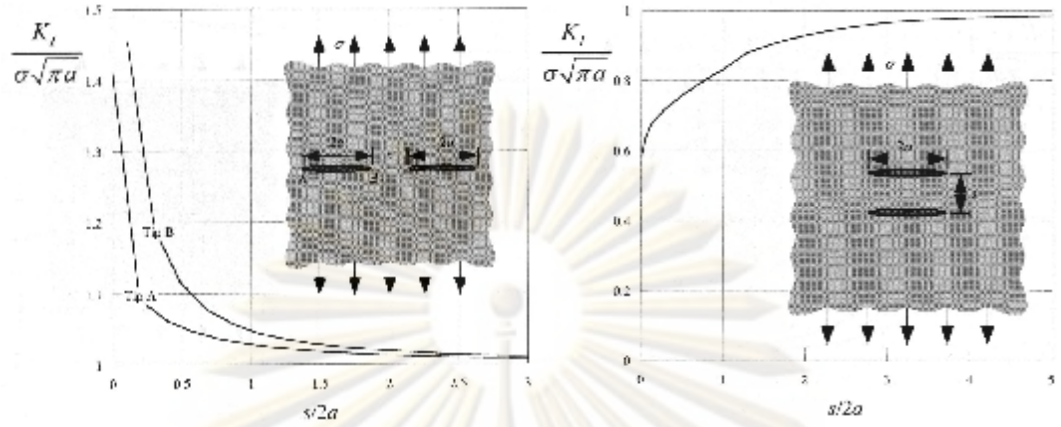
และ

$$W_\theta = 0.99999 + 1.0481\times10^{-5}\theta + 1.5471\times10^{-4}\theta^2 + 3.4141\times10^{-5}\theta^3 - 2.0688\times10^{-6}\theta^4 + 4.4977\times10^{-8}\theta^5 - 4.5751\times10^{-10}\theta^6 + 1.8220\times10^{-12}\theta^7 \quad (3.13)$$

โดย θ คือ มุมเอียง มีหน่วยเป็นองศา

3.5.3 การพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว

ปฏิสัมพันธ์ของรอยร้าว หมายถึง การที่สนามความเค้นของรอยร้าวที่พิจารณาถูกทำให้เปลี่ยนแปลงไปเนื่องจากการมีอยู่ของรอยร้าวในบริเวณใกล้เคียง การเปลี่ยนแปลงสนามความเค้นทำให้พารามิเตอร์ K มีค่าเปลี่ยนแปลง ซึ่งจะมากขึ้นหรือน้อยลงขึ้นอยู่กับการตำแหน่งสัมพัทธ์ของรอยร้าว ยกตัวอย่างเช่น รูปที่ 3.4 (ก) เป็นกรณีรอยร้าวอยู่บนระนาบเดียวกัน (coplanar crack) ในวัตถุแผ่นแบนขนาดไม่จำกัด เนื่องจากรอยร้าวแต่ละรอยต่างส่งผลถึงรอยร้าวอีกรอยเหมือนกัน ดังนั้นจะพิจารณารอยร้าวซ้ายมือ จากรูปแกนตั้งคือพารามิเตอร์ K ที่ถูกนอร์มัลไลซ์ด้วยพารามิเตอร์ K ของรอยร้าว (รอยเดียว) ยาว $2a$ ในแผ่นแบนขนาดไม่จำกัด ดังนั้นถ้ารอยร้าวสองรอยไม่มีปฏิสัมพันธ์กันแล้วอัตราส่วนดังกล่าวจะเท่ากับ 1 แต่จากรูปจะเห็นว่ารอยร้าวขวามือทำให้อัตราส่วนดังกล่าวที่ปลาย A และ B เพิ่มขึ้น (แสดงว่ามีปฏิสัมพันธ์กัน) ปลาย B ได้รับผลรุนแรงกว่า ระดับของการปฏิสัมพันธ์จะเพิ่มขึ้นเมื่อระยะระหว่างปลายรอยร้าวที่ประชิดกัน s ลดลง รูปที่ 3.4 (ข) แสดงปฏิสัมพันธ์ของรอยร้าวที่อยู่ต่างระนาบกัน (noncoplanar crack) จากรูปจะเห็นว่าเมื่อรอยร้าวเข้ามาใกล้กันมากขึ้น (s ลดลง) ระดับของการปฏิสัมพันธ์จะลดลง



(ก) (ข)

รูปที่ 3.4 ผลของตำแหน่งสัมพัทธ์ของรอยร้าวต่อค่าของพารามิเตอร์ K [15]

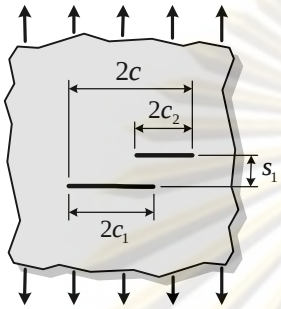
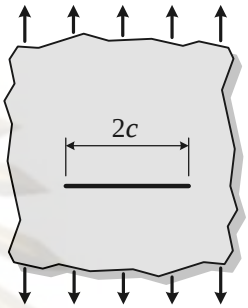
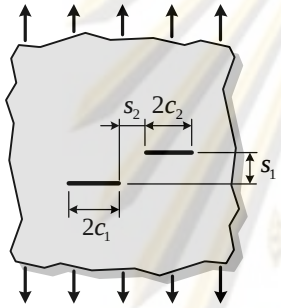
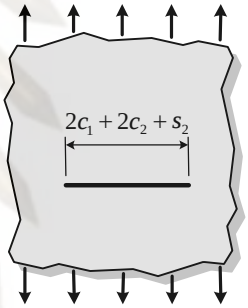
ตัวอย่างในรูปที่ 3.4 แสดงให้เห็นความจำเป็นในการพิจารณาปฏิสัมพันธ์ของรอยร้าว และยังช่วยให้คาดการณ์ได้ว่าปฏิสัมพันธ์ของรอยร้าวขึ้นกับปัจจัยหลายอย่าง เช่น ตำแหน่งสัมพัทธ์ ขนาดสัมพัทธ์ของรอยร้าว เป็นต้น ดังนั้นจึงเป็นเรื่องยากที่จะหาผลเฉลย K ที่ครอบคลุมปัจจัยต่าง ๆ ได้ทั้งหมดแม้จะมีรอยร้าวเพียงสองรอย ด้วยเหตุนี้ระเบียบวิธีประเมินการคงสภาพของโครงสร้างจึงมีข้อแนะนำสำหรับจัดการรอยร้าวที่มีปฏิสัมพันธ์กัน แนวคิดก็คือการรวมรอยร้าวให้เป็นรอยร้าวรอยเดียว รอยร้าวที่รวมแล้วจะรุนแรงกว่าตอนก่อนรวม ทั้งนี้เพื่อให้ผลการประเมินอยู่ในด้านที่ปลอดภัย

เกณฑ์การรวมรอยร้าว และมิติของรอยร้าวหลังรวมกัน จะแตกต่างกันไปในแต่ละระเบียบวิธี อย่างไรก็ดี ทุกระเบียบวิธีจะพิจารณา 2 กรณี คือ 1) ปฏิสัมพันธ์ระหว่างรอยร้าวต่างระนาบ และ 2) ปฏิสัมพันธ์ระหว่างรอยร้าวบนระนาบเดียวกัน กรณีแรกทำเพื่อจัดรอยร้าวที่มีปฏิสัมพันธ์กันแต่อยู่ต่างระนาบให้มาอยู่บนระนาบเดียวกัน จากนั้นจึงใช้เกณฑ์ปฏิสัมพันธ์ระหว่างรอยร้าวบนระนาบเดียวกัน ตารางที่ 3.2 และ 3.3 สรุปเกณฑ์ทั้งสองกลุ่มตามข้อแนะนำในมาตรฐาน API 579

3.5.4 การระบุชนิดรอยร้าวซ้ำอีกครั้ง

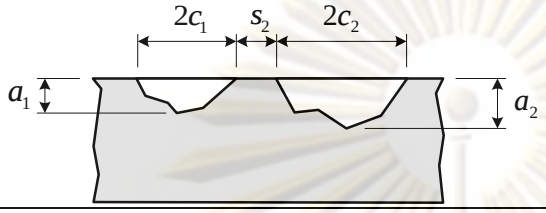
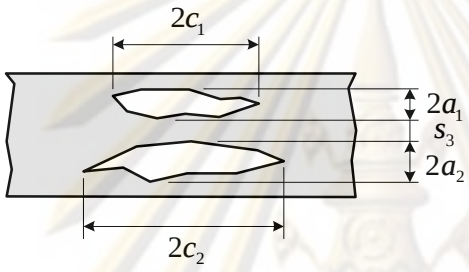
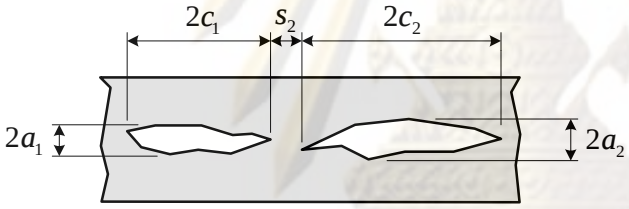
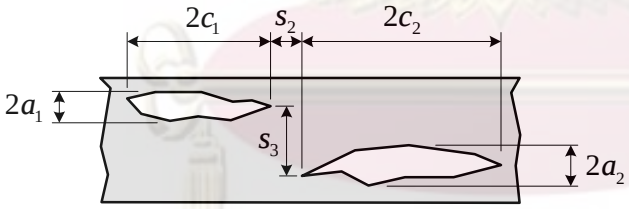
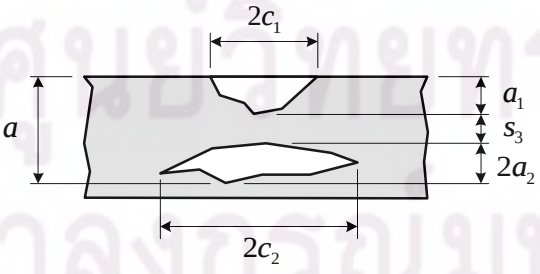
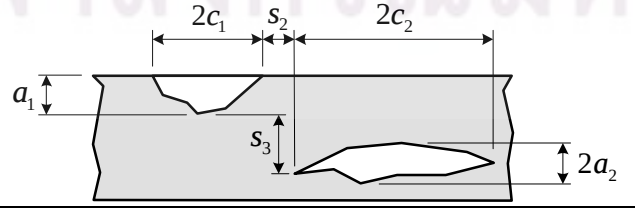
การระบุชนิดของรอยร้าวซ้ำอีกครั้ง จะกล่าวถึงในหัวข้อที่ 3.14

ตารางที่ 3.2 ปฏิสัมพันธ์ของรอยร้าวต่างระนาบ (ระเบียบวิธี API 579)

รอยร้าวก่อนพิจารณา	เงื่อนไขการรวม	รอยร้าวที่รวมแล้ว
	$c_1 + c_2 \geq s_1$	
	$c_1 + c_2 \geq s_1$ และ $c_1 + c_2 \geq s_2$	

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.3 ปฏิสัมพันธ์ของรอยร้าวบนระนาบเดียวกัน (ระเบียบวิธี API 579)

รอยร้าวก่อนพิจารณา	เงื่อนไขการรวม	รอยร้าวที่รวมแล้ว
	$c_1 + c_2 \geq s_2$	รอยร้าวรูปครึ่งวงรี โดย $2c = 2c_1 + 2c_2 + s_2$ $a = \max[a_1, a_2]$
	$a_1 + a_2 \geq s_3$	รอยร้าวฝังรูปวงรี โดย $2c = \max[2c_1, 2c_2]$ $2a = 2a_1 + 2a_2 + s_3$
	$c_1 + c_2 \geq s_2$	รอยร้าวฝังรูปวงรี โดย $2c = 2c_1 + 2c_2 + s_2$ $2a = \max[2a_1, 2a_2]$
	$c_1 + c_2 \geq s_2$ และ $a_1 + a_2 \geq s_3$	รอยร้าวฝังรูปวงรี โดย $2c = 2c_1 + 2c_2 + s_2$ $2a = 2a_1 + 2a_2 + s_3$
	$a_1 + a_2 \geq s_3$	รอยร้าวรูปครึ่งวงรี โดย $2c = \max[2c_1, 2c_2]$ $a = a_1 + 2a_2 + s_3$
	$c_1 + c_2 \geq s_2$ และ $a_1 + a_2 \geq s_3$	รอยร้าวรูปครึ่งวงรี โดย $2c = 2c_1 + 2c_2 + s_2$ $a = a_1 + 2a_2 + s_3$

3.6 การเลือกระดับการประเมิน

ระเบียบวิธี R6 แบ่งระดับการประเมิน ออกเป็น 3 ระดับ ขึ้นกับกลไกการแตกหักที่เกิดขึ้น

การประเมินระดับที่ 1 ใช้เกณฑ์การแตกหักเป็นเกณฑ์การกำเนิดของรอยร้าว โดยที่การแตกหักบริเวณปลายรอยร้าวจะต้องเป็นแบบเปราะเท่านั้น

การประเมินระดับที่ 2 ใช้เกณฑ์การแตกหักเป็นเกณฑ์การกำเนิดของรอยร้าว แต่รวมผลของการเพิ่มขึ้นของความต้านทานการแตกหักเนื่องจากรอยร้าวเติบโตอย่างมีเสถียรภาพไปเป็นระยะทาง Δa_g

การประเมินระดับที่ 3 ใช้เกณฑ์การแตกหักเป็นเกณฑ์การเติบโตของรอยร้าว โดยรวมผลของการเพิ่มขึ้นของความต้านทานการแตกหักเนื่องจากการเติบโตอย่างไร้เสถียรภาพเป็นระยะ Δa ใด ๆ

วิทยานิพนธ์นี้จะกล่าวถึงการประเมินระดับที่ 1 และ 2 เท่านั้น

3.7 การหาค่าความต้านทานการแตกหัก

ความต้านทานการแตกหักที่ใช้ในการประเมิน K_{mat} จะขึ้นกับระดับของการประเมินที่เลือก สำหรับการประเมินระดับที่ 1 ความต้านทานการแตกหัก K_{mat} อาจจะเป็น K_C , K_{IC} หรือ $K_{0.2}$ ส่วนการประเมินระดับที่ 2 ความต้านทานการแตกหัก K_{mat} จะเป็น $K_{0.2}$ หรือ $K_{0.2/BL}$ สำหรับการประเมินครั้งแรก และกำหนดให้เป็น K_g สำหรับการประเมินครั้งที่สอง ความหมายของความต้านทานการแตกหักที่กล่าวไป มีดังนี้

K_C คือ ความต้านทานการแตกหัก ณ จุดที่รอยร้าวเริ่มต้นการแตกหักเปราะ และสถานะความเค้นที่ปลายรอยร้าวไม่ใช่ความเครียดระนาบ

K_{IC} คือ ความต้านทานการแตกหัก ณ จุดที่รอยร้าวเริ่มต้นการแตกหักเปราะ และสถานะความเค้นที่ปลายรอยร้าวเป็นแบบความเครียดระนาบ

$K_{0.2}$ คือ ความต้านทานการแตกหัก ณ จุดที่รอยร้าวเติบโตอย่างมีเสถียรภาพเป็นระยะ 0.2 มิลลิเมตร

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$K_{0.2/BL}$ คือ ความต้านทานการแตกหัก ณ จุดที่เส้นเยื้อง 0.2 มม.ตัดกับเส้นโค้ง J-Δa

K_g คือ ความต้านทานการแตกหัก ณ จุดที่รอยร้าวเติบโตอย่างมีเสถียรภาพเป็นระยะ Δa_g

3.8 การระบุขนาดของรอยร้าว

ขนาดรอยร้าวที่ใช้ในการคำนวณค่าพารามิเตอร์ K จะขึ้นกับระดับการประเมิน สำหรับการประเมินระดับที่ 1 จะใช้ขนาดรอยร้าวเท่ากับขนาดของรอยร้าวหลังการระบุลักษณะ a_0 สำหรับการประเมินระดับที่ 2 จะใช้ขนาดรอยร้าว a_0 สำหรับการประเมินครั้งแรก และจะใช้ขนาด a_g โดย $a_g = a_0 + \Delta a_g$ สำหรับการประเมินครั้งที่สอง

3.9 การคำนวณอัตราส่วนภาระ L_r

อัตราส่วนภาระ L_r คำนวณจากสมการต่อไปนี้

$$L_r = \frac{P}{P_L} \tag{3.14}$$

โดย P คือ ภาระปฐมภูมิ

P_L คือ ภาระขีดจำกัด

สมการที่ (3.14) สามารถเขียนในรูปของความเค้นได้ดังนี้

$$L_r = \frac{\sigma^P}{\sigma_L} \tag{3.15}$$

โดย σ^P คือ ความเค้นปฐมภูมิ

σ_L คือ ความเค้นขีดจำกัด (ความเค้นที่ทำให้เกิดการครากทั้งหน้าตัด)

3.10 การคำนวณอัตราส่วนพารามิเตอร์ K หรือ K_r

อัตราส่วนของพารามิเตอร์ K หรือ K_r สำหรับการเสียรูปในโหมดที่ 1 จะคำนวณจาก

$$K_r = \frac{K_I}{K_{mat}} \tag{3.16}$$

ในกรณีที่โครงสร้างรับภาระปฐมภูมิและภาระทุติยภูมิ อัตราส่วน K_r จะเท่ากับผลรวมของ K_r เนื่องจากภาระปฐมภูมิ K_r^p และทุติยภูมิ K_r^s หรือ

$$K_r = K_r^P + K_r^S \tag{3.17}$$

สำหรับการประเมินระดับที่ 1 K_r^p และ K_r^s หาได้จากสมการต่อไปนี้

$$K_r^P = \frac{K_I^P(a_0)}{K_{mat}} \tag{3.18}$$

$$K_r^S = \frac{K_I^S(a_0)}{K_{mat}} + \rho(a_0) \tag{3.19}$$

โดย K_{mat} คือ K_{IC} หรือ K_C หรือ $K_{0.2}$

ρ คือ ตัวประกอบปรับแก้พลาสติกซิตี้ (plasticity correction factor) คำนวณได้จาก

$$\rho = \begin{cases} \rho_1 & , \quad L_r \le 0.8 \\ 4\rho_1(1.05 - L_r) & , \quad 0.8 < L_r < 1.05 \\ 0 & , \quad 1.05 \le L_r \end{cases} \tag{3.20}$$

และ

$$\rho_1 = \begin{cases} 0 & , \quad K_{\mathrm{I}}^s/\left(K_{\mathrm{I}}^p/L_r\right) < 0 \\ 0.1\chi^2 - 0.007\chi^2 + 0.00003\chi^5 & , \quad 0 \le K_{\mathrm{I}}^s/\left(K_{\mathrm{I}}^p/L_r\right) < 5.2 \\ 0.25 & , \quad 5.2 \le K_{\mathrm{I}}^s/\left(K_{\mathrm{I}}^p/L_r\right) \end{cases} \tag{3.21}$$

และ $$\chi = \frac{K_{\mathrm{I}}^s L_r}{K_{\mathrm{I}}^p} \tag{3.22}$$

สำหรับการประเมินระดับที่ 2 จะต้องทำการประเมินสองครั้ง ครั้งแรกเหมือนกับการประเมินระดับที่ 1 แต่ใช้ K_{mat} เท่ากับ $K_{0.2}$ หรือ $K_{0.2/BL}$ และใช้ความยาวรอยร้าวเริ่มต้น a_0 ดังนั้น K_r^p และ K_r^s หาได้จากสมการต่อไปนี้

$$K_r^P = \frac{K_I^P(a_0)}{K_{0.2}} \tag{3.23}$$

$$K_r^S = \frac{K_I^S(a_0)}{K_{0.2}} + \rho(a_0) \tag{3.24}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ในการประเมินครั้งที่สอง ใช้ K_{mat} เท่ากับ K_g และความยาวรอยร้าว $a_g = a_0 + \Delta a_g$ ดังนั้น K_r^P และ K_r^s หาได้จากสมการต่อไปนี้

$$K_r^P = \frac{K_I^P\left(a_0 + \Delta a_g\right)}{K_g} \tag{3.25}$$

$$K_r^S = \frac{K_I^S\left(a_0 + \Delta a_g\right)}{K_g} + \rho\left(a_0 + \Delta a_g\right) \tag{3.26}$$

3.11 พล็อตจุด (L_r, K_r) บน FAD

สมการที่ (3.15) และสมการ (3.17) จะบ่งบอกสภาวะของโครงสร้างภายใต้ภาระ ในขั้นตอนนี้ ค่าทั้งสองหรือพิกัด (L_r, K_r) จะถูกนำไปพล็อตบน FAD ที่เลือก จากนั้นจึงพิจารณาตำแหน่งของจุดเทียบกับเส้นประเมินความเสียหาย ถ้าจุดประเมินอยู่บนเส้นประเมินความเสียหายหรืออยู่สูงกว่าจะถือว่าโครงสร้างเสียหาย แต่ถ้าอยู่ต่ำกว่าจะถือว่าโครงสร้างยังใช้งานได้

3.12 การพิจารณาว่าต้องการวิเคราะห์การเติบโตของรอยร้าวหรือไม่

เนื่องจากรอยร้าวที่พบในการประเมินอาจเติบโตได้หากโครงสร้างยังถูกใช้งานต่อไป ดังนั้นเพื่อให้มั่นใจว่ารอยร้าวจะไม่เติบโตถึงขนาดวิกฤติก่อนถึงรอบการประเมินครั้งถัดไปแล้ว ระเบียบวิธี R6 จึงมีขั้นตอนนี้ให้ผู้ประเมินพิจารณาว่ารอยร้าวมีโอกาสเติบโตหรือไม่ ถ้าไม่มีโอกาสก็สามารถข้ามขั้นตอนนี้ไปได้ แต่ถ้ารอยร้าวมีโอกาสเติบโตก็ต้องวิเคราะห์หากลไกที่ทำให้รอยร้าวเติบโต เช่น เนื่องจากความล้า หรือเนื่องจากการเติบโตที่ได้รับผลจากสภาพแวดล้อม เป็นต้น

เมื่อทราบกลไกที่ทำให้รอยร้าวเติบโตแล้ว ถัดไปก็คือการระบุจำนวนรอบภาระที่โครงสร้างรับ หรือระยะเวลาการใช้งาน นับจากการประเมินครั้งนี้ถึงการประเมินครั้งถัดไป จากนั้นจึงทดสอบหรือสืบค้นหาอัตราการเติบโตของรอยร้าวของวัสดุในสภาวะใช้งาน สุดท้ายคือคำนวณขนาดรอยร้าวหลังจากรับภาระเป็นจำนวนรอบที่ระบุหรือระยะเวลาที่ระบุ (ด้วยวิธีที่อธิบายในหัวข้อที่ 2.5 หรือ 2.6) ความยาวรอยร้าวที่คาดการณ์นี้จะถูกนำระบุลักษณะรอยร้าว และประเมินการคงสภาพซ้ำอีกครั้ง

3.13 การประเมินนัยสำคัญของผลการประเมิน

การวิเคราะห์เริ่มจากการหาตัวประกอบสำรอง (reserve factor) ซึ่งเป็นพารามิเตอร์ที่บอกให้ทราบว่าสถานะของโครงสร้างอยู่ห่างจากสถานะวิกฤติมากน้อยเพียงใด ตัวประกอบสำรองที่กล่าวถึงในระเบียบวิธี R6 ประกอบด้วย ตัวประกอบสำรองของภาระ ตัวประกอบสำรองของขนาดของรอยร้าว ตัวประกอบสำรองของความต้านทานการแตกหัก และตัวประกอบสำรองของความเค้นคราก อย่างไรก็ดีตัวประกอบสำรองของภาระมีความสำคัญมากที่สุด

ความหมายของตัวประกอบสำรองสามารถเข้าใจได้ง่ายขึ้นหากพิจารณาเส้นทางการเปลี่ยนตำแหน่งของจุดประเมินเนื่องจากการแปรค่าภาระ ขนาดรอยร้าว ฯลฯ รูปที่ 3.5 แสดงแผนภาพประเมินความเสียหาย จุด A คือจุดประเมิน เส้นประแสดงเส้นทางการเคลื่อนที่ของจุดประเมินหากตัวแปรต่าง ๆ แปรค่าอย่างที่ระบุในรูป ยกตัวอย่างเช่น ถ้าภาระเพิ่มขึ้นจุด A จะเคลื่อนไปตามแนวเส้น OA จนกระทั่งถึงจุด B ซึ่งเป็นจุดที่ทำนายว่าโครงสร้างเสียหาย

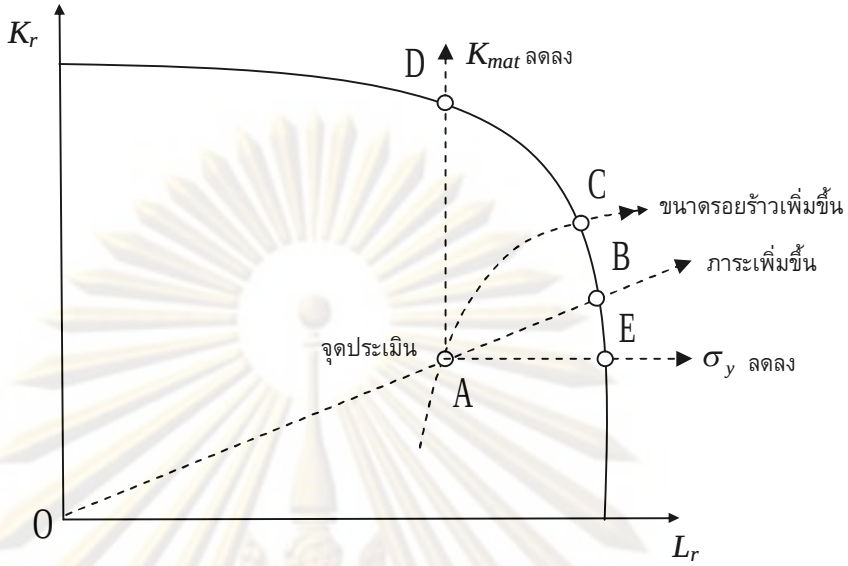
ดังนั้น ตัวประกอบสำรองของภาระ F_L จึงมีนิยามว่า

$$F_L = \frac{P_{critical}}{P_{assess}} \tag{3.27}$$

โดย $P_{critical}$ คือ ภาระที่ทำให้โครงสร้างเสียหาย (ภาระที่ทำให้จุดประเมินเคลื่อนมาอยู่ที่จุด B)
P_{assess} คือ ภาระขณะประเมิน

สำหรับกรณีในรูปนี้นิยาม F_L ในสมการที่ (3.27) จะสมมูลกับอัตราส่วนของระยะ OB ต่อระยะ OA

ในทำนองเดียวกัน ถ้าขนาดรอยร้าวเพิ่มขึ้น ความต้านทานการแตกหัก K_{mat} ลดลง หรือความเค้นครากลดลง จุดประเมินก็จะเคลื่อนที่ไปตัดกับเส้นประเมินความเสียหายที่จุด C, D และ E ตามลำดับ



รูปที่ 3.5 ผลของ ภาระ ขนาดรอยร้าว ความต้านทานการแตกหัก และความเค้นคราก ต่อตำแหน่ง และทิศทางการเคลื่อนที่ของจุดประเมิน

ดังนั้น ตัวประกอบสำรองของขนาดรอยร้าว F_a จึงมีนิยามว่า

$$F_a = \frac{a_{critical}}{a_{assess}} \tag{3.28}$$

โดย $a_{critical}$ คือ ขนาดของรอยร้าววิกฤติ (ขนาดรอยร้าวที่ทำให้จุดประเมินเคลื่อนมาที่จุด C)

a_{assess} คือ ขนาดของรอยร้าวขณะประเมิน

ตัวประกอบสำรองของความต้านทานการแตกหัก F_K จึงมีนิยามว่า

$$F_K = \frac{K_{mat}^{assess}}{K_{mat}^{critical}} \tag{3.29}$$

โดย K_{mat}^{assess} คือ ความต้านทานการแตกหักขณะประเมิน

$K_{mat}^{critical}$ คือ ความต้านทานการแตกหักที่ทำให้โครงสร้างเสียหาย (ความต้านทานการแตกหักที่ทำให้จุดประเมินเคลื่อนมาที่จุด D)

และตัวประกอบสำรองของความเค้นคราก F_σ คำนวณได้จาก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$F_{\sigma} = \frac{\sigma_y^{assess}}{\sigma_y^{critical}} \tag{3.30}$$

โดย σ_y^{assess} คือ ความเค้นคราก ขณะประเมิน

$\sigma_y^{critical}$ คือ ความเค้นครากที่ทำให้โครงสร้างเสียหาย (ความเค้นครากที่ทำให้จุดประเมินเคลื่อนมาที่จุด E)

หลังจากคำนวณตัวประกอบสำรองต่าง ๆ โดยเฉพาะอย่างยิ่ง F_L แล้ว ผู้ประเมินจะต้องวิเคราะห์ผลกระทบของการแปรผันค่าตัวแปรป้อนเข้า ได้แก่ ภาระ ขนาดของรอยร้าว ความต้านทานการแตกหัก ความเค้นคราก มีผลต่อการแปรผันของ F_L มากน้อยเพียงใด ถ้าพบว่า F_L ไว (sensitive) ต่อตัวแปรป้อนเข้าตัวใด ก็จะหาต้องค่าตัวแปรนั้นให้แม่นยำ

3.14 **การประเมินให้ผลลัพธ์ยอมรับได้หรือไม่**

ถ้าผลลัพธ์ที่ได้จากการประเมินออกมาว่าโครงสร้างปลอดภัยการประเมินก็จะสิ้นสุด แต่ถ้าผลออกมาว่าไม่ปลอดภัยแล้ว ระเบียบวิธี R6 แนะนำให้ผู้ประเมินปรับแก้รายละเอียดการประเมินบางขั้นตอนก่อนที่จะสรุปผลอย่างชัดเจน ทางเลือกในการปรับแก้มีทั้งหมด 4 ทางเลือก คือ 1) การประเมินด้วยระดับการประเมินสูงขึ้น 2) การระบุลักษณะรอยร้าวซ้ำอีกครั้ง 3) การเลือก FAD แบบอื่น (หรือ FAD แบบที่สูงขึ้น 4) วิเคราะห์ความเค้นให้แม่นยำขึ้น รายละเอียดของทั้ง 4 หัวข้อจะอธิบายในหัวข้อย่อยต่อไปนี้

3.14.1 การประเมินด้วยระดับการประเมินที่สูงขึ้น

เนื่องจากการประเมินในระดับที่ต่ำกว่าจะให้ผลลัพธ์ในด้านปลอดภัย ซึ่งอาจจะมากไป กล่าวคือโครงสร้างถูกประเมินว่าเสียหายทั้ง ๆ ที่ยังสามารถใช้งานต่อไปได้ การประเมินระดับที่สูงขึ้นจะใช้ค่า K_{mat} ที่พิจารณาผลของการเติบโตอย่างมีเสถียรภาพของรอยร้าว เช่น K_g เนื่องจาก K_{mat} ในกรณีนี้จะมีค่าสูงกว่าการใช้ K_{IC}, K_C, $K_{0.2}$ ดังนั้นจุดประเมินก็จะมีแนวโน้มเคลื่อนต่ำลงในแนวดิ่ง (ดูรูปที่ 3.5 ประกอบ)

3.14.2 การระบุลักษณะรอยร้าวซ้ำอีกครั้ง

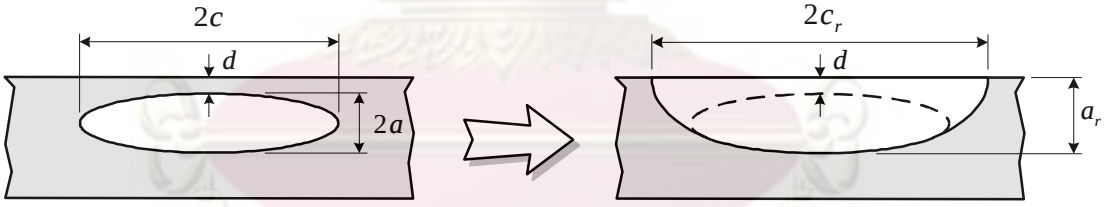
ในกรณีที่ขอบหน้าของรอยร้าวอยู่ใกล้กับผิวอิสระของโครงสร้าง รอยร้าวมีโอกาสจะเติบโต (อย่างไร้เสถียรภาพ) จนทะลุผิวอิสระ และทำให้ชนิดของรอยร้าวเปลี่ยนไป ระเบียบวิธี R6 ชดเชยผลทางพลวัต (dynamic) ต่อการเติบโตของรอยร้าวขณะที่รอยร้าวกำลังทะลุผิวอิสระและเปลี่ยนเป็นรอยร้าวชนิดอื่น โดยการแทนที่รอยร้าวฝังรูปวงรีด้วยรอยร้าวผิวรูปครึ่งวงรี (รูปที่ 3.6) และแทนที่รอยร้าวผิวรูปครึ่งวงรีด้วยรอยร้าวทะลุความหนา (รูปที่ 3.7) นอกจากข้อกำหนดเรื่องชนิดรอยร้าวแล้ว ระเบียบวิธี R6 ยังแนะนำว่าถ้าการแตกหักมีโอกาสเป็นแบบเปราะ ความยาว และความลึกของรอยร้าวผิวรูปครึ่งวงรี (รูปที่ 3.6) จะหาได้จากสมการต่อไปนี้

$$2c_r = \text{ค่ามากกว่าระหว่าง } 2a + 2c + d \text{ และ } 4c \qquad (3.31\ \text{ก})$$

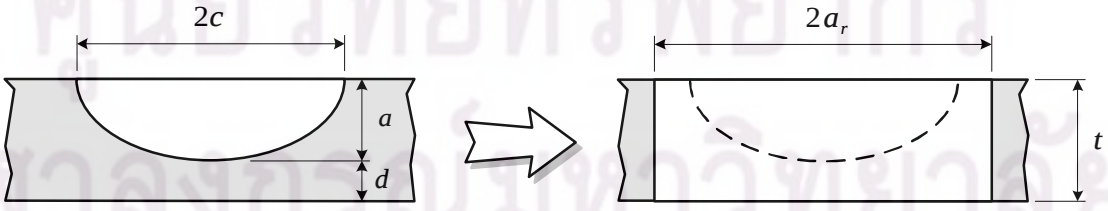
$$a_r = 2a + d \qquad (3.31\ \text{ข})$$

และความยาวของรอยร้าวทะลุความหนา (รูปที่ 3.7) จะหาได้จากสมการต่อไปนี้

$$2a_r = \text{ค่ามากกว่าระหว่าง } a + 2c + d \text{ และ } 4c \qquad (3.32)$$



รูปที่ 3.6 การระบุลักษณะรอยร้าวซ้ำอีกครั้ง ในกรณีรอยร้าวฝังในรูปวงรี



รูปที่ 3.7 การระบุลักษณะรอยร้าวซ้ำอีกครั้ง ในกรณีรอยร้าวผิวรูปครึ่งวงรี

แต่ถ้าหากการแตกหักมีโอกาสเป็นแบบเหนียว ความยาวและความลึกของรอยร้าวผิวรูปครึ่งวงรี (รูปที่ 3.6) จะหาได้จากสมการต่อไปนี้

$$2c_r = 2a + 2c + d \qquad (3.33\ \text{ก})$$

$$a_r = 2a + d \qquad (3.33\ \text{ข})$$

และความยาวของรอยร้าวทะลุความหนา (รูปที่ 3.7) จะหาได้จากสมการต่อไปนี้

$$2a_r = 2a + 2c + d \qquad (3.34)$$

3.14.3 เลือก FAD แบบอื่น

การปรับแก้นี้หมายความว่าให้เลือกใช้ FAD แบบที่สูงขึ้น ยกตัวอย่างเช่น ถ้าเดิมใช้แบบที่ 2 ก็ให้เปลี่ยนไปใช้แบบที่ 3 เนื่องจากระดับความ conservative ของการประเมินจะลดลงเมื่อใช้ FAD เรียงจากแบบที่ 1 ไปจนถึงแบบที่ 3

3.14.4 วิเคราะห์ความเค้นให้แม่นยำขึ้น

การปรับแก้นี้หมายความว่า โครงสร้างมีรูปทรงซับซ้อนหรือรับภาระที่ซับซ้อน การคำนวณความเค้น ณ จุดที่สนใจอาจต้องแทนภาระที่กระจายอยู่บนโครงสร้างด้วยภาระที่สมมูลกันแต่ช่วยให้การคำนวณง่ายขึ้น การทำปัญหาให้ง่ายลงเพื่อให้สามารถใช้สูตรคำนวณความเค้นที่มีทั่ว ๆ ไปได้นั้น อาจทำให้ความแม่นยำของความเค้นที่คำนวณได้ลดลง เช่น ถ้าคำนวณความเค้นได้สูงกว่าที่เกิดขึ้นจริง เมื่อนำไปใช้ในการประเมินก็อาจทำให้ผลการประเมินออกมาว่าโครงสร้างไม่ปลอดภัย ดังนั้นถ้ารูปร่างโครงสร้าง และลักษณะภาระซับซ้อน การประยุกต์วิธีเชิงตัวเลข เช่น ใช้ระเบียบวิธีไฟไนเอลิเมนต์ ก็จะช่วยให้ทำนายความเค้นได้แม่นยำขึ้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

ความรู้เบื้องต้นเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุ

แนวคิดเชิงวัตถุ (object-oriented concept) เริ่มต้นและมีการพัฒนาภาษาสำหรับรองรับการเขียนโปรแกรมเชิงวัตถุ ตั้งแต่ปี ค.ศ. 1967s ภาษาแรกคือ simula-67 จากนั้นราวปี ค.ศ. 1980s ก็มีการพัฒนาภาษา smallTalk และได้รับความนิยมอย่างมาก ในปัจจุบันมีภาษาที่รองรับการเขียนโปรแกรมเชิงวัตถุหลายภาษา เช่น C++ , Java, Delphi เป็นต้น

ในบทนี้จะกล่าวถึงความรู้เบื้องต้นเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุ ประกอบด้วย ข้อจำกัดของแนวคิดเชิงกระบวนการ ข้อดีของแนวคิดเชิงวัตถุ ออบเจ็กต์ (object) และคลาส (class) การจัดกลุ่ม (classification) โครงร่างลำดับชั้นของคลาส (class hierarchy) คุณสมบัติของโปรแกรมเชิงวัตถุ และ design patterns

4.1 ข้อจำกัดของแนวคิดเชิงกระบวนการ

การออกแบบโปรแกรมด้วยแนวคิดเชิงกระบวนการ (procedural concept) จะเริ่มจากการวิเคราะห์ขั้นตอนการทำงานทั้งหมดในการแก้ปัญหา จากนั้นจึงเขียนโปรแกรมย่อยสำหรับการทำงานต่าง ๆ หรือเขียนฟังก์ชันสำหรับการคำนวณต่าง ๆ โปรแกรมย่อยและฟังก์ชันเหล่านี้ก็จะถูกโปรแกรมหลักเรียกใช้ตามลำดับขั้นของการแก้ปัญหา ในการเขียนโปรแกรมแบบนี้จะประกาศ ตัวแปรไว้ในโปรแกรมหลัก ซึ่งทำให้โปรแกรมย่อยสามารถอ่านหรือเปลี่ยนแปลงค่าของตัวแปรเหล่านั้นได้ ด้วยลักษณะเช่นนี้หากโปรแกรมเมอร์เพิ่มเติมโปรแกรมย่อยแล้วการทำงานของโปรแกรมย่อยนี้ไปทำให้ตัวแปรของโปรแกรมหลักมีค่าเปลี่ยนไปโดยไม่ตั้งใจ ก็จะเกิดผลกระทบต่อการทำงานของโปรแกรมย่อยทั้งหมดที่เรียกใช้ตัวแปรนี้ด้วย

4.2 ข้อได้เปรียบของแนวคิดเชิงวัตถุ

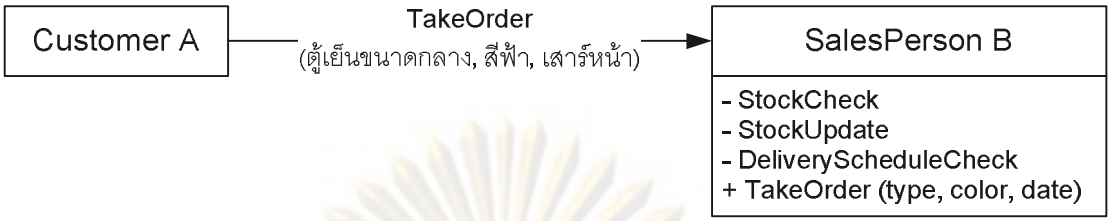
การออกแบบโปรแกรมด้วยแนวคิดเชิงวัตถุ จะเริ่มจากการวิเคราะห์ว่าการหาคำตอบของปัญหาที่สนใจ เกิดจากการสื่อสารอะไรบ้างระหว่างสิ่งที่เกี่ยวข้องกับปัญหา สิ่งต่าง ๆ ที่เกี่ยวข้องกับปัญหาเรียกว่า ออบเจ็กต์ และการสื่อสารระหว่างวัตถุเรียกว่า ข้อความ ด้วยแนวคิดของการใช้ออบเจ็กต์แทนสิ่งต่าง ๆ ในโลกความจริง และใช้การสื่อสารระหว่างออบเจ็กต์แทนกิจกรรมที่เกิด

ขึ้นในโลกความจริง ทำให้การทำงานของโปรแกรมให้สอดคล้องกับกิจกรรมที่เกิดขึ้นในโลกความจริงได้ การพัฒนาและการทำความเข้าใจซอร์สโค้ดจะง่ายกว่า ออบเจ็กต์ที่นิยามขึ้นในระหว่างการทำงานของโปรแกรม จะประกอบด้วยแอตทริบิวต์ (หรือข้อมูล) และเมทธอด (หรือวิธีการหรือฟังก์ชัน) ที่รวมเป็นหน่วยเดียวกัน ซึ่งต่างจากโปรแกรมเชิงกระบวนการที่แยกสองสิ่งนี้จากกัน การรวมทั้งแอตทริบิวต์และเมทธอดเข้าด้วยกันนี้เองที่ทำให้ออบเจ็กต์ในโปรแกรม เทียบเท่ากับวัตถุในโลกความจริง (หรือในปัญหา) ได้ เช่น ดินสอ ก็จะมีแอตทริบิวต์ สีไส้ สีลำตัว ความยาว ฯลฯ และมีเมทธอดคือ ลากเส้น รถยนต์ ก็มี แอตทริบิวต์คือ สี รุ่น ขนาดเครื่องยนต์ จำนวนประตู น้ำหนัก ความเร็ว ฯลฯ และมีเมทธอดคือ เลี้ยว แล่น เบรก ฯลฯ

การรวมแอตทริบิวต์และเมทธอดเข้าด้วยกันเป็นออบเจ็กต์ ยังมีข้อดีอีกประการคือ โปรแกรมเมอร์สามารถเขียนโปรแกรมให้ออบเจ็กต์ไม่สามารถเข้าถึงแอตทริบิวต์ของออบเจ็กต์อื่นได้ ทำให้ลดโอกาสที่แอตทริบิวต์จะถูกเปลี่ยนแปลงโดยไม่ได้ตั้งใจเมื่อมีการปรับปรุงโปรแกรม แม้ว่าการเข้าถึงแอตทริบิวต์ของออบเจ็กต์อื่น (เพื่ออ่านค่า หรือแก้ไขค่า) จะทำไม่ได้โดยตรง แต่ก็สามารถทำโดยเรียกเมทธอดของวัตถุที่เป็นเจ้าของแอตทริบิวต์นั้น

พิจารณาตัวอย่างในโลกความเป็นจริง ได้แก่ กิจกรรมการซื้อของ สมมุติว่ากิจกรรมนี้เกี่ยวข้องกับลูกค้า พนักงานขาย และสินค้า สมมติว่า ลูกค้าต้องการซื้อตู้เย็น หลังจากเลือกตู้เย็นที่ต้องการได้แล้ว ลูกค้าก็จะบอกความต้องการให้กับพนักงานขายทราบ ว่า “ต้องการซื้อตู้เย็นขนาดกลาง สีฟ้า 3 เครื่อง คุณสามารถจัดส่งให้ผมวันเสาร์หน้าได้ไหม” หลังจากรับทราบแล้วพนักงานก็จะต้องไปตรวจสอบสต๊อกว่ามีตู้เย็นรุ่นดังกล่าวเหลือเท่าใด และตรวจสอบตารางการส่งของ ก่อนจะตอบลูกค้าว่าจัดส่งให้ได้หรือไม่ได้

ในส่วนของการเขียนโปรแกรม กิจกรรมข้างต้นสามารถจำลองได้ดังรูปที่ 4.1 จากรูปลูกค้าคือ ออบเจ็กต์ชื่อ Customer A พนักงานคือ ออบเจ็กต์ชื่อ SalesPerson B เมื่อ Customer A ส่งข้อความ TakeOrder ให้กับ SalesPerson B โดยข้อความ TakeOrder ประกอบด้วยพารามิเตอร์ ชนิดของตู้เย็น (คือ ตู้เย็นขนาดกลาง) สีของตู้เย็น (คือ สีฟ้า) วันส่งของ (คือ วันเสาร์หน้า) SalesPerson B ก็จะดำเนินการตรวจสอบสต๊อก และตรวจสอบตารางส่งของ ด้วยเมทธอดที่ทราบเฉพาะตนเอง คือ StockCheck และ DeliverScheduleCheck ก่อนจะส่งผลลัพธ์กลับ ในตัวอย่างนี้ Customer A เรียกใช้เมทธอด TakeOrder ของ SalesPerson B ได้เพราะ SalesPerson B ประกาศเมทธอดนี้เป็นแบบสาธารณะ (public) หรือวัตถุอื่น ๆ สามารถเรียกใช้เมทธอดนี้ได้ ในขณะเดียวกัน Customer A ไม่สามารถเรียกใช้เมทธอด StockCheck และ StockUpdate ได้เพราะ



รูปที่ 4.1 การสื่อสารระหว่าง Customer A และ SalesPerson B

เมทธอดนี้ถูกประกาศเป็นแบบเฉพาะที่ (private) กล่าวอีกอย่างคือ Customer A เข้าไปแก้ไขจำนวนตู้เย็นในสต๊อกด้วยการเรียกใช้เมทธอด StockUpdate ไม่ได้ การที่ออบเจ็กต์สามารถควบคุมการเข้าถึงทั้งแอตทริบิวต์และเมทธอดจากออบเจ็กต์อื่นช่วยให้ส่วนต่าง ๆ ของโปรแกรมเป็นอิสระต่อกันมากขึ้น หรือโอกาสที่การแก้ไขโปรแกรมจะทำให้เกิดผลกระทบในวงกว้างจะลดลง

ในขั้นต้นนี้อาจสรุปได้ว่า แนวคิดเชิงวัตถุนั้นสามารถจำลองโลกความเป็นจริงได้ใกล้เคียง และเมื่อนำแนวคิดนั้นไปเขียนโปรแกรม การทำงานของโปรแกรมก็จะคล้ายคลึงกับสิ่งที่เกิดขึ้นจริง ดังนั้นโปรแกรมจะทำความเข้าใจได้ง่ายกว่า และปรับเปลี่ยนได้ง่ายกว่า

4.3 ออบเจ็กต์และคลาส

ในแนวคิดเชิงวัตถุ คำว่า ออบเจ็กต์ หมายถึง สิ่งต่าง ๆ ในโลกความเป็นจริงที่อยู่รอบตัวเรา ออบเจ็กต์ อาจเป็นรูปธรรม เช่น รถยนต์ โต๊ะ บัญชี ฯลฯ หรือเป็นนามธรรมก็ได้ เช่น บทบาทของคนในองค์กร เป็นต้น

ต่อเนื่องจากตัวอย่างในหัวข้อที่ 4.2 ในความเป็นจริงบริษัทไม่ได้มีลูกค้า customer A (ซึ่งเป็นออบเจ็กต์หนึ่ง) เท่านั้น ในบรรดาลูกค้ารายอื่น ๆ (ซึ่งก็แต่ละรายก็คือออบเจ็กต์แต่ละออบเจ็กต์) จะมีชนิดของแอตทริบิวต์เหมือนกัน คือ ชื่อ (name) ที่อยู่ (address) งบใช้จ่าย (budget) ฯลฯ การรวบรวมออบเจ็กต์ที่มีชนิดแอตทริบิวต์ (และเมทธอด) เหมือนกันเข้ามารวมอยู่ด้วยกันจะทำให้ได้คลาส ยกตัวอย่างในรูปที่ 4.2 จะเห็นว่า Customer A และ Customer C เป็นออบเจ็กต์ของคลาส Customers ในทำนองเดียวกันพนักงานขาย SalesPerson B และ SalesPerson D ก็เป็นออบเจ็กต์ของคลาส SalesPersons เป็นต้น ในทางกลับกันจะเห็นว่าคลาสก็คือ ต้นแบบหรือโครงร่างสำหรับใช้สร้างออบเจ็กต์ ตัวอย่างที่อุปมาอุปมัยเกี่ยวกับคลาสได้ก็คือ แม่พิมพ์ขนม เพราะว่าแม่พิมพ์ขนมคือ ต้นแบบสำหรับการสร้างขนมที่มีชนิดของแอตทริบิวต์ (รูปร่าง ขนาด ฯลฯ) เหมือนกัน ขนมแต่ละชิ้นก็คือออบเจ็กต์แต่ละตัว

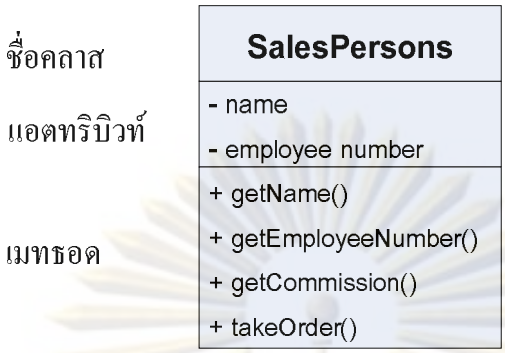
รูปที่ 4.2 คลาส Customers และคลาส SalesPerson

ในโปรแกรมเชิงวัตถุ การทำงานของโปรแกรมคือการสื่อสารระหว่างออบเจ็กต์ ไม่ใช่ระหว่างคลาส ดังนั้นจึงสามารถสร้างออบเจ็กต์หลาย ๆ ตัวจากคลาสเดียวกันในระหว่างการทำงานของโปรแกรมได้ ยกตัวอย่างเช่น ถ้าโปรแกรมต้องการคำนวณงบใช้จ่ายสุทธิของลูกค้าของบริษัท โปรแกรมจะต้องสร้างออบเจ็กต์ Customer A, Customer C ฯลฯ ขึ้นมาจากคลาส Customers ก่อนที่จะอ่านค่างบใช้จ่ายของออบเจ็กต์แต่ละตัวเพื่อนำมารวมกัน เป็นต้น

แม้ว่าออบเจ็กต์ที่สร้างจากคลาสเดียวกันจะมีชนิดของแอตทริบิวต์เหมือนกัน (และมีเมทธอดเหมือนกัน) ก็ตาม แต่ค่าของแอตทริบิวต์ไม่จำเป็นต้องเหมือนกัน ยกตัวอย่างเช่น Customer A และ Customer C มีชนิดแอตทริบิวต์ เช่น ชื่อ เหมือนกัน แต่ออบเจ็กต์ทั้งสองก็มีค่าของแอตทริบิวต์ชื่อไม่เหมือนกัน Customer A อาจมีค่าของแอตทริบิวต์ชื่อ คือ “สมชาย” ขณะที่ Customer B อาจมีค่าของแอตทริบิวต์ชื่อ คือ “สมศักดิ์” เป็นต้น

ขั้นตอนแรกของการออกแบบโปรแกรมเชิงวัตถุ ก็คือการค้นหาคลาสสำหรับใช้สร้างออบ-เจ็กต์ สัญลักษณ์ที่ใช้แทนคลาสคือ รูปสี่เหลี่ยมมุมฉาก แบ่งออกเป็น 3 ส่วน ดังรูปที่ 4.3 ส่วนบนสุดคือ ชื่อของคลาส ส่วนกลางคือ รายการของชนิดแอตทริบิวต์ และส่วนล่างคือ รายการของเมทธอด ข้างหน้าของชนิดแอตทริบิวต์และเมทธอดจะมีเครื่องหมาย “+” หรือ “-” สำหรับบ่งบอกระดับการเข้าถึงแอตทริบิวต์และเมทธอดนั้น เครื่องหมายบวก (“+”) ใช้บอกว่ารายการนั้นเป็นสาธารณะ เครื่องหมายลบ (“-”) ใช้บอกว่ารายการนั้นเป็นเฉพาะตัว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.3 สัญลักษณ์แสดงคลาส (กรณีคลาส SalesPersons)

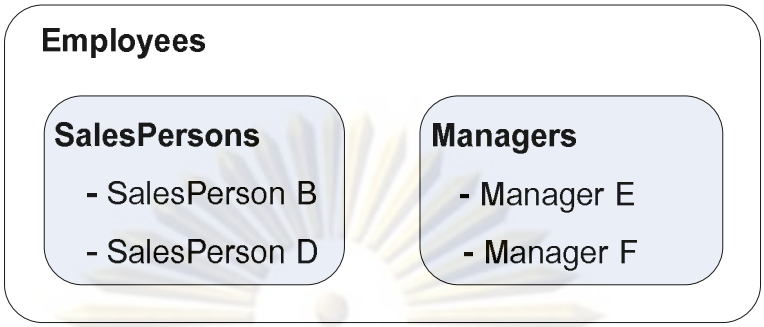
4.4 การจัดกลุ่ม

การจัดกลุ่ม คือ การนำออบเจ็กต์ต่าง ๆ มาคัดแยกเป็นกลุ่ม ๆ โดยออบเจ็กต์ที่อยู่กลุ่มเดียวกันจะมีแอตทริบิวต์และเมทธอดเหมือนกัน ยกตัวอย่างเช่น นกกระจอก นกนางนวล นกอินทรี ก็สามารถจัดอยู่ในกลุ่มของนก หรือคลาสนกได้ เพราะมีชนิดของแอตทริบิวต์ เช่น น้ำหนัก ขนาดปีก ความเร็วในการบิน ฯลฯ และมีเมทธอด เช่น กิน บิน วางไข่ ฯลฯ เหมือนกัน อย่างไรก็ดี เครื่องบินไม่สามารถจัดอยู่ในกลุ่มหรือคลาสนกได้เพราะชนิดของแอตทริบิวต์ไม่เหมือนกับออบเจ็กต์ “นก” แม้ว่าจะมีเมทธอด “บิน” เหมือนกันก็ตาม เป็นต้น

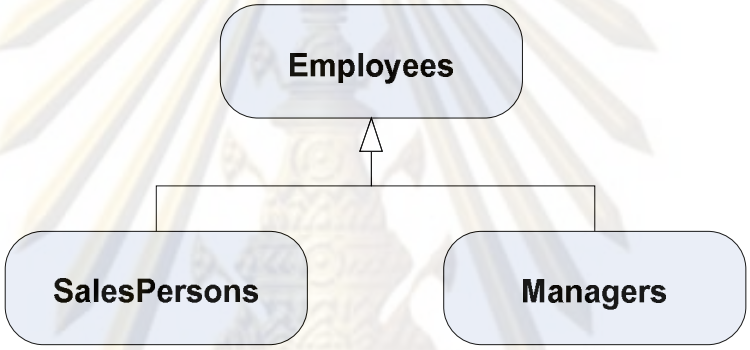
การจัดกลุ่มมีความสำคัญในกระบวนการออกแบบโปรแกรมเชิงวัตถุ เพราะกิจกรรมที่เกิดขึ้นในโลกความจริง (หรือในโลกการทำงานของโปรแกรม) เป็นการสื่อสารระหว่างวัตถุจำนวนมาก หากจัดกลุ่มวัตถุเหล่านั้นไม่เหมาะสม ก็จะได้คลาสที่ไม่มีประสิทธิภาพออกมา ยกตัวอย่างเช่น นักเรียน 50 คนในห้องเรียน และครูผู้สอน ก็สามารถจัดเป็นคลาสได้ 2 คลาสคือ คลาสนักเรียน และคลาสครู เป็นต้น

4.5 ลำดับชั้นของคลาส

บ่อยครั้งที่คลาสมีความสัมพันธ์กัน ในลักษณะของคลาสกลุ่มหนึ่งเป็นเซตย่อย (subset) ของอีกคลาสหนึ่ง ยกตัวอย่างในรูปที่ 4.4 จากรูป คลาส SalePersons และคลาส Managers ต่างก็เป็นเซตย่อยของคลาส Employees เป็นต้น ในตัวอย่างนี้ลำดับชั้นของคลาสจะเขียนได้ดังรูปที่ 4.5



รูปที่ 4.4 ความสัมพันธ์ของคลาส Employee คลาส SalesPerson และคลาส Manager

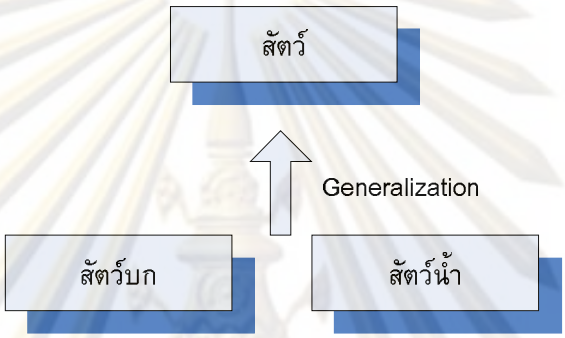


รูปที่ 4.5 โครงร่างลำดับชั้นของคลาส Employees

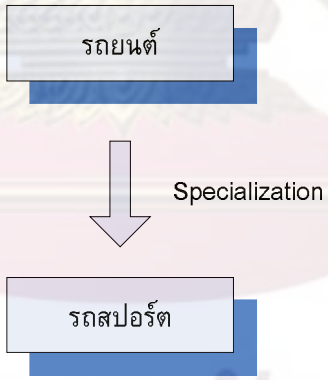
คำศัพท์ที่ใช้เรียกคลาสที่เป็นซุปเปอร์เซ็ตของคลาสอื่น มีหลายชื่อได้แก่ คลาสทั่วไป (general class) หรือซุปเปอร์คลาส (super class) หรือคลาสแม่ ส่วนคลาสที่เป็นเซ็ตย่อยของคลาสอื่น จะเรียกว่า คลาสรอง หรือคลาสย่อย (subclass) หรือคลาสลูก

หากพิจารณาโครงร่างลำดับชั้นของคลาส เช่นตัวอย่างในรูปที่ 4.5 จะเห็นว่าคลาสที่อยู่สูงขึ้นไปจะครอบคลุมคลาสที่อยู่ต่ำกว่า ในอีกมุมมองหนึ่งคลาสที่อยู่ต่ำกว่าจะมีความเฉพาะเจาะจงมากกว่า การสร้างคลาสที่ครอบคลุมคลาสย่อย ๆ (เช่นการสร้างคลาส Employees เพื่อคลุมคลาส SalesPersons และคลาส Managers) เรียกว่า การสร้างลักษณะทั่วไป (generalization) ในทางกลับหากการพิจารณาเริ่มต้นจากคลาสที่กว้าง ๆ หรือทั่ว ๆ ไป การสร้างคลาสที่มีความเฉพาะเจาะจงจากคลาสนั้นจะเรียกว่า การสร้างลักษณะเฉพาะ (specialization) ทั้งการสร้างลักษณะทั่วไป และลักษณะเฉพาะ ช่วยทำให้คลาส (และท้ายที่สุดคือการทำงานของโปรแกรม) เป็นระเบียบ นอกจากนี้ยังสามารถใช้ความสามารถพิเศษของการเขียนโปรแกรมเชิงวัตถุ ซึ่งจะกล่าวในหัวข้อที่ 4.6

รูปที่ 4.6 ถึง 4.8 แสดงตัวอย่างเพิ่มเติมเกี่ยวกับการสร้างลักษณะทั่วไป และลักษณะเฉพาะ ตามลำดับ ในรูปที่ 4.6 คลาสสัตว์บกและคลาสสัตว์น้ำ คือคลาสเฉพาะเจาะจงของคลาสสัตว์ ในรูปที่ 4.7 คลาสรถสปอร์ต คือ คลาสเฉพาะเจาะจงมากกว่าคลาสรถยนต์ ส่วนรูปที่ 4.8 แสดงลำดับชั้นของคลาสบุคคล ซึ่งถ้ามองจากบนลงล่างคือการสร้างลักษณะเฉพาะ แต่ถ้ามองจากล่างขึ้นบนจะเป็นการสร้างลักษณะทั่วไป

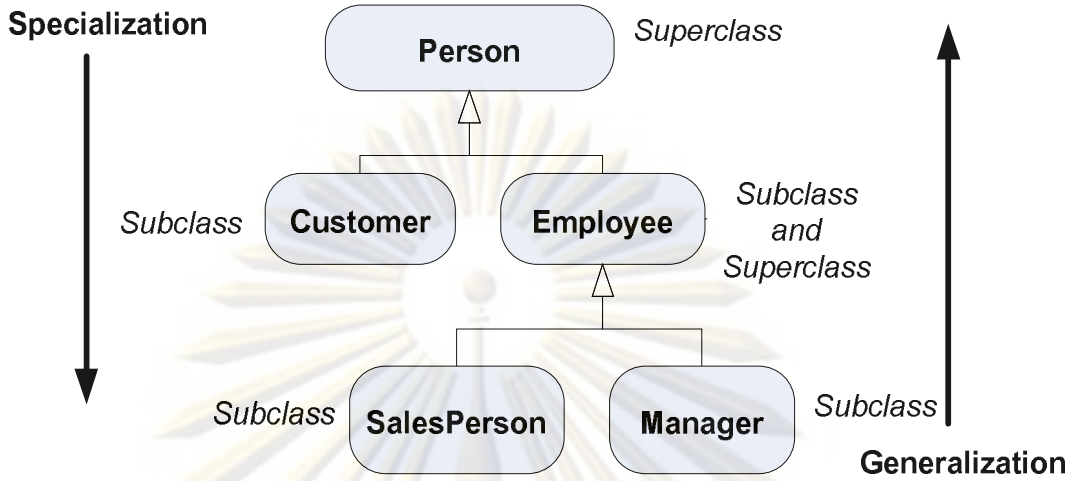


รูปที่ 4.6 การสร้างคลาสทั่วไปจากคลาสที่เฉพาะเจาะจงมากกว่า



รูปที่ 4.7 การสร้างคลาสเฉพาะเจาะจงจากคลาสทั่วไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.8 ลำดับชั้นของคลาสบุคคล

4.6 คุณสมบัติของโปรแกรมเชิงวัตถุ

ในหัวข้อที่ 4.2 ได้กล่าวถึงข้อดีของการออกแบบโปรแกรมด้วยแนวคิดเชิงวัตถุว่า แนวคิดสามารถจำลองโลกความจริงได้ใกล้เคียง เพิ่มความเป็นอิสระต่อกันของส่วนต่าง ๆ ในโปรแกรม ในหัวข้อนี้จะกล่าวถึง ประโยชน์เพิ่มเติมที่จะได้รับจากการเขียนโปรแกรมเชิงวัตถุ ซึ่งประกอบด้วย การควบคุมการเข้าถึง (access control) การสืบทอด (inheritance) และการมีหลายรูปแบบ (polymorphism)

4.6.1 การควบคุมการเข้าถึง

โปรแกรมเชิงวัตถุสามารถควบคุมการเข้าถึงแอตทริบิวต์ และเมทธอดของออบเจกต์ได้ได้ 3 แบบ คือ แบบสาธารณะ แบบเฉพาะที่ และแบบควบคุม (protect) สองแบบแรกได้อธิบายไว้ในหัวข้อที่ 4.2 ส่วนแบบที่ 3 นั้นหมายความว่าแอตทริบิวต์ และเมทธอดสามารถเข้าถึงได้เฉพาะคลาสที่อยู่ในลำดับชั้นเดียวกันเท่านั้น

การประกาศแอตทริบิวต์แบบเข้าถึงได้เฉพาะที่ ช่วยป้องกันไม่ให้การทำงานของออบเจกต์หนึ่งเปลี่ยนแปลงค่าของแอตทริบิวต์ของอีกออบเจ็กต์หนึ่งโดยไม่ได้ตั้งใจ ส่วนการประกาศเมทธอดแบบสาธารณะนั้นช่วยอำนวยความสะดวกกับโปรแกรมเมอร์ ในกรณีต้องการนำคลาสที่

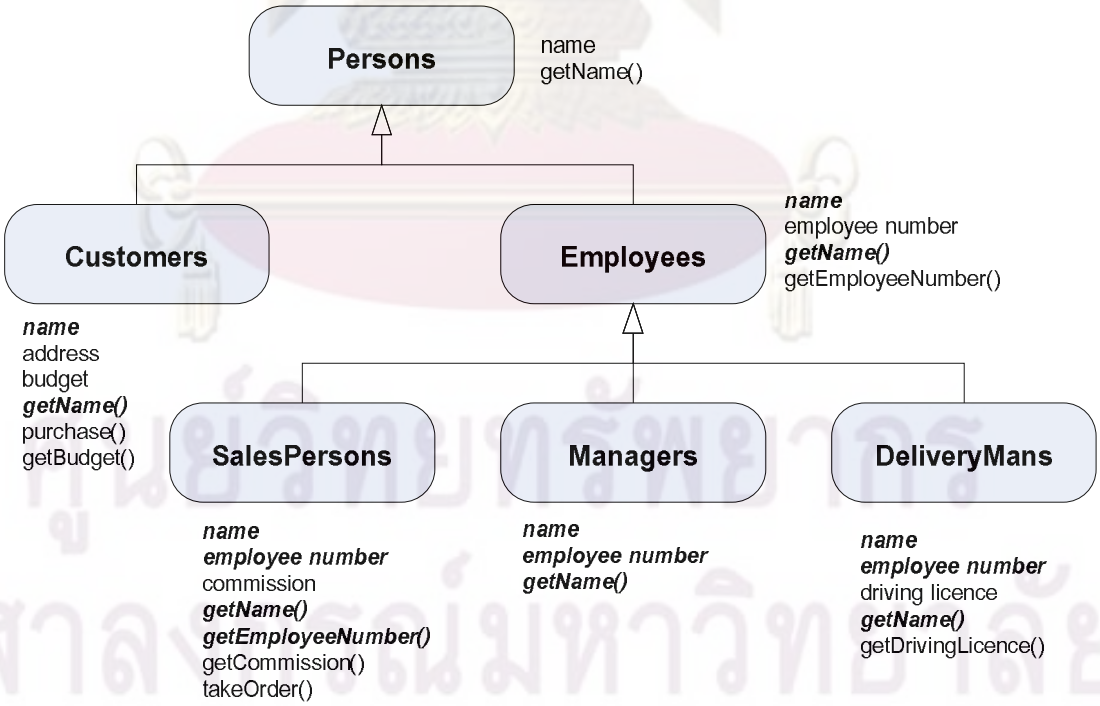
ผู้อื่นเขียนไว้แล้วมาใช้งานร่วมกับโปรแกรมที่ตนพัฒนาอยู่ เพราะโปรแกรมเมอร์จะพิจารณาแค่ว่ามีเมทธอดใดบ้างของคลาสที่สามารถนำมาใช้งานได้ และถ้าต้องการใช้เมทธอดนั้นต้องส่งพารามิเตอร์อะไรบ้าง

4.6.2 การสืบทอด

การสืบทอด หมายถึง การสร้างคลาสใหม่ โดยที่คลาสใหม่สามารถใช้แอตทริบิวต์และเมทธอดของคลาสเดิมได้ นอกจากนี้คลาสที่สืบทอดมาอาจมีแอตทริบิวต์และเมทธอดเพิ่มเติมที่เป็นของตนเองได้ กล่าวอีกอย่างคือ คลาสที่สืบทอดจะมีความเฉพาะเจาะจงมากขึ้น

การสืบทอดช่วยทำให้การเพิ่มความสามารถโปรแกรมทำได้โดยไม่ต้องเขียนโปรแกรมใหม่ทั้งหมด สิ่งที่ต้องทำก็คือใช้สิ่งที่มีอยู่แล้ว และเขียนสิ่งที่ต้องการทำเพิ่มลงไป

รูปที่ 4.9 แสดงลำดับชั้นของคลาส จากรูปนี้ คลาส Employees และคลาส Customers สืบทอดแอตทริบิวต์ name และเมทธอด getName() มาจากคลาส Person (จะเห็นเป็นตัวเข้มเอียง)

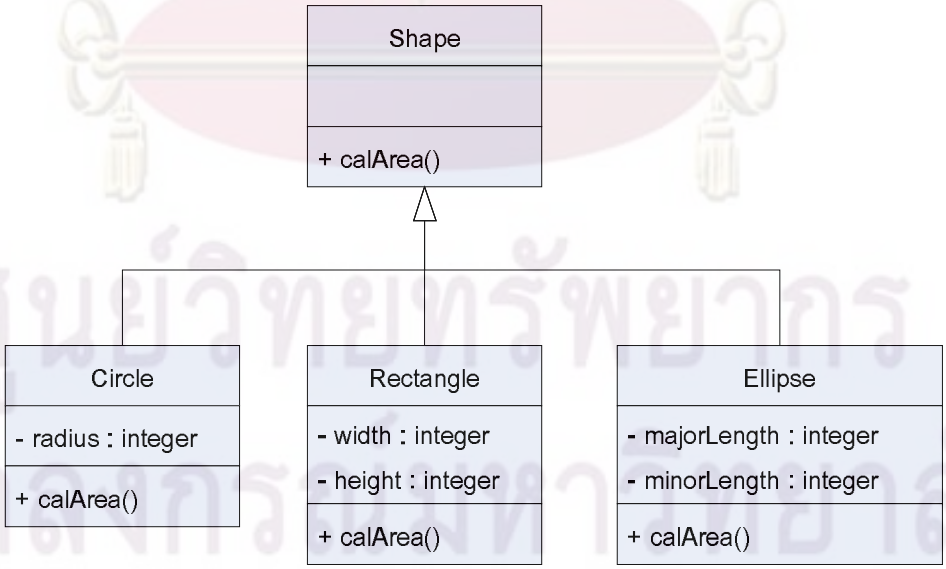


รูปที่ 4.9 การสืบทอดของคลาส

ดังนั้นการเขียนคลาส Customers จึงเพิ่มเติมเฉพาะแอตทริบิวต์ address และ budget และเมทธอด purchase() และ getBudget() เท่านั้น ในทำนองเดียวกันคลาส salesPersons สืบทอดแอตทริบิวต์ name, employee number และเมทธอด getName(), getEmployeeNumber() มาจากคลาส Employees ดังนั้นการเขียนคลาส SalesPersons จึงเพิ่มเติมแต่แอตทริบิวต์ commission และเมทธอด getCommission() และ takeOrder() เท่านั้น ประโยชน์ที่ได้รับจากการไม่ต้องทำซ้ำสิ่งที่มีอยู่ก่อนหน้าแล้วช่วยให้โปรแกรมสั้นลง และเข้าใจง่าย

4.6.3 การมีหลายรูปแบบ

การมีหลายรูปแบบ หมายถึง การที่คลาสแม่ ยินยอมให้คลาสลูกตอบสนองข้อความแบบเดียวกัน ด้วยเมทธอดของคลาสลูกแต่ละคลาส ยกตัวอย่างในรูปที่ 4.10 เช่น คลาส Shape ถูกสืบทอดต่อไปเป็น คลาส Circle คลาส Rectangle และคลาส Ellipse ถ้าสมมุติว่าคลาสแม่มีเมทธอด calArea() สำหรับใช้หาพื้นที่แล้ว คลาสลูกทั้งหมดก็จะมีเมทธอดหาพื้นที่ด้วย อย่างไรก็ดีการคำนวณพื้นที่ของรูปเรขาคณิตทั้งสามชนิดนั้นใช้สูตรต่างกัน การเขียนสูตรคำนวณพื้นที่วงกลม สี่เหลี่ยม และวงรี ในโปรแกรมก็จะเขียนไว้ในเมทธอด calArea()ของคลาส Circle, Rectangle และ Ellipse ในตัวอย่างนี้เมื่อโปรแกรมทำงานหากมีการสร้างออบเจกต์ Circle แล้วออบเจ็กต์นี้ได้รับข้อความ calArea() ออบเจกต์จะคำนวณพื้นที่ด้วยสูตร



รูปที่ 4.10 คลาสไดอะแกรมความสัมพันธ์การสืบทอดของรูปทรงเรขาคณิต

$\pi * radius^2$ แต่ถ้าโปรแกรมสร้างออบเจ็กต์ Rectangle แล้วการตอบสนองต่อข้อความ calArea() จะเป็นการใช้สูตร $width * height$ การตอบสนองที่แตกต่างกันภายใต้ข้อความแบบเดียวกันก็คือคุณสมบัติของการมีหลายรูปแบบ คุณสมบัตินี้ทำให้ออบเจ็กต์เป็นอิสระต่อกัน เพราะว่าออบเจ็กต์จะรับผิดชอบตัวมันเอง

การมีหลายรูปแบบช่วยลดความยุ่งยากในการตั้งชื่อเมทธอด อย่างเช่นกรณีในรูปที่ 4.10 ก็ไม่ต้องตั้งชื่อเมทธอดเป็น calCircleArea(), calRectArea(), calEllipArea() เป็นต้น และอำนวยความสะดวกในการเรียกใช้เมทธอด อย่างเช่นกรณีในรูปที่ 4.10 โปรแกรมหลักจะเรียกใช้เมทธอด calArea() ผ่านออบเจกต์ที่สร้างจากคลาส shape จากนั้นเมื่อผู้ใช้โปรแกรมระบุชัดเจนว่าสนใจรูปเรขาคณิตใด ออบเจกต์ของรูปทรงเรขาคณิตนั้นก็จะถูกสร้างขึ้นแทนที่ออบเจกต์จากคลาส shape และเมทธอด calArea() ที่โปรแกรมหลักเรียกใช้ก็จะหมายถึงเมทธอด calArea() ของออบเจกต์ของรูปทรงเรขาคณิตที่สร้างขึ้น

4.7 Design Pattern

Patterns หมายถึง รูปแบบที่ได้รับการยอมรับว่าถ้าปฏิบัติตามแล้วจะได้ผลลัพธ์อย่างที่ต้องการ การเขียนโปรแกรมเชิงวัตถุก็เช่นกัน มีผู้เชี่ยวชาญที่พยายามจำแนกเหตุการณ์ในโลกความจริงแล้ววิเคราะห์ว่าเหตุการณ์นั้นควรจำลองด้วยคลาสอะไรบ้าง และคลาสเหล่านี้ควรทำงานร่วมกันอย่างไร [16] สำหรับวิทยานิพนธ์นี้มีการนำ Pattern มาใช้เพียงแบบเดียวคือ Factory pattern จึงจะขอกล่าวในรายละเอียดเพียง Pattern นี้เท่านั้น

Factory pattern คือ Pattern ที่จำลองเหตุการณ์การผลิตสินค้าของโรงงาน ดังนี้ โรงงานส่วนใหญ่มีความสามารถในการผลิตสินค้าประเภทหนึ่ง เช่น โรงงานเฟอร์นิเจอร์สามารถผลิตเฟอร์นิเจอร์รูปแบบต่าง ๆ ได้ขึ้นกับการรายการสั่งซื้อจากลูกค้า เป็นต้น ในทำนองเดียวกันการสร้างออบเจ็กต์จากคลาสลูกซึ่งสืบทอดมาจากคลาสแม่เดียวกัน โปรแกรมเมอร์อาจมองว่าคลาสลูกแต่ละคลาสคือคลาสสินค้าแต่ละชนิด การจะสร้างออบเจกต์จากคลาสใดก็ขึ้นกับว่าผู้ใช้โปรแกรมป้อนค่าพารามิเตอร์อะไรให้โปรแกรม รูปที่ 4.11 แสดงคลาส และความสัมพันธ์ระหว่างคลาสเพื่อจำลองเหตุการณ์การผลิตสินค้า ตามแนวคิดของ factory patterns รายละเอียดของคลาสในรูปมีดังนี้

คลาส Product สำหรับแทนสินค้า ซึ่งยังไม่เจาะจงว่าเป็นอะไร (เช่น เฟอร์นิเจอร์)

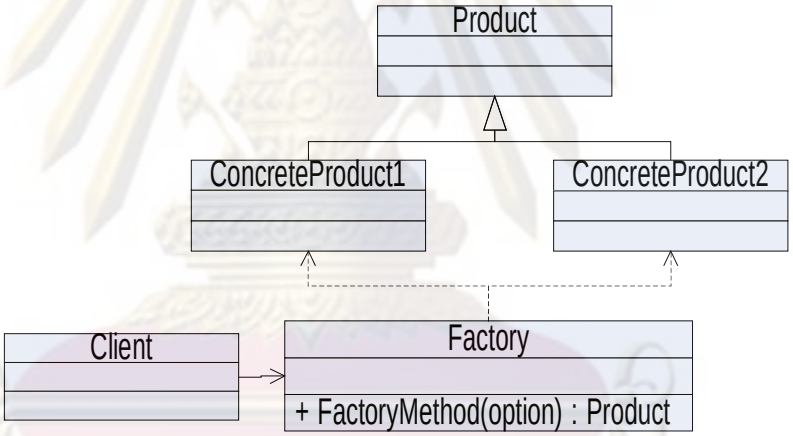
คลาส ConcreteProduct1 สืบทอดจากคลาส Product สำหรับแทนสินค้าชนิดหนึ่งที่โรงงานผลิต (เช่น โต๊ะ)

คลาส ConcreteProduct2 สืบทอดจากคลาส Product สำหรับแทนสินค้าอีกชนิดหนึ่งที่โรงงานผลิต (เช่น เก้าอี้)

คลาส Factory สำหรับแทนโรงงานผลิตสินค้า (กรณีนี้คือ ออบเจ็กต์จากคลาส ConcreteProduct1 หรือจากคลาส ConcreteProduct2)

คลาส Client เป็นตัวแทนของผู้ใช้งานที่เรียกใช้งาน Factory

จากรูปออบเจกต์ของคลาส Client จะส่งข้อความให้ออบเจกต์ของคลาส Factory ทราบว่าต้องการออบเจกต์ออบเจ็กต์ของคลาส ConcreteProduct1 หรือของคลาส ConcreteProduct2



รูปที่ 4.11 คลาสไดอะแกรมสำหรับ Factory pattern

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

การออกแบบโครงสร้างของโปรแกรม และผลการออกแบบ

ในบทนี้จะกล่าวถึงขั้นตอนการออกแบบโครงสร้างของโปรแกรม ตามวิธีที่ Rumbaugh et al. [17] แนะนำ ขั้นตอนการออกแบบแบ่งเป็น 5 ขั้นตอนหลัก คือ 1) การเขียนเนื้อความของปัญหา 2) หาคลาสจากเนื้อความของปัญหา 3) การสร้างคลาสไดอะแกรม 4) การปรับปรุงคลาสไดอะแกรม 5) การสร้าง sequence diagram รายละเอียดของขั้นตอนทั้ง 5 ขั้นจะกล่าวในหัวข้อที่ 5.1 ถึง 5.5 จากนั้นจะกล่าวถึงรายละเอียดของแต่ละคลาสในหัวข้อที่ 5.6 สุดท้ายจะกล่าวถึงโปรแกรมที่ออกแบบเสร็จแล้ว

5.1. เนื้อความของปัญหา

เนื้อความของปัญหามีดังนี้ เมื่อท่อรับภาระใช้งานเป็นระยะเวลาหนึ่งสามารถเสื่อมสภาพและทำให้เกิดรอยร้าว เมื่อหน่วยซ่อมบำรุงตรวจพบรอยร้าวแล้วก็ต้องการทราบว่าท่ออยู่ในสภาพที่ใช้งานต่อไปได้อย่างปลอดภัยหรือไม่ และจะไม่เสียหายก่อนการซ่อมบำรุงครั้งถัดไปหรือไม่ การประเมินว่าโครงสร้างมีสภาพอย่างไรนั้น ผู้ประเมินต้องปฏิบัติตามระเบียบวิธีที่ได้รับการยอมรับ เช่น ระเบียบวิธี R6 ระเบียบวิธี API 579 เป็นต้น สำหรับระเบียบวิธี R6 ขั้นตอนการประเมิน (ตามหัวข้อที่ 3.2) มีดังนี้

- แบ่งประเภทของภาระ และวิเคราะห์ความเค้น
- หาสมบัติแรงดึงของวัสดุ
- เลือกเส้นโค้งประเมินความเสียหาย
- ระบุลักษณะรอยร้าว
- เลือกระดับการประเมิน
- หาค่าความต้านทานการแตกหัก
- คำนวณอัตราส่วนภาระ L_r
- คำนวณอัตราส่วนพารามิเตอร์ K หรือ K_r
- พล็อตจุดประเมิน (L_r, K_r) และบน FAD
- วิเคราะห์การเติบโตของรอยร้าว เนื่องจากจากสภาพแวดล้อมหรือจากภาระล้า

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- พิจารณาผลกระทบต่อตัวประกอบสำรองเนื่องจากการแปรผันค่าของตัวแปรป้อนเข้าต่าง ๆ เช่น ขนาดภาระ ขนาดรอยร้าว เป็นต้น

สภาพความปลอดภัยของโครงสร้างจะพิจารณาจาก F_L หาก F_L มีค่ามากกว่า 1 (ซึ่งหมายความว่าจุดประเมินอยู่ใต้เส้นโค้งประเมินความเสียหาย) จะหมายความว่าโครงสร้างยังปลอดภัย อย่างไรก็ดี รอยร้าวสามารถเติบโตได้หากโครงสร้างยังถูกใช้งานต่อไป ดังนั้นผู้ประเมินจะต้องคาดการณ์ขนาดรอยร้าวในการประเมินครั้งถัดไป จากข้อมูลอัตราการเติบโตของรอยร้าวและข้อมูลสภาวะใช้งานของโครงสร้างนับจากปัจจุบันถึงการประเมินครั้งถัดไป

ระเบียบวิธี R6 ให้ทางเลือกเกี่ยวกับเส้นโค้งประเมินความเสียหาย และระดับการประเมิน เส้นโค้งประเมินความเสียหายแบบที่ 1 นั้นสามารถสร้างได้โดยไม่ต้องอาศัยข้อมูลของโครงสร้างและความสัมพันธ์ระหว่างความเค้น-ความเครียดของวัสดุที่ทำโครงสร้าง ในขณะที่เส้นโค้งประเมินความเสียหายแบบที่ 2 ต้องการข้อมูลทั้งสอง ส่วนการประเมินระดับต่างกันนั้นต่างกันที่ชนิดของความต้านทานการแตกหักที่ใช้สำหรับหา K_r

5.2 การหาคลาสจากเนื้อความของปัญหา

คลาส คือ คำนามในเนื้อความของปัญหา คลาสต่าง ๆ ที่ผู้วิจัยวิเคราะห์จากเนื้อความของปัญหาในหัวข้อที่แล้ว สามารถแบ่งได้ 2 กลุ่ม คือ กลุ่มโครงสร้าง และกลุ่มการคำนวณ ซึ่งแต่ละก็จะมีคลาสย่อย ๆ อีก

ในกลุ่มโครงสร้าง คลาสที่ได้ คือ

- ท่อ
- รอยร้าว
- ภาระ
- สมบัติของวัสดุ

ในกลุ่มการคำนวณ (หรือการประเมิน) ได้คลาส คือ

- FAD
- การระบุลักษณะรอยร้าว
- ภาระขีดจำกัด
- ตัวประกอบความเข้มของความเค้น
- การเติบโตของรอยร้าว
- ความสัมพันธ์ระหว่างความเค้น-ความเครียด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

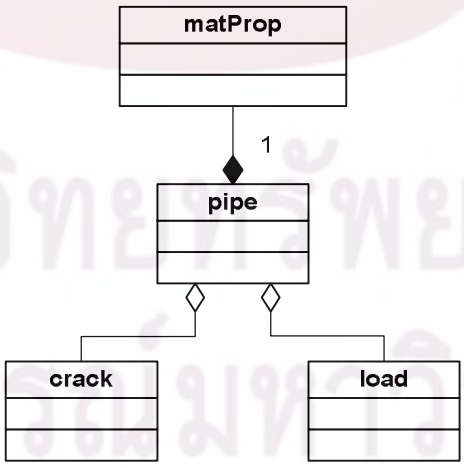
5.3 การสร้างคลาสไดอะแกรม

หลังจากวิเคราะห์หาคลาสได้แล้วจะนำคลาสที่ได้มาสร้างเป็นแผนภาพ เรียกว่า คลาสไดอะแกรม เพื่อแสดงความสัมพันธ์และหน้าที่ของคลาสต่าง ๆ ที่หาได้ ความสัมพันธ์ระหว่างคลาสในกลุ่มโครงสร้างแสดงอยู่ในรูปที่ 5.1 และในกลุ่มการคำนวณแสดงอยู่ในรูปที่ 5.2

จากรูป 5.1 มีความสัมพันธ์ระหว่างคลาสแต่ละคลาสดังนี้ คลาส pipe มีคลาส matProp เป็นส่วนประกอบ ที่จำเป็นและขาดไม่ได้เนื่องจากท่อที่ประเมินทำจากวัสดุเดิมตลอดการประเมิน ความสัมพันธ์ดังกล่าวแสดงด้วยเส้นที่มีหัวเป็นรูปสี่เหลี่ยมข้าวหลามตัดสีดำ ตัวเลขแสดงถึงจำนวนของคลาสที่เป็นส่วนประกอบ กรณีนี้จึงมีจำนวนเท่ากับ 1

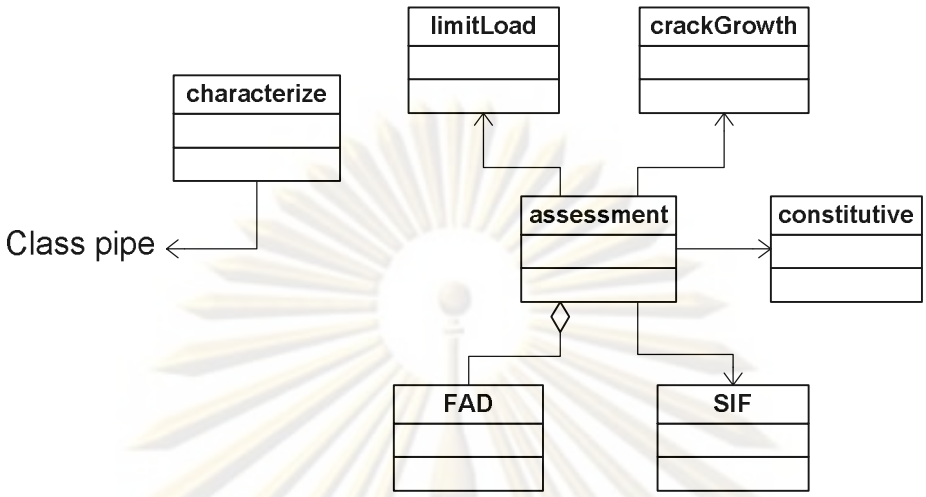
กรณีคลาส crack และคลาส load คลาสทั้งสองต่างเป็นส่วนประกอบของคลาส pipe คลาสทั้งสองเป็นส่วนประกอบที่อาจจะมีหรือไม่มีก็ได้ กล่าวคือ ท่อที่ใช้งานอาจไม่มีรอยร้าว หรือยังไม่มีภาระใด ๆ มากระทำเลยก็ได้ ความสัมพันธ์นี้แสดงได้ด้วยเส้นมีหัวรูปสี่เหลี่ยมข้าวหลามตัดโปร่ง

คลาส characterize ในรูปที่ 5.2 เป็นคลาสในกลุ่มการคำนวณ แทนการระบุลักษณะรอยร้าว กรณีท่อมีรอยร้าวปรากฏอยู่ใกล้กันหลายรอย คลาสมีความสัมพันธ์กับคลาส pipe แบบการกำหนดค่า เนื่องจากรอยร้าวที่ผ่านการระบุลักษณะแล้วจะถูกกำหนดค่าให้กับคลาส crack ผ่านคลาส pipe



รูปที่ 5.1 คลาสไดอะแกรมกลุ่มโครงสร้าง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

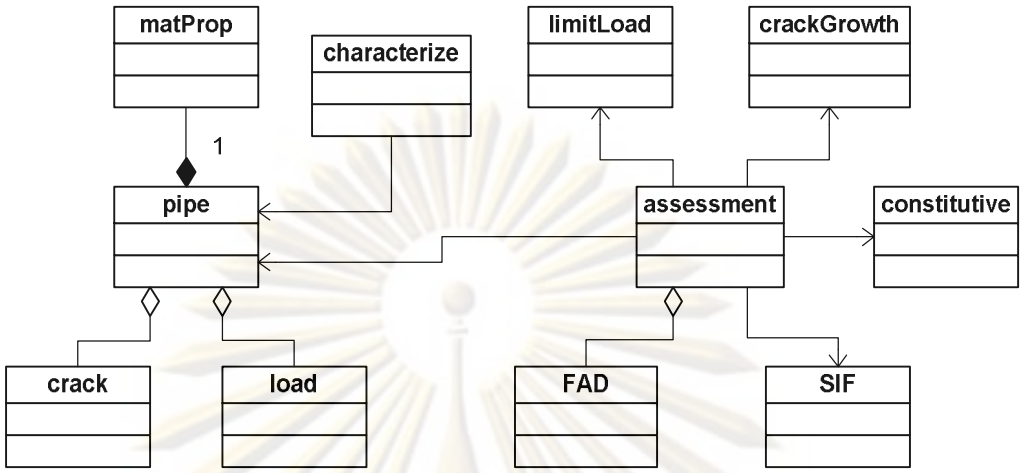


รูปที่ 5.2 คลาสไดอะแกรมกลุ่มการคำนวณ

ความสัมพันธ์ระหว่างคลาสในกลุ่มการคำนวณ คือ คลาส assessment มีคลาส FAD เป็นส่วนประกอบ เนื่องจากการประเมินการคงสภาพของโครงสร้างด้วยระเบียบวิธี R6 ใช้แผนภาพการประเมินความเสียหายเป็นสื่อในการบ่งบอกสถานะของโครงสร้างขณะนั้นว่าใกล้เกิดความเสียหายมากน้อยเพียงใด ความสัมพันธ์นี้แสดงด้วยเส้นมีหัวรูปสี่เหลี่ยมข้าวหลามตัดโปร่ง ผลการประเมินจะแสดงอยู่ในรูปตำแหน่งของจุดการประเมิน จุดดังกล่าวได้จากการคำนวณอัตราส่วนภาระขีดจำกัดและอัตราส่วนพารามิเตอร์ K การคำนวณทั้งสองจึงถูกกำหนดให้เป็นหน้าที่ของคลาส limitLoad และคลาส SIF ตามลำดับ ความสัมพันธ์ระหว่างคลาสทั้งสองกับคลาส assessment เป็นแบบการเรียกใช้ แสดงได้ด้วยเส้นที่มีหัวรูปลูกศร การประเมินยังสามารถประเมินการการเติบโตของรอยร้าวเนื่องจากภาระล้าได้โดยการเรียกใช้จากคลาส crackGrowth ความสัมพันธ์แสดงด้วยเส้นมีหัวรูปลูกศร

คลาส constitutive แทนความสัมพันธ์ระหว่างความเค้น- ความเครียดของวัสดุถูกเรียกใช้จากคลาส assessment เมื่อประเมินโดยใช้แผนภาพการประเมิน FAD ชนิดที่ 2 ความสัมพันธ์แสดงด้วยเส้นมีหัวรูปลูกศร

เมื่อนำคลาสทั้งสองกลุ่มมารวมกันในรูปที่ 5.3 การประเมินเริ่มจากคลาส assessment เรียกใช้คลาส pipe โดยคลาส pipe มีรอยร้าวที่ได้จากการเรียกจากคลาส crack ภาระที่กระทำจากคลาส load และสมบัติวัสดุจากคลาส matProp หลังจากนั้นคลาส assessment เรียกใช้คลาส limitLoad เพื่อคำนวณหาค่าภาระขีดจำกัด คลาส SIF เพื่อคำนวณค่าพารามิเตอร์ K หลังจากนั้นคลาส assessment



รูปที่ 5.3 คลาสไดอะแกรมหลังรวมกลุ่มโครงสร้างและกลุ่มการคำนวณ

คำนวณหาจุดการประเมิน แล้วนำไปหาตัวประกอบสำรองของภาระ โดยการเรียกใช้คลาส FAD หากมีความต้องการประเมินการเติบโตของรอยร้าวแล้วสามารถทำได้โดยการเรียกใช้จากคลาส crackGrowth ในหัวข้อที่ 5.4 จะกล่าวถึงการปรับปรุงคลาสไดอะแกรม

5.4 การปรับปรุงคลาสไดอะแกรม

5.4.1 การปรับปรุงคลาส

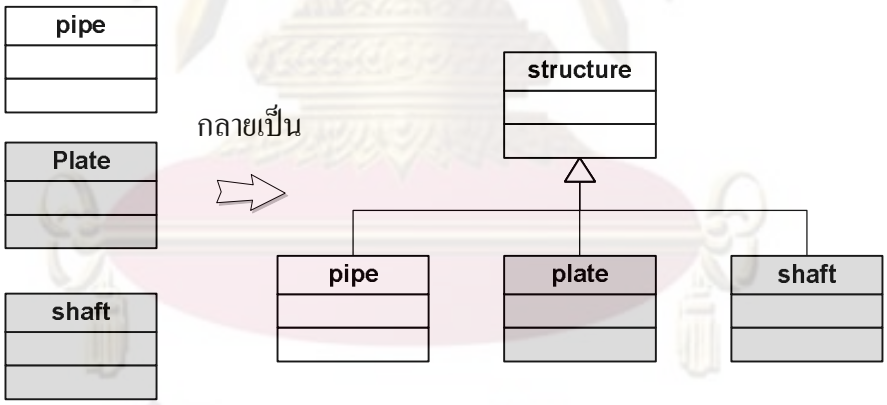
คลาสไดอะแกรมในรูปที่ 5.3 เป็นการแสดงแบบจำลองของการประเมินการคงสภาพของท่อเท่านั้น ในการออกแบบโปรแกรมด้วยแนวคิดเชิงวัตถุ โครงสร้างของโปรแกรมที่ออกแบบต้องสามารถรองรับการปรับปรุงเพิ่มเติมขีดความสามารถของโปรแกรมในอนาคตได้ ดังนั้นคลาสบางคลาสในรูปที่ 5.3 จึงควรแก้ไขให้มีความเฉพาะลดลง ผ่านการสร้างลักษณะทั่วไป ในทางกลับกันคลาสบางคลาสมีความเป็นนามธรรมสูงหรือความเฉพาะน้อยก็สามารถเพิ่มเติมคลาสที่มีความเฉพาะเข้าไปได้ผ่านการสร้างลักษณะเฉพาะ การสร้างลักษณะทั่วไปและลักษณะเฉพาะได้กล่าวถึงแล้วในบทที่ 4 หัวข้อ 4.5 การกระทำดังกล่าวก่อให้เกิดคลาสที่มีการสืบทอดสมบัติจากคลาสหนึ่งไปยังอีกคลาสหนึ่ง ลักษณะเช่นนี้เป็นปัจจัยที่สนับสนุนการนำกลับมาใช้ใหม่ซึ่งเป็นหัวใจของแนวคิดเชิงวัตถุ ส่งผลให้โปรแกรมที่ออกแบบมีความยืดหยุ่นในการเพิ่มเติมสูง จากคลาสไดอะแกรมในรูปที่ 5.3 คลาสที่สามารถนำมาสร้างลักษณะทั่วไปและลักษณะเฉพาะมีดังนี้ คือ

5.4.1.1 การปรับปรุงคลาส pipe

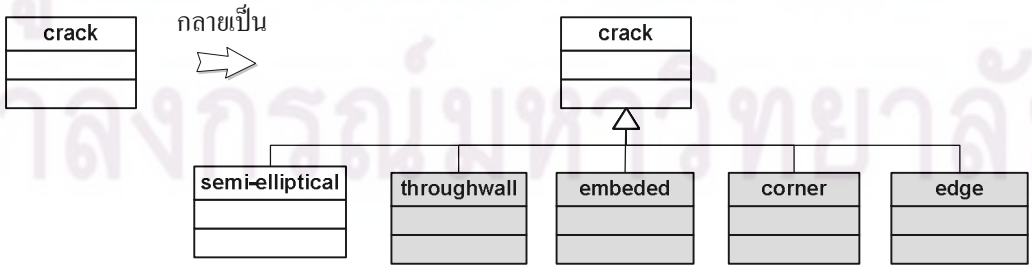
คลาสของโครงสร้างชนิดอื่น ที่นอกเหนือจากคลาส pipe เช่น คลาส plate คลาส shaft คลาสดังกล่าวมีแอตทริบิวต์ คือ ขนาด และเมทธอด คือ สามารถแสดงขนาดได้ คลาสทั้งหมดนี้เป็นคลาสที่มีความเฉพาะสูง ไม่ยืดหยุ่นต่อการเพิ่มเติม การปรับปรุงทำโดยการนำคลาสเหล่านี้มาจัดกลุ่มซึ่งมีการสืบทอดสมบัติมาจากคลาสใหม่ที่มีความเฉพาะน้อยกว่าและมีสมบัติเหมือนกัน คลาสใหม่นี้ คือ คลาส structure ดังรูปที่ 5.4 สำหรับคลาสที่ไม่มีให้ใช้ในวิทยานิพนธ์นี้จะถูกทำเป็นสีเทา

5.4.1.2 การปรับปรุงคลาส crack

คลาส crack เป็นคลาสที่มีความเฉพาะน้อย ด้วยแนวคิดเชิงวัตถุจึงสามารถสืบทอดแอตทริบิวต์และเมทธอดให้กับกลุ่มคลาสรอยร้าวที่มีลักษณะเฉพาะมากกว่าได้ เช่น คลาส semi-elliptical คลาส throughWall คลาส embeded คลาส corner และ คลาส edge ได้ ดังรูปที่ 5.5 สำหรับคลาสที่ไม่มีให้ใช้ในวิทยานิพนธ์นี้จะถูกทำเป็นสีเทา



รูปที่ 5.4 คลาส pipe ที่ถูกปรับปรุงด้วยแนวคิดการสร้างลักษณะทั่วไปและการสืบทอด



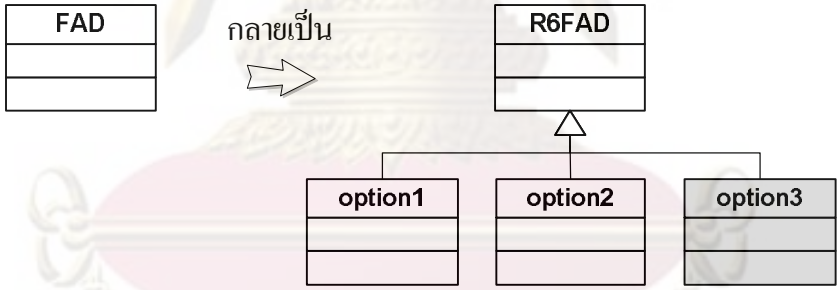
รูปที่ 5.5 คลาส crack ที่ถูกเพิ่มเติมด้วยแนวคิดการสร้างลักษณะเฉพาะและการสืบทอด

5.4.1.3 การปรับปรุงคลาส FAD

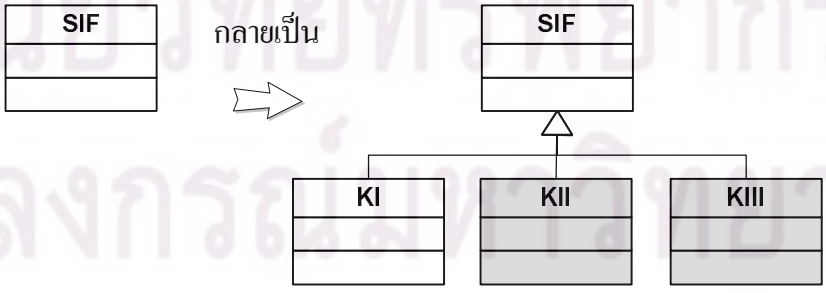
คลาส FAD เป็นคลาสที่มีความเฉพาะน้อย ดังนั้นคลาส FAD จึงสามารถสืบทอดแอตทริบิวต์และเมทธอดให้กับคลาสชนิดของ FAD ที่มีลักษณะเฉพาะมากกว่าได้ เช่น คลาส option1 คลาส option2 และคลาส option3 ได้ ดังรูปที่ 5.6 จากรูปได้มีการเปลี่ยนชื่อคลาสจากคลาส FAD เป็นคลาส R6FAD เพื่อการสื่อความหมายที่ดีขึ้น สำหรับคลาสที่ไม่มีให้ใช้ในวิทยานิพนธ์นี้จะถูกทำเป็นสีเทา

5.4.1.4 การปรับปรุงคลาส SIF

คลาส SIF เป็นคลาสที่มีความเฉพาะน้อย ดังนั้นจึงสามารถสืบทอดแอตทริบิวต์และเมทธอดให้กับกลุ่มคลาสพารามิเตอร์ K ที่มีลักษณะเฉพาะมากกว่าได้ เช่น คลาส K_I คลาส K_{II} และคลาส K_{III} ได้ ดังรูปที่ 5.7 สำหรับคลาสที่ไม่มีให้ใช้ในวิทยานิพนธ์นี้จะถูกทำเป็นสีเทา



รูปที่ 5.6 คลาส FAD ที่ถูกเพิ่มเติมด้วยแนวคิดการสร้างลักษณะเฉพาะและการสืบทอด



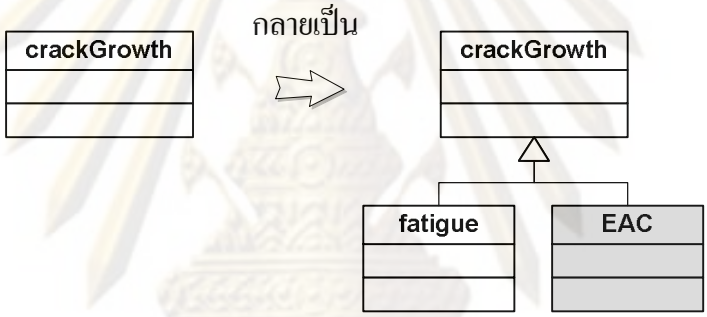
รูปที่ 5.7 คลาส SIF ที่ถูกเพิ่มเติมด้วยแนวคิดการสร้างลักษณะเฉพาะและการสืบทอด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5.4.1.5 การปรับปรุงคลาส crackGrowth

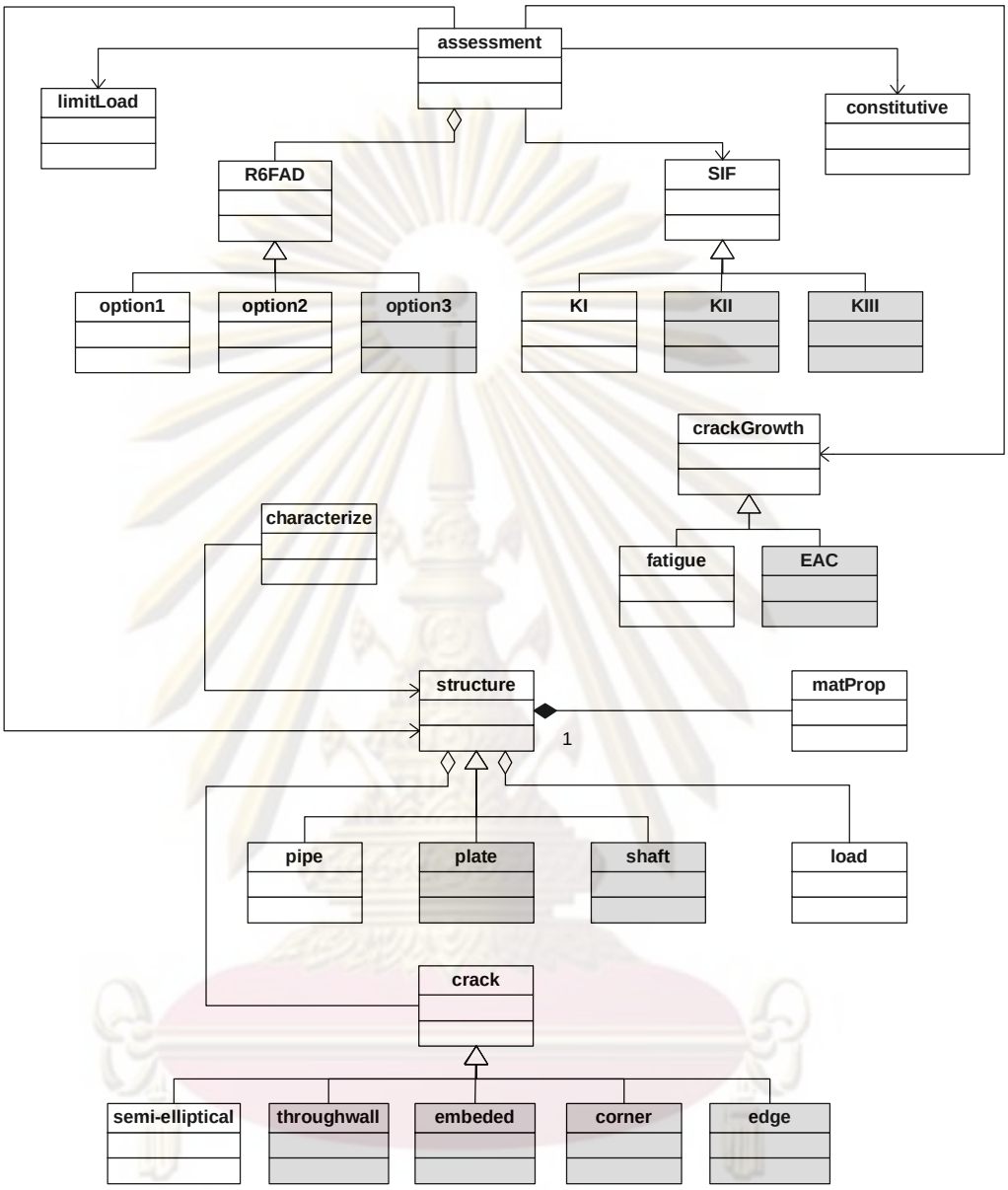
คลาส crackGrowth เป็นคลาสที่มีความเฉพาะน้อย ดังนั้นจึงสามารถสืบทอดแอตทริบิวต์และเมทธอดให้กับกลุ่มคลาสการเติบโตของรอยร้าวที่มีลักษณะเฉพาะมากกว่าได้ เช่น คลาส fatigue และคลาส EAC ได้ ดังรูปที่ 5.8 สำหรับคลาสที่ไม่มีให้ใช้ในวิทยานิพนธ์นี้จะถูกทำเป็นสีเทา

หลังจากเพิ่มขีดความสามารถให้กับคลาสแต่ละคลาสในแบบจำลองแล้ว คลาสไดอะแกรมภายหลังจากการนำคลาสแต่ละคลาสมารวมกัน แสดงได้ดังรูปที่ 5.9



รูปที่ 5.8 คลาส crackGrowth ที่ถูกเพิ่มเติมด้วยแนวคิดการสร้างลักษณะเฉพาะและการสืบทอด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.9 คลาสไดอะแกรมหลังปรับปรุงคลาสด้วยแนวคิดเชิงวัตถุ

5.4.2 การเพิ่มเติมคลาส

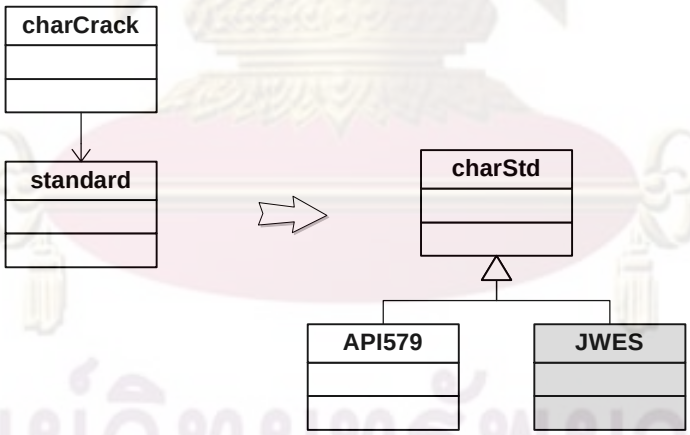
5.4.2.1 การเพิ่มคลาส standard

การระบุลักษณะรอยร้าวด้วยคลาส characterize จะต้องระบุตามข้อแนะนำของมาตรฐาน ซึ่งมาตรฐานการระบุลักษณะมีหลายมาตรฐาน เช่น API 579 และ JWES [18] เป็นต้น ดังนั้นคลาสไดอะแกรมในรูปที่ 5.9 จึงต้องเพิ่มคลาส standard เพื่อใช้เป็นมาตรฐานในการระบุรอย

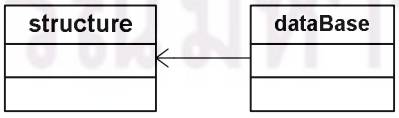
ร้าว คลาส characterize มีความสัมพันธ์กับคลาส standard แบบการเรียกใช้แสดงด้วยเส้นมีหัวรูปลูกศร หากพิจารณาต่อคลาส standard เป็นคลาสที่มีความเฉพาะน้อย ด้วยแนวคิดเชิงวัตถุจึงสามารถสืบทอดแอตทริบิวต์และเมทธอดให้กับกลุ่มคลาสมาตรฐานการระบุลักษณะรอยร้าวที่มีลักษณะเฉพาะมากกว่าได้ เช่น คลาส API 579 และคลาส JWES เป็นต้น ดังรูปที่ 5.10 ในรูปมีการเปลี่ยนชื่อคลาสจากคลาส characterizeไปเป็นคลาส charCrack และจากคลาส standard ไปเป็นคลาส charStd เพื่อความกระชับของชื่อคลาส

5.4.2.2 การเพิ่มคลาส dataBase

วัสดุมีหลายชนิด หากโปรแกรมมีข้อมูลสมบัติของวัสดุเก็บไว้ให้เลือกใช้ก็จะเพิ่มความสะดวกให้กับผู้ใช้มากยิ่งขึ้นนอกจากนั้นยังลดความผิดพลาดจากการกำหนดค่าสมบัติของวัสดุที่ไม่ถูกต้องด้วย ดังนั้น คลาสไดอะแกรมในรูปที่ 5.9 จึงต้องเพิ่มคลาส dataBase ลงไปโดยคลาส database มีความสัมพันธ์กับคลาส structure แบบกำหนดค่า ความสัมพันธ์แสดงด้วยเส้นมีหัวรูปลูกศร ดังรูปที่ 5.11



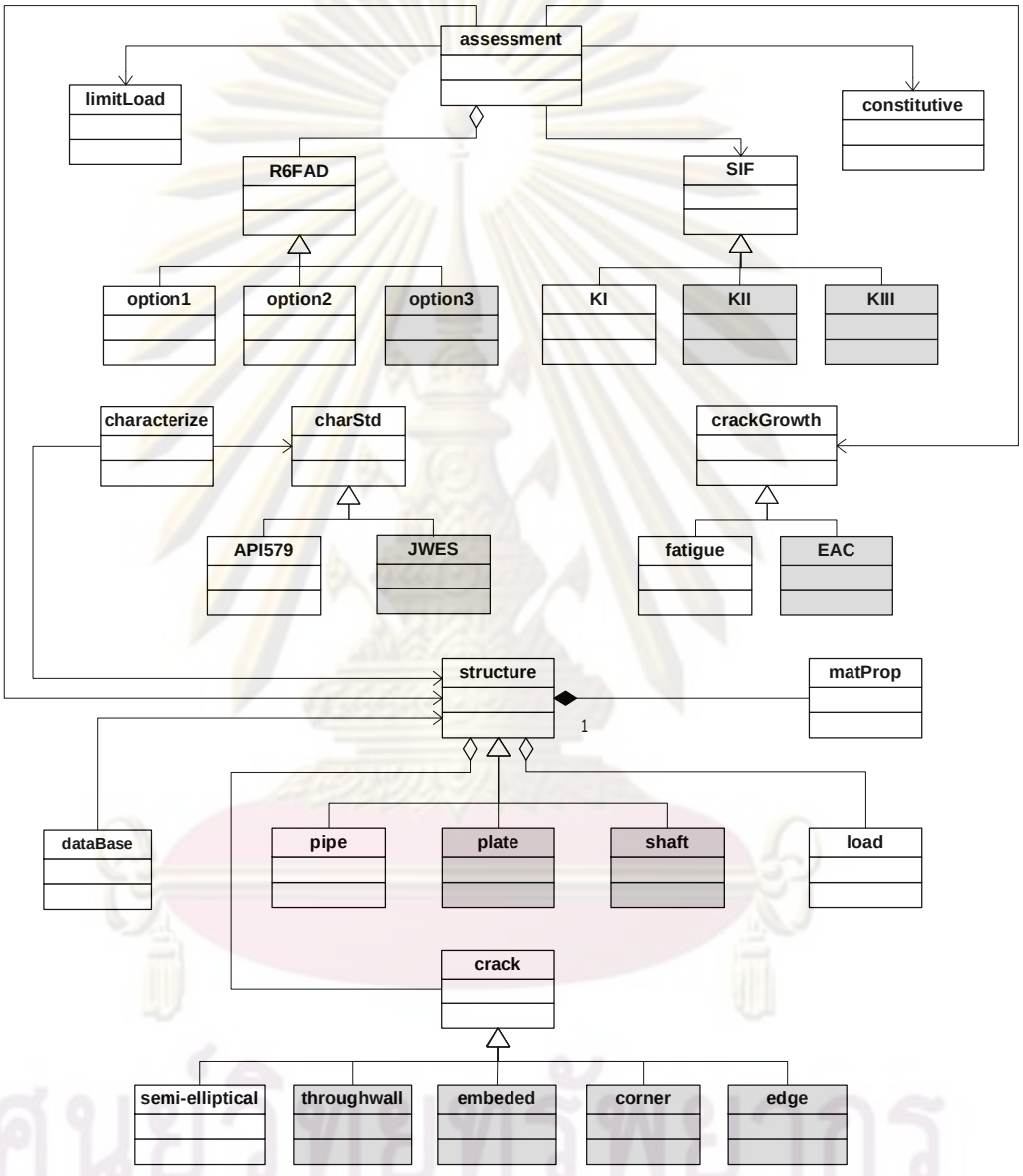
รูปที่ 5.10 คลาส standard ที่ถูกเพิ่มเติมด้วยแนวคิดการสร้างลักษณะเฉพาะและการสืบทอด



รูปที่ 5.11 คลาส dataBase ที่ถูกเพิ่มเติม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หลังจากปรับปรุงคลาสในคลาสไดอะแกรมและเพิ่มเติมคลาสที่เกี่ยวข้องแล้ว คลาสไดอะแกรมที่ได้จะแสดงอยู่ในรูปที่ 5.12



รูปที่ 5.12 คลาสไดอะแกรมหลังปรับปรุงด้วยสมบัติของแนวคิดเชิงวัตถุและเพิ่มเติมคลาสที่เกี่ยวข้องแล้ว

นอกจากนี้แล้วการออกแบบโปรแกรมยังจำเป็นต้องคำนึงผู้ใช้งานด้วย การออกแบบในส่วนนี้ คือ การออกแบบในส่วนติดต่อกับผู้ใช้งาน มีดังนี้

5.4.2.3 การเพิ่มคลาสในส่วนติดต่อกับผู้ใช้งาน

คลาสในส่วนติดต่อกับผู้ใช้งานมีไว้เพื่อเป็นสื่อกลางในการรับค่าที่เกี่ยวข้องกับการประเมินการคงสภาพจากผู้ใช้งาน คลาสในส่วนติดต่อกับผู้ใช้งาน ได้แก่

1) คลาส frmUserIn (รายละเอียดอยู่ในหัวข้อที่ 5.7.1) นอกจากจะใช้รับค่าจากผู้แล้วคลาส frmUserIn ยังใช้สร้างออบเจ็กต์ pipe และออบเจ็กต์ assessment อีกด้วย การสร้างออบเจ็กต์ pipe เลือกใช้ factory pattern (รายละเอียดอธิบายอยู่ในบทที่ 4 หัวข้อที่ 4.7)

2) คลาส frmPipe มีไว้เพื่อรับค่าขนาดของท่อจากผู้ใช้งาน

3) คลาส frmCrack มีไว้เพื่อรับค่าขนาดของรอยร้าวจากผู้ใช้งานและระบุลักษณะของรอยร้าวกรณีท่อมีรอยร้าวสองรอยอยู่ใกล้กัน

4) คลาส frmMatProp มีไว้เพื่อรับค่าสมบัติของวัสดุจากผู้ใช้งานและติดต่อกับฐานข้อมูลเพื่อให้ผู้ใช้สามารถเลือกสมบัติของวัสดุจากฐานข้อมูล

5) คลาส frmCrackGrowth มีไว้เพื่อรับค่าข้อมูลการเติบโตของรอยร้าว กรณีผู้ใช้ต้องการทราบว่าโครงสร้างจะเสียหายก่อนการประเมินครั้งถัดไปหรือไม่

คลาสไดอะแกรมทั้งหมดของปัญหาโดยรวมคลาสส่วนติดต่อกับผู้ใช้งานแล้วแสดงอยู่ในภาคผนวก ก.2

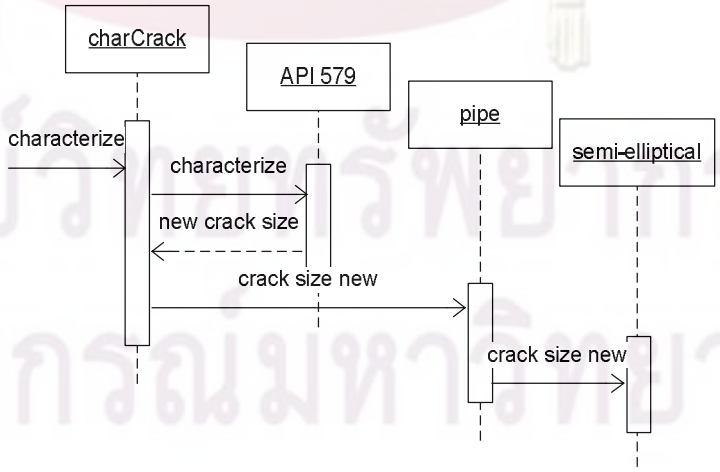
5.5 sequence diagram

คลาสไดอะแกรมในรูปที่ 5.12 ถูกนำมาสร้างเป็นแผนภาพแสดงลำดับขั้นตอนการทำงานของออบเจ็กต์ในแบบจำลอง เรียกแผนภาพดังกล่าวว่า sequence diagram ข้อดีของ sequence diagram คือ สามารถบอกได้ว่าออบเจ็กต์ถูกสร้างและใช้งานเมื่อใดตามลำดับ โดยแบ่งเป็น 3 ส่วน ได้แก่ sequence diagram ของการระบุลักษณะรอยร้าว sequence diagram ของการประเมิน และ sequence diagram ของการเติบโตของรอยร้าวล้า

การประเมินการคงสภาพของท่อหากตรวจพบรอยร้าวหลายรอย ก่อนทำการประเมินต้องระบุรอยร้าวเหล่านั้นให้เหลือรอยเดียว ลำดับขั้นตอนการทำงานของแบบจำลองในส่วนการระบุรอยร้าวแสดงอยู่ในรูปที่ 5.13 ในรูปประกอบด้วยออบเจ็กต์กลุ่มของโครงสร้างคือ ออบเจ็กต์ pipe ที่ถูกสร้างมาจากคลาส pipe ออบเจ็กต์ semi-elliptical ที่ถูกสร้างมาจากคลาส crack ออบเจ็กต์ charCrack ที่ถูกสร้างจากผู้ใช้ (Client) และสุดท้าย คือ ออบเจ็กต์ API579 ที่ถูกสร้างมาจากคลาส charStd การทำงานเรียงลำดับจากด้านบนลงด้านล่างความสัมพันธ์แสดงด้วยเส้นมีหัวรูปลูกศร โดยเส้นทึบแสดงการเรียกใช้ออบเจ็กต์ และเส้นประแสดงการคืนค่าของออบเจ็กต์

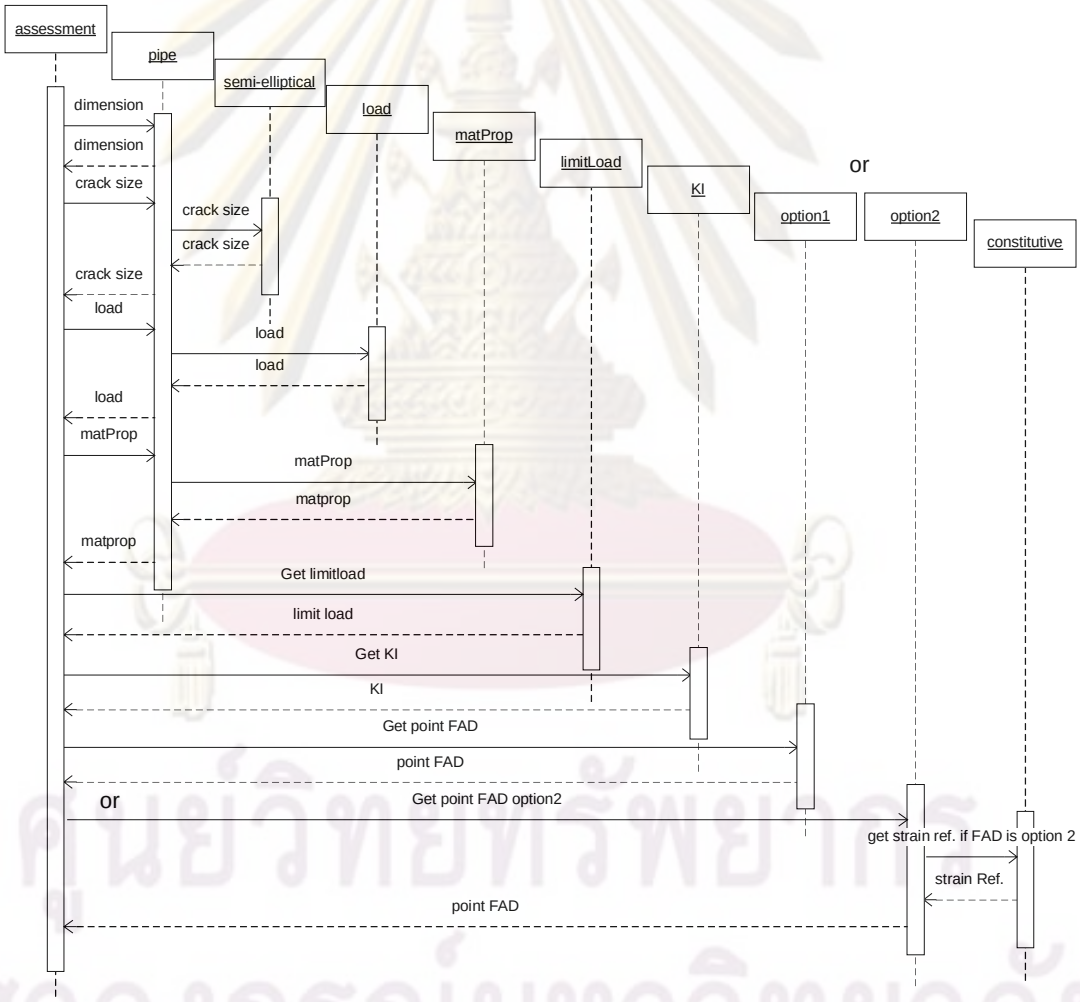
จากรูป ที่ 5.13 การระบุลักษณะของรอยร้าวเริ่มจาก (sequence diagram กรณีรวมออบเจ็กต์ในส่วนการติดต่อกับผู้ใช้แล้วแสดงอยู่ในภาคผนวก ก.3)ออบเจ็กต์ charCrack นำข้อมูลของรอยร้าวหลายรอยส่งให้ออบเจ็กต์ API 579 เพื่อระบุลักษณะของรอยร้าวเหล่านั้น หลังจากออบเจ็กต์ API 579 ระบุรอยร้าวแล้ว (รอยร้าวจะเหลือรอยเดียว) ออบเจ็กต์ API 579 จะส่งคืนกลับมาให้ออบเจ็กต์ charCrack ก่อนที่ออบเจ็กต์ charCrack จะส่งต่อไปยังออบเจ็กต์ pipe หลังจากนั้นออบเจ็กต์ pipe จะนำไปเก็บเป็นข้อมูลของออบเจ็กต์ semi-elliptical การกำหนดค่าข้อมูลรอยร้าวหรือการเรียกใช้ข้อมูลของรอยร้าวสาเหตุที่ต้องทำผ่านออบเจ็กต์ pipe เนื่องจากการออกแบบคลาสไดอะแกรมในหัวข้อที่ 5.3 ที่ผู้ออกแบบได้กำหนดให้คลาส crack เป็นส่วนประกอบของคลาส pipe ดังนั้นเมื่อมีการเรียกใช้ขนาดของรอยร้าวจึงต้องเรียกผ่านคลาส Pipe เสมอ

หลังจากระบุรอยร้าวแล้วต่อไปจะเป็นลำดับขั้นตอนการทำงานของออบเจ็กต์ในกลุ่มการคำนวณ แสดงอยู่ในรูปที่ 5.14 ในรูปประกอบด้วยออบเจ็กต์ในกลุ่มโครงสร้าง คือ ออบเจ็กต์ pipe ออบเจ็กต์ semi-ellitical ออบเจ็กต์ load และออบเจ็กต์ matProp ออบเจ็กต์ทั้งหมดถูกสร้างจากคลาสในส่วนของโครงสร้างในรูปที่ 5.1 ยกเว้นออบเจ็กต์ pipe ที่ถูกสร้างจากจากผู้ใช้ ส่วนออบเจ็กต์ในกลุ่มการคำนวณ คือ ออบเจ็กต์ assessment ออบเจ็กต์ limitLoad ออบเจ็กต์ KI และออบเจ็กต์ option1 (หรือ option2 ขึ้นอยู่กับการเลือก FAD) ออบเจ็กต์ทั้งหมดถูกสร้างจากคลาสในส่วนของการประเมินในรูปที่ 5.2 ยกเว้นออบเจ็กต์ assessment ที่ถูกสร้างจากจากผู้ใช้ การทำงานร่วมกันตามลำดับแสดงด้วยเส้นมีหัวรูปลูกศร รายละเอียดของลูกศรได้กล่าวถึงแล้วในย่อหน้าที่สองของหัวข้อนี้



รูปที่ 5.13 sequence diagram ของการระบุลักษณะรอยร้าว

การประเมินเริ่มจากออบเจ็กต์ assessment (sequence diagram กรณีรวมออบเจ็กต์ในส่วนการติดต่อกับผู้ใช้แล้วแสดงอยู่ในภาคผนวก ก.4) เรียกหาข้อมูลที่ใช้ในการประเมินจากออบเจ็กต์ในกลุ่มโครงสร้าง คือ มิติของโครงสร้างจากออบเจ็กต์ pipe ขนาดของรอยร้าวจากออบเจ็กต์ semi-elliptical ขนาดของภาระจากออบเจ็กต์ load และสมบัติวัสดุจากออบเจ็กต์ matProp การเรียกหาขนาดของรอยร้าว ขนาดของภาระและสมบัติของวัสดุนั้นออบเจ็กต์ assessment เรียกผ่านออบเจ็กต์ pipe เนื่องจากคลาสของออบเจ็กต์ทั้งสามได้ถูกออกแบบให้เป็นส่วนประกอบของคลาส structure ดังนั้น เมื่อมีความต้องการข้อมูลจากออบเจ็กต์ของคลาสทั้งสามนี้ จึงจำเป็นต้องเรียกผ่านออบเจ็กต์ pipe เสมอ

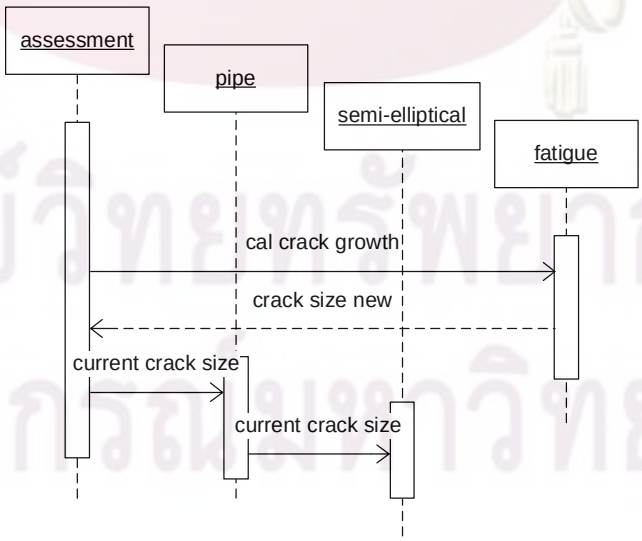


รูปที่ 5.14 sequence diagram ของการประเมินการคงสภาพ

หลังจากออบเจ็กต์ assessment ได้รับการคืนค่าข้อมูลจากออบเจ็กต์ pipe แล้ว การประเมินทำโดย ออบเจ็กต์ assessment เรียกใช้การคำนวณภาระขีดจำกัดจากออบเจ็กต์ limitLoad และเรียกใช้

การคำนวณค่าพารามิเตอร์ K จากออบเจ็กต์ K_I เพื่อใช้ในการหาจุดการประเมิน (L_r, K_r) หลังจากได้จุดการประเมินแล้วจึงนำไปหาตัวประกอบสำรอง การหาตัวประกอบสำรองจำเป็นต้องใช้ตำแหน่งของจุดบนเส้นโค้งประเมินความเสียหายของแผนภาพ FADโดยการเรียกใช้ออบเจ็กต์ option1 หรือ option2 ในการคำนวณขึ้นอยู่กับความต้องการของผู้ประเมิน หากผู้ประเมินเลือก FAD option2 ในการประเมิน ออบเจ็กต์ option2 จะเรียกใช้ออบเจ็กต์ constitutive เพื่อคำนวณความเครียดอ้างอิง (reference strain, $\varepsilon_{ref.}$) และส่งกลับมาให้ออบเจ็กต์ option2 คำนวณจุดบนเส้นโค้งประเมินความเสียหายและส่งกลับให้ออบเจ็กต์ assessment เพื่อคำนวณหาตัวประกอบสำรองต่อไป หากมีความต้องการประเมินการเติบโตของรอยร้าวเนื่องจากความล้าหลังจากรับภาระเป็นจำนวนรอบที่กำหนด ในรูปที่ 5.15 แสดง sequence diagram ของการประเมินการเติบโตของรอยร้าวล้า

การคำนวณการเติบโตของรอยร้าวล้าใน sequence diagram ในรูปที่ 5.15 เริ่มหลังจากการคำนวณจุดประเมินแล้วโดย ออบเจ็กต์ assessment เรียกใช้การคำนวณขนาดของรอยร้าวหลังจากรับภาระล้าจากจำนวนรอบที่กำหนดจากออบเจ็กต์ fatigue หลังจากออบเจ็กต์ fatigue คำนวณหาขนาดของรอยร้าวแล้วคืนค่าขนาดของรอยร้าวใหม่มายังออบเจ็กต์ assessment ออบเจ็กต์ assessment จะนำขนาดของรอยร้าวใหม่ส่งให้กับออบเจ็กต์ pipe เพื่อนำไปเก็บเป็นข้อมูลของรอยร้าวหลังเกิดการเติบโตในออบเจ็กต์ semi-elliptical เพื่อใช้ในการประเมินใหม่อีกรอบตามขั้นตอนในรูปที่ 5.14



รูปที่ 5.15 sequence diagram ของการประเมินการเติบโตของรอยร้าว

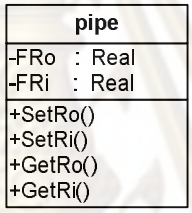
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5.6 รายละเอียดของคลาสและแต่ละคลาส

ในหัวข้อนี้จะวิเคราะห์หาแอตทริบิวต์และเมทธอดของคลาสต่าง ๆ ในหัวข้อที่ 5.4 เพื่อให้สามารถนำไปเขียนโปรแกรมได้สะดวก รายละเอียดทั้งหมดจะแสดงอยู่ในภาคผนวก ก.

5.6.1 รายละเอียดของคลาส pipe

คลาส pipe ภายหลังเพิ่มแอตทริบิวต์และเมทธอดแล้วแสดงอยู่ในรูปที่ 5.16 ส่วนรายละเอียดของแอตทริบิวต์จะอธิบายอยู่ในตารางที่ 5.1 และรายละเอียดของเมทธอดจะอธิบายอยู่ในตารางที่ 5.2



รูปที่ 5.16 แอตทริบิวต์และเมทธอดของคลาส pipe

ตารางที่ 5.1 แอตทริบิวต์ของคลาส pipe และคำอธิบาย

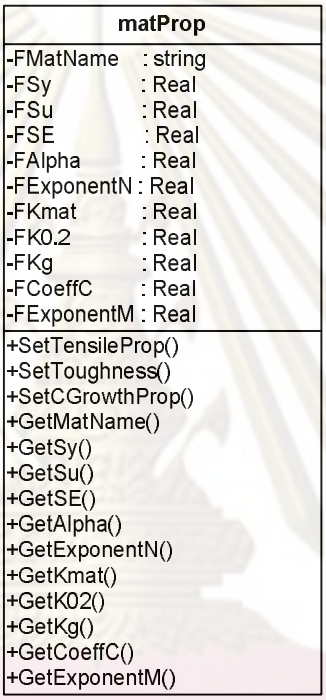
แอตทริบิวต์	คำอธิบาย	หน่วย	หมายเหตุ
FRo	รัศมีภายนอก	*m*	-
FRi	รัศมีภายใน	*m*	-

ตารางที่ 5.2 เมทธอดของคลาส pipe และคำอธิบาย

เมทธอด	คำอธิบาย	หมายเหตุ
SetRo	สำหรับกำหนดค่ารัศมีภายนอกให้กับตัวแปร FRo	-
SetRi	สำหรับกำหนดค่ารัศมีภายในให้กับตัวแปร FRi	-
GetRo	สำหรับอ่านค่ารัศมีภายนอก FRo (หรืออ่านค่าตัวแปร FRo)	-
GetRi	สำหรับอ่านค่ารัศมีภายใน FRi (หรืออ่านค่าตัวแปร FRi)	-

5.6.2 รายละเอียดของคลาส matProp

คลาส matProp ภายหลังเพิ่มแอตทริบิวต์และเมทธอดแล้วแสดงอยู่ในรูปที่ 5.17 ส่วนรายละเอียดของแอตทริบิวต์จะอธิบายอยู่ในตารางที่ 5.3 และรายละเอียดของเมทธอดจะอธิบายอยู่ในตารางที่ 5.4



รูปที่ 5.17 แอตทริบิวต์และเมทธอดของคลาส matProp

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 5.3 แอตทริบิวต์ของคลาส matProp และคำอธิบาย

แอตทริบิวต์	คำอธิบาย	หน่วย	หมายเหตุ
FMatName	ชื่อวัสดุ	-	-
FSy	ความเค้นคราก	MPa	-
FSu	ความเค้นดึงสูงสุด	MPa	-
FSE	มอดูลัสของความยืดหยุ่น	MPa	-
FAlpha	ค่าคงที่ของวัสดุ	-	-
FExponentN	ค่าคงที่ของวัสดุ	-	-
FKmat	ความต้านทานการแตกหักในโหมดที่ 1	$MPa\sqrt{m}$	K_{IC} , K_C , $K_{0.2}$ หรือ $K_{0.2BL}$ สำหรับการประเมินในระดับที่ 1
FK02	ความต้านทานการแตกหักในโหมดที่ 1 กรณีที่รอยร้าวมีการเติบโตไปเป็นระยะ 0.2 mm.	$MPa\sqrt{m}$	สำหรับการประเมินในระดับที่ 2
FKg	ความต้านทานการแตกหักในโหมดที่ 1 กรณีที่รอยร้าวมีการเติบโตอย่างมีเสถียรภาพไปเป็นระยะ Δa_g	$MPa\sqrt{m}$	สำหรับการประเมินในระดับที่ 2
FCoeffC	ค่าคงที่ของวัสดุ	-	จากสมการที่ (2.14ข)
FExponentM	ค่าคงที่ของวัสดุ	-	จากสมการที่ (2.14ข)

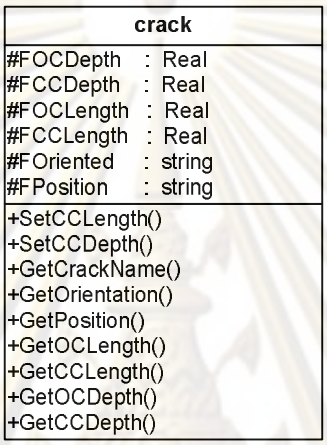
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 5.4 เมทธอดของคลาส matProp และคำอธิบาย

เมทธอด	คำอธิบาย	หมายเหตุ
SetTensileProp	สำหรับกำหนดค่าสมบัติแรงดึงให้กับตัวแปร FSy , FSu และ FSE	-
SetToughness	สำหรับกำหนดค่าความต้านทานการแตกหักให้กับตัวแปร FKmat , FK02 และ FKg	-
SetCGrowth	สำหรับกำหนดค่าสมบัติการเติบโตของรอยร้าวให้กับตัวแปร FCoeffC และ FExponentM	-
GetMatName	สำหรับอ่านค่าชื่อวัสดุ FMatMane (หรืออ่านค่าตัวแปร FMatName)	-
GetSy	สำหรับอ่านค่าความเค้นคราก FSy (หรืออ่านค่าตัวแปร FSy)	-
GetSu	สำหรับอ่านค่าความเค้นดึงสูงสุด FSu (หรืออ่านค่าตัวแปร FSu)	-
GetSE	สำหรับอ่านค่ามอดูลัสของความยืดหยุ่น FSE (หรืออ่านค่าตัวแปร FSE)	-
GetAlpha	สำหรับอ่านค่าคงที่ของวัสดุ FAlpha (หรืออ่านค่าตัวแปร FAlpha)	-
GetExponentN	สำหรับอ่านค่าคงที่ของวัสดุ FExponentN (หรืออ่านค่าตัวแปร FExponentN)	-
GetKmat	สำหรับอ่านค่าความต้านทานการแตกหัก FKmat (หรืออ่านค่าตัวแปร FKmat)	-
GetK02	สำหรับอ่านค่าความต้านทานการแตกหัก FK02 (หรืออ่านค่าตัวแปร FKmat)	-
GetKg	สำหรับอ่านค่าความต้านทานการแตกหัก FKg (หรืออ่านค่าตัวแปร FKg)	-
GetCoeffC	สำหรับอ่านค่าคงที่ของวัสดุ FCoeffC (หรืออ่านค่าตัวแปร FCoeffC)	-
GetExponentM	สำหรับอ่านค่าคงที่ของวัสดุ FExponentM (หรืออ่านค่าตัวแปร FExponentM)	-

5.6.3 รายละเอียดของคลาส crack

คลาส crack ภายหลังเพิ่มแอตทริบิวต์และเมทธอดแล้วแสดงอยู่ในรูปที่ 5.18 ส่วนรายละเอียดของแอตทริบิวต์จะอธิบายอยู่ในตารางที่ 5.5 และรายละเอียดของเมทธอดจะอธิบายอยู่ในตารางที่ 5.6



รูปที่ 5.18 แอตทริบิวต์และเมทธอดของคลาส crack

ตารางที่ 5.5 แอทตริบิวต์ของคลาส crack และคำอธิบาย

แอตทริบิวต์	คำอธิบาย	หน่วย	หมายเหตุ
FOCDepth	ขนาดความลึกของรอยร้าวเริ่มต้น	*m*	-
FCCDepth	ขนาดความลึกของรอยร้าวปัจจุบัน	*m*	-
FOCLength	ขนาดความยาวของรอยร้าวเริ่มต้น	*m*	-
FCCLength	ขนาดความยาวของรอยร้าวปัจจุบัน	*m*	-
FOriented	ทิศทางของรอยร้าวในแนวระนาบ	-	-
FPosition	ตำแหน่งของรอยร้าว	-	วิทยานิพนธ์นี้กำหนดให้รอยร้าวอยู่ภายในท่อเท่านั้น

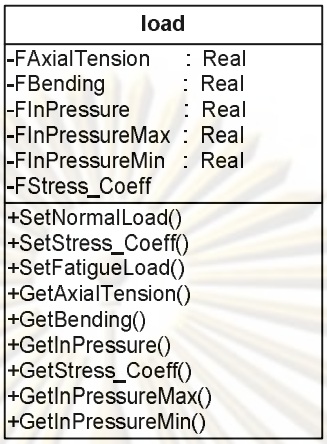
ตารางที่ 5.6 เมทธอดของคลาส crack และคำอธิบาย

เมทธอด	คำอธิบาย	หมายเหตุ
SetCCLength	สำหรับกำหนดค่าขนาดความยาวของรอยร้าวปัจจุบัน ให้กับตัวแปร FCCLength	-
SetCCDepth	สำหรับกำหนดค่าขนาดความลึกของรอยร้าวปัจจุบัน ให้กับตัวแปร FCCDepth	-
GetCrackName	สำหรับอ่านค่าชื่อของรอยร้าว	-
GetOrientation	สำหรับอ่านค่าทิศทางของรอยร้าวในแนวระนาบ FOrientation (หรืออ่านค่าตัวแปร FOrientation)	-
GetPosition	สำหรับอ่านค่าตำแหน่งของรอยร้าว FPosition (หรืออ่านค่าตัวแปร FPosition)	-
GetOCLength	สำหรับอ่านค่าขนาดความยาวของรอยร้าวเริ่มต้น FOCLength (หรืออ่านค่าตัวแปร FOCLength)	-
GetCCLength	สำหรับอ่านค่าขนาดความยาวของรอยร้าวปัจจุบัน FCCLength (หรืออ่านค่าตัวแปร FCCLength)	-
GetOCDepth	สำหรับอ่านค่าขนาดความลึกของรอยร้าวเริ่มต้น FOCDepth (หรืออ่านค่าตัวแปร FOCDepth)	-
GetCCDepth	สำหรับอ่านค่าขนาดความลึกของรอยร้าวปัจจุบัน FCCDepth (หรืออ่านค่าตัวแปร FCCDepth)	-

5.6.4 รายละเอียดของคลาส load

คลาส load ภายหลังเพิ่มแอตทริบิวต์และเมทธอดแล้วแสดงอยู่ในรูปที่ 5.19 ส่วนรายละเอียดของแอตทริบิวต์จะอธิบายอยู่ในตารางที่ 5.7 และรายละเอียดของเมทธอดจะอธิบายอยู่ในตารางที่ 5.8

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.19 แอตทริบิวต์และเมทธอดของคลาส load

ตารางที่ 5.7 แอตทริบิวต์ของคลาส load และคำอธิบาย

แอตทริบิวต์	คำอธิบาย	หน่วย	หมายเหตุ
FAxialTension	แรงดึงในแนวแกน	*N*	-
FBending	โมเมนต์ดัด	*N-m*	-
FInPressure	ความดันภายใน	*MPa*	-
FInPressureMax	ความดันภายในสูงสุด	*MPa*	-
FInPressureMin	ความดันภายในสูงสุด	*MPa*	-
FStress_Coeff	ตัวคูณของความเค้นกระจายไม่คงที่ (arbitrary stress distribution over crack face)	-	-

ตารางที่ 5.8 เมทธอดของคลาส load และคำอธิบาย

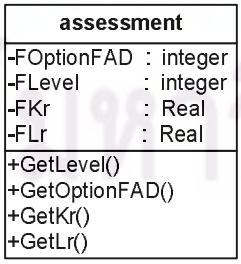
เมทธอด	คำอธิบาย	หมายเหตุ
SetNormalLoad	สำหรับกำหนดค่าภาระสถิตย์ให้กับตัวแปร FAxialTension FBending และ FInPressure	-
SetStress_Coeff	สำหรับกำหนดค่าตัวคูณของความเค้นกระจายไม่คงที่	-
SetFatigueLoad	สำหรับกำหนดค่าภาระล้าให้กับตัวแปร FInPressureMax และ FInPressureMin	-

ตารางที่ 5.8 (ต่อ)

เมทธอด	คำอธิบาย	หมายเหตุ
GetAxialTension	สำหรับอ่านค่าแรงดึงในแนวแกน FAxialTension (หรืออ่านค่าตัวแปร FAxialTension)	-
GetBending	สำหรับอ่านค่าโมเมนต์ FBending (หรืออ่านค่าตัวแปร FBending)	-
GetInPressure	สำหรับอ่านค่าความดันภายใน FInPressure (หรืออ่านค่าตัวแปร FInPressure)	-
GetStress_Coeff	สำหรับอ่านค่าค่าตัวคูณความเค้นกระจายไม่คงที่	-
GetInPressureMax	สำหรับอ่านค่าค่าความดันภายในสูงสุด FInPressureMax (หรืออ่านค่าตัวแปร FInPressureMax)	กรณีท่อรับภาระความดันภายในเปลี่ยนแปลง
GetInPressureMin	สำหรับอ่านค่าความดันภายในต่ำสุด FInPressureMin (หรืออ่านค่าตัวแปร FInPressureMin)	กรณีท่อรับภาระความดันภายในเปลี่ยนแปลง

5.6.5 รายละเอียดของคลาส assessment

คลาส assessment ภายหลังจากเพิ่มแอตทริบิวต์และเมทธอดให้กับคลาสแล้วจะได้คลาสไดอะแกรมดังรูปที่ 5.20 ส่วนรายละเอียดของแอตทริบิวต์จะอธิบายอยู่ในตารางที่ 5.9 และรายละเอียดของเมทธอดจะอธิบายอยู่ในตารางที่ 5.10



รูปที่ 5.20 แอตทริบิวต์และเมทธอดของคลาส assessment

ตารางที่ 5.9 แอตทริบิวต์ของคลาส assessment และคำอธิบาย

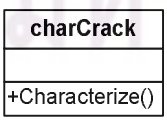
แอตทริบิวต์	คำอธิบาย	หน่วย	หมายเหตุ
FOptionFAD	ชนิดของแผนภาพการประเมิน FAD	-	FAD มี 3 ชนิด
FLevel	ระดับของการประเมิน	-	ระดับการประเมินมี 3 ระดับ
FKr	พิกัดแกนตั้งของจุดการประเมิน	-	-
FLr	พิกัดแกนนอนของจุดการประเมิน	-	-

ตารางที่ 5.10 เมทธอดของคลาส assessment และคำอธิบาย

เมทธอด	คำอธิบาย	หมายเหตุ
GetLevel	สำหรับอ่านค่าระดับการประเมิน FLevel (หรืออ่านค่าตัวแปร FLevel)	-
GetOptionFAD	สำหรับอ่านค่าชนิดของ FAD FOptionFAD (หรืออ่านค่าตัวแปร FOptionFAD)	-
GetKr	สำหรับอ่านค่าพิกัดแกนตั้งของจุดการประเมิน FKr (หรืออ่านค่าตัวแปร FKr)	-
GetLr	สำหรับอ่านอ่านค่าพิกัดแกนนอนของจุดการประเมิน FLr (หรืออ่านค่าตัวแปร FLr)	-

5.6.6 รายละเอียดของคลาส charCrack

คลาส charCrack ภายหลังจากเพิ่มเมทธอดให้กับคลาสแล้วจะได้คลาสไดอะแกรมดังรูปที่ 5.21 จากรูป เมทธอดที่เพิ่มให้กับคลาส ได้แก่ เมทธอด Characterize เป็นเมทธอดสำหรับระบุลักษณะของรอยร้าว

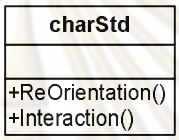


รูปที่ 5.21 แอตทริบิวต์และเมธอดของคลาส charCrack

5.6.7 รายละเอียดของคลาส charStd

คลาส charStd ภายหลังจากเพิ่มเมทธอดให้กับคลาสแล้วจะได้คลาสไดอะแกรมดังรูปที่ 5.22 จากรูป เมทธอดที่เพิ่มให้กับคลาส ได้แก่

- ReOrientation เป็นเมทธอดสำหรับการจัดการวางตัวของรอยร้าวใหม่
- Interaction เป็นเมทธอดสำหรับการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว



รูปที่ 5.22 แอตทริบิวต์และเมทธอดของคลาส charStd

5.6.8 รายละเอียดของคลาส limitLoad

คลาส limitLoad ภายหลังจากเพิ่มเมทธอดให้กับคลาสแล้วจะได้คลาสไดอะแกรมดังรูปที่ 5.23 จากรูป เมทธอดที่เพิ่มให้กับคลาส ได้แก่ เมทธอด GetPL เป็นเมทธอดสำหรับคำนวณค่าภาระขีดจำกัด

รูปที่ 5.23 แอตทริบิวต์และเมทธอดของคลาส limitLoad

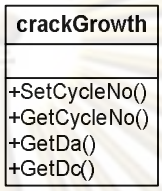
5.6.9 รายละเอียดของคลาส SIF

คลาส SIF ภายหลังจากเพิ่มเมทธอดให้กับคลาสแล้วจะได้คลาสไดอะแกรมดังรูปที่ 5.24 จากรูปเมทธอดที่เพิ่มให้กับคลาส ได้แก่ เมทธอด GetKI เป็นเมทธอดสำหรับคำนวณหาค่าพารามิเตอร์ K ในโหมดการเสียหายโหมดที่ 1(KI)

รูปที่ 5.24 แอตทริบิวต์และเมทธอดของคลาส SIF

5.6.10รายละเอียดของคลาส crackGrowth

คลาส crackGrowth ภายหลังจากเพิ่มเมทธอดให้กับคลาสแล้วจะได้คลาสไดอะแกรมดังรูปที่ 5.25 รายละเอียดของเมทธอดจะแสดงอยู่ในตารางที่ 5.11



รูปที่ 5.25 แอตทริบิวต์และเมทธอดของคลาส crackGrowth

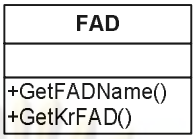
ตารางที่ 5.11 เมทธอดของคลาส crackGrowth และคำอธิบาย

เมทธอด	คำอธิบาย	หมายเหตุ
SetCycleNo	สำหรับกำหนดค่าจำนวนรอบของภาระ	-
GetCycleNo	สำหรับอ่านค่าจำนวนรอบของภาระ	-
GetDa	สำหรับอ่านค่าขนาดความลึกของรอยร้าวหลังประเมินการเติบโตของรอยร้าว	-
GetDc	สำหรับอ่านอ่านค่าขนาดความยาวของรอยร้าวหลังประเมินการเติบโตของรอยร้าว	-

5.6.11 รายละเอียดของคลาส FAD

คลาส FAD ภายหลังจากเพิ่มเมทธอดให้กับคลาสแล้วจะได้คลาสไดอะแกรมดังรูปที่ 5.26 จากรูป เมทธอดที่เพิ่มให้กับคลาส ได้แก่

- GetFADName เป็นเมทธอดสำหรับอ่านค่าชนิดของแผนภาพการประเมิน FAD
- GetKrFAD เป็นเมทธอดสำหรับอ่านค่าพิกัดของจุดการประเมินบนแผนภาพ FAD



รูปที่ 5.26 แอตทริบิวต์และเมทธอดของคลาส FAD

5.7 ผลการออกแบบโปรแกรม

ผลการออกแบบโปรแกรมแบ่งเป็น 2 ส่วน คือ

1) ส่วนติดต่อกับผู้ใช้งาน (หัวข้อที่ 5.7.1)

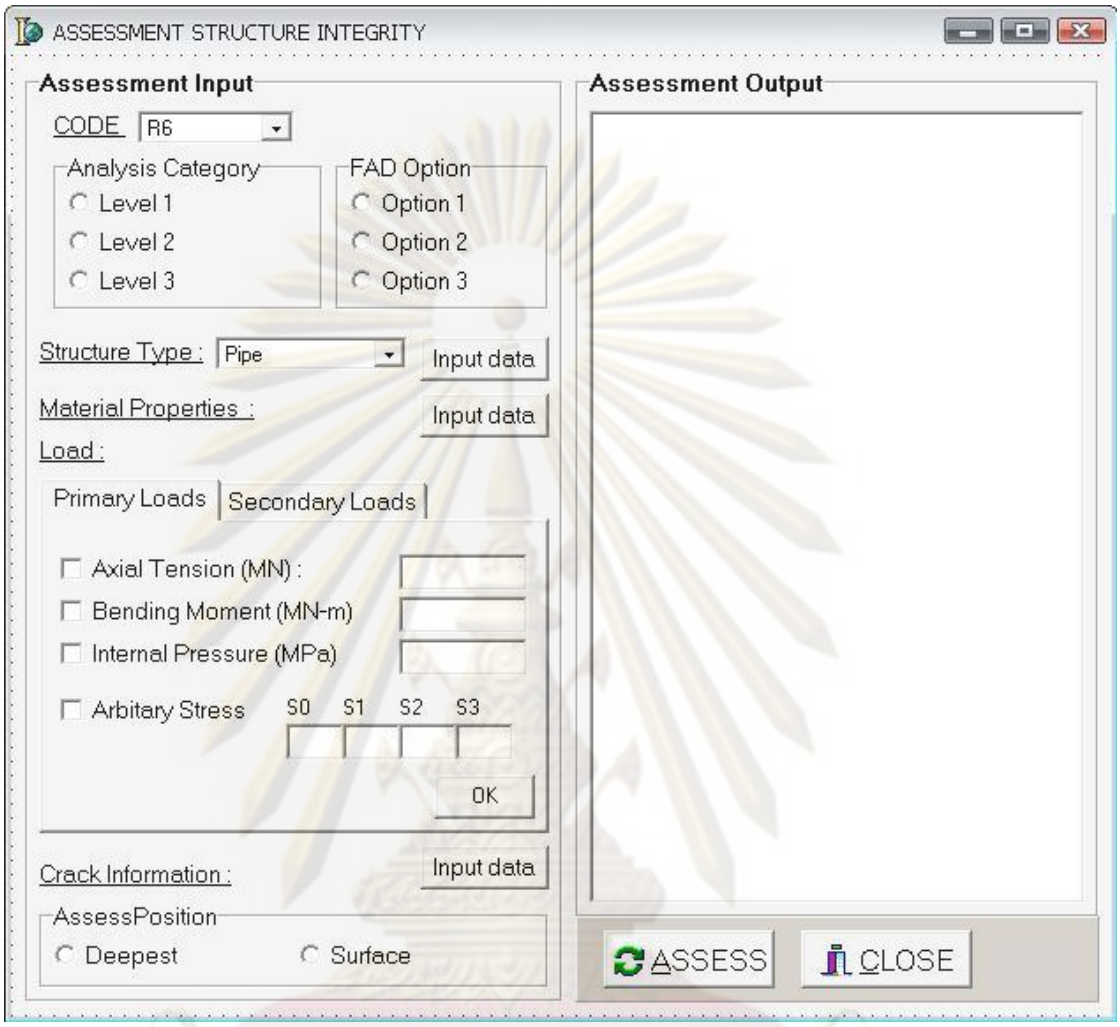
2) ส่วนการประเมิน (หัวข้อที่ 5.7.2)

5.7.1 ส่วนติดต่อกับผู้ใช้งาน

วิทยานิพนธ์นี้เลือกใช้ภาษา Delphi ในการเขียนโปรแกรม ผลการออกแบบโปรแกรมในส่วนติดต่อกับผู้ใช้งานแสดงอยู่ในรูปที่ 5.27 โดยแบ่งเป็นสองส่วน คือ ส่วนรับค่า (Assessment Input) และส่วนแสดงผล (Assessment Output) โดยส่วนรับค่ามีรายละเอียดดังนี้

1) ส่วนระเบียบวิธีการประเมินดังรูปที่ 5.28 ส่วนนี้มีไว้เพื่อให้ผู้ใช้เลือกระเบียบวิธีการประเมิน (CODE) ระดับการประเมิน (Analysis Category) 3 ระดับ และชนิดของแผนภาพประเมิน FAD (FAD Option) 3 ชนิด วิทยานิพนธ์นี้มีระเบียบวิธีการประเมิน ระดับการประเมิน และชนิดของ FAD เฉพาะ ระเบียบวิธี R6 การประเมินระดับที่ 1 และ 2 และ FAD ชนิดที่ 1 และ 2 เท่านั้น ตามลำดับ

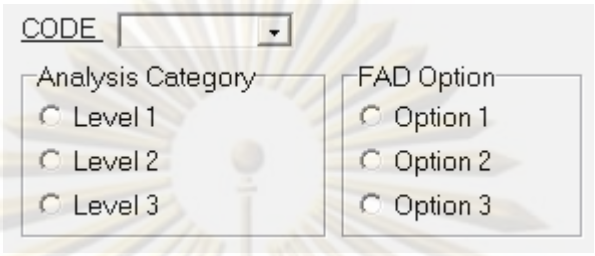
2) ส่วนโครงสร้าง ดังรูปที่ 5.29 (ก) ส่วนนี้มีไว้เพื่อให้ผู้ใช้เลือกชนิดของโครงสร้างที่จะประเมินโดยผู้ใช้สามารถเลือกได้เฉพาะโครงสร้างท่อเท่านั้น การกำหนดขนาดของโครงสร้างผู้ใช้ต้องกดปุ่ม Input Data หลังจากนั้นโปรแกรมจะแสดงหน้าต่างใหม่ขึ้นมาเพื่อให้ผู้ใช้กำหนดค่าขนาดของท่อ คือ รัศมีภายนอกและรัศมีภายใน หลังป้อนค่าแล้วจึงกดปุ่ม OK หากต้องการเปลี่ยนแปลงค่าทั้งหมดกดปุ่ม Clear ดังรูปที่ 5.29 (ข)



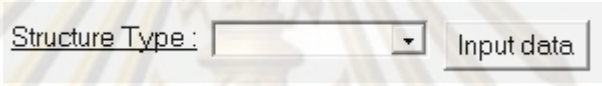
รูปที่ 5.27 ส่วนติดต่อกับผู้ใช้งาน

3) ส่วนสมบัติของวัสดุ ดังรูปที่ 5.30 ส่วนนี้มีไว้เพื่อให้ผู้ใช้กำหนดค่าของสมบัติวัสดุ ผู้ใช้ต้องการกดปุ่ม Input Data โปรแกรมจะแสดงหน้าต่างใหม่ขึ้นมาดังรูปที่ 5.30 (ข) แบ่งเป็น 2 ส่วน คือ สมบัติแรงดึง (Tensile) ประกอบด้วย ความเค้นคราก ความเค้นดึงสูงสุด มอดูลัสของความยืดหยุ่น และค่าคงที่ของวัสดุในสมการความสัมพันธ์ความเค้น-ความเครียดของ Ramberg-Osgood (บทที่ 3 สมการที่ 3.6) และสมบัติความต้านทานการแตกหัก ประกอบด้วย $K_C, K_{IC}, K_{0.2}, K_g$ ขึ้นอยู่กับระดับการประเมิน โปรแกรมถูกออกแบบไว้เพื่อป้องกันความผิดพลาดที่อาจเกิดจากผู้ใช้กำหนดข้อมูลสมบัติของวัสดุผิดพลาดและเพื่อเพิ่มความสะดวกให้กับผู้ใช้จึงออกแบบให้ผู้ใช้สามารถเลือกได้ว่าจะกำหนดสมบัติของวัสดุเองหรือจะเลือกจากฐานข้อมูลสมบัติวัสดุโดยการเลือกจากชื่อของวัสดุ สมบัติของวัสดุค่าต่าง ๆ จะปรากฏบนช่องรับค่า หากยอมรับค่าที่ปรากฏก็กดปุ่ม OK หากผู้ใช้มีความต้องการจะกำหนดข้อมูลสมบัติของวัสดุเองก็สามารถทำได้ หากผู้ใช้ต้องการ

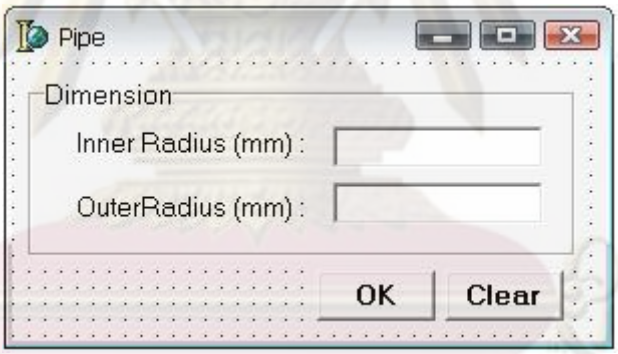
บันทึกข้อมูลของวัสดุชนิดนั้นลงในฐานข้อมูลก็สามารถทำได้โดยการป้อนค่าทั้งหมดแล้วกดปุ่ม Insert



รูปที่ 5.28 ส่วนระเบียบวิธีการประเมิน



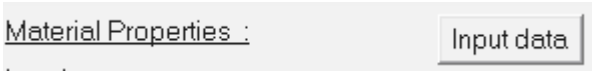
(ก)



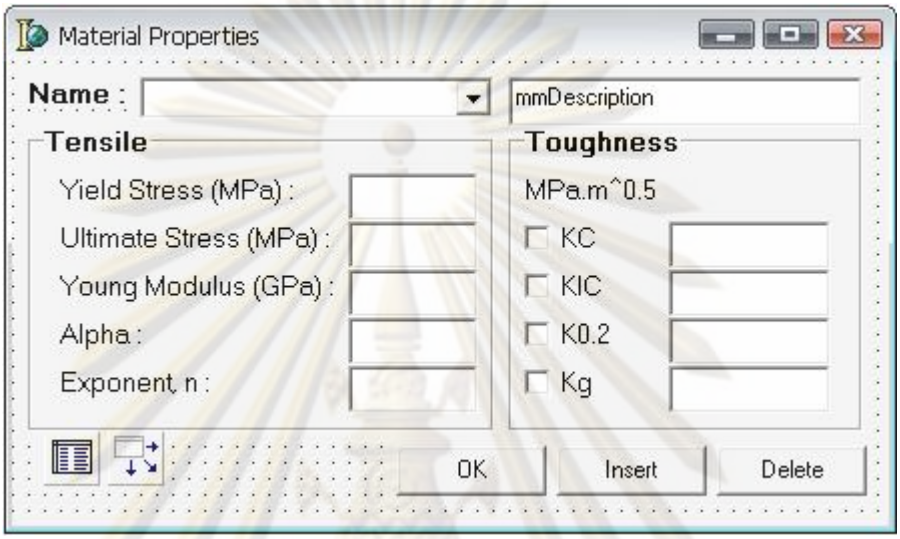
(ข)

รูปที่ 5.29 ส่วนโครงสร้าง
(ก) เลือกชนิดของโครงสร้าง
(ข) หน้าต่างสำหรับกำหนดค่าขนาดของท่อ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



(ก)



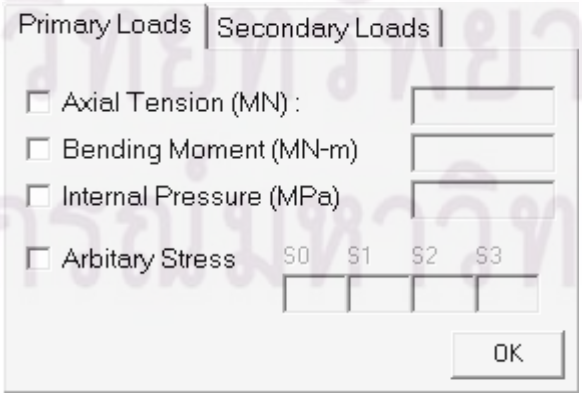
(ข)

รูปที่ 5.30 ส่วนรับค่าสมบัติของวัสดุ

(ก) ปุ่มกดเพื่อจะเข้าสู่หน้าต่าง

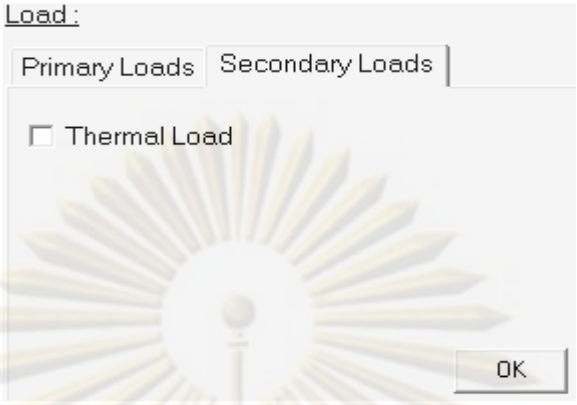
(ข) หน้าต่างสำหรับกำหนดค่าสมบัติขอวัสดุ

4) ส่วนภาระ ดังรูปที่ 5.31 มีไว้เพื่อให้ผู้ใช้กำหนดขนาดของภาระที่กระทำกับโครงสร้าง แบ่งเป็นภาระปฐมภูมิ รูปที่ 5.31 (ก) และภาระทุติยภูมิ รูปที่ 5.31 (ข) โดยผู้ใช้สามารถเลือกได้เฉพาะภาระปฐมภูมิเท่านั้น การกำหนดขนาดของภาระและค่าตัวคูณของความเค้นกระจายจะต้องเลือกก่อนว่าจะกำหนดขนาดของโหลดชนิดใดบ้างจึงจะสามารถป้อนค่าได้



(ก)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



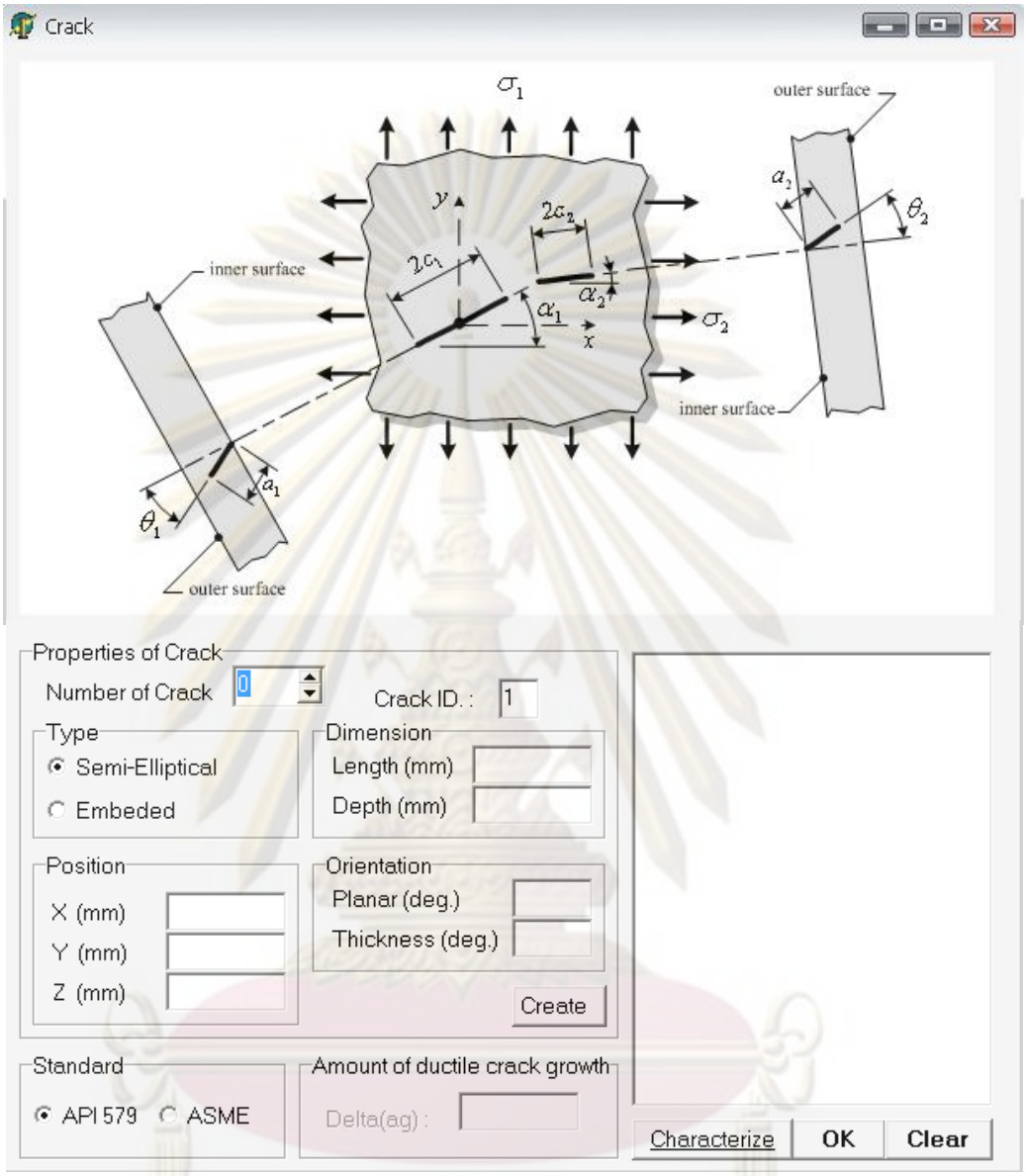
(ข)

รูปที่ 5.31 ส่วนภาระ
(ก) ภาระปฐมภูมิ
(ข) ภาระทุติยภูมิ

5) ส่วนรอยร้าว ดังรูปที่ 5.32 แบ่งเป็น 2 ส่วน คือ ส่วนกำหนดค่า และส่วนแสดงผล ส่วนกำหนกค่ามีไว้เพื่อให้ผู้ใช้สามารถกำหนดขนาดของรอยร้าวที่เกิดขึ้นกับโครงสร้าง โดยผู้ใช้ต้องกำหนดจำนวนของรอยร้าวทั้งหมด (Number of Crack) ไม่เกิน 2 รอย ชนิดของรอยร้าว (Type) ขนาดของรอยร้าว (Dimension) พิกัดตำแหน่งกึ่งกลางของรอยร้าว (Position) มุมในแนวระนาบ (Planar) และมุมในแนวความหนา (Thickness) ตามลำดับ จากนั้นกดปุ่ม Create เพื่อสร้างรอยร้าว จากนั้นเลือกมาตรฐานในการระบุลักษณะของรอยร้าว (เลือกได้เฉพาะมาตรฐาน API 579) หากผู้ใช้เลือกระดับการประเมินระดับที่ 2 แล้วจะต้องกำหนดค่าความยาวรอยร้าวที่เติบโตอย่างมีเสถียรภาพ Δa_g ด้วย การระบุลักษณะรอยร้าวทำโดยการกดปุ่ม Characterize เมื่อเสร็จสิ้นการระบุลักษณะรอยร้าวแล้วก็กดปุ่ม OK เพื่อยืนยันว่าการกำหนดข้อมูลของรอยร้าวให้กับโครงสร้างและการระบุลักษณะรอยร้าวเสร็จแล้ว ส่วนแสดงผลจะแสดงข้อมูลทั้งหมดของรอยร้าวก่อนและหลังการระบุลักษณะ

(ก)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



(ข)

รูปที่ 5.32 ส่วนรอยร้าว

(ก) ปุ่มกดเพื่อจะเข้าสู่หน้าต่าง

(ข) หน้าต่างสำหรับกำหนดค่าข้อมูลของรอยร้าวและระบุลักษณะรอยร้าว

6) ส่วนการกำหนดตำแหน่งที่จะประเมินบนรอยร้าว ดังรูปที่ 5.33 มีไว้เพื่อให้ผู้ใช้เลือกประเมินระหว่างตำแหน่งลึกสุดของรอยร้าวหรือผิวของรอยร้าว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 5.33 ส่วนการกำหนดตำแหน่งที่จะประเมินบนรอยร้าว

7) ส่วนการเติบโตของรอยร้าวล้า ดังรูปที่ 5.34 มีไว้เพื่อหากผู้ใช้มีความต้องการทราบว่าโครงสร้างจะเสียหายก่อนการประเมินครั้งถัดไปหรือไม่ โดยผู้ใช้สามารถประเมินการเติบโตของรอยร้าวที่เกิดจากภาระล้าเท่านั้น โดยใช้สมการการเติบโตของ Paris ในการคำนวณ ผู้ใช้ต้องป้อนค่าภาระสูงสุด ต่ำสุด จำนวนรอบภาระ ค่าคงตัวของวัสดุ (C, m) ในสมการการเติบโตของ Paris (บทที่ 2 สมการที่ 2.12) จากนั้นกดปุ่ม OK

ส่วนการแสดงผลมีรายละเอียดดังรูปที่ 5.35 มีไว้เพื่อแสดงให้ผู้ใช้ทราบว่าได้เลือกหรือกำหนดค่าอะไรลงไปบ้างและเมื่อกดปุ่ม ASSESS โปรแกรมจะแสดงผลการประเมินทั้งหมดให้ผู้ใช้ทราบ หากผู้ใช้ต้องการปิดการทำงานของโปรแกรมก็กดปุ่ม CLOSE

รูปที่ 5.34 ส่วนการเติบโตของรอยร้าวล้า

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 5.35 ส่วนการแสดงผล

5.7.2 ส่วนการประเมิน

ผลการออกแบบโปรแกรมในส่วนการประเมินแบ่งออกเป็น 2 ส่วน คือ 1) การระบุลักษณะรอยร้าว และ 2) การประเมิน

1) การระบุลักษณะรอยร้าว โปรแกรมสามารถระบุลักษณะรอยร้าวได้ไม่เกินสองรอย เนื่องจากการสืบค้นของผู้วิจัยไม่พบว่ามีมาตรฐานใดที่ให้ข้อแนะนำว่าควรทำอย่างไรกับกรณีที่โครงสร้างมีรอยร้าวอยู่ใกล้กันเกินสองรอย

2) การประเมิน โปรแกรมสามารถประเมินการคงสภาพของท่อที่รับภาระปฐมภูมิเท่านั้น และการประเมินไม่สามารถระบุลักษณะของรอยร้าวซ้ำได้ สำหรับการประเมินในระดับที่1 ผู้ใช้เลือกความต้านทานการแตกหักของวัสดุ K_{mat} ได้เฉพาะ K_{IC} เท่านั้น

บทที่ 6

การตรวจสอบความถูกต้องโปรแกรม

ในบทนี้จะกล่าวถึงการตรวจสอบความถูกต้องของโปรแกรมโดยการเปรียบเทียบผลการคำนวณจากโปรแกรมที่ออกแบบกับผลที่ได้จากโปรแกรม MathCAD การตรวจสอบจะแบ่งออกเป็น 2 ส่วน คือ การตรวจสอบการระบุลักษณะรอยร้าว และการตรวจสอบผลการประเมินการคงสภาพของท่อที่มีรอยร้าว ในหัวข้อที่ 6.1 และ 6.2 ตามลำดับ

6.1 การตรวจสอบการระบุลักษณะรอยร้าว

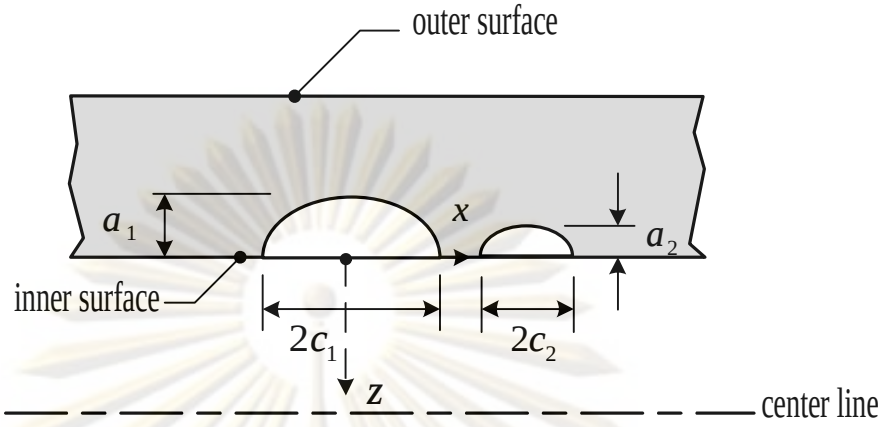
การตรวจสอบการระบุลักษณะรอยร้าวแบ่งเป็น 3 กรณี คือ

1) กรณีระหว่างรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย (หัวข้อที่ 6.1.1)
2) กรณีระหว่างรอยร้าวฝังรูปวงรีทั้งสองรอย (หัวข้อที่ 6.1.2)
3) กรณีระหว่างรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี (หัวข้อที่ 6.1.3)

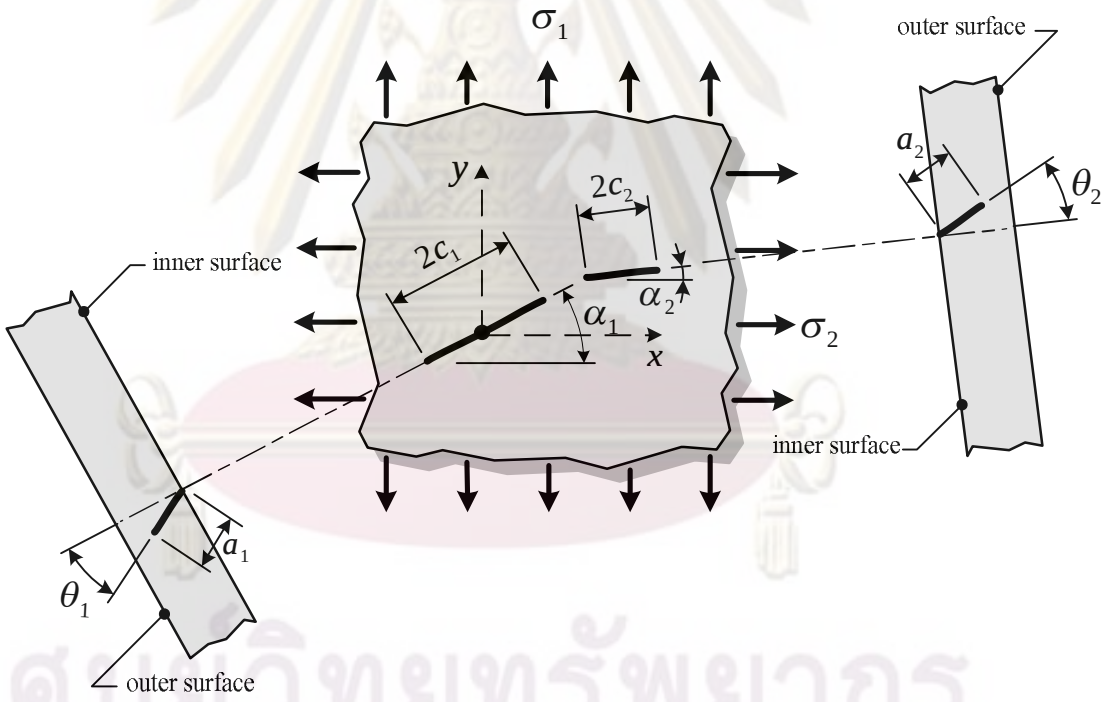
ทั้ง 3 กรณีเป็นรอยร้าวที่วางตัวอยู่ภายในของท่อ กำหนดให้ท่อรับภาระสถิตย์ความดันภายในขนาดเท่ากับ 10 *MPa* โดยมีรายการตรวจสอบ คือ การจัดทิศทางการวางตัวของรอยร้าวและการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว

6.1.1 กรณีระหว่างรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย

รอยร้าวทั้งสองเมื่อมองด้านข้างดังรูปที่ 6.1 และด้านบนดังรูปที่ 6.2 แกน x แทนแนวแกน แกน y แทนแนวเส้นรอบวง และแกน z แทนแนวรัศมีชี้เข้าจุดศูนย์กลาง รายละเอียดของรอยร้าวแสดงอยู่ในตารางที่ 6.1 ตารางคอลัมน์แรกแสดงข้อมูลของรอยร้าว ประกอบด้วย ชนิด ขนาด ทิศทางการวางตัว และพิกัดของตำแหน่งกึ่งกลางรอยร้าว คอลัมน์ที่สองแสดงข้อมูลของรอยร้าวรอยที่ 1 และคอลัมน์ที่สามแสดงข้อมูลของรอยร้าวรอยที่ 2 โดยกำหนดพิกัดจุดเริ่มต้น (x, y, z) อยู่ที่ตำแหน่งกึ่งกลางของรอยร้าวรอยแรก



รูปที่ 6.1 กรณีมองด้านข้างของรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย



รูปที่ 6.2 กรณีมองด้านบนของรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย

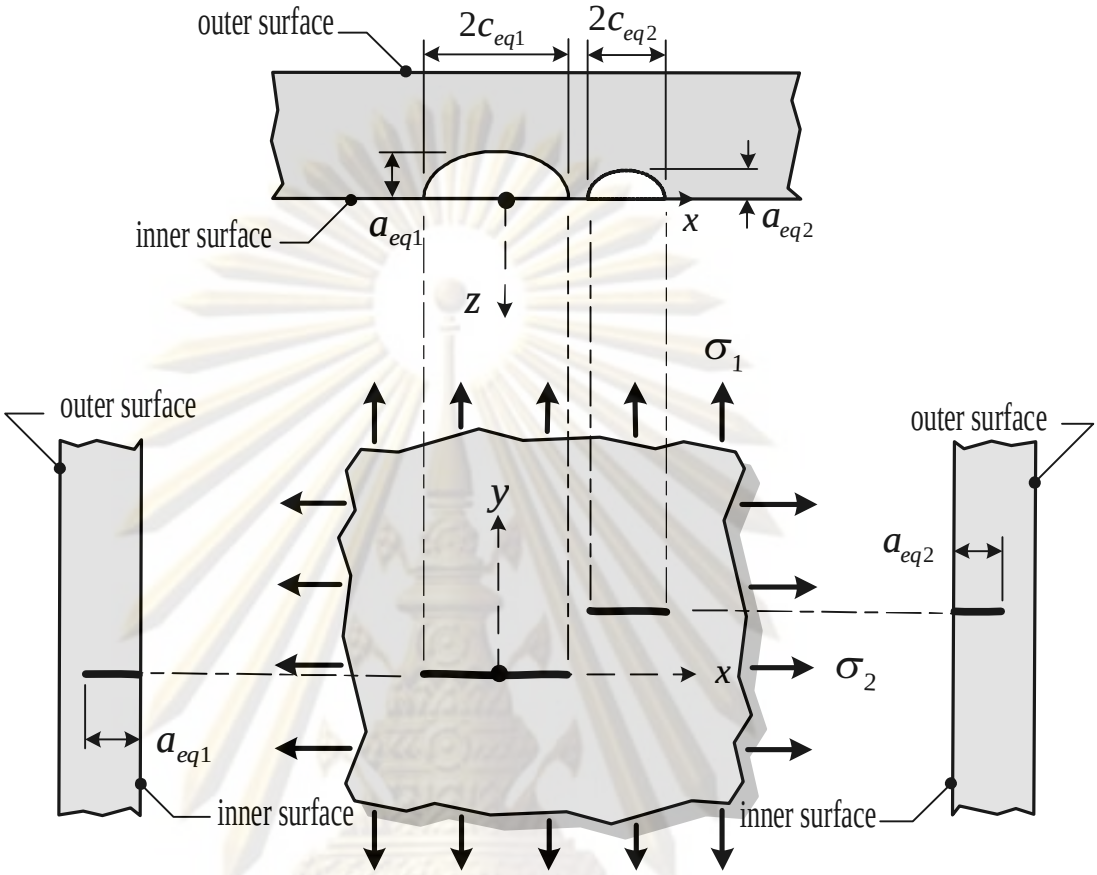
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 6.1 รายละเอียดของรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย

ข้อมูล		รอยที่ 1	รอยที่ 2
ชนิด		รอยร้าวผิวรูปครึ่งวงรี	รอยร้าวผิวรูปครึ่งวงรี
1. ขนาด (mm)	length, $2c$	40	20
	depth, a	10	5
2. ทิศทางการวางตัว (deg)	planar angle, α	4	2
	thickness angle, θ	5	15
3. พิกัดตำแหน่งกึ่งกลางรอยร้าว (mm)	X_m	0	40
	Y_m	0	20
	Z_m	0	0

การจัดทิศทางการวางตัวของรอยร้าวทำตามขั้นตอนในบทที่ 3 หัวข้อย่อยที่ 3.5.2 ผลจากการจัดทิศทางจะได้รอยร้าวใหม่ที่มีทิศทางการวางตัวตั้งฉากกับความเค้นหลักที่มีค่ามาก ความยาวและความลึกของรอยร้าวใหม่เรียกว่าความยาวรอยร้าวสมมูล c_{eq} และความลึกรอยร้าวสมมูล a_{eq} ลักษณะของรอยร้าวหลังจัดทิศทางการวางตัวดังแสดงในรูปที่ 6.3

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.3 รอยร้าวผิวรูปครึ่งวงรีทั้งสองรอยหลังผ่านการจัดทิศทางการวางตัว

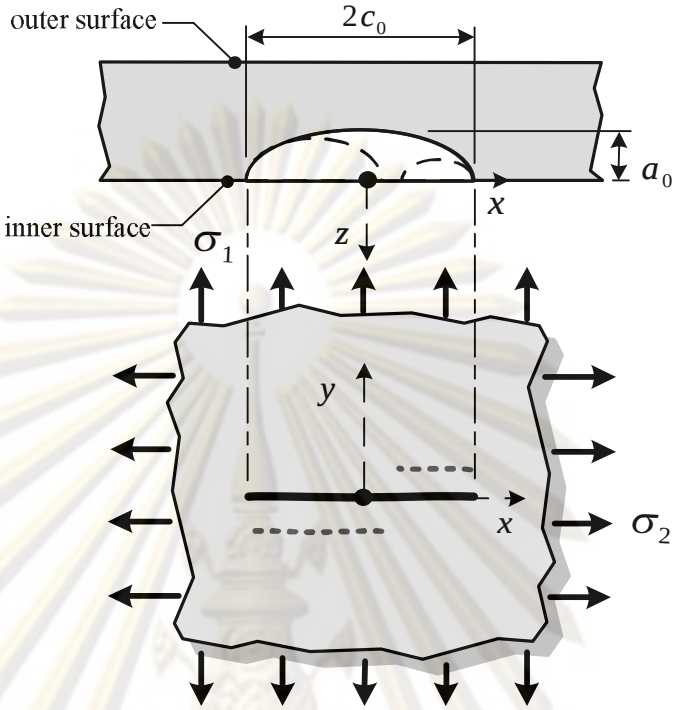
ผลการคำนวณความยาวรอยร้าวสมมูลและความลึกรอยร้าวสมมูลด้วยโปรแกรมที่ออกแบบ (ภาคผนวก ค.) แสดงอยู่ในตารางที่ 6.2 ตารางคอลัมน์แรกแสดงข้อมูลของรอยร้าวประกอบด้วย ชนิด ขนาดสมมูล และพิกัดตำแหน่งกึ่งกลางรอยร้าว คอลัมน์ที่สองแสดงข้อมูลของรอยร้าวรอยที่ 1 และคอลัมน์สุดท้ายแสดงข้อมูลของรอยร้าวรอยที่ 2 เมื่อเปรียบเทียบกับผลการคำนวณด้วยโปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.1.1) พบว่าผลการคำนวณที่ได้จากโปรแกรมที่ออกแบบกับ MathCAD ตรงกัน แสดงว่า โปรแกรมที่ออกแบบมีความถูกต้องในการคำนวณการจัดทิศทางการวางตัวของรอยร้าวกรณีโครงสร้างมีรอยร้าวผิวรูปครึ่งวงรีอยู่ใกล้กันสองรอย

ตารางที่ 6.2 ผลการคำนวณความยาวและความลึกรอยร้าวสมมูลของรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย

ข้อมูล		รอยที่ 1	รอยที่ 2
1. ชนิด		รอยร้าวผิวรูปครึ่งวงรี	รอยร้าวผิวรูปครึ่งวงรี
2. ขนาดสมมูล (mm)	$2c_{eq}$	40.55	20.16
	a_{eq}	10.03	5.19
3. พิกัดตำแหน่งกึ่งกลางรอยร้าว (mm)	X_m	0.0	40.0
	Y_m	0.0	20.0
	Z_m	0.0	0.0

การพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว พิจารณาจากตารางที่ 3.2 และ 3.3 ในบทที่ 3 ผลจากการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าวทั้งสองหลังจากการจัดทิศทางการวางตัวของรอยร้าวแล้ว อาจก่อให้เกิดรอยร้าวใหม่รอยเดียว (รวมกัน) ดังรูปที่ 6.4 ถ้ารายละเอียดของรอยร้าวที่ผ่านการจัดทิศทางตรงกับข้อกำหนด หรืออาจจะยังคงมีรอยร้าวทั้งคู่อยู่เหมือนเดิมหากปฏิสัมพันธ์ดังกล่าวไม่ตรงกับข้อกำหนด สำหรับการตรวจสอบของวิทยานิพนธ์นี้จะตรวจสอบเฉพาะกรณีที่รอยร้าวที่ตรงกับข้อกำหนดเท่านั้น กรณีนี้รอยร้าวหลังพิจารณาปฏิสัมพันธ์จะได้รอยร้าวผิวรูปครึ่งวงรี พิกัดตำแหน่งกึ่งกลางรอยร้าวใหม่จะอยู่ระหว่างขอบเขตของรอยร้าวเดิมทั้งสอง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.4 รอยร้าวผิวรูปครึ่งวงรีทั้งสองรอยหลังผ่านการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว

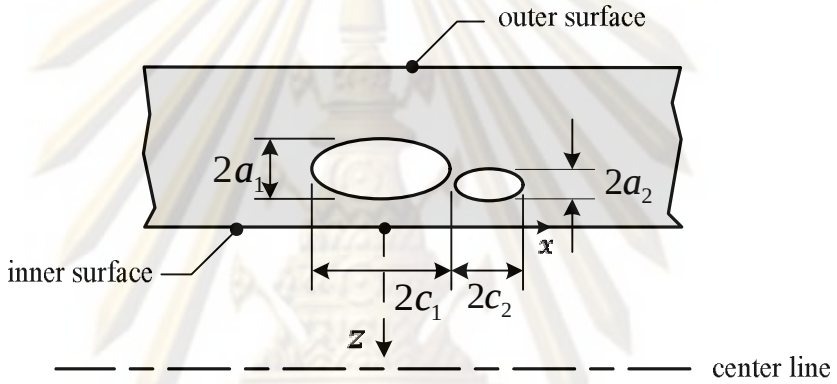
ขนาดของรอยร้าวหลังพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าวแสดงอยู่ในตารางที่ 6.3 เมื่อเปรียบผลการคำนวณขนาดและพิกัดตำแหน่งกึ่งกลางของรอยร้าวจากโปรแกรมที่ออกแบบกับผลการคำนวณของโปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.1.1) พบว่า ตรงกัน แสดงว่าโปรแกรมที่ออกแบบมีความถูกต้องในการระบุลักษณะรอยร้าวกรณีที่โครงสร้างมีรอยร้าวผิวรูปครึ่งวงรีอยู่ใกล้กันสองรอย

ตารางที่ 6.3 ผลการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย

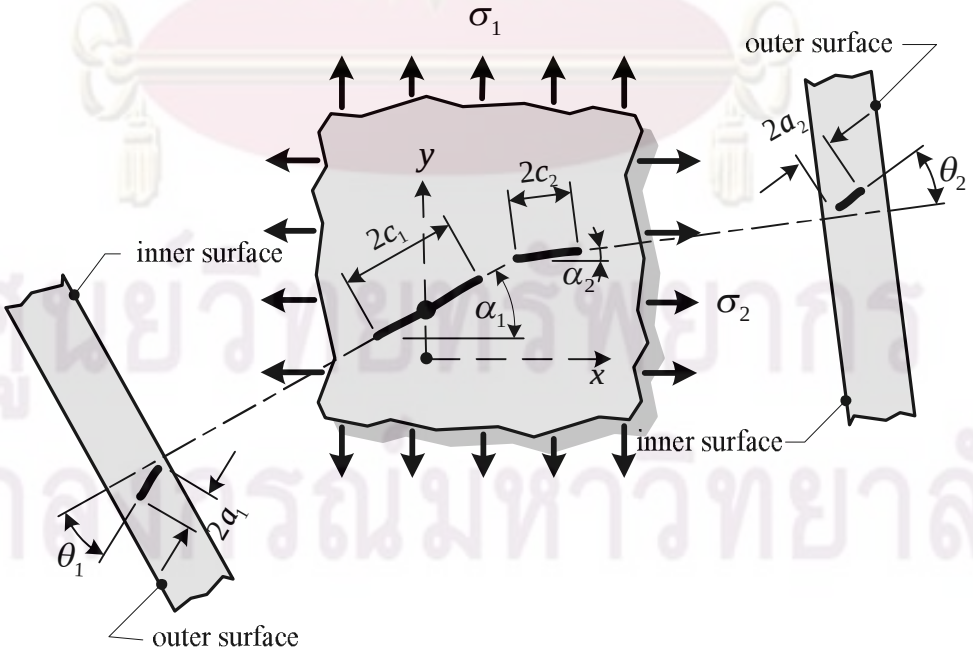
ข้อมูล		รอยร้าวหลังพิจารณา
ชนิด		รอยร้าวผิวรูปครึ่งวงรี
1. ขนาด (*mm*)	length,$2c$	70.35
	depth, a	10.03
2. พิกัดตำแหน่งกึ่งกลางรอยร้าว (*mm*)	X_m	14.9
	Y_m	10.0
	Z_m	0.0

6.1.2 กรณีระหว่างรอยร้าวฝังรูปวงรีทั้งสองรอย

รอยร้าวทั้งสองเมื่อมองด้านข้างดังรูปที่ 6.5 และด้านบนดังรูปที่ 6.6 แกน x แทนแนวแกน แกน y แทนแนวเส้นรอบวง และแกน z แทนแนวรัศมี รายละเอียดของรอยร้าวแสดงอยู่ในตารางที่ 6.4 ในตารางคอลัมน์แรกแสดงข้อมูลของรอยร้าว ประกอบด้วย ชนิด ขนาด ทิศทางการวางตัว และตำแหน่งกึ่งกลางของรอยร้าว คอลัมน์ที่สองแสดงข้อมูลของรอยร้าวรอยที่ 1 และคอลัมน์ที่สามแสดงข้อมูลของรอยร้าวรอยที่ 2 โดยกำหนดพิกัดจุดเริ่มต้น (x, y, z) อยู่ที่ตำแหน่งผิวในของท่อ ห่างจากจุดกึ่งกลางรอยร้าวรอยแรกในแนวแกน y เท่ากับ 10 *mm*



รูปที่ 6.5 กรณีมองด้านข้างของรอยร้าวฝังรูปวงรีทั้งสองรอย



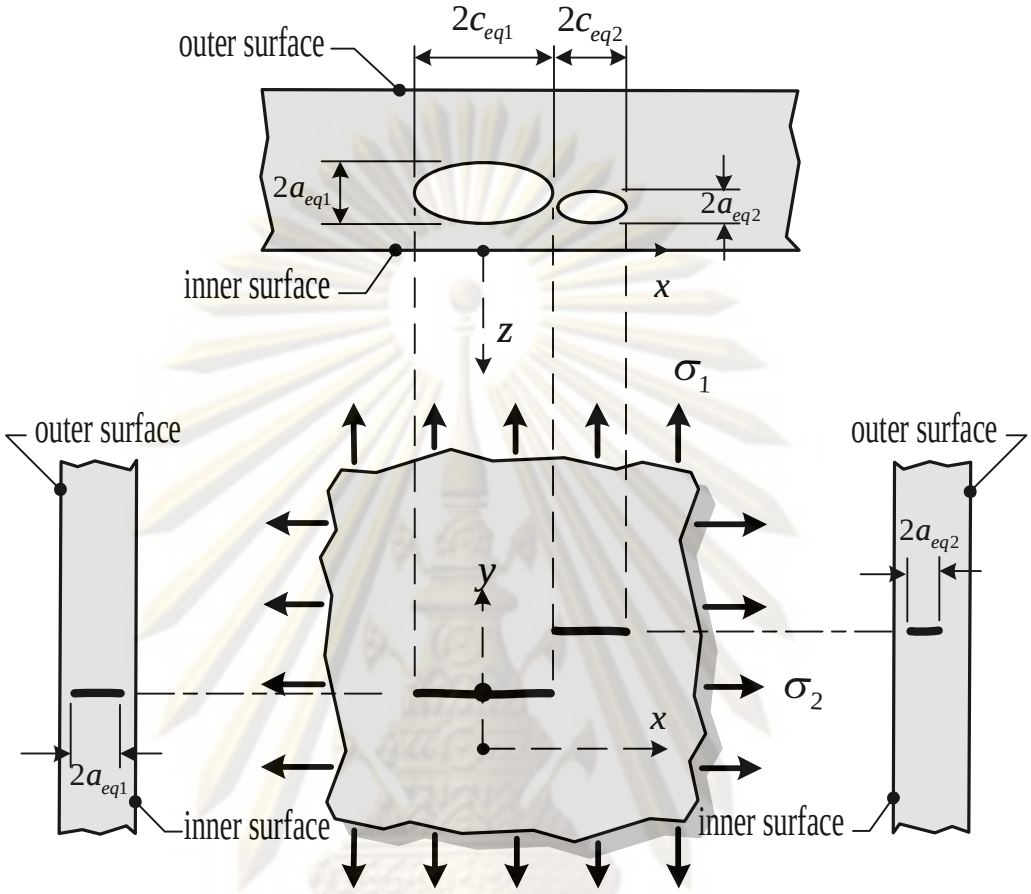
รูปที่ 6.6 กรณีมองด้านบนของรอยร้าวฝังรูปวงรีทั้งสองรอย

ตารางที่ 6.4 รายละเอียดของรอยร้าวฝังรูปวงรีทั้งสองรอย

ข้อมูล		รอยที่ 1	รอยที่ 2
ชนิด		รอยร้าวฝังรูปวงรี	รอยร้าวฝังรูปวงรี
1. ขนาด (mm)	length,$2c$	80.00	40.00
	depth,$2a$	40.00	20.00
2. ทิศทางการวางตัว (deg)	planar angle, α	20.00	5.00
	thickness angle, θ	5.00	15.00
3. พิกัดตำแหน่งกึ่งกลางรอยร้าว (mm)	X_m	0.0	60.0
	Y_m	10.0	40.0
	Z_m	-70.0	-80.0

การจัดทิศทางการวางตัวของรอยร้าวทำตามขั้นตอนในบทที่ 3 หัวข้อย่อยที่ 3.5.2 ผลจากการจัดทิศทางจะได้รอยร้าวใหม่ที่มีทิศทางการวางตัวตั้งฉากกับความเค้นหลักที่มีค่ามาก ความยาวและความลึกของรอยร้าวใหม่เรียกว่าความยาวรอยร้าวสมมูล c_{eq} และความลึกรอยร้าวสมมูล a_{eq} ลักษณะของรอยร้าวหลังจัดทิศทางแสดงดังรูปที่ 6.7

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.7 รอยร้าวฝังรูปวงรีทั้งสองรอยหลังผ่านการจัดทิศทางการวางตัว

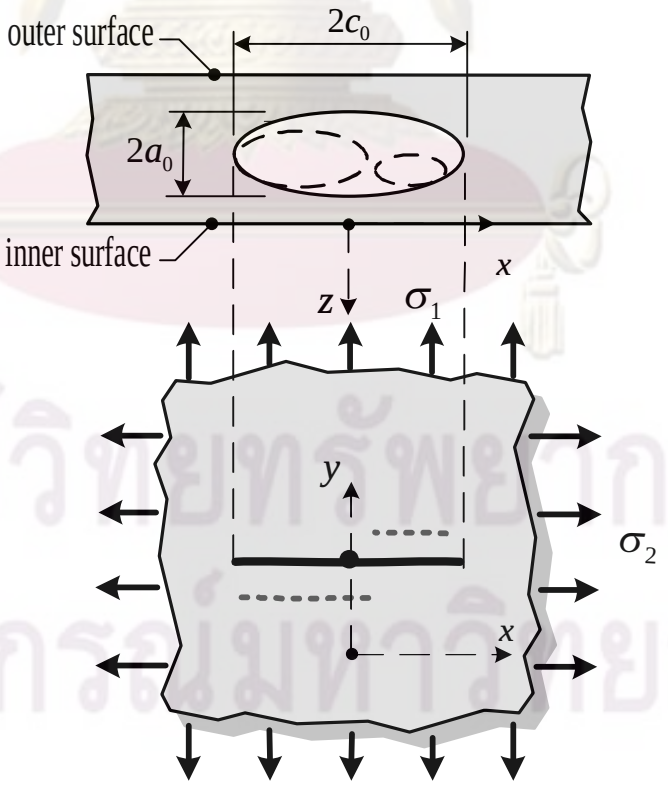
ผลการคำนวณความยาวรอยร้าวสมมูลและความลึกรอยร้าวสมมูลด้วยโปรแกรมที่ออกแบบ (ภาคผนวก ค.) แสดงอยู่ในตารางที่ 6.5 ตารางคอลัมน์แรกแสดงข้อมูลของรอยร้าว ประกอบด้วย ชนิด ขนาดสมมูล และพิกัดตำแหน่งกึ่งกลางรอยร้าว คอลัมน์ที่สองแสดงข้อมูลของรอยร้าวรอยที่ 1 และคอลัมน์สุดท้ายแสดงข้อมูลของรอยร้าวรอยที่ 2 เมื่อเปรียบเทียบกับผลการคำนวณด้วยโปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.1.2) พบว่าผลการคำนวณที่ได้จากโปรแกรมที่ออกแบบกับ MathCAD ตรงกัน แสดงว่า โปรแกรมที่ออกแบบมีความถูกต้องในการคำนวณการจัดทิศทางการวางตัวของรอยร้าวกรณีโครงสร้างมีรอยร้าวฝังรูปวงรีอยู่ใกล้กันสองรอย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 6.5 ผลการคำนวณความยาวและความลึกรอยร้าวสมมูลของรอยร้าวฝังรูปวงรีทั้งสองรอย

ข้อมูล		รอยที่ 1	รอยที่ 2
ชนิด		รอยร้าวฝังรูปวงรี	รอยร้าวฝังรูปวงรี
1. ขนาดสมมูล (*mm*)	$2c_{eq}$	39.70	20.32
	$2a_{eq}$	20.06	10.38
2. พิกัดตำแหน่งกึ่งกลางรอยร้าว (*mm*)	X_m	0.00	60.00
	Y_m	10.00	40.00
	Z_m	-70	-80

การพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว พิจารณาเช่นเดียวกับกรณีแรกหัวข้อที่ 6.1.1 รอยร้าวหลังพิจารณาปฏิสัมพันธ์จะได้รอยร้าวฝังรูปวงรี แสดงอยู่ในรูปที่ 6.8



รูปที่ 6.8 รอยร้าวฝังรูปวงรีทั้งสองรอยหลังผ่านการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

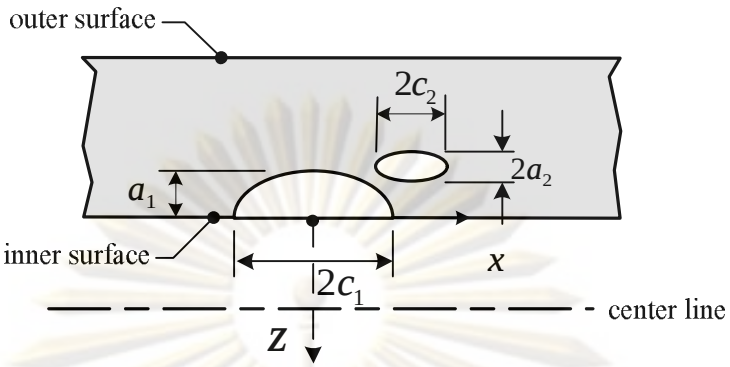
ขนาดของรอยร้าวหลังพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าวแสดงอยู่ในตารางที่ 6.6 เมื่อเปรียบผลการคำนวณขนาดและพิกัดตำแหน่งกึ่งกลางของรอยร้าวจากโปรแกรมที่ออกแบบกับผลการคำนวณของโปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.1.2) พบว่า ตรงกัน แสดงว่าโปรแกรมที่ออกแบบมีความถูกต้องในการระบุลักษณะรอยร้าวกรณีที่โครงสร้างมีรอยร้าวฝังรูปวงรีอยู่ใกล้กันสองรอย

ตารางที่ 6.6 ผลการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าวฝังรูปวงรีทั้งสองรอย

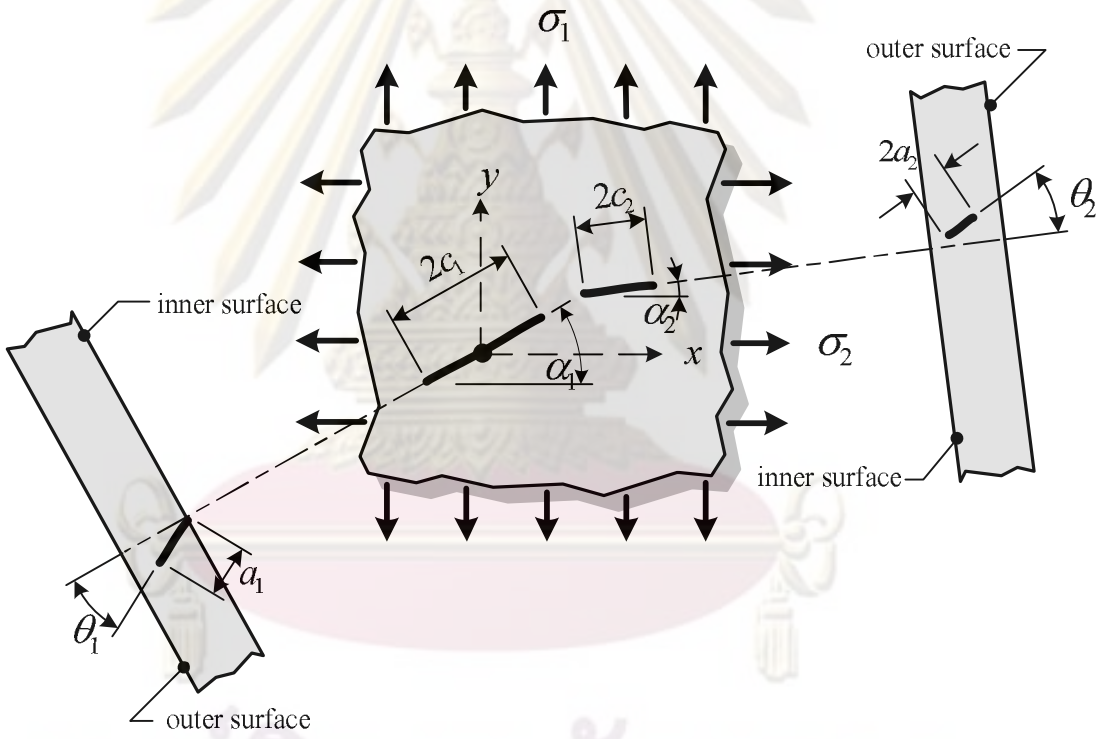
ข้อมูล		รอยร้าวหลังพิจารณา
ชนิด		รอยร้าวฝังรูปวงรี
1. ขนาด (mm)	length,$2c$	120.02
	depth, $2a$	40.44
2. พิกัดตำแหน่งกึ่งกลางรอยร้าว (mm)	X_m	20.31
	Y_m	25.00
	Z_m	-70.16

6.1.3 กรณีระหว่างรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี

รอยร้าวทั้งสองเมื่อมองด้านข้างดังรูปที่ 6.9 และด้านบนดังรูปที่ 6.10 แกน x แทนแนวแกน แกน y แทนแนวเส้นรอบวง และแกน z แทนแนวรัศมี รายละเอียดของรอยร้าวแสดงอยู่ในตารางที่ 6.7 ในตารางคอลัมน์แรกแสดงข้อมูลของรอยร้าว ประกอบด้วย ชนิด ขนาด ทิศทางการวางตัว และตำแหน่งกึ่งกลางของรอยร้าว คอลัมน์ที่สองแสดงข้อมูลของรอยร้าวรอยที่ 1 และคอลัมน์ที่สามแสดงข้อมูลของรอยร้าวรอยที่ 2 โดยกำหนดพิกัดจุดเริ่มต้น (x, y, z) อยู่ที่ตำแหน่งกึ่งกลางรอยร้าวรอยแรก



รูปที่ 6.9 กรณีมองด้านข้างของรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี



รูปที่ 6.10 กรณีมองด้านบนของรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี

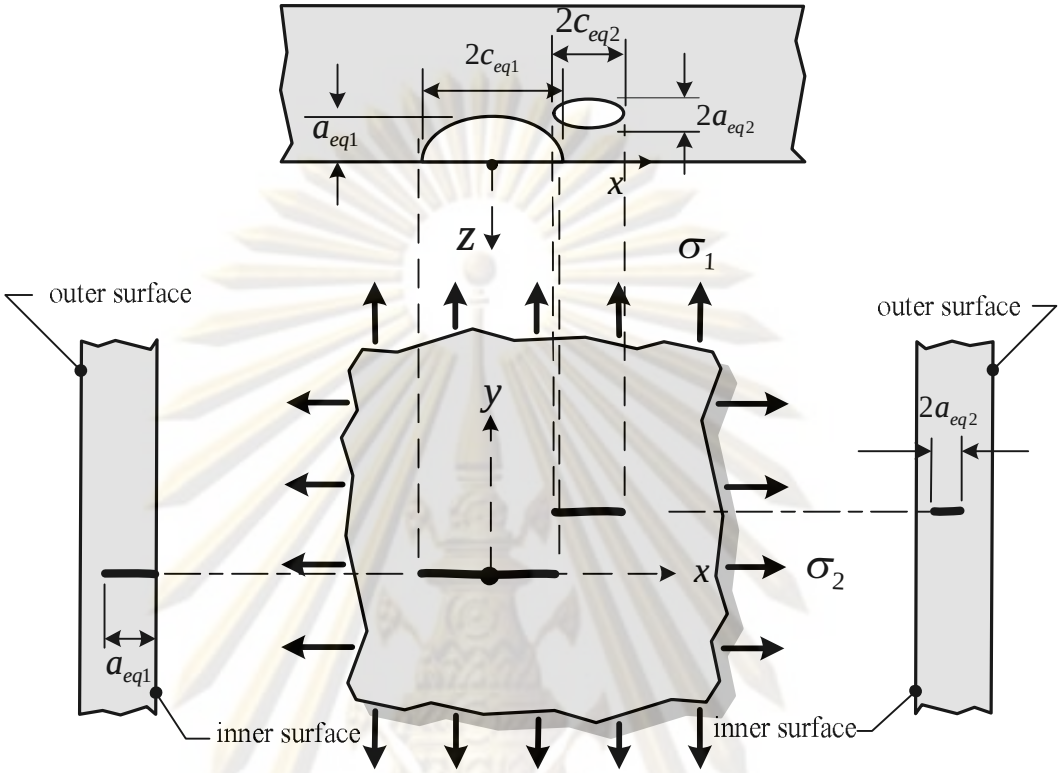
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 6.7 รายละเอียดของรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี

ข้อมูล		รอยที่ 1	รอยที่ 2
ชนิด		รอยร้าวผิวรูปครึ่งวงรี	รอยร้าวฝังรูปวงรี
1. ขนาด (mm)	length,$2c$	60	20
	depth, a	10	5
2. ทิศทางการวางตัว (deg)	planar angle, α	20	5
	thickness angle, θ	5	15
3. พิกัดตำแหน่งกึ่งกลางรอยร้าว (mm)	X_m	0	30
	Y_m	0	30
	Z_m	0	-15

การจัดทิศทางการวางตัวของรอยร้าวทำตามขั้นตอนในบทที่ 3 หัวข้อย่อยที่ 3.5.2 ผลจากการจัดทิศทางจะได้รอยร้าวใหม่ที่มีทิศทางการวางตัวตั้งฉากกับความเค้นหลักที่มีค่ามาก ความยาวและความลึกของรอยร้าวใหม่เรียกว่าความยาวรอยร้าวสมมูล c_{eq} และความลึกรอยร้าวสมมูล a_{eq} ลักษณะของรอยร้าวหลังจัดทิศทางดังรูปที่ 6.11

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.11 รอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรีหลังผ่านการจัดทิศทางการวางตัว

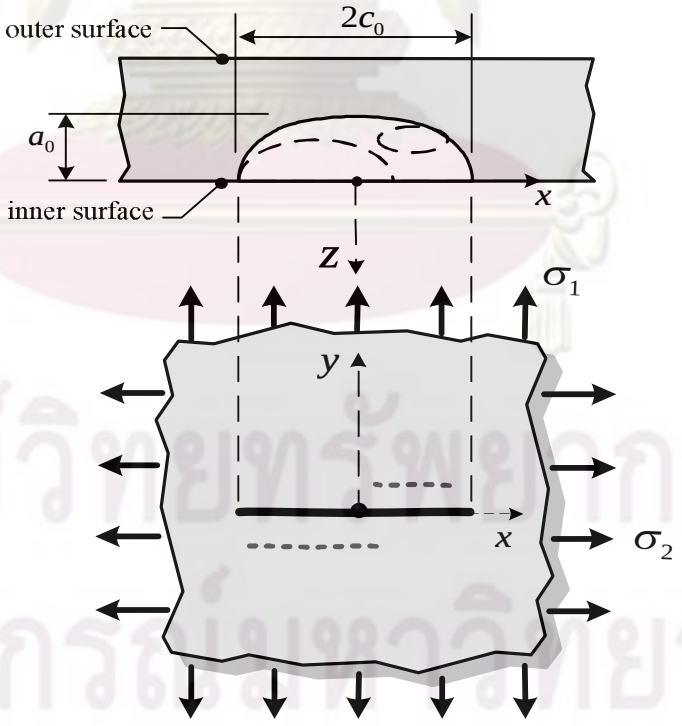
ผลการคำนวณความยาวรอยร้าวสมมูลและความลึกรอยร้าวสมมูลด้วยโปรแกรมที่ออกแบบ (ภาคผนวก ค.) แสดงอยู่ในตารางที่ 6.8 ตารางคอลัมน์แรกแสดงข้อมูลของรอยร้าวประกอบด้วย ชนิด ขนาดสมมูล และพิกัดตำแหน่งกึ่งกลางรอยร้าว คอลัมน์ที่สองแสดงข้อมูลของรอยร้าวรอยที่ 1 และคอลัมน์สุดท้ายแสดงข้อมูลของรอยร้าวรอยที่ 2 เมื่อเปรียบเทียบกับผลการคำนวณด้วยโปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.1.3) พบว่าผลการคำนวณที่ได้จากโปรแกรมที่ออกแบบกับ MathCAD ตรงกัน แสดงว่า โปรแกรมที่ออกแบบมีความถูกต้องในการคำนวณการจัดทิศทางการวางตัวของรอยร้าวกรณีโครงสร้างมีรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรีอยู่ใกล้กัน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 6.8 ผลการคำนวณความยาวและความลึกรอยร้าวสมมูลของรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี

ข้อมูล		รอยที่ 1	รอยที่ 2
ชนิด		รอยร้าวผิวรูปครึ่งวงรี	รอยร้าวฝังรูปวงรี
1. ขนาดสมมูล (mm)	$2c_{eq}$	59.56	20.32
	a_{eq}	10.03	5.19
2. พิกัดตำแหน่งกึ่งกลางรอยร้าว (mm)	X_m	0.00	30.00
	Y_m	0.00	30.00
	Z_m	0.00	-15

การพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว พิจารณาเช่นเดียวกับกรณีแรกหัวข้อที่ 6.1.1 รอยร้าวหลังพิจารณาปฏิสัมพันธ์จะได้รอยร้าวผิวรูปครึ่งวงรีดังรูปที่ 6.12



รูปที่ 6.12 รอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรีหลังผ่านการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ขนาดของรอยร้าวหลังพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าวแสดงอยู่ในตารางที่ 6.9 เมื่อเปรียบผลการคำนวณขนาดและพิกัดตำแหน่งกึ่งกลางของรอยร้าวจากโปรแกรมที่ออกแบบกับผลการคำนวณของโปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.1.3) พบว่า ตรงกัน แสดงว่าโปรแกรมที่ออกแบบมีความถูกต้องในการระบุลักษณะรอยร้าวกรณีที่โครงสร้างมีรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรีอยู่ใกล้กัน

ตารางที่ 6.9 ผลการพิจารณาปฏิสัมพันธ์ระหว่างรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี

ข้อมูล		รอยร้าวหลังพิจารณา
ชนิด		รอยร้าวผิวรูปครึ่งวงรี
1. ขนาด (*mm*)	length,2 *c*	69.94
	depth, *a*	20.19
2. พิกัดตำแหน่งกึ่งกลางรอยร้าว (*mm*)	X_m	5.19
	Y_m	15.00
	Z_m	0.00

6.2 การตรวจสอบผลการประเมิน

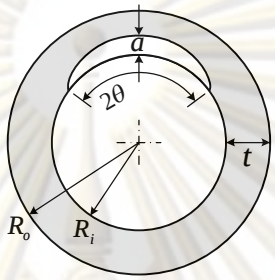
การตรวจสอบความถูกต้องแบ่งตามการวางตัวของรอยร้าวและภาระ เป็น 3 กรณี คือ

1) ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระแรงดึงตามแนวแกน (หัวข้อ 6.2.1)
2) ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระโมเมนต์ดัด (หัวข้อ 6.2.2)
3) ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายใน (หัวข้อ 6.2.3)

การตรวจสอบทั้ง 3 หัวข้อถูกประเมินโดยใช้ระดับการประเมินคู่กับแผนภาพประเมินความเสียหาย FAD ดังนี้คือ FAD ชนิดที่ 1 คู่กับ การประเมินระดับที่ 1 และ ระดับที่ 2 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และระดับที่ 2 ตามลำดับ มีรายละเอียดดังนี้

6.2.1 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระแรงดึงแนวแกน

ท่อมีลักษณะการวางตัวของรอยร้าวและภาระดังรูปที่ 6.19 กรณีนี้ไม่สามารถประเมินการเติบโตของรอยร้าวเนื่องจากภาระล้าได้เนื่องจากไม่มีสมการผลเฉลยพารามิเตอร์ K ตำแหน่งผิวของรอยร้าว



รูปที่ 6.13 ลักษณะของท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระแรงดึงตามแนวแกน

6.2.1.1 ข้อมูล

ขนาดท่อ

รัศมีภายนอก , R_o	0.600	m
รัศมีภายใน , R_i	0.543	m

วัสดุ SA106Gr.C [19]

มอดูลัสของความยืดหยุ่น , E	200.291	GPa
ความเค้นคราก , σ_y	268.700	MPa
ความเค้นดึงสูงสุด , σ_u	413.700	MPa
ค่าคงที่ของวัสดุ (บทที่ 3 สมการที่ (3.6)) , α	1	-
ค่าคงที่ของวัสดุ (บทที่ 3 สมการที่ (3.6)) , n	7.435	-
ความต้านทานการแตกหัก , K_{IC}	59.655	$MPa\sqrt{m}$
ความต้านทานการแตกหัก , $K_{0.2}$	199.191	$MPa\sqrt{m}$
ความต้านทานการแตกหัก , K_g	308.510	$MPa\sqrt{m}$
อัตราการเปลี่ยนแปลงของรอยร้าว , Δa_g	0.002	m

รายละเอียดของรอยร้าว

ความยาวของรอยร้าว , $2c$	0.100	m
ความลึกของรอยร้าว , a	0.020	m

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาระสถิตย์

แรงดึงแนวแกน , P 25 MN

6.2.1.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และระดับที่ 2

หลังจากนำข้อมูลป้อนให้กับโปรแกรม (ภาคผนวก ค.) แล้วทำการประเมินตามขั้นตอน จะได้ผลการประเมินแสดงอยู่ในตารางที่ 6.10 ตารางคอลัมน์แรกแสดงขั้นตอนการประเมิน(คำนวณ) คอลัมน์ที่สองแสดงผลการคำนวณ ประกอบด้วย ผลการประเมินระดับที่ 1 ระดับที่ 2 และหน่วยของผลการคำนวณ เมื่อนำผลการประเมินไปเปรียบเทียบกับผลการประเมินที่ได้จากการใช้โปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.2.1.1และ ข.2.1.2) พบว่า ผลการคำนวณของแต่ละขั้นตอนมีค่าเท่ากัน ผู้วิจัยจึงสรุปว่า โปรแกรมที่ออกแบบให้ผลการประเมินที่ถูกต้อง

ตารางที่ 6.10 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระแรงดึงแนวแกน (FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และ2)

ขั้นตอน	ผลการประเมิน		หน่วย
	ระดับที่ 1	ระดับที่ 2	
$P_L(a_0)$	68.823	68.823	MN
$L_r(a_o)$	0.363	0.363	-
$K_I(a_0)$	29.285	29.285	$MPa\sqrt{m}$
$K_r(a_o)$	0.491	0.147	-
$L_{rc}(a_0)$	0.667	1.093	-
$K_{rc}(a_0)$	0.901	0.442	-
$F_L(a_0)$	1.836	3.009	-
$P_L(a_g)$	-	68.713	MN
$L_r(a_g)$	-	0.364	-
$K_I(a_g)$	-	30.576	$MPa\sqrt{m}$
$K_r(a_g)$	-	0.099	-
$L_{rc}(a_g)$	-	1.194	-
$K_{rc}(a_g)$	-	0.325	-
$F_L(a_g)$	-	3.282	-
สถานะของโครงสร้าง	ปลอดภัย	ปลอดภัย	-

6.2.1.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และระดับที่ 2

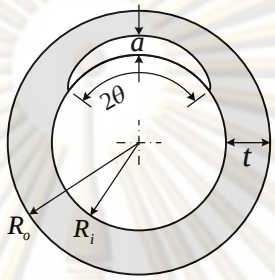
หลังจากนำข้อมูลป้อนให้กับโปรแกรม (ภาคผนวก ค.) แล้วทำการประเมินตามขั้นตอน จะได้ผลการประเมินแสดงอยู่ในตารางที่ 6.11ตารางคอลัมน์แรกแสดงขั้นตอนการประเมิน(คำนวณ) คอลัมน์ที่สองแสดงผลการคำนวณ ประกอบด้วย ผลการประเมินระดับที่ 1 ระดับที่ 2 และหน่วยของผลการคำนวณ เมื่อนำผลการประเมินไปเปรียบเทียบกับผลการประเมินที่ได้จากการใช้โปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.2.1.3 และ ข.2.1.4) พบว่า ผลการคำนวณของแต่ละขั้นตอนมีค่าเท่ากัน ผู้วิจัยจึงสรุปว่า โปรแกรมที่ออกแบบให้ผลการประเมินที่ถูกต้อง

ตารางที่ 6.11 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระแรงดึงแนวแกน (FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และ 2)

ขั้นตอน	ผลการประเมิน		หน่วย
	ระดับที่ 1	ระดับที่ 2	
$P_L(a_0)$	68.823	68.823	MN
$L_r(a_o)$	0.363	0.363	-
$K_I(a_0)$	29.285	29.285	$MPa\sqrt{m}$
$K_r(a_o)$	0.491	0.147	-
$L_{rc}(a_0)$	0.657	1.193	-
$K_{rc}(a_0)$	0.888	0.483	-
$F_L(a_0)$	1.808	3.285	
$P_L(a_g)$	-	68.713	MN
$L_r(a_g)$	-	0.364	-
$K_I(a_g)$	-	30.576	$MPa\sqrt{m}$
$K_r(a_g)$	-	0.099	-
$L_{rc}(a_g)$	-	1.270	-
$K_{rc}(a_g)$	-	0.346	-
$F_L(a_g)$	-	3.489	-
สถานะของโครงสร้าง	ปลอดภัย	ปลอดภัย	-

6.2.2 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระโมเมนต์ดัด

ท่อมีลักษณะการวางตัวของรอยร้าวและภาระดังรูปที่ 6.14 กรณีนี้ไม่สามารถประเมินการเติบโตของรอยร้าวเนื่องจากภาระล้าได้เนื่องจากไม่มีสมการผลเฉลยพารามิเตอร์ K ณ ตำแหน่งผิวของรอยร้าว



รูปที่ 6.14 ลักษณะของท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระโมเมนต์ดัด

6.2.2.1 ข้อมูล

ขนาดท่อ

รัศมีภายนอก , R_o	0.600	m
รัศมีภายใน , R_i	0.543	m

วัสดุ SA106Gr.C

มอดูลัสของความยืดหยุ่น , E	200.291	GPa
ความเค้นคราก , σ_y	269	MPa
ความเค้นดึงสูงสุด , σ_u	414	MPa
ค่าคงที่ของวัสดุ (บทที่ 3 สมการที่ (3.6)) , α	1	-
ค่าคงที่ของวัสดุ (บทที่ 3 สมการที่ (3.6)) , n	7.435	-
ความต้านทานการแตกหัก , K_{IC}	59.655	$MPa\sqrt{m}$
ความต้านทานการแตกหัก , $K_{0.2}$	199.191	$MPa\sqrt{m}$
ความต้านทานการแตกหัก , K_g	308.510	$MPa\sqrt{m}$
อัตราการเปลี่ยนแปลงของรอยร้าว , Δa_g	0.002	m

รายละเอียดของรอยร้าว

ความยาวของรอยร้าว , $2c$	0.100	m
ความลึกของรอยร้าว , a	0.020	m

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาระสถิตย์

โมเมนต์ดัด , M 29 $MN-m$

6.2.2.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และระดับที่ 2

หลังจากนำข้อมูลป้อนให้กับโปรแกรม (ภาคผนวก ค.) แล้วทำการประเมินตามขั้นตอน จะได้ผลการประเมินแสดงอยู่ในตารางที่ 6.12 ตารางคอลัมน์แรกแสดงขั้นตอนการประเมิน(คำนวณ) คอลัมน์ที่สองแสดงผลการคำนวณ ประกอบด้วย ผลการประเมินระดับที่ 1 ระดับที่ 2 และหน่วยของผลการประเมิน เมื่อนำผลการประเมินไปเปรียบเทียบกับผลการประเมินที่ได้จากการใช้โปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.2.2.1 และ ข.2.2.2) พบว่า ผลการคำนวณของแต่ละขั้นตอนมีค่าเท่ากัน ผู้วิจัยจึงสรุปว่า โปรแกรมที่ออกแบบให้ผลการประเมินที่ถูกต้อง

ตารางที่ 6.12 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระโมเมนต์ดัด (FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และ 2)

ขั้นตอน	ผลการประเมิน		หน่วย
	ระดับที่ 1	ระดับที่ 2	
$P_L(a_0)$	25.106	25.106	$MN-m$
$L_r(a_o)$	1.155	1.155	-
$K_I(a_0)$	144.775	144.775	$MPa\sqrt{m}$
$K_r(a_o)$	1.924	0.576	-
$L_{rc}(a_0)$	0.565	1.038	-
$K_{rc}(a_0)$	0.941	0.518	-
$F_L(a_0)$	0.489	0.899	-
$P_L(a_g)$	-	25.074	$MN-m$
$L_r(a_g)$	-	1.157	-
$K_I(a_g)$	-	120.654	$MPa\sqrt{m}$
$K_r(a_g)$	-	0.391	-
$L_{rc}(a_g)$	-	1.138	-
$K_{rc}(a_g)$	-	0.385	-
$F_L(a_g)$	-	0.984	-
สถานะของโครงสร้าง	ไม่ปลอดภัย	ไม่ปลอดภัย	-

6.2.2.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และระดับที่ 2

หลังจากนำข้อมูลป้อนให้กับโปรแกรม (ภาคผนวก ค.) แล้วทำการประเมินตามขั้นตอน จะได้ผลการประเมินแสดงอยู่ในตารางที่ 6.13 ตารางคอลัมน์แรกแสดงขั้นตอนการประเมิน(คำนวณ) คอลัมน์ที่สองแสดงผลการคำนวณ ประกอบด้วย ผลการประเมินระดับที่ 1 ระดับที่ 2 และหน่วยของผลการประเมิน เมื่อนำผลการประเมินไปเปรียบเทียบกับผลการประเมินที่ได้จากการใช้โปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.2.2.3 และ ข.2.2.4) พบว่า ผลการคำนวณของแต่ละขั้นตอนมีค่าเท่ากัน ผู้วิจัยจึงสรุปว่า โปรแกรมที่ออกแบบให้ผลการประเมินที่ถูกต้อง

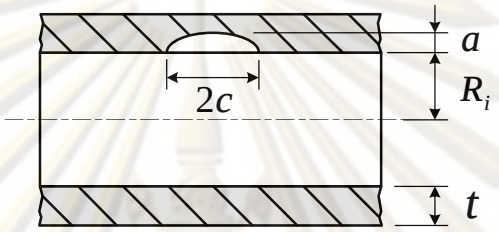
ตารางที่ 6.13 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระโมเมนต์ดัด (FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และ 2)

ขั้นตอน	ผลการประเมิน		หน่วย
	ระดับที่ 1	ระดับที่ 2	
$P_L(a_0)$	25.106	25.106	$MN-m$
$L_r(a_o)$	1.155	1.155	-
$K_I(a_0)$	144.775	144.775	$MPa\sqrt{m}$
$K_r(a_o)$	1.924	0.576	-
$L_{rc}(a_0)$	0.554	1.116	-
$K_{rc}(a_0)$	0.923	0.557	-
$F_L(a_0)$	0.48	0.966	-
$P_L(a_g)$	-	25.074	$MN-m$
$L_r(a_g)$	-	1.157	-
$K_I(a_g)$	-	120.654	$MPa\sqrt{m}$
$K_r(a_g)$	-	0.391	-
$L_{rc}(a_g)$	-	1.258	-
$K_{rc}(a_g)$	-	0.425	-
$F_L(a_g)$	-	1.088	-
สถานะของโครงสร้าง	ไม่ปลอดภัย	ปลอดภัย	-

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

6.2.3 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายใน

ท่อมีลักษณะการวางตัวของรอยร้าวและภาระดังรูปที่ 6.15 กรณีนี้สามารถประเมินการเติบโตของรอยร้าวเนื่องจากภาระล้าได้เนื่องจากมีสมการผลเฉลยพารามิเตอร์ K ณ ตำแหน่งผิวของรอยร้าว และตำแหน่งลึกสุด



รูปที่ 6.15 ลักษณะของท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายใน

6.2.3.1 ข้อมูล

ขนาดท่อ

รัศมีภายนอก , R_o	0.600	m
รัศมีภายใน , R_i	0.543	m

วัสดุ SA 106 Gr.C

มอดูลัสของความยืดหยุ่น , E	200.219	GPa
ความเค้นคราก , σ_y	269	MPa
ความเค้นดึงสูงสุด , σ_u	414	MPa
ค่าคงที่ของวัสดุ (บทที่ 3 สมการที่ (3.6)), α	1	-
ค่าคงที่ของวัสดุ (บทที่ 3 สมการที่ (3.6)), n	7.435	-
ความต้านทานการแตกหัก , K_{IC}	59.655	$MPa\sqrt{m}$
ความต้านทานการแตกหัก , $K_{0.2}$	199.191	$MPa\sqrt{m}$
ความต้านทานการแตกหัก , K_g	308.51	$MPa\sqrt{m}$
อัตราการเปลี่ยนแปลงของรอยร้าว , Δa_g	0.002	m
ค่าคงที่ของวัสดุ (บทที่ 2 สมการที่ (2.12)) , C	1×10^{-9}	$m/cycle$
ค่าคงที่ของวัสดุ (บทที่ 2 สมการที่ (2.12)) , m	3	-

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายละเอียดของรอยร้าว

ความยาวของรอยร้าว , $2c$	0.100	m
ความลึกของรอยร้าว , a	0.020	m
ภาระสถิต		
ความดันภายใน , P	20	MPa
ภาระเปลี่ยนแปลง		
ความดันภายในสูงสุด , P_{max}	4	MPa
ความดันภายในต่ำสุด , P_{min}	0	MPa
จำนวนรอบภาระ	4000	$cycle$

6.2.3.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และระดับที่ 2

หลังจากนำข้อมูลป้อนให้กับโปรแกรม (ภาคผนวก ค.) แล้วแล้วทำการประเมินตามขั้นตอน จะได้ผลการประเมินแสดงอยู่ในตารางที่ 6.14 กรณีไม่ประเมินการเติบโตของรอยร้าว และตารางที่ 6.15 กรณีประเมินการเติบโตของรอยร้าว(ภาคผนวก ข.3) ได้รอยร้าวมีความยาวเท่ากับ 0.024 m และความลึกเท่ากับ 0.103 m เมื่อนำผลการประเมินที่ได้จากตารางทั้งสองไปเปรียบเทียบกับผลการประเมินที่ได้จากการใช้โปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.2.3.1 และ ข.2.3.2) พบว่า ผลการคำนวณของแต่ละขั้นตอนมีค่าเท่ากัน ผู้วิจัยจึงสรุปว่า โปรแกรมที่ออกแบบให้ผลการประเมินที่ถูกต้อง

ตารางที่ 6.14 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายใน(FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และ 2)

ขั้นตอน	ผลการประเมิน		หน่วย
	ระดับที่ 1	ระดับที่ 2	
$P_L(a_0)$	31.480	31.480	MPa
$L_r(a_o)$	0.635	0.635	-
$K_I(a_0)$	49.400	49.400	$MPa\sqrt{m}$
$K_r(a_o)$	0.828	0.248	-
$L_{rc}(a_0)$	0.685	1.102	-
$K_{rc}(a_0)$	0.892	0.430	-

ตารางที่ 6.14 (ต่อ)

ขั้นตอน	ผลการประเมิน		หน่วย
	ระดับที่ 1	ระดับที่ 2	
$F_L(a_0)$	1.077	1.735	-
$P_L(a_g)$	-	31.328	MPa
$L_r(a_g)$	-	0.638	-
$K_I(a_g)$	-	50.712	$MPa\sqrt{m}$
$K_r(a_g)$	-	0.164	-
$L_{rc}(a_g)$	-	1.209	-
$K_{rc}(a_g)$	-	0.312	-
$F_L(a_g)$	-	1.894	-
สถานะของโครงสร้าง	ปลอดภัย	ปลอดภัย	-

ตารางที่ 6.15 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายในและภาระความดันเปลี่ยนแปลง (FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1 และ 2)

ขั้นตอน		ผลการประเมิน		หน่วย
		ระดับที่ 1	ระดับที่ 2	
ก่อนประเมินการเติบโตของรอยร้าว	$P_L(a_0)$	31.480	31.480	MPa
	$L_r(a_o)$	0.635	0.635	-
	$K_I(a_0)$	49.400	49.400	$MPa\sqrt{m}$
	$K_r(a_o)$	0.828	0.248	-
	$P_L(a_g)$	-	31.328	MPa
	$L_r(a_g)$	-	0.638	-
	$K_I(a_g)$	-	50.712	$MPa\sqrt{m}$
	$K_r(a_g)$	-	0.164	-
	$P_L(a_0)$	31.068	31.068	MPa

ตารางที่ 6.15 (ต่อ)

ขั้นตอน		ผลการประเมิน		หน่วย
		ระดับที่ 1	ระดับที่ 2	
หลังประเมินการเติบโตของรอยร้าว	$L_r(a_o)$	0.644	0.644	-
	$K_I(a_0)$	52.736	52.736	$MPa\sqrt{m}$
	$K_r(a_o)$	0.884	0.265	-
	$L_{rc}(a_0)$	0.659	1.089	-
	$K_{rc}(a_0)$	0.905	0.448	-
	$F_L(a_0)$	1.024	1.691	-
	$P_L(a_g)$	-	30.865	MPa
	$L_r(a_g)$	-	0.648	-
	$K_I(a_g)$	-	53.965	$MPa\sqrt{m}$
	$K_r(a_g)$	-	0.175	-
	$L_{rc}(a_g)$	-	1.196	-
	$K_{rc}(a_g)$	-	0.323	-
	$F_L(a_g)$	-	1.847	-
สถานะของโครงสร้าง		ปลอดภัย	ปลอดภัย	-

6.2.3.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และระดับที่ 2

หลังจากนำข้อมูลป้อนให้กับโปรแกรม (ภาคผนวก ค.) แล้วทำการประเมินตามขั้นตอน จะได้ผลการประเมินแสดงอยู่ในตารางที่ 6.16 กรณีไม่ประเมินการเติบโตของรอยร้าว และตารางที่ 6.17 กรณีประเมินการเติบโตของรอยร้าว(ภาคผนวก ข.3) ได้รอยร้าวมีความยาวเท่ากับ 0.024 m และความลึกเท่ากับ 0.103 m เมื่อนำผลการประเมินที่ได้จากตารางทั้งสองไปเปรียบเทียบกับผลการประเมินที่ได้จากการใช้โปรแกรมช่วยคำนวณ MathCAD (ภาคผนวก ข.2.3.3 และ ข. 2.3.4) พบว่า ผลการคำนวณของแต่ละขั้นตอนมีค่าเท่ากัน ผู้วิจัยจึงสรุปว่า โปรแกรมที่ออกแบบให้ผลการประเมินที่ถูกต้อง

ตารางที่ 6.16 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายใน (FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และ 2)

ขั้นตอน	ผลการประเมิน		หน่วย
	ระดับที่ 1	ระดับที่ 2	
$P_L(a_0)$	31.480	31.480	MPa
$L_r(a_o)$	0.635	0.635	-
$K_I(a_0)$	49.400	49.400	$MPa\sqrt{m}$
$K_r(a_o)$	0.828	0.248	-
$L_{rc}(a_0)$	0.675	1.106	-
$K_{rc}(a_0)$	0.880	0.471	-
$F_L(a_0)$	1.063	1.899	-
$P_L(a_g)$	-	31.328	MPa
$L_r(a_g)$	-	0.638	-
$K_I(a_g)$	-	50.712	$MPa\sqrt{m}$
$K_r(a_g)$	-	0.164	-
$L_{rc}(a_g)$	-	1.270	-
$K_{rc}(a_g)$	-	0.327	-
$F_L(a_g)$	-	1.989	-
สถานะของโครงสร้าง	ปลอดภัย	ปลอดภัย	-

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 6.17 ผลการประเมินการคงสภาพของท่อที่มีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายในและภาระความดันเปลี่ยนแปลง (FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1 และ 2)

ขั้นตอน		ผลการประเมิน		หน่วย
		ระดับที่ 1	ระดับที่ 2	
ก่อนประเมินการเติบโตของรอยร้าว	$P_L(a_0)$	31.480	31.480	MPa
	$L_r(a_o)$	0.635	0.635	-
	$K_I(a_0)$	49.400	49.400	$MPa\sqrt{m}$
	$K_r(a_o)$	0.828	0.248	-
	$P_L(a_g)$	-	31.328	MPa
	$L_r(a_g)$	-	0.638	-
	$K_I(a_g)$	-	50.712	$MPa\sqrt{m}$
	$K_r(a_g)$	-	0.164	-
หลังประเมินการเติบโตของรอยร้าว	$P_L(a_0)$	31.068	31.068	MPa
	$L_r(a_o)$	0.644	0.644	-
	$K_I(a_0)$	52.736	52.736	$MPa\sqrt{m}$
	$K_r(a_o)$	0.884	0.265	-
	$L_{rc}(a_0)$	0.649	1.187	-
	$K_{rc}(a_0)$	0.891	0.488	-
	$F_L(a_0)$	1.008	1.844	-
	$P_L(a_g)$	-	30.868	MPa
	$L_r(a_g)$	-	0.648	-
	$K_I(a_g)$	-	53.965	$MPa\sqrt{m}$
	$K_r(a_g)$	-	0.175	-
	$L_{rc}(a_g)$	-	1.270	-
	$K_{rc}(a_g)$	-	0.343	-
	$F_L(a_g)$	-	1.959	-
สถานะของโครงสร้าง		ปลอดภัย	ปลอดภัย	-

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 7

สรุปผล

ในบทนี้จะกล่าวถึงการสรุปผลการวิจัยในหัวข้อที่ 7.1 จะสรุปเกี่ยวกับการออกแบบและผลการตรวจสอบ หัวข้อที่ 7.2 ข้อจำกัดของโปรแกรม และหัวข้อที่ 7.3 แนะนำงานวิจัยต่อเนื่อง

7.1 สรุปเกี่ยวกับการออกแบบและผลตรวจสอบโปรแกรม

การออกแบบโปรแกรมประยุกต์ตามวิธี James Rumbaugh et al. โดยมีวิธีการ คือ การเขียนเนื้อความของปัญหา ถัดไปเป็นการนำเนื้อความของปัญหามาวิเคราะห์หาคลาส จากนั้นจึงนำคลาสที่ได้มาสร้างเป็นคลาสไดอะแกรม ต่อมาจึงปรับปรุงคลาสไดอะแกรมด้วยการแนวคิดการสืบทอด และการจำลองการทำงานตามขั้นตอนด้วยการนำคลาสมาสร้างเป็น sequence diagram และสุดท้ายเป็นการการเพิ่มเติมรายละเอียด (แอตทริบิวต์และเมทธอด) ให้กับคลาสเพื่อให้สามารถนำไปเขียนเป็นโปรแกรมได้ง่าย การเขียนโปรแกรมผู้วิจัยเลือกใช้ภาษา Delphi เนื่องจากเป็นภาษาที่รองรับการเขียนด้วยแนวคิดเชิงวัตถุ โปรแกรมที่ออกแบบยังมีข้อจำกัดบางอย่างซึ่งจะอธิบายในหัวข้อที่ 7.2

การตรวจสอบโปรแกรมทำโดยการสร้างโจทย์ขึ้นมาแล้วคำนวณด้วยโปรแกรมเปรียบเทียบผลการคำนวณด้วยโปรแกรม MathCAD การตรวจสอบแบ่งเป็นสองส่วน คือ ส่วนการระบุลักษณะรอยร้าว และ ส่วนการประเมิน สำหรับส่วนการระบุลักษณะรอยร้าวมีโจทย์ดังนี้คือ

1) ระบุลักษณะของรอยร้าวกรณีรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย
2) ระบุลักษณะของรอยร้าวกรณีรอยร้าวฝังรูปวงรีทั้งสองรอย และ
3) ระลักษณะของรอยร้าวกรณีรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงรี

สำหรับส่วนการประเมินมีโจทย์ดังนี้คือ

1) ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระดึงตามแนวแกน
2) ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระโมเมนต์ดัด
3) ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายใน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ผลการตรวจสอบปรากฏว่า ผลการระบุลักษณะของรอยร้าวและผลการประเมินในโจทย์ทั้งหมดของโปรแกรมที่ออกแบบมีความถูกต้อง

7.2 ข้อจำกัดของโปรแกรม

ข้อจำกัดของโปรแกรมที่ออกแบบมีดังนี้ คือ

1) ข้อจำกัดของโครงสร้าง โปรแกรมสามารถประเมินการคงสภาพของท่อตรงเท่านั้น

2) ข้อจำกัดของสมการที่ใช้หาพารามิเตอร์ K และภาระขีดจำกัด

ข้อจำกัดของสมการที่ใช้หาพารามิเตอร์ K ณ ตำแหน่งผิวหน้ารอยร้าวทำให้การคำนวณการเติบโตของรอยร้าวกรณีที่ต้องการทราบว่าโครงสร้างสามารถใช้งานได้อีกนานเท่าใดนั้น ไม่สามารถทำได้กับกรณีดังนี้ คือ

ก) ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระดึงตามแนวแกน

ข) ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระโมเมนต์ดัด

ค) ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับความเค้นกระจายไม่สม่ำเสมอบนผิวหน้ารอยร้าว

ข้อจำกัดของสมการที่ใช้หาภาระขีดจำกัดทำให้ในกรณีที่ ค) ไม่สามารถประเมินการคงสภาพของท่อได้ ส่วนในกรณีที่ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวงรับภาระความดันภายในนั้นไม่มีสมการที่ใช้หาพารามิเตอร์ K และภาระขีดจำกัด ทำให้ไม่สามารถประเมินการคงสภาพของท่อกรณีนี้ได้

3) ข้อจำกัดในการแสดงผลการประเมินในรูปแบบของแผนภาพ FAD เนื่องจากความยุ่งยากในการเขียนโปรแกรมเพื่อนำเสนอผลการประเมินในรูปแบบของกราฟ

4) ข้อจำกัดของประเภทของภาระ โปรแกรมไม่สามารถประเมินการคงสภาพในกรณีที่โครงสร้างรับภาระทุติยภูมิ

5) ข้อจำกัดของการระบุลักษณะของร้าวซ้ำ โปรแกรมไม่สามารถระบุลักษณะรอยร้าวกรณีท่อมีรอยร้าวอยู่ใกล้กันเกินสองรอยได้

6) ข้อจำกัดของการประเมินการเติบโตของรอยร้าว โปรแกรมไม่สามารถประเมินการเติบโตของรอยร้าวเนื่องจากสิ่งแวดล้อมได้

7) ข้อจำกัดของระดับการประเมินและชนิดของ FAD โปรแกรมไม่สามารถประเมินการคงสภาพด้วยการประเมินระดับที่ 3 และ FAD ชนิดที่ 3

8) ข้อจำกัดของการเลือกค่าความต้านทานการแตกหัก กรณีเลือกระดับการประเมินระดับที่ 1 โปรแกรมไม่สามารถเลือกได้ว่าจะใช้ความต้านทานการแตกหักตัวไหน

7.3 แนะนำงานวิจัยต่อเนื่อง

จากความเข้าใจที่ได้จากการออกแบบโปรแกรมมีประเด็นที่น่าสนใจสำหรับการทำวิจัยต่อเนื่องได้

1) ปรับปรุงเพื่อเพิ่มขีดความสามารถ เช่น การระบุลักษณะของรอยร้าวกรณีที่โครงสร้างมีรอยร้าวอยู่ใกล้กันมากกว่าสองรอย การประเมินกับโครงสร้างและรอยร้าวชนิดชนิดอื่น ๆ การประเมินโดยใช้ระดับการประเมินและ FAD ที่เพิ่มขึ้น เป็นต้น

2) เป็นแนวทางในออกแบบโปรแกรมสำหรับการประเมินการคงสภาพของโครงสร้างด้วยระเบียบวิธีอื่น ๆ ที่ความเสียหายมีผลจากอุณหภูมิได้ เช่น ระเบียบวิธี R5 เป็นต้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] Wikipedia. List of Pipeline Accidents [Online]. Available from: htpp://www.en.wikipedia.org/wiki/List_of_pipeline_accidents#Russia.htm [2007, Jan 5]

[2] Wikipedia. List of Pipeline Accidents [Online]. Available from: htpp://www.en.wikipedia.org/wiki/List_of_pipeline_accidents#United states.htm [2007, Jan 5]

[3] Wikipedia. List of Pipeline Accidents [Online]. Available from: htpp://www.en.wikipedia.org/wiki/List_of_pipeline_accidents.htm [2007, Jan 5]

[4] Ainsworth, R.A., Schwalbe, K.-H., Milne, I., Ritchie, R.O., and Karihaloo, B. Comprehensive Structural Integrity. vol.7. Oxford: Elsevier, 2003.

[5] Milne, I., Ainsworth, R.A., Dowling, A.R., and Stewart, A.T. Assessment of the Integrity of Structures Containing Defects. International Journal of Pressure Vessels and Piping 32 (1988): 4-104.

[6] Budden, P.J., Sharples, J.K., and Dowling, A.R. The R6 Procedure: Recent Developments and Comparison with Alternative Approaches. International Journal of Pressure Vessels and Piping 77 (2000): 895-903.

[7] American Petroleum Institute. Fitness for service: API Recommended Practice 579. 1 st. Washington: NATL PETROCHEMICAL PUBLICATION, 2000

[8] Structure Integrity Assessment . SINTAP: Structure Integrity Assessment Procedure for European Industry. Final Procedure. (n.p.), 1999.

[9] Chattopadhyay, J., Pavankumar, T.V., Dutta, B.K., and Kushwaha, H.S. Fracture Experiments on Through Wall Cracked Elbows Under in-plane Bending Moment: Test Results and Theoretical Analyses. Engineering Fracture Mechanics 72 (2005): 1461-1497.

[10] Marshall, G.W., and Ainsworth, R.A. The Fracture Behavior of a Welded Tubular Joint-an ESIS TCI-3 Round Robin on Failure Assessment Methods Part II: R6 Analysis. Engineering Fracture Mechanics 69 (2002): 1111-1118.

[11] Fitness 4 service [Online]. Available from: htpp://www.Fitness4service.com/software/Rcode.htm [2007, Jan 15]

[12] Brickstad, B., Bergman, M., Andersson, P., Dahlberg, L., Sattari-Far, I., and Nilsson, F. Procedures Used in Sweden for Safety Assessment of Components With Cracks. International Journal of Pressure Vessels and Piping 77 (2000): 877-881.

[13] Kobayashi, H., Sakai, S., Asano, M., Miyazaki, K., Nagasaki, T., and Takahashi, Y. Development of a Flaw Evaluation Handbook of the High Pressure Institute of Japan. International Journal of Pressure Vessels and Piping 77 (2000): 929-936.

[14] Zarko Gajic. Learning Object Oriented Programming with Delphi [Online]. Available from: http://www.delphi.about.com/od/course/a/oop_intro.htm [2007, Jan 25]

[15] Anderson, T.L. Fracture Mechanics: Fundamentals and Application. USA: CRC Press, 1991.

[16] Lethbridge, T. C., and Laganiere, R. Object-Oriented Software Engineering. Singapore: McGrawHill, 2005.

[17] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. Object –Oriented Modeling and Design. USA: Prentice-Hall, 1991.

[18] Japan Welding Engineering Society Standard. Method of Assessment for Flaw in Fusion Welded Joints with respect to Brittle Fracture and Fatigue Crack Growth. Japan: 1997.

[19] Kim, Y. J., Seok, C. S., and Chang, Y. S. Prediction of fracture resistance curves from tensile data for carbon steel piping materials. International Journal of Pressure Vessels and Piping 68 (1996): 209-217.

[20] Zahoor, A., Ductile Fracture Handbook. Vols. 1-3. Palo Alto, California: Electric Power Research Institute, 1989.

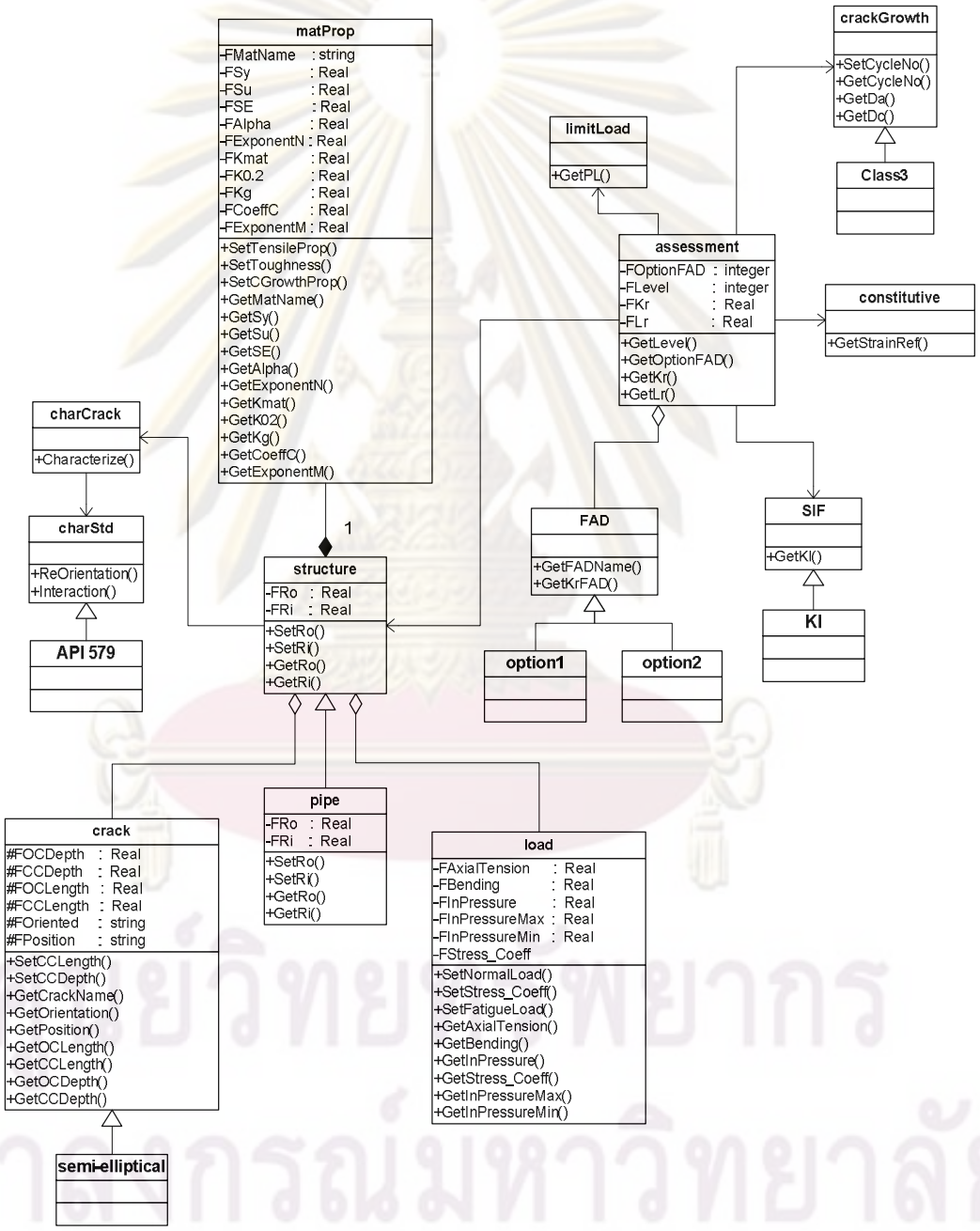
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

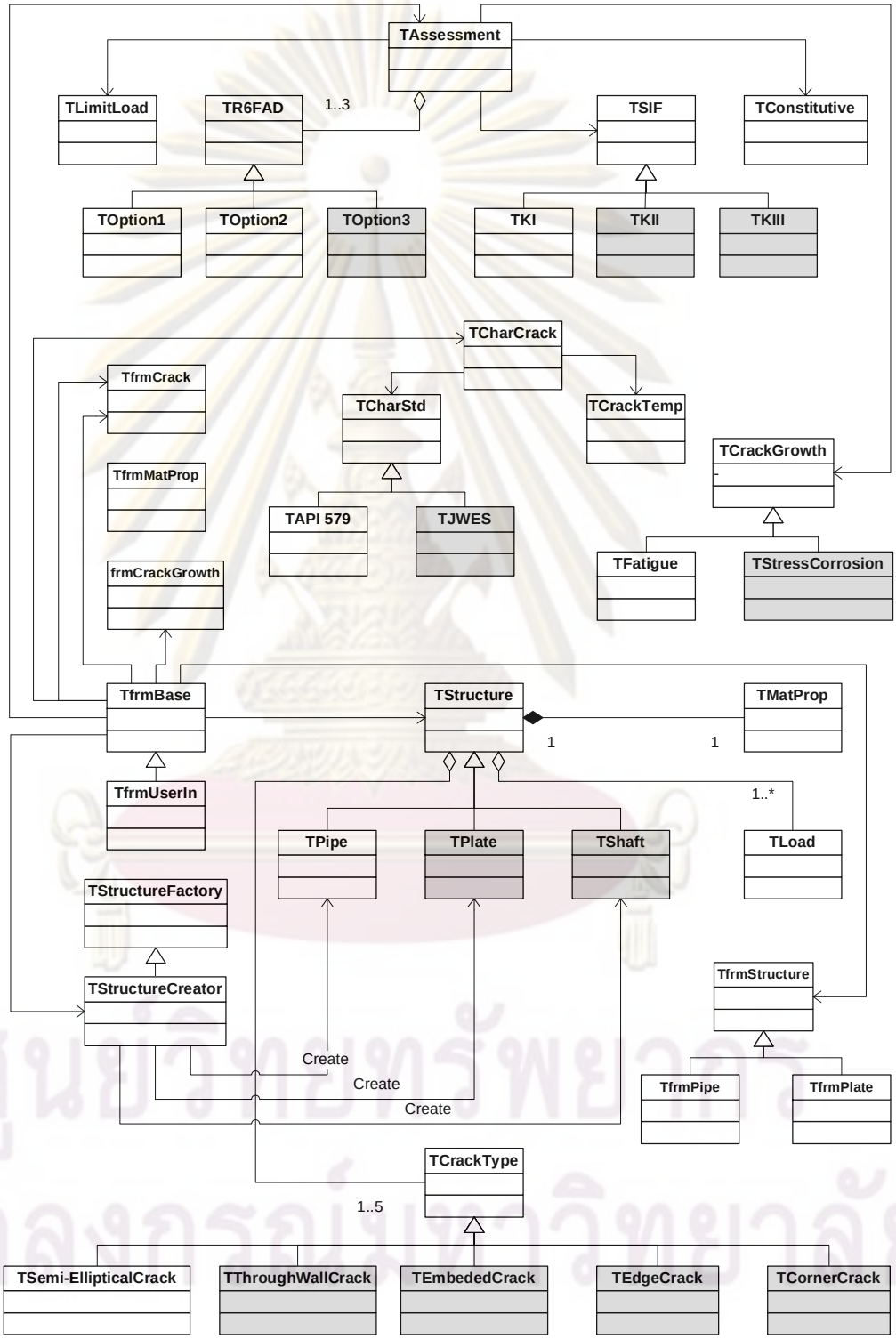
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก.

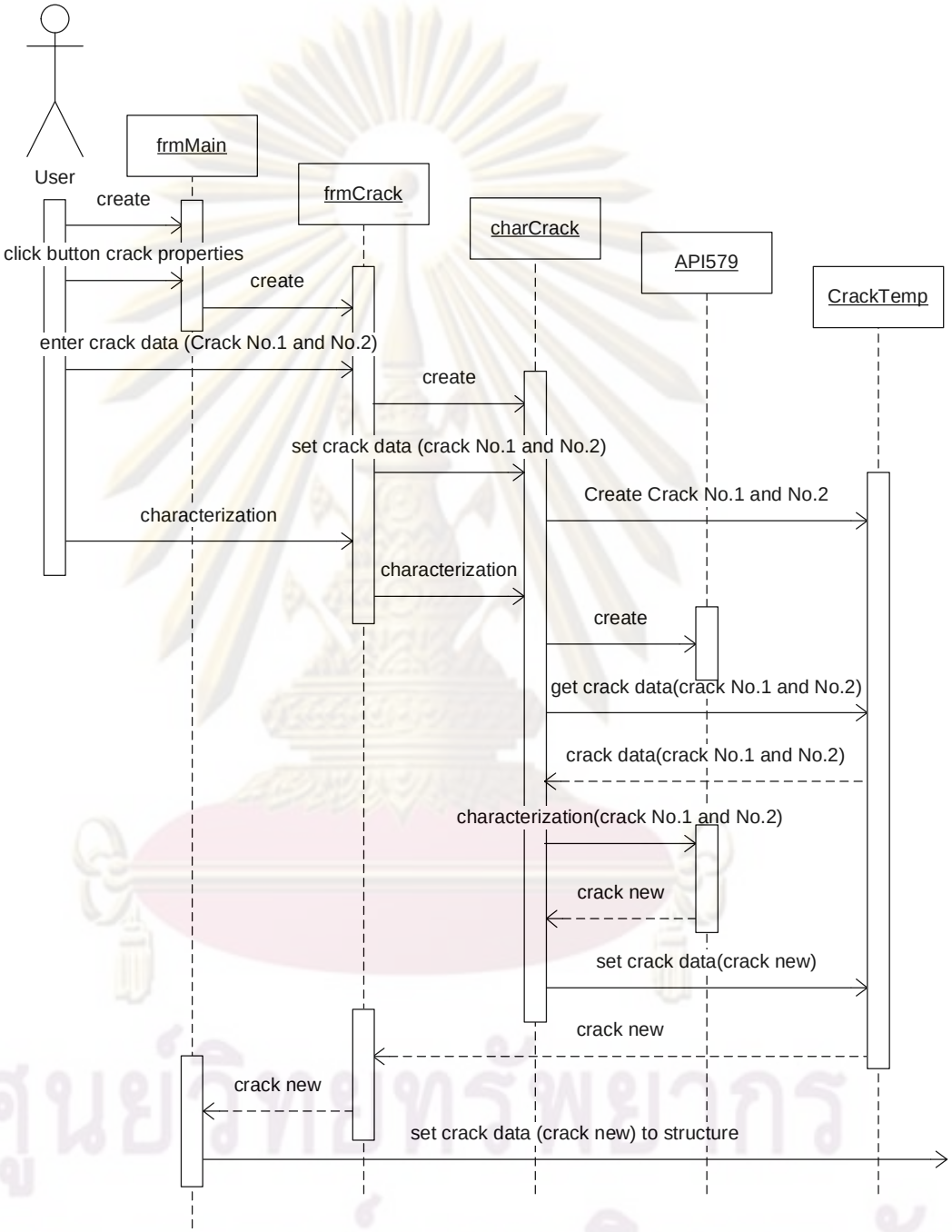
ก.1 คลาสไดอะแกรม



ก.2 คลาสไดอะแกรมที่รวมส่วนติดต่อกับผู้ใช้งานแล้ว

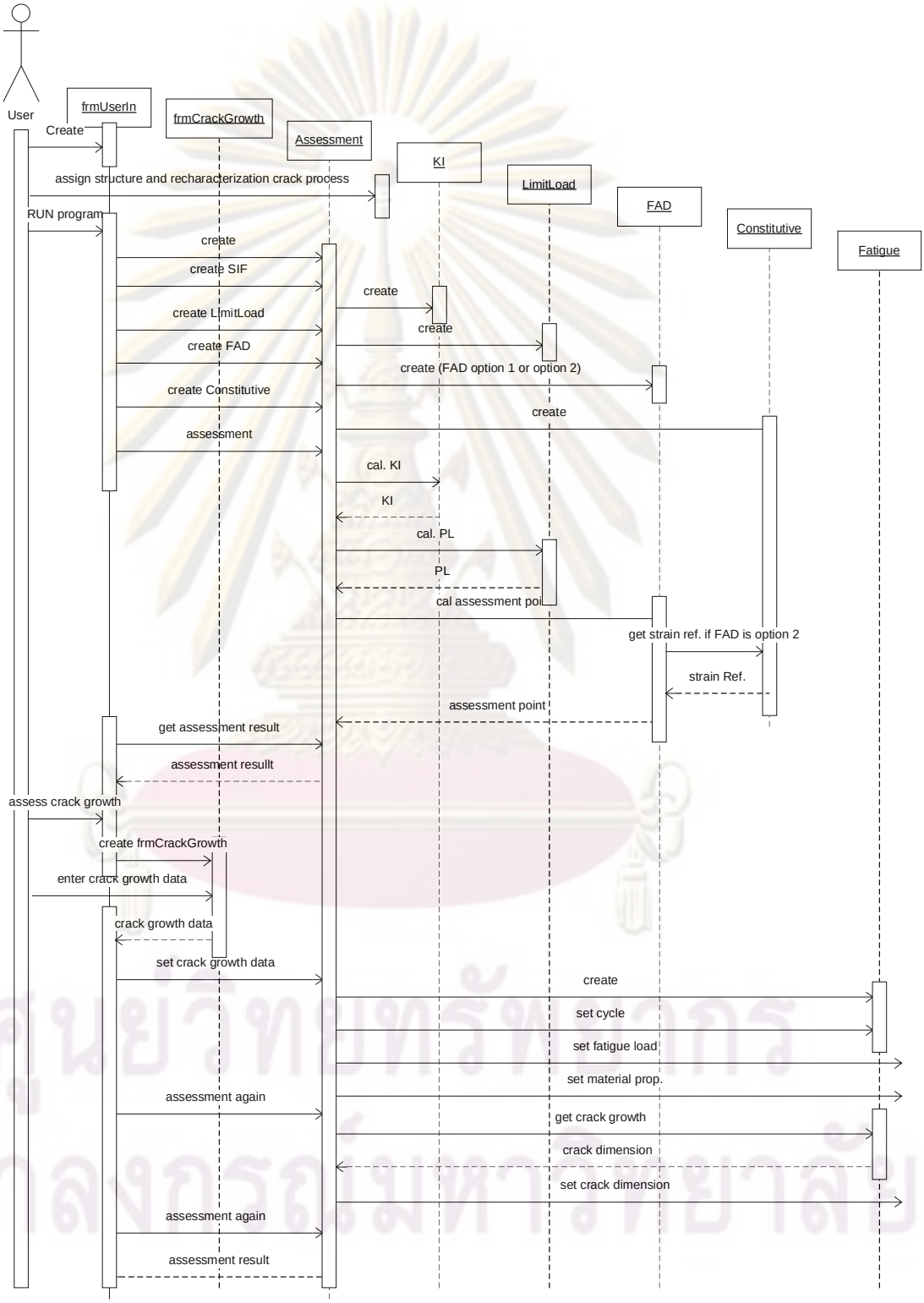


ก.3 sequence diagram **ของการระบุลักษณะรอยร้าว**



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ก.4 sequence diagram ของการประเมิน



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข.

รายละเอียดการคำนวณด้วยโปรแกรมMathCAD

ข.1 การระบุลักษณะรอยร้าว

ข.1.1 การระบุลักษณะรอยร้าวกรณีรอยร้าวผิวรูปครึ่งวงรีทั้งสองรอย

No. of Crack $n_{crack} := 2$ $i := 1 .. n_{crack}$ $MPa := 10^6 Pa$

$ORIGIN := 1$

Crack Data $type := \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ 1 : semi-elliptical

2 : emeded

Crack Position $x_M := \begin{pmatrix} 0 \\ 40 \end{pmatrix}$ $y_M := \begin{pmatrix} 0 \\ 20 \end{pmatrix}$ $z_M := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

inclined angle α $\alpha := \begin{pmatrix} 4 \\ 2 \end{pmatrix} deg$

inclined angle θ $\theta := \begin{pmatrix} 5 \\ 15 \end{pmatrix}$

(Half) crack length $c_o := \begin{pmatrix} 20 \\ 10 \end{pmatrix}$ $2c_o = \begin{pmatrix} 40 \\ 20 \end{pmatrix}$

crack depth $a_o := \begin{pmatrix} 10 \\ 5 \end{pmatrix}$

crack position

$$y_{N_i} := \sqrt{\left[\left(c_{o_i}\right)^2 - \left(c_{o_i} \cdot \cos\left(\alpha_i\right)\right)^2\right]} \quad x_{N_i} := \sqrt{\left[\left(c_{o_i}\right)^2 - \left(y_{N_i}\right)^2\right]}$$

$x_{i,1} := x_{M_i} - x_{N_i}$ $x_{i,2} := x_{M_i} + x_{N_i}$ $x_{i,3} := x_{M_i}$ $x_{i,4} := x_{M_i}$

$y_{i,1} := y_{M_i} - y_{N_i}$ $y_{i,2} := y_{M_i} + y_{N_i}$ $y_{i,3} := y_{M_i}$ $y_{i,4} := y_{M_i}$

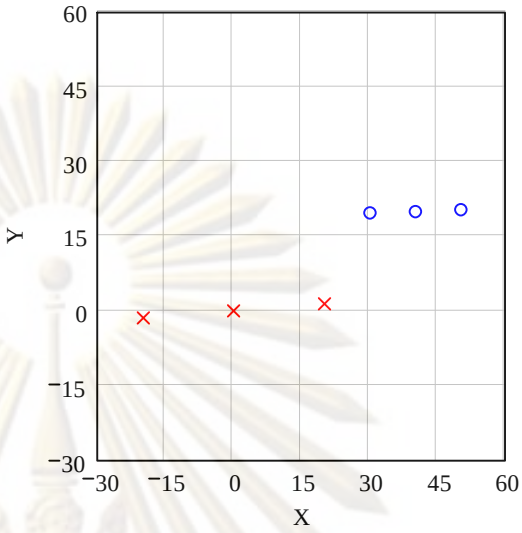
$z_{i,1} := z_{M_i}$ $z_{i,2} := z_{M_i}$ $z_{i,3} := z_{M_i}$ $z_{i,4} := -a_{o_i}$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$x = \begin{pmatrix} -19.95 & 19.95 & 0 & 0 \\ 30.01 & 49.99 & 40 & 40 \end{pmatrix}$$

$$y = \begin{pmatrix} -1.4 & 1.4 & 0 & 0 \\ 19.65 & 20.35 & 20 & 20 \end{pmatrix}$$

$$z = \begin{pmatrix} 0 & 0 & 0 & -10 \\ 0 & 0 & 0 & -5 \end{pmatrix}$$



1) Adjust (half) crack length

$$B := 0.5$$

$$c_i := \begin{cases} \left[\cos(\alpha_i)^2 + \frac{(1 - B)\cdot\sin(\alpha_i)\cdot\cos(\alpha_i)}{2} + B^2\cdot\sin(\alpha_i)^2\right]\cdot c_{0_i} & \text{if } |\alpha_i| \le 45\text{deg} \vee B = 0 \\ \left[\frac{\cos(\alpha_i)^2}{B^2} + \frac{(1 - B)\cdot\sin(\alpha_i)\cdot\cos(\alpha_i)}{2\cdot B^2} + \sin(\alpha_i)^2\right]\cdot c_{0_i} & \text{if } |\alpha_i| > 45\text{deg} \end{cases}$$

$$c^T = (20.27 \quad 10.08) \qquad 2\,c^T = (40.55 \quad 20.16)$$

2) Adjust crack depth

$$w_i := 0.99999 + 1.0481\cdot10^{-5}\cdot\theta_i + 1.5471\cdot10^{-4}\cdot(\theta_i)^2 + 3.4141\cdot10^{-5}\cdot(\theta_i)^3 - 2.0688\cdot10^{-6}\cdot(\theta_i$$
$$+ 4.4977\cdot10^{-8}\cdot(\theta_i)^5 - 4.5751\cdot10^{-10}\cdot(\theta_i)^6 + 1.822\cdot10^{-12}\cdot(\theta_i)^7$$

{θ in deg}

$$w = \begin{pmatrix} 1.007 \\ 1.075 \end{pmatrix}$$

$$a_i := a_{0_i}\cdot\cos(\theta_i\cdot\text{deg})\cdot\max(w_i, 1) \qquad a^T = (10.03 \quad 5.19)$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3) Determine Crack Coordinates

$x_{i,1} := x_{M_i} - c_i$ $x_{i,2} := x_{M_i} + c_i$ $x_{i,3} := x_{M_i}$ $x_{i,4} := x_{M_i}$

$x_{i,1} =$

-20.27
29.92

$x_{i,2} =$

20.27
50.08

$x_{i,3} =$

0
40

$x_{i,4} =$

0
40

$y_{i,1} := y_{i,3}$ $y_{i,2} := y_{i,3}$ $y_{i,3} := y_{i,3}$ $y_{i,4} := y_{i,3}$

$y_{i,1} =$

0
20

$y_{i,2} =$

0
20

$y_{i,3} =$

0
20

$y_{i,4} =$

0
20

$z_{i,1} := z_{i,3}$ $z_{i,2} := z_{i,3}$ $z_{i,3} := z_{i,3}$ $z_{i,4} := -z_{i,3} - a_i$

$z_{i,1} =$

0
0

$z_{i,2} =$

0
0

$z_{i,3} =$

0
0

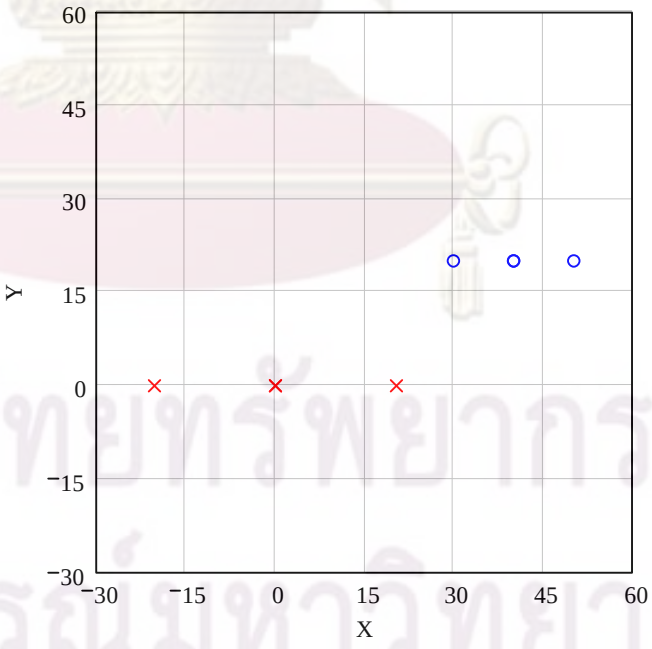
$z_{i,4} =$

-10.03
-5.19

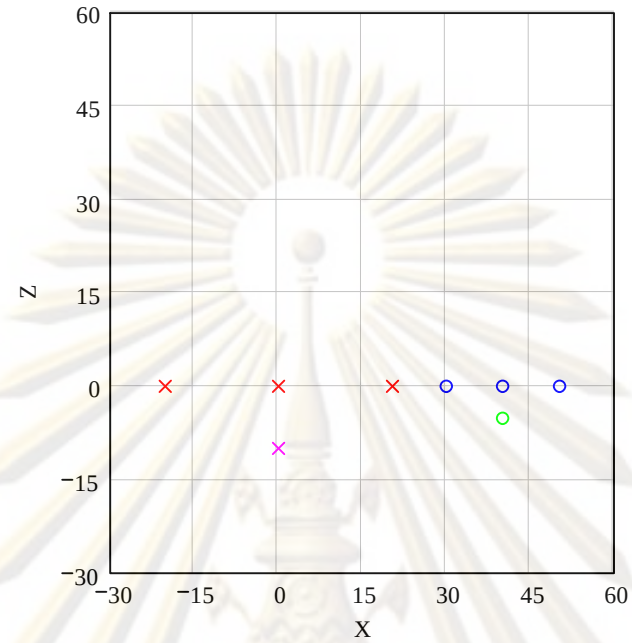
$$x = \begin{pmatrix} -20.27 & 20.27 & 0 & \\ 29.92 & 50.08 & 40 & \end{pmatrix}$$

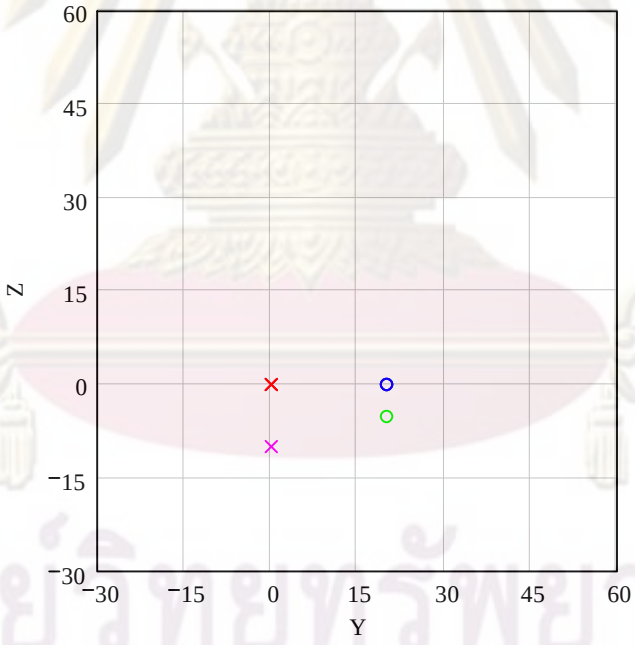
$$y = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 20 & 20 & 20 & 20 \end{pmatrix}$$

$$z = \begin{pmatrix} 0 & 0 & 0 & -10.03 \\ 0 & 0 & 0 & -5.19 \end{pmatrix}$$



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย





4). Calculate Planar distance between crack

$S1 := \max\left(y_{2,3}, y_{1,3}\right) - \min\left(y_{2,3}, y_{1,3}\right)$

$S1 = 20$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5). Calculate Planar overlap length

$$S2 := \max(x_{2,2}, x_{1,2}) - \min(x_{2,1}, x_{1,1}) - |x_{2,2} - x_{2,1}| - |x_{1,2} - x_{1,1}|$$

$$S2 = 9.65$$

6). Calculate Thickness overlap length

$$S3 := |\min(z_{2,4}, z_{1,4})| - |\max(z_{2,3}, z_{1,3})| - |z_{2,4} - z_{2,3}| - |z_{1,4} - z_{1,3}|$$

$$S3 = -5.19$$

$$c^T = (20.27 \quad 10.08) \qquad a^T = (10.03 \quad 5.19)$$

$$c_{sum} := c_1 + c_2 \qquad a_{sum} := a_1 + a_2$$

$$c_{sum} = 30.35 \qquad a_{sum} = 15.22$$

$$c_{eff} := \begin{cases} \left[0.5 \cdot (\max(x_{2,2}, x_{1,2}) - \min(x_{1,1}, x_{2,1}))\right] & \text{if } c_{sum} \geq S1 \wedge c_{sum} \geq S2 \\ \max(c_1, c_2) & \text{otherwise} \end{cases}$$

$$c_{eff} = 35.18$$

$$a_{eff} := \begin{cases} |\min(z_{2,4}, z_{1,4})| - |\max(z_{2,3}, z_{1,3})| & \text{if } c_{sum} \geq S1 \wedge c_{sum} \geq S2 \\ |\max(z_{2,4}, z_{1,4})| & \text{otherwise} \end{cases}$$

$$a_{eff} = 10.03$$

Locate the position of an effective crack

$$x_{1,1} := \min(\min(x_{1,1}, x_{2,1}), \min(x_{2,2}, x_{1,2})) \qquad x_{1,2} := \max(\max(x_{2,2}, x_{1,2}), \max(x_{1,1}, x_{2,1}))$$

$$x_{1,3} := 0.5 \cdot (x_{1,1} + x_{1,2}) \qquad x_{1,4} := x_{1,3}$$

$$y_{1,1} := 0.5 \cdot (y_{M_1} + y_{M_2}) \qquad y_{1,2} := y_{1,1}$$

$$y_{1,3} := y_{1,1} \qquad y_{1,4} := y_{1,3}$$

$$z_{1,1} := z_{M_1} \qquad z_{1,2} := z_{M_1}$$

$$z_{1,3} := z_{M_1} \qquad z_{1,4} := -a_{eff}$$

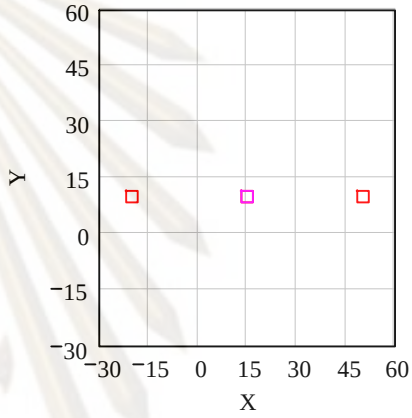
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

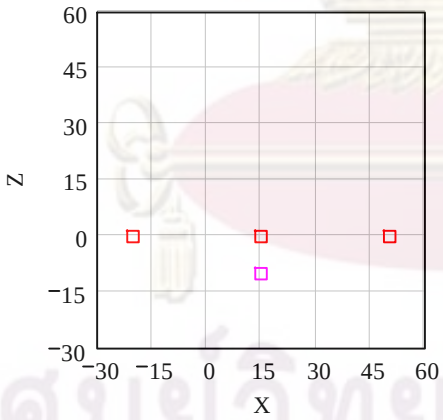
$n_{crack} := n_{crack} - 1$ $\quad i := 1 .. n_{crack}$

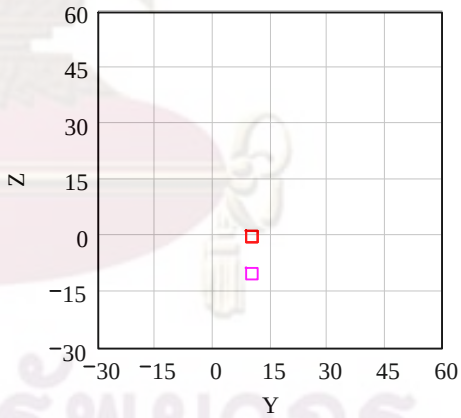
$x_{i,1}$ =	$x_{i,2}$ =	$x_{i,3}$ =	$x_{i,4}$ =
-20.27	50.08	14.9	14.9

$y_{i,1}$ =	$y_{i,2}$ =	$y_{i,3}$ =	$y_{i,4}$ =
10	10	10	10

$z_{i,1}$ =	$z_{i,2}$ =	$z_{i,3}$ =	$z_{i,4}$ =
0	0	0	-10.03







(Half) crack length $\quad c := c_{eff}$

$c = 35.18$ $\quad 2c = 70.35$

crack depth $\quad a := a_{eff}$

$a = 10.03$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.1.2 การระบุลักษณะรอยร้าวกรณีรอยร้าวฝังรูปวงรีทั้งสองรอ

$MPa := 10^6 Pa$ $\quad ORIGIN := 1$

No. of Crack $\quad n_{crack} := 2 \quad i := 1 .. n_{crack}$

Crack Data

$type := \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ 1 : semi-elliptical, 2 : emeded

Crack Position $\quad x_M := \begin{pmatrix} 0 \\ 60 \end{pmatrix} \quad y_M := \begin{pmatrix} 10 \\ 40 \end{pmatrix} \quad z_M := \begin{pmatrix} -70 \\ -80 \end{pmatrix}$

inclined angle α $\quad \alpha := \begin{pmatrix} 20 \\ 5 \end{pmatrix} deg$

inclined angle θ $\quad \theta := \begin{pmatrix} 5 \\ 15 \end{pmatrix}$

(Half) crack length $\quad c0 := \begin{pmatrix} 40 \\ 20 \end{pmatrix} \quad 2c0 = \begin{pmatrix} 80 \\ 40 \end{pmatrix}$

(Half) crack depth $\quad a0 := \begin{pmatrix} 20 \\ 10 \end{pmatrix} \quad 2a0 = \begin{pmatrix} 40 \\ 20 \end{pmatrix}$

crack position

$$y_{N_i} := \sqrt{\left[\left(c0_i\right)^2 - \left(c0_i \cdot \cos\left(\alpha_i\right)\right)^2\right]} \qquad x_{N_i} := \sqrt{\left[\left(c0_i\right)^2 - \left(y_{N_i}\right)^2\right]}$$

$x_{i,1} := x_{M_i} - x_{N_i}$ $\quad x_{i,2} := x_{M_i} + x_{N_i}$ $\quad x_{i,3} := x_{M_i}$ $\quad x_{i,4} := x_{M_i}$

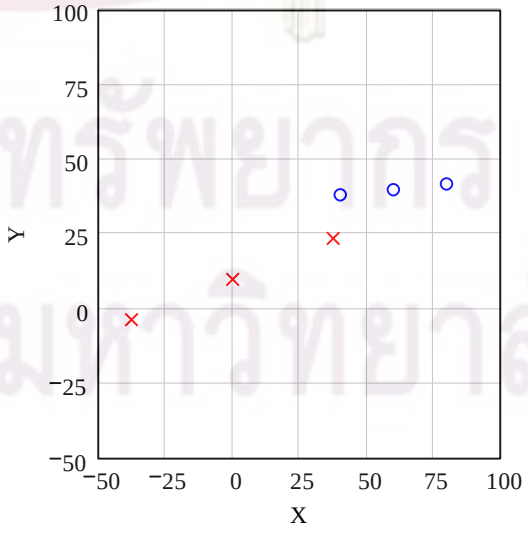
$y_{i,1} := y_{M_i} - y_{N_i}$ $\quad y_{i,2} := y_{M_i} + y_{N_i}$ $\quad y_{i,3} := y_{M_i}$ $\quad y_{i,4} := y_{M_i}$

$z_{i,1} := z_{M_i}$ $\quad z_{i,2} := z_{M_i}$ $\quad z_{i,3} := z_{M_i} + a0_i$ $\quad z_{i,4} := z_{M_i} - a0_i$

$$x = \begin{pmatrix} -37.588 & 37.588 & 0 & 0 \\ 40.076 & 79.924 & 60 & 60 \end{pmatrix}$$

$$y = \begin{pmatrix} -3.681 & 23.681 & 10 & 10 \\ 38.257 & 41.743 & 40 & 40 \end{pmatrix}$$

$$z = \begin{pmatrix} -70 & -70 & -50 & -90 \\ -80 & -80 & -70 & -90 \end{pmatrix}$$



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

1) Adjust (half) crack length

$\sigma 1 := 200MPa \qquad \sigma 2 := 100MPa$

$B := \frac{\sigma 2}{\sigma 1} \qquad B = 0.5$

$$c_i := \begin{cases} \left[\cos(\alpha_i)^2 + \frac{(1-B)\cdot\sin(\alpha_i)\cdot\cos(\alpha_i)}{2} + B^2\cdot\sin(\alpha_i)^2\right]\cdot c0_i & \text{if } |\alpha_i| \le 45deg \vee B = 0 \\ \left[\frac{\cos(\alpha_i)^2}{B^2} + \frac{(1-B)\cdot\sin(\alpha_i)\cdot\cos(\alpha_i)}{2\cdot B^2} + \sin(\alpha_i)^2\right]\cdot c0_i & \text{if } |\alpha_i| > 45deg \end{cases}$$

$c^T = (39.705 \quad 20.32) \qquad 2\,c^T = (79.409 \quad 40.64)$

2) Adjust crack depth

$$w_i := 0.99999 + 1.0481\cdot10^{-5}\cdot\theta_i + 1.5471\cdot10^{-4}\cdot(\theta_i)^2 + 3.4141\cdot10^{-5}\cdot(\theta_i)^3 - 2.0688\cdot10^{-6}\cdot(\theta_i)$$
$$+ 4.4977\cdot10^{-8}\cdot(\theta_i)^5 - 4.5751\cdot10^{-10}\cdot(\theta_i)^6 + 1.822\cdot10^{-12}\cdot(\theta_i)^7$$

{θ in deg}

$w^T = (1.007 \quad 1.075)$

$a_i := a0_i\cdot\cos(\theta_i\cdot deg)\cdot\max(w_i, 1) \qquad a^T = (20.064 \quad 10.381) \quad 2\,a^T = (40.127 \quad 20.762)$

3) Determine Crack Coordinates

$x_{i,1} := x_{M_i} - c_i \qquad x_{i,2} := x_{M_i} + c_i \qquad x_{i,3} := x_{M_i} \qquad x_{i,4} := x_{M_i}$

$x_{i,1} =$

-39.705
39.68

$x_{i,2} =$

39.705
80.32

$x_{i,3} =$

0
60

$x_{i,4} =$

0
60

$y_{i,1} := y_{i,3} \qquad y_{i,2} := y_{i,3} \qquad y_{i,3} := y_{i,3} \qquad y_{i,4} := y_{i,3}$

$y_{i,1} =$

10
40

$y_{i,2} =$

10
40

$y_{i,3} =$

10
40

$y_{i,4} =$

10
40

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$z_{i,1} := z_{M_i}$ $z_{i,2} := z_{M_i}$ $z_{i,3} := z_{M_i} + a_i$ $z_{i,4} := z_{M_i} - a_i$

$z_{i,1} =$

-70
-80

$z_{i,2} =$

-70
-80

$z_{i,3} =$

-49.936
-69.619

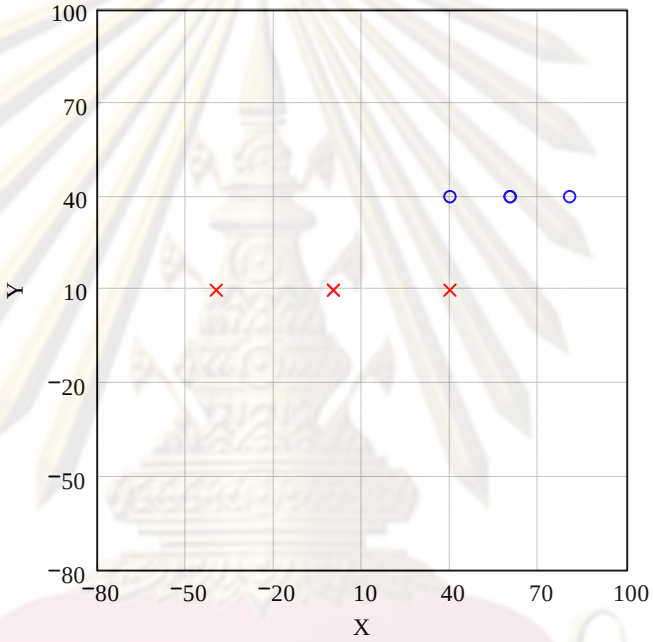
$z_{i,4} =$

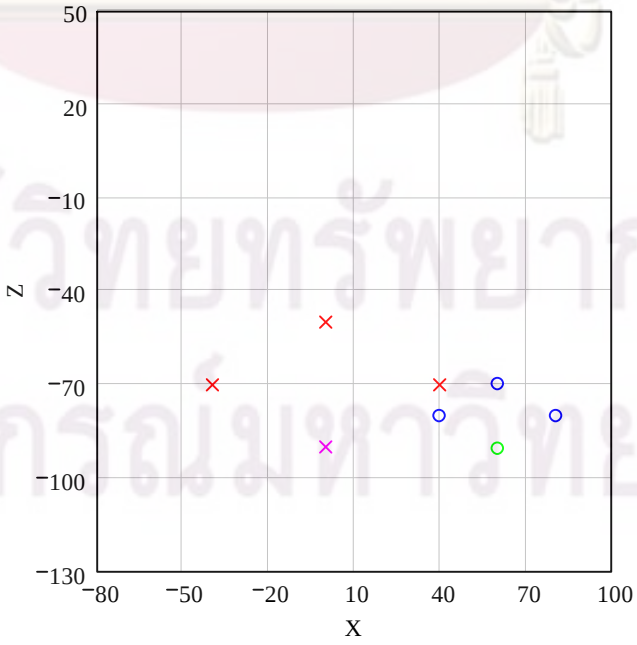
-90.064
-90.381

$$x = \begin{pmatrix} -39.705 & 39.705 & 0 & 0 \\ 39.68 & 80.32 & 60 & 60 \end{pmatrix} \quad y = \begin{pmatrix} 10 & 10 & 10 & 10 \\ 40 & 40 & 40 & 40 \end{pmatrix} \quad z = \begin{pmatrix} -70 & -70 & -49.936 & -90.064 \\ -80 & -80 & -69.619 & -90.381 \end{pmatrix}$$

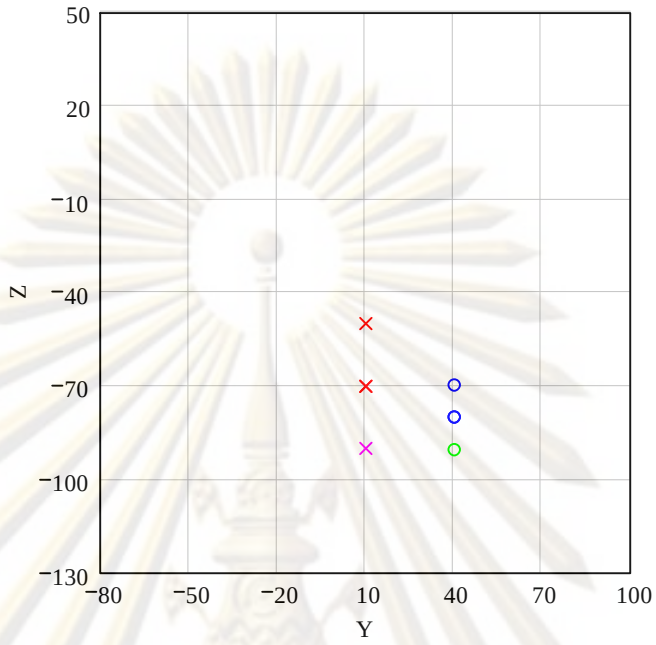
$x_{2,3} = 60$

$x_{2,4} = 60$





ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



4). Calculate Planar distance between crack

$$S1 := \max\left(y_{2,3}, y_{1,3}\right) - \min\left(y_{2,3}, y_{1,3}\right)$$

$$S1 = 30$$

5). Calculate Planar overlap length

$$S2 := \max\left(x_{2,2}, x_{1,2}\right) - \min\left(x_{2,1}, x_{1,1}\right) - \left|x_{2,2} - x_{2,1}\right| - \left|x_{1,2} - x_{1,1}\right|$$

$$S2 = -0.025$$

6). Calculate Thickness overlap length

$$S3 := \left|\min\left(z_{2,4}, z_{1,4}\right)\right| - \left|\max\left(z_{2,3}, z_{1,3}\right)\right| - \left|z_{2,4} - z_{2,3}\right| - \left|z_{1,4} - z_{1,3}\right|$$

$$S3 = -20.445$$

$$c^T = (39.705 \quad 20.32) \qquad a^T = (20.064 \quad 10.381)$$

$$c_{sum} := c_1 + c_2 \qquad a_{sum} := a_1 + a_2$$

$$c_{sum} = 60.025 \qquad a_{sum} = 30.445$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$c_{eff} := \begin{cases} \left\lfloor 0.5\cdot\left(\max\left(x_{2,2}, x_{1,2}\right) - \min\left(x_{1,1}, x_{2,1}\right)\right)\right\rfloor & \text{if } c_{sum} \geq S1 \wedge c_{sum} \geq S2 \wedge a_{sum} \geq S3 \\ \max\left(c_1, c_2\right) & \text{otherwise} \end{cases}$$

$$c_{eff} = 60.012$$

$$a_{eff} := \begin{cases} 0.5\cdot\left(\left|\min\left(z_{2,4}, z_{1,4}\right)\right| - \left|\max\left(z_{2,3}, z_{1,3}\right)\right|\right) & \text{if } c_{sum} \geq S1 \wedge c_{sum} \geq S2 \wedge a_{sum} \geq S3 \\ 0.5\cdot\left(\left|\max\left(z_{2,4}, z_{1,4}\right)\right|\right) & \text{otherwise} \end{cases}$$

$$a_{eff} = 20.222$$

Locate the position of an effective crack

$$x_{1,1} := \min\left(\min\left(x_{1,1}, x_{2,1}\right), \min\left(x_{2,2}, x_{1,2}\right)\right) \qquad x_{1,2} := \max\left(\max\left(x_{1,1}, x_{2,1}\right), \max\left(x_{2,2}, x_{1,2}\right)\right)$$

$$x_{1,3} := 0.5\cdot\left(x_{1,1} + x_{1,2}\right) \qquad x_{1,4} := x_{1,3}$$

$$y_{1,1} := 0.5\cdot\left(y_{M_1} + y_{M_2}\right) \qquad y_{1,2} := y_{1,1}$$

$$y_{1,3} := y_{1,1} \qquad y_{1,4} := y_{1,3}$$

$$y_{M_1} = 10$$

$$y_{M_2} = 40$$

$$z_{1,3} := \max\left(\max\left(z_{1,3}, z_{2,3}\right), \max\left(z_{1,4}, z_{2,4}\right)\right) \qquad z_{1,1} := z_{1,3} - a_{eff}$$

$$z_{1,2} := z_{1,1} \qquad z_{1,4} := z_{1,3} - 2a_{eff}$$

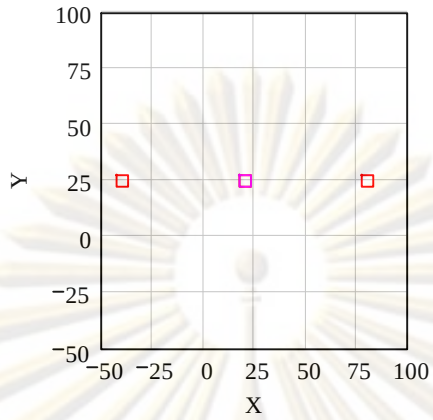
$$n_{crack} := n_{crack} - 1 \qquad i := 1 .. n_{crack}$$

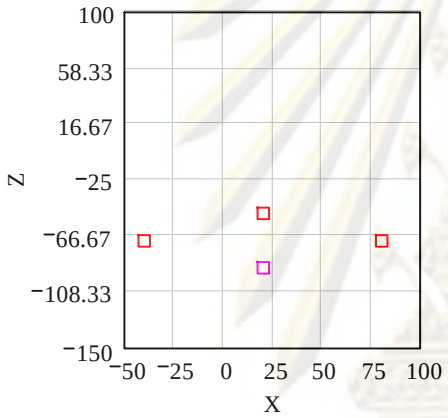
$x_{i,1} =$	$x_{i,2} =$	$x_{i,3} =$	$x_{i,4} =$
-39.705	80.32	20.308	20.308

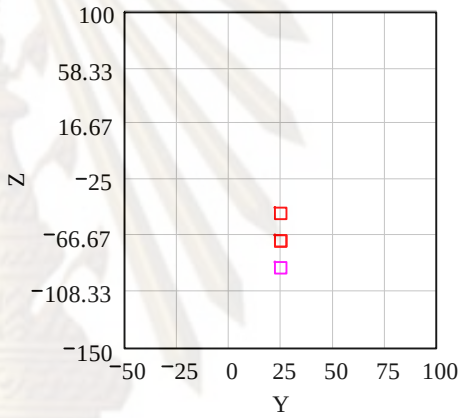
$y_{i,1} =$	$y_{i,2} =$	$y_{i,3} =$	$y_{i,4} =$
25	25	25	25

$z_{i,1} =$	$z_{i,2} =$	$z_{i,3} =$	$z_{i,4} =$
-70.159	-70.159	-49.936	-90.381

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย







(Half) crack length $c := c_{eff}$

$c = 60.012$ $2c = 120.025$

(Half)crack depth $a := a_{eff}$

$a = 20.222$ $2a = 40.445$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.1.3 การระบุลักษณะรอยร้าวกรณีรอยร้าวผิวรูปครึ่งวงรีและรอยร้าวฝังรูปวงร

No. of Crack $n_{crack} := 2$ $i := 1 .. n_{crack}$ $MPa := 10^6 Pa$

Crack Data $type := \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ 1 : semi-elliptical 2 : emeded ORIGIN := 1

Crack Position $x_M := \begin{pmatrix} 0 \\ 30 \end{pmatrix}$ $y_M := \begin{pmatrix} 0 \\ 30 \end{pmatrix}$ $z_M := \begin{pmatrix} 0 \\ -15 \end{pmatrix}$

inclined angle α $\alpha := \begin{pmatrix} 20 \\ 5 \end{pmatrix} deg$

inclined angle θ $\theta := \begin{pmatrix} 5 \\ 15 \end{pmatrix}$

(Half) crack length $c0 := \begin{pmatrix} 30 \\ 10 \end{pmatrix}$ $2c0 = \begin{pmatrix} 60 \\ 20 \end{pmatrix}$

(Half) crack depth $a0 := \begin{pmatrix} 10 \\ 5 \end{pmatrix}$

crack position

$$y_{N_i} := \sqrt{\left[\left(c0_i\right)^2 - \left(c0_i \cdot \cos\left(\alpha_i\right)\right)^2 \right]} \qquad x_{N_i} := \sqrt{\left[\left(c0_i\right)^2 - \left(y_{N_i}\right)^2 \right]}$$

$x_{i,1} := x_{M_i} - x_{N_i}$ $x_{i,2} := x_{M_i} + x_{N_i}$ $x_{i,3} := x_{M_i}$ $x_{i,4} := x_{M_i}$

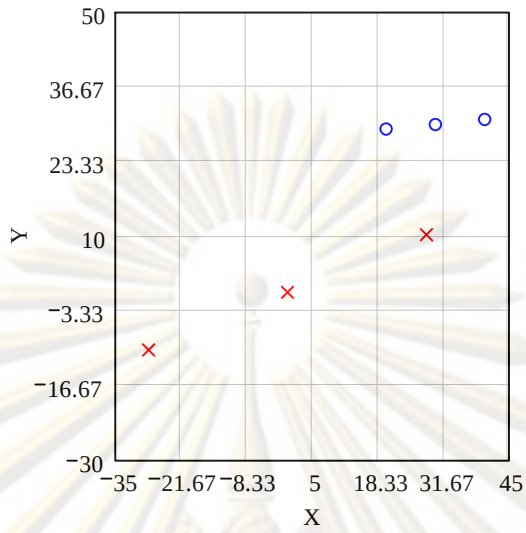
$y_{i,1} := y_{M_i} - y_{N_i}$ $y_{i,2} := y_{M_i} + y_{N_i}$ $y_{i,3} := y_{M_i}$ $y_{i,4} := y_{M_i}$

$$z_{i,1} := \begin{cases} z_{M_i} & \text{if } type_i = 1 \\ -z_{M_i} & \text{otherwise} \end{cases} \qquad z_{i,2} := z_{i,1} \qquad z_{i,3} := \begin{cases} z_{M_i} & \text{if } type_i = 1 \\ z_{M_i} + a0_i & \text{otherwise} \end{cases}$$

$$z_{i,4} := \begin{cases} -a0_i & \text{if } type_i = 1 \\ z_{M_i} - a0_i & \text{otherwise} \end{cases}$$

$$x = \begin{pmatrix} -28.191 & 28.191 & 0 & 0 \\ 20.038 & 39.962 & 30 & 30 \end{pmatrix} \quad y = \begin{pmatrix} -10.261 & 10.261 & 0 & 0 \\ 29.128 & 30.872 & 30 & 30 \end{pmatrix} \quad z = \begin{pmatrix} 0 & 0 & 0 & -10 \\ 15 & 15 & -10 & -20 \end{pmatrix}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



1) Adjust (half) crack length

$\sigma 1 := 200MPa \qquad \sigma 2 := 100MPa$

$B := \frac{\sigma 2}{\sigma 1} \qquad B = 0.5$

$$c_i := \begin{cases} \left[\cos(\alpha_i)^2 + \frac{(1-B)\cdot\sin(\alpha_i)\cdot\cos(\alpha_i)}{2} + B^2\cdot\sin(\alpha_i)^2\right]\cdot c0_i & \text{if } |\alpha_i| \le 45deg \vee B = 0 \\ \left[\frac{\cos(\alpha_i)^2}{B^2} + \frac{(1-B)\cdot\sin(\alpha_i)\cdot\cos(\alpha_i)}{2\cdot B^2} + \sin(\alpha_i)^2\right]\cdot c0_i & \text{if } |\alpha_i| > 45deg \end{cases}$$

$c^T = (29.778 \quad 10.16) \qquad 2\,c^T = (59.557 \quad 20.32)$

2) Adjust crack depth

$$w_i := 0.99999 + 1.0481\cdot10^{-5}\cdot\theta_i + 1.5471\cdot10^{-4}\cdot(\theta_i)^2 + 3.4141\cdot10^{-5}\cdot(\theta_i)^3 - 2.0688\cdot10^{-6}\cdot(\theta_i$$
$$+ 4.4977\cdot10^{-8}\cdot(\theta_i)^5 - 4.5751\cdot10^{-10}\cdot(\theta_i)^6 + 1.822\cdot10^{-12}\cdot(\theta_i)^7$$

$w^T = (1.007 \quad 1.075)$

$a_i := a0_i\cdot\cos(\theta_i\cdot deg)\cdot\max(w_i, 1) \qquad a^T = (10.032 \quad 5.19)$

$$a_i := \begin{cases} a_i & \text{if } type_i = 1 \\ a_i & \text{otherwise} \end{cases} \qquad a^T = (10.032 \quad 5.19)$$

3) Determine Crack Coordinates

$x_{i,1} := x_{M_i} - c_i$ $x_{i,2} := x_{M_i} + c_i$ $x_{i,3} := x_{M_i}$ $x_{i,4} := x_{M_i}$

$x_{i,1} =$

-29.778
19.84

$x_{i,2} =$

29.778
40.16

$x_{i,3} =$

0
30

$x_{i,4} =$

0
30

$y_{i,1} := y_{i,3}$ $y_{i,2} := y_{i,3}$ $y_{i,3} := y_{i,3}$ $y_{i,4} := y_{i,3}$

$y_{i,1} =$

0
30

$y_{i,2} =$

0
30

$y_{i,3} =$

0
30

$y_{i,4} =$

0
30

$z_{i,1} := z_{M_i}$

$z_{i,1} =$

0
-15

$z_{i,2} := z_{M_i}$

$z_{i,2} =$

0
-15

$$z_{i,3} := \begin{cases} z_{M_i} & \text{if } type_i = 1 \\ z_{M_i} + a_i & \text{otherwise} \end{cases}$$

$$z_{i,4} := \begin{cases} -a_i & \text{if } type_i = 1 \\ z_{M_i} - a_i & \text{otherwise} \end{cases}$$

$z_{i,3} =$

0
-9.81

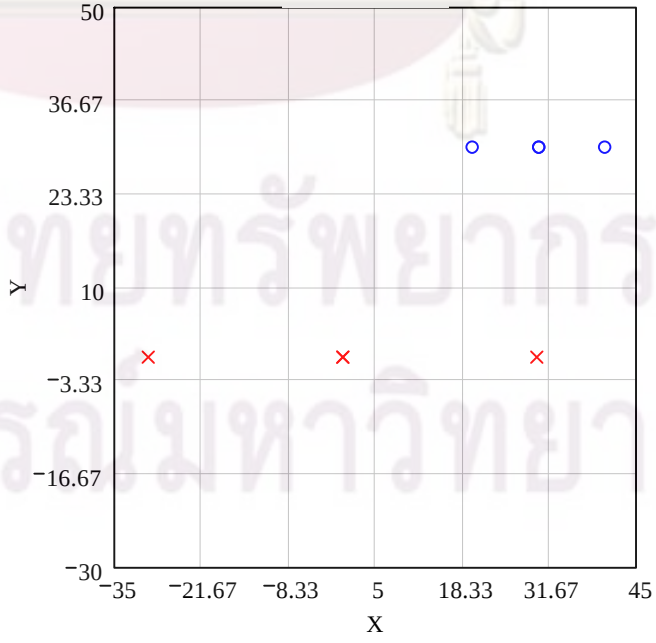
$z_{i,4} =$

-10.032
-20.19

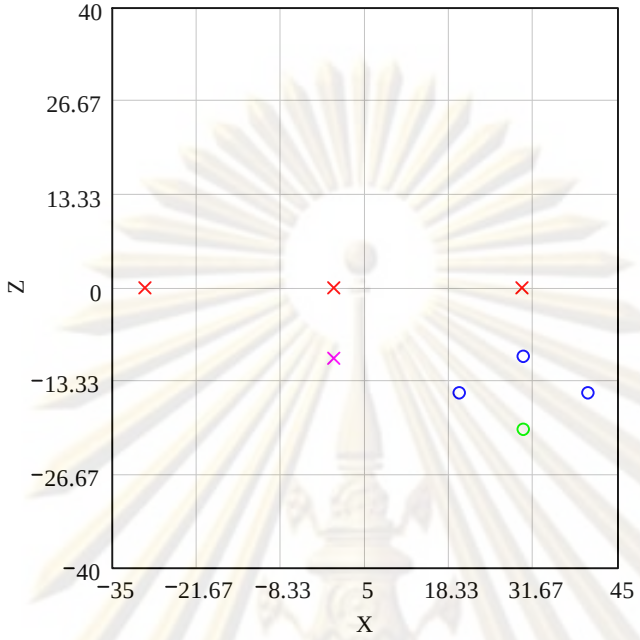
$$x = \begin{pmatrix} -29.778 & 29.778 & 0 \\ 19.84 & 40.16 & 30 \end{pmatrix}$$

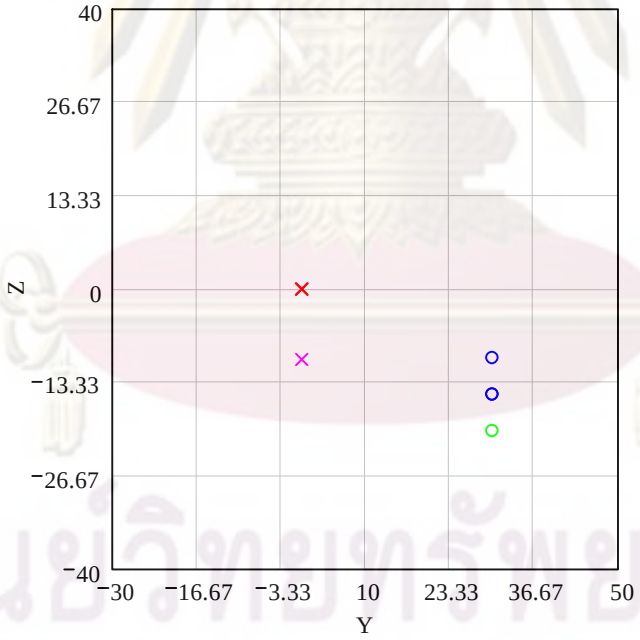
$$y = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 30 & 30 & 30 & 30 \end{pmatrix}$$

$$z = \begin{pmatrix} 0 & 0 & 0 & -10 \\ -15 & -15 & -9.81 & -20 \end{pmatrix}$$



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย





ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4). Calculate Planar distance between crack

$$S1 := \max(y_{2,3}, y_{1,3}) - \min(y_{2,3}, y_{1,3})$$

$$S1 = 30$$

5). Calculate Planar overlap length

$$S2 := \max(x_{2,2}, x_{1,2}) - \min(x_{2,1}, x_{1,1}) - |x_{2,2} - x_{2,1}| - |x_{1,2} - x_{1,1}|$$

$$S2 = -9.939$$

6). Calculate Thickness overlap length

$$S3 := |\min(z_{2,4}, z_{1,4})| - |\max(z_{2,3}, z_{1,3})| - |z_{2,4} - z_{2,3}| - |z_{1,4} - z_{1,3}|$$

$$S3 = -0.222$$

$$c^T = (29.778 \quad 10.16) \qquad a^T = (10.032 \quad 5.19)$$

$$c_{sum} := c_1 + c_2 \qquad a_{sum} := a_1 + a_2$$

$$c_{sum} = 39.939 \qquad a_{sum} = 15.222$$

$$c_{eff} := \begin{cases} [0.5 \cdot (\max(x_{2,2}, x_{1,2}) - \min(x_{1,1}, x_{2,1}))] & \text{if } c_{sum} \geq S1 \wedge c_{sum} \geq S2 \wedge a_{sum} \geq S3 \\ \max(c_1, c_2) & \text{otherwise} \end{cases}$$

$$c_{eff} = 34.969$$

$$a_{eff} := \begin{cases} |\min(z_{2,4}, z_{1,4})| - |\max(z_{2,3}, z_{1,3})| & \text{if } c_{sum} \geq S1 \wedge c_{sum} \geq S2 \wedge a_{sum} \geq S3 \\ |\max(z_{2,4}, z_{1,4})| & \text{otherwise} \end{cases}$$

$$a_{eff} = 20.19$$

Locate the position of an effective crack

$$x_{1,1} := \min(\min(x_{1,1}, x_{2,1}), \min(x_{2,2}, x_{1,2})) \qquad x_{1,2} := \max(\max(x_{1,1}, x_{2,1}), \max(x_{2,2}, x_{1,2}))$$

$$x_{1,3} := 0.5 \cdot (x_{1,1} + x_{1,2}) \qquad x_{1,4} := x_{1,3}$$

$$y_{1,1} := 0.5 \cdot (y_{M_1} + y_{M_2}) \qquad y_{1,2} := y_{1,1}$$

$$y_{1,3} := y_{1,1} \qquad y_{1,4} := y_{1,3}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$z_{1,1} := \max\left(\max\left(z_{1,3}, z_{2,3}\right), \max\left(z_{1,4}, z_{2,4}\right)\right)$ $z_{1,2} := z_{1,1}$

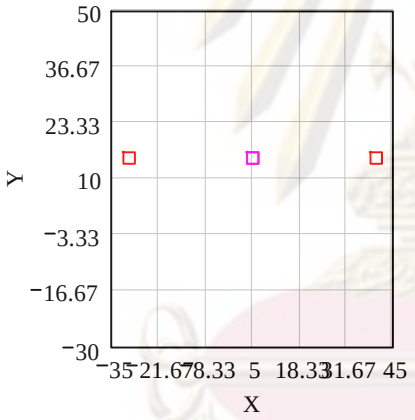
$z_{1,3} := z_{1,1}$ $z_{1,4} := z_{1,3} - a_{eff}$

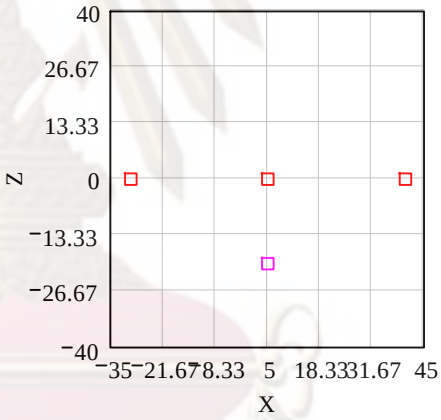
$n_{crack} := n_{crack} - 1$ $i := 1 .. n_{crack}$

$x_{i,1} =$	$x_{i,2} =$	$x_{i,3} =$	$x_{i,4} =$
-29.778	40.16	5.191	5.191

$y_{i,1} =$	$y_{i,2} =$	$y_{i,3} =$	$y_{i,4} =$
15	15	15	15

$z_{i,1} =$	$z_{i,2} =$	$z_{i,3} =$	$z_{i,4} =$
0	0	0	-20.19





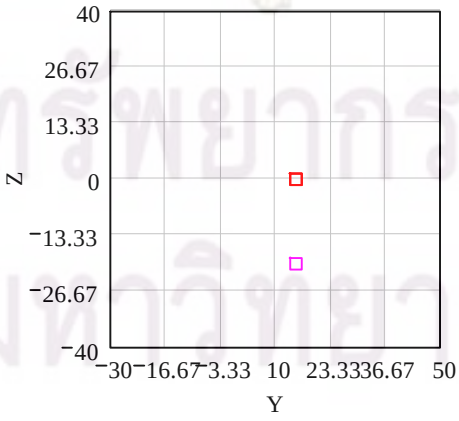
(Half) crack length $c := c_{eff}$

$c = 34.969$

(Half)crack depth $a := a_{eff}$

$a = 20.19$

$2c = 69.939$



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.2 การประเมินการคงสภาพ

ข.2.1 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวง รับภาระแรงดึงตามแนวแกน

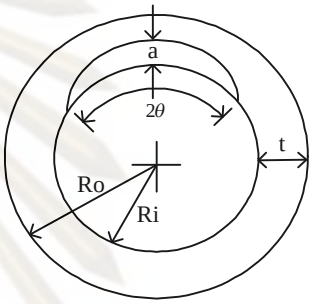
ข.2.1.1 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1

$GPa := 10^9 \cdot Pa$ $MPa := 10^6 \cdot Pa$ $MN := 10^6 newton$

R6 Option 1 & level 1

Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)



Structure's dimensions

Outer radius, RO	$R_o := 600mm$	
Inner radius, Ri	$R_i := 543mm$	
Thickness, t	$t := R_o - R_i$	$t = 57\,mm$
Mean radius, R	$R := R_i + \left(\frac{t}{2}\right)$	$R = 571.5\,mm$

1. Categorise load & stress

Primary load : Axial Tension, P_{op}	$P_{op} := 25MN$	
Secondary load : NONE		
Operating temperature	$T := 121$ {Celcius}	

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

Material properties $E := 207200MPa - 57.1(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269 \cdot MPa$

$\sigma_u := 414 \cdot MPa$

3. Select FAD

Option 1 $K_r(L_r) = \left(1 - 0.14 \cdot L_r^2\right) \cdot \left(0.3 + 0.7 \exp\left(-0.65 \cdot L_r^6\right)\right)$

$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2}$ $\sigma_{flow} = 341.5\,MPa$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Cut-off $L_{r_max} := \frac{\sigma_{flow}}{\sigma_y}$ $L_{r_max} = 1.27$

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

Crack Depth, a $a := 20mm$

5. Select category of analysis category 1

6. Define fracture toughness K_I ,K_{IC}, or $K_{0.2}$

$T_{ref} := 121$

$K_{IC} := \left[36.5 + 3.084\exp\left[0.036\cdot\left(T - T_{ref} + 56\right)\right]\right]\cdot MPa\cdot\sqrt{m}$ From API 579 [5]

$K_{IC} = 59.655\,MPa\cdot\sqrt{m}$

7. Define crack size a_0

$a_0 := a$ $a_0 = 20\,mm$

8. Calculate L_r $L_r = P/P_L$

Limit loadfrom Zahoor [20]

Semi-elliptical Circumferential Axial Tension 5 <= Ro/t <= 20 $\frac{R_o}{t} = 10.526$

$\theta := \pi\cdot\frac{c}{4\cdot R_i}$ $\theta = 0.072$

$\zeta := \frac{t}{R_o}$ $\zeta = 0.095$

$x := \frac{a}{t}$ $x = 0.351$

$A1 := x\cdot\frac{\left[(1-\zeta)\cdot(2 - 2\cdot\zeta + x\cdot\zeta) + (1 - \zeta + x\cdot\zeta)^2\right]}{2\cdot\left[1 + (2-\zeta)\cdot(1-\zeta)\right]}$ $A1 = 0.164$

$\alpha := \operatorname{acos}(A1\cdot\sin(\theta))$ $\alpha = 1.559$

$P_L := \begin{cases} \left[2\cdot\pi R\cdot t\cdot\sigma_{flow}\cdot\left[\left(2\cdot\frac{\alpha}{\pi}\right) - \left(x\cdot\frac{\theta}{\pi}\right)\cdot\frac{(2 - 2\cdot\zeta + x\cdot\zeta)}{(2-\zeta)}\right]\right] & \text{if } 5 \le \frac{R_o}{t} \le 20 \\ 0 \quad \text{otherwise} \end{cases}$

$P_L = 68.823\,MN$

$L_R(P) := \frac{P}{P_L}$ $L_R(P_{op}) = 0.363$

9.Calculate K_r

Parameter K from Zahoor [20] $K_I = \sigma_t \cdot (\pi t)^{0.5} \cdot F_t$

The available K-solution of this cracked structure is valid within the following range

$0.05 \le \frac{a}{t} \le 0.8$ $5 \le \frac{R}{t} \le 160$ $\frac{2 \cdot c}{a} \ge 3$

For our problem $\frac{a}{t} = 0.351$ $\frac{R}{t} = 10.026$ $\frac{2.c}{a} = 5$

where $\sigma_t(P) := \frac{P}{2 \cdot \pi \cdot R \cdot t}$ $\sigma_t(P_{op}) = 122.143\ \text{MPa}$

$\beta := \frac{\left(\frac{a}{t}\right)}{\left(0.25 + \frac{a}{c}\right)^{0.58}}$ $\beta = 0.45$

$$F_t := \begin{cases} \left[3.702 \cdot \beta - 13.475 \cdot \beta^2 + 20.0 \cdot \beta^3 + 0.0086 \cdot \beta \cdot \left(\frac{R}{t} - 5\right)\right] & \text{if } \beta < 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[3.831 \cdot \beta - 13.475 \cdot \beta^2 + 20.0 \cdot \beta^3 + 0.002 \cdot \beta \cdot \left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta < 0.25 \wedge 20 \le \frac{R}{t} \le 1 \\ \left[0.25 + 0.4868 \cdot \beta + 0.3835 \cdot \beta^2 + 0.0086 \cdot \beta \cdot \left(\frac{R}{t} - 5\right)\right] & \text{if } \beta \ge 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[0.25 + 0.6158 \cdot \beta + 0.3835 \cdot \beta^2 + 0.002 \cdot \beta \cdot \left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta \ge 0.25 \wedge 20 < \frac{R}{t} \le 160 \\ 0 & \text{otherwise} \end{cases}$$

$K_I(P) := \sigma_t(P) \cdot (\pi t)^{0.5} \cdot F_t$ $K_I(P_{op}) = 29.285\ \text{MPa} \cdot \sqrt{\text{m}}$

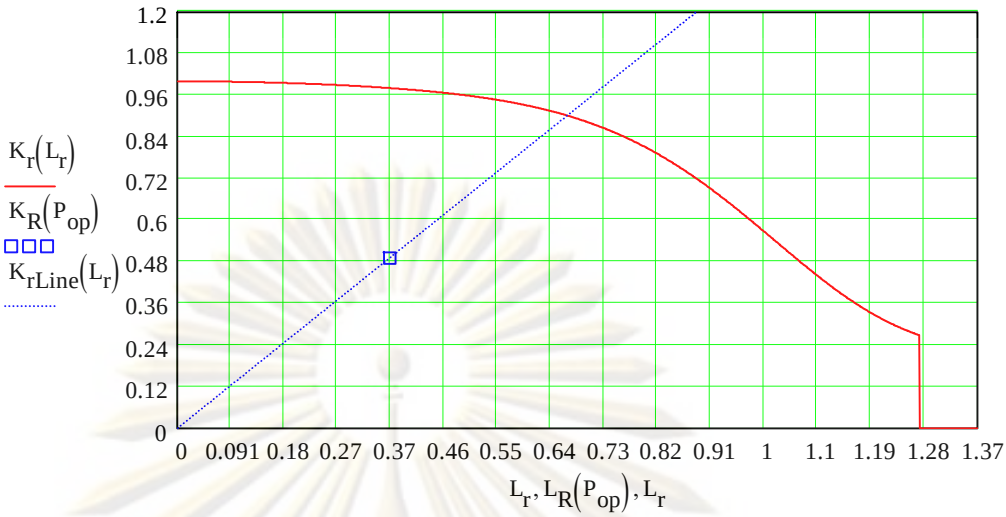
$K_R(P) := \frac{K_I(P)}{K_{IC}}$ $K_R(P_{op}) = 0.491$

10. Plot assessment piont (L_r,K_r) $L_r := 0, 0.001 .. 1.5$

$$K_r(L_r) := \begin{cases} \left[\left(1 - 0.14 \cdot L_r^2\right) \cdot \left(0.3 + 0.7 \exp\left(-0.65 \cdot L_r^6\right)\right)\right] & \text{if } 0 \le L_r \le L_{r_max} \\ 0 & \text{otherwise} \end{cases}$$

$K_{rLine}(L_r) := \frac{K_R(P_{op})}{L_R(P_{op})} \cdot (L_r - L_R(P_{op})) + K_R(P_{op})$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

$$\left[\frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] = \left(1 - 0.14\cdot L_r^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_r^{\,6}}\right)$$

$$f(L_r) := \left[\frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] - \left(1 - 0.14\cdot L_r^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_r^{\,6}}\right)$$

$L_r := 1$

$L_{rC} := \text{root}(f(L_r), L_r)$ $\quad L_{rC} = 0.667$

$K_{rC} := \left(1 - 0.14\cdot L_{rC}^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_{rC}^{\,6}}\right)$ $\quad K_{rC} = 0.901$

$K_{rmax} := \left(1 - 0.14\cdot L_{r_max}^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_{r_max}^{\,6}}\right)$ $\quad K_{rmax} = 0.268$

L_r Critical $\quad L_{rC1} := \begin{cases} L_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R(P_{op})}{L_R(P_{op})} \\ L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$L_{rC1} = 0.667$

K_r Critical $\quad K_{rC1} := \begin{cases} K_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R(P_{op})}{L_R(P_{op})} \\ \dfrac{K_R(P_{op})}{L_R(P_{op})}\cdot L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$K_{rC1} = 0.901$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$d := \sqrt{K_R(P_{op})^2 + L_R(P_{op})^2}$$

$$dC := \sqrt{K_{rC1}{}^2 + L_{rC1}{}^2}$$

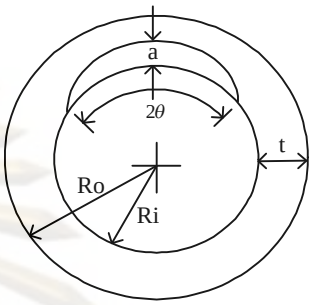
$$FL := \frac{dC}{d} \qquad FL = 1.836$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.2.1.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 2

$GPa := 10^{9}\cdot Pa$ $MPa := 10^{6}$ $MN := 10^{6}\,newton$

R6 Option 1 & Level 2



Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)

Structure's dimensions

Outer radius, RO $R_o := 600mm$

Inner radius, Ri $R_i := 543mm$

Thickness, t $t := R_o - R_i$ $t = 57\,mm$

Mean radius, R $R := R_i + \left(\frac{t}{2}\right)$ $R = 571.5\,mm$

1. Categorise load & stress

Primary load : Axial tension, P_{op} $P_{op} := 25MN$

Secondary load : NONE

Operating temperature $T := 121$ {Celcius}

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

$E := 207200MPa - 57.1\,(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269\cdot MPa$

$\sigma_u := 414\cdot MPa$

3. Select FAD

Option 1 $K_r(L_r) = \left(1 - 0.14\cdot L_r^{2}\right)\cdot\left(0.3 + 0.7\exp\left(-0.65\cdot L_r^{6}\right)\right)$

$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2}$ $\sigma_{flow} = 341.5\,MPa$

Cut-off $L_{r_max} := \frac{\sigma_{flow}}{\sigma_y}$ $L_{r_max} = 1.27$

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Crack Depth, a $a := 20mm$

5. Select category of analysis category 2

6. Define fracture toughness $K_{0.2}$ and $K_{.g}$

Material name : SA106Gr. C

$J = C_1(\Delta a)^{C_2}$ $C_1 = 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c}\right)^{2.29}$ $C_2 := 0.38$

$s_u := \sigma_u$ $s_y := \sigma_y$

$\varepsilon_c := 0.82$ $E = 200.291\,GPa$

$k := 1mm$

$\Delta a := 0.2$

$S_u := \frac{s_u}{E}$ $S_u = 2.067 \times 10^{-3}$

$C_1 := 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c}\right)^{2.29}$ $C_1 = 332.298\,\frac{N}{mm}$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $J(\Delta a) = 180.268\,\frac{N}{mm}$

$K_{0.2} := \sqrt{\frac{(J(\Delta a) \cdot E)}{\left(1 - 0.3^2\right)}}$ $K_{0.2} = 199.191\,MPa \cdot \sqrt{m}$

$\Delta a_g := 2$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $J(\Delta a_g) = 432.433\,\frac{N}{mm}$

$K_g := \sqrt{\frac{\left(J(\Delta a_g) \cdot E\right)}{\left(1 - 0.3^2\right)}}$ $K_g = 308.51\,MPa \cdot \sqrt{m}$

7. Define crack size a_0 and a_g

$a_0 := a$ $a_0 = 20\,mm$

Assume $\Delta a_g := 2mm$

8. Calculate L_r $L_r = P/P_L$

Original crack size a_0

Limit loadfrom Zahoor [20]

Semi-elliptical Circumferential Axial Tension 5 <= Ro/t <= 20 $\frac{R_o}{t} = 10.526$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$\theta := \pi \cdot \frac{c}{4 \cdot R_i} \qquad \theta = 0.072$$

$$\zeta := \frac{t}{R_o} \qquad \zeta = 0.095$$

$$x := \frac{a}{t} \qquad x = 0.351$$

$$A1 := x \cdot \frac{\left[(1-\zeta)\cdot(2-2\cdot\zeta+x\cdot\zeta)+(1-\zeta+x\cdot\zeta)^2\right]}{2\cdot\left[1+(2-\zeta)\cdot(1-\zeta)\right]} \qquad A1 = 0.164$$

$$\alpha := \operatorname{acos}(A1 \cdot \sin(\theta)) \qquad \alpha = 1.559$$

$$P_L := \begin{cases} \left[2\cdot\pi R\cdot t\cdot\sigma_{flow}\cdot\left[\left(2\cdot\frac{\alpha}{\pi}\right)-\left(x\cdot\frac{\theta}{\pi}\right)\cdot\frac{(2-2\cdot\zeta+x\cdot\zeta)}{(2-\zeta)}\right]\right] & \text{if } 5 \le \frac{R_o}{t} \le 2 \\ 0 \quad \text{otherwise} \end{cases}$$

$$P_L = 68.823\,\text{MN}$$

$$L_R(P) := \frac{P}{P_L} \qquad L_R(P_{op}) = 0.363$$

Ductile crack growth a_g

Limit loadfrom Zahoor [20]

Semi-elliptical Circumferential Axial Tension 5 <= Ro/t <= 20 $\qquad \frac{R_o}{t} = 10.526$

$$\theta := \pi \cdot \frac{c}{4 \cdot R_i} \qquad \theta = 0.072$$

$$\zeta := \frac{t}{R_o} \qquad \zeta = 0.095$$

$$x := \frac{a + \Delta a_g}{t} \qquad x = 0.386$$

$$A1 := x \cdot \frac{\left[(1-\zeta)\cdot(2-2\cdot\zeta+x\cdot\zeta)+(1-\zeta+x\cdot\zeta)^2\right]}{2\cdot\left[1+(2-\zeta)\cdot(1-\zeta)\right]} \qquad A1 = 0.181$$

$$\alpha := \operatorname{acos}(A1 \cdot \sin(\theta)) \qquad \alpha = 1.558$$

$$P_L := \begin{cases} \left[2\cdot\pi R\cdot t\cdot\sigma_{flow}\cdot\left[\left(2\cdot\frac{\alpha}{\pi}\right)-\left(x\cdot\frac{\theta}{\pi}\right)\cdot\frac{(2-2\cdot\zeta+x\cdot\zeta)}{(2-\zeta)}\right]\right] & \text{if } 5 \le \frac{R_o}{t} \le 20 \\ 0 \quad \text{otherwise} \end{cases}$$

$$P_L = 68.713\,\text{MN}$$

$$L_{Rg}(P) := \frac{P}{P_L} \qquad L_{Rg}(P_{op}) = 0.364$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

9.Calculate K_r

Parameter K from Zahoor [19] $K_I = \sigma_t \cdot (\pi t)^{0.5} \cdot F_t$

The available K-solution of this cracked structure is valid within the following range

$$0.05 \le \frac{a}{t} \le 0.8 \qquad 5 \le \frac{R}{t} \le 160 \qquad \frac{2 \cdot c}{a} \ge 3$$

For our problem $\frac{a}{t} = 0.351$ $\frac{R}{t} = 10.026$ $\frac{2.c}{a} = 5$

$$\frac{\left(a + \Delta a_g\right)}{t} = 0.386 \qquad \frac{2c}{\left(a + \Delta a_g\right)} = 4.545$$

Original crack size a_0

where $\sigma_t(P) := \frac{P}{2 \cdot \pi \cdot R \cdot t}$ $\sigma_t\left(P_{op}\right) = 122.143\ \text{MPa}$

$$\beta := \frac{\left(\frac{a}{t}\right)}{\left(0.25 + \frac{a}{c}\right)^{0.58}} \qquad \beta = 0.45$$

$$F_t := \begin{cases} \left[3.702 \cdot \beta - 13.475 \cdot \beta^2 + 20.0 \cdot \beta^3 + 0.0086 \cdot \beta \cdot \left(\frac{R}{t} - 5\right)\right] & \text{if } \beta < 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[3.831 \cdot \beta - 13.475 \cdot \beta^2 + 20.0 \cdot \beta^3 + 0.002 \cdot \beta \cdot \left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta < 0.25 \wedge 20 \le \frac{R}{t} \le \\ \left[0.25 + 0.4868 \cdot \beta + 0.3835 \cdot \beta^2 + 0.0086 \cdot \beta \cdot \left(\frac{R}{t} - 5\right)\right] & \text{if } \beta \ge 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[0.25 + 0.6158 \cdot \beta + 0.3835 \cdot \beta^2 + 0.002 \cdot \beta \cdot \left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta \ge 0.25 \wedge 20 < \frac{R}{t} \le 16 \\ 0 & \text{otherwise} \end{cases}$$

$K_I(P) := \sigma_t(P) \cdot (\pi t)^{0.5} \cdot F_t$ $K_I\left(P_{op}\right) = 29.285\ \text{MPa} \cdot \sqrt{\text{m}}$

$K_R(P) := \frac{K_I(P)}{K_{0.2}}$ $K_R\left(P_{op}\right) = 0.147$

Ductile crack growth a_g

Parameter K from Zahoor [20] $K_I = \sigma_t \cdot (\pi t)^{0.5} \cdot F_t$

where $\sigma_t(P) := \frac{P}{2 \cdot \pi \cdot R \cdot t}$ $\sigma_t\left(P_{op}\right) = 122.143\ \text{MPa}$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$\beta := \frac{\left(\frac{a + \Delta a_g}{t}\right)}{\left(0.25 + \frac{a + \Delta a_g}{c}\right)^{0.58}} \qquad \beta = 0.479$$

$$F_t := \begin{cases} \left[3.702\cdot\beta - 13.475\cdot\beta^2 + 20.0\cdot\beta^3 + 0.0086\cdot\beta\cdot\left(\frac{R}{t} - 5\right)\right] & \text{if } \beta < 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[3.831\cdot\beta - 13.475\cdot\beta^2 + 20.0\cdot\beta^3 + 0.002\cdot\beta\cdot\left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta < 0.25 \wedge 20 \le \frac{R}{t} \le \\ \left[0.25 + 0.4868\cdot\beta + 0.3835\cdot\beta^2 + 0.0086\cdot\beta\cdot\left(\frac{R}{t} - 5\right)\right] & \text{if } \beta \ge 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[0.25 + 0.6158\cdot\beta + 0.3835\cdot\beta^2 + 0.002\cdot\beta\cdot\left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta \ge 0.25 \wedge 20 < \frac{R}{t} \le 16 \\ 0 \quad \text{otherwise} \end{cases}$$

$$K_I(P) := \sigma_t(P)\cdot(\pi t)^{0.5}\cdot F_t \qquad K_I(P_{op}) = 30.576\ \text{MPa}\cdot\sqrt{\text{m}}$$

$$K_{Rg}(P) := \frac{K_I(P)}{K_g} \qquad K_{Rg}(P_{op}) = 0.099$$

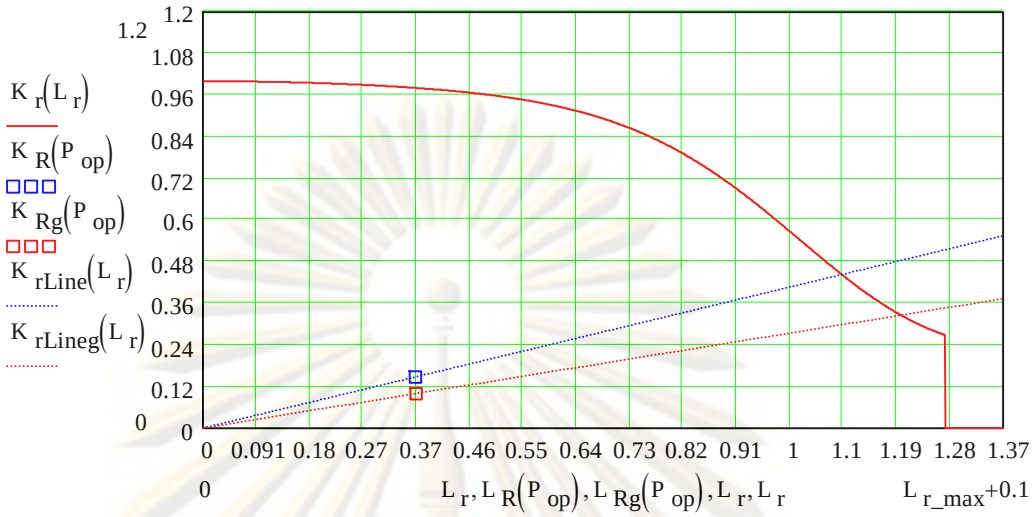
10. Plot assessment piont (L_r, K_r) $\qquad L_r := 0, 0.001 .. 1.5$

$$K_r(L_r) := \begin{cases} \left[\left(1 - 0.14\cdot L_r^2\right)\cdot\left(0.3 + 0.7\exp\left(-0.65\cdot L_r^6\right)\right)\right] & \text{if } 0 \le L_r \le L_{r_max} \\ 0 \quad \text{otherwise} \end{cases}$$

$$K_{rLine}(L_r) := \frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})$$

$$K_{rLineg}(L_r) := \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})}\cdot\left(L_r - L_{Rg}(P_{op})\right) + K_{Rg}(P_{op})$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

Original crack size a_0

$$\left[\frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] = \left(1 - 0.14\cdot L_r^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_r^{\,6}}\right)$$

$$f(L_r) := \left[\frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] - \left(1 - 0.14\cdot L_r^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_r^{\,6}}\right)$$

$L_r := 1$

$L_{rC} := \text{root}\left(f(L_r), L_r\right)$ $\quad L_{rC} = 1.093$

$K_{rC} := \left(1 - 0.14\cdot L_{rC}^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_{rC}^{\,6}}\right)$ $\quad K_{rC} = 0.442$

$K_{rmax} := \left(1 - 0.14\cdot L_{r_max}^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_{r_max}^{\,6}}\right)$ $\quad K_{rmax} = 0.268$

L_r Critical $\quad L_{rC1} := \begin{cases} L_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R(P_{op})}{L_R(P_{op})} \\ L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$L_{rC1} = 1.093$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

K_r Critical $\quad K_{rC1} := \begin{cases} K_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R(P_{op})}{L_R(P_{op})} \\ \dfrac{K_R(P_{op})}{L_R(P_{op})} \cdot L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$K_{rC1} = 0.442$

$d := \sqrt{K_R(P_{op})^2 + L_R(P_{op})^2}$ $\qquad dC := \sqrt{K_{rC1}{}^2 + L_{rC1}{}^2}$

$FL := \dfrac{dC}{d}$ $\qquad FL = 3.009$

Ductile crack growth a_g

$$\left[\frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \cdot \left(L_r - L_{Rg}(P_{op})\right) + K_{Rg}(P_{op})\right] = \left(1 - 0.14 \cdot L_r{}^2\right) \cdot \left(0.3 + 0.7 \cdot e^{-0.65 \cdot L_r{}^6}\right)$$

$$f(L_r) := \left[\frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \cdot \left(L_r - L_{Rg}(P_{op})\right) + K_{Rg}(P_{op})\right] - \left(1 - 0.14 \cdot L_r{}^2\right) \cdot \left(0.3 + 0.7 \cdot e^{-0.65 \cdot L_r{}^6}\right)$$

$L_r := 1$

$L_{rC} := \text{root}(f(L_r), L_r)$ $\qquad L_{rC} = 1.194$

$K_{rC} := \left(1 - 0.14 \cdot L_{rC}{}^2\right) \cdot \left(0.3 + 0.7 \cdot e^{-0.65 \cdot L_{rC}{}^6}\right)$ $\qquad K_{rC} = 0.325$

L_r Critical $\quad L_{rC1} := \begin{cases} L_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \\ L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \end{cases}$

$L_{rC1} = 1.194$

K_r Critical $\quad K_{rC1} := \begin{cases} K_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \\ \dfrac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \cdot L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \end{cases}$

$K_{rC1} = 0.325$

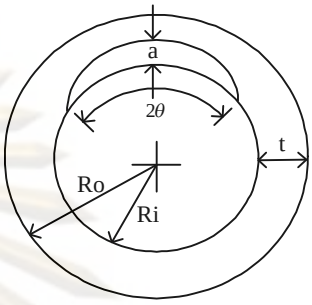
$d := \sqrt{K_{Rg}(P_{op})^2 + L_{Rg}(P_{op})^2}$ $\qquad dC := \sqrt{K_{rC1}{}^2 + L_{rC1}{}^2}$

$FL := \dfrac{dC}{d}$ $\qquad FL = 3.282$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.2.1.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1

$GPa := 10^9 \cdot Pa$ $MPa := 10^6 \cdot Pa$ $MN := 10^6 newton$



R6 Option 2 & level 1

Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)

Structure's dimensions

Outer radius, RO	$R_o := 600mm$	
Inner radius, Ri	$R_i := 543mm$	
Thickness, t	$t := R_o - R_i$	$t = 57\,mm$
Mean radius, R	$R := R_i + \left(\frac{t}{2}\right)$	$R = 571.5\,mm$

1. Categorise load & stress

Primary load : Axial Tension, P_{op} $P_{op} := 25MN$

Secondary load : NONE

Operating temperature $T := 121$ {Celcius}

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

Material properties $E := 207200MPa - 57.1(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269 \cdot MPa$

$\sigma_u := 414 \cdot MPa$ $\alpha := 1$

Estinate n from R6

$$n_E := \left| \begin{array}{l} \text{for } i \in 0..N \\ \quad \left| \begin{array}{l} n_0 \leftarrow 0.01 \\ n_{i+1} \leftarrow \dfrac{\ln\left(\dfrac{1}{0.002 \cdot n_i}\right) - 1}{\ln\left(\dfrac{\sigma_u}{\sigma_y}\right)} \end{array} \right. \\ n \end{array} \right.$$

$N := 15$

$n_E =$

	0
0	0.01
1	22.776
2	4.845
3	8.435

$n := n_{E_{15}}$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3. Select FAD

Option 2 $$K_r(L_r) = \left(\frac{E \cdot \varepsilon_{ref}}{L_r \cdot \sigma_y} + \frac{L_r^3 \cdot \sigma_y}{2 \cdot E \cdot \varepsilon_{ref}} \right)^{\frac{-1}{2}}$$

$$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2} \qquad \sigma_{flow} = 341.5\,MPa$$

Ramberg-Osgood Equation $$\frac{\varepsilon_{ref}}{\varepsilon_0} = \frac{\sigma_{ref}}{\sigma_0} + \alpha \cdot \left(\frac{\sigma_{ref}}{\sigma_0} \right)^n$$

$$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2} \qquad \sigma_{ref} = L_r \cdot \sigma_y$$

$$\sigma_0 := \sigma_y \qquad \varepsilon_0 := \frac{\sigma_0}{E}$$

$$\varepsilon_{ref} = \varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right]$$

$$K_{rOption2}(L_r) := \left[\frac{E \cdot \left[\varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right] \right]}{L_r \cdot \sigma_y} + \frac{L_r^3 \cdot \sigma_y}{2 \cdot E \cdot \left[\varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right] \right]} \right]^{\frac{-1}{2}}$$

Cut-off $$L_{r_max} := \frac{\sigma_{flow}}{\sigma_y} \qquad L_{r_max} = 1.27$$

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

Crack Depth, a $a := 20mm$

5. Select category of analysis category 1

6. Define fracture toughness K_I ,K_{IC}, or $K_{0.2}$

$$T_{ref} := 121$$

$$K_{IC} := \left[36.5 + 3.084 \exp\left[0.036 \cdot \left(T - T_{ref} + 56 \right) \right] \right] \cdot MPa \cdot \sqrt{m}$$ From API 579 [5]

$$K_{IC} = 59.655\,MPa \cdot \sqrt{m}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

7. Define crack size a_0

$a_0 := a$ $\quad a_0 = 20\,mm$

8. Calculate L_r $\quad L_r = P/P_L$

Limit load from Zahoor [20]

Semi-elliptical Circumferential Axial Tension 5 <= Ro/t <= 20 $\quad \frac{R_o}{t} = 10.526$

$\theta := \pi \cdot \frac{c}{4 \cdot R_i}$ $\quad \theta = 0.072$

$\zeta := \frac{t}{R_o}$ $\quad \zeta = 0.095$

$x := \frac{a}{t}$ $\quad x = 0.351$

$A1 := x \cdot \frac{\left[(1-\zeta)\cdot(2-2\cdot\zeta+x\cdot\zeta)+(1-\zeta+x\cdot\zeta)^2\right]}{2\cdot\left[1+(2-\zeta)\cdot(1-\zeta)\right]}$ $\quad A1 = 0.164$

$\alpha_1 := \operatorname{acos}(A1 \cdot \sin(\theta))$ $\quad \alpha_1 = 1.559$

$P_L := \begin{cases} \left[2\cdot\pi R\cdot t\cdot\sigma_{flow}\cdot\left[\left(2\cdot\frac{\alpha_1}{\pi}\right)-\left(x\cdot\frac{\theta}{\pi}\right)\cdot\frac{(2-2\cdot\zeta+x\cdot\zeta)}{(2-\zeta)}\right]\right] & \text{if } 5 \le \frac{R_o}{t} \le 20 \\ 0 \quad \text{otherwise} \end{cases}$

$P_L = 68.823\,MN$

$L_R(P) := \frac{P}{P_L}$ $\quad L_R(P_{op}) = 0.363$

9.Calculate K_r

Parameter K from Zahoor [20] $\quad K_I = \sigma_t \cdot (\pi t)^{0.5} \cdot F_t$

The available K-solution of this cracked structure is valid within the following range

$0.05 \le \frac{a}{t} \le 0.8$ $\quad 5 \le \frac{R}{t} \le 160$ $\quad \frac{2\cdot c}{a} \ge 3$

For our problem $\frac{a}{t} = 0.351$ $\quad \frac{R}{t} = 10.026$ $\quad \frac{2.c}{a} = 5$

where $\sigma_t(P) := \frac{P}{2\cdot\pi\cdot R\cdot t}$ $\quad \sigma_t(P_{op}) = 122.143\,MPa$

$\beta := \frac{\left(\frac{a}{t}\right)}{\left(0.25+\frac{a}{c}\right)^{0.58}}$ $\quad \beta = 0.45$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$F_t := \begin{cases} \left[3.702\cdot\beta - 13.475\cdot\beta^2 + 20.0\cdot\beta^3 + 0.0086\cdot\beta\cdot\left(\frac{R}{t} - 5\right)\right] & \text{if } \beta < 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[3.831\cdot\beta - 13.475\cdot\beta^2 + 20.0\cdot\beta^3 + 0.002\cdot\beta\cdot\left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta < 0.25 \wedge 20 \le \frac{R}{t} \le \\ \left[0.25 + 0.4868\cdot\beta + 0.3835\cdot\beta^2 + 0.0086\cdot\beta\cdot\left(\frac{R}{t} - 5\right)\right] & \text{if } \beta \ge 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[0.25 + 0.6158\cdot\beta + 0.3835\cdot\beta^2 + 0.002\cdot\beta\cdot\left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta \ge 0.25 \wedge 20 < \frac{R}{t} \le 16 \\ 0 \quad \text{otherwise} \end{cases}$$

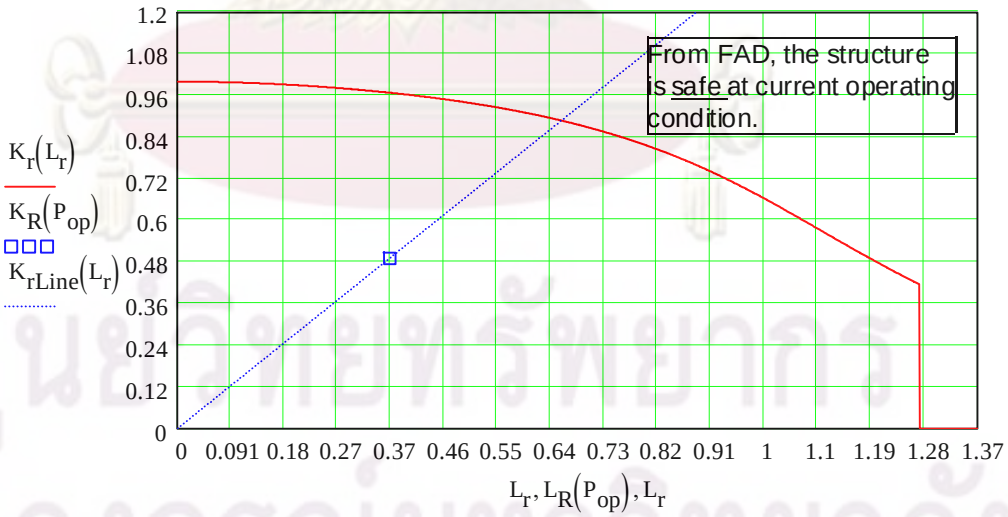
$$K_I(P) := \sigma_t(P)\cdot(\pi t)^{0.5}\cdot F_t \qquad K_I(P_{op}) = 29.285\ \text{MPa}\cdot\sqrt{\text{m}}$$

$$K_R(P) := \frac{K_I(P)}{K_{IC}} \qquad K_R(P_{op}) = 0.491$$

10. Plot assessment piont (L_r,K_r)

$$K_r(L_r) := \begin{cases} K_{rOption2}(L_r) & \text{if } 0 \le L_r \le L_{r_n} \\ 0 \quad \text{otherwise} \end{cases} \qquad L_r := 0, 0.001 .. 1.5$$

$$K_{rLine}(L_r) := \frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})$$



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

$$\left[\frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] = K_{rOption2}(L_r)$$

$$f(L_r) := \left[\frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] - K_{rOption2}(L_r)$$

$L_r := 1$

$L_{rC} := \text{root}(f(L_r), L_r)$ $\quad L_{rC} = 0.657$

$K_{rC} := K_{rOption2}(L_{rC})$ $\quad K_{rC} = 0.888$

$K_{rmax} := K_{rOption2}(L_{r_max})$ $\quad K_{rmax} = 0.416$

L_r Critical $\quad L_{rC1} := \begin{cases} L_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R(P_{op})}{L_R(P_{op})} \\ L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$L_{rC1} = 0.657$

K_r Critical $\quad K_{rC1} := \begin{cases} K_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R(P_{op})}{L_R(P_{op})} \\ \frac{K_R(P_{op})}{L_R(P_{op})}\cdot L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$K_{rC1} = 0.888$

$d := \sqrt{K_R(P_{op})^2 + L_R(P_{op})^2}$

$dC := \sqrt{K_{rC1}{}^2 + L_{rC1}{}^2}$

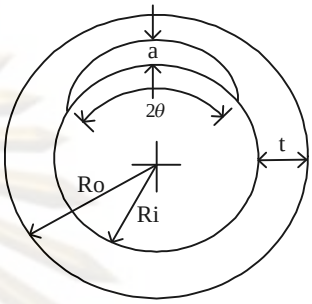
$FL := \frac{dC}{d}$ $\quad FL = 1.808$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.2.1.4 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 2

$GPa := 10^9 \cdot Pa$ $MPa := 10^6 \cdot Pa$ $MN := 10^6 newton$

R6 Option 2 & level 2



Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)

Structure's dimensions

Outer radius, RO	$R_o := 600mm$	
Inner radius, Ri	$R_i := 543mm$	
Thickness, t	$t := R_o - R_i$	$t = 57\,mm$
Mean radius, R	$R := R_i + \left(\frac{t}{2}\right)$	$R = 571.5\,mm$

1. Categorise load & stress

Primary load : Axial Tension, P_{op} $P_{op} := 25MN$

Secondary load : NONE

Operating temperature $T := 121$ {Celcius}

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

Material properties $E := 207200MPa - 57.1(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269 \cdot MPa$

$\sigma_u := 414 \cdot MPa$ $\alpha := 1$

Estinate n from R6

$N := 15$

$$n_E := \left| \begin{array}{l} \text{for } i \in 0..N \\ \quad n_0 \leftarrow 0.01 \\ \quad n_{i+1} \leftarrow \dfrac{\ln\left(\dfrac{1}{0.002 \cdot n_i}\right) - 1}{\ln\left(\dfrac{\sigma_u}{\sigma_y}\right)} \\ n \end{array} \right.$$

$n_E =$

	0
0	0.01
1	22.776
2	4.845
3	8.435

$n := n_{E_{15}}$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3. Select FAD

Option 2 $$K_r(L_r) = \left(\frac{E \cdot \varepsilon_{ref}}{L_r \cdot \sigma_y} + \frac{L_r^3 \cdot \sigma_y}{2 \cdot E \cdot \varepsilon_{ref}} \right)^{\frac{-1}{2}}$$

$$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2} \qquad \sigma_{flow} = 341.5\,MPa$$

Ramberg-Osgood Equation $$\frac{\varepsilon_{ref}}{\varepsilon_0} = \frac{\sigma_{ref}}{\sigma_0} + \alpha \cdot \left(\frac{\sigma_{ref}}{\sigma_0} \right)^n$$

$$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2} \qquad \sigma_{ref} = L_r \cdot \sigma_{flow}$$

$$\sigma_0 := \sigma_y \qquad \varepsilon_0 := \frac{\sigma_0}{E}$$

$$\varepsilon_{ref} = \varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_{flow}}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_{flow}}{\sigma_0} \right)^n \right]$$

$$K_{rOption2}(L_r) := \left[\frac{E \cdot \left[\varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right] \right]}{L_r \cdot \sigma_y} + \frac{L_r^3 \cdot \sigma_y}{2 \cdot E \cdot \left[\varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right] \right]} \right]^{\frac{-1}{2}}$$

Cut-off $$L_{r_max} := \frac{\sigma_{flow}}{\sigma_y} \qquad L_{r_max} = 1.27$$

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

Crack Depth, a $a := 20mm$

5. Select category of analysis category 2

6. Define fracture toughness $K_{0.2}$ and K_g

Material name : SA106Gr. C

$$J = C_1(\Delta a)^{C_2} \qquad C_1 = 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c} \right)^{2.29} \qquad C_2 := 0.38$$

$$s_u := \sigma_u \qquad s_y := \sigma_y$$

$$\varepsilon_c := 0.82 \qquad E = 200.291\,GPa$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$k := 1mm$

$\Delta a := 0.2$

$S_u := \frac{s_u}{E}$ $\quad S_u = 2.067 \times 10^{-3}$

$C_1 := 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c}\right)^{2.29}$ $\quad C_1 = 22.153\,N\,\frac{N}{mm}$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $\quad J(\Delta a) = 12.018\,N\,\frac{N}{mm}$

$K_{0.2} := \sqrt{\frac{(J(\Delta a) \cdot E)}{\left(1 - 0.3^2\right)}}$ $\quad K_{0.2} = 199.191\,MPa \cdot \sqrt{m}$

$\Delta a_g := 2$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $\quad J(\Delta a_g) = 28.829\,N\,\frac{N}{mm}$

$K_g := \sqrt{\frac{\left(J(\Delta a_g) \cdot E\right)}{\left(1 - 0.3^2\right)}}$ $\quad K_g = 308.51\,MPa \cdot \sqrt{m}$

7. Define crack size a_0

Assume $a_0 := a$ $\quad a_0 = 20\,mm$

$\Delta a_g := 2mm$

8. Calculate L_r $\quad L_r = P/P_L$

Original crack size a_0

Limit load from Zahoor [20] $\quad \sigma_{ref} := \sigma_y$

Semi-elliptical Circumferential Axial Tension 5 <= Ro/t <= 20 $\quad \frac{R_o}{t} = 10.526$

$\theta := \pi \cdot \frac{c}{4 \cdot R_i}$ $\quad \theta = 0.072$

$\zeta := \frac{t}{R_o}$ $\quad \zeta = 0.095$

$x := \frac{a}{t}$ $\quad x = 0.351$

$A1 := x \cdot \frac{\left[(1 - \zeta) \cdot (2 - 2 \cdot \zeta + x \cdot \zeta) + (1 - \zeta + x \cdot \zeta)^2\right]}{2 \cdot \left[1 + (2 - \zeta) \cdot (1 - \zeta)\right]}$ $\quad A1 = 0.164$

$\alpha_1 := \operatorname{acos}(A1 \cdot \sin(\theta))$ $\quad \alpha_1 = 1.559$

$P_L := \begin{cases} \left[2 \cdot \pi R \cdot t \cdot \sigma_{flow} \cdot \left[\left(2 \cdot \frac{\alpha_1}{\pi}\right) - \left(x \cdot \frac{\theta}{\pi}\right) \cdot \frac{(2 - 2 \cdot \zeta + x \cdot \zeta)}{(2 - \zeta)}\right]\right] & \text{if } 5 \le \frac{R_o}{t} \le 20 \\ 0 \quad \text{otherwise} \end{cases}$

$P_L = 68.823\,MN$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$L_R(P) := \frac{P}{P_L} \qquad L_R(P_{op}) = 0.363$$

Ductile crack growth a_g

$$\theta := \pi \cdot \frac{c}{4 \cdot R_i} \qquad \theta = 0.072$$

$$\zeta := \frac{t}{R_o} \qquad \zeta = 0.095$$

$$x := \frac{a + \Delta a_g}{t} \qquad x = 0.386$$

$$A1 := x \cdot \frac{\left[(1-\zeta)\cdot(2 - 2\cdot\zeta + x\cdot\zeta) + (1 - \zeta + x\cdot\zeta)^2\right]}{2\cdot\left[1 + (2-\zeta)\cdot(1-\zeta)\right]} \qquad A1 = 0.181$$

$$\alpha_1 := \operatorname{acos}(A1 \cdot \sin(\theta)) \qquad \alpha_1 = 1.558$$

$$P_L := \begin{cases} \left[2\cdot\pi R\cdot t\cdot\sigma_{flow}\cdot\left[\left(2\cdot\frac{\alpha_1}{\pi}\right) - \left(x\cdot\frac{\theta}{\pi}\right)\cdot\frac{(2 - 2\cdot\zeta + x\cdot\zeta)}{(2-\zeta)}\right]\right] & \text{if } 5 \le \frac{R_o}{t} \le 20 \\ 0 \quad \text{otherwise} \end{cases}$$

$$P_L = 68.713\,\text{MN}$$

$$L_{Rg}(P) := \frac{P}{P_L} \qquad L_{Rg}(P_{op}) = 0.364$$

9.Calculate K_r

Parameter K from Zahoor [20] $\quad K_I = \sigma_t \cdot (\pi t)^{0.5} \cdot F_t$

The available K-solution of this cracked structure is valid within the following range

$$0.05 \le \frac{a}{t} \le 0.8 \qquad 5 \le \frac{R}{t} \le 160 \qquad \frac{2\cdot c}{a} \ge 3$$

For our problem $\quad \frac{a}{t} = 0.351 \qquad \frac{R}{t} = 10.026 \qquad \frac{2.c}{a} = 5$

$$\frac{a + \Delta a_g}{t} = 0.386 \qquad \frac{2.c}{a + \Delta a_g} = 4.545$$

Original crack size a_0

where $\quad \sigma_t(P) := \frac{P}{2\cdot\pi\cdot R\cdot t} \qquad \sigma_t(P_{op}) = 122.143\,\text{MPa}$

$$\beta := \frac{\left(\frac{a}{t}\right)}{\left(0.25 + \frac{a}{c}\right)^{0.58}} \qquad \beta = 0.45$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$F_t := \begin{cases} \left[3.702\cdot\beta - 13.475\cdot\beta^2 + 20.0\cdot\beta^3 + 0.0086\cdot\beta\cdot\left(\frac{R}{t} - 5\right)\right] & \text{if } \beta < 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[3.831\cdot\beta - 13.475\cdot\beta^2 + 20.0\cdot\beta^3 + 0.002\cdot\beta\cdot\left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta < 0.25 \wedge 20 \le \frac{R}{t} \le 1 \\ \left[0.25 + 0.4868\cdot\beta + 0.3835\cdot\beta^2 + 0.0086\cdot\beta\cdot\left(\frac{R}{t} - 5\right)\right] & \text{if } \beta \ge 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[0.25 + 0.6158\cdot\beta + 0.3835\cdot\beta^2 + 0.002\cdot\beta\cdot\left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta \ge 0.25 \wedge 20 < \frac{R}{t} \le 160 \\ 0 & \text{otherwise} \end{cases}$$

$$K_I(P) := \sigma_t(P)\cdot(\pi t)^{0.5}\cdot F_t \qquad K_I(P_{op}) = 29.285\ \text{MPa}\cdot\sqrt{\text{m}}$$

$$K_R(P) := \frac{K_I(P)}{K_{0.2}} \qquad K_R(P_{op}) = 0.147$$

Original crack size a_g

Parameter K from Zahoor [20] $K_I = \sigma_t\cdot(\pi t)^{0.5}\cdot F_t$

where $\sigma_t(P) := \frac{P}{2\cdot\pi\cdot R\cdot t}$ $\qquad \sigma_t(P_{op}) = 122.143\ \text{MPa}$

$$\beta := \frac{\left(\frac{a + \Delta a_g}{t}\right)}{\left(0.25 + \frac{a + \Delta a_g}{c}\right)^{0.58}} \qquad \beta = 0.479$$

$$F_t := \begin{cases} \left[3.702\cdot\beta - 13.475\cdot\beta^2 + 20.0\cdot\beta^3 + 0.0086\cdot\beta\cdot\left(\frac{R}{t} - 5\right)\right] & \text{if } \beta < 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[3.831\cdot\beta - 13.475\cdot\beta^2 + 20.0\cdot\beta^3 + 0.002\cdot\beta\cdot\left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta < 0.25 \wedge 20 \le \frac{R}{t} \le 1 \\ \left[0.25 + 0.4868\cdot\beta + 0.3835\cdot\beta^2 + 0.0086\cdot\beta\cdot\left(\frac{R}{t} - 5\right)\right] & \text{if } \beta \ge 0.25 \wedge 5 \le \frac{R}{t} \le 20 \\ \left[0.25 + 0.6158\cdot\beta + 0.3835\cdot\beta^2 + 0.002\cdot\beta\cdot\left(\frac{R}{t} - 20\right)^{0.7}\right] & \text{if } \beta \ge 0.25 \wedge 20 < \frac{R}{t} \le 160 \\ 0 & \text{otherwise} \end{cases}$$

$$K_I(P) := \sigma_t(P)\cdot(\pi t)^{0.5}\cdot F_t \qquad K_I(P_{op}) = 30.576\ \text{MPa}\cdot\sqrt{\text{m}}$$

$$K_{Rg}(P) := \frac{K_I(P)}{K_g} \qquad K_{Rg}(P_{op}) = 0.099$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

10. Plot assessment piont (L_r,K_r)

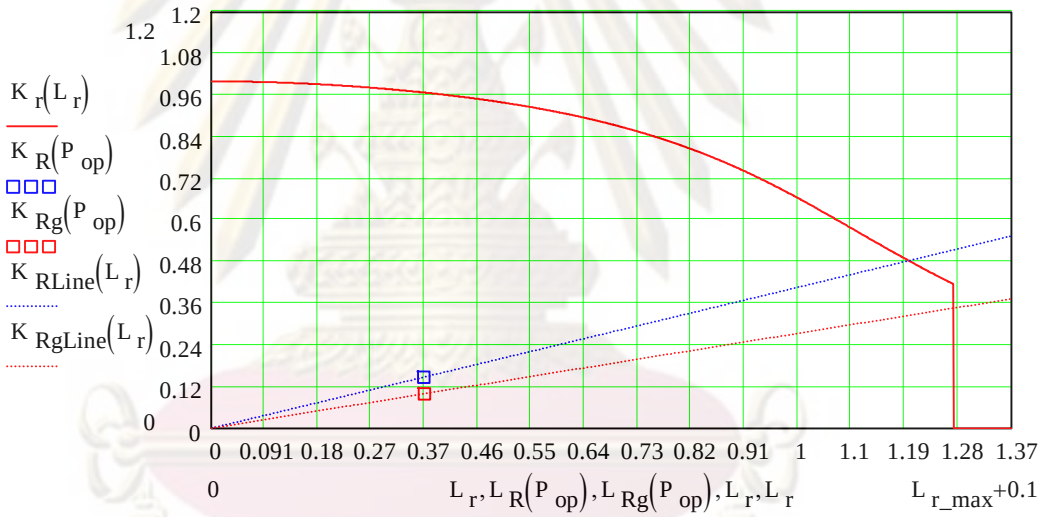
$$K_r(L_r) := \begin{cases} K_{rOption2}(L_r) & \text{if } 0 \le L_r \le L_{r_max} \\ 0 & \text{otherwise} \end{cases} \quad : 0, 0.001 .. 1.5$$

For Original crack size a_0

$$K_{RLine}(L_r) := \frac{K_R(P_{op})}{L_R(P_{op})} \cdot (L_r - L_R(P_{op})) + K_R(P_{op})$$

For Ductile crack growth a_g

$$K_{RgLine}(L_r) := \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \cdot (L_r - L_{Rg}(P_{op})) + K_{Rg}(P_{op})$$



11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

For Original crack size a_0

$$\left[\frac{K_R(P_{op})}{L_R(P_{op})} \cdot (L_r - L_R(P_{op})) + K_R(P_{op})\right] = K_{rOption2}(L_r)$$

$$f(L_r) := \left[\frac{K_R(P_{op})}{L_R(P_{op})} \cdot (L_r - L_R(P_{op})) + K_R(P_{op})\right] - K_{rOption2}(L_r)$$

$L_r := 1$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$L_{rC} := root\left(f\left(L_r\right), L_r\right)$ $L_{rC} = 1.193$

$K_{rC} := K_{rOption2}\left(L_{rC}\right)$ $K_{rC} = 0.483$

$K_{rmax} := K_{rOption2}\left(L_{r_max}\right)$ $K_{rmax} = 0.416$

L_r Critical $L_{rC1} := \begin{cases} L_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R\left(P_{op}\right)}{L_R\left(P_{op}\right)} \\ L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R\left(P_{op}\right)}{L_R\left(P_{op}\right)} \end{cases}$

$L_{rC1} = 1.193$

K_r Critical $K_{rC1} := \begin{cases} K_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R\left(P_{op}\right)}{L_R\left(P_{op}\right)} \\ \dfrac{K_R\left(P_{op}\right)}{L_R\left(P_{op}\right)} \cdot L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R\left(P_{op}\right)}{L_R\left(P_{op}\right)} \end{cases}$

$K_{rC1} = 0.483$

$d := \sqrt{K_R\left(P_{op}\right)^2 + L_R\left(P_{op}\right)^2}$

$dC := \sqrt{K_{rC1}{}^2 + L_{rC1}{}^2}$

$FL := \dfrac{dC}{d}$ $FL = 3.285$

For Ductile crack growth a_g

$$\left[\frac{K_{Rg}\left(P_{op}\right)}{L_{Rg}\left(P_{op}\right)} \cdot \left(L_r - L_{Rg}\left(P_{op}\right)\right) + K_{Rg}\left(P_{op}\right)\right] = K_{rOption2}\left(L_r\right)$$

$$f\left(L_r\right) := \left[\frac{K_{Rg}\left(P_{op}\right)}{L_{Rg}\left(P_{op}\right)} \cdot \left(L_r - L_{Rg}\left(P_{op}\right)\right) + K_{Rg}\left(P_{op}\right)\right] - K_{rOption2}\left(L_r\right)$$

$L_r := 1$

$L_{rC} := root\left(f\left(L_r\right), L_r\right)$ $L_{rC} = 1.335$

$K_{rC} := K_{rOption2}\left(L_{rC}\right)$ $K_{rC} = 0.364$

L_r Critical $\quad L_{rC1} := \begin{cases} L_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \\ L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \end{cases}$

$L_{rC1} = 1.27$

K_r Critical $\quad K_{rC1} := \begin{cases} K_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \\ \dfrac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \cdot L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \end{cases}$

$K_{rC1} = 0.346$

$d := \sqrt{K_{Rg}(P_{op})^2 + L_{Rg}(P_{op})^2}$

$dC := \sqrt{K_{rC1}{}^2 + L_{rC1}{}^2}$

$FL := \dfrac{dC}{d} \qquad FL = 3.489$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.2.2 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวเส้นรอบวง รับภาระโมเมนต์ดัด

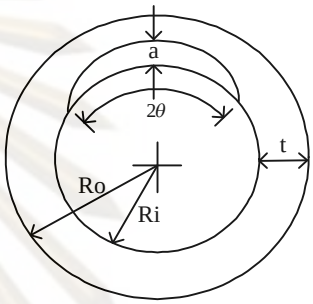
ข.2.2.1 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1

$GPa := 10^9 \cdot Pa$ $MPa := 10^6 \cdot Pa$ $MN := 10^6 newton$

R6 Option 1 & level 1

Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)



Structure's dimensions

Outer radius, RO	$R_o := 600mm$	
Inner radius, Ri	$R_i := 543mm$	
Thickness, t	$t := R_o - R_i$	$t = 57\,mm$
Mean radius, R	$R := R_i + \left(\frac{t}{2}\right)$	$R = 571.5\,mm$

1. Categorise load & stress

Primary load : Bending moment, M_{op} $M_{op} := 29MN \cdot m$

Secondary load : NONE

Operating temperature $T := 121$ {Celcius}

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

Material properties $E := 207200MPa - 57.1(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269 \cdot MPa$

$\sigma_u := 414 \cdot MPa$

3. Select FAD

Option 1 $K_r(L_r) = \left(1 - 0.14 \cdot L_r^2\right) \cdot \left(0.3 + 0.7\exp\left(-0.65 \cdot L_r^6\right)\right)$

$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2}$ $\sigma_{flow} = 341.5\,MPa$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Cut-off $L_{r_max} := \frac{\sigma_{flow}}{\sigma_y}$ $L_{r_max} = 1.27$

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

Crack Depth, a $a := 20mm$

5. Select category of analysis category 1

6. Define fracture toughness K_I ,K_{IC}, or $K_{0.2}$

$T_{ref} := 121$

$K_{IC} := \left[36.5 + 3.084\exp\left[0.036\cdot\left(T - T_{ref} + 56\right)\right]\right]\cdot MPa\cdot\sqrt{m}$ From API 579 [5]

$K_{IC} = 59.655\,MPa\cdot\sqrt{m}$

7. Define crack size a_0

$a_0 := a$ $a_0 = 20\,mm$

8. Calculate L_r $L_r = M/M_L$

LimitLoad from Zahoor [20]

ombined Axial Load and Bending Moment R/t >= 10 and σt/σf < 0.25

Axial Load ,P is equal to zero $P := 0MN$ $\frac{R}{t} = 10.026$

$\theta := \pi\cdot\frac{c}{4R_i}$ $\theta = 0.072$

$\sigma_t := \frac{P}{2\pi R\cdot t}$ $\sigma_t = 0.000\,Pa$ $\frac{\sigma_t}{\sigma_{flow}} = 0$

$\beta 1 := 0.5\pi\cdot\left[1 - \left(\frac{\theta}{\pi}\right)\cdot\left(\frac{a}{t}\right) - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right]$ $\beta 1 = 1.558$

$\beta 2 := \pi\cdot\frac{\left[1 - \frac{a}{t} - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right]}{\left(2 - \frac{a}{t}\right)}$ $\beta 2 = 1.237$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$M_L := \begin{cases} \left[2\cdot\sigma_{flow}\cdot R^2\cdot t\cdot\left[2\cdot\sin(\beta 1)-\left(\frac{a}{t}\right)\cdot\sin(\theta)\right]\right] & \text{if } \frac{R}{t}\geq 10 \wedge \frac{\sigma_t}{\sigma_{flow}} < 0.25 \\ \left[2\cdot\sigma_{flow}\cdot R^2\cdot t\cdot\left(2-\frac{a}{t}\right)\cdot\sin(\beta 2)\right] & \text{if } (\theta+\beta 1) > \pi \\ 0 & \text{otherwise} \end{cases}$$

$\theta + \beta 1 = 1.63$

$M_L = 25106258.73256\,\text{N m}$ $\quad M_L = 25.106\,\text{MN·m}$

$$L_R := \frac{M_{op}}{M_L} \qquad L_R = 1.155$$

9.Calculate K_r

Parameter K from Zahoor [20]

The available K-solution of this cracked structure is valid within the following range

Bending Moment 5 <= R/t <= 160 $\quad 2\cdot\frac{c}{a} = 5$ $\quad \frac{R}{t} = 10.026$

2c/a >= 3

θ/π <= 1 $\quad \frac{\theta}{\pi} = 0.023$

0.05 <= a/t <= 0.8 $\quad \frac{a}{t} = 0.351$

$$B00 := -0.0002\cdot\left(\frac{R}{t}\right)^2 + 0.0173\cdot\left(\frac{R}{t}\right) + 3.3567 \qquad B00 = 3.51$$

$$B01 := -0.00006\cdot\left(\frac{R}{t}\right)^2 + 0.0101\cdot\left(\frac{R}{t}\right) + 1.3153 \qquad B01 = 1.411$$

$$B02 := 0.00002\cdot\left(\frac{R}{t}\right)^2 - 0.0019\cdot\left(\frac{R}{t}\right) - 0.0066 \qquad B02 = -0.024$$

$$B10 := 0.000006\cdot\left(\frac{R}{t}\right)^2 - 0.0017\cdot\left(\frac{R}{t}\right) - 4.2841 \qquad B10 = -4.301$$

$$B11 := 0.00006\cdot\left(\frac{R}{t}\right)^2 - 0.013\cdot\left(\frac{R}{t}\right) - 1.1867 \qquad B11 = -1.311$$

$$B12 := -0.000002\cdot\left(\frac{R}{t}\right)^2 + 0.0004\cdot\left(\frac{R}{t}\right) - 0.0741 \qquad B12 = -0.07$$

$$B20 := -0.0264\ln\left(\frac{R}{t}\right) + 1.7928 \qquad B20 = 1.732$$

$$B21 := 0.147\ln\left(\frac{R}{t}\right) + 0.7238 \qquad B21 = 1.063$$

$$B22 := 0.0058\ln\left(\frac{R}{t}\right) + 0.2269 \qquad B22 = 0.24$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$A0 := B00\cdot\left(\frac{a}{t}\right)^2 + B01\cdot\left(\frac{a}{t}\right) + B02 \qquad A0 = 0.903$$

$$A1 := B10\cdot\left(\frac{a}{t}\right)^2 + B11\cdot\left(\frac{a}{t}\right) + B12 \qquad A1 = -1.06$$

$$A2 := B20\cdot\left(\frac{a}{t}\right)^2 + B21\cdot\left(\frac{a}{t}\right) + B22 \qquad A2 = 0.826$$

$$Fb := A0\cdot\left(\frac{a}{c}\right)^2 + A1\cdot\left(\frac{a}{c}\right) + A2 \qquad Fb = 0.547$$

$$\sigma b := \frac{M_{op}}{\left(\pi\cdot R^2\cdot t\right)} \qquad \sigma b = 495.839\,\text{MPa}$$

$$K_I := \begin{cases} \sigma b\cdot(\pi\cdot t)^{0.5}\cdot Fb & \text{if } 5 \le \frac{R}{t} \le 160 \wedge 0.05 \le \frac{a}{t} \le 0.8 \wedge 2\cdot\frac{c}{a} \ge 3 \wedge \frac{\theta}{\pi} \le 1 \\ 0 & \text{otherwise} \end{cases}$$

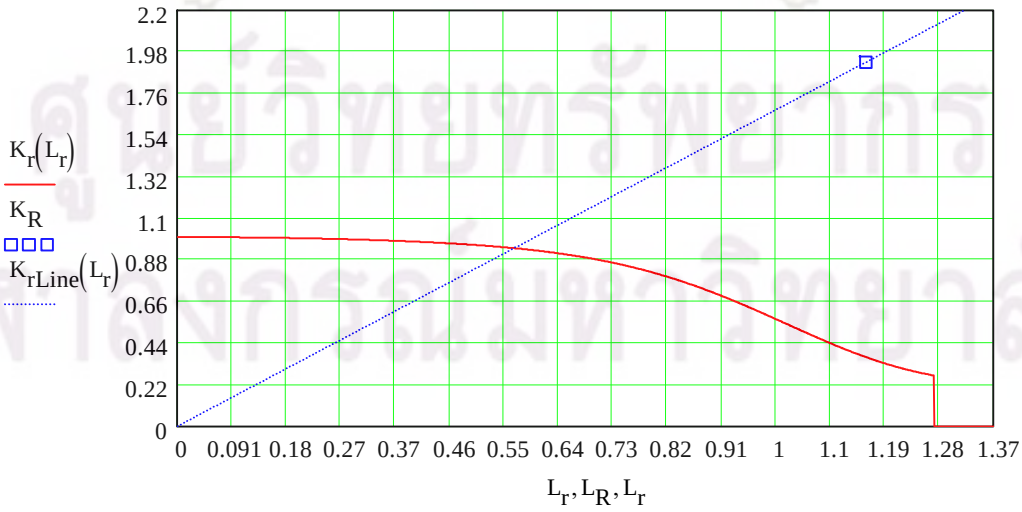
$$K_I = 114775167\,\text{Pa}\cdot\sqrt{\text{m}} \qquad K_I = 114.775\,\text{MPa}\cdot\sqrt{\text{m}}$$

$$K_R := \frac{K_I}{K_{IC}} \qquad K_R = 1.924$$

10. Plot assessment piont (L_r, K_r) $\qquad L_r := 0, 0.001 .. 1.5$

$$K_r(L_r) := \begin{cases} \left[\left(1 - 0.14\cdot L_r^2\right)\cdot\left(0.3 + 0.7\exp\left(-0.65\cdot L_r^6\right)\right)\right] & \text{if } 0 \le L_r \le L_{r_max} \\ 0 & \text{otherwise} \end{cases}$$

$$K_{rLine}(L_r) := \frac{K_R}{L_R}\cdot(L_r - L_R) + K_R$$



11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

$$\left[\frac{K_R}{L_R}\cdot\left(L_r - L_R\right) + K_R\right] = \left(1 - 0.14\cdot L_r^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_r^{\,6}}\right)$$

$$f\left(L_r\right) := \left[\frac{K_R}{L_R}\cdot\left(L_r - L_R\right) + K_R\right] - \left(1 - 0.14\cdot L_r^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_r^{\,6}}\right)$$

$L_r := 1$

$L_{rC} := \mathrm{root}\left(f\left(L_r\right), L_r\right)$ $\quad L_{rC} = 0.565$

$K_{rC} := \left(1 - 0.14\cdot L_{rC}^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_{rC}^{\,6}}\right)$ $\quad K_{rC} = 0.941$

$K_{rmax} := \left(1 - 0.14\cdot L_{r_max}^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_{r_max}^{\,6}}\right)$ $\quad K_{rmax} = 0.268$

L_r Critical

$$L_{rC1} := \begin{cases} L_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R}{L_R} \\ L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R}{L_R} \end{cases}$$

$L_{rC1} = 0.565$

K_r Critical

$$K_{rC1} := \begin{cases} K_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R}{L_R} \\ \dfrac{K_R}{L_R}\cdot L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R}{L_R} \end{cases}$$

$K_{rC1} = 0.941$

$d := \sqrt{K_R^{\,2} + L_R^{\,2}}$

$dC := \sqrt{K_{rC1}^{\,2} + L_{rC1}^{\,2}}$

$FL := \dfrac{dC}{d}$ $\quad FL = 0.489$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

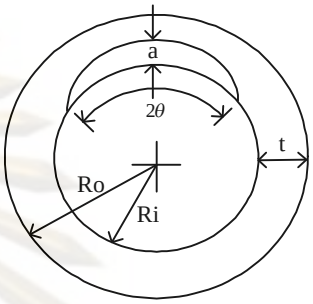
ข.2.2.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 2

$GPa := 10^9 \cdot Pa$ $MPa := 10^6 \cdot Pa$ $MN := 10^6 newton$

R6 Option 1 & Level 2

Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)



Structure's dimensions

Outer radius, RO $R_o := 600mm$

Inner radius, Ri $R_i := 543mm$

Thickness, t $t := R_o - R_i$ $t = 57\,mm$

Mean radius, R $R := R_i + \left(\frac{t}{2}\right)$ $R = 571.5\,mm$

1. Categorise load & stress

Primary load : Bemding moment, M_{op} $M_{op} := 29MN \cdot m$

Secondary load : NONE

Operating temperature $T := 121$ {Celcius}

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

$E := 207200MPa - 57.1(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269 \cdot MPa$

$\sigma_u := 414 \cdot MPa$

3. Select FAD

Option 1 $K_r(L_r) = \left(1 - 0.14 \cdot L_r^2\right) \cdot \left(0.3 + 0.7 \exp\left(-0.65 \cdot L_r^6\right)\right)$

$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2}$ $\sigma_{flow} = 341.5\,MPa$

Cut-off $L_{r_max} := \frac{\sigma_{flow}}{\sigma_y}$ $L_{r_max} = 1.27$

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Crack Depth, a $a := 20mm$

5. Select category of analysis category 2

6. Define fracture toughness $K_{0.2}$ and K_g

Material name : SA106Gr. C

$J = C_1(\Delta a)^{C_2}$ $C_1 = 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c}\right)^{2.29}$ $C_2 := 0.38$

$s_u := \sigma_u$ $s_y := \sigma_y$

$\varepsilon_c := 0.82$ $E = 200.291\,GPa$

$k := 1mm$

$\Delta a := 0.2$

$S_u := \frac{s_u}{E}$ $S_u = 2.067 \times 10^{-3}$

$C_1 := 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c}\right)^{2.29}$ $C_1 = 332.298\,\frac{N}{mm}$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $J(\Delta a) = 180.268\,\frac{N}{mm}$

$K_{0.2} := \sqrt{\frac{(J(\Delta a) \cdot E)}{\left(1 - 0.3^2\right)}}$ $K_{0.2} = 199.191\,MPa \cdot \sqrt{m}$

$\Delta a_g := 2$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $J(\Delta a_g) = 432.433\,\frac{N}{mm}$

$K_g := \sqrt{\frac{\left(J(\Delta a_g) \cdot E\right)}{\left(1 - 0.3^2\right)}}$ $K_g = 308.51\,MPa \cdot \sqrt{m}$

7. Define crack size a_0 and a_g

$a_0 := a$ $a_0 = 20\,mm$

Assume $\Delta a_g := 2mm$

8. Calculate L_r $L_r = M/M_L$

Original crack size a_0

Limit load from Zahoor [20]

ombined Axial Load and Bending Moment R/t >= 10 and $\sigma t/\sigma f < 0.25$

Axial Load ,P is equal zero $P := 0MN$ $\frac{R}{t} = 10.026$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$\theta := \pi \cdot \frac{c}{4R_i}$ $\theta = 0.072$

$\sigma_t := \frac{P}{2\pi R \cdot t}$ $\sigma_t = 0.000\,\text{Pa}$ $\frac{\sigma_t}{\sigma_{flow}} = 0$

$\beta 1 := 0.5\pi \cdot \left[1 - \left(\frac{\theta}{\pi}\right) \cdot \left(\frac{a}{t}\right) - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right]$ $\beta 1 = 1.558$

$\beta 2 := \pi \cdot \frac{\left[1 - \frac{a}{t} - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right]}{\left(2 - \frac{a}{t}\right)}$ $\beta 2 = 1.237$

$$M_L := \begin{cases} \left[2 \cdot \sigma_{flow} \cdot R^2 \cdot t \cdot \left[2 \cdot \sin(\beta 1) - \left(\frac{a}{t}\right) \cdot \sin(\theta)\right]\right] & \text{if } \frac{R}{t} \geq 10 \wedge \frac{\sigma_t}{\sigma_{flow}} < 0.25 \\ \left[2 \cdot \sigma_{flow} \cdot R^2 \cdot t \cdot \left(2 - \frac{a}{t}\right) \cdot \sin(\beta 2)\right] & \text{if } (\theta + \beta 1) > \pi \\ 0 & \text{otherwise} \end{cases}$$

$M_L = 25.106\,\text{MN}\cdot\text{m}$

$L_R := \frac{M_{op}}{M_L}$ $L_R = 1.155$

Ductile crack growth a_g

Limit load from Zahoor [20]

ombined Axial Load and Bending Moment R/t >= 10 and σt/σf < 0.25

Axial Load ,P is equal zero $P := 0\text{MN}$ $\frac{R}{t} = 10.026$

$\theta := \pi \cdot \frac{c}{4R_i}$ $\theta = 0.072$

$\sigma_t := \frac{P}{2\pi R \cdot t}$ $\sigma_t = 0.000\,\text{Pa}$ $\frac{\sigma_t}{\sigma_{flow}} = 0$

$\beta 1 := 0.5\pi \cdot \left[1 - \left(\frac{\theta}{\pi}\right) \cdot \left(\frac{a + \Delta a_g}{t}\right) - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right]$ $\beta 1 = 1.557$

$\beta 2 := \pi \cdot \frac{\left[1 - \frac{a + \Delta a_g}{t} - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right]}{\left(2 - \frac{a + \Delta a_g}{t}\right)}$ $\beta 2 = 1.237$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$M_L := \begin{cases} \left[2\cdot\sigma_{flow}\cdot R^2\cdot t\cdot\left[2\cdot\sin(\beta 1) - \left(\frac{a+\Delta a_g}{t}\right)\cdot\sin(\theta)\right]\right] & \text{if } \frac{R}{t} \geq 10 \wedge \frac{\sigma_t}{\sigma_{flow}} < 0.25 \\ \left[2\cdot\sigma_{flow}\cdot R^2\cdot t\cdot\left(2 - \frac{a+\Delta a_g}{t}\right)\cdot\sin(\beta 2)\right] & \text{if } (\theta+\beta 1) > \pi \\ 0 & \text{otherwise} \end{cases}$$

$M_L = 25.074\,MN\cdot m$

$L_{Rg} := \frac{M_{op}}{M_L}$ $L_{Rg} = 1.157$

9.Calculate K_r

Original crack size a_0

Parameter K from Zahoor [20]

The available K-solution of this cracked structure is valid within the following range

Bending Moment 5 <= R/t <= 160 $2\cdot\frac{c}{a} = 5$ $\frac{R}{t} = 10.026$

2c/a >= 3

θ/π <= 1 $\frac{\theta}{\pi} = 0.023$

0.05 <= a/t <= 0.8 $\frac{a}{t} = 0.351$

$B00 := -0.0002\cdot\left(\frac{R}{t}\right)^2 + 0.0173\cdot\left(\frac{R}{t}\right) + 3.3567$ $B00 = 3.51$

$B01 := -0.00006\cdot\left(\frac{R}{t}\right)^2 + 0.0101\cdot\left(\frac{R}{t}\right) + 1.3153$ $B01 = 1.411$

$B02 := 0.00002\cdot\left(\frac{R}{t}\right)^2 - 0.0019\cdot\left(\frac{R}{t}\right) - 0.0066$ $B02 = -0.024$

$B10 := 0.000006\cdot\left(\frac{R}{t}\right)^2 - 0.0017\cdot\left(\frac{R}{t}\right) - 4.2841$ $B10 = -4.301$

$B11 := 0.00006\cdot\left(\frac{R}{t}\right)^2 - 0.013\cdot\left(\frac{R}{t}\right) - 1.1867$ $B11 = -1.311$

$B12 := -0.000002\cdot\left(\frac{R}{t}\right)^2 + 0.0004\cdot\left(\frac{R}{t}\right) - 0.0741$ $B12 = -0.07$

$B20 := -0.0264\ln\left(\frac{R}{t}\right) + 1.7928$ $B20 = 1.732$

$B21 := 0.147\ln\left(\frac{R}{t}\right) + 0.7238$ $B21 = 1.063$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$B22 := 0.0058 \ln\left(\frac{R}{t}\right) + 0.2269$ $B22 = 0.24$

$A0 := B00 \cdot \left(\frac{a}{t}\right)^2 + B01 \cdot \left(\frac{a}{t}\right) + B02$ $A0 = 0.903$

$A1 := B10 \cdot \left(\frac{a}{t}\right)^2 + B11 \cdot \left(\frac{a}{t}\right) + B12$ $A1 = -1.06$

$A2 := B20 \cdot \left(\frac{a}{t}\right)^2 + B21 \cdot \left(\frac{a}{t}\right) + B22$ $A2 = 0.826$

$Fb := A0 \cdot \left(\frac{a}{c}\right)^2 + A1 \cdot \left(\frac{a}{c}\right) + A2$ $Fb = 0.547$

$\sigma b := \frac{M_{op}}{\left(\pi \cdot R^2 \cdot t\right)}$ $\sigma b = 495.839\ \text{MPa}$

$$K_I := \begin{cases} \sigma b \cdot (\pi \cdot t)^{0.5} \cdot Fb & \text{if } 5 \le \frac{R}{t} \le 160 \wedge 0.05 \le \frac{a}{t} \le 0.8 \wedge 2 \cdot \frac{c}{a} \ge 3 \wedge \frac{\theta}{\pi} \le 1 \\ 0 & \text{otherwise} \end{cases}$$

$K_I = 114775167\ \text{Pa} \cdot \sqrt{\text{m}}$ $K_I = 114.775\ \text{MPa} \cdot \sqrt{\text{m}}$

$K_R := \frac{K_I}{K_{0.2}}$ $K_R = 0.576$

Ductile crack growth a_g

Bending Moment 5 <= R/t <= 160 $2 \cdot \frac{c}{a + \Delta a_g} = 4.545$ $\frac{R}{t} = 10.026$

2c/a >= 3

$\theta/\pi <= 1$ $\frac{\theta}{\pi} = 0.023$

0.05 <= a/t <= 0.8 $\frac{a + \Delta a_g}{t} = 0.386$

$B00 := -0.0002 \cdot \left(\frac{R}{t}\right)^2 + 0.0173 \cdot \left(\frac{R}{t}\right) + 3.3567$ $B00 = 3.51$

$B01 := -0.00006 \cdot \left(\frac{R}{t}\right)^2 + 0.0101 \cdot \left(\frac{R}{t}\right) + 1.3153$ $B01 = 1.411$

$B02 := 0.00002 \cdot \left(\frac{R}{t}\right)^2 - 0.0019 \cdot \left(\frac{R}{t}\right) - 0.0066$ $B02 = -0.024$

$B10 := 0.000006 \cdot \left(\frac{R}{t}\right)^2 - 0.0017 \cdot \left(\frac{R}{t}\right) - 4.2841$ $B10 = -4.301$

$B11 := 0.00006 \cdot \left(\frac{R}{t}\right)^2 - 0.013 \cdot \left(\frac{R}{t}\right) - 1.1867$ $B11 = -1.311$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$B12 := -0.000002\cdot\left(\frac{R}{t}\right)^2 + 0.0004\cdot\left(\frac{R}{t}\right) - 0.0741 \qquad B12 = -0.07$$

$$B20 := -0.0264\ln\left(\frac{R}{t}\right) + 1.7928 \qquad B20 = 1.732$$

$$B21 := 0.147\ln\left(\frac{R}{t}\right) + 0.7238 \qquad B21 = 1.063$$

$$B22 := 0.0058\ln\left(\frac{R}{t}\right) + 0.2269 \qquad B22 = 0.24$$

$$A0 := B00\cdot\left(\frac{a + \Delta a_g}{t}\right)^2 + B01\cdot\left(\frac{a + \Delta a_g}{t}\right) + B02 \qquad A0 = 1.044$$

$$A1 := B10\cdot\left(\frac{a + \Delta a_g}{t}\right)^2 + B11\cdot\left(\frac{a + \Delta a_g}{t}\right) + B12 \qquad A1 = -1.217$$

$$A2 := B20\cdot\left(\frac{a + \Delta a_g}{t}\right)^2 + B21\cdot\left(\frac{a + \Delta a_g}{t}\right) + B22 \qquad A2 = 0.908$$

$$Fb := A0\cdot\left(\frac{a + \Delta a_g}{c}\right)^2 + A1\cdot\left(\frac{a + \Delta a_g}{c}\right) + A2 \qquad Fb = 0.575$$

$$\sigma b := \frac{M_{op}}{\left(\pi\cdot R^2\cdot t\right)} \qquad \sigma b = 495.839\,\text{MPa}$$

$$K_I := \begin{cases} \sigma b\cdot(\pi\cdot t)^{0.5}\cdot Fb & \text{if } 5 \le \frac{R}{t} \le 160 \wedge 0.05 \le \frac{a + \Delta a_g}{t} \le 0.8 \wedge 2\cdot\frac{c}{a} \ge 3 \wedge \frac{\theta}{\pi} \le \\ 0 & \text{otherwise} \end{cases}$$

$$K_I = 120653929\,\text{Pa}\cdot\sqrt{\text{m}} \qquad K_I = 120.654\,\text{MPa}\cdot\sqrt{\text{m}}$$

$$K_{Rg} := \frac{K_I}{K_g} \qquad K_{Rg} = 0.391$$

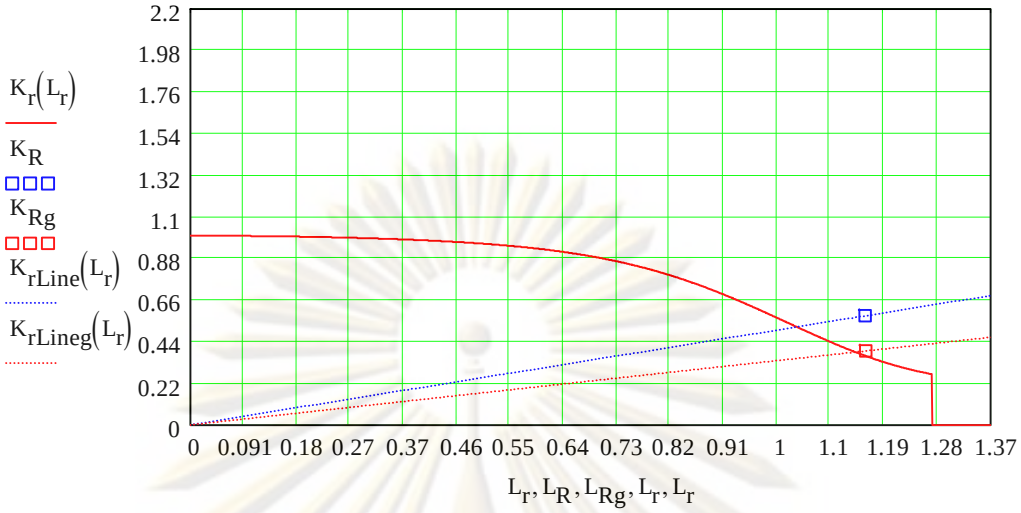
10. Plot assessment piont (L_r, K_r) $\qquad L_r := 0, 0.001 .. 1.5$

$$K_r(L_r) := \begin{cases} \left[\left(1 - 0.14\cdot L_r^2\right)\cdot\left(0.3 + 0.7\exp\left(-0.65\cdot L_r^6\right)\right)\right] & \text{if } 0 \le L_r \le L_{r_max} \\ 0 & \text{otherwise} \end{cases}$$

$$K_{rLine}(L_r) := \frac{K_R}{L_R}\cdot\left(L_r - L_R\right) + K_R$$

$$K_{rLineg}(L_r) := \frac{K_{Rg}}{L_{Rg}}\cdot\left(L_r - L_{Rg}\right) + K_{Rg}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

Original crack size a_0

$$\left[\frac{K_R}{L_R}\cdot\left(L_r - L_R\right) + K_R\right] = \left(1 - 0.14\cdot L_r^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_r^{\,6}}\right)$$

$$f(L_r) := \left[\frac{K_R}{L_R}\cdot\left(L_r - L_R\right) + K_R\right] - \left(1 - 0.14\cdot L_r^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_r^{\,6}}\right) \qquad L_r := 1$$

$L_{rC} := \text{root}(f(L_r), L_r)$ $\qquad L_{rC} = 1.038$

$K_{rC} := \left(1 - 0.14\cdot L_{rC}^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_{rC}^{\,6}}\right)$ $\qquad K_{rC} = 0.518$

$K_{rmax} := \left(1 - 0.14\cdot L_{r_max}^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_{r_max}^{\,6}}\right)$ $\qquad K_{rmax} = 0.268$

L_r Critical $\qquad L_{rC1} := \begin{cases} L_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R}{L_R} \\ L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R}{L_R} \end{cases}$

$L_{rC1} = 1.038$

K_r Critical $\qquad K_{rC1} := \begin{cases} K_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R}{L_R} \\ \dfrac{K_R}{L_R}\cdot L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R}{L_R} \end{cases}$

$K_{rC1} = 0.518$

$$d := \sqrt{K_R^{\,2} + L_R^{\,2}} \qquad dC := \sqrt{K_{rC}^{\,2} + L_{rC}^{\,2}}$$

$$FL := \frac{dC}{d} \qquad FL = 0.899$$

Ductile crack growth a_g

$$f(L_r) := \left[\frac{K_{Rg}}{L_{Rg}}\cdot(L_r - L_{Rg}) + K_{Rg}\right] - \left(1 - 0.14\cdot L_r^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_r^{\,6}}\right) \qquad L_r := 1$$

$$L_{rC} := \text{root}(f(L_r), L_r) \qquad L_{rC} = 1.138$$

$$K_{rC} := \left(1 - 0.14\cdot L_{rC}^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_{rC}^{\,6}}\right) \qquad K_{rC} = 0.385$$

L_r Critical

$$L_{rC1} := \begin{cases} L_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_{Rg}}{L_{Rg}} \\ L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_{Rg}}{L_{Rg}} \end{cases}$$

$$L_{rC1} = 1.138$$

K_r Critical

$$K_{rC1} := \begin{cases} K_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_{Rg}}{L_{Rg}} \\ \dfrac{K_{Rg}}{L_{Rg}}\cdot L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_{Rg}}{L_{Rg}} \end{cases}$$

$$K_{rC1} = 0.385$$

$$d := \sqrt{K_{Rg}^{\,2} + L_{Rg}^{\,2}}$$

$$dC := \sqrt{K_{rC1}^{\,2} + L_{rC1}^{\,2}}$$

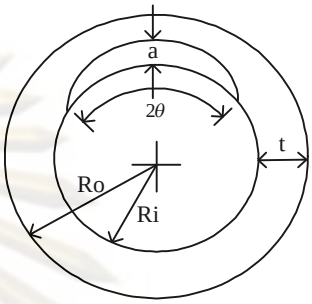
$$FL := \frac{dC}{d} \qquad FL = 0.984$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.2.2.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1

$GPa := 10^9 \cdot Pa$ $MPa := 10^6 \cdot Pa$ $MN := 10^6 newton$

R6 Option 2 & level 1



Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)

Structure's dimensions

Outer radius, RO $R_o := 600mm$

Inner radius, Ri $R_i := 543mm$

Thickness, t $t := R_o - R_i$ $t = 57\,mm$

Mean radius, R $R := R_i + \left(\frac{t}{2}\right)$ $R = 571.5\,mm$

1. Categorise load & stress

Primary load : Bending moment, M_{op} $M_{op} := 29MN \cdot m$

Secondary load : NONE

Operating temperature $T := 121$ {Celcius}

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

Material properties $E := 207200MPa - 57.1(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269 \cdot MPa$

$\sigma_u := 414 \cdot MPa$ $\alpha := 1$

Estinate n from R6

$N := 15$

$$n_E := \left| \begin{array}{l} \text{for } i \in 0..N \\ \quad n_0 \leftarrow 0.01 \\ \quad n_{i+1} \leftarrow \dfrac{\ln\left(\dfrac{1}{0.002 \cdot n_i}\right) - 1}{\ln\left(\dfrac{\sigma_u}{\sigma_y}\right)} \\ n \end{array} \right.$$

$n_E =$

	0
0	0.01
1	22.776
2	4.845
3	8.435

$n := n_{E_{15}}$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3. Select FAD

Option 2 $$K_r(L_r) = \left(\frac{E\cdot\varepsilon_{ref}}{L_r\cdot\sigma_y} + \frac{L_r^{3}\cdot\sigma_y}{2\cdot E\cdot\varepsilon_{ref}}\right)^{\frac{-1}{2}}$$

$$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2} \qquad \sigma_{flow} = 341.5\,MPa$$

Ramberg-Osgood Equation $$\frac{\varepsilon_{ref}}{\varepsilon_0} = \frac{\sigma_{ref}}{\sigma_0} + \alpha\cdot\left(\frac{\sigma_{ref}}{\sigma_0}\right)^{n}$$

$$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2} \qquad \sigma_{ref} = L_r\cdot\sigma_y$$

$$\sigma_0 := \sigma_y \qquad \varepsilon_0 := \frac{\sigma_0}{E}$$

$$\varepsilon_{ref} = \varepsilon_0\cdot\left[\frac{L_r\cdot\sigma_y}{\sigma_0} + \alpha\cdot\left(\frac{L_r\cdot\sigma_y}{\sigma_0}\right)^{n}\right]$$

$$K_{rOption2}(L_r) := \left[\frac{E\cdot\left[\varepsilon_0\cdot\left[\frac{L_r\cdot\sigma_y}{\sigma_0} + \alpha\cdot\left(\frac{L_r\cdot\sigma_y}{\sigma_0}\right)^{n}\right]\right]}{L_r\cdot\sigma_y} + \frac{L_r^{3}\cdot\sigma_y}{2\cdot E\cdot\left[\varepsilon_0\cdot\left[\frac{L_r\cdot\sigma_y}{\sigma_0} + \alpha\cdot\left(\frac{L_r\cdot\sigma_y}{\sigma_0}\right)^{n}\right]\right]}\right]^{\frac{-1}{2}}$$

Cut-off $$L_{r_max} := \frac{\sigma_{flow}}{\sigma_y} \qquad L_{r_max} = 1.27$$

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

Crack Depth, a $a := 20mm$

5. Select category of analysis category 1

6. Define fracture toughness K_I ,K_{IC}, or $K_{0.2}$

$T_{ref} := 121$

$K_{IC} := \left[36.5 + 3.084\exp\left[0.036\cdot\left(T - T_{ref} + 56\right)\right]\right]\cdot MPa\cdot\sqrt{m}$ From API 579 [5]

$K_{IC} = 59.655\,MPa\cdot\sqrt{m}$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

7. Define crack size a_0

$a_0 := a$ $\quad a_0 = 20\,mm$

8. Calculate L_r $\quad L_r = M/M_L$

Limit load from Zahoor [20]

ombined Axial Load and Bending Moment R/t >= 10 and σt/σf < 0.25

Axial Load ,P is equal to zero $\quad P := 0MN$ $\quad \frac{R}{t} = 10.026$

$\theta := \pi \cdot \frac{c}{4R_i}$ $\quad \theta = 0.072$

$\sigma_t := \frac{P}{2\pi R \cdot t}$ $\quad \sigma_t = 0.000\,Pa$ $\quad \frac{\sigma_t}{\sigma_{flow}} = 0$

$\beta 1 := 0.5\pi \cdot \left[1 - \left(\frac{\theta}{\pi}\right) \cdot \left(\frac{a}{t}\right) - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right]$ $\quad \beta 1 = 1.558$

$\beta 2 := \pi \cdot \frac{\left[1 - \frac{a}{t} - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right]}{\left(2 - \frac{a}{t}\right)}$ $\quad \beta 2 = 1.237$

$$M_L := \begin{cases} \left[2 \cdot \sigma_{flow} \cdot R^2 \cdot t \cdot \left[2 \cdot \sin(\beta 1) - \left(\frac{a}{t}\right) \cdot \sin(\theta)\right]\right] & \text{if } \frac{R}{t} \geq 10 \wedge \frac{\sigma_t}{\sigma_{flow}} < 0.25 \\ \left[2 \cdot \sigma_{flow} \cdot R^2 \cdot t \cdot \left(2 - \frac{a}{t}\right) \cdot \sin(\beta 2)\right] & \text{if } (\theta + \beta 1) > \pi \\ 0 & \text{otherwise} \end{cases}$$

$\theta + \beta 1 = 1.63$

$M_L = 25.106\,MN \cdot m$

$L_R := \frac{M_{op}}{M_L}$ $\quad L_R = 1.155$

9.Calculate K_r

Parameter K from Zahoor [20]

The available K-solution of this cracked structure is valid within the following range

Bending Moment 5 <= R/t <= 160 $\quad 2 \cdot \frac{c}{a} = 5$ $\quad \frac{R}{t} = 10.026$

2c/a >= 3

$\theta/\pi <= 1$ $\quad \frac{\theta}{\pi} = 0.023$

0.05 <= a/t <= 0.8 $\quad \frac{a}{t} = 0.351$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$B00 := -0.0002\cdot\left(\frac{R}{t}\right)^2 + 0.0173\cdot\left(\frac{R}{t}\right) + 3.3567 \qquad B00 = 3.51$$

$$B01 := -0.00006\cdot\left(\frac{R}{t}\right)^2 + 0.0101\cdot\left(\frac{R}{t}\right) + 1.3153 \qquad B01 = 1.411$$

$$B02 := 0.00002\cdot\left(\frac{R}{t}\right)^2 - 0.0019\cdot\left(\frac{R}{t}\right) - 0.0066 \qquad B02 = -0.024$$

$$B10 := 0.000006\cdot\left(\frac{R}{t}\right)^2 - 0.0017\cdot\left(\frac{R}{t}\right) - 4.2841 \qquad B10 = -4.301$$

$$B11 := 0.00006\cdot\left(\frac{R}{t}\right)^2 - 0.013\cdot\left(\frac{R}{t}\right) - 1.1867 \qquad B11 = -1.311$$

$$B12 := -0.000002\cdot\left(\frac{R}{t}\right)^2 + 0.0004\cdot\left(\frac{R}{t}\right) - 0.0741 \qquad B12 = -0.07$$

$$B20 := -0.0264\ln\left(\frac{R}{t}\right) + 1.7928 \qquad B20 = 1.732$$

$$B21 := 0.147\ln\left(\frac{R}{t}\right) + 0.7238 \qquad B21 = 1.063$$

$$B22 := 0.0058\ln\left(\frac{R}{t}\right) + 0.2269 \qquad B22 = 0.24$$

$$A0 := B00\cdot\left(\frac{a}{t}\right)^2 + B01\cdot\left(\frac{a}{t}\right) + B02 \qquad A0 = 0.903$$

$$A1 := B10\cdot\left(\frac{a}{t}\right)^2 + B11\cdot\left(\frac{a}{t}\right) + B12 \qquad A1 = -1.06$$

$$A2 := B20\cdot\left(\frac{a}{t}\right)^2 + B21\cdot\left(\frac{a}{t}\right) + B22 \qquad A2 = 0.826$$

$$Fb := A0\cdot\left(\frac{a}{c}\right)^2 + A1\cdot\left(\frac{a}{c}\right) + A2 \qquad Fb = 0.547$$

$$\sigma b := \frac{M_{op}}{\left(\pi\cdot R^2\cdot t\right)} \qquad \sigma b = 495.839\ \text{MPa}$$

$$K_I := \begin{cases} \sigma b\cdot(\pi\cdot t)^{0.5}\cdot Fb & \text{if } 5 \le \frac{R}{t} \le 160 \wedge 0.05 \le \frac{a}{t} \le 0.8 \wedge 2\cdot\frac{c}{a} \ge 3 \wedge \frac{\theta}{\pi} \le 1 \\ 0 & \text{otherwise} \end{cases}$$

$$K_I = 114775167\ \text{Pa}\cdot\sqrt{\text{m}} \qquad K_I = 114.775\ \text{MPa}\cdot\sqrt{\text{m}}$$

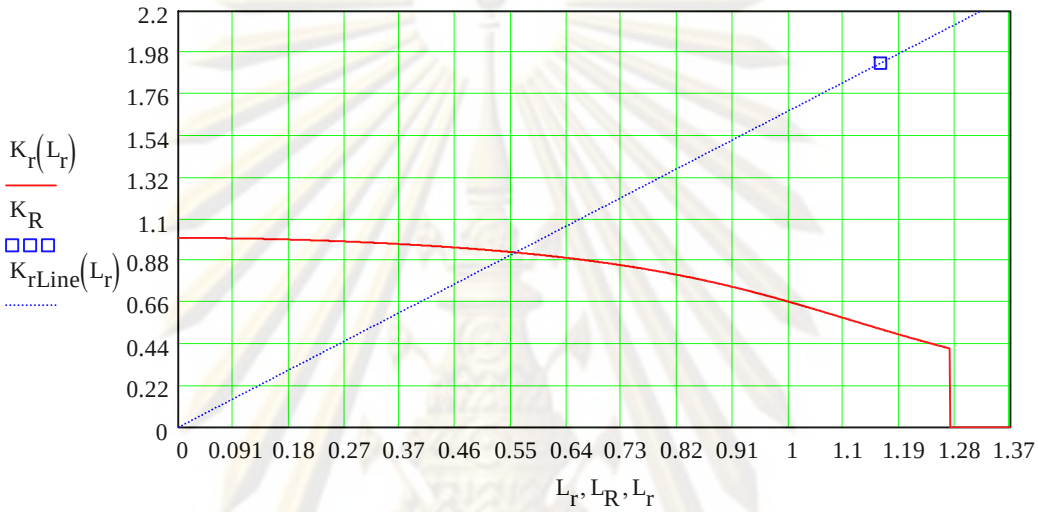
$$K_R := \frac{K_I}{K_{IC}} \qquad K_R = 1.924$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

10. Plot assessment piont (L_r,K_r)

$$K_r(L_r) := \begin{cases} K_{rOption2}(L_r) & \text{if } 0 \le L_r \le L_{r_max} \\ 0 & \text{otherwise} \end{cases} \quad 0, 0.001 .. 1.5$$

$$K_{rLine}(L_r) := \frac{K_R}{L_R} \cdot (L_r - L_R) + K_R$$



11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

$$\left[\frac{K_R}{L_R} \cdot (L_r - L_R) + K_R\right] = K_{rOption2}(L_r)$$

$$f(L_r) := \left[\frac{K_R}{L_R} \cdot (L_r - L_R) + K_R\right] - K_{rOption2}(L_r) \qquad L_r := 1$$

$$L_{rC} := \text{root}(f(L_r), L_r) \qquad L_{rC} = 0.554$$

$$K_{rC} := K_{rOption2}(L_{rC}) \qquad K_{rC} = 0.923$$

$$K_{rmax} := \left(1 - 0.14 \cdot L_{r_max}^{2}\right) \cdot \left(0.3 + 0.7 \cdot e^{-0.65 \cdot L_{r_max}^{6}}\right) \qquad K_{rmax} = 0.268$$

L_r Critical

$$L_{rC1} := \begin{cases} L_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R}{L_R} \\ L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R}{L_R} \end{cases}$$

$$L_{rC1} = 0.554$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

K_r Critical

$$K_{rC1} := \begin{cases} K_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R}{L_R} \\ \frac{K_R}{L_R} \cdot L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R}{L_R} \end{cases}$$

$K_{rC1} = 0.923$

$$d := \sqrt{K_R^{\,2} + L_R^{\,2}}$$

$$dC := \sqrt{K_{rC1}^{\,2} + L_{rC1}^{\,2}}$$

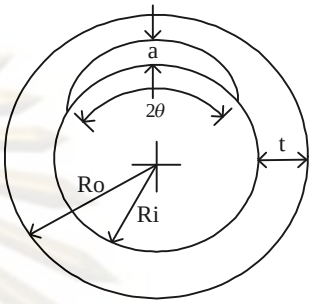
$$FL := \frac{dC}{d} \qquad FL = 0.48$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.2.2.4 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 2

$GPa := 10^9 \cdot Pa$ $MPa := 10^6 \cdot Pa$ $MN := 10^6 newton$

R6 Option 2 & level 2



Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)

Structure's dimensions

Outer radius, RO	$R_o := 600mm$	
Inner radius, Ri	$R_i := 543mm$	
Thickness, t	$t := R_o - R_i$	$t = 57\,mm$
Mean radius, R	$R := R_i + \left(\frac{t}{2}\right)$	$R = 571.5\,mm$

1. Categorise load & stress

Primary load : Bendign moment, M_{op} $M_{op} := 29MN \cdot m$

Secondary load : NONE

Operating temperature $T := 121$ {Celcius}

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

Material properties $E := 207200MPa - 57.1(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269 \cdot MPa$

$\sigma_u := 414 \cdot MPa$ $\alpha := 1$

Estinate n from R6

$N := 15$

$$n_E := \left| \begin{array}{l} \text{for } i \in 0..N \\ \quad n_0 \leftarrow 0.01 \\ \quad n_{i+1} \leftarrow \dfrac{\ln\left(\dfrac{1}{0.002 \cdot n_i}\right) - 1}{\ln\left(\dfrac{\sigma_u}{\sigma_y}\right)} \\ n \end{array} \right.$$

$n_E =$

	0
0	0.01
1	22.776
2	4.845
3	8.435

$n := n_{E_{15}}$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3. Select FAD

Option 2 $$K_r(L_r) = \left(\frac{E \cdot \varepsilon_{ref}}{L_r \cdot \sigma_y} + \frac{L_r^3 \cdot \sigma_y}{2 \cdot E \cdot \varepsilon_{ref}} \right)^{\frac{-1}{2}}$$

$$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2} \qquad \sigma_{flow} = 341.5\,MPa$$

Ramberg-Osgood Equation $$\frac{\varepsilon_{ref}}{\varepsilon_0} = \frac{\sigma_{ref}}{\sigma_0} + \alpha \cdot \left(\frac{\sigma_{ref}}{\sigma_0} \right)^n$$

$$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2} \qquad \sigma_{ref} = L_r \cdot \sigma_{flow}$$

$$\sigma_0 := \sigma_y \qquad \varepsilon_0 := \frac{\sigma_0}{E}$$

$$\varepsilon_{ref} = \varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_{flow}}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_{flow}}{\sigma_0} \right)^n \right]$$

$$K_{rOption2}(L_r) := \left[\frac{E \cdot \left[\varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right] \right]}{L_r \cdot \sigma_y} + \frac{L_r^3 \cdot \sigma_y}{2 \cdot E \cdot \left[\varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right] \right]} \right]^{\frac{-1}{2}}$$

Cut-off $$L_{r_max} := \frac{\sigma_{flow}}{\sigma_y} \qquad L_{r_max} = 1.27$$

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

Crack Depth, a $a := 20mm$

5. Select category of analysis category 2

6. Define fracture toughness $K_{0.2}$ and K_g

Material name : SA106Gr. C

$$J = C_1(\Delta a)^{C_2} \qquad C_1 = 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c} \right)^{2.29} \qquad C_2 := 0.38$$

$s_u := \sigma_u$ $s_y := \sigma_y$

$\varepsilon_c := 0.82$ $E = 200.291\,GPa$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$k := 1mm$

$\Delta a := 0.2$

$S_u := \frac{s_u}{E}$ $\quad S_u = 2.067 \times 10^{-3}$

$C_1 := 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c}\right)^{2.29}$ $\quad C_1 = 22.153\,N\frac{N}{mm}$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $\quad J(\Delta a) = 12.018\,N\frac{N}{mm}$

$K_{0.2} := \sqrt{\frac{(J(\Delta a) \cdot E)}{\left(1 - 0.3^2\right)}}$ $\quad K_{0.2} = 199.191\,MPa\cdot\sqrt{m}$

$\Delta a_g := 2$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $\quad J(\Delta a_g) = 28.829\,N\frac{N}{mm}$

$K_g := \sqrt{\frac{(J(\Delta a_g) \cdot E)}{\left(1 - 0.3^2\right)}}$ $\quad K_g = 308.51\,MPa\cdot\sqrt{m}$

7. Define crack size a_0

Assume $a_0 := a$ $\quad a_0 = 20\,mm$

$\Delta a_g := 2mm$

8. Calculate L_r $\quad L_r = P/P_L$

Original crack size a_0

Limit load from Zahoor [20]

ombined Axial Load and Bending Moment R/t >= 10 and σt/σf < 0.25

Axial tension ,P is equal zero $P := 0MN$ $\quad \frac{R}{t} = 10.026$

$\theta := \pi \cdot \frac{c}{4R_i}$ $\quad \theta = 0.072$

$\sigma_t := \frac{P}{2\pi R \cdot t}$ $\quad \sigma_t = 0.000\,Pa$ $\quad \frac{\sigma_t}{\sigma_{flow}} = 0$

$\beta 1 := 0.5\pi \cdot \left[1 - \left(\frac{\theta}{\pi}\right) \cdot \left(\frac{a}{t}\right) - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right]$ $\quad \beta 1 = 1.558$

$\beta 2 := \pi \cdot \frac{\left[1 - \frac{a}{t} - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right]}{\left(2 - \frac{a}{t}\right)}$ $\quad \beta 2 = 1.237$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$M_L := \begin{cases} \left[2\cdot\sigma_{flow}\cdot R^2\cdot t\cdot\left[2\cdot\sin(\beta 1) - \left(\frac{a}{t}\right)\cdot\sin(\theta)\right]\right] & \text{if } \frac{R}{t} \geq 10 \wedge \frac{\sigma_t}{\sigma_{flow}} < 0.25 \\ \left[2\cdot\sigma_{flow}\cdot R^2\cdot t\cdot\left(2 - \frac{a}{t}\right)\cdot\sin(\beta 2)\right] & \text{if } (\theta + \beta 1) > \pi \\ 0 & \text{otherwise} \end{cases}$$

$M_L = 25.106\,\text{MN}\cdot\text{m}$

$$L_R := \frac{M_{op}}{M_L} \qquad L_R = 1.155$$

Ductile crack growth a_g

$$\theta := \pi\cdot\frac{c}{4R_i} \qquad \theta = 0.072$$

$$\sigma_t := \frac{P}{2\pi R\cdot t} \qquad \sigma_t = 0.000\,\text{Pa}$$

$$\beta 1 := 0.5\pi\cdot\left[1 - \left(\frac{\theta}{\pi}\right)\cdot\left(\frac{a + \Delta a_g}{t}\right) - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right] \qquad \beta 1 = 1.557$$

$$\beta 2 := \pi\cdot\frac{\left[1 - \frac{a + \Delta a_g}{t} - \left(\frac{\sigma_t}{\sigma_{flow}}\right)\right]}{\left(2 - \frac{a + \Delta a_g}{t}\right)} \qquad \beta 2 = 1.237$$

$$M_L := \begin{cases} \left[2\cdot\sigma_{flow}\cdot R^2\cdot t\cdot\left[2\cdot\sin(\beta 1) - \left(\frac{a + \Delta a_g}{t}\right)\cdot\sin(\theta)\right]\right] & \text{if } \frac{R}{t} \geq 10 \wedge \frac{\sigma_t}{\sigma_{flow}} < 0.25 \\ \left[2\cdot\sigma_{flow}\cdot R^2\cdot t\cdot\left(2 - \frac{a + \Delta a_g}{t}\right)\cdot\sin(\beta 2)\right] & \text{if } (\theta + \beta 1) > \pi \\ 0 & \text{otherwise} \end{cases}$$

$M_L = 25.074\,\text{MN}\cdot\text{m}$

$$L_{Rg} := \frac{M_{op}}{M_L} \qquad L_{Rg} = 1.157$$

9. Calculate K_r

Original crack size a_0

Parameter K from Zahoor [20]

The available K-solution of this cracked structure is valid within the following range

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Bending Moment 5 <= R/t <= 160

$2\cdot\frac{c}{a} = 5$ $\quad \frac{R}{t} = 10.026$

2c/a >= 3

$\frac{\theta}{\pi} = 0.023$

θ/π <= 1

$\frac{a}{t} = 0.351$

0.05 <= a/t <= 0.8

$B00 := -0.0002\cdot\left(\frac{R}{t}\right)^2 + 0.0173\cdot\left(\frac{R}{t}\right) + 3.3567$ $\quad B00 = 3.51$

$B01 := -0.00006\cdot\left(\frac{R}{t}\right)^2 + 0.0101\cdot\left(\frac{R}{t}\right) + 1.3153$ $\quad B01 = 1.411$

$B02 := 0.00002\cdot\left(\frac{R}{t}\right)^2 - 0.0019\cdot\left(\frac{R}{t}\right) - 0.0066$ $\quad B02 = -0.024$

$B10 := 0.000006\cdot\left(\frac{R}{t}\right)^2 - 0.0017\cdot\left(\frac{R}{t}\right) - 4.2841$ $\quad B10 = -4.301$

$B11 := 0.00006\cdot\left(\frac{R}{t}\right)^2 - 0.013\cdot\left(\frac{R}{t}\right) - 1.1867$ $\quad B11 = -1.311$

$B12 := -0.000002\cdot\left(\frac{R}{t}\right)^2 + 0.0004\cdot\left(\frac{R}{t}\right) - 0.0741$ $\quad B12 = -0.07$

$B20 := -0.0264\ln\left(\frac{R}{t}\right) + 1.7928$ $\quad B20 = 1.732$

$B21 := 0.147\ln\left(\frac{R}{t}\right) + 0.7238$ $\quad B21 = 1.063$

$B22 := 0.0058\ln\left(\frac{R}{t}\right) + 0.2269$ $\quad B22 = 0.24$

$A0 := B00\cdot\left(\frac{a}{t}\right)^2 + B01\cdot\left(\frac{a}{t}\right) + B02$ $\quad A0 = 0.903$

$A1 := B10\cdot\left(\frac{a}{t}\right)^2 + B11\cdot\left(\frac{a}{t}\right) + B12$ $\quad A1 = -1.06$

$A2 := B20\cdot\left(\frac{a}{t}\right)^2 + B21\cdot\left(\frac{a}{t}\right) + B22$ $\quad A2 = 0.826$

$Fb := A0\cdot\left(\frac{a}{c}\right)^2 + A1\cdot\left(\frac{a}{c}\right) + A2$ $\quad Fb = 0.547$

$\sigma b := \frac{M_{op}}{\left(\pi\cdot R^2\cdot t\right)}$ $\quad \sigma b = 495.839\ \text{MPa}$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$K_I := \begin{cases} \sigma b \cdot (\pi \cdot t)^{0.5} \cdot Fb & \text{if } 5 \le \frac{R}{t} \le 160 \wedge 0.05 \le \frac{a}{t} \le 0.8 \wedge 2 \cdot \frac{c}{a} \ge 3 \wedge \frac{\theta}{\pi} \le 1 \\ 0 & \text{otherwise} \end{cases}$$

$K_I = 114775167\ Pa \cdot \sqrt{m}$ $\qquad$ $K_I = 114.775\ MPa \cdot \sqrt{m}$

$$K_R := \frac{K_I}{K_{0.2}} \qquad K_R = 0.576$$

Ductile crack growth a_g

Bending Moment $5 <= R/t <= 160$

$2c/a >= 3$

$\theta/\pi <= 1$

$0.05 <= a/t <= 0.8$

$$2 \cdot \frac{c}{a + \Delta a_g} = 4.545 \qquad \frac{R}{t} = 10.026$$

$$\frac{\theta}{\pi} = 0.023$$

$$\frac{a + \Delta a_g}{t} = 0.386$$

$$B00 := -0.0002 \cdot \left(\frac{R}{t}\right)^2 + 0.0173 \cdot \left(\frac{R}{t}\right) + 3.3567 \qquad B00 = 3.51$$

$$B01 := -0.00006 \cdot \left(\frac{R}{t}\right)^2 + 0.0101 \cdot \left(\frac{R}{t}\right) + 1.3153 \qquad B01 = 1.411$$

$$B02 := 0.00002 \cdot \left(\frac{R}{t}\right)^2 - 0.0019 \cdot \left(\frac{R}{t}\right) - 0.0066 \qquad B02 = -0.024$$

$$B10 := 0.000006 \cdot \left(\frac{R}{t}\right)^2 - 0.0017 \cdot \left(\frac{R}{t}\right) - 4.2841 \qquad B10 = -4.301$$

$$B11 := 0.00006 \cdot \left(\frac{R}{t}\right)^2 - 0.013 \cdot \left(\frac{R}{t}\right) - 1.1867 \qquad B11 = -1.311$$

$$B12 := -0.000002 \cdot \left(\frac{R}{t}\right)^2 + 0.0004 \cdot \left(\frac{R}{t}\right) - 0.0741 \qquad B12 = -0.07$$

$$B20 := -0.0264 \ln\left(\frac{R}{t}\right) + 1.7928 \qquad B20 = 1.732$$

$$B21 := 0.147 \ln\left(\frac{R}{t}\right) + 0.7238 \qquad B21 = 1.063$$

$$B22 := 0.0058 \ln\left(\frac{R}{t}\right) + 0.2269 \qquad B22 = 0.24$$

$$A0 := B00 \cdot \left(\frac{a + \Delta a_g}{t}\right)^2 + B01 \cdot \left(\frac{a + \Delta a_g}{t}\right) + B02 \qquad A0 = 1.044$$

$$A1 := B10 \cdot \left(\frac{a + \Delta a_g}{t}\right)^2 + B11 \cdot \left(\frac{a + \Delta a_g}{t}\right) + B12 \qquad A1 = -1.217$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$A2 := B20\cdot\left(\frac{a + \Delta a_g}{t}\right)^2 + B21\cdot\left(\frac{a + \Delta a_g}{t}\right) + B22 \qquad A2 = 0.908$$

$$Fb := A0\cdot\left(\frac{a + \Delta a_g}{c}\right)^2 + A1\cdot\left(\frac{a + \Delta a_g}{c}\right) + A2 \qquad Fb = 0.575$$

$$\sigma b := \frac{M_{op}}{\left(\pi\cdot R^2\cdot t\right)} \qquad \sigma b = 495.839\,\text{MPa}$$

$$K_I := \begin{cases} \sigma b\cdot(\pi\cdot t)^{0.5}\cdot Fb & \text{if } 5 \le \frac{R}{t} \le 160 \wedge 0.05 \le \frac{a + \Delta a_g}{t} \le 0.8 \wedge 2\cdot\frac{c}{a} \ge 3 \wedge \frac{\theta}{\pi} \le \\ 0 & \text{otherwise} \end{cases}$$

$$K_I = 120653929\,\text{Pa}\cdot\sqrt{\text{m}} \qquad K_I = 120.654\,\text{MPa}\cdot\sqrt{\text{m}}$$

$$K_{Rg} := \frac{K_I}{K_g} \qquad K_{Rg} = 0.391$$

10. Plot assessment piont (L_r,K_r)

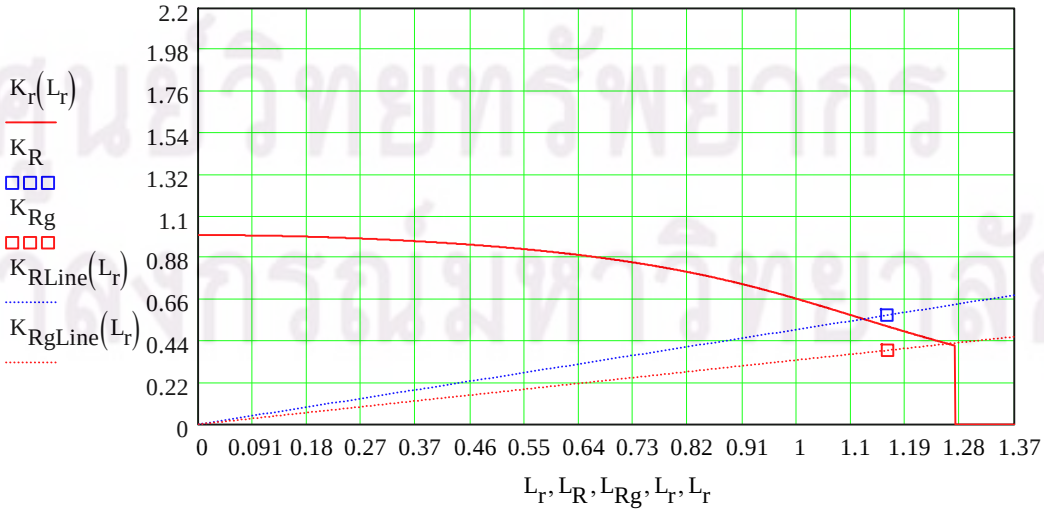
$$K_r(L_r) := \begin{cases} K_{rOption2}(L_r) & \text{if } 0 \le L_r \le L_{r_max} \\ 0 & \text{otherwise} \end{cases} \quad 0, 0.001 .. 1.5$$

For Original crack size a_0

$$K_{RLine}(L_r) := \frac{K_R}{L_R}\cdot(L_r - L_R) + K_R$$

For Ductile crack growth a_g

$$K_{RgLine}(L_r) := \frac{K_{Rg}}{L_{Rg}}\cdot(L_r - L_{Rg}) + K_{Rg}$$



11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

For Original crack size a_0

$$\left[\frac{K_R}{L_R}\cdot\left(L_r - L_R\right) + K_R\right] = K_{rOption2}\left(L_r\right)$$

$$f\left(L_r\right) := \left[\frac{K_R}{L_R}\cdot\left(L_r - L_R\right) + K_R\right] - K_{rOption2}\left(L_r\right) \qquad L_r := 1$$

$L_{rC} := \text{root}\left(f\left(L_r\right), L_r\right)$ $\qquad L_{rC} = 1.116$

$K_{rC} := K_{rOption2}\left(L_{rC}\right)$ $\qquad K_{rC} = 0.557$

$K_{rmax} := K_{rOption2}\left(L_{r_max}\right)$ $\qquad K_{rmax} = 0.416$

L_r Critical

$$L_{rC1} := \begin{cases} L_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R}{L_R} \\ L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R}{L_R} \end{cases}$$

$L_{rC1} = 1.116$

K_r Critical

$$K_{rC1} := \begin{cases} K_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R}{L_R} \\ \frac{K_R}{L_R}\cdot L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R}{L_R} \end{cases}$$

$K_{rC1} = 0.557$

$d := \sqrt{K_R^{\,2} + L_R^{\,2}}$ $\qquad dC := \sqrt{K_{rC}^{\,2} + L_{rC}^{\,2}}$

$FL := \frac{dC}{d}$ $\qquad FL = 0.966$

For Ductile crack growth a_g

$$f\left(L_r\right) := \left[\frac{K_{Rg}}{L_{Rg}}\cdot\left(L_r - L_{Rg}\right) + K_{Rg}\right] - K_{rOption2}\left(L_r\right) \qquad L_r := 1$$

$L_{rC} := \text{root}\left(f\left(L_r\right), L_r\right)$ $\qquad L_{rC} = 1.258$

$K_{rC} := K_{rOption2}\left(L_{rC}\right)$ $\qquad K_{rC} = 0.425$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

L_r Critical $\quad L_{rC1} := \begin{cases} L_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_{Rg}}{L_{Rg}} \\ L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_{Rg}}{L_{Rg}} \end{cases}$

$L_{rC1} = 1.258$

K_r Critical $\quad K_{rC1} := \begin{cases} K_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_{Rg}}{L_{Rg}} \\ \dfrac{K_{Rg}}{L_{Rg}} \cdot L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_{Rg}}{L_{Rg}} \end{cases}$

$K_{rC1} = 0.425$

$d := \sqrt{K_{Rg}^{\,2} + L_{Rg}^{\,2}}$

$dC := \sqrt{K_{rC1}^{\,2} + L_{rC1}^{\,2}} \qquad FL := \dfrac{dC}{d} \qquad FL = 1.088$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.2.3 ท่อมีรอยร้าวผิวรูปครึ่งวงรีวางตัวภายในตามแนวแกนรับภาระความดันภายใน

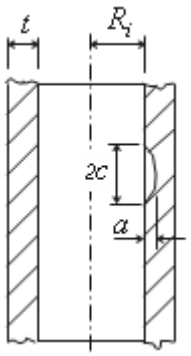
ข.2.3.1 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 1

$GPa := 10^9 \cdot Pa$ $MPa := 10^6 \cdot Pa$ $MN := 10^6 newton$

R6 Option 1 & Level 1

Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)



Structure's dimensions

Outer radius, RO	$R_o := 600mm$	
Inner radius, Ri	$R_i := 543mm$	
Thickness, t	$t := R_o - R_i$	$t = 57\,mm$
Mean radius, R	$R := R_i + \left(\frac{t}{2}\right)$	$R = 571.5\,mm$

1. Categorise load & stress

Primary load : Internal Pressure, P_{op} $P_{op} := 20MPa$

Secondary load : NONE

Operating temperature $T := 121$ {Celcius}

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

Material name $E := 207200MPa - 57.1(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269 \cdot MPa$

$\sigma_u := 414 \cdot MPa$

3. Select FAD

Option 1 $K_r(L_r) = \left(1 - 0.14 \cdot L_r^2\right) \cdot \left(0.3 + 0.7 \exp\left(-0.65 \cdot L_r^6\right)\right)$

$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2}$ $\sigma_{flow} = 341.5\,MPa$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Cut-off $L_{r_max} := \frac{\sigma_{flow}}{\sigma_y}$ $L_{r_max} = 1.27$

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

Crack Depth, a $a := 20mm$

5. Select category of analysis category 1

6. Define fracture toughness K_I ,K_{IC}, or $K_{0.2}$

$T_{ref} := 121$

$K_{IC} := \left[36.5 + 3.084\exp\left[0.036\cdot\left(T - T_{ref} + 56\right)\right]\right]\cdot MPa\cdot\sqrt{m}$ From API 579 [5]

$K_{IC} = 59.655387\,MPa\cdot\sqrt{m}$

7. Define crack size a_0

$a_0 := a$ $a_0 = 20\,mm$

8. Calculate L_r $L_r = P/P_L$

Limit loadfrom Zahoor [20] $\sigma_{ref} := \sigma_{flow}$

Internal Pressure Where the flaw is assumed to be of rectangular shape

where $Po := \sigma_{flow}\cdot\left(\frac{t}{R_o}\right)$ $Po = 32.442\,MPa$

$x := \frac{a}{t}$ $x = 0.351$

$M := \left[1 + 1.61\cdot\frac{c^2}{(R\cdot t)}\right]^{0.5}$ $M = 1.06$

$P_L := Po\cdot\left[\frac{(1 - x)}{\left(1 - \frac{x}{M}\right)}\right]$ $P_L = 31.480\,MPa$

$L_R(P) := \frac{P}{P_L}$ $L_R(P_{op}) = 0.635$

9.Calculate K_r

Parameter K from Zahoor [20] $K_I = \sigma_h\cdot(\pi\cdot t)^{0.5}\cdot F_1$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

The available K-solution of this cracked structure is valid within the following range

$$0.05 \le \frac{a}{t} \le 0.85 \qquad 1 \le \frac{R_i}{t} \le 10 \qquad 0.1 \le \frac{a}{c} \le 1 \qquad \alpha \ge 0.2$$

For our problem $\frac{a}{t} = 0.351$ $\frac{R_i}{t} = 9.526$ $\frac{a}{c} = 0.4$

where $\alpha := \frac{\left(\frac{a}{t}\right)}{\left(\frac{a}{c}\right)^{0.58}}$ $\alpha = 0.597$

$$\sigma_h(P) := P \cdot \frac{\left(R_o^{\,2} + R_i^{\,2}\right)}{\left(R_o^{\,2} - R_i^{\,2}\right)} \qquad \sigma_h(P_{op}) = 201.025\ \text{MPa}$$

$$F_1 := 0.25 + \frac{\left(0.4759 \cdot \alpha + 0.1262 \cdot \alpha^2\right)}{\left[0.102 \cdot \left(\frac{R_i}{t}\right) - 0.02\right]^{0.1}} \qquad F_1 = 0.581$$

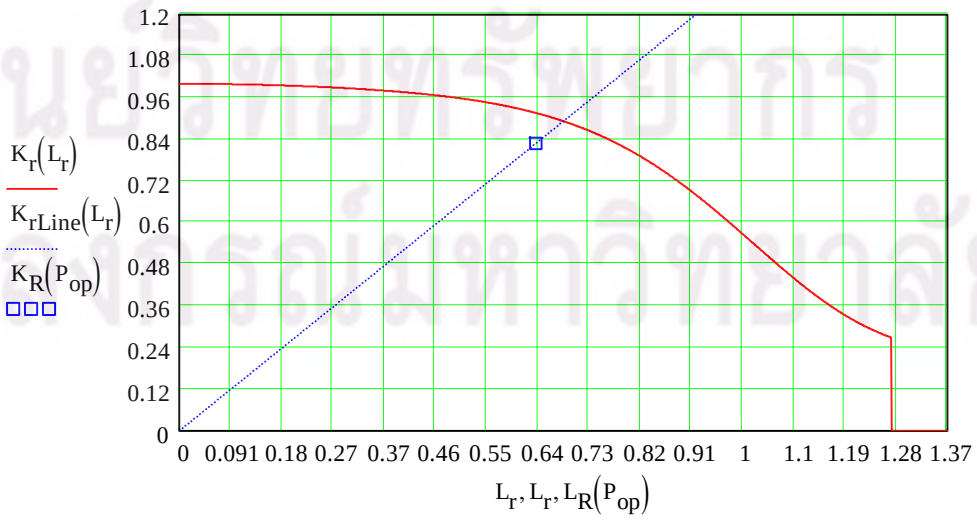
$$K_I(P) := \sigma_h(P) \cdot (\pi \cdot t)^{0.5} \cdot F_1 \qquad K_I(P_{op}) = 49.4\ \text{MPa}\cdot\sqrt{\text{m}}$$

$$K_R(P) := \frac{K_I(P)}{K_{IC}} \qquad K_R(P_{op}) = 0.828$$

10. Plot assessment piont (L_r, K_r) $L_r := 0, 0.001 .. 1.5$

$$K_r(L_r) := \begin{cases} \left[\left(1 - 0.14 \cdot L_r^{\,2}\right) \cdot \left(0.3 + 0.7 \exp\left(-0.65 \cdot L_r^{\,6}\right)\right)\right] & \text{if } 0 \le L_r \le L_{r_max} \\ 0 & \text{otherwise} \end{cases}$$

$$K_{rLine}(L_r) := \frac{K_R(P_{op})}{L_R(P_{op})} \cdot \left(L_r - L_R(P_{op})\right) + K_R(P_{op})$$



11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

$$\left[\frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] = \left(1 - 0.14\cdot L_r^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_r^{\,6}}\right)$$

$$f(L_r) := \left[\frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] - \left(1 - 0.14\cdot L_r^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_r^{\,6}}\right)$$

$L_r := 1$

$L_{rC} := \text{root}(f(L_r), L_r)$ $\quad L_{rC} = 0.685$

$K_{rC} := \left(1 - 0.14\cdot L_{rC}^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_{rC}^{\,6}}\right)$ $\quad K_{rC} = 0.892$

$K_{rmax} := \left(1 - 0.14\cdot L_{r_max}^{\,2}\right)\cdot\left(0.3 + 0.7\cdot e^{-0.65\cdot L_{r_max}^{\,6}}\right)$ $\quad K_{rmax} = 0.268$

L_r Critical $\quad L_{rC1} := \begin{cases} L_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R(P_{op})}{L_R(P_{op})} \\ L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$L_{rC1} = 0.685$

K_r Critical $\quad K_{rC1} := \begin{cases} K_{rC} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} < \dfrac{K_R(P_{op})}{L_R(P_{op})} \\ \dfrac{K_R(P_{op})}{L_R(P_{op})}\cdot L_{r_max} & \text{if } \dfrac{K_{rmax}}{L_{r_max}} > \dfrac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$K_{rC1} = 0.892$

$d := \sqrt{K_R(P_{op})^2 + L_R(P_{op})^2}$

$dC := \sqrt{K_{rC1}^{\,2} + L_{rC1}^{\,2}}$

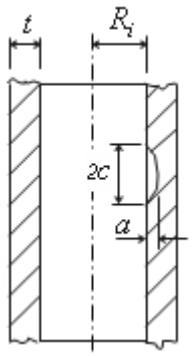
$FL := \dfrac{dC}{d}$ $\quad FL = 1.077$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.2.3.2 FAD ชนิดที่ 1 คู่กับการประเมินระดับที่ 2

$GPa := 10^9 \cdot Pa$ $MPa := 10^6 \cdot Pa$ $MN := 10^6 newton$

R6 Option 1 & Level 2



Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)

Structure's dimensions

Outer radius, RO $R_o := 600mm$

Inner radius, Ri $R_i := 543mm$

Thickness, t $t := R_o - R_i$ $t = 57\,mm$

Mean radius, R $R := R_i + \left(\frac{t}{2}\right)$ $R = 571.5\,mm$

1. Categorise load & stress

Primary load : Internal Pressure, P_{op} $P_{op} := 20MPa$

Secondary load : NONE

Operating temperature $T := 121$ {Celcius}

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

Material name $E := 207200MPa - 57.1\,(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269 \cdot MPa$

$\sigma_u := 414 \cdot MPa$

3. Select FAD

Option 1 $K_r(L_r) = \left(1 - 0.14 \cdot L_r^{\,2}\right) \cdot \left(0.3 + 0.7 \exp\left(-0.65 \cdot L_r^{\,6}\right)\right)$

$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2}$ $\sigma_{flow} = 341.5\,MPa$

Cut-off $L_{r_max} := \frac{\sigma_{flow}}{\sigma_y}$ $L_{r_max} = 1.27$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

Crack Depth, a $a := 20mm$

5. Select category of analysis category 2

6. Define fracture toughness $K_{0.2}$ and K_g

Material name : SA106Gr. C

$J = C_1(\Delta a)^{C_2}$ $C_1 = 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c}\right)^{2.29}$ $C_2 := 0.38$

$s_u := \sigma_u$ $s_y := \sigma_y$

$\varepsilon_c := 0.82$ $E = 200.291\,GPa$

$k := 1mm$

$\Delta a := 0.2$

$S_u := \frac{s_u}{E}$ $S_u = 2.067 \times 10^{-3}$

$C_1 := 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c}\right)^{2.29}$ $C_1 = 332.298\,\frac{N}{mm}$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $J(\Delta a) = 180.268\,\frac{N}{mm}$

$K_{0.2} := \sqrt{\frac{(J(\Delta a) \cdot E)}{\left(1 - 0.3^2\right)}}$ $K_{0.2} = 199.191\,MPa \cdot \sqrt{m}$

$\Delta a_g := 2$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $J(\Delta a_g) = 432.433\,\frac{N}{mm}$

$K_g := \sqrt{\frac{\left(J(\Delta a_g) \cdot E\right)}{\left(1 - 0.3^2\right)}}$ $K_g = 308.51\,MPa \cdot \sqrt{m}$

7. Define crack size a_0 and a_g

$a_0 := a$ $a_0 = 20\,mm$

Assume $\Delta a_g := 2mm$

8. Calculate L_r $L_r = P/P_L$

Original crack size a_0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Limit loadfrom Zahoor [20] $\sigma_{ref} := \sigma_{flow}$

Internal Pressure Where the flaw is assumed to be of rectangular shape

where $Po := \sigma_{flow} \cdot \left(\frac{t}{R_o} \right)$ $Po = 32.442\,MPa$

$x := \frac{a}{t}$ $x = 0.351$

$M := \left[1 + 1.61 \cdot \frac{c^2}{(R \cdot t)} \right]^{0.5}$ $M = 1.06$

$P_L := Po \cdot \left[\frac{(1 - x)}{\left(1 - \frac{x}{M} \right)} \right]$ $P_L = 31.480\,MPa$

$L_R(P) := \frac{P}{P_L}$ $L_R(P_{op}) = 0.635$

Ductile crack growth a_g

Limit load from Zahoor $\sigma_{ref} := \sigma_{flow}$

Internal Pressure Where the flaw is assumed to be of rectangular shape

where $Po := \sigma_{flow} \cdot \left(\frac{t}{R_o} \right)$ $Po = 32.442\,MPa$

$x := \frac{a + \Delta a_g}{t}$ $x = 0.386$

$M := \left[1 + 1.61 \cdot \frac{c^2}{(R \cdot t)} \right]^{0.5}$ $M = 1.06$

$P_L := Po \cdot \left[\frac{(1 - x)}{\left(1 - \frac{x}{M} \right)} \right]$ $P_L = 31.328\,MPa$

$L_{Rg}(P) := \frac{P}{P_L}$ $L_{Rg}(P_{op}) = 0.638$

9.Calculate K_r

Parameter K from Zahoor [20] $K_I = \sigma_h \cdot (\pi \cdot t)^{0.5} \cdot F_1$

The available K-solution of this cracked structure is valid within the following range

$0.05 \le \frac{a}{t} \le 0.85$ $1 \le \frac{R_i}{t} \le 10$ $0.1 \le \frac{a}{c} \le 1$ $\alpha \ge 0.2$

For our problem $\frac{a}{t} = 0.351$ $\frac{R_i}{t} = 9.526$ $\frac{a}{c} = 0.4$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$\frac{a + \Delta a_g}{t} = 0.386 \qquad \frac{a + \Delta a_g}{c} = 0.44$$

Original crack size a_0

where $\alpha := \dfrac{\left(\frac{a}{t}\right)}{\left(\frac{a}{c}\right)^{0.58}}$ $\alpha = 0.597$

$$\sigma_h(P) := P \cdot \frac{\left(R_o^{\,2} + R_i^{\,2}\right)}{\left(R_o^{\,2} - R_i^{\,2}\right)} \qquad \sigma_h(P_{op}) = 201.025\ \text{MPa}$$

$$F_1 := 0.25 + \frac{\left(0.4759 \cdot \alpha + 0.1262 \cdot \alpha^2\right)}{\left[0.102 \cdot \left(\frac{R_i}{t}\right) - 0.02\right]^{0.1}} \qquad F_1 = 0.581$$

$$K_I(P) := \sigma_h(P) \cdot (\pi \cdot t)^{0.5} \cdot F_1 \qquad K_I(P_{op}) = 49.4\ \text{MPa} \cdot \sqrt{\text{m}}$$

$$K_R(P) := \frac{K_I(P)}{K_{0.2}} \qquad K_R(P_{op}) = 0.248$$

Ductile crack growth a_g

Parameter K from Zahoor $K_I = \sigma_h \cdot (\pi \cdot t)^{0.5} \cdot F_1$

where $\alpha := \dfrac{\left(\frac{a + \Delta a_g}{t}\right)}{\left(\frac{a + \Delta a_g}{c}\right)^{0.58}}$ $\alpha = 0.621$

$$\sigma_h(P) := P \cdot \frac{\left(R_o^{\,2} + R_i^{\,2}\right)}{\left(R_o^{\,2} - R_i^{\,2}\right)} \qquad \sigma_h(P_{op}) = 201.025\ \text{MPa}$$

$$F_1 := 0.25 + \frac{\left(0.4759 \cdot \alpha + 0.1262 \cdot \alpha^2\right)}{\left[0.102 \cdot \left(\frac{R_i}{t}\right) - 0.02\right]^{0.1}} \qquad F_1 = 0.596$$

$$K_I(P) := \sigma_h(P) \cdot (\pi \cdot t)^{0.5} \cdot F_1 \qquad K_I(P_{op}) = 50.712\ \text{MPa} \cdot \sqrt{\text{m}}$$

$$K_{Rg}(P) := \frac{K_I(P)}{K_g} \qquad K_{Rg}(P_{op}) = 0.164$$

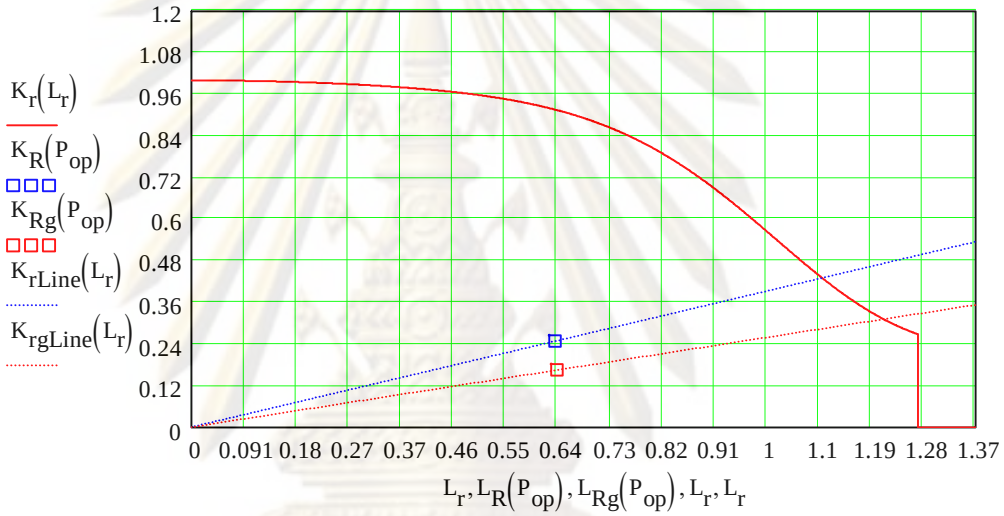
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

10. Plot assessment piont (L_r,K_r) $L_r := 0, 0.001 .. 1.5$

$$K_r(L_r) := \begin{cases} \left[\left(1 - 0.14 \cdot L_r^{\,2}\right) \cdot \left(0.3 + 0.7 \exp\left(-0.65 \cdot L_r^{\,6}\right)\right)\right] & \text{if } 0 \le L_r \le L_{r_max} \\ 0 & \text{otherwise} \end{cases}$$

$$K_{rLine}(L_r) := \frac{K_R(P_{op})}{L_R(P_{op})} \cdot \left(L_r - L_R(P_{op})\right) + K_R(P_{op})$$

$$K_{rgLine}(L_r) := \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \cdot \left(L_r - L_{Rg}(P_{op})\right) + K_{Rg}(P_{op})$$



11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

$$\left[\frac{K_R(P_{op})}{L_R(P_{op})} \cdot \left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] = \left(1 - 0.14 \cdot L_r^{\,2}\right) \cdot \left(0.3 + 0.7 \cdot e^{-0.65 \cdot L_r^{\,6}}\right)$$

Original crack size a_0

$$f(L_r) := \left[\frac{K_R(P_{op})}{L_R(P_{op})} \cdot \left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] - \left(1 - 0.14 \cdot L_r^{\,2}\right) \cdot \left(0.3 + 0.7 \cdot e^{-0.65 \cdot L_r^{\,6}}\right)$$

$L_r := 1$

$L_{rC} := \text{root}(f(L_r), L_r)$ $\quad L_{rC} = 1.102$

$K_{rC} := \left(1 - 0.14 \cdot L_{rC}^{\,2}\right) \cdot \left(0.3 + 0.7 \cdot e^{-0.65 \cdot L_{rC}^{\,6}}\right)$ $\quad K_{rC} = 0.43$

$K_{rmax} := \left(1 - 0.14 \cdot L_{r_max}^{\,2}\right) \cdot \left(0.3 + 0.7 \cdot e^{-0.65 \cdot L_{r_max}^{\,6}}\right)$ $\quad K_{rmax} = 0.268$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

L_r Critical $L_{rC1} := \begin{cases} L_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R(P_{op})}{L_R(P_{op})} \\ L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$L_{rC1} = 1.102$

K_r Critical $K_{rC1} := \begin{cases} K_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R(P_{op})}{L_R(P_{op})} \\ \frac{K_R(P_{op})}{L_R(P_{op})} \cdot L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$K_{rC1} = 0.43$

$d := \sqrt{K_R(P_{op})^2 + L_R(P_{op})^2}$ $dC := \sqrt{K_{rC1}^2 + L_{rC1}^2}$

$FL := \frac{dC}{d}$ $FL = 1.735$

Ductile crack growth a_g

$f(L_r) := \left[\frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \cdot (L_r - L_{Rg}(P_{op})) + K_{Rg}(P_{op})\right] - \left(1 - 0.14 \cdot L_r^2\right) \cdot \left(0.3 + 0.7 \cdot e^{-0.65 \cdot L_r^6}\right)$

$L_r := 1$

$L_{rC} := \text{root}(f(L_r), L_r)$ $L_{rC} = 1.209$

$K_{rC} := \left(1 - 0.14 \cdot L_{rC}^2\right) \cdot \left(0.3 + 0.7 \cdot e^{-0.65 \cdot L_{rC}^6}\right)$ $K_{rC} = 0.312$

L_r Critical $L_{rC1} := \begin{cases} L_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \\ L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \end{cases}$

$L_{rC1} = 1.209$

K_r Critical $K_{rC1} := \begin{cases} K_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \\ \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \cdot L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \end{cases}$

$K_{rC1} = 0.312$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$d := \sqrt{K_{Rg}\left(P_{op}\right)^2 + L_{Rg}\left(P_{op}\right)^2} \qquad dC := \sqrt{K_{rC}{}^2 + L_{rC}{}^2}$$

$$FL := \frac{dC}{d} \qquad FL = 1.894$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

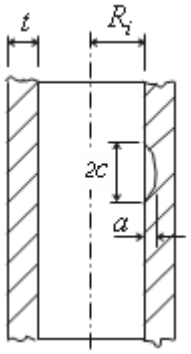
ข.2.3.3 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 1

$GPa := 10^9 \cdot Pa$ $MPa := 10^6 \cdot Pa$ $MN := 10^6 newton$

R6 Option 2 & Level 1

Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)



Structure's dimensions

Outer radius, RO	$R_o := 600mm$	
Inner radius, Ri	$R_i := 543mm$	
Thickness, t	$t := R_o - R_i$	$t = 57\,mm$
Mean radius, R	$R := R_i + \left(\frac{t}{2}\right)$	$R = 571.5\,mm$

1. Categorise load & stress

Primary load : Internal Pressure, P_{op} $P_{op} := 20MPa$

Secondary load : NONE

Operating temperature $T := 121$ {Celcius}

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

Material name $E := 207200MPa - 57.1(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269 \cdot MPa$

$\sigma_u := 414 \cdot MPa$ $\alpha := 1$

Estinate n from R6

$N := 15$

$$n_E := \left| \begin{array}{l} \text{for } i \in 0..N \\ \quad \left| \begin{array}{l} n_0 \leftarrow 0.01 \\ n_{i+1} \leftarrow \dfrac{\ln\left(\dfrac{1}{0.002 \cdot n_i}\right) - 1}{\ln\left(\dfrac{\sigma_u}{\sigma_y}\right)} \end{array} \right. \\ n \end{array} \right.$$

$n_E =$

	0
0	0.01
1	22.776
2	4.845
3	8.435

$n := n_{E_{15}}$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3. Select FAD

Option 2 $K_r(L_r) = \left(\frac{E \cdot \varepsilon_{ref}}{L_r \cdot \sigma_y} + \frac{L_r^3 \cdot \sigma_y}{2 \cdot E \cdot \varepsilon_{ref}} \right)^{\frac{-1}{2}}$

Ramberg-Osgood Equation $\frac{\varepsilon_{ref}}{\varepsilon_0} = \frac{\sigma_{ref}}{\sigma_0} + \alpha \cdot \left(\frac{\sigma_{ref}}{\sigma_0} \right)^n$

$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2}$ $\sigma_{ref} = L_r \cdot \sigma_y$

$\sigma_0 := \sigma_y$ $\varepsilon_0 := \frac{\sigma_0}{E}$

$\varepsilon_{ref} = \varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right]$

$$K_{rOption2}(L_r) := \left[\frac{E \cdot \left[\varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right] \right]}{L_r \cdot \sigma_y} + \frac{L_r^3 \cdot \sigma_y}{2 \cdot E \cdot \left[\varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right] \right]} \right]^{\frac{-1}{2}}$$

Cut-off $L_{r_max} := \frac{\sigma_{flow}}{\sigma_y}$ $L_{r_max} = 1.27$

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

Crack Depth, a $a := 20mm$

5. Select category of analysis category 1

6. Define fracture toughness K_I ,K_{IC}, or $K_{0.2}$

$T_{ref} := 121$

$K_{IC} := \left[36.5 + 3.084 \exp\left[0.036 \cdot \left(T - T_{ref} + 56 \right) \right] \right] \cdot MPa \cdot \sqrt{m}$ From API 579 [5]

$K_{IC} = 59.655387\,MPa \cdot \sqrt{m}$

7. Define crack size a_0

$a_0 := a$ $a_0 = 20\,mm$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

8. Calculate L_r $L_r = P/P_L$

Limit load from Zahoor [20] $\sigma_{ref} := \sigma_{flow}$

Internal Pressure Where the flaw is assumed to be of rectangular shape

where $Po := \sigma_{flow}\cdot\left(\frac{t}{R_o}\right)$ $Po = 32.442\,MPa$

$x := \frac{a}{t}$ $x = 0.351$

$M := \left[1 + 1.61\cdot\frac{c^2}{(R\cdot t)}\right]^{0.5}$ $M = 1.06$

$P_L := Po\cdot\left[\frac{(1-x)}{\left(1-\frac{x}{M}\right)}\right]$ $P_L = 31.480\,MPa$

$L_R(P) := \frac{P}{P_L}$ $L_R(P_{op}) = 0.635$

9.Calculate K_r

Parameter K from Zahoor [20] $K_I = \sigma_h\cdot(\pi\cdot t)^{0.5}\cdot F_1$

The available K-solution of this cracked structure is valid within the following range

$0.05 \le \frac{a}{t} \le 0.85$ $1 \le \frac{R_i}{t} \le 10$ $0.1 \le \frac{a}{c} \le 1$ $\alpha \ge 0.2$

For our problem $\frac{a}{t} = 0.351$ $\frac{R_i}{t} = 9.526$ $\frac{a}{c} = 0.4$

where $\alpha_1 := \frac{\left(\frac{a}{t}\right)}{\left(\frac{a}{c}\right)^{0.58}}$ $\alpha_1 = 0.597$

$\sigma_h(P) := P\cdot\frac{\left(R_o^2 + R_i^2\right)}{\left(R_o^2 - R_i^2\right)}$ $\sigma_h(P_{op}) = 201.025\,MPa$

$F_1 := 0.25 + \frac{\left(0.4759\cdot\alpha_1 + 0.1262\cdot\alpha_1^2\right)}{\left[0.102\cdot\left(\frac{R_i}{t}\right) - 0.02\right]^{0.1}}$ $F_1 = 0.581$

$K_I(P) := \sigma_h(P)\cdot(\pi\cdot t)^{0.5}\cdot F_1$ $K_I(P_{op}) = 49.4\,MPa\cdot\sqrt{m}$

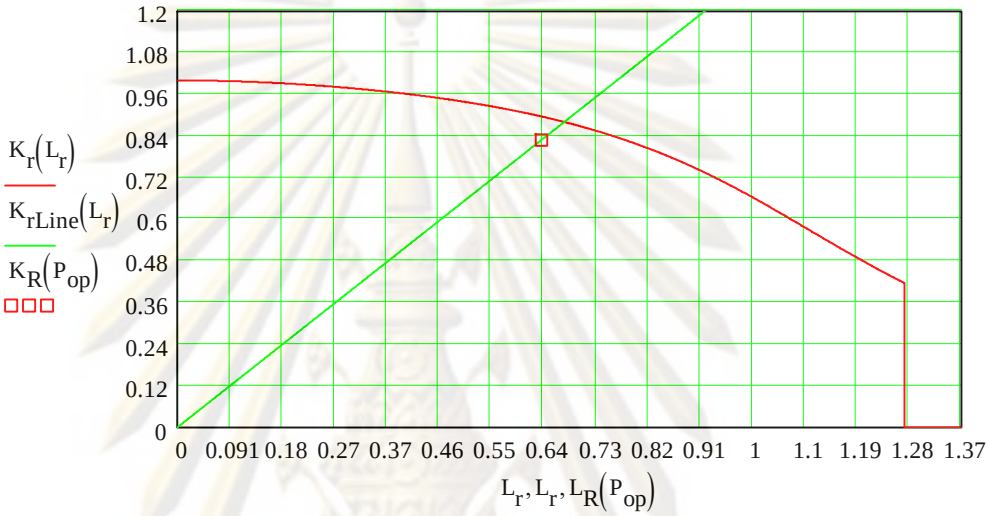
$K_R(P) := \frac{K_I(P)}{K_{IC}}$ $K_R(P_{op}) = 0.828$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

10. Plot assessment piont (L_r, K_r) $L_r := 0, 0.001 .. 1.5$

$$K_r(L_r) := \begin{cases} K_{rOption2}(L_r) & \text{if } 0 \le L_r \le L_{r_max} \\ 0 & \text{otherwise} \end{cases}$$

$$K_{rLine}(L_r) := \frac{K_R(P_{op})}{L_R(P_{op})} \cdot (L_r - L_R(P_{op})) + K_R(P_{op})$$



11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

$$\left[\frac{K_R(P_{op})}{L_R(P_{op})} \cdot (L_r - L_R(P_{op})) + K_R(P_{op})\right] = K_{rOption2}(L_r)$$

$$f(L_r) := \left[\frac{K_R(P_{op})}{L_R(P_{op})} \cdot (L_r - L_R(P_{op})) + K_R(P_{op})\right] - K_{rOption2}(L_r)$$

$L_r := 1$

$L_{rC} := \text{root}(f(L_r), L_r)$ $L_{rC} = 0.675$

$K_{rC} := K_{rOption2}(L_{rC})$ $K_{rC} = 0.88$

$K_{rmax} := K_{rOption2}(L_{r_max})$ $K_{rmax} = 0.416$

L_r Critical $L_{rC1} := \begin{cases} L_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R(P_{op})}{L_R(P_{op})} \\ L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$L_{rC1} = 0.675$

K_r Critical $K_{rC1} := \begin{cases} K_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R(P_{op})}{L_R(P_{op})} \\ \frac{K_R(P_{op})}{L_R(P_{op})} \cdot L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$K_{rC1} = 0.88$

$d := \sqrt{K_R(P_{op})^2 + L_R(P_{op})^2}$ $dC := \sqrt{K_{rC1}^2 + L_{rC1}^2}$

$FL := \frac{dC}{d}$ $FL = 1.063$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

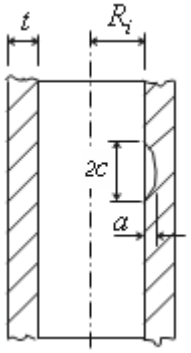
ข.2.3.4 FAD ชนิดที่ 2 คู่กับการประเมินระดับที่ 2

$GPa := 10^9 \cdot Pa$ $MPa := 10^6 \cdot Pa$ $MN := 10^6 newton$

R6 Option 2 & Level 2

Name of cracked structure :

Finite length,semi-elliptical, internal circumferential part-thoughwall flaw (Axial semi-elliptical crack)



Structure's dimensions

Outer radius, RO	$R_o := 600mm$	
Inner radius, Ri	$R_i := 543mm$	
Thickness, t	$t := R_o - R_i$	$t = 57\,mm$
Mean radius, R	$R := R_i + \left(\frac{t}{2}\right)$	$R = 571.5\,mm$

1. Categorise load & stress

Primary load : Internal Pressure, P_{op} $P_{op} := 20MPa$

Secondary load : NONE

Operating temperature $T := 121$ {Celcius}

2. Meterial's tensile properties

Material name : SA106Gr. C [19]

Material name $E := 207200MPa - 57.1\,(T)MPa$ $E = 200.291\,GPa$

$\sigma_y := 269 \cdot MPa$

$\sigma_u := 414 \cdot MPa$ $\alpha := 1$

Estinate n from R6

$N := 15$

$$n_E := \left| \begin{array}{l} \text{for } i \in 0..N \\ \quad n_0 \leftarrow 0.01 \\ \quad n_{i+1} \leftarrow \dfrac{\ln\left(\dfrac{1}{0.002 \cdot n_i}\right) - 1}{\ln\left(\dfrac{\sigma_u}{\sigma_y}\right)} \\ n \end{array} \right.$$

$n_E =$

	0
0	0.01
1	22.776
2	4.845
3	8.435

$n := n_{E_{15}}$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3. Select FAD

Option 2 $K_r(L_r) = \left(\frac{E \cdot \varepsilon_{ref}}{L_r \cdot \sigma_y} + \frac{L_r^3 \cdot \sigma_y}{2 \cdot E \cdot \varepsilon_{ref}} \right)^{\frac{-1}{2}}$

Ramberg-Osgood Equation $\frac{\varepsilon_{ref}}{\varepsilon_0} = \frac{\sigma_{ref}}{\sigma_0} + \alpha \cdot \left(\frac{\sigma_{ref}}{\sigma_0} \right)^n$

$\sigma_{flow} := \frac{\sigma_y + \sigma_u}{2}$ $\sigma_{ref} = L_r \cdot \sigma_{flow}$

$\sigma_0 := \sigma_y$ $\varepsilon_0 := \frac{\sigma_0}{E}$

$\varepsilon_{ref} = \varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right]$

$$K_{rOption2}(L_r) := \left[\frac{E \cdot \left[\varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right] \right]}{L_r \cdot \sigma_y} + \frac{L_r^3 \cdot \sigma_y}{2 \cdot E \cdot \left[\varepsilon_0 \cdot \left[\frac{L_r \cdot \sigma_y}{\sigma_0} + \alpha \cdot \left(\frac{L_r \cdot \sigma_y}{\sigma_0} \right)^n \right] \right]} \right]^{\frac{-1}{2}}$$

Cut-off $L_{r_max} := \frac{\sigma_{flow}}{\sigma_y}$ $L_{r_max} = 1.27$

4. Charecterize flaw

Crack Length, 2C $c := 50mm$ $2c = 100\,mm$

Crack Depth, a $a := 20mm$

5. Select category of analysis category 2

6. Define fracture toughness $K_{0.2}$ and K_g

Material name : SA106Gr. C

$J = C_1(\Delta a)^{C_2}$ $C_1 = 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c} \right)^{2.29}$ $C_2 := 0.38$

$s_u := \sigma_u$ $s_y := \sigma_y$

$\varepsilon_c := 0.82$ $E = 200.291\,GPa$

$k := 1mm$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$\Delta a := 0.2$

$S_u := \frac{s_u}{E}$ $\quad S_u = 2.067 \times 10^{-3}$

$C_1 := 2927 \cdot E \cdot k \cdot \left(S_u \cdot \sqrt{\varepsilon_c}\right)^{2.29}$ $\quad C_1 = 22.153\,N\,\frac{N}{mm}$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $\quad J(\Delta a) = 12.018\,N\,\frac{N}{mm}$

$K_{0.2} := \sqrt{\frac{(J(\Delta a) \cdot E)}{\left(1 - 0.3^2\right)}}$ $\quad K_{0.2} = 199.191\,MPa \cdot \sqrt{m}$

$\Delta a_g := 2$

$J(\Delta a) := C_1 \cdot \Delta a^{C_2}$ $\quad J(\Delta a_g) = 28.829\,N\,\frac{N}{mm}$

$K_g := \sqrt{\frac{\left(J(\Delta a_g) \cdot E\right)}{\left(1 - 0.3^2\right)}}$ $\quad K_g = 308.51\,MPa \cdot \sqrt{m}$

7. Define crack size a_0

$a_0 := a$ $\quad a_0 = 20\,mm$

Assume $\Delta a_g := 2mm$

8. Calculate L_r $\quad L_r = P/P_L$

Original crack size a_0

Limit load from Zahoor [20] $\quad \sigma_{ref} := \sigma_{flow}$

Internal Pressure Where the flaw is assumed to be of rectangular shape

where $Po := \sigma_{flow} \cdot \left(\frac{t}{R_o}\right)$ $\quad Po = 32.442\,MPa$

$x := \frac{a}{t}$ $\quad x = 0.351$

$M := \left[1 + 1.61 \cdot \frac{c^2}{(R \cdot t)}\right]^{0.5}$ $\quad M = 1.06$

$P_L := Po \cdot \left[\frac{(1 - x)}{\left(1 - \frac{x}{M}\right)}\right]$ $\quad P_L = 31.480\,MPa$

$L_R(P) := \frac{P}{P_L}$ $\quad L_R(P_{op}) = 0.635$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Ductile crack growth a_g

Limit Load from Zahoor $\sigma_{ref} := \sigma_{flow}$

Internal Pressure Where the flaw is assumed to be of rectangular shape

where $Po := \sigma_{flow} \cdot \left(\frac{t}{R_o} \right)$ $Po = 32.442\,MPa$

$x := \frac{a + \Delta a_g}{t}$ $x = 0.386$

$M := \left[1 + 1.61 \cdot \frac{c^2}{(R \cdot t)} \right]^{0.5}$ $M = 1.06$

$P_L := Po \cdot \left[\frac{(1 - x)}{\left(1 - \frac{x}{M} \right)} \right]$ $P_L = 31.328\,MPa$

$L_{Rg}(P) := \frac{P}{P_L}$ $L_{Rg}(P_{op}) = 0.638$

9.Calculate K_r

Parameter K from Zahoor [20] $K_I = \sigma_h \cdot (\pi \cdot t)^{0.5} \cdot F_1$

The available K-solution of this cracked structure is valid within the following range

$0.05 \le \frac{a}{t} \le 0.85$ $1 \le \frac{R_i}{t} \le 10$ $0.1 \le \frac{a}{c} \le 1$ $\alpha \ge 0.2$

For our problem $\frac{a}{t} = 0.351$ $\frac{R_i}{t} = 9.526$ $\frac{a}{c} = 0.4$

$\frac{a + \Delta a_g}{t} = 0.386$ $\frac{a + \Delta a_g}{c} = 0.44$

Original crack size a_0

where $\alpha_1 := \frac{\left(\frac{a}{t} \right)}{\left(\frac{a}{c} \right)^{0.58}}$ $\alpha_1 = 0.597$

$\sigma_h(P) := P \cdot \frac{\left(R_o^2 + R_i^2 \right)}{\left(R_o^2 - R_i^2 \right)}$ $\sigma_h(P_{op}) = 201.025\,MPa$

$F_1 := 0.25 + \frac{\left(0.4759 \cdot \alpha_1 + 0.1262 \cdot \alpha_1^2 \right)}{\left[0.102 \cdot \left(\frac{R_i}{t} \right) - 0.02 \right]^{0.1}}$ $F_1 = 0.581$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$K_I(P) := \sigma_h(P)\cdot(\pi\cdot t)^{0.5}\cdot F_1 \qquad K_I(P_{op}) = 49.4\,MPa\cdot\sqrt{m}$$

$$K_R(P) := \frac{K_I(P)}{K_{0.2}} \qquad K_R(P_{op}) = 0.248$$

Ductile crack growth a_g

Parameter K from Zahoor $\qquad K_I = \sigma_h\cdot(\pi\cdot t)^{0.5}\cdot F_1$

where $\qquad \alpha_1 := \dfrac{\left(\dfrac{a + \Delta a_g}{t}\right)}{\left(\dfrac{a + \Delta a_g}{c}\right)^{0.58}} \qquad \alpha_1 = 0.621$

$$\sigma_h(P) := P\cdot\frac{\left(R_o^{\,2} + R_i^{\,2}\right)}{\left(R_o^{\,2} - R_i^{\,2}\right)} \qquad \sigma_h(P_{op}) = 201.025\,MPa$$

$$F_1 := 0.25 + \frac{\left(0.4759\cdot\alpha_1 + 0.1262\cdot\alpha_1^{\,2}\right)}{\left[0.102\cdot\left(\dfrac{R_i}{t}\right) - 0.02\right]^{0.1}} \qquad F_1 = 0.596$$

$$K_I(P) := \sigma_h(P)\cdot(\pi\cdot t)^{0.5}\cdot F_1 \qquad K_I(P_{op}) = 50.712\,MPa\cdot\sqrt{m}$$

$$K_{Rg}(P) := \frac{K_I(P)}{K_g} \qquad K_{Rg}(P_{op}) = 0.164$$

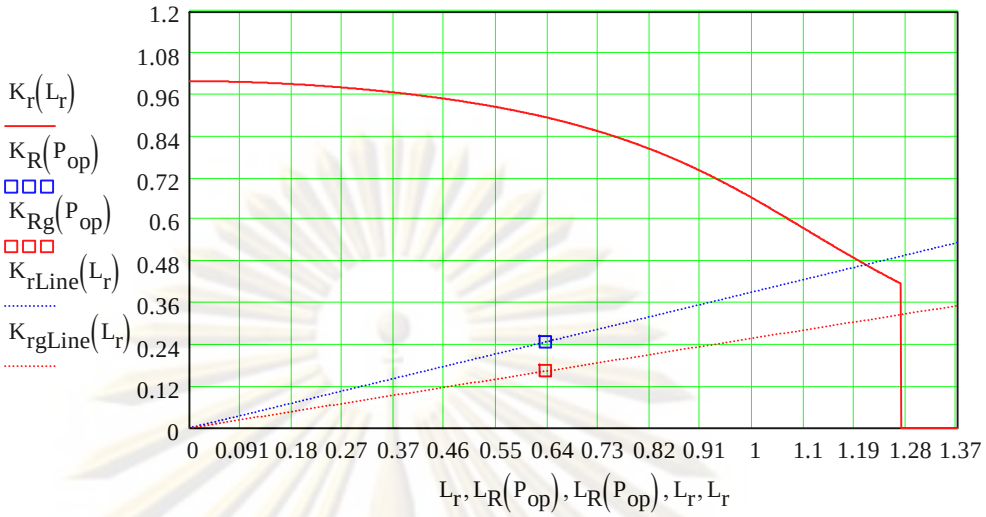
10. Plot assessment piont (L_r, K_r) $\qquad L_r := 0, 0.001 .. 1.5$

$$K_r(L_r) := \begin{cases} K_{rOption2}(L_r) & \text{if } 0 \le L_r \le L_{r_max} \\ 0 & \text{otherwise} \end{cases}$$

$$K_{rLine}(L_r) := \frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})$$

$$K_{rgLine}(L_r) := \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})}\cdot\left(L_r - L_{Rg}(P_{op})\right) + K_{Rg}(P_{op})$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



11. Not consider crack growth caused by fatigue or EAC

12. Reserve factor, F_L

$$\left[\frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] = K_{rOption2}(L_r)$$

Original crack size a_0

$$f(L_r) := \left[\frac{K_R(P_{op})}{L_R(P_{op})}\cdot\left(L_r - L_R(P_{op})\right) + K_R(P_{op})\right] - K_{rOption2}(L_r)$$

$L_r := 1$

$L_{rC} := root(f(L_r), L_r)$ $\quad L_{rC} = 1.206$

$K_{rC} := K_{rOption2}(L_{rC})$ $\quad K_{rC} = 0.471$

$K_{rmax} := K_{rOption2}(L_{r_max})$ $\quad K_{rmax} = 0.416$

L_r Critical $\quad L_{rC1} := \begin{cases} L_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R(P_{op})}{L_R(P_{op})} \\ L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$L_{rC1} = 1.206$

K_r Critical $\quad K_{rC1} := \begin{cases} K_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_R(P_{op})}{L_R(P_{op})} \\ \frac{K_R(P_{op})}{L_R(P_{op})}\cdot L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_R(P_{op})}{L_R(P_{op})} \end{cases}$

$K_{rC1} = 0.471$

$$d := \sqrt{K_R(P_{op})^2 + L_R(P_{op})^2} \qquad dC := \sqrt{K_{rC}^2 + L_{rC}^2}$$

$$FL := \frac{dC}{d} \qquad FL = 1.899$$

Ductile crack growth a_g

$$f(L_r) := \left[\frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \cdot (L_r - L_{Rg}(P_{op})) + K_{Rg}(P_{op})\right] - K_{rOption2}(L_r)$$

$$L_r := 1$$

$$L_{rC} := root(f(L_r), L_r) \qquad L_{rC} = 1.355$$

$$K_{rC} := K_{rOption2}(L_{rC}) \qquad K_{rC} = 0.35$$

L_r Critical

$$L_{rC1} := \begin{cases} L_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \\ L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \end{cases}$$

$$L_{rC1} = 1.27$$

K_r Critical

$$K_{rC1} := \begin{cases} K_{rC} & \text{if } \frac{K_{rmax}}{L_{r_max}} < \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \\ \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \cdot L_{r_max} & \text{if } \frac{K_{rmax}}{L_{r_max}} > \frac{K_{Rg}(P_{op})}{L_{Rg}(P_{op})} \end{cases}$$

$$K_{rC1} = 0.327$$

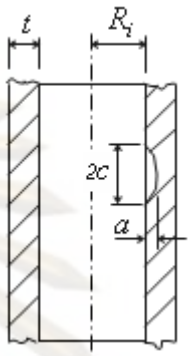
$$d := \sqrt{K_{Rg}(P_{op})^2 + L_{Rg}(P_{op})^2} \qquad dC := \sqrt{K_{rC1}^2 + L_{rC1}^2}$$

$$FL := \frac{dC}{d} \qquad FL = 1.989$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข.3 การประเมินการเติบโตของรอยร้าวล้า

$cyc := 1 \qquad MPa := 10^6 Pa$



Structure's dimensions

Outer radius, RO $R_o := 600mm$

Inner radius, Ri $R_i := 543mm$

Thickness, t $t := R_o - R_i$

Meterial properties

$C_{paris} := 1 \cdot 10^{-9} \cdot \frac{m}{cyc}$

$n := 3$

Crack size

Crack Length, 2C $c_0 := 50mm \qquad 2c_0 = 100\,mm$

Crack Depth, a $a_0 := 20mm$

Fatigue load

maximun load $P_{max} := 4MPa$

minimum load $P_{min} := 0MPa$

$\Delta P := P_{max} - P_{min}$

$\Delta P = 4 \times 10^6\,Pa$

Number of cycles, N $N := 4000$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

$$
\begin{aligned}
\text{crackSize} := \;& c_0 \leftarrow 50\cdot mm \\
& a_0 \leftarrow 20\cdot mm \\
& \text{for } i \in 0..N \\
& \quad \alpha \leftarrow \frac{\left(\frac{a_i}{t}\right)}{\left(\frac{a_i}{c_i}\right)^{0.58}} \\
& \quad \sigma_h \leftarrow \Delta P\cdot\frac{\left(R_o^{\,2} + R_i^{\,2}\right)}{\left(R_o^{\,2} - R_i^{\,2}\right)} \\
& \quad F \leftarrow 0.25 + \frac{\left(0.4759\cdot\alpha + 0.1262\cdot\alpha^{2}\right)}{\left[0.102\cdot\left(\frac{R_i}{t}\right) - 0.02\right]^{0.1}} \\
& \quad F_s \leftarrow F\cdot\left[1.06 + 0.28\cdot\left(\frac{a_i}{t}\right)^{2}\right]\cdot\left(\frac{a_i}{c_i}\right)^{0.41} \\
& \quad \Delta KA_i \leftarrow \sigma_h\cdot(\pi\cdot t)^{0.5}\cdot F \\
& \quad \Delta KB_i \leftarrow \sigma_h\cdot(\pi\cdot t)^{0.5}\cdot F_s \\
& \quad da \leftarrow C_{paris}\cdot\left[\Delta KA_i\cdot\left(MPa\cdot\sqrt{m}\right)^{-1}\right]^{n} \\
& \quad dc \leftarrow 0.9^{n}\cdot C_{paris}\cdot\left[\Delta KB_i\cdot\frac{\left(Pa\cdot\sqrt{m}\right)^{-1}}{10^{6}}\right]^{n} \\
& \quad a_{i+1} \leftarrow a_i + da \\
& \quad c_{i+1} \leftarrow c_i + dc \\
& \begin{pmatrix} a\cdot mm^{-1} \\ 2\cdot c\cdot mm^{-1} \end{pmatrix}
\end{aligned}
$$

$$\left(\text{crackSize}_0\right)_N = 24.252557386$$

$$\left(\text{crackSize}_1\right)_N = 102.995487643$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค.

ซอร์สโค้ดของโปรแกรม

ค.1 ซอร์สโค้ดของฟอร์ม base

```
unit frmBaseU;

interface

uses
   Windows, Messages, SysUtils, Variants, Classes, Graphics,
   Controls,   Forms,Dialogs, StructureU, StructureFactoryU,
   AssessmentU;

type
  TfrmBase = class(TForm)
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);

  private
    { Private declarations }
  protected
    Structure       : TStructure;
    Assessment      : TAssessment;
    StructureCreator: TStructureCreator;

    //descendant of TfrmBase can link through to TAssessment
  public
    { Public declarations }
  end;

var
  frmBase: TfrmBase;
implementation

{$R *.dfm}

{ TfrmBase }
procedure TfrmBase.FormCreate(Sender: TObject);
begin
  Assessment       := TAssessment.Create;
  StructureCreator := TStructureCreator.Create;
end;

procedure TfrmBase.FormDestroy(Sender: TObject);
begin
  Assessment.Free;
  StructureCreator.Free;
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.2 ซอร์สโค้ดของฟอร์ม UserIn

```
unit frmUserInU;

interface

uses
   Windows, Messages, SysUtils, Variants, Classes, Graphics,
   Controls, Forms,Dialogs, frmBaseU, StdCtrls, Buttons, ExtCtrls,
   ComCtrls;

type
  TfrmUserIn = class(TfrmBase)
    gpbInput          : TGroupBox;
    lblCodeProcedure  : TLabel;
    lblStructureType  : TLabel;
    lblCrackType      : TLabel;
    lblSourceMatProp  : TLabel;
    Label1            : TLabel;
    Label3            : TLabel;
    cmbCode           : TComboBox;
    cmbStructure      : TComboBox;
    cmbCrack          : TComboBox;
    btnGeometry       : TButton;
    ragLevel          : TRadioGroup;
    ragOption         : TRadioGroup;
    pgcLoads          : TPageControl;
    tbsPrimary        : TTabSheet;
    edtAxial          : TEdit;
    edtBen            : TEdit;
    edtInPress        : TEdit;
    chkAxial          : TCheckBox;
    chkBen            : TCheckBox;
    chkInPress        : TCheckBox;
    chkAr             : TCheckBox;
    lblEdt0           : TLabeledEdit;
    lblEdt1           : TLabeledEdit;
    lblEdt2           : TLabeledEdit;
    lblEdt3           : TLabeledEdit;
    tbsSecondary      : TTabSheet;
    chkThermal        : TCheckBox;
    btnOKLoad         : TButton;
    btnCrackProp      : TButton;
    btnMatCustom      : TButton;
    gpbOutput         : TGroupBox;
    lstShow           : TListBox;
    PanelBtn          : TPanel;
    bbtRunTotal       : TBitBtn;
    bbtClose          : TBitBtn;
    ragAssessPos      : TRadioGroup;
    procedure cmbCodeChange    (Sender: TObject);
    procedure ragLevelClick    (Sender: TObject);
    procedure ragOptionClick   (Sender: TObject);
    procedure cmbStructureChange(Sender: TObject);
    procedure btnGeometryClick (Sender: TObject);
    procedure btnMatCustomClick(Sender: TObject);
    procedure pgcLoadsChange   (Sender: TObject);
```

```
    procedure chkAxialClick     (Sender: TObject);
    procedure chkBenClick       (Sender: TObject);
    procedure chkInPressClick   (Sender: TObject);
    procedure chkArClick        (Sender: TObject);
    procedure btnOKLoadClick    (Sender: TObject);
    procedure edtAxialKeyPress  (Sender: TObject; var Key: Char);
    procedure edtBenKeyPress    (Sender: TObject; var Key: Char);
    procedure edtInPressKeyPress(Sender: TObject; var Key: Char);
    procedure FormCreate        (Sender: TObject);
    procedure btnCrackPropClick (Sender: TObject);
    procedure bbtRunTotalClick  (Sender: TObject);
    procedure bbtCloseClick     (Sender: TObject);
  private
    { Private declarations }
    function  CheckBlank      (pgcLoads: TPageControl): string;
    procedure Hide_Edit_Loads (aaa     : TComponent);
    procedure Hide            (bbb     : TComponent);
    procedure CrackGrowthM;
    procedure assign_show_crack(AType: integer; ALength, ADepth:Real;
                                AOriented, APosition: string);
    function  LoadStatus    : string;

    procedure ShowMatpropLevel1_Option1(AKName: string);
    procedure ShowMatpropLevel1_Option2(AKName: string);
    procedure ShowMatpropLevel2_Option1;
    procedure ShowMatpropLevel2_Option2;

    procedure CalAndShowLevel1_Option;
    procedure CalAndShowLevel2_Option;
  public
    { Public declarations }
    function GetPrinStressH : Real;
    function GetPrinStressA : Real;
    function GetAxialTension: Real;
    function GetBending     : Real;
    function GetInPressure  : Real;
    function GetROut        : Real;
    function GetRIn         : Real;
  end;

var
  frmUserIn: TfrmUserIn;

implementation

uses frmStructureU, frmPipeU, frmCrackU, StructureU, frmCrackGrowthU,
     frmMatPropMain;

{$R *.dfm}

///----------------------------Code----------------------------///
procedure TfrmUserIn.cmbCodeChange(Sender: TObject);
begin
  inherited;
    cmbCode.ItemIndex      := 0;
    cmbStructure.ItemIndex := 0;
end;
///--------------------------Level---------------------------///
procedure TfrmUserIn.ragLevelClick(Sender: TObject);
```

```
begin
  inherited;
  Assessment.Level(ragLevel.ItemIndex);
  if ragLevel.ItemIndex >= 2 then
    begin
      ShowMessage(ragLevel.Items[ragLevel.ItemIndex] +
                                                ' Not Available');
      ragLevel.ItemIndex := -1;
    end;
end;

///----------------------------Option----------------------------///
procedure TfrmUserIn.ragOptionClick(Sender: TObject);
begin
  inherited;
  Assessment.AssignFAD(ragOption.ItemIndex);
  if ragOption.ItemIndex >= 2 then
    begin
      ShowMessage(ragOption.Items[ragOption.ItemIndex] +
                                                ' Not Available');
      ragOption.ItemIndex := -1;
    end;
end;

///----------------------------Structure-------------------------///
procedure TfrmUserIn.cmbStructureChange(Sender: TObject);
begin
  inherited;
   if cmbStructure.ItemIndex <> 0 then
    begin
      ShowMessage(cmbStructure.Items[cmbStructure.ItemIndex] +
                                          ' is not available ');
      cmbStructure.ItemIndex := 0;
    end;
end;

///----------------------Structure Geometry---------------------///
procedure TfrmUserIn.btnGeometryClick(Sender: TObject);
begin
  inherited;
  if cmbStructure.ItemIndex > -1 then
    begin
      FreeAndNil (Structure);
      Structure := StructureCreator.CreatStructure
                   (cmbStructure.ItemIndex);

      case cmbStructure.ItemIndex of
      0: begin
           frmStructure := TfrmPipe.Create(Self);
         end;
      1: begin

         end;
      end;
      frmStructure.Initial;
      if frmStructure.InfoValid = True then
         begin
           Structure.SetDimension(frmStructure.Dimension1,
                                  frmStructure.Dimension2);
```

```
            lstShow.Items.Add('Structure is '+
                              Structure.StructureName);
            lstShow.Items.Add('  - Inner Radius: ' +
                             floatToStrF(Structure.Dimension1,
                              ffNumber,5,3) + ' m.');
            lstShow.Items.Add('  - Outer Radius: ' +
                              FloatToStrF(Structure.Dimension2,
                              ffNumber,5,3) + ' m.');
          end
        else
          ShowMessage(' You not have assigned structural geometry '
                      + #10#13
                      + ' Please define structural geometry again ');
    end
  else
    ShowMessage(' Please Select Structure Type ');
end;

///---------------------Material properties-----------------------///
procedure TfrmUserIn.btnMatCustomClick(Sender: TObject);
var Sy      : Real;
    Su      : Real;
    SE      : Real;
    KOption1: Real;
    KName   : string;
    K02     : Real;
    Kg      : Real;
    alpha   : Real;
    n       : Real;
    MatName : string;
begin
  inherited;
  if assigned(Structure) then
    if ragLevel.ItemIndex <> -1 then
      if ragOption.ItemIndex <> -1 then
        begin //2
          frmMatPropM := TfrmMatPropM.Create(self);
                         frmMatPropM.Initial(ragLevel.ItemIndex,
                         ragOption.ItemIndex);
          case ragLevel.ItemIndex of  //2.1
          0: begin      // begin level 1
               case ragOption.ItemIndex of
               0: begin      // begin level 1 option 1
                    if frmMatPropM.InfoValid = true then
                      begin
                        MatName  := frmMatPropM.GetMatName;
                        Sy       := frmMatPropM.GetSy;
                        Su       := frmMatPropM.GetSu;
                        SE       := frmMatPropM.GetSE;
                        KOption1 := frmMatPropM.GetKOption1;
                        KName    := frmMatPropM.GetKName;
                        Structure.AssignMatProp(MatName, Sy,
                                                Su, SE, 0, 0);
                        Structure.SetToughness(KOption1, 0, 0);
                        ShowMatpropLevel1_Option1(KName);
                      end
                    else
                      begin
                        ShowMessage(' You not have assigned
```

```
                        material properties' + ' ' +
                        #10#13 +
                        ' Please define material
                        properties again ');
              end;
           end;     // end level 1 option 1
        1: begin    // begin level 1 option 2
             if frmMatPropM.InfoValid = true then
               begin
                 MatName  := frmMatPropM.GetMatName;
                 Sy       := frmMatPropM.GetSy;
                 Su       := frmMatPropM.GetSu;
                 SE       := frmMatPropM.GetSE;
                 alpha    := frmMatPropM.GetAlpha;
                 n        := frmMatPropM.GetExponentN;
                 KOption1 := frmMatPropM.GetKOption1;
                 KName    := frmMatPropM.GetKName;
                 Structure.AssignMatProp(MatName, Sy,
                                         Su, SE, alpha, n);
                 Structure.SetToughness(KOption1, 0, 0);
                 Assessment.AssignConstitutive(Structure);
                 ShowMatpropLevel1_Option2(KName);
               end
             else
               begin
                 ShowMessage(' You not have assigned
                        material properties'
                        + ' ' + #10#13 +
                        ' Please define material
                        properties again ');
               end;
           end;     // end level 1 option 2
         end;     // end case ragOption.ItemIndex
       end;     // end level 1
    1: begin    // begin level 2
         case ragOption.ItemIndex of
         0: begin    // begin level 2 option 1
              if frmMatPropM.InfoValid = true then
                begin
                  MatName  := frmMatPropM.GetMatName;
                  Sy       := frmMatPropM.GetSy;
                  Su       := frmMatPropM.GetSu;
                  SE       := frmMatPropM.GetSE;
                  K02      := frmMatPropM.GetK02;
                  Kg       := frmMatPropM.GetKg;
                  Structure.AssignMatProp(MatName, Sy,
                                          Su, SE, 0, 0);
                  Structure.SetToughness(0, K02, Kg);
                  ShowMatpropLevel2_Option1;
                end
              else
                begin
                  ShowMessage(' You not have assigned
                        material properties'
                        + ' ' + #10#13 +
                        ' Please define material
                        properties again ');
                end;
```

```
                end;     // end level 2 option 1
              1: begin    // begin level 2 option 2
                   if frmMatPropM.InfoValid = true then
                     begin
                       MatName  := frmMatPropM.GetMatName;
                       Sy       := frmMatPropM.GetSy;
                       Su       := frmMatPropM.GetSu;
                       SE       := frmMatPropM.GetSE;
                       alpha    := frmMatPropM.GetAlpha;
                       n        := frmMatPropM.GetExponentN;
                       K02      := frmMatPropM.GetK02;
                       Kg       := frmMatPropM.GetKg;
                       Structure.AssignMatProp(MatName, Sy, Su,
                                               SE, alpha, n);
                       Structure.SetToughness(0, K02, Kg);
                       Assessment.AssignConstitutive(Structure);
                       ShowMatpropLevel2_Option2;
                     end
                   else
                     begin
                       ShowMessage(' You not have assigned
                                   material properties'
                                   + ' ' + #10#13 +
                                   ' Please define material
                                   properties again ');
                     end;
                end;     // end level 2 option 2
              end;       // end case ragOption.ItemIndex
            end;    // end level 2
          end;   // end case level.ItemIndex
        end   // end 2
      else
        ShowMessage(' Please select option of FAD')
    else
      ShowMessage(' Please select level of analysis')
  else
    ShowMessage(' Please assigned structure first');
end;

procedure TfrmUserIn.ShowMatpropLevel1_Option1(AKName: string);
begin
  lstShow.Items.Add('CODE Procedure : ' +
                    cmbCode.Items[cmbCode.ItemIndex]);
  lstShow.Items.Add('Analysis Level : ' +
                    ragLevel.Items[ragLevel.ItemIndex]);
  lstShow.Items.Add('FAD Option : ' +
                    ragOption.Items[ragOption.ItemIndex]);
  lstShow.Items.Add('Material Name : ' + Structure.GetMatName);
  lstShow.Items.Add('material properties ');
  lstShow.Items.Add('  Tensile Properties ');
  lstShow.Items.Add('    - Yield Stress : ' +
                    FloatToStrF(Structure.GetSy, ffNumber,5,3)
                    + ' Pa.');
  lstShow.Items.Add('    - Ultimate Stress : ' +
                    FloatToStrF(Structure.GetSu, ffNumber,5,3)
                    + ' Pa.');
  lstShow.Items.Add('    - Young Modulus : ' +
                    FloatToStrF(Structure.GetSE, ffNumber,5,3)
                    + ' Pa.');
```

```
  lstShow.Items.Add('  Toughness properties ');
  lstShow.Items.Add('    - ' + AKName + ' : '
                   +    FloatToStrF(Structure.GetKmat,
                   ffNumber,5,3) + ' Pa./m^2');
end;

procedure TfrmUserIn.ShowMatpropLevel1_Option2(AKName: string);
begin
  lstShow.Items.Add('CODE Procedure : '
                   + cmbCode.Items[cmbCode.ItemIndex]);
  lstShow.Items.Add('Analysis Level : '
                   +   ragLevel.Items[ragLevel.ItemIndex]);
  lstShow.Items.Add('FAD Option : '
                   + ragOption.Items[ragOption.ItemIndex]);
  lstShow.Items.Add('Material Name : ' + Structure.GetMatName);
  lstShow.Items.Add('material properties ');
  lstShow.Items.Add('  Tensile Properties ');
  lstShow.Items.Add('    - Yield Stress : ' +
                   FloatToStrF(Structure.GetSy, ffNumber,5,3)
                   + ' Pa.');
  lstShow.Items.Add('    - Ultimate Stress : ' +
                   FloatToStrF(Structure.GetSu, ffNumber,5,3)
                   + ' Pa.');
  lstShow.Items.Add('    - Young Modulus : ' +
                   FloatToStrF(Structure.GetSE, ffNumber,5,3)
                   + ' Pa.');
  lstShow.Items.Add('    - Coefficient, alpha : ' +
                   FloatToStrF(Structure.GetAlpha, ffNumber,5,3));
  lstShow.Items.Add('    - Exponent, n : ' +
                   FloatToStrF(Structure.GetExponentN,
                    ffNumber,5,3));
  lstShow.Items.Add('  Toughness properties ');
  lstShow.Items.Add('    - ' + AKName + ' : ' +
                    FloatToStrF(Structure.GetKmat,
                    ffNumber,5,3) + ' Pa./m^2');
end;

procedure TfrmUserIn.ShowMatpropLevel2_Option1;
begin
  lstShow.Items.Add('CODE Procedure : '
                   + cmbCode.Items[cmbCode.ItemIndex]);
  lstShow.Items.Add('Analysis Level : '
                   + ragLevel.Items[ragLevel.ItemIndex]);
  lstShow.Items.Add('FAD Option : '
                   + ragOption.Items[ragOption.ItemIndex]);
  lstShow.Items.Add('Material Name : ' + Structure.GetMatName);
  lstShow.Items.Add('material properties ');
  lstShow.Items.Add('  Tensile Properties ');
  lstShow.Items.Add('    - Yield Stress : ' +
                   FloatToStrF(Structure.GetSy, ffNumber,5,3)
                   + ' Pa.');
  lstShow.Items.Add('    - Ultimate Stress : ' +
                   FloatToStrF(Structure.GetSu, ffNumber,5,3)
                   + ' Pa.');
  lstShow.Items.Add('    - Young Modulus : ' +
                   FloatToStrF(Structure.GetSE, ffNumber,5,3)
                   + ' Pa.');
  lstShow.Items.Add('  Toughness properties ');
  lstShow.Items.Add('    - K0.2 : ' +
```

```
                    FloatToStrF(Structure.GetK02,ffNumber,5,3)
                    + ' Pa./m^2');
  lstShow.Items.Add('    - Kg : ' +
                    FloatToStrF(Structure.GetKg,ffNumber,5,3)
                    + ' Pa./m^2');
end;

procedure TfrmUserIn.ShowMatpropLevel2_Option2;
begin
  lstShow.Items.Add('CODE Procedure : '
                    + cmbCode.Items[cmbCode.ItemIndex]);
  lstShow.Items.Add('Analysis Level : '
                    + ragLevel.Items[ragLevel.ItemIndex]);
  lstShow.Items.Add('FAD Option : '
                    + ragOption.Items[ragOption.ItemIndex]);
  lstShow.Items.Add('Material Name : ' + Structure.GetMatName);
  lstShow.Items.Add('material properties ');
  lstShow.Items.Add('  Tensile Properties ');
  lstShow.Items.Add('    - Yield Stress : ' +
                    FloatToStrF(Structure.GetSy, ffNumber,5,3)
                    + ' Pa.');
  lstShow.Items.Add('    - Ultimate Stress : ' +
                    FloatToStrF(Structure.GetSu, ffNumber,5,3)
                    + ' Pa.');
  lstShow.Items.Add('    - Young Modulus : ' +
                    FloatToStrF(Structure.GetSE, ffNumber,5,3)
                    + ' Pa.');
  lstShow.Items.Add('    - Coefficient, alpha : ' +
                    FloatToStrF(Structure.GetAlpha, ffNumber,5,3));
  lstShow.Items.Add('    - Exponent, n : ' +
                    FloatToStrF(Structure.GetExponentN,
                     ffNumber,5,3));
  lstShow.Items.Add('  Toughness properties ');
  lstShow.Items.Add('    - K0.2 : ' +
                    FloatToStrF(Structure.GetK02,ffNumber,5,3)
                    + ' Pa./m^2');
  lstShow.Items.Add('    - Kg : ' +
                    FloatToStrF(Structure.GetKg,ffNumber,5,3)
                    + ' Pa./m^2');
end;


///----------------------------Loads----------------------------///
procedure TfrmUserIn.pgcLoadsChange(Sender: TObject);
begin
  inherited;
    if pgcLoads.ActivePageIndex = 0 then
    ShowMessage(' Do you want to difine '
                + pgcLoads.ActivePage.Caption + ' ?')
  else
    ShowMessage(' Do you want to difine '
                + pgcLoads.ActivePage.Caption + ' ?');
end;

procedure TfrmUserIn.chkAxialClick(Sender: TObject);
begin
  inherited;
  if chkAxial.Checked = True then
    begin
```

```
      edtAxial.Enabled := True;
      edtAxial.Color   := clWhite;
      edtAxial.SetFocus;
    end
  else
    begin
      edtAxial.Enabled := False;
      edtAxial.Clear;
      edtAxial.Color   := clBtnFace;
    end;
end;

procedure TfrmUserIn.chkBenClick(Sender: TObject);
begin
  inherited;
  if chkBen.Checked = True then
    begin
      edtBen.Enabled := True;
      edtBen.Color   := clWhite;
      edtBen.SetFocus;
    end
  else
    begin
      edtBen.Enabled := False;
      edtBen.Clear;
      edtBen.Color   := clBtnFace;
    end;
end;

procedure TfrmUserIn.chkInPressClick(Sender: TObject);
begin
  inherited;
  if chkInPress.Checked = True then
    begin
      edtInPress.Enabled := True;
      edtInPress.Color   := clWhite;
      edtInPress.SetFocus;
    end
  else
    begin
      edtInPress.Enabled := False;
      edtInPress.Clear;
      edtInPress.Color   := clBtnFace;
    end;
end;

procedure TfrmUserIn.chkArClick(Sender: TObject);
var i: integer;
begin
  inherited;
  if chkAr.Checked = True then
    begin
      for i := 0 to ComponentCount - 1 do
        if Components[i] is TLabeledEdit then
        begin
          TLabeledEdit(Components[i]).Enabled := True;
          TLabeledEdit(Components[i]).Color   := clWhite;
          TLabeledEdit(Components[i]).SetFocus;
        end;
```

```
    end
  else
    begin
      for i := 0 to ComponentCount - 1 do
        if Components[i] is TLabeledEdit then
        begin
          TLabeledEdit(Components[i]).Clear;
          TLabeledEdit(Components[i]).Enabled := False;
          TLabeledEdit(Components[i]).Color   := clBtnFace;
        end;
    end
end;

procedure TfrmUserIn.btnOKLoadClick(Sender: TObject);
var text        : string;
    Tension     : Real;
    Bending     : Real;
    InPressure  : Real;
    Stress_Coeff: array[0..3] of Real;
begin
  inherited;
  if assigned(Structure) then
  begin
    text := CheckBlank(pgcLoads);
    if text = '' then
      begin
        Tension    := 0;
        Bending    := 0;
        InPressure := 0;

        if (chkAxial.Checked = false)and(chkBen.Checked = false)and
           (chkInPress.Checked = false) and (chkAr.Checked = False)
         then
          begin
            ShowMessage(' Please Select Loads type ');
            exit;
          end;
        if chkAxial.Checked = True then
          Tension := (StrToFloat(edtAxial.Text))*1E6;      // MN
        if chkBen.Checked = True then
          Bending := (StrToFloat(edtBen.Text))*1E6;        // MN- m
        if chkInPress.Checked = True then
          InPressure := (StrToFloat(edtInPress.Text))*1E6; // MPa
        if chkAr.Checked = True then
          begin
            Stress_Coeff[0] := StrToFloat(lblEdt0.Text);
            Stress_Coeff[1] := StrToFloat(lblEdt1.Text);
            Stress_Coeff[2] := StrToFloat(lblEdt2.Text);
            Stress_Coeff[3] := StrToFloat(lblEdt3.Text);
          end;
        Structure.AssignLoads(Tension,Bending,InPressure);
        Structure.SetStress_Coeff(Stress_Coeff);
        lstShow.Items.Add('Loads : ');
        lstShow.Items.Add('  -Axial Tension : ' +
                          FloatToStrF(Structure.GetAxialTension,
                          ffNumber,5,3) + ' N.');
        lstShow.Items.Add('  -Bending Moment : ' +
                          FloatToStrF(Structure.GetBending,
                           ffNumber,5,3) + ' N-m.');
```

```
          lstShow.Items.Add('  -Internal Pressure : ' +
                            FloatToStrF(Structure.GetInPressure,
                             ffNumber,5,3) + ' Pa.');
          lstShow.Items.Add('  -Stress Coefficient : ');
          lstShow.Items.Add('    -S0 : ' +
                            floatToStrF(Structure.GetStress_Coeff[0],
                            ffNumber,5,3));
          lstShow.Items.Add('    -S1 : ' +
                            floatToStrF(Structure.GetStress_Coeff[1],
                            ffNumber,5,3));
          lstShow.Items.Add('    -S2 : ' +
                            floatToStrF(Structure.GetStress_Coeff[2],
                            ffNumber,5,3));
          lstShow.Items.Add('    -S3 : ' +
                            floatToStrF(Structure.GetStress_Coeff[3],
                            ffNumber,5,3));
        end
      else
        begin
          Showmessage('   Please enter :              ' + text);
        end;
    end
    else
      ShowMessage(' Please assigned structure ');
end;

procedure TfrmUserIn.edtAxialKeyPress(Sender: TObject;
                                      var Key:   Char);
begin
  inherited;
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmUserIn.edtBenKeyPress(Sender: TObject; var Key: Char);
begin
  inherited;
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmUserIn.edtInPressKeyPress(Sender: TObject;
                                        var Key: Char);
begin
  inherited;
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;
```

```
procedure TfrmUserIn.FormCreate(Sender: TObject);
begin
  inherited;
  Hide_Edit_Loads(tbsPrimary);
  Hide_Edit_Loads(tbsSecondary);
end;

function TfrmUserIn.CheckBlank(pgcLoads: TPageControl): string;
var  i      : integer;
     temp,
     temp1,
     temp2  : string;
begin
  for i := 0 to self.ComponentCount-1 do
  begin
    if self.Components[i] is TEdit then
      begin
        with self.Components[i] as TEdit do
          begin
            if Enabled = true then
               begin
                 if text = '' then
                   begin
                     temp1 := self.Components[i].Name;
                     if temp1 = 'edtAxial' then
                       temp1 := '  -Axial Tension';
                     if temp1 = 'edtBen' then
                       temp1 := '  -Bending Moment';
                     if temp1 = 'edtInPress' then
                       temp1 := '  -Internal Pressure';
                    temp := temp + #13#10 +'    '+ temp1;
                    setfocus;
                    end;
               end
          end;
      end;
    if self.Components[i] is TLabeledEdit then
      begin
        with self.Components[i] as TLabeledEdit do
          begin
            if Enabled = true then
              begin
                if text = '' then
                  begin
                    temp2 := self.Components[i].Name;
                    if temp2 = 'lblEdt0' then
                      temp2 := '  -Stress Coefficient S0';
                    if temp2 = 'lblEdt1' then
                      temp2 := '  -Stress Coefficient S1';
                    if temp2 = 'lblEdt2' then
                      temp2 := '  -Stress Coefficient S2';
                    if temp2 = 'lblEdt3' then
                      temp2 := '  -Stress Coefficient S3';
                    temp := temp + #13#10 + '    ' + temp2;
                    setfocus;
                  end;
              end;
          end;
      end;
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
  end;
  Result := temp;
end;

procedure TfrmUserIn.Hide_Edit_Loads(aaa: TComponent);
var i: integer;
begin
  if aaa is TTabSheet then
    for i := 0 to (aaa as TTabSheet).ControlCount-1 do
      Hide((aaa as TTabSheet).Controls[i]);
end;

procedure TfrmUserIn.Hide(bbb: TComponent);
begin
  if bbb is TEdit then
    begin
      (bbb as TEdit).Color   := clBtnFace;
      (bbb as TEdit).Enabled := False;
    end;
  if bbb is TLabeledEdit then
    begin
      (bbb as TLabeledEdit).Color   := clBtnFace;
      (bbb as TLabeledEdit).Enabled := False;
    end;
end;

///-------------------------Crack--------------------------------///
procedure TfrmUserIn.btnCrackPropClick(Sender: TObject);
var Depth        : Real;
    Length       : Real;
    Oriented     : string;
    Position     : string;
    CrackType    : string;
    CType        : integer;

    Ri, Ro, t    : Real;    // Pipe dimension
    AperC        : Real;    // a/c
    AperT        : Real;    // a/t
    TwoCperA     : Real;    // 2c/a

    Load         : string;
begin
  inherited;
  if Assigned(Structure) then
    if ragLevel.ItemIndex <> -1 then
      if ragOption.ItemIndex <> -1 then
        begin    //1
          frmCrack := TfrmCrack.Create(self);
          frmCrack.Initial(ragLevel.ItemIndex);
          if frmCrack.Infovalid = True then
            begin    //1.1
              Depth     := frmCrack.Depth;
              Length    := frmCrack.Length;
              Oriented  := frmCrack.Oriented;
              Position  := frmCrack.Position;
              CrackType := frmCrack.CrackType;

              CType := 0;
              if CrackType = 'Semi-Elliptical' then
```

```
  begin
    CType := 0;
  end
else if CrackType = 'Embeded' then
  begin
    CType := 1;
  end;

Ri       := Structure.Dimension1;
Ro       := Structure.Dimension2;
t        := Ro - Ri;
AperC    := Depth/(Length/2);
AperT    := Depth/t;
TwoCperA := Length/Depth;

Load := LoadStatus;
if (Load = 'ATL') or (Load = 'BDL') or
   (Load = 'ATLBDL') then
  begin
    if (0.05 <= AperT) and (AperT <= 0.8) then
      begin
        if TwoCperA >= 3 then
          begin
            assign_show_crack(CType, Length,
                              Depth, Oriented,
                              Position);
          end
        else
          begin
            ShowMessage('range of 2c/a >= 3' + #13#10
                        + '  2c/a = ' +
                         FloatToStr(TwoCperA)
                        + ' out of range');
          end;
      end
    else
      begin
        ShowMessage('range of  0.05 <= a/t <= 0.8 '
                    + #13#10 +
                    '  a/t = ' + FloatToStr(AperT) +
                    ' out of range');
      end;
  end
else if (Load = 'IPL') then
  begin
    if (0.05 <= AperT) and (AperT <= 0.85) then
      begin
        if (0.1 <= AperC) and (AperC <= 1) then
          begin
            assign_show_crack(CType, Length,
                              Depth, Oriented,
                              Position);
          end
        else
          begin
            ShowMessage('range of 0.1 <= a/c <= 1'
                        + #13#10 +
                        '  a/c = ' + FloatToStr(AperC)
                       + ' out of range');
```

```
                    end;
                  end
                else
                  begin
                    ShowMessage('range of  0.05 <= a/t <= 0.85 '
                                + #13#10 +
                                '  a/t = ' + FloatToStr(AperT) +
                                ' out of range');
                  end;
              end
            else  // Load combine
              begin
                if (0.05 <= AperT) and (AperT <= 0.8) then
                  begin
                    if (TwoCperA >= 3) or (0.1 <= AperC)
                                       and (AperC <= 1) then
                      begin
                        assign_show_crack(CType, Length,
                                          Depth, Oriented,
                                          Position);
                      end
                    else
                      begin
                        ShowMessage('range of 2c/a >= 3' + #13#10 +
                                    '  2c/a = ' +
                                    FloatToStr(TwoCperA) +
                                    'range of 0.1 <= a/c <= 1' +
                                    #13#10 +
                                    '  a/c = ' + FloatToStr(AperC)
                                    + ' out of range');
                      end;
                  end
                else
                  begin
                    ShowMessage('range of  0.05 <= a/t <= 0.8 '
                                + #13#10 +
                                '  a/t = ' + FloatToStr(AperT) +
                                ' out of range');
                  end;
              end;
          end  //1.1
        else
          ShowMessage(' You not have assigned crack properties '
                      +  #10#13 +
                      ' Please define crack properties again ');
      end  //1
    else
      ShowMessage(' Please Select Option of FAD')
  else
    ShowMessage(' Please Select Level of analysis')
else
  ShowMessage(' Please Assigned Structure first ')
end;

procedure TfrmUserIn.assign_show_crack(AType: integer;
                                       ALength, ADepth: Real;
                                       AOriented, APosition: string);
var  CrackDepth_ag : Real;
begin
```

```
  case ragLevel.ItemIndex of
  0: begin
       Structure.AssignCrack(AType, ALength, ADepth,
                             AOriented, APosition);
       lstShow.Items.Add('Crack is '+ Structure.GetCrackType);
       lstShow.Items.Add('  - Position: ' + Structure.GetPosition);
       lstShow.Items.Add('  - Orientation: '
                         + Structure.GetOrientation + ' Direction');
       lstShow.Items.Add('  - Depth: ' +
                         FloatToStrF(Structure.GetOCDepth,
                         ffNumber,5,3) + ' m.');
       lstShow.Items.Add('  - Length: ' +
                         FloatToStrF(Structure.GetOCLength,
                         ffNumber,5,3)+ ' m.');
     end;
  1: begin
       CrackDepth_ag := frmCrack.DepthDelta;
       Structure.AssignCrack(AType, ALength, ADepth,
                             AOriented, APosition);
       Structure.SetCCDepth(CrackDepth_ag);
       lstShow.Items.Add('Crack is '+ Structure.GetCrackType);
       lstShow.Items.Add('  - Position: ' + Structure.GetPosition);
       lstShow.Items.Add('  - Orientation: ' +
                         Structure.GetOrientation + ' Direction');
       lstShow.Items.Add('  - Depth: ' +
                         FloatToStrF(Structure.GetOCDepth,
                         ffNumber,5,3) + ' m.');
       lstShow.Items.Add('  - Length: ' +
                         FloatToStrF(Structure.GetOCLength,
                         ffNumber,5,3) + ' m.');
       lstShow.Items.Add('  - Depth Delta, ag: ' +
                         FloatToStrF(CrackDepth_ag,
                         ffNumber,5,3) + ' m.');
     end;
  end;
end;

////--------------------------- RUN -----------------------------////
procedure TfrmUserIn.bbtRunTotalClick(Sender: TObject);
begin
  inherited;
  if assigned (Structure) then
    begin  //1
      Assessment.AssignFAD(ragOption.ItemIndex); // Create FAD Objecj
      Assessment.AssignSIF;                      // Create SIF Object
      Assessment.AssignLimitLoads;               // Create LimitLoad
      Assessment.CalKI(Structure, ragAssessPos.ItemIndex);
      Assessment.CalLimitLoad(Structure);
      Assessment.CalKr(Structure);
      Assessment.CalLr(Structure);

      case ragLevel.ItemIndex of //2 case of ragLevel.ItemIndex
      0: begin                          // begin level 1
           case ragOption.ItemIndex of
           0: begin                     // begin level 1 option1
                CalAndShowLevel1_Option;
              end;                      // end   level 1 option 1
           1: begin                     // begin level 1 option 2
                CalAndShowLevel1_Option;
```

```
            end;                          // end   level 1 option 2
          end;
        end;                              // end   level 1
      1: begin                            // begin level 2
          case ragOption.ItemIndex of
          0: begin                        // begin level 2 option1
               CalAndShowLevel2_Option;
             end;
          1: begin                        // begin level 2 option 2
               CalAndShowLevel2_Option;
             end;                         // end   level 2 option 2
          end;
        end;                              // end   level 2
      end;                          // end 2 case of ragLevel.ItemIndex
      end
  else
    begin
      ShowMessage('Please Assigned Structure first');
    end;
end;

procedure TfrmUserIn.bbtCloseClick(Sender: TObject);
begin
  inherited;
  frmStructure.Free;
  frmCrack.Free;
  StructureCreator.Free;
  Structure.Free;
end;

function TfrmUserIn.GetPrinStressH: Real;
begin
  Result := Structure.GetPrinStressH;
end;

function TfrmUserIn.GetPrinStressA: Real;
begin
  Result := Structure.GetPrinStressA;
end;

procedure TfrmUserIn.CrackGrowthM;
var Mechanism : integer;
    NOCycle   : integer;
    Equation  : integer;
    LoadMax   : Real;
    LoadMin   : Real;
    CoeffC    : Real;
    ExponentM : Real;
    CycleC    : integer;
begin
  Mechanism := frmCrackGrowth.Mechanism;
  NOCycle   := frmCrackGrowth.NOCycle;
  Equation  := frmCrackGrowth.Equation;
  LoadMax   := frmCrackGrowth.LoadMax;
  LoadMin   := frmCrackGrowth.LoadMin;
  CoeffC    := frmCrackGrowth.CoffC;
  ExponentM := frmCrackGrowth.CoffM;

  Assessment.AssignCrackGrowth(Mechanism, NOCycle, Equation);
```

```
  Structure.SetFatigueLoad(LoadMax, LoadMin);
  Structure.SetCGrowthProp(CoeffC, ExponentM);

  lstShow.Items.Add('Mechanism  : ' + Assessment.GetMechanism);
  lstShow.Items.Add('Number of Cycles ' +
                  FloatToStr(Assessment.GetLoop) + ' Cycle');

  if Assessment.GetNODirector = 0 then
  lstShow.Items.Add('Crack Growth Model  : ' + 'Paris');
  lstShow.Items.Add('Load Max.  : ' +
                    FloatToStr(Structure.GetInPressureMax));
  lstShow.Items.Add('Load Min.  : ' +
                    FloatToStr(Structure.GetInPressureMin));

  lstShow.Items.Add('Coefficient, c  : ' +
                    FloatToStr(Structure.GetCoeffC));
  lstShow.Items.Add('Exponent, m  : ' +
                    FloatToStr(Structure.GetExponentM));
  Assessment.CalCrackGrowth(Structure);
             //-----------------------------------------------//
  CycleC := Assessment.GetFCycleC;
    if CycleC > 0 then
      begin
        lstShow.Items.Add('critical cycle  : ' +
                          FloatToStr(Assessment.GetFCycleC) + '
                          Cycle');
        lstShow.Items.Add('critical crack depth  : ' +
                          FloatToStrF(Assessment.GetCrackDepthNew,
                          ffNumber,11,9) + ' m.');
        lstShow.Items.Add('critical crack length  : ' +
                          FloatToStrF(Assessment.GetCrackLengthNew,
                          ffNumber,11,9) + ' m.');
      end
    else
      begin
        lstShow.Items.Add('Cycle number : ' +
                          FloatToStr(Assessment.GetLoop) + ' Cycle');
        lstShow.Items.Add('Crack Depth New : ' +
                          FloatToStrF(Assessment.GetCrackDepthNew,
                          ffNumber,11,9) + ' m.');
        lstShow.Items.Add('Crack Length New : ' +
                          FloatToStrF(Assessment.GetCrackLengthNew,
                          ffNumber,11,9) + ' m.');
      end;
end;

function TfrmUserIn.GetAxialTension: Real;
begin
  Result := Structure.GetAxialTension;
end;

function TfrmUserIn.GetBending: Real;
begin
  Result := Structure.GetBending;
end;

function TfrmUserIn.GetInPressure: Real;
begin
  Result := Structure.GetCInPressure;
```

```
end;

function TfrmUserIn.GetRIn: Real;
begin
  Result := Structure.Dimension1;
end;

function TfrmUserIn.GetROut: Real;
begin
  Result := Structure.Dimension2;
end;

function TfrmUserIn.LoadStatus: string;
var i       : integer;
    Load    : Real;
    temp    : string;
    LStatus : string;
begin
  Load := 0;
  for i := 0 to 3 do
    begin
      if i = 0 then
        begin
          Load := Structure.GetAxialTension;
          temp := 'ATL';
        end
      else if i = 1 then
        begin
          Load := Structure.GetBending;
          temp := 'BDL';
        end
      else if i = 2 then
        begin
          Load := Structure.GetInPressure;
          temp := 'IPL';
        end
      else if i = 3 then
        begin
          Load := Structure.GetStress_Coeff[0];
          temp := 'SDL';
        end;

      if Load > 0 then
        begin
          LStatus := LStatus + temp;
        end;
    end;
  Result := LStatus;
end;

procedure TfrmUserIn.CalAndShowLevel1_Option;
var resp: word;
    Load: string;
begin
  if ragAssessPos.ItemIndex = 0 then
    lstShow.Items.Add('Parameter K is deepest point')
  else
    lstShow.Items.Add('Parameter K is sureface point');
```

```
lstShow.Items.Add('    PL(ao) : ' +
                  FloatToStrF(Assessment.GetLimitLoad_a0,
                  ffNumber,5,3) + ' MPa');
lstShow.Items.Add('    Lr(ao) : ' +
                  FloatToStrF(Assessment.GetLr_a0,
                  ffNumber,5,3));
lstShow.Items.Add('    KI(ao) : ' +
                  FloatToStrF(Assessment.GetKI_a0,
                  ffNumber,5,3)
                  + ' Pa./m^2');
lstShow.Items.Add('    Kr(ao) : ' +
                  FloatToStrF(Assessment.GetKr_a0,
                  ffNumber,5,3));

Load := LoadStatus;
if load = 'IPL' then // cal crack growth
  begin
    resp := MessageDlg(' Do you want to estimate amount of ' +
                       'flaw growth in service ?',
                        mtConfirmation,[mbOK, mbCancel], 0);
  end
else
  begin
    resp := mrCancel;
  end;

if resp = mrOK then
  begin    // begin resp = OK
    frmCrackGrowth := TfrmCrackGrowth.Create(self);
    frmCrackGrowth.Initial;
    if frmCrackGrowth.Infovalid = true then
      begin      // begin Infovalid = true
        CrackGrowthM;  // crack growth
        Assessment.CalKI(Structure, ragAssessPos.ItemIndex);
        Assessment.CalLimitLoad(Structure);
        Assessment.CalKr(Structure);
        Assessment.CalLr(Structure);
        Assessment.CalAssessPoint(self, Structure);

        lstShow.Items.Add('    PL(ao) : ' +
                          FloatToStrF(Assessment.GetLimitLoad_a0,
                          ffNumber,5,3) + ' MPa');
        lstShow.Items.Add('    Lr(ao) : ' +
                          FloatToStrF(Assessment.GetLr_a0,
                          ffNumber,5,3));
        lstShow.Items.Add('    KI(ao) : ' +
                          FloatToStrF(Assessment.GetKI_a0,
                          ffNumber,5,3) + ' Pa./m^2');
        lstShow.Items.Add('    Kr(ao) : ' +
                          FloatToStrF(Assessment.GetKr_a0,
                          ffNumber,5,3));
        lstShow.Items.Add('    LrC(ao) :' +
                          FloatToStrF(Assessment.GetLrC,
                          ffNumber,5,3));
        lstShow.Items.Add('    KrC(ao) :' +
                          FloatToStrF(Assessment.GetKrC,
                          ffNumber,5,3));
        lstShow.Items.Add('    FL(ao) :' +
                          FloatToStrF(Assessment.GetFL,
```

```
                              ffNumber,5,3));
          lstShow.Items.Add('STRUCTURE IS : ' +
                              Assessment.GetAssessmentResult);
        end     // end Infovalid = true
      else      // Infovalid = false
        begin
          ShowMessage(' You not have assigned crack growth' +
                      ' properties ' + #10#13 + ' Please define crack
                     growth' + ' properties again ');
        end;
    end     // end resp = OK
  else
    begin   // begine resp <> OK
      lstShow.Items.Add('Do not assessment crack growth ');
      Assessment.CalAssessPoint(self, Structure);
      lstShow.Items.Add('    LrC(ao) :' +
                        FloatToStrF(Assessment.GetLrC,
                        ffNumber,5,3));
      lstShow.Items.Add('    KrC(ao) :' +
                        FloatToStrF(Assessment.GetKrC,
                        ffNumber,5,3));
      lstShow.Items.Add('    FL(ao) :' +
                        FloatToStrF(Assessment.GetFL,
                        ffNumber,5,3));
      lstShow.Items.Add('STRUCTURE IS : ' +
                        Assessment.GetAssessmentResult);
    end;   // end resp <> OK
end;

procedure TfrmUserIn.CalAndShowLevel2_Option;
var resp: word;
    Load: string;
begin
  if ragAssessPos.ItemIndex = 0 then
    lstShow.Items.Add('Parameter K is deepest point')
  else
    lstShow.Items.Add('Parameter K is sureface point');
  lstShow.Items.Add('    PL(ao) : ' +
                    FloatToStrF(Assessment.GetLimitLoad_a0,
                    ffNumber,5,3) + ' MPa');
  lstShow.Items.Add('    PL(ag) : ' +
                    FloatToStrF(Assessment.GetLimitLoad_ag,
                    ffNumber,5,3) + ' MPa');
  lstShow.Items.Add('    Lr(ao) : ' +
                    FloatToStrF(Assessment.GetLr_a0,
                    ffNumber,5,3));
  lstShow.Items.Add('    Lr(ag) : ' +
                    FloatToStrF(Assessment.GetLr_ag,
                    ffNumber,5,3));
  lstShow.Items.Add('    KI(ao) : ' +
                    FloatToStrF(Assessment.GetKI_a0,
                    ffNumber,5,3) + ' Pa./m^2');
  lstShow.Items.Add('    KI(ag) : ' +
                    FloatToStrF(Assessment.GetKI_ag,
                    ffNumber,5,3) + ' Pa./m^2');
  lstShow.Items.Add('    Kr(ao) : ' +
                    FloatToStrF(Assessment.GetKr_a02,
                    ffNumber,5,3));
  lstShow.Items.Add('    Kr(ag) : ' +
```

```
                    FloatToStrF(Assessment.GetKr_ag,
                    ffNumber,5,3));
Load := LoadStatus;
if load = 'IPL' then // cal crack growth
  begin
    resp := MessageDlg(' Do you want to estimate amount of ' +
                       'flaw growth in service ?',
                       mtConfirmation,[mbOK, mbCancel], 0);
  end
else
  begin
    resp := mrCancel;
  end;

if resp = mrOK then
  begin                  // begin resp = OK   level 1
    frmCrackGrowth := TfrmCrackGrowth.Create(self);
    frmCrackGrowth.Initial;
    if frmCrackGrowth.Infovalid = true then
      begin      // begin Infovalid = true   level 1
        CrackGrowthM;  // crack grow

        Assessment.CalKI(Structure, ragAssessPos.ItemIndex);
        Assessment.CalLimitLoad(Structure);
        Assessment.CalKr(Structure);
        Assessment.CalLr(Structure);
        Assessment.CalAssessPoint(self, Structure);

        lstShow.Items.Add('    PL(ao) : ' +
                          FloatToStrF(Assessment.GetLimitLoad_a0,
                          ffNumber,5,3) + ' MPa');
        lstShow.Items.Add('    PL(ag) : ' +
                          FloatToStrF(Assessment.GetLimitLoad_ag,
                          ffNumber,5,3) + ' MPa');
        lstShow.Items.Add('    Lr(ao) : ' +
                          FloatToStrF(Assessment.GetLr_a0,
                          ffNumber,5,3));
        lstShow.Items.Add('    Lr(ag) : ' +
                          FloatToStrF(Assessment.GetLr_ag,
                          ffNumber,5,3));
        lstShow.Items.Add('    KI(ao) : ' +
                          FloatToStrF(Assessment.GetKI_a0,
                          ffNumber,5,3) + ' Pa./m^2');
        lstShow.Items.Add('    KI(ag) : ' +
                          FloatToStrF(Assessment.GetKI_ag,
                          ffNumber,5,3) + ' Pa./m^2');
        lstShow.Items.Add('    Kr(ao) : ' +
                          FloatToStrF(Assessment.GetKr_a02,
                          ffNumber,5,3));
        lstShow.Items.Add('    Kr(ag) : ' +
                          FloatToStrF(Assessment.GetKr_ag,
                          ffNumber,5,3));
        lstShow.Items.Add('    LrC(ao) :' +
                          FloatToStrF(Assessment.GetLrC,
                          ffNumber,5,3));
        lstShow.Items.Add('    LrC(ag) :' +
                          FloatToStrF(Assessment.GetLrC_g,
                          ffNumber,5,3));
        lstShow.Items.Add('    KrC(ao) :' +
```

```
                          FloatToStrF(Assessment.GetKrC,
                          ffNumber,5,3));
            lstShow.Items.Add('    KrC(ag) :' +
                          FloatToStrF(Assessment.GetKrC_g,
                          ffNumber,5,3));
            lstShow.Items.Add('    FL(ao) :' +
                          FloatToStrF(Assessment.GetFL,
                          ffNumber,5,3));
            lstShow.Items.Add('    FL(ag) :' +
                          FloatToStrF(Assessment.GetFL_g,
                          ffNumber,5,3));
            lstShow.Items.Add('STRUCTURE IS : ' +
                          Assessment.GetAssessmentResult);

          end       // end Infovalid = true     level 1
        else      // Infovalid = false        level 1
          begin
            ShowMessage(' You not have assigned crack growth' +
                        ' properties ' + #10#13 + ' Please define crack
                       growth' + ' properties again ');
          end;
      end   // end resp = OK      level 1
    else
      begin  // begine resp <> OK    level 1
        lstShow.Items.Add('Do not assessment crack growth ');
        Assessment.CalAssessPoint(self, Structure);

        lstShow.Items.Add('    LrC(ao) :' +
                          FloatToStrF(Assessment.GetLrC,
                          ffNumber,5,3));
        lstShow.Items.Add('    LrC(ag) :' +
                          FloatToStrF(Assessment.GetLrC_g,
                          ffNumber,5,3));
        lstShow.Items.Add('    KrC(ao) :' +
                          FloatToStrF(Assessment.GetKrC,
                          ffNumber,5,3));
        lstShow.Items.Add('    KrC(ag) :' +
                          FloatToStrF(Assessment.GetKrC_g,
                          ffNumber,5,3));
        lstShow.Items.Add('    FL(ao) :' +
                          FloatToStrF(Assessment.GetFL,
                          ffNumber,5,3));
        lstShow.Items.Add('    FL(ag) :' +
                          FloatToStrF(Assessment.GetFL_g,
                          ffNumber,5,3));
        lstShow.Items.Add('STRUCTURE IS : ' +
                          Assessment.GetAssessmentResult);
      end;  // end resp <> OK     level 1
    //end;   // end level = 1
end;


end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.3 ซอร์สโค้ดของฟอร์ม structure

```
unit frmStructureU;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,Dialogs, StdCtrls;

type
  TfrmStructure = class(TForm)
    gpbDimension : TGroupBox;
    lblDimension1: TLabel;
    edtDimension1: TEdit;
    btnOK        : TButton;
    btnClear     : TButton;
    procedure FormCreate(Sender: TObject);
    procedure edtDimension1KeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  protected
    FInfoValid: Boolean;
    function GetDimension1: Real; virtual; abstract;
    function GetDimension2: Real; virtual; abstract;
  public
    { Public declarations }
    procedure Initial; virtual; abstract;
    property  InfoValid: Boolean Read FInfoValid;
    Property  Dimension1: Real Read GetDimension1;
    property  Dimension2: Real Read GetDimension2;
  end;

var
  frmStructure: TfrmStructure;

implementation

{$R *.dfm}

procedure TfrmStructure.FormCreate(Sender: TObject);
begin
  SetWindowLong(edtDimension1.Handle, GWL_ExStyle,
  GetWindowLong(edtDimension1.Handle, GWL_EXStyle)+WS_EX_RIGHT);
end;

procedure TfrmStructure.edtDimension1KeyPress(Sender: TObject;
          var Key: Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

end.
```

ค.4 ซอร์สโค้ดของฟอร์ม pipe

```
unit frmPipeU;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls,  Forms, Dialogs, frmStructureU, StdCtrls;

type
  TfrmPipe = class(TfrmStructure)
    edtDimension2: TEdit;
    lblDimension2: TLabel;
    procedure edtDimension2KeyPress(Sender: TObject; var Key: Char);
    procedure btnOKClick(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
  private
    { Private declarations }
    function CheckBlank(Sender: TObject): string;
  protected
    function GetDimension1: Real; override;
    function GetDimension2: Real; override;
  public
    { Public declarations }
    procedure Initial; override;
  end;

var
  frmPipe: TfrmPipe;

implementation

{$R *.dfm}

procedure TfrmPipe.edtDimension2KeyPress(Sender: TObject; var Key:
Char);
begin
  inherited;
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

function TfrmPipe.GetDimension1: Real;
begin
  Result := StrToFloat(edtDimension1.Text)/1000; // Inner Radius
end;

function TfrmPipe.GetDimension2: Real;
begin
  Result := StrToFloat(edtDimension2.Text)/1000;  // Outer Radius
end;

procedure TfrmPipe.Initial;
```

```
begin
  inherited;
  edtDimension1.Clear;
  edtDimension2.Clear;
  Self.ShowModal;
end;

procedure TfrmPipe.btnOKClick(Sender: TObject);
var text   : string;
    Ro     : Real;
    Ri     : Real;
    t      : Real;
    R      : Real;
    RPerT  : Real;
    RiPerT : Real;
begin
  inherited;
  text := CheckBlank(self);
  if text = '' then
    begin
      FInfoValid := True;
      Ro         := StrToFloat(edtDimension2.Text)/1000;  // Outer
                                                             Radius
      Ri         := StrToFloat(edtDimension1.Text)/1000;  // Inner
                                                             Radius
      t          := Ro - Ri;
      R          := Ri + (t/2);
      RPerT      := R/t;
      RiPerT     := Ri/t;

      if (5 <= RPerT) and (RPerT <= 20) or (5 <= RiPerT) and (RiPerT
         <= 10) then
        begin
          close ;
        end
      else
        begin
          ShowMessage('dimension out of range');
        end;
    end
  else
    begin
      Showmessage('   Please enter :            ' + text);
      Setfocus;
    end;
end;

function TfrmPipe.CheckBlank(Sender: TObject): string;
var i: integer;
    temp, temp1: string;
begin
 for i:= 0 to self.ComponentCount-1 do
    if self.Components[i] is TEdit then
      with self.Components[i] as TEdit do
        begin
          if text = '' then
            begin
              temp1 := copy(self.Components[i].Name,4,15);
              if temp1 = 'Dimension1' then
```

```
                begin
                  temp1 := '  - Inner Radius ';
                end
              else
                begin
                  temp1 := '  - Outer Radius ';
                end;
                  temp := temp + #10#13 + '      '+temp1;
            end
        end;
  Result := temp;
end;

procedure TfrmPipe.btnClearClick(Sender: TObject);
begin
  inherited;
  edtDimension2.Clear;
  edtDimension1.Clear;
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.5 ซอร์สโค้ดของฟอร์ม matProp

```
unit frmMatPropMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,Dialogs, StdCtrls, Mask, DBCtrls, DB,
  DBTables,   Grids, DBGrids;

type
  TfrmMatPropM = class(TForm)
    Label1       : TLabel;
    GroupBox1    : TGroupBox;
    lblSy        : TLabel;
    lblSu        : TLabel;
    lblSE        : TLabel;
    lblAlpha     : TLabel;
    lblExpoN     : TLabel;
    GroupBox2    : TGroupBox;
    chkKC        : TCheckBox;
    chkKIC       : TCheckBox;
    chkK02       : TCheckBox;
    lblSIFUnit   : TLabel;
    DataSource1  : TDataSource;
    TableAssess  : TTable;
    chkKg        : TCheckBox;
    edtSy        : TEdit;
    edtSu        : TEdit;
    edtSE        : TEdit;
    edtAlpha     : TEdit;
    edtEXpoN     : TEdit;
    edtKC        : TEdit;
    edtKIC       : TEdit;
    edtK02       : TEdit;
    edtKg        : TEdit;
    mmDescription: TMemo;
    btninsert    : TButton;
    btnDelete    : TButton;
    btnOK        : TButton;
    cmbName      : TComboBox;
    procedure btnOKClick(Sender    : TObject);
    procedure FormCreate(Sender    : TObject);
    procedure cmbNameChange(Sender : TObject);
    procedure btninsertClick(Sender: TObject);

  private
    { Private declarations }
    FInfoValid: Boolean;
    FLevel    : integer;
    FOption   : integer;
    procedure CheckTensileLvel1_Option1;
    procedure CheckToughnessLevel1_Option1;

  public
    { Public declarations }
```

```
    procedure Initial(ALevel, AOption: integer);
    property  InfoValid: Boolean Read FInfoValid;

    function GetSy       : Real;
    function GetSu       : Real;
    function GetSE       : Real;
    function GetMatName  : string;
    function GetKOption1 : Real;
    function GetKName    : string;
    function GetAlpha    : Real;
    function GetExponentN: Real;
    function GetK02      : Real;
    function GetKg       : Real;
  end;

var
  frmMatPropM: TfrmMatPropM;

implementation

{$R *.dfm}

{ TfrmMatPropM }

procedure TfrmMatPropM.Initial(ALevel, AOption: integer);
var i: integer;
begin
  inherited;
  FLevel := ALevel;
  FOption := AOption;
  for i := 0 to self.ComponentCount -1  do
    begin
      if self.Components[i] is TEdit then
         TEdit(Components[i]).Clear;
    end;
      if ALevel = 0 then  // level 1
        begin
          if AOption = 0 then      // option 1
            begin
              Self.ShowModal;
            end
          else if AOption = 1 then      // option 2
            begin
              Self.ShowModal;
            end;
        end
      else if Alevel = 1 then  // level 2
        begin
          if AOption = 0 then       // option 1
            begin
              Self.ShowModal;
            end
          else if AOption = 1 then  // option 2
            begin
              Self.ShowModal;
            end;
        end;
end;
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
procedure TfrmMatPropM.btnOKClick(Sender: TObject);
begin
 FInfoValid := true;
 Self.Close;
end;

procedure TfrmMatPropM.FormCreate(Sender: TObject);
begin
  cmbName.Clear;
  TableAssess.Open;
  TableAssess.First;

  while not TableAssess.Eof do
    begin
      cmbName.Items.Add(TableAssess.FieldByName('Name').AsString);
      TableAssess.Next;
    end;
    TableAssess.First;
end;

procedure TfrmMatPropM.cmbNameChange(Sender: TObject);
var
  NameOld: string;
  NameNew: string;
begin
  NameOld := cmbName.Text;
  TableAssess.First;
  while not TableAssess.Eof do
    begin
      NameNew := TableAssess.FieldByName('Name').Asstring;
      if NameOld = NameNew then
        begin

          TableAssess.Locate('Name',cmbName.Text,[loCaseInsensitive]
          );
          edtSy.Clear;
          edtSu.Clear;
          edtSE.Clear;
          edtAlpha.Clear;
          edtEXpoN.Clear;
          edtKC.Clear;
          edtKIC.Clear;
          edtK02.Clear;
          edtKg.Clear;
          mmDescription.Clear;

          if FLevel = 0 then // level 1
            begin
              if FOption = 0 then // option = 1
                begin
                 CheckTensileLvel1_Option1;
                 CheckToughnessLevel1_Option1;
                end
              else                  // option = 2
                begin
                  CheckTensileLvel1_Option1;
                  CheckToughnessLevel1_Option1;
                end;
            end
```

```
          else              // level = 2
            begin
              if FOption = 0 then // option = 1
                begin
                 CheckTensileLvel1_Option1;
                 CheckToughnessLevel1_Option1;
                end
              else                  // option = 2
                begin
                  CheckTensileLvel1_Option1;
                  CheckToughnessLevel1_Option1;
                end;
            end;
          Break;
        end
      else
        begin
          edtSy.Clear;
          edtSu.Clear;
          edtSE.Clear;
          edtAlpha.Clear;
          edtEXpoN.Clear;
          edtKC.Clear;
          edtKIC.Clear;
          edtK02.Clear;
          edtKg.Clear;

          chkKC.Checked  := false;
          chkKIC.Checked := false;
          chkK02.Checked := false;
          chkKg.Checked  := false;

        end;
        TableAssess.Next;
    end;
end;
procedure TfrmMatPropM.CheckToughnessLevel1_Option1;
begin
  if TableAssess.fieldByName('KC').AsVariant = Null then
    begin
      chkKC.Checked := false;
      edtKC.Text := '';
    end
  else
    begin
      chkKC.Checked := true;
      edtKC.Text := TableAssess.fieldByName('KC').AsVariant;
    end;

  if TableAssess.FieldByName('KIC').AsVariant = Null then
    begin
      chkKIC.Checked := false;
      edtKIC.Text := '';
    end
  else
    begin
      chkKIC.Checked := true;
      edtKIC.Text   := TableAssess.FieldByName('KIC').AsVariant;
    end;
```

```
  if TableAssess.FieldByName('K02').AsVariant = Null then
    begin
      chkK02.Checked := false;
      edtK02.Text:= '';
    end
  else
    begin
      chkK02.Checked := true;
      edtK02.Text   := TableAssess.FieldByName('K02').AsVariant;
    end;

  if TableAssess.FieldByName('Kg').AsVariant = Null then
    begin
      chkKg.Checked := false;
      edtK02.Text:= '';
    end
  else
    begin
      chkKg.Checked := true;
      edtKg.Text   := TableAssess.FieldByName('Kg').AsVariant;
    end;
end;

procedure TfrmMatPropM.CheckTensileLvel1_Option1;
begin
  if TableAssess.FieldByName('Yield').AsVariant = Null then
    begin
      edtSy.Text := '';
    end
  else
    begin
      edtSy.Text := TableAssess.FieldByName('Yield').AsVariant;
    end;

  if TableAssess.FieldByName('Ultimated').AsVariant = Null then
    begin
      edtSu.Text := '';
    end
  else
    begin
      edtSu.Text := TableAssess.FieldByName('Ultimated').AsVariant;
    end;

  if TableAssess.FieldByName('Young').AsVariant = Null then
    begin
      edtSE.Text := '';
    end
  else
    begin
      edtSE.Text := TableAssess.FieldByName('Young').AsVariant;
    end;
  if TableAssess.FieldByName('Alpha').AsVariant = Null then
    begin
      edtAlpha.Text := '';
    end
  else
    begin
      edtAlpha.Text := TableAssess.FieldByName('Alpha').AsVariant;
```

```
    end;

  if TableAssess.FieldByName('Exponent').AsVariant = Null then
    begin
      edtEXpoN.Text := '';
    end
  else
    begin
      edtEXpoN.Text := TableAssess.FieldByName('Exponent').AsVariant;
    end;
end;

function TfrmMatPropM.GetSy: Real;
begin
  Result := StrToFloat(edtSy.Text) * 1E6;
end;

function TfrmMatPropM.GetSu: Real;
begin
  Result := StrToFloat(edtSu.Text) * 1E6;
end;

function TfrmMatPropM.GetSE: Real;
begin
  Result := StrToFloat(edtSE.Text) * 1E9;
end;

function TfrmMatPropM.GetMatName: string;
begin
  Result := cmbName.Items[cmbName.ItemIndex];
end;

function TfrmMatPropM.GetKOption1: Real;
begin
  if chkKIC.Checked = true then
    begin
      Result := StrToFloat(edtKIC.Text) * 1E6;
    end
  else if chkKC.Checked = true then
    begin
      Result := StrToFloat(edtKC.Text) * 1E6;
    end
  else if chkK02.Checked = true then
    begin
      Result := StrToFloat(edtK02.Text) * 1E6;
    end;
end;

function TfrmMatPropM.GetKName: string;
begin
  if chkKIC.Checked = true then
    begin
      Result := chkKIC.Caption;
    end
  else if chkKC.Checked = true then
    begin
      Result := chkKC.Caption;
    end
  else if chkK02.Checked = true then
```

```
    begin
      Result := chkK02.Caption;
    end;
end;

function TfrmMatPropM.GetAlpha: Real;
begin
  Result := StrToFloat(edtAlpha.Text);
end;

function TfrmMatPropM.GetExponentN: Real;
begin
  Result := StrTofloat(edtEXpoN.Text);
end;

function TfrmMatPropM.GetK02: Real;
begin
  Result := StrToFloat(edtK02.Text) * 1E6;
end;

function TfrmMatPropM.GetKg: Real;
begin
  Result := StrToFloat(edtKg.Text) * 1E6;
end;

procedure TfrmMatPropM.btninsertClick(Sender: TObject);
var
  new: real;
begin
  TableAssess.Last;
  new := TableAssess.FieldByName('ACode').AsFloat;
  TableAssess.Insert;
  TableAssess.FieldByName('ACode').AsFloat :=  new + 1;
  TableAssess.FieldByName('Name').AsString := cmbName.Text;
  TableAssess.FieldByName('Yield').AsFloat := StrToFloat(edtSy.Text);
  TableAssess.FieldByName('Ultimated').AsFloat :=
                        StrToFloat(edtSu.Text);
  TableAssess.FieldByName('Young').AsFloat := StrToFloat(edtSE.Text);
  TableAssess.FieldByName('Alpha').AsFloat :=
                        StrToFloat(edtAlpha.Text);
  TableAssess.FieldByName('Exponent').AsFloat :=
                        StrToFloat(edtEXpoN.Text);
  TableAssess.FieldByName('KC').AsFloat := StrToFloat(edtKC.Text);
  TableAssess.FieldByName('KIC').AsFloat := StrToFloat(edtKIC.Text);
  TableAssess.FieldByName('K02').AsFloat := StrToFloat(edtK02.Text);
  TableAssess.FieldByName('Kg').AsFloat := StrToFloat(edtKg.Text);

  TableAssess.Post;
  FormCreate(self);
  ShowMessage('Insert Compleated');
end;

end.
```

จุฬาลงกรณ์มหาวิทยาลัย

ค.6 ซอร์สโค้ดของฟอร์ม crack

```
unit frmCrackU;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, Spin, ExtCtrls, CharCrackU;

type
  TfrmCrack = class(TForm)
    btnOK              : TButton;
    btnClear           : TButton;
    gpbCrack           : TGroupBox;
    lblNoCrack         : TLabel;
    Label5             : TLabel;
    ragCrackType       : TRadioGroup;
    gpbCrackDimension  : TGroupBox;
    lblLength          : TLabel;
    lblDepth           : TLabel;
    edtLength          : TEdit;
    edtDepth           : TEdit;
    gpbCrackAng        : TGroupBox;
    lblAngPlan         : TLabel;
    lblAngThk          : TLabel;
    edtAngPlan         : TEdit;
    edtAngThk          : TEdit;
    gpbCrackPositon    : TGroupBox;
    lblCrackPosX       : TLabel;
    lblCrackPosY       : TLabel;
    lblCrackPosZ       : TLabel;
    edtPosX            : TEdit;
    edtPosY            : TEdit;
    edtPosZ            : TEdit;
    btnNext            : TButton;
    Memo1              : TMemo;
    spdNOCrack         : TSpinEdit;
    lstCrack           : TListBox;
    gpbSpecial         : TGroupBox;
    lblDepthDelta      : TLabel;
    edtDepthDelta      : TEdit;
    btnCharacterization: TButton;
    ragCharStd: TRadioGroup;
    procedure edtCrackDepthDeltaKeyPress(Sender: TObject;
                                         var Key: Char);
    procedure edtPosXKeyPress           (Sender: TObject;
                                         var Key: Char);
    procedure edtPosYKeyPress           (Sender: TObject;
                                         var Key: Char);
    procedure edtPosZKeyPress           (Sender: TObject;
                                         var Key: Char);
    procedure edtLengthKeyPress         (Sender: TObject;
                                         var Key: Char);
    procedure edtDepthKeyPress          (Sender: TObject;
                                         var Key: Char);
    procedure edtAngPlanKeyPress        (Sender: TObject;
```

```
                                          var Key: Char);
    procedure edtAngThkKeyPress          (Sender: TObject;
                                          var Key: Char);
    procedure btnOKClick                 (Sender: TObject);
    procedure btnClearClick              (Sender: TObject);
    procedure btnNextClick               (Sender: TObject);
    procedure spdNOCrackChange           (Sender: TObject);
    procedure btnCharacterizationClick   (Sender: TObject);
    procedure ragCharStdClick            (Sender: TObject);
  private
    { Private declarations }
    CharCrack  : TCharCrack; // the intermediate class
    FInfovalid : Boolean;
    FCounter   : integer;
    function  CheckBlank(Sender: TObject): string;
    function  GetCrackOriented  : string;
    function  GetCrackPosition  : string;
    function  GetCrackDepth     : Real;
    function  GetCrackLength    : Real;
    function  GetCrackDepthDelta: Real;
    function  GetCrackType      : string;
    procedure UpdateCounter;

    function  CheckLoadStatus: string;
    procedure AssignCrackAndShow;
    procedure Check_LoaddimCrack_Assign;
  public
    { Public declarations }
    procedure Initial(ALevel: integer);
    Property  Infovalid : Boolean Read FInfovalid Write FInfovalid;
    property  Oriented  : string  Read GetCrackOriented;
    property  Position  : string  Read GetCrackPosition;
    property  Depth     : Real    Read GetCrackDepth;
    property  Length    : Real    Read GetCrackLength;
    property  DepthDelta: Real    Read GetCrackDepthDelta;
    property  CrackType : string  Read GetCrackType;
    function  GetPrinStressH: Real;
    function  GetPrinStressA: Real;
  end;

var
  frmCrack: TfrmCrack;

implementation

uses frmUserInU;

{$R *.dfm}

procedure TfrmCrack.Initial(ALevel: integer);
var i: integer;
begin
  for i := 0 to self.ComponentCount -1  do
    begin
      if self.Components[i] is TRadioGroup then
         TRadioGroup(Components[i]).ItemIndex := 0;
      if self.Components[i] is TEdit then
         TEdit(Components[i]).Clear;
    end;
```

```
  if ALevel = 0 then
    begin
      lblDepthDelta.Enabled := false;
      edtDepthDelta.Enabled := false;
      edtDepthDelta.Color   := clBtnFace;
    end
  else
    begin
      lblDepthDelta.Enabled := true;
      edtDepthDelta.Enabled := true;
      edtDepthDelta.Color   := clWhite;
    end;
  CharCrack := TCharCrack.Create;
  self.ShowModal;
end;

procedure TfrmCrack.edtLengthKeyPress(Sender: TObject; var Key:
Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end
end;

procedure TfrmCrack.edtDepthKeyPress(Sender: TObject; var Key: Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmCrack.edtCrackDepthDeltaKeyPress(Sender: TObject;
  var Key: Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmCrack.edtPosXKeyPress(Sender: TObject; var Key: Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmCrack.edtPosYKeyPress(Sender: TObject; var Key: Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
```

```
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmCrack.edtPosZKeyPress(Sender: TObject; var Key: Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmCrack.edtAngPlanKeyPress(Sender: TObject; var Key:
                                       Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmCrack.edtAngThkKeyPress(Sender: TObject; var Key:
                                      Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmCrack.btnOKClick(Sender: TObject);
begin
      FInfoValid := True;
      close ;
end;

procedure TfrmCrack.btnClearClick(Sender: TObject);
var i: integer;
begin
  for i := 0 to self.ComponentCount -1  do
    begin
    if self.Components[i] is TRadioGroup then
       TRadioGroup(Components[i]).ItemIndex := 0;
    if self.Components[i] is TEdit then
       TEdit(Components[i]).Clear;
    end;
end;

function TfrmCrack.CheckBlank(Sender: TObject): string;
var i    : integer;
    temp,
    temp1: string;
begin
  for i:= 0 to self.ComponentCount-1 do
    if self.Components[i] is TEdit then
      with self.Components[i] as TEdit do
```

```
        begin
          if text = '' then
            begin
              if edtDepthDelta.Enabled = true then
              begin
              temp1 := copy(self.Components[i].Name,4,15);
              if temp1 = 'Depth' then
                 temp1 := '  - Crack Depth '
              else if temp1 = 'Length' then
                 temp1 := '  - Crack Length '
              else if temp1 = 'PosX' then
                 temp1 := '  - Position X mean'
              else if temp1 = 'PosY' then
                 temp1 := '  - Position Y mean'
              else if temp1 = 'PosZ' then
                 temp1 := '  - position Z mean'
              else if temp1 = 'AngPlan' then
                 temp1 := '  - Planar Angle '
              else if temp1 = 'AngThk' then
                 temp1 := '  - Thickness Angle'
              else if temp1 = 'DepthDelta' then
                 temp1 := '  - Crack Depth Delta (ag)';
              temp := temp + #10#13 + '      '+temp1;
              end
              else
              begin
              temp1 := copy(self.Components[i].Name,4,15);
              if temp1 = 'Depth' then
                 temp1 := '  - Crack Depth '
              else if temp1 = 'Length' then
                 temp1 := '  - Crack Length '
              else if temp1 = 'PosX' then
                 temp1 := '  - Position X mean'
              else if temp1 = 'PosY' then
                 temp1 := '  - Position Y mean'
              else if temp1 = 'PosZ' then
                 temp1 := '  - position Z mean'
              else if temp1 = 'AngPlan' then
                 temp1 := '  - Planar Angle '
              else if temp1 = 'AngThk' then
                 temp1 := '  - Thickness Angle'
              else if temp1 = 'DepthDelta' then
                 temp1 := '  - Crack Depth Delta (ag)';
              temp := temp + #10#13 + '      '+temp1;
              end;

            end
        end;
  Result := temp;
end;

function TfrmCrack.GetCrackLength: Real;
begin
  Result := CharCrack.GetDim(0)[0]*1E-03;
end;

function TfrmCrack.GetCrackDepth: Real;
begin
  Result := CharCrack.GetDim(0)[1]*1E-03;
```

```
end;

function TfrmCrack.GetCrackDepthDelta: Real;
begin
  Result := (StrToFloat(edtDepthDelta.Text))*1E-03

end;

function TfrmCrack.GetCrackOriented: string;
var Direction: string;
begin
  Direction := CharCrack.GetDiFlaw(0);
  if Direction = 'X' then
     Direction := 'Longitudinal'
  else
     Direction := 'Circumferential';
  Result := Direction;
end;

function TfrmCrack.GetCrackPosition: string;
begin
  Result := 'Internal';  // All crack located on the inner radius
end;


procedure TfrmCrack.btnNextClick(Sender: TObject);
var posZ : Real;
begin
  // Create Crack
  // delete previous and set to nill
  // plus No of Crack but don't more than crack amount in structure

  if spdNOCrack.Value > 0 then
    begin
      posZ := strToFloat(edtPosZ.Text);
      if ragCrackType.ItemIndex = 1 then  // embeded crack type
        begin
          if posZ = 0 then
            begin
              ShowMessage('position Z must be more then ' +
                          'zero and half crack depth');
              edtPosZ.Clear;
              edtPosZ.SetFocus;
            end
          else   // posZ <> 0
            begin
              Check_LoaddimCrack_Assign;
            end;
        end
      else                                  // semi-elliptical crack
                                               type
        begin
          if posZ <> 0 then
            begin
              ShowMessage('position Z must be equal zero');
              edtPosZ.Clear;
              edtPosZ.SetFocus;
            end
          else
```

```
            begin
              Check_LoaddimCrack_Assign;
            end;
        end;
    end
  else
    ShowMessage('Please define total crack amount in box');
end;


procedure TfrmCrack.spdNOCrackChange(Sender: TObject);
begin
  btnNext.Enabled     := True;
end;

procedure TfrmCrack.ragCharStdClick(Sender: TObject);
begin
  if ragCharStd.ItemIndex <> 0 then
    begin
      ShowMessage(ragCharStd.Items[ragCharStd.ItemIndex] +
                  ' standars is not available');
      ragCharStd.ItemIndex := 0;
    end;
end;

procedure TfrmCrack.UpdateCounter;
var i: integer;
begin
  inc(FCounter);
  Memo1.Lines.Delete(0);

  if spdNOCrack.Value = FCounter then     //  one crack or final
                                               crack
    begin
      Memo1.Lines.Text :=  IntToStr(FCounter);
      btnNext.Enabled  := False;
      if spdNOCrack.Value = 1 then
    end
  else if spdNOCrack.Value > FCounter then
    begin
      Memo1.Lines.Text :=  IntToStr(FCounter+1);
      for i := 0 to self.ComponentCount -1  do
        begin
          if self.Components[i] is TRadioGroup then
            TRadioGroup(Components[i]).ItemIndex := 0;
          if self.Components[i] is TEdit then
            TEdit(Components[i]).Clear;
        end;
    end;
end;

procedure TfrmCrack.btnCharacterizationClick(Sender: TObject);
var i: integer;       // Crack NO.
begin
  CharCrack.assignCrackStd(ragCharStd.ItemIndex);
  CharCrack.Reorientation(self);
  CharCrack.Position;
  for i := 0 to FCounter - 1 do
    begin    // begin loop
```

```
      lstCrack.Items.Add('Crack NO. ' + IntToStr(i + 1));
      lstCrack.Items.Add('Type   ' + CharCrack.GetType(i) + '   is
                         rotated');
      lstCrack.Items.Add('Principal Stress is in ' +
                         CharCrack.GetDiPrinS(i) + ' direction');
      lstCrack.Items.Add('Crack is in ' + CharCrack.GetDiFlaw(i) +
                         ' direction');
      lstCrack.Items.Add('Dimension (mm.)');
      lstCrack.Items.Add('Effective Length   : ' +
                          FloatToStrF(CharCrack.GetDim(i)[0],
                         ffNumber,5,3) + ' ,  ' + 'Depth   :' +
                          FloatToStrF(CharCrack.GetDim(i)[1],
                         ffNumber,5,3));

      lstCrack.Items.Add('Crack No ' + IntToStr(i + 1) + ' is rotated
                         to ');
      lstCrack.Items.Add('Planar :' +
                          FloatToStr(CharCrack.GetAng(i)[0]) + ' ,  '
                         + 'Thickness  :' +
                          FloatToStr(CharCrack.GetAng(i)[1]));
      lstCrack.Items.Add('Crack Tip Coordinates: Point 1');
      lstCrack.Items.Add(FloatToStrF(CharCrack.GetPos(i,1)[0],
                         ffNumber,5,3) + ' / ' +
                        FloatToStrF(CharCrack.GetPos(i,1)[1],
                         ffNumber,5,3) + ' / ' +
                        FloatToStrF(CharCrack.GetPos(i,1)[2],
                         ffNumber,5,3));
      lstCrack.Items.Add('Crack Tip Coordinates: Point 2');
      lstCrack.Items.Add(FloatToStrF(CharCrack.GetPos(i,2)[0],
                         ffNumber,5,3) + ' / ' +
                        FloatToStrF(CharCrack.GetPos(i,2)[1],
                         ffNumber,5,3) + ' / ' +
                        FloatToStrF(CharCrack.GetPos(i,2)[2],
                         ffNumber,5,3));
      lstCrack.Items.Add('Crack Tip Coordinates: Point 3');
      lstCrack.Items.Add(FloatToStrF(CharCrack.GetPos(i,3)[0],
                         ffNumber,5,3) + ' / ' +
                        FloatToStrF(CharCrack.GetPos(i,3)[1],
                         ffNumber,5,3) +
                        ' / ' +
                        FloatToStrF(CharCrack.GetPos(i,3)[2],
                         ffNumber,5,3));
      lstCrack.Items.Add('Crack Tip Coordinates: Point 4');
      lstCrack.Items.Add(FloatToStrF(CharCrack.GetPos(i,4)[0],
                         ffNumber,5,3) +
                        ' / ' +
                        FloatToStrF(CharCrack.GetPos(i,4)[1],
                         ffNumber,5,3) +
                        ' / ' +
                        FloatToStrF(CharCrack.GetPos(i,4)[2],
                         ffNumber,5,3));
      lstCrack.Items.Add('-------------------------');
    end;   // end loop
  if FCounter > 1 then
    begin
      // can interaction
    CharCrack.Interaction;
      for i := 0 to CharCrack.NOCrack - 1 do
        begin
```

```
            lstCrack.Items.Add('FINAL CRACK');
            lstCrack.Items.Add('Type   ' + CharCrack.GetType(i));
            lstCrack.Items.Add('Principal Stress is in ' +
                               CharCrack.GetDiPrinS(i) + ' direction');
            lstCrack.Items.Add('Crack is in ' +
                               CharCrack.GetDiFlaw(i) + ' direction');
            lstCrack.Items.Add('Dimension (mm.)');
            lstCrack.Items.Add('final Crack Length   : ' +
                               FloatToStrF(CharCrack.GetDim(i)[0],
                               ffNumber,5,3) + ' ,  ' + 'Depth   :' +
                               FloatToStrF(CharCrack.GetDim(i)[1],
                               ffNumber,5,3));
            lstCrack.Items.Add('Planar :' +
                               FloatToStr(CharCrack.GetAng(i)[0]) +
                               ' ,  ' + 'Thickness  :' +
                               FloatToStr(CharCrack.GetAng(i)[1]));
            lstCrack.Items.Add('Crack Tip Coordinates: Point 1');
            lstCrack.Items.Add(FloatToStrF(CharCrack.GetPos(i,1)[0],
                              ffNumber,5,3)+ ' / ' +
                              FloatToStrF(CharCrack.GetPos(i,1)[1],
                              ffNumber,5,3)+ ' / ' +
                              FloatToStrF(CharCrack.GetPos(i,1)[2],
                              ffNumber,5,3));
            lstCrack.Items.Add('Crack Tip Coordinates: Point 2');

             lstCrack.Items.Add(FloatToStrF(CharCrack.GetPos(i,2)[0],
                              ffNumber,5,3)+ ' / ' +
                              FloatToStrF(CharCrack.GetPos(i,2)[1],
                              ffNumber,5,3)+ ' / ' +
                              FloatToStrF(CharCrack.GetPos(i,2)[2],
                              ffNumber,5,3));
            lstCrack.Items.Add('Crack Tip Coordinates: Point 3');

             lstCrack.Items.Add(FloatToStrF(CharCrack.GetPos(i,3)[0],
                              ffNumber,5,3)+ ' / ' +
                             FloatToStrF(CharCrack.GetPos(i,3)[1],
                             ffNumber,5,3)+ ' / ' +
                             FloatToStrF(CharCrack.GetPos(i,3)[2],
                             ffNumber,5,3));
            lstCrack.Items.Add('Crack Tip Coordinates: Point 4');

             lstCrack.Items.Add(FloatToStrF(CharCrack.GetPos(i,4)[0],
                              ffNumber,5,3)+ ' / ' +
                              FloatToStrF(CharCrack.GetPos(i,4)[1],
                               ffNumber,5,3)+ ' / ' +

                              FloatToStrF(CharCrack.GetPos(i,4)[2],ffN
                              umber,5,3));

            lstCrack.Items.Add('-------------------------');

        end;
      end
    else
      begin
        // can't interaction
        ShowMessage('Can not interaction process becouse crack current
                    is one');
      end;
```

```
  if edtDepthDelta.Enabled = true then
     edtDepthDelta.SetFocus;
end;

function TfrmCrack.GetPrinStressH: Real;
begin
  Result := frmUserIn.GetPrinStressH;
end;

function TfrmCrack.GetPrinStressA: Real;
begin
  Result := frmUserIn.GetPrinStressA;
end;


function TfrmCrack.CheckLoadStatus: string;  // Check status of load
var i       : integer;
    Load    : Real;
    temp    : string;
    LStatus : string;
begin
  Load := 0;
  for i := 0 to 2 do
    begin
      if i = 0 then
        begin
          Load := frmUserIn.GetAxialTension;
          temp := 'ATL';
        end
      else if i = 1 then
        begin
          Load := frmUserIn.GetBending;
          temp := 'BDL';
        end
      else if i = 2 then
        begin
          Load := frmUserIn.GetInPressure;
          temp := 'IPL';
        end;
      if Load > 0 then
        begin
          LStatus := LStatus + temp;
        end;
    end;
  Result := LStatus;
end;

procedure TfrmCrack.AssignCrackAndShow;
var  CrackType : string;
     Position  : array[0..2] of real;
     Dimension : array[0..1] of real;
     Angle     : array[0..1] of real;
     alpha     : Real;
     zeta      : Real;
begin
      alpha      := StrToFloat(edtAngPlan.Text);  // Plannar angle
      zeta       := StrToFloat(edtAngThk.Text);   // Thickness angle
      if alpha <= 90 then
        begin
```

```
          if zeta < 90 then
            begin                                  // total crack amount
              CharCrack.NOCrack := spdNOCrack.Value
                                                   // Crack Type
              CrackType         := ragCrackType.Items
                                   [ragCrackType.ItemIndex];

                                                   //Position X
              Position[0]       :=   StrToFloat(edtPosX.Text);
                                                   // Position y
              Position[1]       :=   StrToFloat(edtPosY.Text);
              if ragCrackType.ItemIndex = 0 then
              Position[2]  :=    StrToFloat(edtPosZ.Text)
                                                   //Position Z of

//semi-ellip
              else
              Position[2]  :=  - StrToFloat(edtPosZ.Text);
                                                  // Position Z of

//embeded

              Dimension[0] :=   StrToFloat(edtLength.Text);
                                                  // Crack Length
              Dimension[1] :=   StrToFloat(edtDepth.Text);
                                                  // Crack Depth
              Angle[0]     :=   alpha;
                                                  // Plannar angle
              Angle[1]     :=   zeta;
                                                  // Thickness angle

// alpha angle, crack length and crack depth must be positive only

              CharCrack.SetCrack(FCounter, CrackType,
                                 Position, Dimension, Angle);
              lstCrack.Items.Add('Crack NO. ' +
                                 IntToStr(FCounter +         1) +
                                 ' is created');
              lstCrack.Items.Add('-------------------------');
              lstCrack.Items.Add('Type : ' +
                                 CharCrack.GetType
                                 (FCounter));
              lstCrack.Items.Add('Location:' );
              lstCrack.Items.Add('X : ' +
                                 FloatToStr(CharCrack.GetPos(FCounter
                                 ,3)[0]) + ' ,   ' + 'Y : ' +

                                 FloatToStr(CharCrack.GetPos(FCounter
                                 ,3)[1])+
                                 ' ,   ' +
                                 'Z : ' +

                                 FloatToStr(CharCrack.GetPos(FCounter
                                 ,3)[2]));
              lstCrack.Items.Add('Dimension (mm.)');
              lstCrack.Items.Add('Length   : ' +

                                 FloatToStr(CharCrack.GetDim(FCounter
                                 )[0]) +
```

```
                                ' ,  ' + 'Depth   :' +

                                FloatToStr(CharCrack.GetDim(FCounter
                                )[1]));
                lstCrack.Items.Add('Orientation (deg.)');
                lstCrack.Items.Add('Planar :' +

                                FloatToStr(CharCrack.GetAng(FCounter
                                )[0]) +
                                ' ,  ' + 'Thickness  :' +

                                FloatToStr(CharCrack.GetAng(FCounter
                                )[1]));
                lstCrack.Items.Add('-------------------------');
                UpdateCounter;
              end
            else
              begin
                ShowMessage('you much be enter zeta less then 90
                            degree');
                edtAngThk.Clear;
              end;
          end
        else
          begin
            ShowMessage('program is available for alpha less then or
                        equal 90' + 'degree');
            edtAngPlan.Clear;
          end;
end;

procedure TfrmCrack.Check_LoaddimCrack_Assign;
var  alpha     : Real;
     LoadStatus: string;
     ROut      : Real;
     RIn       : Real;
     thickness : Real;
     twoC      : Real;
     a         : Real;
     APerT     : Real;
     TwoCPerA  : Real;
     APerC     : Real;
     // check load status and alpha angle
        {   5 <= R/t <= 160}
        {0.05 <= a/t <= 0.8}
        {   3 <= 2c/a}          // for axial tension  TUB_SECDI_AT

        {   5 <= R/t <= 160}
        {0.05 <= a/t <= 0.8}
        {   3 <= 2c/a}          // for bending moment  TUB_SECDI_BD

        {   1 <= Ri/t <= 10}
        {0.05 <= a/t <= 0.85}
        { 0.1 <= a/c <= 1}      // for internal pressure

begin
  ROut       := (frmUserIn.GetROut) * 1000 ;
  RIn        := (frmUserIn.GetRIn) * 1000;
  thickness  := ROut - RIn;
```

```
LoadStatus := CheckLoadStatus;
alpha      := StrToFloat(edtAngPlan.Text);    // Plannar angle
twoC       := StrToFloat(edtLength.Text);
a          := StrToFloat(edtDepth.Text);
TwoCPerA   := twoC/a;
AperT      := a/thickness;
AperC      := a/(twoC/2);

if (LoadStatus = 'ATL')    or  (LoadStatus = 'BDL') or
   (LoadStatus = 'ATLBDL') then
  begin
    if alpha <> 0 then  // can do
      begin             //check leng per depth
        if  3 <= TwoCPerA then
          begin         //check depth per thickness
            if (0.05 <= AperT) and (AperT <= 0.8) then
              begin
                AssignCrackAndShow;
              end
            else  // a/t out of range
              begin
                ShowMessage('range of  0.05 <= a/t <= 0.8 ' +
                            #13#10 +
                            'a/t = ' + FloatToStr(AperT) +
                            ' out of range');
                edtLength.Clear;
                edtLength.SetFocus;
              end;
          end
        else  // 2c/a out of range
          begin
            ShowMessage('range of 2c/a >= 3' + #13#10 +
                        '2c/a = ' + FloatToStr(TwoCPerA) +
                        ' out of range');
            edtLength.Clear;
            edtLength.SetFocus;
          end;
      end
    else              //  alpha = 0 can not
      begin
        ShowMessage('alpha must be not equal zero' + #13#10 +
                    'solution not available');
        edtAngPlan.Clear;
        edtAngPlan.SetFocus;
      end;
  end
else  if LoadStatus = 'IPL' then
                              // LoadStatus = ATLIPL, BDLIPL, IPL
  begin
    if (0.05 <= AperT) and (AperT <= 0.85) then
      begin
        if (0.1 <= AperC) and (AperC <= 1) then
          begin
            AssignCrackAndShow;
          end
        else
          begin
            ShowMessage('range of 0.1 <= a/c <= 1' + #13#10 +
                        '  a/c = ' + FloatToStr(AperC) +
```

```
                              ' out of range');
            end;
        end
      else
        begin
          ShowMessage('range of  0.05 <= a/t <= 0.85 ' + #13#10 +
                      '  a/t = ' + FloatToStr(AperT) +
                      ' out of range');
        end;
    end
  else  // Load combine
    begin
      if (0.05 <= AperT) and (AperT <= 0.8) then
        begin
          if (TwoCperA >= 3) or (0.1 <= AperC) and (AperC <= 1) then
            begin
              AssignCrackAndShow;
            end
          else
            begin
              ShowMessage('range of 2c/a >= 3' + #13#10 +
                          '  2c/a = ' + FloatToStr(TwoCperA) +
                          'range of 0.1 <= a/c <= 1' + #13#10 +
                          '  a/c = ' + FloatToStr(AperC) +
                          ' out of range');
            end;
        end
      else
        begin
          ShowMessage('range of  0.05 <= a/t <= 0.8 ' + #13#10 +
                      '  a/t = ' + FloatToStr(AperT) +
                      ' out of range');
        end;
    end;
end;

function TfrmCrack.GetCrackType: string;
begin
  Result := CharCrack.GetType(0);
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.7 ซอร์สโค้ดของฟอร์ม crackGrowth

```
unit frmCrackGrowthU;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, ExtCtrls;

type
  TfrmCrackGrowth = class(TForm)
    ragCrackGrowth: TRadioGroup;
    gpbFATIGUE    : TGroupBox;
    lblLoadMax    : TLabel;
    lblLoadMin    : TLabel;
    lblC          : TLabel;
    lblM          : TLabel;
    lblNoCycle    : TLabel;
    lblEquation   : TLabel;
    lblLoadType   : TLabel;
    edtLoadMax    : TEdit;
    edtLoadMin    : TEdit;
    cmbEquation   : TComboBox;
    edtNoCycle    : TEdit;
    edtC          : TEdit;
    edtM          : TEdit;
    cmbLoadType   : TComboBox;
    btnOK         : TButton;
    btnCancel     : TButton;
    procedure btnOKClick        (Sender: TObject);
    procedure btnCancelClick    (Sender: TObject);
    procedure edtLoadMaxKeyPress(Sender: TObject; var Key: Char);
    procedure edtLoadMinKeyPress(Sender: TObject; var Key: Char);
    procedure edtNoCycleKeyPress(Sender: TObject; var Key: Char);
    procedure edtCKeyPress      (Sender: TObject; var Key: Char);
    procedure edtMKeyPress      (Sender: TObject; var Key: Char);
  private
    { Private declarations }
    FInfovalid : Boolean;
    function  CheckBlank(Sender: TObject): string;
    function GetMechanism: integer;
    function GetNOCycle  : integer;
    function GetEquation : integer;
    function GetLoadMax  : Real;
    function GetLoadMin  : Real;
    function GetCoffc    : Real;
    function GetCoffm    : Real;
  public
    { Public declarations }
    procedure Initial;
    Property  Infovalid: Boolean Read FInfovalid Write FInfovalid;
    property  Mechanism: integer Read GetMechanism;
    property  NOCycle:   integer Read GetNOCycle;
    property  Equation:  integer Read GetEquation;
    property  LoadMax:      Real Read GetLoadMax;
```

```
    property  LoadMin:      Real Read GetLoadMin;
    property  CoffC:        Real Read GetCoffc;
    property  CoffM:        Real Read GetCoffm;
  end;

var
  frmCrackGrowth: TfrmCrackGrowth;

implementation

{$R *.dfm}

{ TfrmCrackGrowth }

procedure TfrmCrackGrowth.Initial;
var i: integer;
begin
  for i := 0 to self.ComponentCount -1  do
    begin
      if self.Components[i] is TRadioGroup then
         TRadioGroup(Components[i]).ItemIndex := 0;
      if self.Components[i] is TEdit then
         TEdit(Components[i]).Clear;
    end;
  self.ShowModal;
end;

procedure TfrmCrackGrowth.btnOKClick(Sender: TObject);
begin
      FInfoValid := True;
      close ;
end;

procedure TfrmCrackGrowth.btnCancelClick(Sender: TObject);
var i: integer;
begin
  for i := 0 to self.ComponentCount -1  do
    begin
    if self.Components[i] is TRadioGroup then
       TRadioGroup(Components[i]).ItemIndex := 0;
    if self.Components[i] is TComboBox then
       TColorBox(Components[i]).ItemIndex := 0;
    if self.Components[i] is TEdit then
      TEdit(Components[i]).Clear;
    end;
end;

function TfrmCrackGrowth.CheckBlank(Sender: TObject): string;
var i    : integer;
    temp,
    temp1: string;
begin
  for i:= 0 to self.ComponentCount-1 do
    if self.Components[i] is TEdit then
      with self.Components[i] as TEdit do
        begin
          if text = '' then
            begin
              temp1 := copy(self.Components[i].Name,4,15);
```

```
              if temp1 = 'LoadMax' then
                 temp1 := '  - Load Max'
              else if temp1 = 'LoadMin' then
                 temp1 := '  - Load Min'
              else if temp1 = 'NOCycle' then
                 temp1 := '  - Number of Cycle'
              else if temp1 = 'C' then
                 temp1 := '  - Cofficient C'
              else if temp1 = 'M' then
                 temp1 := '  - Cofficient m';
              temp := temp + #10#13 + '      '+temp1;
            end
        end;
  Result := temp;
end;

procedure TfrmCrackGrowth.edtLoadMaxKeyPress(Sender: TObject; var
Key: Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmCrackGrowth.edtLoadMinKeyPress(Sender: TObject; var
Key: Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmCrackGrowth.edtNoCycleKeyPress(Sender: TObject; var
Key: Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmCrackGrowth.edtCKeyPress(Sender: TObject; var Key:
Char);
begin
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

procedure TfrmCrackGrowth.edtMKeyPress(Sender: TObject; var Key:
Char);
begin
```

```
  if not (key in ['0'..'9','.',#8,#13]) then
    begin
      key := #0;
      ShowMessage('Please enter real number only');
    end;
end;

function TfrmCrackGrowth.GetMechanism: integer;
begin
  Result := ragCrackGrowth.ItemIndex;
end;

function TfrmCrackGrowth.GetNOCycle: integer;
begin
  Result := StrToInt(edtNoCycle.Text);
end;

function TfrmCrackGrowth.GetEquation: integer;
begin
  Result := cmbEquation.ItemIndex;
end;

function TfrmCrackGrowth.GetLoadMax: Real;
begin
  Result := StrToFloat(edtLoadMax.Text);
end;

function TfrmCrackGrowth.GetLoadMin: Real;
begin
  Result := StrToFloat(edtLoadMin.Text);
end;

function TfrmCrackGrowth.GetCoffc: Real;
begin
  Result := StrToFloat(edtC.Text);
end;

function TfrmCrackGrowth.GetCoffm: Real;
begin
  Result := StrToFloat(edtM.Text);
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.8 ซอร์สโค้ดของคลาส structureFactory

```
unit StructureFactoryU;

interface

Uses  StructureU;

Type
  TStructureFactory = class(TObject)
  public
    function CreatStructure (ASelectStructure: integer): TStructure;
                              virtual; abstract;
  end;    // end TStructureFactory = class(TObject)

  TStructureCreator = class(TStructureFactory)
  public
    function CreatStructure (ASelectStructure: integer): TStructure;
                             override;
  end;     // end TStructureCreator = class(TStructureFactory)

implementation

{ TStructureCreator }

function TStructureCreator.CreatStructure(ASelectStructure: integer):
                                          TStructure;
begin
  case ASelectStructure of
  0: Result := TPipe.Create;
  else
     Result := nil;
  end;
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.9 ซอร์สโค้ดของคลาส structure

```
unit StructureU;

interface

uses

CrackU, LoadU, MatPropU;

type
  TStructure = class(TObject)
  private
  protected
    CrackTypeObj : TCrack;
    LoadObj      : TLoad;
    MatProp      : TMatProp;
    function  GetStructureType: string;
    function  GetDimension1   : Real; virtual; abstract;
                                                  // Internal Radius
    function  GetDimension2   : Real; virtual; abstract;
                                                  // Outer Radius
  public
    property  StructureName   : string Read GetStructureType;

///---------------------------Dimension----------------------------///
    property  Dimension1      : Real Read GetDimension1;
    property  Dimension2      : Real Read GetDimension2;
    procedure SetDimension(ADimension1, ADimension2: Real); virtual;
                                                            abstract;

///----------------------Material Properties---------------------///
    procedure AssignMatProp(AMatName: string; ASy, ASu, ASE, Aalpha,
                            AExponentN: Real);
    procedure SetToughness(AKmat, AK02, AKg      : Real);
    Procedure SetCGrowthProp(ACoeffC, AExponentM : Real);

    function  GetMatName  : string;
    function  GetSy       : Real;
    function  GetSu       : Real;
    function  GetSE       : Real;
    function  GetAlpha    : Real;
    function  GetExponentN: Real;

    function  GetKmat     : Real;
    function  GetK02      : Real;
    function  GetKg       : Real;

    function  GetCoeffC   : Real;
    function  GetExponentM: Real;

///---------------------------Crack-----------------------------///
    // CC : Current Crack
    // OC : Original Crack
    procedure AssignCrack(ACrackType: integer; ALength, ADepth: Real;
                          AOrientation, APosition: string); virtual;
                          abstract;
```

```
    procedure SetCCDepth (ADepth    : Real); virtual; abstract;
    procedure SetCCLength(ALength   : Real); virtual; abstract;

    function  GetCrackType   : string; virtual; abstract;
    function  GetOCLength    : Real;   virtual; abstract;
    function  GetCCLength    : Real;   virtual; abstract;
    function  GetOCDepth     : Real;   virtual; abstract;
    function  GetCCDepth     : Real;   virtual; abstract;
    function  GetOrientation : string; virtual; abstract;
    function  GetPosition    : string; virtual; abstract;

///---------------------------Loads-----------------------------///
    procedure AssignLoads(AnAxialTension, ABending, AInPressure:
                          Real); virtual; abstract;
    procedure SetStress_Coeff(AStress_Coeff: array of Real);
                                                  virtual; abstract;
    procedure SetFatigueLoad(AMax, AMin    : Real);
                                                  virtual; abstract;
    procedure SetCInPressure(AValue        : Real);
                                                  virtual; abstract;

    function  GetAxialTension  : Real;          virtual; abstract;
    function  GetBending       : Real;          virtual; abstract;
    function  GetInPressure    : Real;          virtual; abstract;
    function  GetStress_Coeff  : TStress_Coeff; virtual; abstract;

    function  GetInPressureMax : Real;          virtual; abstract;
    function  GetInPressureMin : Real;          virtual; abstract;

    function  GetCInPressure   : Real;          virtual; abstract;
    function  GetPrinStressA   : Real;          virtual; abstract;
    function  GetPrinStressH   : Real;          virtual; abstract;
    destructor Destroy;                         override;
  end;   // end TStructure = class(TObject)

  TPipe = class(TStructure)
  private
    FInnerRadius: Real;
    FOuterRadius: Real;
  protected
    function GetDimension1: Real; override;     // Internal Radius
    function GetDimension2: Real; override;     // Outer Radius
  public
    procedure SetDimension(ADimension1, ADimension2: Real); override;

///---------------------------Crack-----------------------------///
    procedure AssignCrack(ACrackType: integer; ALength, ADepth: Real;
                          AOrientation, APosition: string); override;
    procedure SetCCDepth(ADepth  : Real);                   override;
    procedure SetCCLength(ALength: Real);                   override;

    function  GetCrackType   : string; override;
    function  GetOCLength    : Real;   override;
    function  GetCCLength    : Real;   override;
    function  GetOCDepth     : Real;   override;
    function  GetCCDepth     : Real;   override;
    function  GetOrientation : string; override;
    function  GetPosition    : string; override;
```

```
///----------------------------Loads--------------------------///
    procedure AssignLoads(AnAxialTension, ABending, AInPressure:
Real);override;
    procedure SetStress_Coeff(AStress_Coeff: array of Real);
override;
    procedure SetFatigueLoad(AMax, AMin    : Real);
override;
    procedure SetCInPressure(AValue        : Real);
override;

    function  GetAxialTension  : Real;          override;
    function  GetBending       : Real;          override;
    function  GetInPressure    : Real;          override;
    function  GetStress_Coeff  : TStress_Coeff; override;

    function  GetInPressureMax : Real;          override;
    function  GetInPressureMin : Real;          override;

    function  GetCInPressure   : Real;          override;

    function  GetPrinStressA   : Real;          override;
    function  GetPrinStressH   : Real;          override;
  end;    //end TPipe = class(TStructure)

implementation

uses  SysUtils, Math;

{TStructure}

function TStructure.GetStructureType: string;
begin
  Result := Self.ClassName;
  Delete(Result,1,1);
end;

destructor TStructure.Destroy;
begin
  CrackTypeObj.Free;
  LoadObj.Free;
  MatProp.Free;
  inherited;
end;

///----------------------Material Properties-------------------///
procedure TStructure.AssignMatProp(AMatName: string;
                                   ASy, ASu, ASE, Aalpha, AExponentN:
                                   Real);
begin
  FreeAndNil(MatProp);
  MatProp := TMatProp.Create(AMatName, ASy, ASu, ASE, Aalpha,
                             AExponentN);
end;

function TStructure.GetMatName: string;
begin
  if assigned (MatProp) then
    Result := MatProp.GetMatName
  else
```

```
    Result := '';
end;
function TStructure.GetK02: Real;
begin
  if assigned (MatProp) then
    Result := MatProp.GetK02
  else
    Result := 0;
end;

function TStructure.GetSu: Real;
begin
  if assigned (MatProp) then
    Result := MatProp.GetSu
  else
    Result := 0;
end;

function TStructure.GetSy: Real;
begin
  if assigned (MatProp) then
    Result := MatProp.GetSy
  else
    Result := 0;
end;

function TStructure.GetSE: Real;
begin
  if assigned (MatProp) then
    Result := MatProp.GetSE
  else
    Result := 0;
end;

function TStructure.GetAlpha: Real;
begin
  if assigned (MatProp) then
    Result := MatProp.Getalpha
  else
    Result := 0;
end;

function TStructure.GetExponentN: Real;
begin
  if assigned (MatProp) then
    Result := MatProp.GetExponentN
  else
    Result := 0;
end;

procedure TStructure.SetToughness(AKmat, AK02, AKg: Real);
begin
  MatProp.SetToughness(AKmat, AK02, AKg);
end;

function TStructure.GetKg: Real;
begin
  if assigned (MatProp) then
    Result := MatProp.GetKg
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
  else
    Result := 0;
end;

function TStructure.GetKmat: Real;
begin
  if assigned (MatProp) then
    Result := MatProp.GetKmat
  else
    Result := 0;
end;

procedure TStructure.SetCGrowthProp(ACoeffC, AExponentM: Real);
begin
  MatProp.SetCGrowthProp(ACoeffC,AExponentM);
end;

function TStructure.GetCoeffC: Real;
begin
  if assigned(MatProp) then
    Result := MatProp.GetCoeffC
  else
    Result := 0;
end;

function TStructure.GetExponentM: Real;
begin
  if assigned(MatProp) then
    Result := MatProp.GetExponentM
  else
    Result := 0;
end;

{ TPipe }

///-------------------------Dimension----------------------------///
procedure TPipe.SetDimension(ADimension1, ADimension2: Real);
begin
  inherited;
  FInnerRadius := ADimension1;
  FOuterRadius := ADimension2;
end;

function TPipe.GetDimension1: Real;
begin
  Result := FInnerRadius;
end;

function TPipe.GetDimension2: Real;
begin
  Result := FOuterRadius;
end;

///-------------------------Crack-------------------------------///
procedure TPipe.AssignCrack(ACrackType: integer; ALength, ADepth:
                            Real; AOrientation, APosition: string);
begin
  inherited;
  FreeAndNil(CrackTypeObj);
```

```
  case ACrackType of
    0: CrackTypeObj := TSemiEllCrack.Create(ALength, ADepth,
                                            AOrientation, APosition);
  else
      CrackTypeObj := nil;
  end;
end;

procedure TPipe.SetCCDepth(ADepth: Real);
begin
  inherited;
  CrackTypeObj.SetCCDepth(ADepth);
end;

procedure TPipe.SetCCLength(ALength: Real);
begin
  inherited;
  CrackTypeObj.SetCCLength(ALength);
end;

function TPipe.GetCrackType: string;
begin
  if assigned (CrackTypeObj) then
    Result := CrackTypeObj.GetCrackName
  else
    Result := '';
end;

function TPipe.GetOCDepth: Real;
begin
  if assigned (CrackTypeObj) then
    Result := CrackTypeObj.GetOCDepth
  else
    Result := 0;
end;

function TPipe.GetOCLength: Real;
begin
  if assigned (CrackTypeObj) then
    Result := CrackTypeObj.GetOCLength
  else
    Result := 0;
end;

function TPipe.GetOrientation: string;
begin
  if assigned (CrackTypeObj) then
    Result := CrackTypeObj.GetOriented
  else
    Result := '';
end;

function TPipe.GetPosition: string;
begin
  if assigned (CrackTypeObj) then
    Result := CrackTypeObj.GetPosition
  else
    Result := '';
end;
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
function TPipe.GetCCDepth: Real;
begin
  if assigned(CrackTypeObj) then
    Result := CrackTypeObj.GetCCDepth
  else
    Result := 0;
end;

function TPipe.GetCCLength: Real;
begin
  if assigned(CrackTypeObj) then
    Result := CrackTypeObj.GetCCLength
  else
    Result := 0;
end;

///------------------------------Loads---------------------------///
procedure TPipe.AssignLoads(AnAxialTension, ABending, AInPressure:
                            Real);
begin
  inherited;
  FreeAndNil(LoadObj);
  LoadObj := TLoad.Create(AnAxialTension, ABending, AInPressure);
end;

function TPipe.GetAxialTension: Real;
begin
  if assigned (LoadObj) then
    Result := LoadObj.GetAxialTension
  else
    Result := 0;
end;

function TPipe.GetBending: Real;
begin
  if assigned (LoadObj) then
    Result := LoadObj.GetBending
  else
    Result := 0;
end;

function TPipe.GetInPressure: Real;
begin
  if assigned (LoadObj) then
    Result := LoadObj.GetInPressure
  else
    Result := 0;
end;

procedure TPipe.SetStress_Coeff(AStress_Coeff: array of Real);
begin
  inherited;
  LoadObj.SetStress_Coeff(AStress_Coeff);
end;

function TPipe.GetStress_Coeff: TStress_Coeff;
var Coeff: TStress_Coeff;
begin
  if assigned(LoadObj) then
```

```
    begin
      Coeff := LoadObj.GetStress_Coeff;
      Result := Coeff;
    end;
end;

procedure TPipe.SetFatigueLoad(AMax, AMin: Real);
begin
  inherited;
  LoadObj.SetFatigueLoad(AMax, AMin);
end;

function TPipe.GetInPressureMax: Real;
begin
  if assigned(LoadObj) then
    Result := LoadObj.GetInPressureMax
  else
    Result := 0;
end;

function TPipe.GetInPressureMin: Real;
begin
  if assigned(LoadObj) then
    Result := LoadObj.GetInPressureMin
  else
    Result := 0;
end;

procedure TPipe.SetCInPressure(AValue: Real);
begin
  inherited;
  loadObj.SetCInPressure(AValue);
end;

function TPipe.GetCInPressure: Real;
begin
  if assigned(LoadObj) then
    Result := LoadObj.GetCInPressure
  else
    Result := 0;
end;

function TPipe.GetPrinStressA: Real;
var t                          : Real;
    Tension, Bending, Pressure : Real;
    SigmaT, SigmaB, SigmaP     : Real;
begin
  t       := FOuterRadius - FInnerRadius;
  Tension  := LoadObj.GetAxialTension;
  Bending  := LoadObj.GEtBending;
  Pressure := LoadObj.GetInPressure;

  SigmaT   := Tension/(Pi*(sqr(FOuterRadius) - sqr(FInnerRadius)));
  SigmaB   := Bending*FOuterRadius/(Pi*0.25*(power(FOuterRadius,4) -
              power(FInnerRadius,4)));
  SigmaP   := Pressure*FInnerRadius/(2*t);

  Result   := sigmaT + sigmaB + sigmaP;
end;
```

```
function TPipe.GetPrinStressH: Real;
var t       : Real;
    Pressure: Real;
    SigmaP  : Real;
begin
  t        := FOuterRadius - FInnerRadius;
  Pressure := LoadObj.GetInPressure;
  SigmaP   := Pressure*FInnerRadius/(t);
  Result   := SigmaP;
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.10 ซอร์สโค้ดของคลาส crack

```
unit CrackU;

interface

type
  TCrack = class(TObject)

  protected
    FOCLength : Real;                    // Original crack length
    FCCLength : Real;                    // Current crack length
    FOCDepth  : Real;                    // Original crack depth
    FCCDepth  : Real;                    // Current crack depth

    FOriented : string;
    FPosition : string;

    property  OCLength    : Real   read FOCLength write FOCLength;
    property  CCLength    : Real   read FCCLength write FCCLength;
    property  OCDepth     : Real   read FOCDepth  write FOCDepth;
    property  CCDepth     : Real   read FCCDepth  write FCCDepth;
    property  Oriented    : string read FOriented write FOriented;
    property  Position    : string read FPosition write FPosition;

  public
    procedure SetCCLength(ALengthNew: Real); virtual; abstract;
    procedure SetCCDepth(ADepthNew  : Real); virtual; abstract;

    function  GetCrackName: string;
    function  GetOCLength : Real;   virtual; abstract;
    function  GetCCLength : Real;   virtual; abstract;
    function  GetOCDepth  : Real;   virtual; abstract;
    function  GetCCDepth  : Real;   virtual; abstract;
    function  GetOriented : string; virtual; abstract;
    function  GetPosition : string; virtual; abstract;
  end;  //end TCrack = class(TObject)

  TSemiEllCrack = class(TCrack)
  protected
  public
    Constructor Create (ALength, ADepth: Real; AOrientation,
                        APosition: string);
    function  GetOCLength : Real;   override;
    function  GetCCLength : Real;   override;
    function  GetOCDepth  : Real;   override;
    function  GetCCDepth  : Real;   override;
    function  GetOriented : string; override;
    function  GetPosition : string; override;

    procedure SetCCLength(ALengthNew : Real); override;
    procedure SetCCDepth(ADepthNew   : Real); override;
  end;

implementation

{ TCrackType }
```

```
function TCrack.GetCrackName: string;
begin
  Result := Self.ClassName;
  Delete (Result,1,1);
end;

{ TSemiEllCrack }

constructor TSemiEllCrack.Create(ALength, ADepth: Real; AOrientation,
APosition: string);
begin
  OCLength := ALength;
  CCLength := ALength;
  OCDepth  := ADepth;
  CCDepth  := ADepth;
  Oriented := AOrientation;
  Position := APosition;
end;

function TSemiEllCrack.GetCCDepth: Real;
begin
  Result := CCDepth;
end;

function TSemiEllCrack.GetCCLength: Real;
begin
  Result := CCLength;
end;

function TSemiEllCrack.GetOCDepth: Real;
begin
  Result := OCDepth;
end;

function TSemiEllCrack.GetOCLength: Real;
begin
  Result := OCLength;
end;

function TSemiEllCrack.GetOriented: string;
begin
  Result := Oriented;
end;

function TSemiEllCrack.GetPosition: string;
begin
  Result := Position;
end;

procedure TSemiEllCrack.SetCCDepth(ADepthNew: Real);
begin
  inherited;
  CCDepth := ADepthNew;
end;

procedure TSemiEllCrack.SetCCLength(ALengthNew: Real);
begin
  inherited;
```

```
  CCLength := ALengthNew;
end;
end.
```


ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.11 ซอร์สโค้ดของคลาส load

```
unit LoadU;

interface

type
  TStress_Coeff = array [0..3] of real;

  TLoad = class(TObject)
  private
    FAxialTension : Real;
    FBending      : Real;
    FInPressure   : Real;
    FInPressureMax: Real;
    FInPressureMin: Real;
    FCInPressure  : Real;
  public
    Constructor Create (AnAxialTension, ABending, AInPressure: Real);
    procedure SetStress_Coeff(AStress_Coeff: array of Real);
    procedure SetFatigueLoad(AMax, AMin    : Real);
    procedure SetCInPressure(AValue        : Real);

    function  GetAxialTension  : Real;
    function  GetBending       : Real;
    function  GetInPressure    : Real;
    function  GetStress_Coeff  : TStress_Coeff;
    function  GetInPressureMax : Real;
    function  GetInPressureMin : Real;
    function  GetCInPressure   : Real;
  end;
var
Stress_Coeff: TStress_Coeff;
implementation

{ TLoad }

constructor TLoad.Create(AnAxialTension, ABending, AInPressure:
Real);
begin
  FAxialTension := AnAxialTension;
  FBending      := ABending;
  FInPressure   := AInPressure;
  FCInPressure  := AInPressure;
end;

function TLoad.GetAxialTension: Real;
begin
  Result := FAxialTension;
end;

function TLoad.GetBending: Real;
begin
  Result := FBending;
end;

function TLoad.GetCInPressure: Real;
```

```
begin
  Result := FCInPressure;
end;

function TLoad.GetInPressure: Real;
begin
  Result := FInPressure;
end;

function TLoad.GetInPressureMax: Real;
begin
  Result := FInPressureMax;
end;

function TLoad.GetInPressureMin: Real;
begin
  Result := FInPressureMin;
end;

function TLoad.GetStress_Coeff: TStress_Coeff;
begin
  Result := Stress_Coeff;
end;

procedure TLoad.SetCInPressure(AValue: Real);
begin
  FCInPressure := AValue;
end;

procedure TLoad.SetFatigueLoad(AMax, AMin: Real);
begin
  FInPressureMax := AMax;
  FInPressureMin := AMin;
end;

procedure TLoad.SetStress_Coeff(AStress_Coeff: array of Real);
var i: integer;
begin
 for i := 0 to 3 do
   Stress_Coeff[i] := AStress_Coeff[i];
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.12 ซอร์สโค้ดของคลาส matProp

```
unit MatPropU;

interface

type
  TMatProp = class(TObject)
  private
    FMatName  : string;
    FSy       : Real;
    FSu       : Real;
    FSE       : Real;
    FAlpha    : Real;
    FExponentN: Real;
    FKmat     : Real;
    FK02      : Real;
    FKg       : Real;
    FCoeffC   : Real;
    FExponentM: Real;
  public
    constructor Create (AMatName: string; ASy, ASu, ASE, Aalpha,
                        AExponentN: Real);
    procedure SetToughness(AKmat, AK02, AKg: Real);
    procedure SetCGrowthProp(ACoeffC, AExponentM: Real);

    function  GetMatName   : string;

    function  GetSy        : Real;
    function  GetSu        : Real;
    function  GetSE        : Real;

    function  GetAlpha     : Real;
    function  GetExponentN : Real;
    function  GetKmat      : Real;
    function  GetK02       : Real;
    function  GetKg        : Real;

    function  GetCoeffC    : Real;
    function  GetExponentM : Real;
  end;

implementation

{ TMatProp }

constructor TMatProp.Create(AMatName: string; ASy, ASu, ASE, Aalpha,
                            AExponentN: Real);
begin
  FMatName   := AMatName;
  FSy        := ASy;
  FSu        := ASu;
  FSE        := ASE;
  Falpha     := Aalpha;
  FExponentN := AExponentN;
end;
```

```
function TMatProp.GetAlpha: Real;
begin
  Result := FAlpha;
end;

function TMatProp.GetCoeffC: Real;
begin
  Result := FCoeffC;
end;

function TMatProp.GetExponentM: Real;
begin
  Result := FExponentM;
end;

function TMatProp.GetExponentN: Real;
begin
  Result := FExponentN;
end;

function TMatProp.GetK02: Real;
begin
  Result := FK02;
end;

function TMatProp.GetKg: Real;
begin
  Result := FKg;
end;

function TMatProp.GetKmat: Real;
begin
  Result := FKmat;
end;

function TMatProp.GetMatName: string;
begin
  Result := FMatName;
end;

function TMatProp.GetSE: Real;
begin
  Result := FSE;
end;

function TMatProp.GetSu: Real;
begin
  Result := FSu;
end;

function TMatProp.GetSy: Real;
begin
  Result := FSy;
end;

procedure TMatProp.SetCGrowthProp(ACoeffC, AExponentM: Real);
begin
  FCoeffC    := ACoeffC;
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
  FExponentM := AExponentM;
end;

procedure TMatProp.SetToughness(AKmat, AK02, AKg: Real);
begin
  FKmat := AKmat;
  FK02  := AK02;
  FKg   := AKg;
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.13 ซอร์สโค้ดของคลาส charCrack

```
unit CharCrackU;      // delegstor class

interface

uses CrackTempU, CharStdU;

type
   TCharCrack = class(TObject)
   private
     FNOCrack : integer;
     Crack    : array of TCrack;  // dynamic array
     CharStd  : TCharStd;
     FDiPrinS : String;
   public
     property  NOCrack: integer Read FNOCrack Write FNOCrack;
     procedure SetCrack(ACounter: integer; ACrackType: string;
                        APosition, ADimension, AnAngle: array of
                        Real);
     function  GetType(ACounter: integer): string;
     function  GetDim(ACounter: integer): TDimension;
     function  GetAng(ACounter: integer): TAngle;
     function  GetPos(ACounter, APos: integer): TCoorD;
     function  GetDiPrinS(ACounter: integer): string;
     function  GetDiFlaw(ACounter: integer): string;

     procedure assignCrackStd(AIndexStd: integer);
     procedure Reorientation(Sender: TObject);
     procedure Position;
     procedure Interaction;
  end;

implementation

uses SysUtils, frmCrackU;

{ TCharCrack }

procedure TCharCrack.SetCrack(ACounter: integer; ACrackType: String;
                              APosition, ADimension, AnAngle: array
                              of Real);
var i: integer;
begin
  i := ACounter;   // initial : 0
  SetLength(Crack, NOCrack);  // set length of dynamic array
  if i <= NOCrack -1 then
    begin
      Crack[i] := TCrack.Create(ACrackType, APosition, ADimension,
AnAngle);
    end;
end;

function TCharCrack.GetType(ACounter: integer): string;
begin
  Result := Crack[ACounter].CrackType;
end;
```

```
function TCharCrack.GetAng(ACounter: integer): TAngle;
var Angle: TAngle;
begin
  Angle := Crack[ACounter].GetAngle;
  Result := Angle;
end;

function TCharCrack.GetDim(ACounter: integer): TDimension;
var Dimension: TDimension;
begin
  Dimension := Crack[ACounter].GetDimension;
  Result := Dimension;
end;

function TCharCrack.GetPos(ACounter, APos: integer): TCoorD;
var Pos: TCoorD;
begin
  Pos := Crack[ACounter].GetPos(APos);
  Result := Pos
end;

procedure TCharCrack.assignCrackStd(AIndexStd: integer);
begin
  FreeAndNil(CharStd);
  case AIndexStd of
  0: CharStd := TAPI579.Create
  else
     CharStd := nil;
  end;
end;

procedure TCharCrack.Reorientation(Sender: TObject);
var  i             : integer;    // Crack NO
     Angle         : array [0..1] of Real;
     InFoReOrient  : TInFoReOrient;
     SigmaTemp1,
     SigmaTemp2    : Real;
     Sigma1, Sigma2: Real;
begin
  if assigned(CharStd) then
    begin
      SigmaTemp1 := (Sender as TfrmCrack).GetPrinStressH;
      SigmaTemp2 := (Sender as TfrmCrack).GetPrinStressA;
      for i := 0 to NOCrack - 1 do
        begin
          if SigmaTemp1 < SigmaTemp2 then     // sigmaH < sigmaA
            begin
              Sigma1  := SigmaTemp2;
              Sigma2  := SigmaTemp1;
              FDiPrinS := 'X';
                    //sigma 1 is in to the X direction:Longitudinal
            end
          else                                // sigmaH > sigmaA
            begin
              Sigma1  := SigmaTemp1;
              Sigma2  := SigmaTemp2;
              FDiPrinS := 'Y';
                  //sigma 1 is in to the Y direction:Curcumferiential
```

```
            end;

          InFoReOrient := CharStd.ReOrientation(Crack, i, Sigma1,
                                                Sigma2, FDiPrinS);
          Crack[i].SetEffDimention(InFoReOrient);

          if InFoReOrient[2] = 1 then  // flaw is in X direction
            begin
              Angle[0] := 0;
              Angle[1] := 0;
              Crack[i].DiFlaw := 'X';
            end
          else                         // flaw is in Y direction
            begin
              Angle[0] := 90;
              Angle[1] := 0;
              Crack[i].DiFlaw := 'Y';
            end;
          Crack[i].SetAngle(Angle);
        end;  // end Loop
    end
end;

procedure TCharCrack.Position;
var i: integer;
    Pos1, pos2, Pos3, Pos4: TCoor;
begin
  if assigned(CharStd) then
    begin
      for i := 0 to NOCrack - 1 do
        begin
          Pos1 := CharStd.GetPosition(Crack, i, 1);
          Pos2 := CharStd.GetPosition(Crack, i, 2);
          Pos3 := CharStd.GetPosition(Crack, i, 3);
          Pos4 := CharStd.GetPosition(Crack, i, 4);

          Crack[i].SetPos(1, Pos1);
          Crack[i].SetPos(2, Pos2);
          Crack[i].SetPos(3, Pos3);
          Crack[i].SetPos(4, Pos4);
        end;
    end;
end;

procedure TCharCrack.Interaction;
var CrackNew: TCrackNew;
    DiFlaw1 : string;
    DiFlaw2 : string;
begin
  if assigned(CharStd) then
    begin
      if NOCrack > 1 then
        begin
          DiFlaw1 := Crack[0].DiFlaw;
          DiFlaw2 := Crack[1].DiFlaw;

          CrackNew := CharStd.GetInteraction(Crack, 0, 1, FDiPrinS,
                                             DiFlaw1, DiFlaw2);
          if CrackNew.Valid = true then
```

```
            begin
              Crack[1].FreeInstance;
              NOCrack := 1;
              Crack[0].CrackType := CrackNew.CrackType;
              Crack[0].SetEffDimention(CrackNew.Dim);
              Crack[0].SetPos(1, CrackNew.Pos1);
              Crack[0].SetPos(2, CrackNew.Pos2);
              Crack[0].SetPos(3, CrackNew.Pos3);
              Crack[0].SetPos(4, CrackNew.Pos4);
              Crack[0].SetAngle(CrackNew.Angle);
            end
          else
        end
    end
end;

function TCharCrack.GetDiFlaw(ACounter: integer): string;
begin
  Result := Crack[ACounter].DiFlaw;
end;

function TCharCrack.GetDiPrinS(ACounter: integer): string;
begin
  Result := FDiPrinS;
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.14 ซอร์สโค้ดของคลาส charStd

```
unit CharStdU;    // delegate  class

interface

Uses CrackTempU, Math;

type
   TInFoReOrient = array [0..2] of Real;
   TCoor         = array [0..2] of Real;
   TCrackNew     = record
     Valid    : boolean;
     CrackType: string;
     Dim      : array [0..1] of Real;
     Pos1     : array [0..2] of Real;
     Pos2     : array [0..2] of Real;
     Pos3     : array [0..2] of Real;
     Pos4     : array [0..2] of Real;
     Angle    : array [0..1] of Real;

   end;

   TCharStd = class(TObject)
   protected
     InFoReOrient : TInFoReOrient;
     Coord        : TCoor;
     CrackNew     : TCrackNew;
   public
     function ReOrientation(ACrack: array of TCrack; ANO: integer;
                            ASigma1, ASigma2: Real;
                            ADiPrinS: string):TInFoReOrient;
                            virtual; abstract;
     function GetInteraction(ACrack: array of TCrack;
                             ANO1,ANO2: integer;
                             ADiPrinS, ADiFlaw1, ADiFlaw2: string):
                            TCrackNew; virtual; abstract;
     function GetPosition(ACrack: array of TCrack;
                          ANO, APos: integer): TCoor;
                          virtual; abstract;
   end;

   TAPI579 = class(TCharStd)
   private
     procedure EffLength(ACrack: array of TCrack;
                         ANO: integer;
                         ASigma1, ASigma2: Real;
                         ADiPrinS: string);
     procedure EffDepth(ACrack: array of TCrack; ANO: integer);
     procedure Interaction(ACrack: array of TCrack;
                           ANO1, ANO2: integer);
   public
     function ReOrientation(ACrack: array of TCrack;
                            ANO: integer;
                            ASigma1, ASigma2: Real;
                            ADiPrinS:  string):
                            TInFoReOrient; override;
```

```
     function GetInteraction(ACrack: array of TCrack;
                             ANO1, ANO2: integer;
                            ADiPrinS, ADiFlaw1, ADiFlaw2:
                             string):TCrackNew; override;
     function GetPosition(ACrack: array of TCrack;
                          ANO, APos: integer): TCoor; override;
   end;


implementation

Uses CharCrackU;

{ TAPI579 }

function TAPI579.ReOrientation(ACrack: array of TCrack; ANO: integer;
                               ASigma1, ASigma2: Real;
                               ADiPrinS: string): TInFoReOrient;
begin
  EffLength(ACrack, ANO, ASigma1, ASigma2, ADiPrinS);
  EffDepth (ACrack, ANO);
  InFoReOrient[0] := 2*InFoReOrient[0];

  if ACrack[ANO].CrackType = 'Semi-Elliptical' then
     InFoReOrient[1] := InFoReOrient[1]
  else
     InFoReOrient[1] := 2*InFoReOrient[1];
  Result := InFoReOrient;
end;

procedure TAPI579.EffDepth(ACrack: array of TCrack; ANO: integer);
var WZeta   : Real;
    W       : Real;
    zeta    : Real;
    EffDepth: Real;
begin
  zeta  := ACrack[ANO].GetAngle[1];

  WZeta := 0.99999 + (1.0481E-5*zeta) + (1.5471E-04*Power(zeta,2)) +
           (3.4141E-5*Power(zeta,3))  - (2.0688E-06*Power(zeta,4))
           +
           (4.4977E-8*Power(zeta,5))  - (4.5751E-10*Power(zeta,6))
           +
           (1.8220E-12*Power(zeta,7));

  W       := Max(WZeta,1);
  EffDepth := 0;      // initial

  if ACrack[ANO].CrackType = 'Semi-Elliptical' then
     EffDepth := ACrack[ANO].GetDimension[1] * cos(Pi*zeta/180) * W;
  if ACrack[ANO].CrackType = 'Embeded' then
     EffDepth := ACrack[ANO].GetDimension[1] * cos(Pi*zeta/180) * W *
0.5;
  InFoReOrient[1] := EffDepth;
end;

procedure TAPI579.EffLength(ACrack: array of TCrack; ANO: integer;
                            ASigma1, ASigma2: Real;
                            ADiPrinS: string);
```

```
var B, alpha, alphaN : Real;
    HalfLength       : Real;
begin  // 1
       // alpha is angle between flaw and principle plane sigma 2
  B := ASigma2/ASigma1;
  alpha := ACrack[ANO].GetAngle[0];
       // alpha initial is positive only
  HalfLength := 0;
  if B <> 0 then
    begin  // begin B <> 0
      if ADiPrinS = 'X' then
                     // PrinS is in to the X direction: Longitudinal
        begin  // begin X
          alphaN := ACrack[ANO].GetAngle[0] + 90;
          if (alphaN > 90) and (alphaN < 180) then
              alpha := 180 - alphaN;
          if (alphaN > 180) and(alphaN < 270) then
              alpha := alphaN - 180;
          if (alphaN > 270) and(alphaN < 360) then
              alpha := 360 - alphaN;
          if (alphaN = 0) or (alphaN = 180) or (alphaN = 360) then
              alpha := 0;
          if (alphaN = 90) or (alphaN = 270) then
              alpha := 90;

          if abs(alpha) <= 45 then
            begin
              HalfLength := sqr(cos(Pi*alpha/180)) +
                                0.5*(1 - B)*sin(Pi*alpha/180)*
                                cos(Pi*alpha/180) +
                                sqr(B)*sqr(sin(Pi*alpha/180));
              InFoReOrient[2] := 2;
                                 //flaw perpendicula to X direction
            end
          else
            begin
              HalfLength := sqr(cos(Pi*alpha/180))/sqr(B) +
                               (1 - B)*sin(Pi*alpha/180)*
                                cos(Pi*alpha/180)/ (2*sqr(B)) +
                               sqr(sin(Pi*alpha/180));
              InFoReOrient[2] := 1;
                //flaw perpendicula to Y direction
            end;
        end    // end X
      else     // PrinS is in to the Y direction: Longitudinal
        begin  // begin Y
          alphaN := ACrack[ANO].GetAngle[0];
          if (alphaN > 90) and (alphaN < 180) then
              alpha := 180 - alphaN;
          if (alphaN > 180) and (alphaN < 270) then
              alpha := alphaN - 180;
          if (alphaN > 270) and (alphaN < 360) then
              alpha := 360 - alphaN;
          if (alphaN = 0) or (alphaN = 180) or (alphaN = 360) then
              alpha := 0;
          if (alphaN = 90) or (alphaN = 270) then
              alpha := 90;

          if abs(alpha) <= 45 then
```

```
            begin
              HalfLength := sqr(cos(Pi*alpha/180)) +
                                0.5*(1 - B)*sin(Pi*alpha/180)*
                                cos(Pi*alpha/180) +
                                sqr(B)*sqr(sin(Pi*alpha/180));
              InFoReOrient[2] := 1;
                 //flaw is in to to X direction
            end
          else
            begin
              HalfLength := sqr(cos(Pi*alpha/180))/sqr(B) +
                            (1 - B)*sin(Pi*alpha/180)*
                            cos(Pi*alpha/180)/(2*sqr(B)) +
                             sqr(sin(Pi*alpha/180));
              InFoReOrient[2] := 2;
               //flaw perpendicula X direction
            end;
        end;   // end Y
    end    // end B <> 0
  else // special load: axial tension or bending moment B equal zero
    begin  // begin B = 0
      if ADiPrinS = 'X' then
        begin  // begin X
          alphaN := ACrack[ANO].GetAngle[0] + 90;
          if (alphaN > 90) and (alphaN < 180) then
                  alpha := 180 - alphaN;
          if (alphaN > 180) and(alphaN < 270) then
                  alpha := alphaN - 180;
          if (alphaN > 270) and(alphaN < 360) then
                  alpha := 360 - alphaN;
          if (alphaN = 0) or (alphaN = 180) or (alphaN = 360) then
                  alpha := 0;
          if (alphaN = 90) or (alphaN = 270) then
                  alpha := 90;

          HalfLength := sqr(cos(Pi*alpha/180)) +
                            0.5*sin(Pi*alpha/180)*cos(Pi*alpha/180);
          InFoReOrient[2] := 2;   //flaw perpendicula to X direction
        end;  // end X
    end;   // end B = 0
  InFoReOrient[0] := 0.5*ACrack[ANO].GetDimension[0]*HalfLength;
end;  // 1

function TAPI579.GetPosition(ACrack: array of TCrack; ANO, APos:
integer):TCoor;
var alpha: Real;
begin
  alpha := ACrack[ANO].GetAngle[0];
  if alpha = 0 then     // crack is in X direction
    begin
      if ACrack[ANO].CrackType = 'Semi-Elliptical' then
        begin
          if APos = 1 then
            begin
              Coord[0] :=   ACrack[ANO].GetPos(3)[0] -
                                          // (Position)[x,y,z]
                            0.5*ACrack[ANO].GetDimension[0];
              Coord[1] :=   ACrack[ANO].GetPos(3)[1];
              Coord[2] :=   ACrack[ANO].GetPos(3)[2];
```

```
                                                  // must be equal to zero
            end
          else if APos = 2 then
            begin
              Coord[0] :=   ACrack[ANO].GetPos(3)[0] +
                            0.5*ACrack[ANO].GetDimension[0];
              Coord[1] :=   ACrack[ANO].GetPos(3)[1];
              Coord[2] :=   ACrack[ANO].GetPos(3)[2];
                                                  // must be equal to zero
            end
          else if APos = 3 then
            begin
              Coord[0] :=   ACrack[ANO].GetPos(3)[0];
              Coord[1] :=   ACrack[ANO].GetPos(3)[1];
              Coord[2] :=   ACrack[ANO].GetPos(3)[2];
                                                  // must be equal to zero
            end
          else if APos = 4 then
            begin
              Coord[0] :=   ACrack[ANO].GetPos(3)[0];
              Coord[1] :=   ACrack[ANO].GetPos(3)[1];
              Coord[2] := - ACrack[ANO].GetDimension[1];
                                                  // equal depth
            end;
        end
      else if ACrack[ANO].CrackType = 'Embeded' then
        begin
          if APos = 1 then
            begin
              Coord[0] := ACrack[ANO].GetPos(3)[0] -
                          0.5*ACrack[ANO].GetDimension[0];
              Coord[1] := ACrack[ANO].GetPos(3)[1];
              Coord[2] := ACrack[ANO].GetPos(3)[2];
            end
          else if APos = 2 then
            begin
              Coord[0] := ACrack[ANO].GetPos(3)[0] +
                          0.5*ACrack[ANO].GetDimension[0];
              Coord[1] := ACrack[ANO].GetPos(3)[1];
              Coord[2] := ACrack[ANO].GetPos(3)[2];
            end
          else if APos = 3 then
            begin
              Coord[0] := ACrack[ANO].GetPos(3)[0];
              Coord[1] := ACrack[ANO].GetPos(3)[1];
              Coord[2] := ACrack[ANO].GetPos(3)[2] +
                          0.5*ACrack[ANO].GetDimension[1];
            end
          else if APos = 4 then
            begin
              Coord[0] := ACrack[ANO].GetPos(3)[0];
              Coord[1] := ACrack[ANO].GetPos(3)[1];
              Coord[2] := ACrack[ANO].GetPos(3)[2] -
                          0.5*ACrack[ANO].GetDimension[1];
            end;
        end;
    end
  else        // crack is in Y direction
    begin
```

```
if ACrack[ANO].CrackType = 'Semi-Elliptical' then
  begin
    if APos = 1 then
      begin
        Coord[0] :=   ACrack[ANO].GetPos(3)[0];
                                        // (Position)[x,y,z]
        Coord[1] :=   ACrack[ANO].GetPos(3)[1] +
                      0.5*ACrack[ANO].GetDimension[0];;
        Coord[2] :=   ACrack[ANO].GetPos(3)[2];
                                        // must be equal to zero
      end
    else if APos = 2 then
      begin
        Coord[0] :=   ACrack[ANO].GetPos(3)[0];
        Coord[1] :=   ACrack[ANO].GetPos(3)[1] -
                      0.5*ACrack[ANO].GetDimension[0];
        Coord[2] :=   ACrack[ANO].GetPos(3)[2];
                                        // must be equal to zero
      end
    else if APos = 3 then
      begin
        Coord[0] :=   ACrack[ANO].GetPos(3)[0];
        Coord[1] :=   ACrack[ANO].GetPos(3)[1];
        Coord[2] :=   ACrack[ANO].GetPos(3)[2];
                                        // must be equal to zero
      end
    else if APos = 4 then
      begin
        Coord[0] :=   ACrack[ANO].GetPos(3)[0];
        Coord[1] :=   ACrack[ANO].GetPos(3)[1];
        Coord[2] := - ACrack[ANO].GetDimension[1];
                                        // equal depth
      end;
  end
else if ACrack[ANO].CrackType = 'Embeded' then
  begin
    if APos = 1 then
      begin
        Coord[0] := ACrack[ANO].GetPos(3)[0];
        Coord[1] := ACrack[ANO].GetPos(3)[1] +
                    0.5*ACrack[ANO].GetDimension[0];
        Coord[2] := ACrack[ANO].GetPos(3)[2];
      end
    else if APos = 2 then
      begin
        Coord[0] := ACrack[ANO].GetPos(3)[0];
        Coord[1] := ACrack[ANO].GetPos(3)[1] +
                    0.5*ACrack[ANO].GetDimension[0];
        Coord[2] := ACrack[ANO].GetPos(3)[2];
      end
    else if APos = 3 then
      begin
        Coord[0] := ACrack[ANO].GetPos(3)[0];
        Coord[1] := ACrack[ANO].GetPos(3)[1];
        Coord[2] := ACrack[ANO].GetPos(3)[2] +
                    0.5*ACrack[ANO].GetDimension[1];
      end
    else if APos = 4 then
      begin
```

```
              Coord[0] := ACrack[ANO].GetPos(3)[0];
              Coord[1] := ACrack[ANO].GetPos(3)[1];
              Coord[2] := ACrack[ANO].GetPos(3)[2] -
                          0.5*ACrack[ANO].GetDimension[1];
            end;
        end;
    end;
  Result := Coord;
end;

function TAPI579.GetInteraction(ACrack: array of TCrack;
                                ANO1, ANO2: integer;
                                ADiPrinS, ADiFlaw1,
                                ADiFlaw2: string): TCrackNew;
begin  //1
// check  principle stree max in direction ? : curcumferiantial of
//longitudinal
// check direction of flaw ? : curcumferiantial of longitudinal
  if ((ADiPrinS = 'Y')  and (ADiFlaw1 = 'X')  and
     (ADiFlaw2 = 'X'))  or ((ADiPrinS = 'X' ) and
     (ADiFlaw1 = 'Y')   and (ADiFlaw2 = 'Y')) then
    begin
      Interaction(ACrack, ANO1, ANO2);
    end
  else
    begin
      if ADiPrinS = 'X' then
        begin
          if ADiFlaw1 = 'Y' then
            begin
              //check length , check depth
              CrackNew.Valid     := true;
              CrackNew.CrackType := ACrack[ANO1].CrackType;
              CrackNew.Dim[0]    := ACrack[ANO1].GetDimension[0];
              CrackNeW.Dim[1]    := ACrack[ANO1].GetDimension[1];

              CrackNew.Pos1[0]   := ACrack[ANO1].GetPos(1)[0];
              CrackNew.Pos1[1]   := ACrack[ANO1].GetPos(1)[1];
              CrackNew.Pos1[2]   := ACrack[ANO1].GetPos(1)[2];

              CrackNew.Pos2[0]   := ACrack[ANO1].GetPos(2)[0];
              CrackNew.Pos2[1]   := ACrack[ANO1].GetPos(2)[1];
              CrackNew.Pos2[2]   := ACrack[ANO1].GetPos(2)[2];

              CrackNew.Pos3[0]   := ACrack[ANO1].GetPos(3)[0];
              CrackNew.Pos3[1]   := ACrack[ANO1].GetPos(3)[1];
              CrackNew.Pos3[2]   := ACrack[ANO1].GetPos(3)[2];

              CrackNew.Pos4[0]   := ACrack[ANO1].GetPos(4)[0];
              CrackNew.Pos4[1]   := ACrack[ANO1].GetPos(4)[1];
              CrackNew.Pos4[2]   := ACrack[ANO1].GetPos(4)[2];

              CrackNew.Angle[0]  := ACrack[ANO1].GetAngle[0];
              CrackNew.Angle[1]  := ACrack[ANO1].GetAngle[1];
            end
          else if ADiFlaw2 = 'Y' then
            begin
              //check length , check depth
              CrackNew.Valid     := true;
```

```
          CrackNew.CrackType := ACrack[ANO2].CrackType;
          CrackNew.Dim[0]    := ACrack[ANO2].GetDimension[0];
          CrackNeW.Dim[1]    := ACrack[ANO2].GetDimension[1];

          CrackNew.Pos1[0]   := ACrack[ANO2].GetPos(1)[0];
          CrackNew.Pos1[1]   := ACrack[ANO2].GetPos(1)[1];
          CrackNew.Pos1[2]   := ACrack[ANO2].GetPos(1)[2];

          CrackNew.Pos2[0]   := ACrack[ANO2].GetPos(2)[0];
          CrackNew.Pos2[1]   := ACrack[ANO2].GetPos(2)[1];
          CrackNew.Pos2[2]   := ACrack[ANO2].GetPos(2)[2];

          CrackNew.Pos3[0]   := ACrack[ANO2].GetPos(3)[0];
          CrackNew.Pos3[1]   := ACrack[ANO2].GetPos(3)[1];
          CrackNew.Pos3[2]   := ACrack[ANO2].GetPos(3)[2];

          CrackNew.Pos4[0]   := ACrack[ANO2].GetPos(4)[0];
          CrackNew.Pos4[1]   := ACrack[ANO2].GetPos(4)[1];
          CrackNew.Pos4[2]   := ACrack[ANO2].GetPos(4)[2];

          CrackNew.Angle[0]  := ACrack[ANO2].GetAngle[0];
          CrackNew.Angle[1]  := ACrack[ANO2].GetAngle[1];
        end;

    end
  else if ADiPrinS = 'Y' then
    begin
      if ADiFlaw1 = 'X' then
        begin
          //check length , check depth
          CrackNew.Valid     := true;
          CrackNew.CrackType := ACrack[ANO1].CrackType;
          CrackNew.Dim[0]    := ACrack[ANO1].GetDimension[0];
          CrackNeW.Dim[1]    := ACrack[ANO1].GetDimension[1];

          CrackNew.Pos1[0]   := ACrack[ANO1].GetPos(1)[0];
          CrackNew.Pos1[1]   := ACrack[ANO1].GetPos(1)[1];
          CrackNew.Pos1[2]   := ACrack[ANO1].GetPos(1)[2];

          CrackNew.Pos2[0]   := ACrack[ANO1].GetPos(2)[0];
          CrackNew.Pos2[1]   := ACrack[ANO1].GetPos(2)[1];
          CrackNew.Pos2[2]   := ACrack[ANO1].GetPos(2)[2];

          CrackNew.Pos3[0]   := ACrack[ANO1].GetPos(3)[0];
          CrackNew.Pos3[1]   := ACrack[ANO1].GetPos(3)[1];
          CrackNew.Pos3[2]   := ACrack[ANO1].GetPos(3)[2];

          CrackNew.Pos4[0]   := ACrack[ANO1].GetPos(4)[0];
          CrackNew.Pos4[1]   := ACrack[ANO1].GetPos(4)[1];
          CrackNew.Pos4[2]   := ACrack[ANO1].GetPos(4)[2];

          CrackNew.Angle[0]  := ACrack[ANO1].GetAngle[0];
          CrackNew.Angle[1]  := ACrack[ANO1].GetAngle[1];
        end
      else if ADiFlaw2 = 'X' then
        begin
          //check length , check depth
          CrackNew.Valid     := true;
          CrackNew.CrackType := ACrack[ANO2].CrackType;
```

```
                CrackNew.Dim[0]    := ACrack[ANO2].GetDimension[0];
                CrackNeW.Dim[1]    := ACrack[ANO2].GetDimension[1];

                CrackNew.Pos1[0]   := ACrack[ANO2].GetPos(1)[0];
                CrackNew.Pos1[1]   := ACrack[ANO2].GetPos(1)[1];
                CrackNew.Pos1[2]   := ACrack[ANO2].GetPos(1)[2];

                CrackNew.Pos2[0]   := ACrack[ANO2].GetPos(2)[0];
                CrackNew.Pos2[1]   := ACrack[ANO2].GetPos(2)[1];
                CrackNew.Pos2[2]   := ACrack[ANO2].GetPos(2)[2];

                CrackNew.Pos3[0]   := ACrack[ANO2].GetPos(3)[0];
                CrackNew.Pos3[1]   := ACrack[ANO2].GetPos(3)[1];
                CrackNew.Pos3[2]   := ACrack[ANO2].GetPos(3)[2];

                CrackNew.Pos4[0]   := ACrack[ANO2].GetPos(4)[0];
                CrackNew.Pos4[1]   := ACrack[ANO2].GetPos(4)[1];
                CrackNew.Pos4[2]   := ACrack[ANO2].GetPos(4)[2];

                CrackNew.Angle[0]  := ACrack[ANO2].GetAngle[0];
                CrackNew.Angle[1]  := ACrack[ANO2].GetAngle[1];
              end;
          end;
      end;
    Result := CrackNew;
end;  //1

procedure TAPI579.Interaction(ACrack: array of TCrack; ANO1, ANO2:
integer);
var S1   : Real; // distance between cracks' center in y-direction
    S2   : Real; // distance between cracks' tip    in x-direction
    S3   : Real; // distance between cracks' tip    in z-direction

    Ceff : Real; // effective half crack length (if interaction
occurs)
    Aeff : Real; // effective (half or total) crack depth(if
interaction occurs)

    DepthTemp1: Real;
    DepthTemp2: Real;
begin
// Calculate distance
  // Y direction
  S1 := Abs(ACrack[ANO1].GetPos(3)[1] - ACrack[ANO2].GetPos(3)[1]);

  // X direction
  S2 := Max(ACrack[ANO1].GetPos(2)[0],  ACrack[ANO2].GetPos(2)[0]) -
        Min(ACrack[ANO1].GetPos(1)[0],  ACrack[ANO2].GetPos(1)[0]) -
        Abs(ACrack[ANO1].GetPos(2)[0] - ACrack[ANO1].GetPos(1)[0]) -
        Abs(ACrack[ANO2].GetPos(2)[0] - ACrack[ANO2].GetPos(1)[0]);

  // Z direction
  S3 := Abs(min(ACrack[ANO1].GetPos(4)[2],
         ACrack[ANO2].GetPos(4)[2])) -
         Abs(max(ACrack[ANO1].GetPos(3)[2],
         ACrack[ANO2].GetPos(3)[2])) -
         Abs(ACrack[ANO1].GetPos(4)[2] - ACrack[ANO1].GetPos(3)[2]) -
         Abs(ACrack[ANO2].GetPos(4)[2] - ACrack[ANO2].GetPos(3)[2]);
```

```
// Apply interaction rule
    if (0.5*ACrack[ANO1].GetDimension[0] +
        0.5*ACrack[ANO2].GetDimension[0] >= S1) and
                                          //Planar interaction rule
       (0.5*ACrack[ANO1].GetDimension[0] +
        0.5*ACrack[ANO2].GetDimension[0] >= S2) and
                                          //Planar interaction rule
       (0.5*ACrack[ANO1].GetDimension[1] +
        0.5*ACrack[ANO2].GetDimension[1] >= S3) then
                                         //Coplanar interaction rule

      begin  // begin interaction
             // Planar interaction occurs
        CrackNew.Valid := true;
        Ceff :=  0.5*(max(ACrack[ANO1].GetPos(2)[0],
                 ACrack[ANO2].GetPos(2)[0]) -
                 min(ACrack[ANO1].GetPos(1)[0],
                 ACrack[ANO1].GetPos(1)[0]));
// Assign the first crack as a combined crack
// Calculate cracks' position (planar)

        CrackNew.Pos1[0] := min(min(ACrack[ANO1].GetPos(1)[0],
//Pos1[X]                      ACrack[ANO1].GetPos(2)[0]),
                             min(ACrack[ANO2].GetPos(1)[0],
                             ACrack[ANO2].GetPos(2)[0]));
        CrackNew.Pos1[1] := 0.5*(ACrack[ANO1].GetPos(1)[1] +
//Pos1[Y]                      ACrack[ANO2].GetPos(1)[1]);

        CrackNew.Pos2[0] := max(max(ACrack[ANO1].GetPos(1)[0],
//Pos2[X]                      ACrack[ANO1].GetPos(2)[0]),
                             max(ACrack[ANO2].GetPos(1)[0],
                             ACrack[ANO2].GetPos(2)[0]));
        CrackNew.Pos2[1] := 0.5*(ACrack[ANO1].GetPos(2)[1] +
//Pos2[Y]                      ACrack[ANO2].GetPos(2)[1]);

        if (ACrack[ANO1].CrackType = 'Embeded') and
           (ACrack[ANO2].CrackType = 'Embeded') then
          begin
            Aeff := 0.5*(Abs(min(ACrack[ANO1].GetPos(4)[2],
                    ACrack[ANO2].GetPos(4)[2])) -
                    Abs(max(ACrack[ANO1].GetPos(3)[2],
                    ACrack[ANO2].GetPos(3)[2])));

            CrackNew.Pos3[0] := 0.5*(CrackNew.Pos1[0] +
//Pos3[X]                      CrackNew.Pos2[0]);
            CrackNew.Pos3[1] := CrackNew.Pos1[1];
//Pos3[Y]
            CrackNew.Pos3[2] := max(max(ACrack[ANO1].GetPos(3)[2],
                                  ACrack[ANO1].GetPos(4)[2]),
                                  max(ACrack[ANO2].GetPos(3)[2],
                                  ACrack[ANO2].GetPos(4)[2]));
//Pos3[Z]

            CrackNew.Pos4[0] := CrackNew.Pos3[0];
//Pos4[X]
            CrackNew.Pos4[1] := CrackNew.Pos3[1];
//Pos4[Y]
            CrackNew.Pos4[2] := CrackNew.Pos3[2] - (2 * Aeff);
//Pos4[Z]
```

```
            CrackNew.Pos1[2] := CrackNew.Pos3[2] - Aeff;
//Pos1[Z]
            CrackNew.Pos2[2] := CrackNew.Pos1[2];
//Pos2[Z]

            CrackNew.CrackType := 'Embeded';
            CrackNew.Dim[0]    := 2 * Ceff;
            CrackNew.Dim[1]    := 2 * Aeff;
            CrackNew.Angle[0]  := 0;
            CrackNew.Angle[1]  := 0;
          end
        else         // (semi-ell, semi-ell), (semi-ell, emb)
          begin
            Aeff := Abs(min(ACrack[ANO1].GetPos(4)[2],
                      ACrack[ANO2].GetPos(4)[2])) -
                      Abs(max(ACrack[ANO1].Getpos(3)[2],
                      ACrack[ANO2].GetPos(3)[2]));

            CrackNew.Pos3[0] := 0.5*(CrackNew.Pos2[0] +
                                CrackNew.Pos1[0]);
//Pos3[X]
            CrackNew.Pos3[1] := CrackNew.Pos1[1];
//Pos3[Y]
            CrackNew.Pos3[2] := 0;
//Pos3[Z]

            CrackNew.Pos4[0] := CrackNew.Pos3[0];
//Pos4[X]
            CrackNew.Pos4[1] := CrackNew.Pos3[1];
//Pos4[Y]
            CrackNew.Pos4[2] := - Aeff;
//Pos4[Z]

            CrackNew.Pos1[2] := CrackNew.Pos3[2];
//Pos1[Z]
            CrackNew.Pos2[2] := CrackNew.Pos3[2];
//Pos2[Z]

            CrackNew.CrackType := 'Semi-Elliptical';
            CrackNew.Dim[0]    := 2 * Ceff;
            CrackNew.Dim[1]    := Aeff;
            CrackNew.Angle[0]  := 0;
            CrackNew.Angle[1]  := 0;
          end;
          // Ligament Yielding
      end   // end interaction
    else    // non interaction  Check crack depth only
      begin
        CrackNew.Valid := true;
        DepthTemp1 := ACrack[ANO1].GetDimension[1];
        DepthTemp2 := ACrack[ANO2].GetDimension[1];

        if DepthTemp1 > DepthTemp2 then
          begin
            CrackNew.CrackType := ACrack[ANO1].CrackType;
            CrackNew.Dim[0]    := ACrack[ANO1].GetDimension[0];
            CrackNeW.Dim[1]    := ACrack[ANO1].GetDimension[1];
```

```
            CrackNew.Pos1[0]   := ACrack[ANO1].GetPos(1)[0];
            CrackNew.Pos1[1]   := ACrack[ANO1].GetPos(1)[1];
            CrackNew.Pos1[2]   := ACrack[ANO1].GetPos(1)[2];

            CrackNew.Pos2[0]   := ACrack[ANO1].GetPos(2)[0];
            CrackNew.Pos2[1]   := ACrack[ANO1].GetPos(2)[1];
            CrackNew.Pos2[2]   := ACrack[ANO1].GetPos(2)[2];

            CrackNew.Pos3[0]   := ACrack[ANO1].GetPos(3)[0];
            CrackNew.Pos3[1]   := ACrack[ANO1].GetPos(3)[1];
            CrackNew.Pos3[2]   := ACrack[ANO1].GetPos(3)[2];

            CrackNew.Pos4[0]   := ACrack[ANO1].GetPos(4)[0];
            CrackNew.Pos4[1]   := ACrack[ANO1].GetPos(4)[1];
            CrackNew.Pos4[2]   := ACrack[ANO1].GetPos(4)[2];

            CrackNew.Angle[0]  := ACrack[ANO1].GetAngle[0];
            CrackNew.Angle[1]  := ACrack[ANO1].GetAngle[1];
          end
        else
          begin
            CrackNew.CrackType := ACrack[ANO2].CrackType;
            CrackNew.Dim[0]    := ACrack[ANO2].GetDimension[0];
            CrackNeW.Dim[1]    := ACrack[ANO2].GetDimension[1];

            CrackNew.Pos1[0]   := ACrack[ANO2].GetPos(1)[0];
            CrackNew.Pos1[1]   := ACrack[ANO2].GetPos(1)[1];
            CrackNew.Pos1[2]   := ACrack[ANO2].GetPos(1)[2];

            CrackNew.Pos2[0]   := ACrack[ANO2].GetPos(2)[0];
            CrackNew.Pos2[1]   := ACrack[ANO2].GetPos(2)[1];
            CrackNew.Pos2[2]   := ACrack[ANO2].GetPos(2)[2];

            CrackNew.Pos3[0]   := ACrack[ANO2].GetPos(3)[0];
            CrackNew.Pos3[1]   := ACrack[ANO2].GetPos(3)[1];
            CrackNew.Pos3[2]   := ACrack[ANO2].GetPos(3)[2];

            CrackNew.Pos4[0]   := ACrack[ANO2].GetPos(4)[0];
            CrackNew.Pos4[1]   := ACrack[ANO2].GetPos(4)[1];
            CrackNew.Pos4[2]   := ACrack[ANO2].GetPos(4)[2];

            CrackNew.Angle[0]  := ACrack[ANO2].GetAngle[0];
            CrackNew.Angle[1]  := ACrack[ANO2].GetAngle[1]
          end;
      end;
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.15 ซอร์สโค้ดของคลาส crackTemp

```
unit CrackTempU;

interface

type
   TCoorD      = array [0..2] of Real;// Coordinate
   TDimension  = array [0..1] of Real;// Length, Depth
   TAngle      = array [0..1] of Real;// Planar, Thickness directions

   TCrack = class(TObject)
   private
     FCrackType    : String;
     FPos1, FPos2,
     FPos3, FPos4  : TCoorD;
     FDimension    : TDimension;
     FAngle        : TAngle;
     FDiFlaw       : string;
   public
     constructor Create(ACrackType: String;
                        APosition, ADimension, AnAngle: array of
                        Real);
     property CrackType: string Read FCrackType Write FCrackType;
     property DiFlaw: string Read FDiFlaw Write FDiFlaw;

     function GetPos(APos: integer): TCoorD;
     function GetDimension: TDimension;
     function GetAngle: TAngle;

     procedure SetEffDimention(AEffDim: array of Real);
     procedure SetPos(APos: integer; ACoor: array of Real);
     procedure SetAngle(AAngle: array of Real);
   end;

implementation

{ TCrack }

constructor TCrack.Create(ACrackType: String; APosition, ADimension,
                          AnAngle: array of Real);
var i: integer;
begin
  CrackType := ACrackType;
  for i := 0 to 2 do
    begin
      FPos3[i] := APosition[i];  // initial at position 3: mean point
    end;
    for i := 0 to 1 do
      begin
        FDimension[i] := ADimension[i];
        FAngle[i]     := AnAngle[i];
      end;
end;

function TCrack.GetAngle: TAngle;
begin
```

```
  Result := FAngle;
end;

function TCrack.GetDimension: TDimension;
begin
  Result := FDimension;
end;

function TCrack.GetPos(APos: integer): TCoorD;
begin
  if APos = 1 then
     Result := FPos1
  else if APos = 2 then
     Result := FPos2
  else if APos = 3 then
     Result := FPos3
  else if APos = 4 then
     Result := FPos4;
end;

procedure TCrack.SetAngle(AAngle: array of Real);
var i: integer;
begin
  for i := 0 to 1 do
    begin
      FAngle[i] := AAngle[i];
    end;
end;

procedure TCrack.SetPos(APos: integer; ACoor: array of Real);
var i: integer;
begin
  for i := 0 to 2 do
    begin
      if APos = 1 then
        FPos1[i] := ACoor[i]
      else if APos = 2 then
        FPos2[i] := ACoor[i]
      else if APos = 3 then
        FPos3[i] := ACoor[i]
      else if APos = 4 then
        FPos4[i] := ACoor[i];
    end;
end;

procedure TCrack.SetEffDimention(AEffDim: array of Real);
var i: integer;
begin
  for i := 0 to 1 do
    begin
      FDimension[i] := AEffDim[i];
    end;
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.16 ซอร์สโค้ดของคลาส assessment

```
unit AssessmentU;

interface

Uses StructureU, CrackTPMU, LimitLoadsU, R6FADU,
     ConstitutiveU, CrackGrowthU, Windows, Messages,
     SysUtils,  Variants, Classes, Graphics, Controls, Forms,
     Dialogs, StdCtrls, Buttons, ExtCtrls, ComCtrls;

type

  TAssessment = class(TObject)
  private
    FKI_a0         : Real;   // Stress intensity factor
    FKI_ag         : Real;
    FPL_a0         : Real;   // Limit Load
    FPL_ag         : Real;
    FKr_a0         : Real;   // Assessment point
    FKr_a02        : Real;
    FKr_ag         : Real;
    FLr_a0         : Real;   // Assessment point
    FLr_ag         : Real;
    FAssessResult  : String;
    FKrC           : Real;
                       //critical point of Assessment line Kr0, Kr0.2
    FLrC           : Real;
    FKrC_g         : Real;
                      //critical point of Assessment line at Krg, Lrg
    FLrC_g         : Real;
    FFL            : Real;   //  Resurve factor of Kr0, Kr0.2
    FFL_g          : Real;   //  Resurve factor of Krg
    FBDL_ATL,
    FATL_BDL       : Real;   // for check limit load
    FaTotal        : Real;
    FcTotal        : Real;
    FGrowth        : Boolean;
    FCycleC        : integer;  // fatigue critical cycle

    FLevel         : integer;
    LimitLoad      : TLimitLoad;
    SIFObj         : TCrackTPM;
    FADObj         : TR6FAD;
    Constitutive   : TConstitutive;
    CrackGrowthObj: TCrackGrowth;
    FCaller        : TObject;
    function CalTotalKI(AStructure: TStructure;AnAssessPos: integer):
                        Real;
    function StructureStatus(AStructure: TStructure): string;
    function LoadStatus(AStructure: TStructure)     : string;
    function CheckAndCalPL(AStructure: TStructure)  : Real;
    function AssessPointLevel1(Sender: TObject;
                               AStructure: TStructure): String;
    function AssessPointLevel2(Sender: TObject;
                               AStructure: TStructure): String;
```

```
    function CalRootLr(Sender: TObject; AKrA, ALrA  : Real;
                       AStructure: TStructure)      : Real;
    function Line(AKrA, ALrA, ALr: Real)            : Real;
   public
    procedure AssignSIF;                      //Create SIFObj
    procedure AssignLimitLoads;               //Create LimitLoad
    procedure AssignFAD(AOptionFAD: integer); //Create FADObj
    procedure AssignConstitutive(AStructure: TStructure);
                                              //Create Constitutive

                                              //Create Crack Growth
    procedure AssignCrackGrowth(AMeNism, ACyOrTime, ADirector:
                                integer);
    procedure Level(ALevel: integer);
    function  GetFADName            : string;
    procedure CalKI(AStructure: TStructure; AnAssessPos: integer);
    function  GetKI_a0              : Real;
    function  GetKI_ag              : Real;
    procedure CalLimitLoad(AStructure: TStructure);
    function  GetLimitLoad_a0       : Real;
    function  GetLimitLoad_ag       : Real;
    procedure CalKr(AStructure: TStructure);
    function  GetKr_a0              : Real;
    function  GetKr_a02             : Real;
    function  GetKr_ag              : Real;
    procedure CalLr(AStructure: TStructure);
    function  GetLr_a0              : Real;
    function  GetLr_ag              : Real;
    procedure CalAssessPoint(Sender: TObject; AStructure:
                             TStructure);
    function  GetAssessmentResult: string;
// call back
    function  GetStrainRef(ALr: Real): Real;
// end call back
    function  GetKrC                : Real;
    function  GetLrC                : Real;
    function  GetKrC_g              : Real;
    function  GetLrC_g              : Real;
    function  GetFL                 : Real;
    function  GetFL_g               : Real;
    procedure CalCrackGrowth(AStructure: TStructure);
    function  GetCrackDepthNew      : Real;
    function  GetCrackLengthNew     : Real;
    function  GetFCycleC            : integer;

    function GetMechanism           : string;
    function GetLoop                : integer;
    function GetNODirector          : integer;
  end;

implementation

{ TAssessment }

function TAssessment.GetStrainRef(ALr: Real): Real;      // Call Back
begin
  if assigned (Constitutive) then
    Result := Constitutive.GetStrainRef_Romberg(ALr)
```

```
  else
    Result := 0;
end;

procedure TAssessment.AssignFAD(AOptionFAD: integer);
begin
  FreeAndNil(FADObj);
  FGrowth := false;
  case AOptionFAD of
  0: FADObj := TOption1.Create;
  1: begin
       FADObj := TOption2.Create;
     end;
  end;
end;

procedure TAssessment.AssignSIF;
begin
  FreeAndNil(SIFObj);
  SIFObj := TKI.Create;
end;

procedure TAssessment.AssignLimitLoads;
begin
  LimitLoad := TLimitLoad.Create;
end;

function TAssessment.GetFADName: string;
begin
  if assigned(FADObj) then
    Result := FADObj.GetFADName
  else
    Result := '';
end;

procedure TAssessment.Level(ALevel: integer);
begin
  FLevel := ALevel;
end;

procedure TAssessment.AssignConstitutive(AStructure: TStructure);
begin
  FreeAndNil(Constitutive);
  Constitutive := TConstitutive.Create(AStructure);
end;

procedure TAssessment.AssignCrackGrowth(AMeNism, ACyOrTime,
                                        ADirector: integer);
begin
  freeAndnil(CrackGrowthObj);
  case AMeNism of
  0: CrackGrowthObj := TFatigue.Create(ACyOrTime, ADirector);
  1: CrackGrowthObj := TEAC.Create;
  end;
end;

procedure TAssessment.CalKI(AStructure: TStructure; AnAssessPos:
                            integer);
```

```
var  a0 : Real;
     ag : Real;
begin
  a0 := AStructure.GetOCDepth;  // crack depth a0
  ag := AStructure.GetCCDepth;  // crack depth ag is original depth
of CCDepth
  if FLevel = 0 then
    begin
//CalCrack tip parameter with Crack depth ao and then keep result(KI)
//in parameter KI_a0
      if FGrowth = False then
        begin
          AStructure.SetCCDepth(a0);
          FKI_a0 := CalTotalKI(AStructure, AnAssessPos);
          AStructure.SetCCDepth(ag);
        end
      else
        begin
          AStructure.SetCCDepth(FaTotal);
          AStructure.SetCCLength(FcTotal);
          FKI_a0 := CalTotalKI(AStructure, AnAssessPos);
          AStructure.SetCCDepth(ag);             // Delta ag
        end;
    end
  else if FLevel = 1 then
    begin
//CalCrack tip parameter with Crack depth ao and then keep result(KI)
//in parameter KI_a0
      if FGrowth = False then
        begin
          AStructure.SetCCDepth(a0);
          FKI_a0 := CalTotalKI(AStructure, AnAssessPos);
          AStructure.SetCCDepth(ag + a0);

          FKI_ag := CalTotalKI(AStructure, AnAssessPos);
          AStructure.SetCCDepth(ag);
        end
      else
        begin
          AStructure.SetCCDepth(FaTotal);
          AStructure.SetCCLength(FcTotal);
          FKI_a0 := CalTotalKI(AStructure, AnAssessPos);
          AStructure.SetCCDepth(ag + FaTotal);

          FKI_ag := CalTotalKI(AStructure, AnAssessPos);
          AStructure.SetCCDepth(ag);
        end;
//CalCrack tip parameter with Crack depth ag and then keep result(KI)
//in parameter KI_ag
    end;
end;

function TAssessment.CalTotalKI(AStructure: TStructure;
                                      AnAssessPos: integer): Real;
var SStatus     : string;
    LStatus     : string;
    TotalStatus : string;
    Phi         : Real;
```

```
    Load        : Real;
    KI_AT       : Real;
    KI_BD       : Real;
    KI_IP       : Real;
    KI_SD       : Real;
begin
  if AnAssessPos = 0 then     // for Deepest points
    begin
      Phi := 90;
    end
  else                        // for surface points
    begin
      Phi := 0;
    end;
  SStatus := StructureStatus(AStructure);
                              // Check Status of CrackStructure
  Load := AStructure.GetAxialTension; // Check Status of Load
  if Load > 0 then
    begin
      LStatus     := '_AT';     // AT : Axial Tension
      TotalStatus := SStatus + LStatus;
      KI_AT       := SIFObj.GetKI(AStructure,TotalStatus, Phi);
    end
  else
      KI_AT       := 0;

  Load := AStructure.GetBending;
  if Load > 0 then
    begin
      LStatus     := '_BD';    // BD : Bending Moment
      TotalStatus := SStatus + LStatus;
      KI_BD       := SIFObj.GetKI(AStructure,TotalStatus, Phi);
    end
  else
      KI_BD       := 0;

  Load := AStructure.GetInPressure;
  if Load > 0 then
    begin
      LStatus     := '_IP';    // IP : Internal Pressure
      TotalStatus := SStatus + LStatus;
      KI_IP       := SIFObj.GetKI(AStructure,TotalStatus, Phi);
    end
  else
      KI_IP       := 0;

  load := AStructure.GetStress_Coeff[0];
  if Load > 0 then
    begin
      LStatus     := '_SD';    // SD : Arbitary stress distribution
      TotalStatus := SStatus + LStatus;
      KI_SD       := SIFObj.GetKI(AStructure,TotalStatus, Phi);
    end
  else
      KI_SD := 0;

  Result := KI_AT + KI_BD + KI_IP + KI_SD;
end;
```

```
function TAssessment.StructureStatus(AStructure: TStructure): string;
var StructureType : string;
    Orientation   : string;
    CrackType     : string;
    Position      : string;
begin
  StructureType := AStructure.StructureName;
    if StructureType = 'Pipe' then
       StructureType := 'TUB'          //------- available
    else if StructureType = 'Plate' then
       StructureType := 'PLA';         //------- not available

  CrackType := AStructure.GetCrackType;
    if CrackType = 'SemiEllCrack' then
       CrackType := '_SE';             //------- avilable

  Orientation := AStructure.GetOrientation;
    if Orientation = 'Longitudinal' then
       Orientation := 'LD'             //------- available
    else if Orientation = 'Circumferential' then
       Orientation := 'CD';            //------- available

  Position := AStructure.GetPosition;
    if Position = 'Internal' then
       Position := 'I'                 //------- available
    else if Position = 'External' then
       Position := 'E';                //------- not available
  Result := StructureType + CrackType  + Orientation + Position;
end;

function TAssessment.GetKI_a0: Real;
begin
  Result := FKI_a0;
end;

function TAssessment.GetKI_ag: Real;
begin
  Result := FKI_ag;
end;

procedure TAssessment.CalLimitLoad(AStructure: TStructure);
var  a0 : Real;
     ag : Real;
begin
  a0 := AStructure.GetOCDepth;    // Crack Depth a0
  ag := AStructure.GetCCDepth;    // Crack Depth ag

  if FLevel = 0 then
    begin
      if FGrowth = False then
        begin
          AStructure.SetCCDepth(a0);
          FPL_a0 := CheckAndCalPL(AStructure);
          AStructure.SetCCDepth(ag);
        end
      else
        begin
          AStructure.SetCCDepth(FaTotal);
```

```
          AStructure.SetCCLength(FcTotal);
          FPL_a0 := CheckAndCalPL(AStructure);
          AStructure.SetCCDepth(ag);             // Delta ag
        end;
    end
  else if FLevel = 1 then
    begin
      if FGrowth = False then
        begin
          AStructure.SetCCDepth(a0);
          FPL_a0 := CheckAndCalPL(AStructure);
          AStructure.SetCCDepth(ag + a0);

          FPL_ag := CheckAndCalPL(AStructure);
        end
      else
        begin
          AStructure.SetCCDepth(FaTotal);
          AStructure.SetCCLength(FcTotal);
          FPL_a0 := CheckAndCalPL(AStructure);
          AStructure.SetCCDepth(ag + FaTotal);  // Delta ag
          FPL_ag := CheckAndCalPL(AStructure);
        end;
    end;
end;

function TAssessment.CheckAndCalPL(AStructure: TStructure): Real;
var PL           : Real;
    SStatus      : string;
    LStatus      : string;
    LStatusN     : string;
    TotalStatus  : string;
    TotalStatusN : string;
begin
  SStatus     := StructureStatus(AStructure);
                              // Check status of CrackStructure
  LStatus     := LoadStatus(AStructure);
                              // Check status of load
  TotalStatus := SStatus + LStatus;  // Total status
  PL := LimitLoad.GetPL(AStructure, TotalStatus);
  Result := PL;
end;

function TAssessment.LoadStatus(AStructure: TStructure): string;
var i      : integer;
    Load   : Real;
    temp   : string;
    LStatus : string;
begin
  Load := 0;
  for i := 0 to 3 do                  // Check status of load
    begin
      if i = 0 then
        begin
          Load := AStructure.GetAxialTension;
          temp := '_ATL';
        end
      else if i = 1 then
```

```
        begin
          Load := AStructure.GetBending;
          temp := '_BDL';
        end
      else if i = 2 then
        begin
          Load := AStructure.GetInPressure;
          temp := '_IPL';
        end
      else if i = 3 then
        begin
          Load := AStructure.GetStress_Coeff[0];
          temp := '_SDL';
        end;
      if Load > 0 then
        begin
          LStatus := LStatus + temp;
        end;
    end;
  Result := LStatus;
end;

function TAssessment.GetLimitLoad_a0: Real;
begin
  Result := FPL_a0;
end;

function TAssessment.GetLimitLoad_ag: Real;
begin
  Result := FPL_ag;
end;

procedure TAssessment.CalKr(AStructure: TStructure);
begin
  if FLevel = 0 then
    begin
      FKr_a0 := FKI_a0/AStructure.GetKmat;
    end
  else if FLevel = 1 then
    begin
      FKr_a02 := FKI_a0/AStructure.GetK02;
      FKr_ag  := FKI_ag/AStructure.GetKg;
    end;
end;

function TAssessment.GetKr_a0: Real;
begin
  Result := FKr_a0;
end;

function TAssessment.GetKr_a02: Real;
begin
  Result := FKr_a02;
end;

function TAssessment.GetKr_ag: Real;
begin
  Result := FKr_ag;
```

```
end;

procedure TAssessment.CalLr(AStructure: TStructure);
var  LoadName: string;
     Load: Real;
begin
  if FPL_a0 <> 0 then
    begin
      Load := 0;
      LoadName := LoadStatus(AStructure);
      if LoadName = '_ATL' then
        Load := AStructure.GetAxialTension
      else if LoadName = '_BDL' then
        Load := AStructure.GetBending
      else if LoadName = '_IPL' then
        Load := AStructure.GetCInPressure
      else if LoadName = '_ATL_BDL' then
        begin
          if FBDL_ATL <= FATL_BDL then
            Load := AStructure.GetBending
          else
            Load := AStructure.GetAxialTension;
        end;

      if FLevel = 0 then
        begin
          FLr_a0 := Load/FPL_a0;
        end
      else if FLevel = 1 then
        begin
          if FPL_ag <> 0 then
            begin
              FLr_a0 := Load/FPL_a0;
              FLr_ag := Load/FPL_ag;
            end
          else
            begin
              FLr_a0 := 0;
              FLr_ag := 0;
            end;
        end;
    end
  else
    FLr_a0 := 0;
end;

function TAssessment.GetLr_a0: Real;
begin
  Result := FLr_a0;
end;

function TAssessment.GetLr_ag: Real;
begin
  Result := FLr_ag;
end;

procedure TAssessment.CalAssessPoint(Sender: TObject;
  AStructure: TStructure);
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
begin
  if FLevel = 0 then       // assessment point = 1
    FAssessResult := AssessPointLevel1(Sender, AStructure)
  else if FLevel = 1 then  // assessment point = 2
    FAssessResult := AssessPointLevel2(Sender, AStructure);
end;
                           //  Level 1 ; Option 1 or Option 2
function TAssessment.AssessPointLevel1(Sender: TObject;
                                       AStructure: TStructure):
String;
var LrMax   : Real;
    SFlow   : Real;
    Kr_LrMax: Real;
    dist    : Real;
    distC   : Real;
    AssessPointLevel1: String;
begin
  SFlow := (AStructure.GetSy + AStructure.GetSu)/2;
  LrMax := SFlow/AStructure.GetSy;

   if FLr_a0 < LrMax then
     begin
       Kr_LrMax := FADObj.GetKrFAD(self, LrMax, AStructure);
                                              //Kr and Lr at LrMax

// Cal Crittical Point @ Line LrMax
       if Kr_LrMax/LrMax > FKr_a0/FLr_a0 then
         begin
           FLrC  := LrMax;  // LrCL = LrMax
           FKrC  := (FKr_a0 * FLrC)/FLr_a0;
           dist  := sqrt(sqr(FKr_a0) + sqr(FLr_a0));
           distC := sqrt(sqr(FKrC) + sqr(FLrC));
           FFL   := distC/dist;
         end

// Cal crittical Point @ Assessment Curve
       else if Kr_LrMax/LrMax <= FKr_a0/FLr_a0 then
         begin
           FLrC  := CalRootLr(self, FKr_a0, FLr_a0, AStructure);
           FKrC  := FADObj.GetKrFAD(self, FLrC, AStructure);
           dist  := sqrt(sqr(FKr_a0) + sqr(FLr_a0));
           distC := sqrt(sqr(FKrC) + sqr(FLrC));
           FFL   := distC/dist;
         end;
     end
   else
     begin
       AssessPointLevel1 := ' NOT SAFE ';
     end;

   if FFL <= 1 then
     begin
       AssessPointLevel1 := ' NOT SAFE ';
     end
   else
     begin
       AssessPointLevel1 := ' SAFE ';
     end;
```

```
  Result := AssessPointLevel1;
end;

function TAssessment.GetKrC: Real;
begin
  Result := FKrC;
end;

function TAssessment.GetLrC: Real;
begin
  Result := FLrC;
end;

function TAssessment.GetFL: Real;
begin
  Result := FFL;
end;

function TAssessment.CalRootLr(Sender: TObject; AKrA, ALrA: Real;
                               AStructure: TStructure): Real;
var i     : integer;
    SFlow : Real;
    LrMax : Real;
    LrL   : Real;
    LrR   : Real;
    LrM   : Real;
    LrN   : Real;

    FLrL  : Real;
    FLrR  : Real;
    FLrM  : Real;

    AA    : Real;
    ES    : Real;
    TOL   : Real;

    RootLr: Real;
begin
  RootLr := 0;
  SFlow  := (AStructure.GetSy + AStructure.GetSu)/2;
  LrMax  := SFlow/AStructure.GetSy;

  LrL    := 0.01;
  LrR    := LrMax;
  ES     := 0.001;

  FLrL   := FADObj.GetKrFAD(self, LrL, AStructure) - Line(AKrA, ALrA,
            LrL);
  FLrR   := FADObj.GetKrFAD(self, LrR, AStructure) - Line(AKrA, ALrA,
            LrR);
  AA     := FLrL * FLrR;

  ShowMessage(' AA :  ' + floatToStr(AA));
  if AA >= 0 then
    begin
      ShowMessage('root is not in the given range');
    end;
      for i := 1 to 20 do
```

```
        begin    //2
          // check whether the root is in given range
          LrM  := (LrL + LrR)/2;
          FLrM := FADObj.GetKrFAD(self, LrM, AStructure) - Line(AKrA,
                  ALrA, LrM);
          FLrR := FADObj.GetKrFAD(self, LrR, AStructure) - Line(AKrA,
                  ALrA, LrR);
          AA   := FLrM * FLrR;

          if AA > 0 then  // case :  LrL < root < LrM
            LrR := LrM
          else            // case :  LrL < root < LrM
            LrL := LrM;
            LrN := (LrL + LrM)/2;
            TOL := ABS((LrN - LrR) * 100 / LrN);
          if TOL < ES then
            begin
              RootLr := LrN;
              Break;
            end;
        end;  //2
      Result := RootLr;
end;

function TAssessment.Line(AKrA, ALrA, ALr: Real): Real;
begin
  Result := (AKrA * (ALr - ALrA)/ALrA) + AKrA;
end;
                              //  Level 2 ; Option 1 or Option 2
function TAssessment.AssessPointLevel2(Sender: TObject;
                                       AStructure:TStructure): String;
var LrMax    : Real;
    SFlow    : Real;
    Kr_LrMax : Real;

    dist     : Real;
    distC    : Real;
    AssessPointLevel2: String;
begin
  FCaller := Sender;
  SFlow   := (AStructure.GetSy + AStructure.GetSu)/2;
  LrMax   := SFlow/AStructure.GetSy;

   if FLr_a0 < LrMax then
     begin  //1
       Kr_LrMax := FADObj.GetKrFAD(self, LrMax, AStructure);

// assessment point 1
// Kr02, Lr02    @ Kmat = K0.2
// Cal Crittical Point @ Line L
        if Kr_LrMax/LrMax > FKr_a02/FLr_a0 then
          begin
            FLrC  := LrMax;    //LrCL = LrMax
            FKrC  := (FKr_a02 * FLrC)/FLr_a0;
            dist  := sqrt(sqr(FKr_a02) + sqr(FLr_a0));
            distC := sqrt(sqr(FKrC) + sqr(FLrC));
            FFL   := distC/dist;
          end
```

```
// Cal crittical Point @ Assessment Curve
       else if Kr_LrMax/LrMax <= FKr_a02/FLr_a0 then
         begin
           FLrC  := CalRootLr(self, FKr_a02, FLr_a0, AStructure);
           FKrC  := FADObj.GetKrFAD(self, FLrC, AStructure);
           dist  := sqrt(sqr(FKr_a02) + sqr(FLr_a0));
           distC := sqrt(sqr(FKrC) + sqr(FLrC));
           FFL   := distC/dist;
         end;

// assessment point 2
// Krg, Lrg      @ Kmat = Kg
// Cal Crittical Point @ Line LrMax
       if Kr_LrMax/LrMax > FKr_ag/FLr_ag then
         begin
           FLrC_g := LrMax;   //LrCL = LrMax
           FKrC_g := (FKr_ag * FLrC_g)/FLr_ag;
           dist   := sqrt(sqr(FKr_ag) + sqr(FLr_ag));
           distC  := sqrt(sqr(FKrC_g) + sqr(FLrC_g));
           FFL_g  := distC/dist;
         end

// Cal crittical Point @ Assessment Curve
       else if Kr_LrMax/LrMax <= FKr_ag/FLr_ag then
         begin
           FLrC_g := CalRootLr(self, FKr_ag, FLr_ag, AStructure);
           FKrC_g := FADObj.GetKrFAD(self, FLrC_g, AStructure);
           dist   := sqrt(sqr(FKr_ag) + sqr(FLr_ag));
           distC  := sqrt(sqr(FKrC_g) + sqr(FLrC_g));
           FFL_g  := distC/dist;
         end;
     end  //1
   else
     begin
       AssessPointLevel2 := ' NOT SAFE ';
     end;

   if (FFL > 1) then   // Check FL
     begin
       AssessPointLevel2 := ' SAFE ';
     end
   else
     begin
       AssessPointLevel2 := ' NOT SAFE ';
     end;
  Result := AssessPointLevel2;
end;

function TAssessment.GetKrC_g: Real;
begin
  Result := FKrC_g;
end;

function TAssessment.GetLrC_g: Real;
begin
  Result := FLrC_g;
end;
```

```
function TAssessment.GetFL_g: Real;
begin
  Result := FFL_g;
end;

function TAssessment.GetAssessmentResult: string;
begin
  Result := FAssessResult;
end;

procedure TAssessment.CalCrackGrowth(AStructure: TStructure);
var N                              : integer;
    i                              : integer;
    a0, ag, a, da, aN, aCrit       : Real;
    twoc0, twoc, dc, twocN         : Real;
    FatigueLoad                    : Real;
    SLoad, FLoadMax, FLoadMin, DeltaLoad    : Real;
    DeltaKIA                       : Real;
    DeltaKIC                       : Real;
    SStatus, LStatus, TotalStatus: string;
    K11, K22                       : Real;
    K1, K02, Kg                    : Real;
    Ro, Ri, t                      : Real;
begin
  SStatus := StructureStatus(AStructure);    // check load status
  SLoad   := AStructure.GetInPressure;       // static load
  FatigueLoad := AStructure.GetInPressureMax;
  if FatigueLoad > 0 then
                              // Fatigue load : internal pressure only
    LStatus := '_IP'
  else
    LStatus := '';

{  FatigueLoad := AStructure.GetAxialTensionMax;
  if FatigueLoad > 0 then
    LStatus := '_AT'
  else
    LStatus := '';   }

    TotalStatus := SStatus + LStatus;

    a0    := AStructure.GetOCDepth;
    ag    := AStructure.GetCCDepth;

    twoc0 := AStructure.GetCCLength;          // 2C

    Ri    := AStructure.Dimension1;
    Ro    := AStructure.Dimension2;
    t     := Ro - Ri;

    aCrit := 0.8*t;
    FCycleC := 0;

    AStructure.SetCCDepth(a0);
    AStructure.SetCCLength(twoc0);
    N := CrackGrowthObj.GetLoop;

    FLoadMax := AStructure.GetInPressureMax;     // fatigue load max
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
  FLoadMin := AStructure.GetInPressureMin;     // fatigue load min
  DeltaLoad := FLoadMax - FLoadmin;
  AStructure.SetCInPressure(DeltaLoad);

if LStatus = '_IP' then
  begin
    for i := 0 to N - 1 do
      begin
        DeltaKIA := SIFObj.GetKI(AStructure, TotalStatus, 90);
        DeltaKIC := SIFObj.GetKI(AStructure, TotalStatus, 0);

        a        := AStructure.GetCCDepth;
        da       := CrackGrowthObj.GetDa(AStructure, DeltaKIA);
        aN       := a + da;

        twoc     := AStructure.GetCCLength;                  // 2C
        dc       := CrackGrowthObj.GetDc(AStructure, DeltaKIC);// C
        twocN    := twoc + (2 * dc);                         // 2C

        AStructure.SetCCDepth(aN);
        AStructure.SetCCLength(TwocN);
        AStructure.SetCInPressure(SLoad);
        K11 := SIFObj.GetKI(AStructure, TotalStatus, 90);
        K22 := SIFObj.GetKI(AStructure, TotalStatus, 0);

        if FLevel = 0 then
          begin
            K1 := AStructure.GetKmat;
              if (K11 >= K1) or (aN >= aCrit) then
                                        // check K @ Deep point
                begin
                  FaTotal := aN - da;
                  FcTotal := twocN;
                  FCycleC := i;
                  Break;
                end
              else if K22 >= K1 then     // check K @ near surface
                begin
                  FaTotal := aN;
                  FcTotal := twocN - (2 * dc);
                  FCycleC := i;
                  Break;
                end
              else
                begin
                  FaTotal := aN;
                  FcTotal := twocN;
                end;
          end
        else if FLevel = 1 then
          begin
            K02 := AStructure.GetK02;
            Kg  := AStructure.GetKg;    // check K @ Deep point
            if (K11 >= K02) or (K11 >= Kg) or (aN >= aCrit) then
              begin
                FaTotal := aN - da;
                FcTotal := twocN;
                FCycleC := i;
```

```
                Break;
              end
            else if (K22 >= K02) or (K22 >= Kg) then
                                        // check K @ near surface
              begin
                FaTotal := aN;
                FcTotal := twocN - (2 * dc);
                FCycleC := i;
                Break;
              end
            else

              begin
                FaTotal := aN;
                FcTotal := twocN;
                //FCycleC := i;
              end;
          end;
          AStructure.SetCCDepth(aN);
          AStructure.SetCCLength(twocN);
          AStructure.SetCInPressure(DeltaLoad);
        end;      // end Loop of cyclic
       AStructure.SetCCDepth(ag);
       AStructure.SetCCLength(twoc0);

       AStructure.SetCInPressure(SLoad);
       FGrowth := True;
    end;
end;

function TAssessment.GetCrackDepthNew: Real;
begin
  Result := Fatotal;
end;

function TAssessment.GetCrackLengthNew: Real;
begin
  Result := FcTotal;
end;

function TAssessment.GetMechanism: string;
begin
  if assigned(CrackGrowthObj) then
    Result := CrackGrowthObj.GetMechanism
  else
    Result := '';
end;

function TAssessment.GetLoop: integer;
begin
  if assigned(CrackGrowthObj) then
    Result := CrackGrowthObj.GetLoop
  else
    Result := 0;
end;

function TAssessment.GetNODirector: integer;
begin
```

```
  if assigned(CrackGrowthObj) then
    Result := CrackGrowthObj.GetNODirector
  else
    Result := 0;
end;

function TAssessment.GetFCycleC: integer;
begin
  Result := FCycleC;
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.17 ซอร์สโค้ดของคลาส limitLoad

```
unit LimitLoadsU;

interface

Uses Math, StructureU;

Type
  TLimitLoad = class(TObject)
  private
// Cercumferential
    function TUB_SECDI_ATL(AStructure: TStructure): Real;
    function TUB_SECDI_BDL(AStructure: TStructure): Real;
    function TUB_SECDI_BDL_ATL(AStructure: TStructure): Real;
             // limit bending moment
// Longitudinal
    function TUB_SELDI_IPL(AStructure: TStructure): Real;
  public
    function GetPL(AStructure: TStructure; AIndex: string): Real;
  end;

implementation

{ LimitLoads }

//----------------------Cal limit load--------------------------//
function TLimitLoad.GetPL(AStructure: TStructure; AIndex: string):
Real;
var  Ri, Ro, t, R  : Real;  // Pipe dimension
     P             : Real;  // Load : Axial tension
     sigmaT, sigmaY,
     sigmaU, sigmaF: Real;  // Material properties
     PL            : Real;  // Limit load : Limit tension
     RoperT        : Real;  // Ro/t
     RperT         : Real;  // R/t
begin
  Ri     := AStructure.Dimension1;
  Ro     := AStructure.Dimension2;
  t      := Ro - Ri;
  R      := Ri + (t/2);
  RoperT := Ro/t;
  RperT  := R/t;
  PL     := 0;

  if AIndex = 'TUB_SECDI_ATL' then             // limit axial tension
    begin
      if (5 <= RoperT) and (RoperT <= 20) then
        PL := TUB_SECDI_ATL(AStructure)
      else
        PL := 0;
    end
  else if AIndex = 'TUB_SECDI_BDL' then       // limit bending moment
    begin
      PL := TUB_SECDI_BDL(AStructure);
    end
  else if AIndex = 'TUB_SECDI_BDL_ATL' then   // limit bending moment
```

```
    begin
      P      := AStructure.GetAxialTension;
      sigmaT := P/(2*Pi*R*t);
      sigmaY := AStructure.GetSy;
      sigmaU := AStructure.GetSu;
      sigmaF := (sigmaY + sigmaU)/2;
      if (RperT >= 10) and (sigmaT/sigmaF < 0.25) then
        PL   := TUB_SECDI_BDL_ATL(AStructure)
      else
        PL   := 0;
    end
  else if AIndex = 'TUB_SELDI_IPL' then    // limit internal pressure
    begin
      PL := TUB_SELDI_IPL(AStructure);
    end;
  Result := PL;
end;

///----------------------TUB_SECDI_ATL---------------------------///
function TLimitLoad.TUB_SECDI_ATL(AStructure: TStructure): Real;
var  Ri, Ro, t, R  : Real;       // Pipe dimension
     twoc, c, a    : Real;       //Crack size
     sigmay, sigmaU,
     sigmaF        : Real;       // Material properties
                                 //No Loads
     si            : Real;
     zeta          : Real;
     x             : Real;
     A1            : Real;
     alpha         : Real;
begin
  Ri     := AStructure.Dimension1;
  Ro     := AStructure.Dimension2;
  t      := Ro - Ri;
  R      := Ri + t/2;

  sigmaY := AStructure.GetSy;
  sigmaU := AStructure.GetSu;
  sigmaF := (sigmaY + sigmaU)/2;

  twoc   := AStructure.GetCCLength;
  c      := twoc/2;
  a      := AStructure.GetCCDepth;

  si     := t/Ro;
  zeta   := (Pi*c)/(4*Ri);
  x      := a/t;
  A1     := x*((1 - si)*(2 - 2*si + x*si) + sqr(1 - si + x*si))/
            (2*(1 + (2 - si)*(1 - si)));
  alpha  := arccos(A1*sin(zeta));
  Result := 2*Pi*R*t*sigmaF*((2*alpha/Pi) - (x*zeta/Pi)*(2 - 2*si +
            x*si)/(2 - si));
end;


///-----------------TUB_SECDI_BDL_ATL---------------------------///
function TLimitLoad.TUB_SECDI_BDL_ATL(AStructure: TStructure): Real;
var  Ri, Ro, t, R   : Real;      // Pipe dimension
     twoc, c, a     : Real;      // Crack Properties
```

```
      sigmaY, sigmaU,
      sigmaF          : Real;      // Material Properties
      P               : Real;      // Load Axial Tension
      sigmaT          : Real;      // Stress
      zeta            : Real;
      beta            : Real;
      ML              : Real;      // Limit load : Limit moment
begin
  Ri     := AStructure.Dimension1;
  Ro     := AStructure.Dimension2;
  t      := Ro - Ri;
  R      := Ri + t/2;

  sigmaY := AStructure.GetSy;
  sigmaU := AStructure.GetSu;
  sigmaF := (sigmaY + sigmaU)/2;

  twoc   := AStructure.GetCCLength;
  c      := twoc/2;
  a      := AStructure.GetCCDepth;

  P      := AStructure.GetAxialTension;

  sigmaT := P/(2*Pi*R*t);
  zeta   := (Pi*c)/(4*Ri);
  beta   := 0.5*Pi*(1 - (zeta/Pi)*(a/t) - (sigmaT/sigmaF));

  if (zeta + beta) > Pi then
    begin
      beta := Pi*(1 - (a/t) - (sigmaT/sigmaF)/(2 - (a/t)));
      ML   := 2*sigmaF*sqr(R)*t*(2 - a/t)*sin(beta);
    end
  else
    ML     := 2*sigmaF*sqr(R)*t*(2*sin(beta) - (a/t)*sin(zeta));
  Result   := ML;              // limit bending moment
end;

//---------------------TUB_SECDI_BDL-----------------------------//
function TLimitLoad.TUB_SECDI_BDL(AStructure: TStructure): Real;
var  Ri, Ro, t, R   : Real;        // Pipe dimension
     twoc, c, a     : Real;        // Crack size
     sigmaY, sigmaU,
     sigmaF         : Real;        // Material Properties
     P              : Real;        // Load : Axial Tension

     sigmaT         : Real;
     zeta           : Real;
     beta           : Real;
     PL             : Real;      // Limit Load : Limit moment
begin
  Ri     := AStructure.Dimension1;
  Ro     := AStructure.Dimension2;
  t      := Ro - Ri;
  R      := Ri + t/2;

  sigmaY := AStructure.GetSy;
  sigmaU := AStructure.GetSu;
  sigmaF := (sigmaY + sigmaU)/2;
```

```
  twoc   := AStructure.GetCCLength;
  c      := twoc/2;
  a      := AStructure.GetCCDepth;
  P      := 0;
  sigmaT := P/(2*Pi*R*t);
  zeta   := (Pi*c)/(4*Ri);
  beta   := 0.5*Pi*(1 - (zeta/Pi)*(a/t) - (sigmaT/sigmaF));

  if (zeta + beta) > Pi then
    begin
      beta := Pi*(1 - (a/t) - (sigmaT/sigmaF)/(2 - (a/t)));
      PL   := 2*sigmaF*sqr(R)*t*(2 - a/t)*sin(beta);
    end
  else
     PL    := 2*sigmaF*sqr(R)*t*(2*sin(beta) - (a/t)*sin(zeta));

  Result := PL;
end;

///------------------TUB_SELDI_IPL-----------------------------///
function TLimitLoad.TUB_SELDI_IPL(AStructure: TStructure): Real;
var  Ri, Ro, t, R   : Real;    // Pipe dimension
     twoc, c, a     : Real;    // Crack size
     sigmaY, sigmaU,
     sigmaF         : Real;    // Material Properties
                               //--- No Loads ---//
     x              : Real;
     M              : Real;
     Po             : Real;    // stress
begin
  Ri     := AStructure.Dimension1;
  Ro     := AStructure.Dimension2;
  t      := Ro - Ri;
  R      := Ri + t/2;

  sigmaY := AStructure.GetSy;
  sigmaU := AStructure.GetSu;
  sigmaF := (sigmaY + sigmaU)/2;

  twoc   := AStructure.GetCCLength;
  c      := twoc/2;
  a      := AStructure.GetCCDepth;

  x      := a/t;
  M      := sqrt(1 + 1.61*sqr(c)/(R*t));
  Po     := sigmaF*(t/Ro);

  Result := Po*((1 - x)/(1 - x/M));
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.18 ซอร์สโค้ดของคลาส SIF

```
unit CrackTPMU;

interface

Uses Math, StructureU, Windows, Messages, SysUtils,
     Variants,   Classes, Graphics, Controls,   Forms,
     Dialogs,  StdCtrls, Buttons, ExtCtrls, ComCtrls;

Type
  TCrackTPM = class(TObject)
  protected

  public
    function GetKI(AStructure: TStructure; AIndex: string;
                   APhi: Real): Real; virtual; abstract;
  end;   // end  TCrackTPM = class(TObject)

  TKI = class(TCrackTPM)
  private
// Circumferential         // for Deepest & surface points
    function TUB_SECDI_AT(AStructure: TStructure; APhi: Real): Real;
    function TUB_SECDI_BD(AStructure: TStructure): Real;

// Loangitudinal
    function TUB_SELDI_IP(AStructure: TStructure; APhi: Real): Real;
    function TUB_SELDI_SD(AStructure: TStructure): Real;

  public
    function GetKI(AStructure: TStructure; AIndex: string;
                   APhi: Real): Real; override;
  end;  // end TKI = class(TCrackTPM)

implementation

{ TKI }

function TKI.GetKI(AStructure: TStructure; AIndex: string;
                   APhi: Real): Real;
var Ri, Ro, t   : Real;
                  // inner radius, outer radius and wall thickness
    c           : Real;    // half surface length
    a           : Real;    // Crack depth
    twoC        : Real;    // 2c
    AperC       : Real;    // a/c
    AperT       : Real;    // a/t
    TwoCperA    : Real;    // 2c/a
    KI          : Real;    // Mode 1 Stress intensity factor
begin
  Ri       := AStructure.Dimension1;
  Ro       := AStructure.Dimension2;
  t        := Ro - Ri;

  twoC     := AStructure.GetCCLength;
  c        := twoC/2;
```

```
a        := AStructure.GetCCDepth;
KI       := 0;

AperC    := a/c;
AperT    := a/t;
TwoCperA := TwoC/a;

if AIndex = 'TUB_SECDI_AT' then
  begin
    if (0.05 <= AperT) and (AperT <= 0.8) then
      begin
        if 3 <= TwoCperA then
          begin
            KI := TUB_SECDI_AT(AStructure, APhi);
          end
        else
          begin
            ShowMessage('2C/a : 3 <= 2C/a' + #13#10 +
                        '2C/a = ' + FloatToStr(TwoCperA) +
                        'out of range');
          end;
      end
    else
      begin
        ShowMessage('a/t : 0.05 <= a/t <= 0.8' + #13#10 +
                    'a/t = ' + FloatToStr(AperT) + 'out of range');
      end;
  end
else if AIndex = 'TUB_SECDI_BD' then
  begin
    if (0.05 <= AperT) and (AperT <= 0.8) then
      begin
        if 3 <= TwoCperA then
          begin
            KI := TUB_SECDI_BD(AStructure);
          end
        else
          begin
            ShowMessage('2C/a : 3 <= 2C/a' + #13#10 +
                        '2C/a = ' + FloatToStr(TwoCperA) +
                        'out of range');
          end;
      end
    else
      begin
        ShowMessage('a/t : 0.05 <= a/t <= 0.8' + #13#10 +
                    'a/t = ' + FloatToStr(AperT) + 'out of range');
      end;
  end
else if AIndex = 'TUB_SELDI_IP' then
  begin
    if (0.05 <= AperT) and (AperT <= 0.85) then
      begin
        if (0.1 <= AperC) and (AperC <= 1) then
          begin
            KI := TUB_SELDI_IP(AStructure, APhi);
          end
        else
          begin
```

```
              ShowMessage('a/c : 0.1 <= a/c <= 1' + #13#10 +
                          'a/c = ' + FloatToStr(AperC) +
                          'out of range');
            end;
        end
      else
        begin
          ShowMessage('a/t : 0.05 <= a/t <= 0.85' + #13#10 +
                      'a/t = ' + FloatToStr(AperT) + 'out of range');
        end;
    end
  else if AIndex = 'TUB_SELDI_SD' then
    begin
      if (0.05 <= AperT) and (AperT <= 0.85) then
        begin
          if (0.1 <= AperC) and (AperC <= 1) then
            begin
              KI := TUB_SELDI_SD(AStructure);
            end
          else
            begin
              ShowMessage('a/c : 0.1 <= a/c <= 1' + #13#10 +
                          'a/c = ' + FloatToStr(AperC) +
                          'out of range');
            end;
        end
      else
        begin
          ShowMessage('a/t : 0.05 <= a/t <= 0.85' + #13#10 +
                      'a/t = ' + FloatToStr(AperT) + 'out of range');
        end;
    end;

  Result := KI;
end;

///-------------TUB_SECDI_AT------for  Deepest point ------///

function TKI.TUB_SECDI_AT(AStructure: TStructure; APhi: Real): Real;
var Ri, Ro, R, t : Real;  // Pipe Dimension
    twoC, c, a   : Real;  // Crack size
    P            : Real;  // Load : Axial tension
    Beta         : Real;
    Ft, Fst      : Real;  // Shape factor
    sigmaT       : Real;  // Stress
    KI           : Real;  // Stress intensity factor
begin
  Ri     := AStructure.Dimension1;
  Ro     := AStructure.Dimension2;
  t      := Ro - Ri;
  R      := Ri +t/2;
  twoC   := AStructure.GetCCLength;
  c      := twoc/2;
  a      := AStructure.GetCCDepth;
  P      := AStructure.GetAxialTension;

  sigmaT := P/(2*Pi*R*t);
  Beta   := (a/t)/power((0.25 + a/c),0.58);
  Ft     := 0;
```

```
  KI     := 0;

  if (Beta < 0.25) and (5 <= R/t) and (R/t <= 20) then
    Ft := 3.702*Beta - 13.475*sqr(Beta) + 20.0*Power(Beta, 3) +
          0.0086*Beta*((R/t) - 5)

  else if (Beta < 0.25) and (20 <= R/t) and (R/t <= 160) then
    Ft := 3.831*Beta - 13.475*sqr(Beta) + 20.0*Power(Beta,3) +
          0.002*Beta*Power((R/t) - 20, 0.7)

  else if (Beta >= 0.25) and (5 <= R/t) and (R/t <= 20) then
    Ft := 0.25 + 0.4868*Beta + 0.3835*sqr(Beta) +
          0.0086*Beta*(R/t - 5)

  else if (Beta >= 0.25) and (20 < R/t) and (R/t <= 160) then
    Ft := 0.25 + 0.6158*Beta + 0.3835*sqr(Beta) +
          0.002*Beta*Power(R/t - 20, 0.7);

  if APhi = 0 then          // Point B : for surface point
    begin
      Fst := (0.5354 + 0.123*(a/t))*Ft;
      KI  := sigmaT*sqrt(pi*t)*Fst;
    end
  else if APhi = 90 then    // Point A : for maximum deep point
    begin
      KI  := sigmaT*sqrt(pi*t)*Ft;
    end;
  Result  := KI;
end;

///---------- TUB_SECDI_BD --- for  deepest point------------///
function TKI.TUB_SECDI_BD(AStructure: TStructure): Real;
var Ri, Ro, R, t : Real;    // Pipe Dimension
    twoC, c, a   : Real;    // Crack size
    M            : Real;    // Load magnitude: bending moment
    zeta         : Real;
    sigmaB       : Real;    // Stress
    B00, B01,
    B02, B10,
    B11, B12,
    B20, B21,
    B22, A0,
    A1,  A2,
    Fb           : Real;    // Shape factor
begin
  Ri   := AStructure.Dimension1;
  Ro   := AStructure.Dimension2;
  t    := Ro - Ri;
  R    := Ri + t/2;
  twoC := AStructure.GetCCLength;
  c    := twoC/2;
  a    := AStructure.GetCCDepth;
  M    := AStructure.GetBending;

  zeta := (Pi*c)/(4*Ri);

 if (zeta/Pi <= 1) then
   begin
     sigmaB := M/(Pi*t*sqr(R));
```

```
      B00 := -0.0002*sqr(R/t)  + 0.0173*(R/t) + 3.3567;
      B01 := -0.00006*sqr(R/t) + 0.0101*(R/t) + 1.3153;
      B02 :=  0.00002*sqr(R/t) - 0.0019*(R/t) - 0.0066;

      B10 :=  0.000006*sqr(R/t) - 0.0017*(R/t) - 4.2841;
      B11 :=  0.00006*sqr(R/t)  - 0.013*(R/t)  - 1.1867;
      B12 := -0.000002*sqr(R/t) + 0.0004*(R/t) - 0.0741;

      B20 := -0.0264*Ln(R/t) + 1.7928;
      B21 :=  0.147*Ln(R/t)  + 0.7238;
      B22 :=  0.0058*Ln(R/t) + 0.2269;

      A0  := B00*sqr(a/t) + B01*(a/t) + B02;
      A1  := B10*sqr(a/t) + B11*(a/t) + B12;
      A2  := B20*sqr(a/t) + B21*(a/t) + B22;

      Fb  := A0*sqr(a/c) + A1*(a/c) + A2;

      Result := sigmaB*Fb*sqrt(Pi*t);
    end
    else
      Result := 0;
end;

///------------TUB_SELDI_IP----for Deepest point -----------------///
function TKI.TUB_SELDI_IP(AStructure: TStructure; APhi: Real): Real;
var  Ro, Ri, t  : Real;   // Pipe dimension
     twoC, c, a : Real;   // Crack size
     IP         : Real;   // Load : Internal pressure
     sigmaH     : Real;   // Stress
     alpha      : Real;
     F, Fs      : Real;   // Shape factor
     KI         : Real;   // Stress intensity factor
begin
  Ri     := AStructure.Dimension1;
  Ro     := AStructure.Dimension2;
  t      := Ro - Ri;
  twoC   := AStructure.GetCCLength;
  c      := twoC/2;
  a      := AStructure.GetCCDepth;
  IP     := AStructure.GetCInPressure;

  KI     := 0;    // initial of stress intensity factor equal zero
  sigmaH := IP*(sqr(Ro) + sqr(Ri))/(sqr(Ro) - sqr(Ri));
  alpha  := (a/t)/power(a/c,0.58);
  F      := 0.25 + (0.4759*alpha + 0.1262*sqr(alpha))/
            (power(0.102*(Ri/t) - 0.02, 0.1));

  if alpha >= 0.2 then
    begin
      if APhi = 0 then        // Point B : for surface point
        begin
          Fs := F*(1.06 + 0.28*sqr(a/t))*power(a/c, 0.41);
          KI := sigmaH*power(Pi*t, 0.5)*Fs;
        end
      else if APhi = 90 then   // Point A : for maximum deep point
        begin
          KI := sigmaH*power(Pi*t, 0.5)*F;
```

```
        end;
    end
  else
    KI := 0;

  Result := KI;
end;


///----------------TUB_SELDI_SD –for deepest point   -------------///
function TKI.TUB_SELDI_SD(AStructure: TStructure): Real;
var i,j         : integer;
    Ri, Ro, t   : Real;                       // Pipe dimension
    twoC, c, a  : Real;                       // Crack size
    alpha       : array[0..3] of Real;
    G           : array[0..3] of Real;
    Stress_Coeff: array[0..3] of Real;        // Stress coefficient
    GA          : array[0..4, 0..6] of Real;
    GsA         : array[0..3, 0..2] of Real;
                                             // for point near surface
    SUM         : Real;
    SUMTotal    : Real;
begin
  Ri       := AStructure.Dimension1;
  Ro       := AStructure.Dimension2;
  t        := Ro - Ri;
  twoC     := AStructure.GetCCLength;
  c        := twoC/2;
  a        := AStructure.GetCCDepth;

  GA[0,0]  :=  0.00;
  GA[0,1]  :=  1.7767;
  GA[0,2]  := -2.5975;
  GA[0,3]  :=  2.7520;
  GA[0,4]  := -1.3237;
  GA[0,5]  :=  0.2363;
  GA[0,6]  :=  0.58;

  GA[1,0]  := 0.02;
  GA[1,1]  := 0.5000;
  GA[1,2]  := 0.0962;
  GA[1,3]  := 0.00;
  GA[1,4]  := 0.00;
  GA[1,5]  := 0.00;
  GA[1,6]  := 0.58;

  GA[2,0]  := 0.00;
  GA[2,1]  := 0.1045;
  GA[2,2]  := 0.4189;
  GA[2,3]  := 0.00;
  GA[2,4]  := 0.00;
  GA[2,5]  := 0.00;
  GA[2,6]  := 0.22;

  GA[3,0]  :=  0.00;
  GA[3,1]  :=  0.02038;
  GA[3,2]  := -0.00397;
  GA[3,3]  :=  0.42126;
  GA[3,4]  :=  0.00;
```

```
  GA[3,5]  :=  0.00;
  GA[3,6]  :=  0.10;

  GA[4,0]  :=  0.00;
  GA[4,1]  :=  0.07283;
  GA[4,2]  := -0.36006;
  GA[4,3]  :=  0.66883;
  GA[4,4]  :=  0.00;
  GA[4,5]  :=  0.00;
  GA[4,6]  :=  0.05;

// for point near surface
  GsA[0,0] := 1.06;
  GsA[0,1] := 0.28;
  GsA[0,2] := 0.41;

  GsA[1,0] := 0.25;
  GsA[1,1] := 0.20;
  GsA[1,2] := 0.26;

  GsA[2,0] := 0.07;
  GsA[2,1] := 0.16;
  GsA[2,2] := 0.06;

  GsA[3,0] := 0.085;
  GsA[3,1] := 0.02;
  GsA[3,2] := 0.00;

  SUMTotal := 0;
  i        := 0;
  j        := 0;
  while i <> 4 do
    while j <> 5 do
      begin
        Stress_Coeff[i] := AStructure.GetStress_Coeff[i];
        alpha[i]        := (a/t)/power(a/c,GA[j,6]);
        if i = 0 then    // for  i = 0
          begin
            if not (a/c < 0.2) and (alpha[i] > 2) then
              begin
                G[i] := GA[j,0] + (GA[j,1]*alpha[i] +
                        GA[j,2]*power(alpha[i],2)+
                        GA[j,3]*power(alpha[i],3) +
                        GA[j,4]*power(alpha[i],4) +
                        GA[j,5]*power(alpha[i],5)) /
                        power((0.102*(Ri/t) - 0.02),0.05);
              end
            else
              begin
                G[i] := GA[1,0] + (GA[1,1]*alpha[i] +
                        GA[1,2]*power(alpha[i],2)+
                        GA[1,3]*power(alpha[i],3) +
                        GA[1,4]*power(alpha[i],4) +
                        GA[1,5]*power(alpha[i],5)) /
                        power((0.102*(Ri/t) - 0.02),0.05);
              end;
            j := 1;
          end
        else            // for i <> 0
```

```
            begin
              G[i] := GA[j,0] + (GA[j,1]*alpha[i] +
                       GA[j,2]*power(alpha[i],2) +
                      GA[j,3]*power(alpha[i],3) +
                       GA[j,4]*power(alpha[i],4) +
                      GA[j,5]*power(alpha[i],5)) /
                      power((0.102*(Ri/t) - 0.02),0.05);
            end;
          SUM      := Stress_Coeff[i]*G[i];
          SUMTotal := SUMTotal + SUM;
          i        := i + 1;
          j        := j + 1;
        end;
      Result := sqrt(Pi*t)*SUMToTal;

end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.19 ซอร์สโค้ดของคลาส FAD

```
unit R6FADU;

interface

Uses Math, StructureU;

type
  TR6FAD = class(TObject)
  private
  public
    function GetFADName: string;
    function GetKrFAD(Secder: TObject; ALr: Real;
                      AStructure: TStructure): Real;
                      virtual; abstract;
  end;

  TOption1 = class(TR6FAD)
  public
    function GetKrFAD(Secder: TObject; ALr: Real;
                      AStructure: TStructure): Real;
                      override;
  end;

  TOption2 = class(TR6FAD)
  public
    function GetKrFAD(Sender: TObject; ALr: Real;
                      AStructure: TStructure): Real;
                      override;
  end;

implementation

uses AssessmentU;     // Call back

{ TR6FAD }

function TR6FAD.GetFADName: string;
begin
  Result := self.ClassName;
  Delete(Result,1,1);
end;

{ TOption1 }

function TOption1.GetKrFAD(Secder: TObject; ALr: Real;
                           AStructure: TStructure): Real;
begin
  Result := (1 - 0.14*ALr*ALr)*(0.3 + 0.7*Exp(-0.65*Power(ALr,6)));
end;

{ TOption2 }

function TOption2.GetKrFAD(Sender: TObject; ALr: Real;
                           AStructure: TStructure): Real;
var E,Sy     : Real;
```

```
    StrainRef: Real;
begin
  E         := AStructure.GetSE;
  Sy        := AStructure.GetSy;
  StrainRef := (Sender as TAssessment).GetStrainRef(ALr);
  Result    := power((E*StrainRef/(ALr*Sy)) +
               (power(ALr,3)*Sy/(2*E*StrainRef)),-1/2);
end;

end.
```


ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.20 ซอร์สโค้ดของคลาส crackGrowth

```
unit CrackGrowthU;

interface

uses Math, StructureU;

type
  TCrackGrowth = class(TObject)
  private

  Protected

  public
    function GetLoop: integer;       virtual; abstract;
    function GetDa(AStructure: TStructure; ADeltaKI: Real): Real;
                                     virtual; abstract;
    function GetDc(AStructure: TStructure; ADeltaKI: Real): Real;
                                     virtual; abstract;
    function GetMechanism : string;
    function GetNODirector: integer; virtual; abstract;
  end;

  TFatigue = class(TCrackGrowth)
  private
    FNoCycle   : integer;
    FNoDirector: integer;
  public
    constructor create(ALoop, ADirector: integer);
    function    GetLoop         : integer; override;
    function    GetNODirector   : integer; override;
    function    GetDa(AStructure: TStructure; ADeltaKI: Real):
                      Real; override;
    function    GetDc(AStructure: TStructure; ADeltaKI: Real):
                      Real; override;
  end;

implementation

{ TFatigue }

constructor TFatigue.create(ALoop, ADirector: integer);
begin
  FNoCycle    := ALoop;
  FNoDirector := ADirector;
end;

function TFatigue.GetDa(AStructure: TStructure; ADeltaKI:
                        Real): Real;
var C  : Real;
    m  : Real;
    da : Real;
begin
  C      := AStructure.GetCoeffC;
  m      := AStructure.GetExponentM;
  da     := C*Power(ADeltaKI/1E06,m);
```

```
  Result :=  da;
end;

function TFatigue.GetDc(AStructure: TStructure; ADeltaKI:
                        Real): Real;
var C  : Real;
    m  : Real;
    dc : Real;
begin
  C      := AStructure.GetCoeffC;
  m      := AStructure.GetExponentM;
  dc     := power(0.9,m)*C*Power(ADeltaKI/1E06,m);
  Result :=  dc;
end;

function TFatigue.GetLoop: integer;
begin
  Result := FNoCycle;
end;

function TFatigue.GetNODirector: integer;
begin
  Result := FNoDirector;
end;

{ TCrackGrowth }

function TCrackGrowth.GetMechanism: string;
begin
  Result := Self.ClassName;
  Delete(Result,1,1);
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค.21ซอร์สโค้ดของคลาส constitutive

```
unit ConstitutiveU;

interface

Uses Math, StructureU;

type
  TConstitutive = class(TObject)
  private
    FSy       : Real;
    FSU       : Real;
    FSE       : Real;
    FAlpha    : Real;
    FExponentN: Real;
  public
    constructor Create(AStructure: TStructure);
    function    GetStrainRef_Romberg(ALr: Real): Real;
  end;
implementation

{ TConstitutive }

constructor TConstitutive.Create(AStructure: TStructure);
begin
  FSy        := AStructure.GetSy;
  FSu        := AStructure.GetSu;
  FSE        := AStructure.GetSE;
  FAlpha     := AStructure.GetAlpha;
  FExponentN := AStructure.GetExponentN;
end;

function TConstitutive.GetStrainRef_Romberg(ALr: Real): Real;
var
     StrainO : Real;
     StressO : Real;
begin
  StressO   := FSy;
  StrainO   := StressO/FSE;
  Result    := StrainO*((ALr*FSy/StressO) +
               (FAlpha*power(ALr*FSy/StressO,FExponentN)));
end;

end.
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายเนติพันธ์ พุทธรักษ์ เกิดเมื่อวันที่ 14 เมษายน 2524 ที่อำเภอพระพรหม จังหวัดนครศรีธรรมราช สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต จากภาควิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เมื่อปีการศึกษา 2547 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่ภาควิชาวิศวกรรมเครื่องกล จุฬาลงกรณ์มหาวิทยาลัยเมื่อปีการศึกษา 2548

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย