

การกำหนดค่าความคลาดเคลื่อนที่เหมาะสมโดยใช้ Rooted-Tree Graph

นางสาวอรรณพ นงนุช

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมอุตสาหการ

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2554

**OPTIMAL TOLERANCE ALLOCATION VIA A
ROOTED-TREE GRAPH**

Orawan Nongnuch

**A Thesis submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Industrial Engineering
Suranaree University of Technology
Academic Year 2011**

อรรณ นนุช : การกำหนดค่าความคลาดเคลื่อนที่เหมาะสมโดยใช้ Rooted-Tree Graph
(OPTIMAL TOLERANCE ALLOCATION VIA A ROOTED-TREE GRAPH)

อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.ยงยุทธ เสริมสุริยวัฒน์, 126 หน้า.

ในการผลิตชิ้นงานด้วยกรรมวิธีการตัดโลหะ (machining) ถ้าระบุอ้างอิงในการผลิต (manufacturing datums) หรือในการตรวจสอบขนาดชิ้นงาน (inspection datums) มีความแตกต่างไปจากระบบอ้างอิงสำหรับการบอกขนาดแล้ว ปัญหาที่จะเกิดขึ้นตามมาก็คือ ความคลาดเคลื่อนสะสม (tolerance stacks) ซึ่งส่งผลให้ผู้วางแผนกระบวนการผลิตจำเป็นต้องเปลี่ยนแปลงค่าความคลาดเคลื่อนของมิติในการผลิตหรือในการวัดขนาด เพื่อคงไว้ซึ่งความต้องการใช้งานของชิ้นงาน (functional requirement) ที่ผู้ออกแบบได้กำหนดไว้ ปัญหาเช่นนี้จัดเป็นปัญหาของการแบ่งหรือกระจายค่าความคลาดเคลื่อน (tolerance allocation problems) ซึ่งมักจะเกิดขึ้นอยู่เสมอในการวางแผนการผลิต การผลิต และการตรวจสอบขนาด และเป็นปัญหาที่ต้องอาศัยประสบการณ์และความชำนาญของผู้วางแผนกระบวนการผลิต ผู้ออกแบบเกจตรวจสอบ (gauge designer) หรือวิศวกรควบคุมคุณภาพ

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาโปรแกรมคอมพิวเตอร์เพื่อใช้สำหรับการแบ่งค่าความคลาดเคลื่อนให้เหมาะสมกับมิติชุดใหม่เมื่อจุดอ้างอิงของการผลิตและการตรวจสอบแตกต่างไปจากจุดอ้างอิงของการบอกขนาด โดยประยุกต์เทคนิคของกราฟต้นไม้ (rooted-tree graph) เพื่อสร้างความสัมพันธ์ระหว่างมิติและเทคนิคการหาค่าที่เหมาะสมที่สุดด้วยโปรแกรมเชิงเส้นตรง (linear programming) เพื่อให้ได้ผลลัพธ์ความคลาดเคลื่อนสำหรับมิติชุดใหม่ทั้งหมดมีค่ามากที่สุดภายใต้เงื่อนไขความสัมพันธ์ของมิติใช้งานและขณะเดียวกันต้องอยู่ในเกณฑ์ที่สามารถผลิตได้ในทางปฏิบัติด้วยกระบวนการผลิตโดยทั่ว ๆ ไป (processing conditions)

โปรแกรมที่พัฒนาขึ้นสามารถหาผลลัพธ์ที่เหมาะสมที่สุดให้กับปัญหาการแบ่งค่าความคลาดเคลื่อนได้ครั้งละหลาย ๆ ชุดคำตอบ โดยไม่ต้องสร้างสมการเชิงเส้นตรงด้วยมือและป้อนค่าให้กับโปรแกรมใหม่ทุกครั้งเมื่อมีการปรับเปลี่ยนเกณฑ์ความคลาดเคลื่อน ช่วยทำให้ลดขั้นตอนและเวลาในการทำงานลงได้มาก โดยที่ผู้ใช้โปรแกรมไม่จำเป็นต้องมีความรู้และความชำนาญในทางปฏิบัติมากนัก ก็สามารถกระจายความคลาดเคลื่อนให้กับมิติในการผลิตหรือในการวัดขนาดใหม่ เพื่อคงไว้ซึ่งความต้องการใช้งานของชิ้นงานที่ผู้ออกแบบกำหนดไว้

ORAWAN NONGNUCH : OPTIMAL TOLERANCE ALLOCATION
VIA A ROOTED-TREE GRAPH. THESIS ADVISOR : ASST. PROF.
YONGYOOH SREMSUTIANUWAT, Ph.D., 126 PP.

TOLERANCE ALLOCATION/OPTIMIZATION/LINEAR PROGRAMMING

The problems of tolerance allocation are normally encountered in various phases of engineering production ranging from product design, process planning, production, to product quality control. To solve such problems requires not only high practical experience but also time and efforts. A proper tolerance analysis computer package is therefore in need of those personnel involving in aforementioned activities.

This thesis reports the development of a computerized tolerance allocation system, namely, Tolerance Optimization Program (TOLOP). The program is capable of optimally allocating blueprint tolerances to a new set of dimensions corresponding to the datum for manufacturing. By making use of a rooted-tree graph, all tolerance relationship among functional dimensions on a machined part, before and after the baseline for dimensioning has been modified, can be created. These relationships and other practical machining limitations are taken as the constraints for allocating of tolerances with a linear programming algorithm.

The Visual Basic language has been used for developing TOLOP for its abundant tools for a program development. The inputs required by the package include: the number of dimensions involved in a problem, the basic sizes and the tolerances together with the end-faces of all old and new sets of dimensions. TOLOP

also requires to know a range of the International Tolerance (IT) numbers for the machining operation making each dimension. This piece of information enables the program to give a solution within a practical feasibility. Furthermore, it requires to be entered the number of cycles to execute the routines to obtain the best solution, or a set of possible solutions. TOLOP then constructs a rooted-tree for those dimensions resulting from the change of datum for dimensioning. This graph records all functional dimensions relating to the blueprint dimensions.

Starting from a set of the minimum IT numbers for the new set of dimensions, TOLOP creates an objective function and a set of linear programming constraints, and solves for an optimum solution. It will repeat the routine with different IT numbers the number of times specified by the number of executions entered. The user can interact with the application to modify the method of applying the IT numbers and the number of program runs.

TOLOP has been validated by comparing the results from its executions with those obtained from two commercial packages Excel and Lindo, and they all agree. With 10 linear dimensions and 2 tolerance grades attaching to each dimension, TOLOP can give an optimum solution within 10 seconds. It can handle more dimensions and tolerance grades at an expense of computing time.

School of Industrial Engineering

Academic Year 2011

Student's signature _____

Advisor's signature _____

การกำหนดค่าความคลาดเคลื่อนที่เหมาะสมโดยใช้ Rooted-Tree Graph

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(ผศ. ดร.พงษ์ชัย จิตตะมัย)

ประธานกรรมการ

(ผศ. ดร.ขงยุทธ เสริมสุขธินุวัฒน์)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(อ. ดร.วีระชัย มโนพิเชฐวัฒนา)

กรรมการ

(รศ. ดร.เกตุศักดิ์ เกิดประสพ)

กรรมการ

(ศ. ดร.ชูกิจ ลิมปิจำนงค์)

รองอธิการบดีฝ่ายวิชาการ

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคลและกลุ่มบุคคลที่ได้ให้คำปรึกษา แนะนำ ช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการและด้านการดำเนินงานวิจัย อาทิ เช่น

ผู้ช่วยศาสตราจารย์ ดร.ยงยุทธ เสริมสุขอินุวัฒน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ให้การสนับสนุน ช่วยเหลือ ให้คำปรึกษาที่มีคุณค่าด้วยดีเสมอมา รวมทั้งสละเวลาแก้ไขตรวจทานเนื้อหาวิทยานิพนธ์จนเสร็จสมบูรณ์

ผู้ช่วยศาสตราจารย์ ดร.พรศิริ จงกล ที่ได้ให้โอกาสทางการศึกษา และให้กำลังใจแก่ผู้วิจัยมาโดยตลอด

คุณปราณี กลิ่นใหม่ เจ้าหน้าที่บริหารงานทั่วไป ประจำสาขาวิชาวิศวกรรมอุตสาหกรรม ที่แนะนำขั้นตอนการดำเนินการให้เป็นไปด้วยความเรียบร้อย และให้กำลังใจแก่ผู้วิจัยมาโดยตลอด

คุณปฏิกรณ์ แสนสิ่ง คุณพรพิพัฒน์ มีดีจันทร์ขันธ์ และคุณพุดพิงษ์ ของทอง ที่ให้ความช่วยเหลือและให้คำแนะนำในการดำเนินงานวิจัยด้วยดีมาโดยตลอด

ขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่านที่ให้คำชี้แนะเพื่อให้วิทยานิพนธ์เล่มนี้สมบูรณ์มากยิ่งขึ้น

อรรณณ นงนุช

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญตาราง.....	ช
สารบัญรูป.....	ฉ
บทที่	
1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 การเขียนแผนภาพวงรอบ	3
1.2.1 วิธีการของ Drake.....	4
1.2.2 วิธีการของ Ligget.....	6
1.2.3 วิธีการของ Gladman.....	8
1.3 ปัญหาของวิธีการเขียนแผนภาพวงรอบ.....	9
1.4 วิธีการทั่วไปสำหรับแบ่งค่าความคลาดเคลื่อน (manual allocation of tolerances)....	10
1.5 กราฟต้นไม้	12
1.6 วัตถุประสงค์	14
1.7 ขอบเขตและสมมติฐานของงานวิจัย.....	14
1.8 สรุป.....	14
2 ปรัชญาวรรณกรรมและงานวิจัยที่เกี่ยวข้อง.....	16
2.1 วิธีการวิเคราะห์ความคลาดเคลื่อน.....	16
2.1.1 การวิเคราะห์ความคลาดเคลื่อนในชิ้นงานประกอบ.....	16
2.1.2 การวิเคราะห์ความคลาดเคลื่อนในชิ้นงานชิ้นเดียว	17

สารบัญ (ต่อ)

หน้า

2.2	การใช้คอมพิวเตอร์ช่วยในการวิเคราะห์ความคลาดเคลื่อน	18
2.3	การประยุกต์ใช้เทคนิคกราฟในการวิเคราะห์ความคลาดเคลื่อน	18
2.4	การใช้เทคนิคกราฟต้นไม้ในการทำแผนภูมิความคลาดเคลื่อน	19
2.5	วิธีการสร้างกราฟต้นไม้แสดงความสัมพันธ์ของความคลาดเคลื่อน	22
2.6	การโปรแกรมเชิงเส้นตรงกับการกระจายความคลาดเคลื่อน	24
2.6.1	สมการเป้าหมาย (objective function)	24
2.6.2	สมการข้อจำกัด (constraints)	24
2.7	ข้อจำกัดของโปรแกรมสำเร็จรูปทั่วไป	26
2.8	สรุป	27
3	โครงสร้างโปรแกรม	28
3.1	องค์ประกอบของโปรแกรม TOLOP	28
3.1.1	โมดูลหลัก (main module)	29
3.1.2	โมดูลป้อนข้อมูล (Input module)	32
3.1.3	โมดูลกราฟต้นไม้ (tree graph module)	34
3.1.4	โมดูลฐานข้อมูลค่าความคลาดเคลื่อนมาตรฐาน (International tolerance grades data base module)	35
3.1.5	โมดูลการคำนวณหาค่าที่เหมาะสม (optimization module)	35
3.1.6	โมดูลปรับปรุงความคลาดเคลื่อน (modification module)	36
3.2	สรุป	36
4	โปรแกรม TOLOP	37
4.1	โมดูลกราฟต้นไม้ (tree graph module)	37
4.2	การหามิติใหม่ที่มีค่าความคลาดเคลื่อนสัมพันธ์กับมิติเก่า	40
4.3	โมดูลฐานข้อมูลค่าความคลาดเคลื่อนมาตรฐาน (International tolerance grades data base module)	43
4.3.1	การสร้างตารางฐานข้อมูลค่าความคลาดเคลื่อนมาตรฐาน	43
4.3.2	สร้างรูปแบบของเกรด IT No.	43

สารบัญ (ต่อ)

หน้า

4.4	โมดูลการคำนวณหาค่าที่เหมาะสม (optimization module).....	45
4.4.1	โปรแกรมย่อย Sub create_initial_equation	46
4.4.2	โปรแกรมย่อย Sub create_msgrid	47
4.4.3	โปรแกรมย่อย Sub find_all_solution_step	49
4.5	โมดูลปรับความคลาดเคลื่อน (modification module).....	51
4.6	สรุป.....	53
5	สรุปผลการวิจัย	54
5.1	สรุป.....	54
5.2	ข้อจำกัดของโปรแกรม TOLOP	55
5.3	ข้อเสนอแนะในการพัฒนาต่อไป	55
	รายการอ้างอิง	56
	ภาคผนวก	
	ภาคผนวก ก. ตารางเกณฑ์ความคลาดเคลื่อนมาตรฐาน (ISO 286-2)	58
	ภาคผนวก ข. คู่มือการใช้งานโปรแกรม TOLOP	60
	ภาคผนวก ค. รายการ โปรแกรม (Program Listings)	91
	ภาคผนวก ง. บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่	119
	ประวัติผู้เขียน	126

สารบัญตาราง

ตารางที่	หน้า
2.1	แสดงความสัมพันธ์ของผิวที่ถูกตัดและผิวที่เป็นจุดอ้างอิงในการวัดขนาด20
2.2	ความสัมพันธ์ของความคลาดเคลื่อนของมิติเก่าและมิติใหม่24
4.1	แสดงข้อมูลของตัวแปรสำหรับการหาเส้นทางความสัมพันธ์ระหว่างมิติเก่ากับมิติใหม่ ...41
4.2	ความคลาดเคลื่อนที่เกิดจากกระบวนการผลิตต่าง ๆ44
4.3	แสดงข้อมูลของตัวแปรที่ใช้ในโปรแกรมย่อย Sub find_all_solution_step.....49

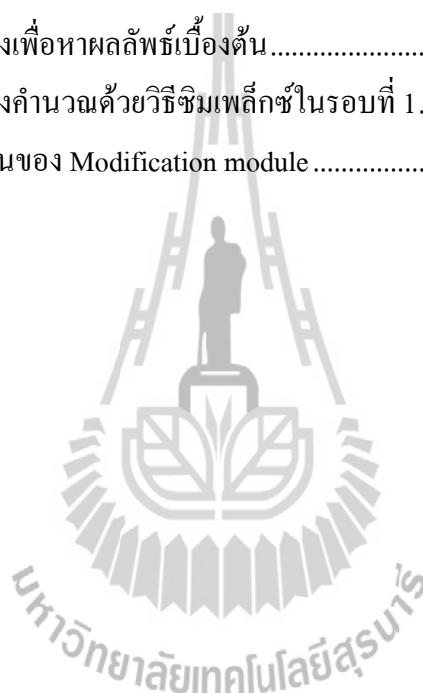


สารบัญรูป

รูปที่	หน้า
1.1	แบบชิ้นงานที่ใช้ผลิตจริงในอุตสาหกรรม..... 2
1.2	ชิ้นส่วนประกอบของมอเตอร์(นิว)..... 4
1.3	แผนภาพวงรอบสำหรับหาขนาดมูลฐานและความคลาดเคลื่อนของมิติ GAP..... 5
1.4	ชิ้นส่วนประกอบของดิสเบรก 6
1.5	แผนภาพวงรอบสำหรับหาขนาดมูลฐานและความคลาดเคลื่อนของมิติ G..... 7
1.6	ชิ้นงานประกอบ 8
1.7	ชิ้นงานกลึง 10
1.8	แบบของชิ้นงานตัด-กลึง 12
1.9	กราฟต้นไม้ 13
2.1	ชิ้นงานสำเร็จจากการตัดโลหะมิติต่าง ๆ (Whybrew <i>et al.</i> , 1990) 19
2.2	แผนภูมิความคลาดเคลื่อน (Whybrew <i>et al.</i> , 1990)..... 19
2.3	กราฟต้นไม้ (Whybrew <i>et al.</i> , 1990) 21
2.4	ชิ้นงานตัวอย่างสำหรับการคำนวณมิติความยาว 4 มิติ..... 22
2.5	กราฟต้นไม้สำหรับมิติใหม่..... 23
3.1	องค์ประกอบหลักของโปรแกรม TOLOP..... 28
3.2	ส่วนประกอบของหน้าจอเริ่มต้นหรือโมดูลหลัก..... 30
3.3	ส่วนประกอบของแถบเมนู..... 30
3.4	ขั้นตอนการทำงานภายในโมดูลหลัก..... 31
3.5	แสดงหน้าจอป้อนข้อมูล..... 33
3.6	แสดงหน้าต่าง Open 33
3.7	ขั้นตอนการทำงานของโมดูลป้อนข้อมูล..... 34
3.8	คอนโทรลทรีวิว 35
4.1	ขั้นตอนการสร้างกราฟต้นไม้ 38
4.2	ผลการแสดงกราฟต้นไม้ที่ได้จากคอนโทรลทรีวิว..... 40
4.3	ขั้นตอนการหามิติใหม่ที่มีค่าความคลาดเคลื่อนสัมพันธ์กับมิติเก่า 42

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.4	ตารางฐานข้อมูลค่าความคลาดเคลื่อนมาตรฐาน43
4.5	การกำหนดกระบวนการผลิตให้กับโปรแกรม44
4.6	ลำดับการทำงานของโปรแกรมย่อยในโมดูลการคำนวณหาค่าที่เหมาะสม45
4.7	แสดงการสร้างสมการในรูปแบบของการโปรแกรมเชิงเส้นตรง.....46
4.8	รูปแบบของตารางคำนวณ48
4.9	รูปแบบของตารางเพื่อหาผลลัพธ์เบื้องต้น49
4.10	รูปแบบของตารางคำนวณด้วยวิธีซิมเพล็กซ์ในรอบที่ 1.....51
4.11	แผนภูมิการทำงานของ Modification module52



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การผลิตชิ้นงานด้วยกรรมวิธีการตัดโลหะ (machining) ถ้าหากระนาบอ้างอิงในการผลิต (manufacturing datums) หรือในการตรวจสอบขนาดชิ้นงาน (inspection datums) แตกต่างไปจากรนาบอ้างอิงในการบอกขนาดแล้ว ผลที่จะเกิดขึ้นก็คือปัญหาความคลาดเคลื่อนสะสม (tolerance stacks) ซึ่งผู้วางแผนกระบวนการผลิตจำเป็นต้องกระจายค่าความคลาดเคลื่อนให้กับมิติ (dimensions) ในการผลิตหรือในการวัดขนาดใหม่เพื่อคงไว้ซึ่งความต้องการใช้งานของชิ้นงาน (functional requirement) ที่ผู้ออกแบบกำหนด เราเรียกปัญหานี้ว่า การกระจายหรือแบ่งค่าความคลาดเคลื่อน (tolerance allocation) และถึงแม้ว่าในปัจจุบันเครื่องมือกลที่ใช้ในการผลิตมีความเที่ยงตรงสูงก็ตาม การกระจายค่าความคลาดเคลื่อนยังคงเป็นเรื่องจำเป็นทั้งในด้านการออกแบบ การวางแผนการผลิต การผลิต การตรวจสอบ และการประกอบชิ้นงาน ทั้งนี้เพราะงานตัดโลหะชนิดหนึ่งสามารถทำได้บนเครื่องมือกลที่แตกต่างกัน และการตรวจสอบมิติเดียวกันก็สามารถทำได้หลายวิธี นอกจากนี้ยังมีชิ้นงานจำนวนมากที่ต้องเปลี่ยนตำแหน่งการจับชิ้นงานหลายครั้งกว่าการผลิตชิ้นงานจะเสร็จสมบูรณ์ เพราะฉะนั้นการวิเคราะห์ความคลาดเคลื่อน (tolerance analysis) ชนิดนี้จึงยังคงมีความสำคัญต่ออุตสาหกรรมการตัดโลหะ โดยเฉพาะอย่างยิ่งชิ้นงานที่มีหลายมิติ และมีขั้นตอนในการขึ้นรูปที่ซับซ้อน ดังตัวอย่างชิ้นงานในรูปที่ 1.1 ซึ่งเป็นแบบของชิ้นงานที่ใช้จริงในอุตสาหกรรม จะเห็นได้ว่าถ้าหากระนาบอ้างอิงของการผลิตเปลี่ยนแปลง การกระจายค่าความคลาดเคลื่อนให้กับมิติของชิ้นงานเพื่อการผลิตจะต้องใช้เวลานานในการพิจารณาเนื่องจากชิ้นงานมีรูปร่างที่ซับซ้อนและมีงานตัดโลหะหลายชนิดที่เกี่ยวข้อง

การกระจายหรือแบ่งค่าความคลาดเคลื่อนให้กับมิติในการผลิตใหม่นั้นมีหลักเกณฑ์ว่า ค่าความคลาดเคลื่อนมากผลิตได้ง่ายแต่ไม่รับประกันว่าชิ้นงานจะใช้ได้หรือไม่ ค่าความคลาดเคลื่อนน้อยผลิตได้ยากและต้นทุนสูงแต่ชิ้นงานที่ผลิตได้สามารถทำหน้าที่ได้เสมอ ดังนั้นการกระจายหรือแบ่งค่าความคลาดเคลื่อนจึงต้องพยายามให้มิติที่เกี่ยวข้องกับความต้องการใช้งานมีความคลาดเคลื่อนมากที่สุด ขณะเดียวกันก็ต้องไม่ทำให้หน้าที่ในการใช้งานมีความบกพร่อง

หน้าที่ใช้งาน (functional requirement) ของชิ้นงานแต่ละชิ้นเป็นสถานะที่ชิ้นงานที่พิจารณาสัมพันธ์กับส่วนประกอบอื่น ๆ ของชุดงานประกอบ (assembly) สถานะเช่นนี้จะถูกถ่ายทอดให้กับชิ้นงานประกอบแต่ละชิ้นด้วยการกำหนดค่าความคลาดเคลื่อนให้กับมิติของชิ้นงานที่เกี่ยวข้อง ซึ่ง

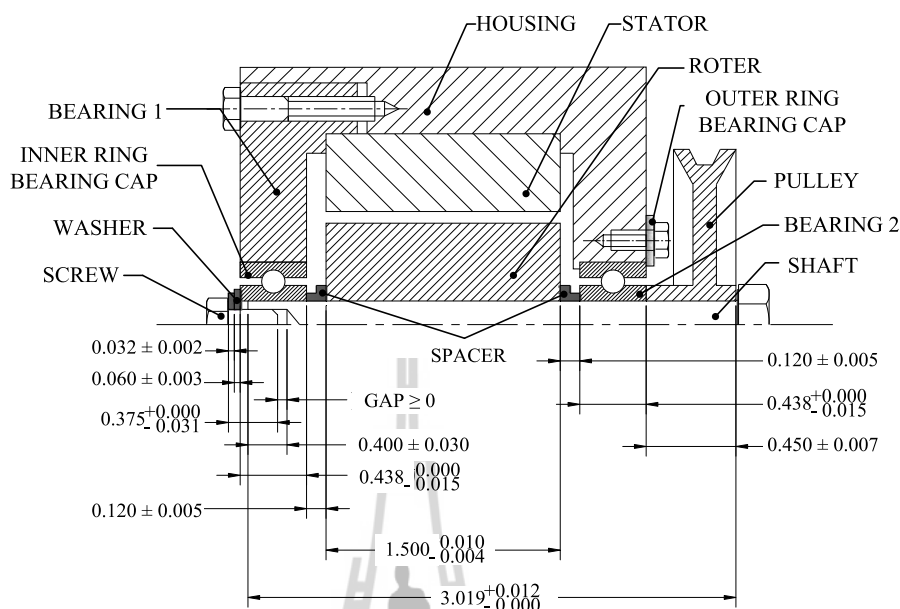
เราเรียกมิติที่สัมพันธ์กับความต้องการใช้งานของชุดงานประกอบว่า มิติใช้งาน (function dimension) ในการผลิตชิ้นงานแต่ละชิ้น มิติใช้งานของชิ้นงานแต่ละชิ้นก็คือ ความต้องการใช้งานหรือเป้าหมายในการผลิตของชิ้นงานนั้น ๆ เพราะถ้าหากไม่สามารถบรรลุเป้าหมายนี้ได้ก็จะมีผลเสียต่อหน้าที่ใช้งานของชุดงานประกอบโดยรวม

ในปัจจุบันมีวิธีการสร้างความสัมพันธ์ของความคลาดเคลื่อนในมิติใช้งานของชิ้นส่วนประกอบกับความต้องการใช้งานของชุดงานประกอบอยู่หลายวิธี วิธีการเหล่านี้เป็นพื้นฐานของการกระจายความคลาดเคลื่อนในชิ้นงานชิ้นใดชิ้นหนึ่งด้วย

1.2 การเขียนแผนภาพวงรอบ

แผนภาพวงรอบ (loop diagram) คือ แผนภาพที่แสดงความสัมพันธ์ของความต้องการใช้งานของชุดงานประกอบกับมิติใช้งานในชิ้นส่วนประกอบแต่ละชิ้น การเขียนแผนภาพชนิดนี้เป็นเทคนิคของการออกแบบชุดงานประกอบซึ่งเป็นที่นิยมใช้มาเป็นเวลานานในอุตสาหกรรม และในงานวิจัยทางด้านความคลาดเคลื่อน เพราะพบในบทความและหนังสือทางวิชาการมาไม่น้อยกว่า 50 ปี และ C.A. Gladman ได้บันทึกไว้อย่างเป็นระบบในหนังสือเมื่อปี 1972 นอกจาก Gladman แล้ว P.Drake และ J.Ligget ก็มีการใช้แผนภาพในลักษณะเดียวกันนี้ด้วย ถึงแม้ว่าแผนภาพจะมีลักษณะเดียวกันแต่รายละเอียดของการกำหนดมิติที่เกี่ยวข้องหรือเรียกว่า “มิติใช้งาน” (functional dimensions) มีความแตกต่างกันโดยเฉพาะอย่างยิ่ง Gladman ไม่ได้กล่าวถึงขั้นตอนหรือวิธีการเหล่านี้ไว้เลย จึงน่าจะเข้าใจได้ว่าการกำหนดมิติใช้งานน่าที่จะต้องอาศัยประสบการณ์ในการทำงานทางปฏิบัติด้วย ซึ่งความเข้าใจเช่นนี้ได้รับการยืนยันด้วยความแตกต่างของวิธีการของ Drake และ Ligget ในที่นี้จะกล่าวถึงวิธีการทั้ง 2 นี้ก่อนแล้วจึงขยายความวิธีการของ Gladman ในภายหลัง

1.2.1 วิธีการของ Drake



รูปที่ 1.2 ชิ้นส่วนประกอบของมอเตอร์ (นิ้ว)

พิจารณารูปที่ 1.2 กำหนดให้ GAP คือ ความต้องการใช้งานที่ต้องมีค่ามากกว่าหรือเท่ากับ ศูนย์ โดย Drake ได้สรุปขั้นตอนการเขียนแผนภาพวงรอบไว้ดังนี้

- 1) กำหนดให้ผิวด้านซ้ายของ GAP เป็นจุดเริ่มต้นและกำหนดให้ผิวด้านขวาของ GAP เป็นจุดสิ้นสุด
- 2) ใช้เวกเตอร์สร้างแผนภาพวงรอบแบบปิด จากจุดเริ่มต้นจนถึงจุดสิ้นสุด โดยไม่รวม GAP (รูปที่ 1.3)
- 3) ใช้หัวลูกศรแสดงทิศทางของแต่ละเวกเตอร์ โดยระบุเวกเตอร์ด้วยเครื่องหมายบวก (+) เมื่อลูกศรชี้จากซ้ายไปขวา หรือ ระบุเวกเตอร์ด้วยเครื่องหมายลบ (-) เมื่อลูกศรชี้จากขวาไปซ้าย
- 4) ในรูปเดียวกันนี้กำหนดชื่อตัวแปรให้แต่ละมิติและเปลี่ยนวิธีการบอกขนาดความคลาดเคลื่อนให้เป็นแบบเบี่ยงเบนจากขนาดมูลฐานเท่า ๆ กันทั้งสองข้าง (bilateral tolerances) โดยมีตัวอักษรแทนมิติต่าง ๆ ดังนี้ $X \pm x$ คือ ขอบเขตของขนาด GAP ที่ต้องการทราบค่า; A คือ ขนาดของปลายสกรูถึงผิวด้านของหัวสลักเกลียว หรือ คือ ความยาวของสลักเกลียว; B คือ ความหนาของแหวนรอง (washer);

C คือ ความหนาของแหวนกดแบร์ริง (inner ring bearing cap); D คือ ความหนาของแบร์ริงตัวแรก (bearing 1); E คือ ความหนาของ spacer; F คือ ความยาวของ rotor; G คือ ความหนาของ spacer; H คือ ความหนาของแบร์ริงตัวที่ 2 (bearing 2); I คือ ความยาวของ ป्लอกพูลเลย์ (pulley hub); J คือ ความยาวของเพลา (shaft); K คือ ระยะจากผิวปลายเพลาถึงก้นรู (hole bottom)

$$GAP = X \pm x$$

$$-A = (0.3595) \pm 0.0155$$

$$+B = (0.0320) \pm 0.0020$$

$$+C = (0.0600) \pm 0.0030$$

$$+D = (0.4305) \pm 0.0075$$

$$+E = (0.1200) \pm 0.0050$$

$$+F = (1.5030) \pm 0.0070$$

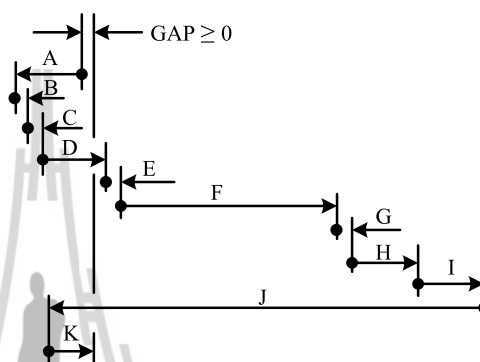
$$+G = (0.1200) \pm 0.0050$$

$$+H = (0.4500) \pm 0.0070$$

$$-J = (0.4305) \pm 0.0075$$

$$+I = (3.0250) \pm 0.0060$$

$$+K = (0.3000) \pm 0.0300$$



รูปที่ 1.3 แผนภาพวงรอบสำหรับหาขนาดมูลฐานและความคลาดเคลื่อนของมิติ GAP

5) สร้างสมการวงรอบขนาดมูลฐานและสมการวงรอบสำหรับความคลาดเคลื่อนได้ดังนี้

สมการวงรอบขนาดมูลฐาน :

$$X = -A+B+C+D+E+F+G+H-I-J+K \quad (1.1)$$

$$= -0.3595 + 0.0320 + 0.0600 + 0.4305 + 0.1200 + 1.5030 + 0.1200 + 0.4305 + 0.4500 \\ - 3.0250 + 0.4000$$

$$X = 0.1615$$

สมการวงรอบสำหรับความคลาดเคลื่อน :

$$\pm x = \pm (a+b+c+d+e+f+g+h+i+j+k) \quad (1.2)$$

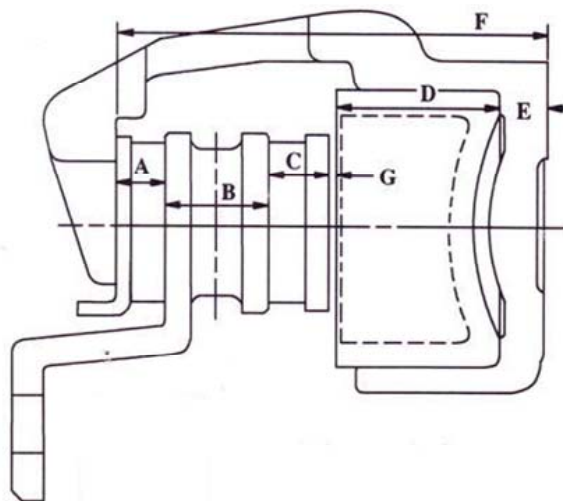
$$= 0.0155 + 0.0020 + 0.0030 + 0.0075 + 0.0050 + 0.0070 + 0.0050 + 0.0075 \\ + 0.0070 + 0.0060 + 0.0300$$

$$\pm x = \pm 0.0955$$

$$\therefore \text{GAP หรือ } X \pm x = 0.1615 \pm 0.0955$$

เป็นที่น่าสังเกตว่า ขั้นตอนที่ Drake เสนอจะเน้นเรื่องของทิศทางการพิจารณาจาก functional requirement และไม่มีกฎเกณฑ์อย่างอื่นกำกับ ซึ่งมีโอกาสที่จะเกิดมิติของชิ้นงานเดียวกันซ้ำซ้อนในสมการวงรอบได้

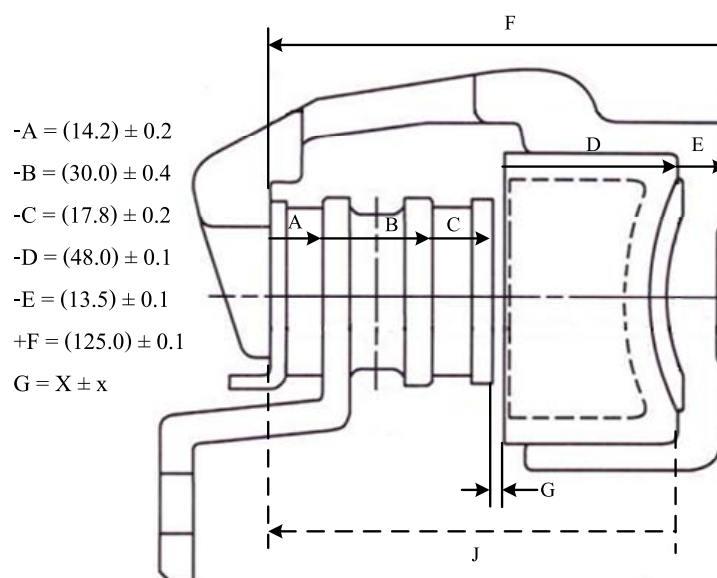
1.2.2 วิธีการของ Ligget



รูปที่ 1.4 ชิ้นส่วนประกอบของคิสเบรก

พิจารณารูปที่ 1.4 กำหนดให้มิติ A ถึง F คือมิติในชิ้นส่วนที่เป็นองค์ประกอบของคิสเบรก และ G คือ ความต้องการใช้งานที่ต้องการทราบค่า โดย Ligget มีขั้นตอนการหาสมการวงรอบดังนี้

- 1) กำหนดให้ผิวด้านขวาของ G เป็นจุดเริ่มต้นและให้ผิวด้านซ้ายของ G เป็นจุดสิ้นสุด
- 2) ใช้เวกเตอร์สร้างแผนภาพวงรอบแบบปิด จากจุดเริ่มต้นจนถึงจุดสิ้นสุด โดยไม่รวมช่องว่าง G
- 3) ใช้หัวลูกศรแสดงทิศทางของแต่ละเวกเตอร์ โดยระบุเวกเตอร์ด้วยเครื่องหมายบวก (+) เมื่อลูกศรชี้จากขวาไปซ้าย หรือ ระบุเวกเตอร์ด้วยเครื่องหมายลบ (-) เมื่อลูกศรชี้จากซ้ายไปขวา
- 4) กำหนดข้อผิดพลาดแปรให้แต่ละมิติและเปลี่ยนขนาดมูลฐานและความคลาดเคลื่อนให้เท่ากันทั้ง 2 ด้าน (bilateral tolerances) ดังรูปที่ 1.5



รูปที่ 1.5 แผนภาพวงรอบสำหรับหาขนาดมูลฐานและความคลาดเคลื่อนของมิติ G

- 5) คำนวณหาขนาดมูลฐานโดยใช้วิธีทางพีชคณิตคำนวณตามเครื่องหมายที่แสดงทิศทางของแต่ละมิติได้ดังนี้

สมการวงรอบขนาดมูลฐาน :

$$X = -A - B - C - D - E + F \quad (1.3)$$

$$= -14.2 - 30 - 17.8 - 48 - 13.5 + 125$$

$$X = 1.5$$

- 6) คำนวณค่าความคลาดเคลื่อนโดยการนำความคลาดเคลื่อนทั้งหมดมารวมกันจะได้

สมการวงรอบสำหรับความคลาดเคลื่อนดังนี้

สมการวงรอบสำหรับความคลาดเคลื่อน :

$$\pm x = \pm(a + b + c + d + e + f) \quad (1.4)$$

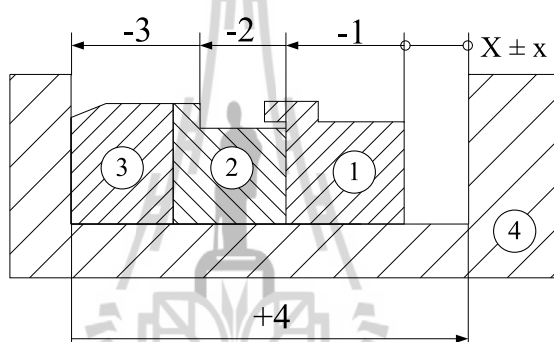
$$= \pm(0.2 + 0.4 + 0.2 + 0.1 + 0.1 + 0.1)$$

$$\pm x = \pm 1.1$$

$$\therefore G \text{ หรือ } X \pm x = 1.5 \pm 1.1$$

1.2.3 วิธีการของ Gladman

อันที่จริง Gladman ใช้สมการวงรอบในการวิเคราะห์ความสัมพันธ์ระหว่างมิติมา โดยตลอดโดยแยกเป็นสมการสำหรับขนาดมูลฐานและสมการสำหรับความคลาดเคลื่อน ดังที่ได้กล่าวมาแล้วในตอนต้น และยังไม่มีการตั้งกฎเกณฑ์การพิจารณากำหนดมิติใช้งาน (function dimensions) ที่เกี่ยวข้องกับความต้องการใช้งานแต่อย่างใด ที่เป็นเช่นนี้อาจเป็นเพราะในงานด้านวิศวกรรมเครื่องมือ (tool engineering) ความรู้ในเรื่องนี้เป็นความรู้พื้นฐานของช่างหรือผู้มีประสบการณ์ในทางปฏิบัติก็ได้ จนกระทั่ง Sermutsi-anuwat (2009) ได้สรุปแนวทางการประยุกต์ใช้ของ Gladman และได้อธิบายขั้นตอนการเขียนแผนภาพวงรอบสำหรับงานประกอบไว้อย่างเป็นระบบและชัดเจน เป็นขั้นตอนดังนี้



รูปที่ 1.6 ชิ้นงานประกอบ

จากรูปที่ 1.6 แสดงชิ้นงานประกอบที่มีชิ้นส่วนประกอบทั้งหมด 4 ชิ้น (หมายเลข 1-4) ความต้องการใช้งานในการประกอบนี้ คือ ระยะห่างระหว่างชิ้นส่วนที่ 4 และชิ้นส่วนที่ 1 ซึ่งเขียนแทนด้วยมิติ $X \pm x$ เพื่อที่จะหามิติใช้งานที่ส่งผลทำให้ขนาดของช่องว่างเปลี่ยนแปลงโดยใช้แผนภาพวงรอบ สามารถทำได้ง่ายโดยมีขั้นตอนการวิเคราะห์ดังต่อไปนี้

- 1) เขียนลูกศรจากผิวแรกของมิติความต้องการใช้งานแล้วเขียนหัวลูกศรไปยังอีกผิวหนึ่งของชิ้นส่วนเดียวกัน ซึ่งผิว ๆ นี้จะต้องมีความสัมผัสกับผิวชิ้นส่วนอื่น ลูกศรหมายเลข 1 ในรูปที่ 1.6 คือผลลัพธ์ของขั้นตอนนี้ โดยเริ่มจากผิวของชิ้นงานหมายเลข 1 ที่ติดกับช่องว่างแล้วเขียนลูกศรไปยังผิวของชิ้นส่วนหมายเลข 2 ในตำแหน่งที่ตรงข้ามกัน ลูกศรนี้แสดงมิติใช้งานของชิ้นส่วนหมายเลข 1

ในการพิจารณาว่ามิติใดเป็นมิติใช้งานให้สังเกตดูว่าความต้องการใช้งานเปลี่ยนแปลงหรือไม่ เมื่อมิติที่พิจารณาเปลี่ยนแปลงในขณะที่มิติอื่น ๆ คงที่

- 2) เขียนลูกศรหมายเลข 2 จากผิวที่สัมผัสกับหัวลูกศรหมายเลข 1 ไปยังอีกผิวหนึ่งของชิ้นส่วนหมายเลข 2 ที่ติดกับผิวของชิ้นส่วนหมายเลข 3 ลูกศรนี้แสดงมิติใช้งานของชิ้นส่วนหมายเลข 2
- 3) ทำซ้ำในข้อที่ 2) จนกระทั่งลูกศรย้อนกลับไปหาอีกขอบหนึ่งของมิติความต้องการใช้งาน จากนั้นให้หยุดกระบวนการทำงานก็จะได้แผนภาพวงรอบที่เสร็จสมบูรณ์ มิติใช้งานทั้งหมดที่มีผลต่อขนาดของช่องว่างจะรวมอยู่ใน

แผนภาพ

ซึ่งขั้นตอนที่กล่าวมานี้จะคล้ายกับเทคนิคทั้งสองที่กล่าวมาข้างต้น แต่ในบทความนี้ได้ให้หลักเกณฑ์ในการพิจารณามิติใช้งานที่เกี่ยวข้องกับความต้องการใช้งานไว้ว่า ชิ้นส่วนประกอบหนึ่งชิ้นจะต้องที่มีมิติใช้งานเพียงหนึ่งมิติเท่านั้น เนื่องจากชิ้นส่วนในงานประกอบหนึ่งชิ้นจะมีเพียงสองด้านที่สัมผัสกับชิ้นส่วนอื่น ๆ ซึ่งด้านทั้งสองเป็นมิติเชิงเส้นตรงมิติเดียว นอกจากนั้นยังได้เสนอแนะกฎเกณฑ์อื่นเป็นวิธีการสร้างความสัมพันธ์ของตัวแปรในสมการวงรอบไว้อีกด้วย

1.3 ปัญหาของวิธีการเขียนแผนภาพวงรอบ

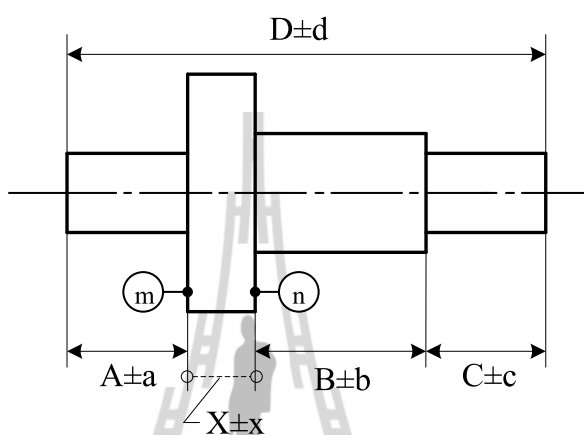
แม้ว่าโดยทั่วไปแผนภาพวงรอบจะเป็นที่ยอมรับกันในเรื่องการหาความสัมพันธ์ระหว่างมิติที่เกี่ยวข้องกับความต้องการใช้งาน แต่แต่ละวิธีการหรือแต่ละเทคนิคจะมีรายละเอียดที่สะท้อนถึงประสบการณ์ของผู้ใช้ที่แตกต่างกัน ตัวอย่างเช่น Drake และ Ligget ไม่ได้เน้นถึงจำนวนมิติของชิ้นงานแต่ละชิ้นในสมการวงรอบ แต่ Gladman เน้น

ถ้าพิจารณาตัวอย่างชิ้นงานประกอบของ Ligget โดยใช้หลักเกณฑ์ที่กล่าวไว้ในบทความของ Sermsuti-anuwat (2009) จะพบว่ามิติ J (แสดงเป็นเส้นประ) ของโครงคิสเบรกเป็นมิติใช้งานที่เกี่ยวข้องกับ G ส่วน E และ F เป็นผลของการเปลี่ยนแปลงเดมของการบอกขนาดในโครงคิสเบรกเท่านั้น แต่ด้วยประสบการณ์ของ Ligget มิติ E และ F สามารถนำมาใช้แทน J ได้ เพราะความคลาดเคลื่อนของ E และ F เป็นผลลัพธ์จากการกระจายความคลาดเคลื่อนของ J

แผนภาพวงรอบสามารถแสดงความสัมพันธ์ของมิติระหว่างชิ้นงานได้ดี แต่กฎเกณฑ์ในการเขียนแผนภาพวงรอบเพื่อให้ได้มาซึ่งสมการวงรอบนั้น ยังไม่สามารถครอบคลุมไปถึงกรณีของมิติหลายมิติในชิ้นงานเดียวกันได้

1.4 วิธีการทั่วไปสำหรับแบ่งค่าความคลาดเคลื่อน (manual allocation of tolerances)

พิจารณาชิ้นงานกลึง (stepped-shaft) ในรูปที่ 1.7 ซึ่งมีการบอกขนาดความยาวของส่วนต่าง ๆ 4 มิติ คือ $A \pm a$, $B \pm b$, $C \pm c$ และ $D \pm d$ จะเห็นได้ว่า 4 มิติบอกตำแหน่งรูปลักษณะ 5 รูปลักษณะ (หรือ 5 ผิว) หรือพูดได้ว่าจำนวนมิติจะน้อยกว่าจำนวนรูปลักษณะที่มีมิติอ้างอิงอยู่ 1 เสมอ



รูปที่ 1.7 ชิ้นงานกลึง

โดยหลักการในการบอกขนาดในแบบจะไม่มีมิติ $X \pm x$ แต่ถ้าต้องการทราบผิว m และ n จะห่างกันเท่าใด เมื่อชิ้นงานได้รับการตัดกลึงตามมิติอื่น ๆ ของชิ้นงาน $X \pm x$ จึงถูกสมมุติขึ้นเป็นระยะห่างระหว่างผิว m และ n ดังนั้น ถ้า X สูงสุด หรือ $X_{\max}$ หรือ $(X + x)$ จึงหาได้จาก:

$$\begin{aligned} X_{\max} &= D_{\max} - A_{\min} - B_{\min} - C_{\min} \\ &= (D + d) - (A - a) - (B - b) - (C - c) \\ &= (D - A - B - C) + (a + b + c + d) \end{aligned} \quad (1.5)$$

ในทำนองเดียวกัน

$$\begin{aligned} X_{\min} \text{ หรือ } (X - x) &= D_{\min} - A_{\max} - B_{\max} - C_{\max} \\ &= (D - d) - (A + a) - (B + b) - (C + c) \\ &= (D - A - B - C) - (a + b + c + d) \end{aligned} \quad (1.6)$$

$$\therefore \frac{X_{\max}}{X_{\min}} = \frac{(D - A - B - C) + (a + b + c + d)}{(D - A - B - C) - (a + b + c + d)}$$

หรือเขียนได้เป็น

$$X \pm x = (D - A - B - C) \pm (a + b + c + d) \quad (1.7)$$

เมื่อเขียนแยกเป็นสมการวงรอบ 2 สมการจะได้

สมการสำหรับขนาดมูลฐาน :

$$X = D - A - B - C \quad (1.8)$$

สมการสำหรับความคลาดเคลื่อน :

$$\pm x = \pm(a + b + c + d) \quad (1.9)$$

จะเห็นได้ว่าความเบี่ยงเบน $\pm x$ หรือ (ความคลาดเคลื่อน $2x$) * นี้เป็นผลรวมของความเบี่ยงเบน (หรือความคลาดเคลื่อน) ของมิติอื่น ๆ ทั้งหมด เราเรียกความเบี่ยงเบน $\pm x$ หรือ (ความคลาดเคลื่อน $2x$) * นี้ว่า ความคลาดเคลื่อนสะสม (tolerance stack)

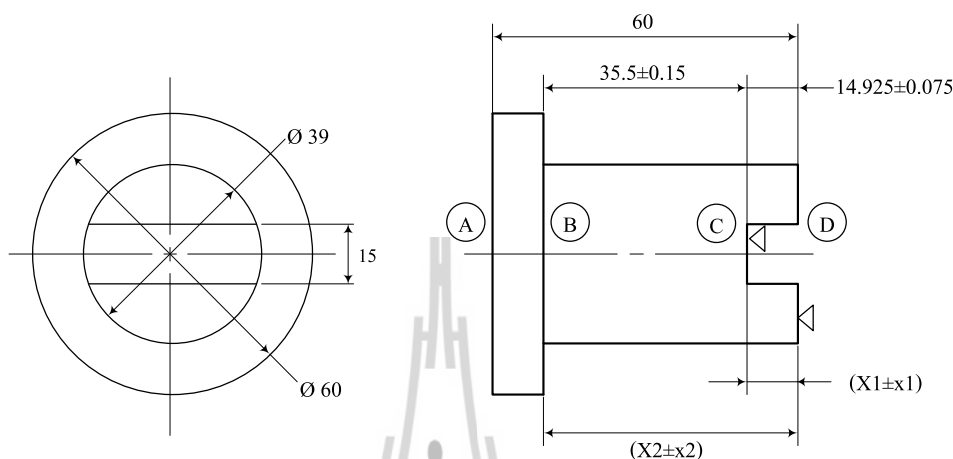
การประยุกต์ใช้งานที่เกิดขึ้นอยู่เสมอ คือ เมื่อเดกซ์ของการบอกขนาดในแบบไม่เหมือนกับเดกซ์ของการผลิต (manufacturing datum) ตัวอย่างเช่น ในรูปที่ 1.8 แสดงแบบชิ้นงานหล่อ ซึ่งต้องการกลึงเพลและกั๊ด (milling) ให้เป็นร่อง ถ้าต้องทำตามแบบจริง ๆ ก็ต้องปาดหน้า (facing) ผิว A และ D ก่อนแล้วจึงกัดร่องลึก 14.925 ± 0.075 mm แล้วจากนั้นจึงกลึง (เพราะต้องวัด 35.5 ± 0.15 mm จากก้นร่อง) แต่ในทางปฏิบัติจะทำงานกลึงก่อนแล้วจึงกัดร่อง นั่นคือ ต้องทราบมิติ $X2 \pm x2$ ถึงจะกลึงได้ ซึ่งสามารถหาได้จากสมการ 1.8 และ 1.9 ดังนี้

$$\text{สมการขนาดมูลฐาน :} \quad X2 - 14.925 = 35.5$$

$$\therefore X2 = 50.425$$

* แม้ว่าความคลาดเคลื่อนของมิติคือความแตกต่างของขนาดใหญ่สุดกับขนาดเล็กสุดก็ตาม แต่ในเรื่องของการวิเคราะห์ความคลาดเคลื่อนนั้น นิยมเรียกความเบี่ยงเบนเป็นความคลาดเคลื่อนเช่น ความคลาดเคลื่อน $\pm x$ ซึ่งในความหมายจริง คือ $2x$

$$\begin{aligned} \text{สมการความคลาดเคลื่อน : } & \pm (x_2 + 0.075) \leq \pm 0.15 \\ & \therefore x_2 = \pm 0.075 \\ & \therefore X_2 \pm x_2 = 50.425 \pm 0.075 \text{ mm} \end{aligned}$$



รูปที่ 1.8 แบบของชิ้นงานตัด-กลึง

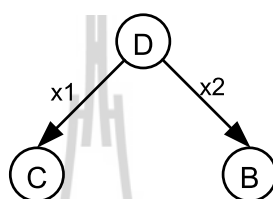
1.5 กราฟต้นไม้ (rooted-tree graph)

เนื่องจากการใช้ค่าสูงสุดและค่าต่ำสุดคำนวณความคลาดเคลื่อนสะสมเป็นวิธีที่ล่าช้า ยากแก่การตรวจสอบและไม่สะดวกต่อการผูกเป็นสมการ จึงจำเป็นต้องค้นหาวิธีการอื่นที่เหมาะสมกว่า การใช้กราฟต้นไม้โดย Whybrew, Britton, Robinson and Sermsuti-anuwat (1990) ประสบผลสำเร็จ ในการสร้างความสัมพันธ์ระหว่างรอยตัดของแผนภูมิความคลาดเคลื่อน (tolerance charts) โดยกราฟดังกล่าวสามารถบันทึกความสัมพันธ์ของความคลาดเคลื่อนของทุกรอยตัดตั้งแต่เริ่มต้นจนสำเร็จเป็นชิ้นงานตามข้อกำหนดในแบบ

ในงานวิจัยนี้อาศัยแนวความคิดดังกล่าว โดยถือว่ามิติแต่ละมิติในชิ้นงาน คือ รอยตัดที่เกิดจากแต่ละขั้นตอนของการตัดโลหะ เพราะฉะนั้นจึงสามารถที่จะนำกราฟต้นไม้มาใช้สร้างความสัมพันธ์ระหว่างความคลาดเคลื่อนได้ และมีการนำแนวความคิดของการใช้กราฟต้นไม้ไปสร้างเงื่อนไขสำหรับแก้ปัญหาด้วยสมการเชิงเส้นแล้วหาคำตอบด้วย Excel (อรรณณ นงนุช และ ยงยุทธ เสริมสุริอวัฒน์, 2552) ซึ่งได้ผลเป็นที่น่าพอใจ

เพื่อแสดงการประยุกต์ใช้วิธีการกราฟต้นไม้ พิจารณารูปที่ 1.8 อีกครั้ง ตัวเลขบอกขนาดด้านบนของรูป คือ ตัวเลขบอกขนาดตามแบบ เราจะเรียกว่า มิติชุดเก่า ซึ่งใช้ผิว C เป็นระนาบอ้างอิงให้กับผิว B และผิว D เป็นการบอกขนาดที่ไม่เหมาะสมสำหรับการผลิต เนื่องจากถ้าให้

ผิว C เป็นผิวอ้างอิงของการบอกขนาด ดังนั้นผิว C จะต้องได้รับการกัดร่องก่อนเพื่อเป็นผิวอ้างอิงให้กับผิว B และผิว D แต่ปัญหาที่จะเกิดตามมาก็คือ ความยุ่งยากในการตั้งเครื่องมือตัด (tool set-up) จะเพิ่มมากขึ้นและการวัดขนาดของความยาวที่จะต้องกลิ้งปอกของ Ø39 ซึ่งสัมพันธ์กับผิว C เพราะฉะนั้นจึงจำเป็นต้องเปลี่ยนแปลงระนาบอ้างอิงของการบอกขนาดในแบบให้เหมาะสมกับระนาบอ้างอิงในการผลิตและการตรวจสอบใหม่ คือ ใช้ผิว D เป็นระนาบอ้างอิงให้กับผิว B และผิว C โดยการบอกขนาดของมิติชุดใหม่นี้อยู่ทางด้านล่างของรูป สำหรับปัญหาในรูปที่ 1.8 เราสามารถเขียนกราฟได้ดังรูปที่ 1.9 และสามารถสร้างสมการวงรอบจากกราฟได้ดังนี้



รูปที่ 1.9 กราฟต้นไม้

สมการวงรอบขนาดมูลฐาน:

$$X1 = Y1 \quad (1.10)$$

$$X2 - X1 = Y2 \quad (1.11)$$

สมการวงรอบสำหรับความคลาดเคลื่อน :

$$x1 \leq y1 \quad (1.12)$$

$$x1 + x2 \leq y2 \quad (1.13)$$

จะเห็นว่าสมการสำหรับขนาดมูลฐานจะบวกหรือลบกันตามเครื่องหมาย แต่สมการของค่าความคลาดเคลื่อนจะเป็นผลบวกเสมอ ซึ่งในการกระจายความคลาดเคลื่อนนั้นจะอาศัยสมการที่ (1.12) และ (1.13) กล่าวคือ ผลรวมของความคลาดเคลื่อนของมิติใหม่ต้องไม่เกินความคลาดเคลื่อนของมิติเดิมที่กำหนดไว้แล้ว

1.6 วัตถุประสงค์

วัตถุประสงค์หลักในการทำงานวิจัยครั้งนี้ คือ การเขียนโปรแกรมคอมพิวเตอร์เพื่อนำมาใช้สำหรับแบ่งค่าความคลาดเคลื่อนให้เหมาะสมกับมิติชุดใหม่เมื่อจุดอ้างอิงของการผลิตและการตรวจสอบแตกต่างกันไปจากจุดอ้างอิงของการบอกขนาดในแบบ โดยอาศัยเทคนิคของกราฟต้นไม้ในการหาความสัมพันธ์ของสมการให้อยู่ในรูปแบบของการโปรแกรมเชิงเส้นตรง (linear programming) และหาผลลัพธ์โดยใช้เทคนิคการหาค่าที่เหมาะสม (optimization techniques) เพื่อให้แน่ใจได้ว่าความคลาดเคลื่อนที่ถูกกำหนดให้กับมิติทั้งหมดนั้นมีค่ามากที่สุดภายใต้เงื่อนไขความสัมพันธ์ของมิติและขณะเดียวกันก็ต้องอยู่ในเกณฑ์ที่สามารถผลิตได้ด้วยกระบวนการผลิตโดยเฉลี่ย (average processing conditions) และนอกจากนี้ยังมีวัตถุประสงค์รองเพื่อหาวิธีที่สะดวกและรวดเร็วสำหรับคำนวณหรือตรวจสอบความคลาดเคลื่อนสะสมเพื่อใช้เป็นพื้นฐานสำหรับการวางแผนขั้นตอนในการตัดโลหะ (machining sequence) ตลอดจนสามารถนำไปใช้ในการออกแบบอุปกรณ์ยึด (work holding devices) เช่น จิกและฟิกซ์เจอร์ได้

1.7 ขอบเขตและสมมุติฐานของงานวิจัย

การวิจัยนี้จะเป็นเรื่องของความคลาดเคลื่อนของมิติเชิงเส้นตรงธรรมดา (conventional tolerancing method) เท่านั้น โดยสมมุติว่าเครื่องมือกลมีความสามารถในการสภาพเฉลี่ย ซึ่งสามารถให้ค่าความคลาดเคลื่อนตามมาตรฐาน ISO 286-2 และ ไม่รวมถึงความคลาดเคลื่อนเชิงเรขาคณิต (geometric tolerance) และยังคงอยู่กับแนวความคิดของการผลิตแบบสับเปลี่ยนได้ 100% (100% interchangeable manufacture) เพียงอย่างเดียว นั่นคือ ยังไม่รวมไปถึงการประยุกต์ใช้หลักสถิติกับความคลาดเคลื่อนทั้งหลายที่เกี่ยวข้อง

1.8 สรุป

ปัญหาการกระจายหรือแบ่งค่าความคลาดเคลื่อน เกิดขึ้นจากปัญหาความคลาดเคลื่อนสะสมของชิ้นงานที่มีระนาบอ้างอิงในการผลิตแตกต่างกันไปจากระนาบอ้างอิงในการบอกขนาด ดังนั้นผู้วางแผนกระบวนการผลิตจำเป็นต้องกระจายค่าความคลาดเคลื่อนให้กับมิติ (dimensions) ในการผลิตหรือในการวัดขนาดใหม่เพื่อให้ความต้องการใช้งานของชิ้นงาน (functional requirement) คงเดิม การวิเคราะห์ความคลาดเคลื่อนชนิดนี้อาศัยเทคนิคของแผนภาพวงรอบ ซึ่งเป็นวิธีการที่นิยมใช้ในอุตสาหกรรมมาเป็นเวลานานกว่า 50 ปี

แผนภาพวงรอบเป็นเทคนิคที่ใช้ในการแสดงความสัมพันธ์ของความต้องการใช้งานของชุดงานประกอบกับมิติใช้งานของชิ้นส่วนประกอบแต่ละชิ้นเพื่อใช้สร้างสมการวงรอบ ซึ่งเป็นวิธีการ

ที่ใช้สำหรับการคำนวณเพื่อปรับปรุงแบบของชิ้นงานให้สะดวกต่อการผลิต นอกจากนั้นการใช้สมการวงรอบสามารถลดความชำนาญและประสบการณ์ของผู้ออกแบบกระบวนการผลิต (process planner) ได้ แต่อย่างไรก็ตามวิธีการนี้ก็ยังมีย่อจำกัดในเรื่องของการนำไปใช้กับชิ้นงานที่มีรูปร่างซับซ้อน ซึ่งยังคงเป็นปัญหาของการออกแบบทางวิศวกรรม ดังนั้นจึงได้มีการพัฒนารูปแบบของการคำนวณโดยการประยุกต์ใช้เทคนิคกราฟต้นไม้กับเทคนิคการหาค่าที่เหมาะสมให้อยู่ในรูปแบบของโปรแกรมคอมพิวเตอร์เพื่อให้สามารถนำไปใช้ในการออกแบบชิ้นงานที่มีรูปร่างซับซ้อนได้ง่ายและรวดเร็วมากยิ่งขึ้น



บทที่ 2

ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

การวิเคราะห์ความคลาดเคลื่อน (tolerance analysis) เป็นเรื่องเกี่ยวกับการพิจารณาแยกองค์ประกอบของความคลาดเคลื่อนของความต้องการใช้งานออกเป็น ส่วน ๆ กำหนดความคลาดเคลื่อนที่เหมาะสมให้กับแต่ละองค์ประกอบย่อยและประเมินความสัมพันธ์ขององค์ประกอบทั้งหลายที่มีผลต่อการใช้งาน เพราะฉะนั้น เรื่องของการวิเคราะห์ความคลาดเคลื่อนของชุดงานประกอบจึงครอบคลุมตั้งแต่แยกความคลาดเคลื่อนของชุดงานประกอบหรือของมิติในชิ้นส่วนแต่ละชิ้นแล้วแบ่งหรือกำหนดค่าความคลาดเคลื่อนที่เหมาะสม ซึ่งเรียกว่า tolerance allocation ไปจนถึงการสร้างความสัมพันธ์ระหว่างความคลาดเคลื่อนที่เรียกว่า tolerance chain หรือ tolerance relationship

2.1 วิธีการวิเคราะห์ความคลาดเคลื่อน

วิธีการวิเคราะห์ความคลาดเคลื่อนแยกออกได้เป็น 2 แบบ คือ วิธีการวิเคราะห์ในชิ้นงานประกอบและวิธีวิเคราะห์ในชิ้นงานเดี่ยว ซึ่งแต่ละวิธีแม้ว่าจะมีความแตกต่างกันในรายละเอียดของการดำเนินการ แต่ถ้าสามารถสรุปได้เป็นสมการแสดงความสัมพันธ์ระหว่างมิติหรือความคลาดเคลื่อนในรูปแบบเดียวกัน คือ สมการวงรอบ (loop equation) สำหรับขนาดมูลฐานและสำหรับความคลาดเคลื่อน

2.1.1 การวิเคราะห์ความคลาดเคลื่อนในชิ้นงานประกอบ

การวิเคราะห์ความคลาดเคลื่อนในชิ้นงานประกอบส่วนใหญ่อาศัยการพิจารณาจากแผนภาพประกอบแล้วใช้ประสบการณ์กำหนดมิติใช้งาน (functional dimensions) ที่มีผลกระทบต่อความต้องการใช้งาน (functional requirement) ของชุดงานประกอบ ดังที่ได้กล่าวมาแล้วในบทที่ 1 ในเรื่องของการสร้างสมการวงรอบจากกฎเกณฑ์ของ Drake หรือของ Liggett เป็นต้น ซึ่งผลของการวิเคราะห์ก็คือ สมการแสดงความสัมพันธ์ของความคลาดเคลื่อนของมิติในชิ้นงานที่เป็นองค์ประกอบ วิธีการเหล่านี้เป็นวิธีการที่ผู้ชำนาญใช้ในทางปฏิบัติอยู่เสมอ และมีหลักการใหญ่ ๆ คล้าย ๆ กัน เพียงแต่มีรายละเอียดที่แตกต่างกันตามประสบการณ์ของแต่ละบุคคลเท่านั้น นอกจากนี้ยังมีความแตกต่างกันในการเรียกความสัมพันธ์ เช่น สมการวงรอบ (loop equations) และ ลูกโซ่ความสัมพันธ์ (chain relationship) เป็นต้น

นักวิจัยทางด้านเทคโนโลยีความคลาดเคลื่อน (tolerance technology) ตระหนักถึงความสัมพันธ์เช่นนี้มานานไม่น้อยกว่า 70 ปี ดังที่ Early and Johnson (1962) ได้ให้ความสำคัญต่อการวิเคราะห์ความคลาดเคลื่อนในงานทางด้านวิศวกรรมเครื่องมือและในการวางแผนกระบวนการผลิต (process planning) ตามลำดับ

การประยุกต์ใช้ความสัมพันธ์ลูกโซ่ที่เด่นชัดที่สุดปรากฏอยู่ในงานของ Björke (1978) โดยการนำสมการดังกล่าวไว้ในโปรแกรมที่เขาพัฒนาขึ้นสำหรับวิเคราะห์ความคลาดเคลื่อนเชิงเส้นตรงของชุดงานประกอบ อย่างไรก็ตาม เนื่องจากความแตกต่างในกฎเกณฑ์การสร้างสมการหรือความสัมพันธ์ของความคลาดเคลื่อนของมิติทำให้ยากแก่การประยุกต์ใช้ Sermstuanuwat (2009) จึงได้สรุปหลักการของ Gladman เพื่อให้การสร้างความสัมพันธ์ของมิติเชิงเส้นตรงสามารถให้คำตอบเดียวกัน เป็นกฎเกณฑ์ 3 ข้อ ดังที่ได้กล่าวมาแล้ว ที่สำคัญคือ มิติใช้งานในชิ้นส่วนประกอบหนึ่ง ๆ จะมีเพียงมิติเดียวเท่านั้น

2.1.2 การวิเคราะห์ความคลาดเคลื่อนในชิ้นงานชิ้นเดียว

เป็นการวิเคราะห์ความคลาดเคลื่อนที่เกิดจากระนาบอ้างอิงในการผลิตแตกต่างจากระนาบอ้างอิงในการออกแบบ วิธีวิเคราะห์ที่นิยมใช้ในทางปฏิบัติและเป็นวิธีดั้งเดิม คือ การหาค่าสูงสุดและค่าต่ำสุดของมิติที่สนใจ จากมิติที่มีอยู่ทั้งหมด ดังที่ได้กล่าวมาแล้วในบทที่ 1 แต่ข้อเสียของวิธีนี้คือ ต้องอาศัยประสบการณ์ของผู้วิเคราะห์และเสียเวลาในการตัดสินใจเลือกมิติที่เกี่ยวข้อง การวิเคราะห์ความคลาดเคลื่อนที่จัดว่าเป็นระบบมากที่สุด คือ การสร้างแผนภูมิความคลาดเคลื่อน ซึ่งใช้ในทางปฏิบัติตั้งแต่ต้นศตวรรษที่ 20 จนมีบทความแสดงผลงานวิจัยอย่างต่อเนื่องในช่วง ค.ศ. 1950 เช่น ๆ อย่างไรก็ตาม เทคนิคหรือวิธีการนี้ยังคงจำกัอยู่ในวงแคบเพราะต้องใช้เวลาทำแผนภูมิมากและหลักการของแผนภูมิเองก็ยังไม่ชัดเจนนักจนกระทั่งเมื่อปี ค.ศ. 1984 Ahluwalia and Korolin (1984) ได้ตีพิมพ์บทความของการประยุกต์ใช้คอมพิวเตอร์สร้างแผนภูมิความคลาดเคลื่อน เทคนิคนี้จึงฟื้นกลับมาอีกครั้ง และในปีเดียวกันนั้นเอง Farmer and Harris (1984) ได้แสดงการกระจายความคลาดเคลื่อนให้กับมิติต่าง ๆ ด้วยคอมพิวเตอร์

ทั้งแผนภูมิความคลาดเคลื่อนและผลงานของ Farmer and Harris (1984) ต่างก็เป็นการวิเคราะห์ความคลาดเคลื่อนสะสมและมีผลต่อความต้องการใช้งานของชิ้นส่วนที่พิจารณา ถึงแม้ว่า Farmer and Harris จะใช้สมการความคลาดเคลื่อนเป็นหลักในการกระจายความคลาดเคลื่อน แต่การกำหนดมิติใช้งานในสมการความคลาดเคลื่อนยังคงต้องอาศัยประสบการณ์ในทางปฏิบัติเป็นสำคัญ

2.2 การใช้คอมพิวเตอร์ช่วยในการวิเคราะห์ความคลาดเคลื่อน

การใช้คอมพิวเตอร์วิเคราะห์ความคลาดเคลื่อนอย่างจริงจังมีมานานกว่า 30 ปี ผลงานของ Bjørke (1978) แสดงให้เห็นถึงความสำเร็จในการพัฒนาโปรแกรมสำหรับคำนวณค่าความคลาดเคลื่อนสะสม และทำการหาค่าความคลาดเคลื่อนที่เหมาะสมในเครื่องคอมพิวเตอร์ขนาดใหญ่ (mainframe) แต่ในเวลานั้นงานวิจัยด้านนี้ยังอยู่ในวงแคบ ซึ่งอาจเป็นเพราะว่าขีดความสามารถที่จำกัดของเครื่องคอมพิวเตอร์เองหรือราคาคอมพิวเตอร์ที่ยังสูงอยู่ บทบาทของคอมพิวเตอร์เริ่มชัดเจนขึ้นเมื่อ Ahaluwalia and Karolin (1984) รายงานการใช้คอมพิวเตอร์สร้างแผนภูมิความคลาดเคลื่อน (tolerance chart) และในปีเดียวกัน Farmer and Harris (1984) ได้พัฒนาโปรแกรมคอมพิวเตอร์สำหรับคำนวณค่าคลาดเคลื่อนให้กับมิติต่าง ๆ ที่ถูกกำหนดขึ้นเมื่อเกิดการเปลี่ยนแปลงเดกซ์ของการบอกขนาดในแบบวิศวกรรมโดยใช้วิธีการสร้างสมการวงรอบเป็นพื้นฐานของการกระจายค่าความคลาดเคลื่อน ผลงานที่กล่าวข้างต้นอาจถือได้ว่าเป็นจุดเริ่มต้นของการประยุกต์ใช้คอมพิวเตอร์สำหรับวิเคราะห์ความคลาดเคลื่อนให้เป็นไปอย่างกว้างขวาง และหลังจากนั้น Fainguelernt, Weill and Bourdet (1986) พยายามแสดงให้เห็นถึงความเป็นไปได้ในการคำนวณหาขนาดและค่าความคลาดเคลื่อนบนเครื่องคอมพิวเตอร์ขนาดเล็ก (micro computer) โดยใช้เทคนิคการหาค่าที่เหมาะสมด้วยวิธีการโปรแกรมเชิงเส้นตรง

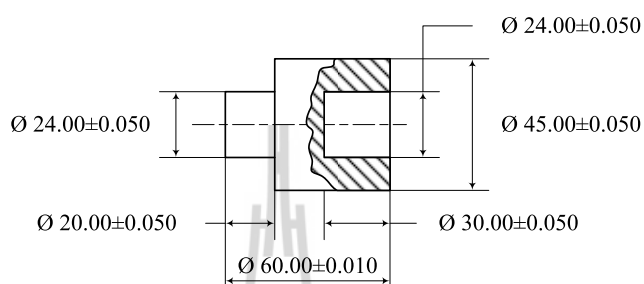
2.3 การประยุกต์ใช้เทคนิคกราฟในการวิเคราะห์ความคลาดเคลื่อน

การนำเทคนิคของกราฟมาประยุกต์ใช้กับการสร้างความสัมพันธ์ของความคลาดเคลื่อนเป็นวิธีหนึ่งที่ทำได้ง่ายและสามารถนำไปสร้างเป็นแบบจำลองทางคณิตศาสตร์ (mathematical model) โดยใช้ความเชื่อมโยงของเส้นทาง (path) ที่แสดงความสัมพันธ์ของมิติต่าง ๆ เป็นตัวสร้างสมการความสัมพันธ์ของความคลาดเคลื่อน เพื่อใช้ในการคำนวณหาค่าที่เหมาะสมด้วยวิธีการโปรแกรมเชิงเส้นตรง

จากงานวิจัยของ Irani, Mittal and Lehtihet (1989) ได้นำเสนอการใช้ directed graph เพื่อสร้างความสัมพันธ์ระหว่างรอยตัดของแผนภูมิความคลาดเคลื่อนโดยใช้วิธีการที่เรียกว่า path-tracing ในการบ่งชี้ความเชื่อมโยงของความคลาดเคลื่อน อย่างไรก็ตาม วิธีการนี้ก็ยังเป็นวิธีที่ยากต่อการนำไปใช้ในทางปฏิบัติเพราะไม่สามารถบ่งชี้ผิวหน้าในการตัด (machined surface) ที่แน่นอนได้ และมีกฎเกณฑ์ที่ซับซ้อน

2.4 การใช้เทคนิคกราฟต้นไม้ในการทำแผนภูมิความคลาดเคลื่อน

Whybrew, Britton, Robinson and Sermsuti-anuwat (1990) ได้ประยุกต์ใช้กราฟต้นไม้ (rooted-tree graph) สำหรับช่วยสร้างแผนภูมิความคลาดเคลื่อน ซึ่งได้อธิบายถึงวิธีการสร้างกราฟต้นไม้ (rooted-tree graph) สำหรับการลำดับขั้นตอนการตัดโลหะในแผนภูมิความคลาดเคลื่อน โดยมีรายละเอียดดังนี้



รูปที่ 2.1 ชิ้นงานสำเร็จจากการตัดโลหะมิติต่าง ๆ (Whybrew *et al.*, 1990)

Line no.	Op.no.	Face	M/C	Working dim.		A	B	C	D	Stock removal	
				basic	$\pm$ Tol.					basic	$\pm$ Tol.
0											
1	10	A	saw	61.20	0.200						
2	30	B	lathe	40.60	0.025					20.60	solid
3	40	A1	lathe	60.70	0.025					0.50	0.225
4	50	D1	lathe	60.20	0.025					0.50	0.050
5	60	C	drill	30.60	0.025					29.60	solid
6	70	C1	lathe	30.10	0.025					0.50	0.050
7	80	A2	grind	60.10	0.005					0.10	0.030
8	90	D2	grind	60.00	0.005					0.10	0.010
				blueprint						resultant	
				20.00	0.050					20.00	0.080
				30.00	0.050					30.00	0.060
				60.00	0.010					60.00	0.005

รูปที่ 2.2 แผนภูมิความคลาดเคลื่อน (Whybrew *et al.*, 1990)

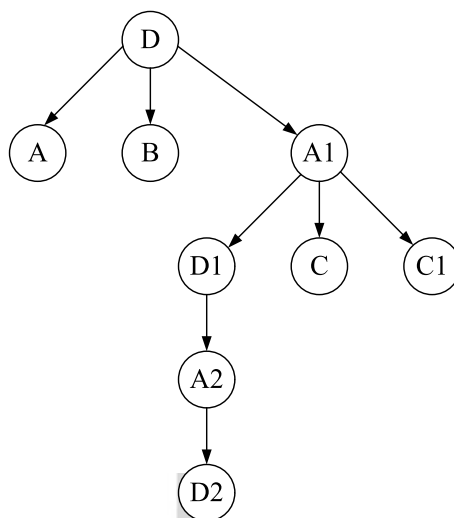
จากรูปที่ 2.2 แสดงขั้นตอนการตัดโลหะที่เรียกว่า แผนภูมิความคลาดเคลื่อนซึ่งสามารถสรุปขั้นตอนได้ดังนี้

ขั้นตอนที่ 1 Operation 10 เลื่อยตัด	ได้ผิว A	โดยอ้างอิงจากผิว D
ขั้นตอนที่ 2 Operation 30 กลึงตัด	ได้ผิว B	โดยอ้างอิงจากผิว D
ขั้นตอนที่ 3 Operation 40 กลึงตัด	ได้ผิว A1	โดยอ้างอิงจากผิว D
ขั้นตอนที่ 4 Operation 50 กลึงตัด	ได้ผิว D1	โดยอ้างอิงจากผิว A1
ขั้นตอนที่ 5 Operation 60 เจาะรู	ได้ผิว C	โดยอ้างอิงจากผิว A1
ขั้นตอนที่ 6 Operation 70 กลึงตัด	ได้ผิว C1	โดยอ้างอิงจากผิว A1
ขั้นตอนที่ 7 Operation 80 เจียรระไน	ได้ผิว A2	โดยอ้างอิงจากผิว D1
ขั้นตอนที่ 8 Operation 90 เจียรระไน	ได้ผิว D2	โดยอ้างอิงจากผิว A2

จากการพิจารณาขั้นตอนทั้งหมดสามารถสรุปความสัมพันธ์ที่ถูกตัดและผิวที่เป็นจุดอ้างอิงในการวัดขนาดได้ดังตารางที่ 2.1 และสามารถแสดงเป็นกราฟต้นไม้ได้ดังรูปที่ 2.3

ตารางที่ 2.1 แสดงความสัมพันธ์ของผิวที่ถูกตัดและผิวที่เป็นจุดอ้างอิงในการวัดขนาด (Whybrew *et al.*, 1990)

การดำเนินการ (Operation)	ผิวที่ถูกตัด (machined face)	จุดอ้างอิงในการวัดขนาด (location face)
10	A	D
30	B	D
40	A1	D
50	D1	A1
60	C	A1
70	C1	A1
80	A2	D1
90	D2	A2



รูปที่ 2.3 กราฟต้นไม้ (Whybrew *et al.*, 1990)

จากกราฟต้นไม้ในรูปที่ 2.3 เราสามารถคำนวณความคลาดเคลื่อนของมิติที่สนใจได้ เช่น มิติระหว่างผิว A2 กับผิว B จะมีความคลาดเคลื่อนเท่ากับผลรวมของความคลาดเคลื่อนของการตัดที่อยู่ในเส้นทางระหว่างผิว A2 และผิว B นั่นคือ ผลรวมของความคลาดเคลื่อนของ B - D, D - A1, A1 - D1 และ D1 - A2 เท่ากับ $\pm (0.025+0.025+0.025+0.005) = \pm 0.080$ mm เป็นต้น

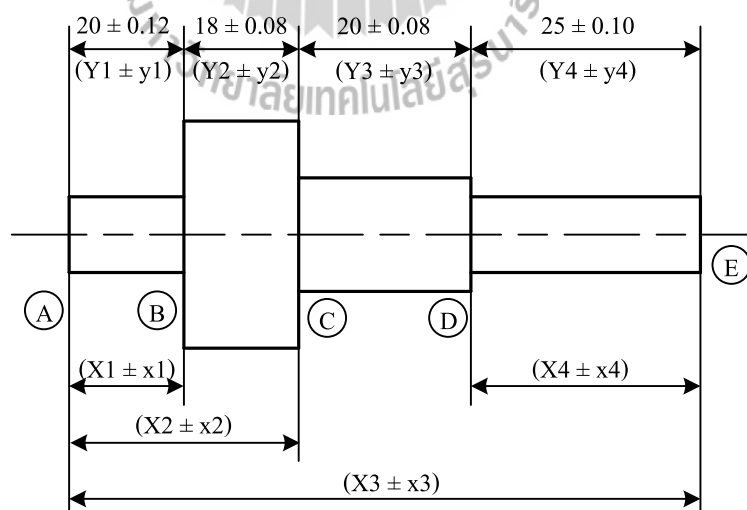
วิธีการของ Whybrew *et al.* (1990) เป็นวิธีที่ง่ายต่อการทำความเข้าใจและไม่ต้องอาศัยกฎเกณฑ์ที่ยู่ยากซับซ้อนและสามารถนำมาประยุกต์ใช้ในการสร้างความสัมพันธ์ของมิติได้อย่างดี จึงเป็นจุดเริ่มต้นในการนำไปสร้างเป็นแบบจำลองทางคณิตศาสตร์ของสมการเชิงเส้นตรงเพื่อให้สามารถดำเนินการได้ง่ายบนคอมพิวเตอร์ โดย Ngoi (1992) และ Ji (1993) ได้ใช้เทคนิคของกราฟต้นไม้สร้างแบบจำลองทางคณิตศาสตร์สำหรับการกำหนดความคลาดเคลื่อนในแผนภูมิความคลาดเคลื่อน โดยนำเสนอตัวอย่างในการเปรียบเทียบ การแก้ปัญหาระหว่างวิธีโปรแกรมเชิงเส้นตรงกับวิธีการคำนวณแบบลองผิดลองถูกด้วยมือ (Eary and Johnson, 1962) จากนั้นก็มีการนำแบบจำลองทางคณิตศาสตร์นี้ไปพัฒนาต่อเพื่อหาขนาดที่เกิดจากการตัดในแต่ละครั้ง (working dimensions) และแบ่งความคลาดเคลื่อนโดยอัตโนมัติในแผนภูมิความคลาดเคลื่อน (Ngoi and Ong, 1993a; 1993b) ซึ่งใช้เทคนิคที่เรียกว่า path-tracing technique และแก้ปัญหาโดยใช้ระบบของสมการเชิงเส้นตรง นอกจากนั้นยังสามารถนำไปใช้อธิบายวิธีการจัดลำดับกระบวนการผลิตในแผนภูมิความคลาดเคลื่อนโดยอัตโนมัติ (Ngoi and Fang, 1994; Ngoi and Cheong, 1998) ซึ่งงานวิจัยที่กล่าวมาข้างต้นยังคงต้องอาศัยโปรแกรม LINDO ในการหาผลลัพธ์ที่เหมาะสมที่สุดให้กับแบบจำลองทางคณิตศาสตร์ที่สร้างขึ้นมา

นอกจากนี้เทคนิคของกราฟต้นไม้ยังสามารถนำไปประยุกต์ใช้ในเขียนโปรแกรมคอมพิวเตอร์ช่วยในการวางแผนกระบวนการผลิต (Sermutsi-anuwat, Whybrew and McCallion, 1995) และยังสามารถนำไปใช้แสดงความสัมพันธ์ระหว่างการออกแบบกับการวางแผนการผลิตในชิ้นงานที่มีรูปทรงสามมิติได้อีกด้วย (Thimm, Britton and Fok, 2004)

2.5 วิธีการสร้างกราฟต้นไม้แสดงความสัมพันธ์ของความคลาดเคลื่อน

เมื่อพิจารณาการใช้เทคนิคกราฟต้นไม้ในการบันทึกรอยตัดในแผนภูมิความคลาดเคลื่อน จะเห็นได้ว่าความสัมพันธ์ของความคลาดเคลื่อนของรอยตัดทุกรอยตั้งแต่ต้น จนเป็นชิ้นงานสำเร็จรูป สามารถสรุปได้เป็นกราฟชุดเดียว เพราะฉะนั้นเราอาจพิจารณาว่าเมื่อมิติชุดใหม่มีระนาบอ้างอิงของการบอกขนาดเปลี่ยนแปลงเกิดจากมิติชุดเก่าที่ถูกตัดมาก่อน นั่นคือ เราสามารถใช้เทคนิคกราฟต้นไม้แสดงความสัมพันธ์ของความคลาดเคลื่อนระหว่างมิติชุดเก่าและมิติชุดใหม่ได้

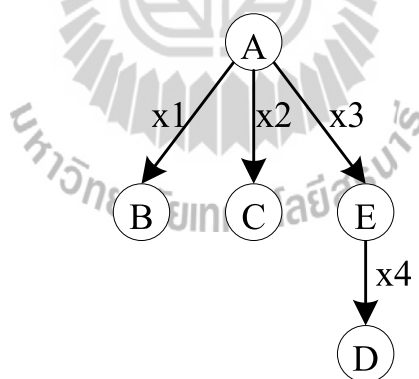
รูปที่ 2.4 แสดงชิ้นงานกลึงเพลาลูกกลิ้งสี่เหลี่ยมซึ่งมีการบอกขนาดความคลาดเคลื่อนของมิติความยาวเป็นค่าเบี่ยงเบนเท่า ๆ กันจากขนาดมูลฐาน (equally bilateral tolerancing) เพื่อให้เข้าใจได้ง่ายขึ้น ในรูปนี้ได้เขียนกำกับมิติเก่าทุกมิติด้วยสัญลักษณ์ของการบอกขนาดเป็น $(Y1 \pm y1)$ ถึง $(Y4 \pm y4)$ ซึ่งเป็นมิติที่อยู่ด้านบนทั้งหมดของรูปและกำหนดให้ $(X1 \pm x1)$ ถึง $(X4 \pm x4)$ เป็นมิติใหม่เมื่อระนาบอ้างอิงของการบอกขนาดเปลี่ยนแปลง ซึ่งอยู่ด้านล่างทั้งหมดของรูป และใช้ตัวอักษร A ถึง E แทนผิวชิ้นงานที่เกี่ยวข้องกับมิติของขนาด



รูปที่ 2.4 ชิ้นงานตัวอย่างสำหรับการคำนวณมิติความยาว 4 มิติ

วิธีการสร้างความสัมพันธ์ของความคลาดเคลื่อนของมิติชุดเก่าและมิติชุดใหม่จะอาศัยการสร้างกราฟต้นไม้ของมิติใหม่เป็นหลักในการกำหนดความสัมพันธ์ โดยมีขั้นตอนดังนี้

- 1) สกัดภาพฉายตั้งฉากของชิ้นงานที่แสดงมิติซึ่งเกี่ยวข้องกับการเปลี่ยนแปลงระนาบอ้างอิง และกำหนดชื่อเป็นตัวอักษรให้กับผิวงานที่เป็นขอบเขตของขนาดทั้งหมด (ผิว A, B, C, D, E ในวงกลมของรูปที่ 2.4)
- 2) สร้างกราฟสำหรับมิติใหม่ในรูปที่ 2.4 โดยการเขียนตัวอักษรของผิวเริ่มต้นไว้ตอนบนสุด โดยเริ่มจากผิวใดของชิ้นงานก็ได้ ในรูปที่ 2.5 เริ่มจาก A และเรียก A ว่า โหนด (node) A
- 3) พิจารณามิติใดบ้างที่เกี่ยวข้องกับโหนดเริ่มต้นนี้ ให้ลากลูกศรออกจากโหนดเริ่มต้นชี้ไปยังอักษรที่ระบุผิวปลายของมิติเหล่านั้น ผลของขั้นตอนนี้ คือเส้นเชื่อมโยง (links) AB, AC, และ AE ของมิติ X1 และ X2 และ X3 ตามลำดับ และมีโหนดที่เกิดขึ้นใหม่คือ B, C และ E
- 4) พิจารณาโหนดที่เกิดขึ้นใหม่คือ B, C และ E โดยลากลูกศรออกจากโหนดแล้วทำเช่นเดียวกับขั้นตอนที่ 3 จนครบทุกมิติ จะได้กราฟต้นไม้ของมิติใหม่ที่สมบูรณ์ ดังรูปที่ 2.5



รูปที่ 2.5 กราฟต้นไม้สำหรับมิติใหม่

จากกราฟต้นไม้รูปที่ 2.5 เราสามารถสรุปความสัมพันธ์ระหว่างความคลาดเคลื่อนของมิติชุดเก่าและเส้นทางของกราฟที่เกี่ยวข้องกับมิติชุดใหม่ได้ดังตารางที่ 2.2

ตารางที่ 2.2 ความสัมพันธ์ของความคลาดเคลื่อนของมิติเก่าและมิติใหม่

ความคลาดเคลื่อนของมิติเก่า	ผิวชิ้นงานของมิติเก่า	ความคลาดเคลื่อนของมิติใหม่
y1 (± 0.12)	AB	x1
y2 (± 0.08)	BC	x1, x2
y3 (± 0.08)	CD	x2, x3, x4
y4 (± 0.10)	DE	x4

2.6 การโปรแกรมเชิงเส้นตรงกับการกระจายความคลาดเคลื่อน

การโปรแกรมเชิงเส้นตรง หมายถึง ระเบียบวิธีทางคณิตศาสตร์เพื่อหาค่าที่เหมาะสมที่สุด ซึ่งเป็นค่าสูงสุด (maximize) หรือต่ำสุด (minimize) ของเป้าหมายที่กำหนด ภายใต้เงื่อนไขบางประการ โดยเป้าหมายจะต้องแสดงอยู่ในรูปสมการเส้นตรง สำหรับเงื่อนไขอาจอยู่ในรูปสมการหรืออสมการเส้นตรงก็ได้

2.6.1 สมการเป้าหมาย (objective function)

สมการเป้าหมายสำหรับปัญหาการกระจายค่าความคลาดเคลื่อน คือ การสร้างสมการเชิงเส้นตรงแสดงความสัมพันธ์ระหว่างมิติเก่ากับมิติใหม่ โดยใช้หลักการคือ ผลรวมของความคลาดเคลื่อนสะสมของมิติใหม่จะต้องมีค่ามากที่สุดเพื่อให้สามารถผลิตได้ง่ายและชิ้นงานทำหน้าที่ได้เสมอ โดยสามารถเขียนเป็นรูปแบบของสมการเป้าหมายได้ดังนี้

$$\text{Max } Z = x_1 + x_2 + \dots + x_n \quad \text{โดยที่ } i=1, \dots, n \quad (2.1)$$

กำหนดให้ x_i = ค่าความคลาดเคลื่อนที่มิติใหม่ใด ๆ

n = จำนวนของมิติทั้งหมด

2.6.2 สมการข้อจำกัด (constraints)

สมการข้อจำกัดของปัญหาปัญหาการกระจายค่าความคลาดเคลื่อน นั้นสามารถแบ่งออกเป็น 2 เงื่อนไขดังนี้

1) ผลรวมของความคลาดเคลื่อนสะสมของมิติใหม่จะต้องมีค่าน้อยกว่าความคลาดเคลื่อนของมิติเก่าที่เกี่ยวข้องเพื่อให้ชิ้นงานสามารถใช้งานได้เสมอ ซึ่งสามารถสร้างสมการแสดงความสัมพันธ์ระหว่างมิติชุดเก่าและมิติชุดใหม่ได้ เช่น มิติเก่า Y3 ระหว่างผิว C และ D ในแบบ (รูปที่ 2.4) จะเกิดจากเส้นทางในกราฟ (รูปที่ 2.5) จาก C ถึง D ประกอบด้วยมิติ X2 X3 และ X4 ซึ่งจะเขียนเป็นอสมการแสดงความสัมพันธ์ของความคลาดเคลื่อนของมิติได้

เป็น $x_2 + x_3 + x_4 \leq y_3$ และสามารถเขียนรูปแบบของสมการข้อจำกัดได้ดังนี้

$$x_1 \leq y_1 \quad \text{โดยที่ } i = 1, \dots, n \quad (2.2)$$

$$x_1 + x_2 \leq y_2 \quad \text{โดยที่ } i = 1, \dots, n \quad (2.3)$$

$$x_2 + x_3 + x_4 \leq y_3 \quad \text{โดยที่ } i = 1, \dots, n \quad (2.4)$$

$$x_4 \leq y_4 \quad \text{โดยที่ } i = 1, \dots, n \quad (2.5)$$

กำหนดให้ $y_i =$ ค่าความคลาดเคลื่อนที่มีมิติใหม่ใด ๆ

$n =$ จำนวนของมิติทั้งหมด

2) ข้อจำกัดที่เกี่ยวกับกระบวนการผลิตต้องมีความสามารถในการผลิตไม่น้อยกว่าค่าเกรดความคลาดเคลื่อนสากล (International Tolerance Grade ISO 286-1) (รายละเอียดแสดงในภาคผนวก ก) ซึ่งสามารถเขียนรูปแบบของสมการข้อจำกัดได้ดังนี้

$$x_i \geq \text{เกรดความคลาดเคลื่อนสากล} \quad \text{โดยที่ } i = 1, \dots, n \quad (2.6)$$

จากชิ้นงานตัวอย่างในรูปที่ 2.4 สามารถสรุปเป็นตัวแทนของปัญหาโปรแกรมเชิงเส้นตรงทั้งหมดได้ดังนี้

สมการเป้าหมาย

$$\text{Max } Z = x_1 + x_2 + x_3 + x_4 \quad (2.7)$$

สมการข้อจำกัด

$$x_1 \leq 0.12 \quad (2.8)$$

$$x_1 + x_2 \leq 0.08 \quad (2.9)$$

$$x_2 + x_3 + x_4 \leq 0.08 \quad (2.10)$$

$$x_4 \leq 0.10 \quad (2.11)$$

$$x_1 \geq 0.042 \quad (2.12)$$

$$x_2 \geq 0.035 \quad (2.13)$$

$$x_3 \geq 0.042 \quad (2.14)$$

$$x_4 \geq 0.042 \quad (2.15)$$

สมการที่ (2.12) ถึง (2.15) เป็นข้อจำกัดที่เกี่ยวกับกระบวนการผลิต ในที่นี้กำหนดให้มีความสามารถในการผลิตได้ไม่น้อยกว่าค่าเกรดความคลาดเคลื่อนสากล 10 หรือ IT 10 ทั้งหมด

2.7 ข้อจำกัดของโปรแกรมสำเร็จรูปทั่วไป

โปรแกรมสำเร็จรูปส่วนใหญ่ออกแบบมาเพื่อใช้สำหรับการแก้ปัญหาทั่วไป โดยไม่เฉพาะเจาะจงต่องานใดงานหนึ่ง เช่น การหาผลลัพธ์ที่เหมาะสมให้กับปัญหาการแบ่งค่าความคลาดเคลื่อน ดังนั้นการใช้โปรแกรมสำเร็จรูปทั่วไปจึงมีข้อจำกัดคือ ผู้ใช้ต้องทำการสร้างสมการขึ้นใหม่ทุกครั้งที่มีการเปลี่ยนแปลงของค่า IT No ดังตัวอย่างในหัวข้อ 2.6 จะพบว่าในตอนเริ่มต้นได้กำหนดค่า IT No ไว้ที่ IT 10 ทั้งหมด แต่ผลการคำนวณด้วยโปรแกรมสำเร็จรูปทั่วไปไม่สามารถคำนวณหาคำตอบได้ จึงต้องมีการปรับเปลี่ยนและกำหนดค่า IT No เริ่มต้นใหม่ ซึ่งอาจต้องเสียเวลาทำการทดลองเปลี่ยนค่าใหม่หลายครั้งกว่าจะได้คำตอบที่ดีที่สุด ส่งผลให้ต้องสิ้นเปลืองเวลาในการทำงานมากขึ้น ดังนั้นในงานวิจัยนี้จึงได้พัฒนาโปรแกรมคอมพิวเตอร์ขึ้นใหม่ โดยรวบรวมขั้นตอนการทำงานตั้งแต่วิธีในการสร้างกราฟต้นไม้ไปจนถึงการคำนวณหาผลลัพธ์ที่เหมาะสมที่สุดของปัญหาการแบ่งค่าความคลาดเคลื่อนโดยใช้วิธีซิมเพล็กซ์ โปรแกรมสามารถปรับเปลี่ยนค่า IT No ได้เองโดยอัตโนมัติและให้ครอบคลุมได้มากที่สุด เพื่อลดขั้นตอนการทำงานที่ไม่จำเป็นและให้ผู้ใช้สามารถทำงานได้ง่ายและมีความรวดเร็วยิ่งขึ้น

2.8 สรุป

วิธีการวิเคราะห์ความคลาดเคลื่อนได้รับการพัฒนามาตลอด 50 ปีที่ผ่านมา เพราะขนาดของความคลาดเคลื่อนที่กำหนดให้กับมิติของชิ้นงานมีผลโดยตรงต่อค่าใช้จ่ายในการผลิต งานวิจัยทางด้านวิศวกรรมความคลาดเคลื่อน (Tolerance Engineering) จึงเป็นที่สนใจของนักวิจัยทั้งหลายที่เกี่ยวข้องกับการวางแผนกระบวนการ (process planning) การออกแบบผลิตภัณฑ์ (product design) การวัดละเอียด (metrology) และการควบคุมคุณภาพ (quality control) ยิ่งมีการประยุกต์ใช้คอมพิวเตอร์ด้วยแล้วจึงเป็นแรงส่งให้เกิดการขยายตัวของงานวิจัยทางการวิเคราะห์ความคลาดเคลื่อนอย่างแพร่หลาย กราฟต้นไม้ (rooted-tree graph) เป็นผลพวงจากวิวัฒนาการนั้นด้วย ซึ่งเราสามารถนำเทคนิคของกราฟต้นไม้มาประยุกต์ใช้กับการสร้างความสัมพันธ์ของความคลาดเคลื่อนจึงเป็นวิธีที่สามารถทำได้ง่าย รวดเร็วและสามารถนำไปสร้างเป็นแบบจำลองทางคณิตศาสตร์ โดยใช้ความเชื่อมโยงของเส้นทางที่แสดงความสัมพันธ์ของมิติต่าง ๆ เป็นตัวสร้างสมการความสัมพันธ์ของความคลาดเคลื่อน เพื่อใช้ในการคำนวณหาค่าที่เหมาะสมด้วยวิธีการโปรแกรมเชิงเส้นตรง ซึ่งในงานวิจัยนี้จะได้ใช้เทคนิคที่กล่าวมาข้างต้นเพื่อนำเสนอในรูปแบบของโปรแกรมคอมพิวเตอร์สำเร็จรูป



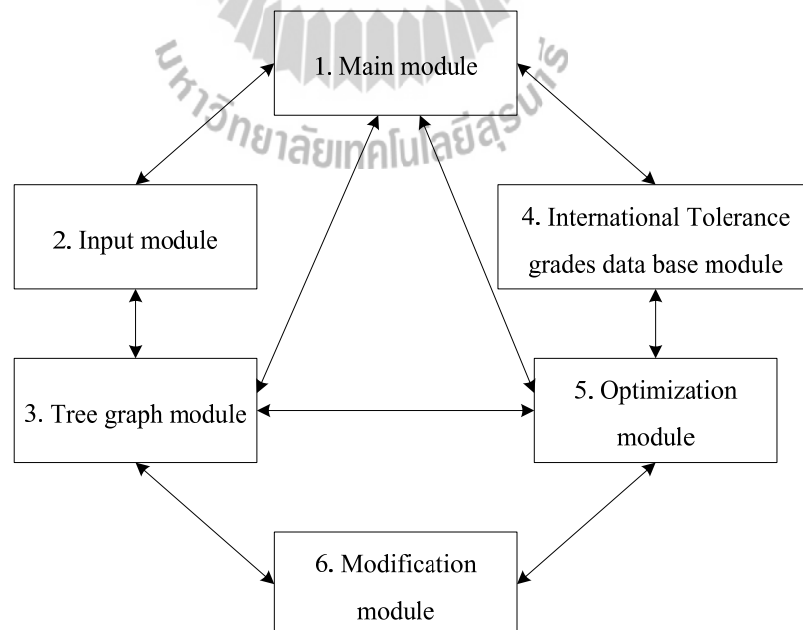
บทที่ 3

โครงสร้างโปรแกรม

งานวิจัยนี้ได้นำเทคนิคของกราฟต้นไม้มาใช้เป็นหลักเกณฑ์ในการสร้างสมการความสัมพันธ์ของความคลาดเคลื่อนในรูปแบบของสมการเชิงเส้นตรงเพื่อคำนวณหาผลลัพธ์ที่เหมาะสมภายใต้เงื่อนไขของค่าความคลาดเคลื่อนมาตรฐานที่กำหนด มาพัฒนาเป็นโปรแกรมชื่อว่า “TOLOP” (Tolerance Optimization Program) โดยการพัฒนาโปรแกรมได้เลือกใช้โปรแกรมไมโครซอฟท์วิซวลเบสิก 6 (Microsoft Visual Basic 6 หรือ VB 6) สำหรับเขียนคำสั่ง เนื่องจากเป็นโปรแกรมที่สามารถสร้างส่วนติดต่อกับผู้ใช้ (user interface) ได้อย่างรวดเร็วและสามารถเรียนรู้วิธีการเขียนโปรแกรมได้ด้วยตนเอง

3.1 องค์ประกอบของโปรแกรม TOLOP

องค์ประกอบหลักของโปรแกรมสามารถแบ่งออกเป็น 6 ส่วน หรือ 6 โมดูล ซึ่งมีความสัมพันธ์เชื่อมโยงกันแสดงดังในรูปที่ 3.1



รูปที่ 3.1 องค์ประกอบหลักของโปรแกรม TOLOP

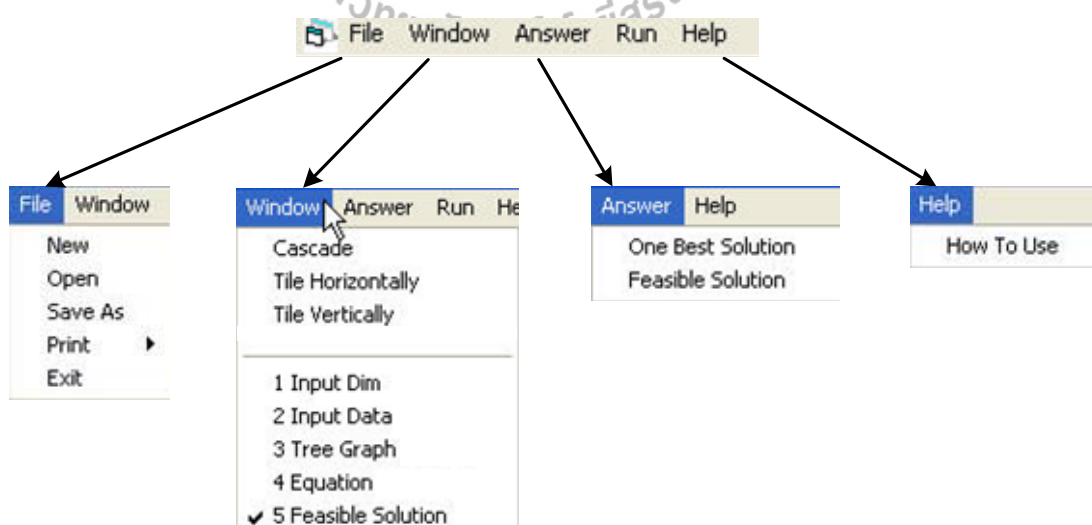
3.1.1 โมดูลหลัก (main module)

โมดูลหลักมีหน้าที่เป็นศูนย์กลางสำหรับเรียกใช้งานในส่วนต่าง ๆ ของโปรแกรม โดยพื้นฐานแล้วโมดูลหลัก จะเป็นหน้าจอหน้าต่างแรกที่ปรากฏขึ้นหลังจากที่เข้าสู่โปรแกรมครั้งแรก (รูปที่ 3.2) มีหน้าจอที่ประกอบด้วย ชื่อโปรแกรม ชื่อผู้พัฒนาโปรแกรม นอกจากนั้นยังมีแถบเมนูและปุ่มคำสั่งต่าง ๆ เหมือนกับหน้าจอของโปรแกรมใช้งานทั่วไป มีไว้ให้ผู้ใช้ติดต่อและสั่งงานโปรแกรม (รูปที่ 3.3) ปุ่มคำสั่งเหล่านี้ได้แก่

- File สำหรับจัดการเกี่ยวกับไฟล์ เช่น เปิดไฟล์ บันทึกข้อมูล หรือ พิมพ์ข้อมูล เป็นต้น
 - Window สำหรับแสดงรายละเอียดหน้าจอของโมดูลต่าง ๆ ที่มีการทำงานอยู่ทั้งหมดได้แก่ หน้าจอแสดงการป้อนจำนวนมิติ (Input Dim) หน้าจอแสดงการป้อนข้อมูล (Input Data) หน้าจอแสดงกราฟต้นไม้ (Tree Graph) หน้าจอแสดงสมการเชิงเส้น (Equation) หน้าจอแสดงคำตอบที่เหมาะสมทั้งหมด (Feasible Solution) และหน้าจอแสดงคำตอบที่ดีที่สุด (One Best Solution)
 - Answer สำหรับแสดงคำตอบ สามารถแสดงได้ทั้งคำตอบที่ดีที่สุดซึ่งมีค่า objective function มากที่สุด (One Best Solution) และคำตอบที่เหมาะสมอื่น ๆ (Feasible Solution) ซึ่งหมายถึงคำตอบที่ผู้ใช้สามารถเลือกมาใช้งานได้ ถึงแม้จะไม่ใช้คำตอบที่มีค่าผลรวมของความคลาดเคลื่อนมากที่สุด แต่ก็ยังเป็นคำตอบที่อยู่ในเงื่อนไขของเกรดความคลาดเคลื่อนมาตรฐานที่เรากำหนดให้
 - Run สำหรับเริ่มต้นใช้งานโปรแกรมเพื่อเข้าสู่หน้าจอสำหรับป้อนจำนวนมิติ
 - Help เป็นปุ่มที่มีไว้เพื่อแนะนำรายละเอียดและวิธีการใช้งานของโปรแกรม
- ส่วนด้านล่างของหน้าจอจะมีแถบสถานะ (status bar) เพื่อแสดงสถานะของไฟล์ที่เก็บบันทึกข้อมูลพร้อมทั้งแสดงวันที่และเวลาขณะที่ใช้โปรแกรม

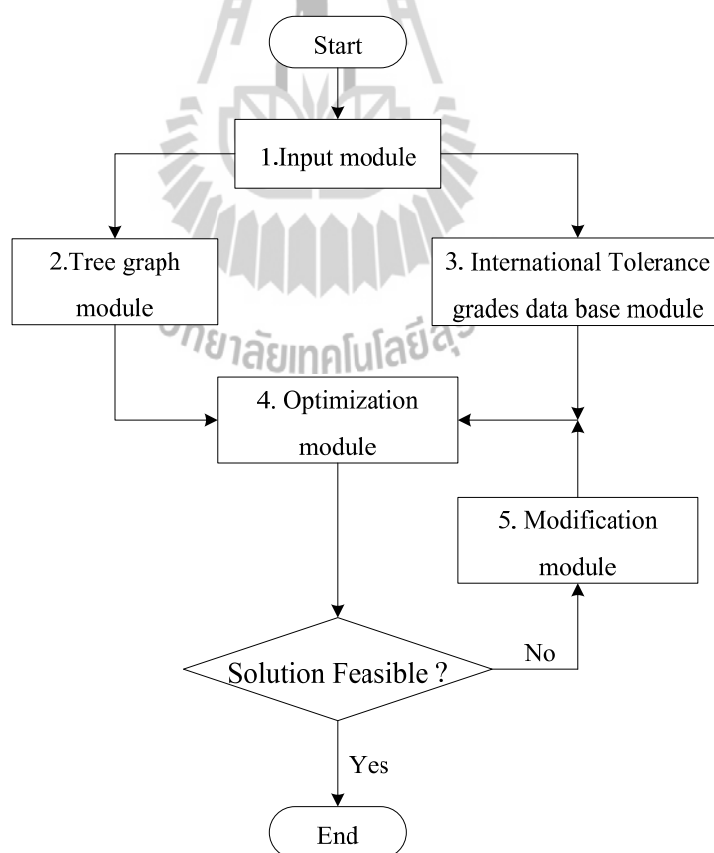


รูปที่ 3.2 ส่วนประกอบของหน้าจอเริ่มต้นหรือโมดูลหลัก



รูปที่ 3.3 ส่วนประกอบของแถบเมนู

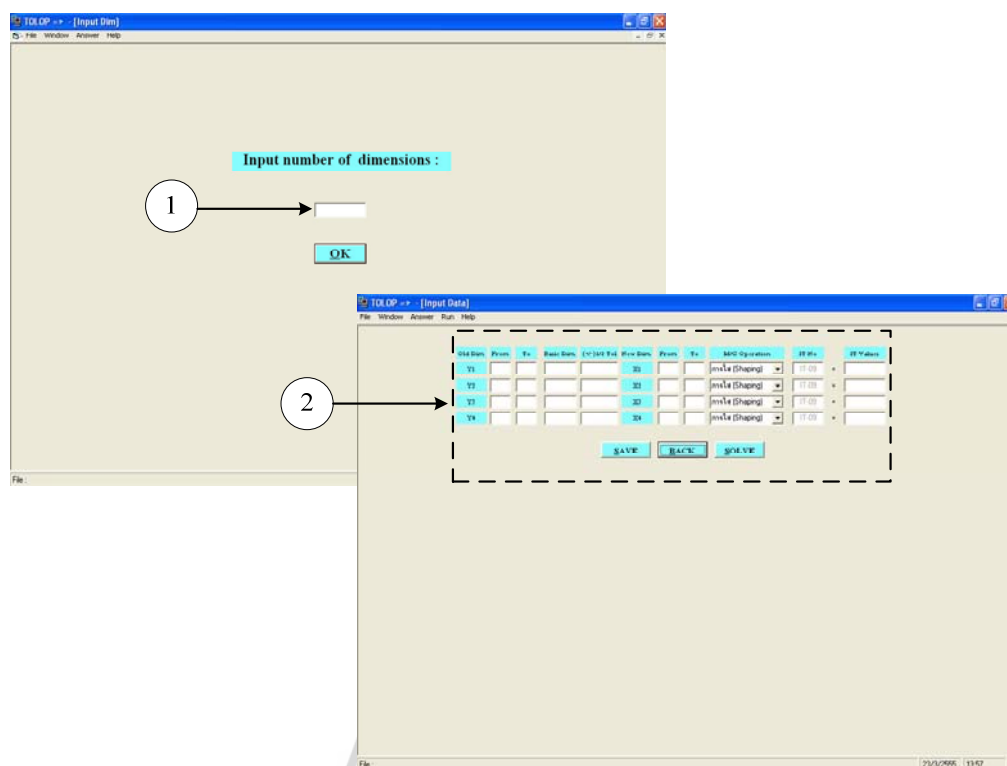
หลังจากเข้าสู่โปรแกรมแล้วการแก้ปัญหาแต่ละครั้งจะเกิดขึ้นในโมดูลหลักซึ่งครอบคลุมการทำหน้าที่ของโมดูลอื่น ๆ โดยมีขั้นตอนทั่วไปแสดงได้ดังแผนภูมิรูปที่ 3.4 เริ่มจากการรับข้อมูลจากผู้ใช้มาสร้างกราฟต้นไม้เพื่อแสดงความสัมพันธ์ระหว่างมิติชุดเก่าและมิติชุดใหม่ และสร้างตารางค่าความคลาดเคลื่อนมาตรฐาน จากนั้นนำความสัมพันธ์ที่ได้จากกราฟต้นไม้ไปสร้างเป็นสมการเชิงเส้นตรงและดึงข้อมูลค่าความคลาดเคลื่อนมาตรฐานมาจากตารางข้อมูล โดยหาคำตอบของสมการได้จากโมดูลการคำนวณหาค่าที่เหมาะสม (optimization module) การคำนวณในรอบแรกจะใช้รูปแบบของเกรด IT No จำนวน 3,000 รูปแบบ เมื่อโปรแกรมทำการคำนวณครบ 3,000 รูปแบบแล้ว จะทำการตรวจสอบว่ามีคำตอบหรือไม่ ถ้ายังไม่มีคำตอบจะเรียกใช้โมดูลปรับปรุงความคลาดเคลื่อน (modification module) เพื่อปรับปรุงเงื่อนไขของค่า IT No โดยอัตโนมัติ จากนั้นโปรแกรมจะกลับไปทำการคำนวณหาคำตอบในโมดูลการคำนวณหาค่าที่เหมาะสมใหม่นจนกว่าจะได้คำตอบที่เหมาะสมต่อไป



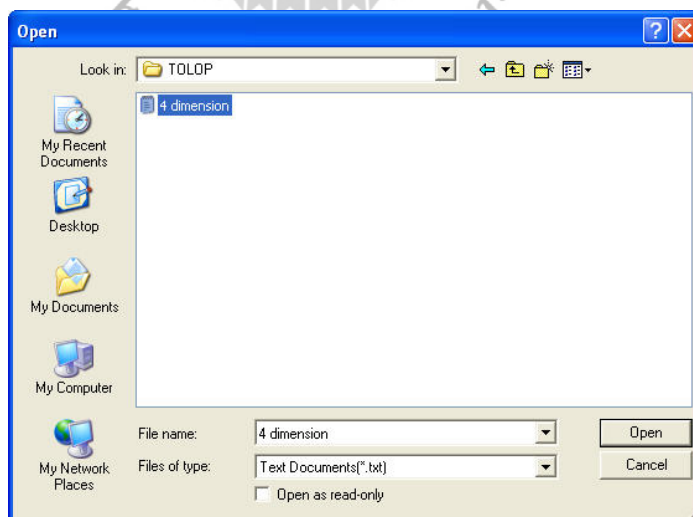
รูปที่ 3.4 ขั้นตอนการทำงานภายในโมดูลหลัก

3.1.2. โมดูลป้อนข้อมูล (Input module)

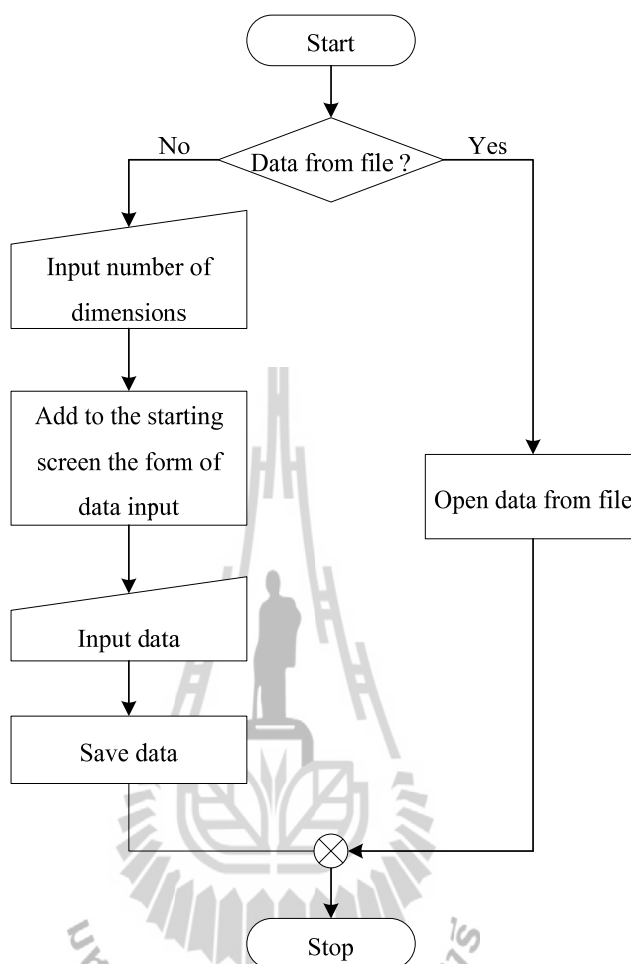
โมดูลป้อนข้อมูล มีหน้าที่รับข้อมูลต่าง ๆ ที่ผู้ใช้ป้อนให้กับโปรแกรมและเก็บบันทึกไว้ในฐานข้อมูลเพื่อใช้ประมวลผล เมื่อผู้ใช้กดปุ่ม Run จากหน้าจอของโมดูลหลัก หรือ เรียกใช้คำสั่ง New ในแถบเมนู จะปรากฏหน้าจอของโมดูลป้อนข้อมูลขึ้นมาดังรูปที่ 3.5 ซึ่งหน้าจอจะประกอบด้วย ช่องว่างหมายเลข 1 สำหรับให้ผู้ใช้ป้อนจำนวนมิติที่ต้องการลงไป หลังจากนั้นโมดูลป้อนข้อมูลจะสร้างรูปแบบของการป้อนข้อมูลตามจำนวนมิติที่กำหนดดังแสดงในหมายเลข 2 สำหรับมิติจำนวน 4 มิติ ข้อมูลที่ต้องทำการป้อนให้กับโปรแกรม ได้แก่ ตัวอักษรที่ใช้แทนผิวชิ้นงานของมิติที่เกิดจากการบอกขนาดในแบบและที่เกิดจากการกระบวนการตัดโลหะ ขนาดมูลฐานของมิติเก่า ค่าความคลาดเคลื่อนของมิติเก่าและสำหรับการกำหนดค่าความคลาดเคลื่อนมาตรฐานให้กับมิติใหม่นั้นจะทำได้โดยการเลือกค่าความคลาดเคลื่อนจากกระบวนการ (M/C Operation) ที่จะนำไปใช้ในการตัดมิตินั้น ๆ ของชิ้นงาน เป็นต้น เมื่อผู้ใช้ทำการป้อนข้อมูลจนครบแล้วทำการบันทึกข้อมูลโดยกดปุ่ม Save โปรแกรมจะทำการเก็บข้อมูลเป็นไฟล์นามสกุล .txt ในกรณีที่มีข้อมูลเก่าอยู่ในไฟล์แล้ว และต้องการนำมาประมวลผลก็สามารถเลือกคำสั่งได้จากแถบเมนู File เลือกใช้คำสั่ง Open จะปรากฏหน้าต่างดังรูปที่ 3.6 เลือกไฟล์ที่ต้องการแล้วกดปุ่ม Open ข้อมูลจะถูกนำมาใส่ไว้ที่หน้าจอป้อนข้อมูล กล่าวโดยสรุปการทำงานของโมดูลป้อนข้อมูลสามารถแสดงเป็นขั้นตอนได้ดังรูปที่ 3.7



รูปที่ 3.5 แสดงหน้าจอป้อนข้อมูล



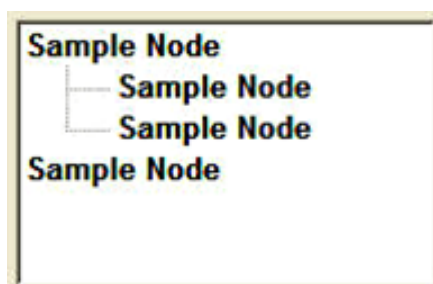
รูปที่ 3.6 แสดงหน้าต่าง Open



รูปที่ 3.7 แสดงขั้นตอนการทำงานของโมดูลป้อนข้อมูล

3.1.3 โมดูลกราฟต้นไม้ (tree graph module)

เป็นส่วนของการสร้างกราฟต้นไม้ของมิติชุดใหม่ โดยอาศัยเครื่องมือของไมโครซอฟท์พาวเวอร์เบสิก 6 ที่เรียกว่า คอนโทรลทรีวิว (control tree view) มีหน้าที่ในการแสดงผลข้อมูลออกมาในลักษณะของโครงสร้างต้นไม้ (รูปที่ 3.8) และใช้กราฟที่สร้างขึ้นนี้เพื่อหาเส้นทางที่แสดงถึงความสัมพันธ์ระหว่างมิติชุดเก่ากับมิติชุดใหม่ จากนั้นนำเส้นทางที่ได้ไปใช้ในการสร้างสมการเชิงเส้นตรงในโมดูลการคำนวณหาค่าที่เหมาะสมต่อไป



รูปที่ 3.8 คอนโทรลทรีวิว

3.1.4 โมดูลฐานข้อมูลค่าความคลาดเคลื่อนมาตรฐาน (International tolerance grades data base module)

โมดูลนี้แบ่งการทำงานเป็น 2 ส่วนคือ ส่วนแรกจะเป็นการสร้างตารางฐานข้อมูลของค่าความคลาดเคลื่อนมาตรฐาน ISO 286-1 (IT No) โดยใช้คอนโทรล MSHFlexGrid ซึ่งเป็นเครื่องมือของไมโครซอฟท์วิซวลเบสิก 6 มีหน้าที่ในการแสดงผลข้อมูลให้อยู่ในรูปแบบของตาราง ซึ่งการแสดงผลค่าความคลาดเคลื่อนให้อยู่ในรูปแบบของตารางทำให้สะดวกต่อการดึงข้อมูลไปใช้ในการคำนวณ สำหรับส่วนที่สองจะเป็นการสร้างรูปแบบของเกรด IT No

3.1.5 โมดูลการคำนวณหาค่าที่เหมาะสม (optimization module)

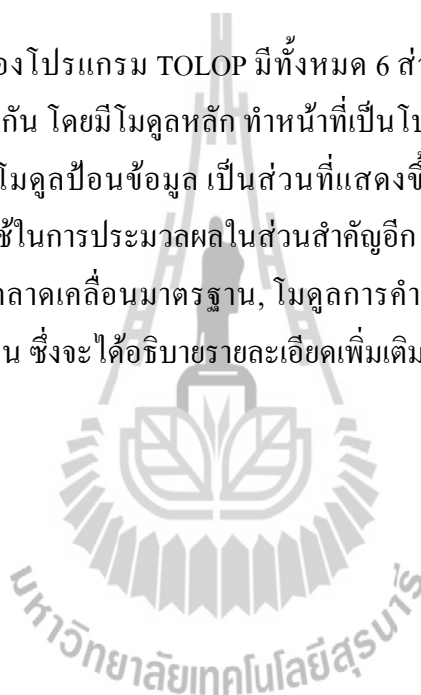
เป็นโมดูลการประมวลผลเพื่อหาคำตอบที่เหมาะสมด้วยวิธีการโปรแกรมเชิงเส้นตรง มีหน้าที่สร้างสมการเชิงเส้นตรงโดยอาศัยความสัมพันธ์จากโมดูลกราฟดิ้งไม้และดึงข้อมูลค่า IT No จากโมดูลฐานข้อมูลของค่าความคลาดเคลื่อนมาตรฐาน มาสร้างเป็นสมการเริ่มต้น โดยโปรแกรมจะใช้คอนโทรล MSHFlexGrid ในการดึงข้อมูลต่าง ๆ ให้อยู่ในรูปแบบของตารางเพื่อให้ง่ายต่อการคำนวณโดยใช้วิธีตารางซิมเพล็กซ์ เมื่อทำการคำนวณในรอบแรกเสร็จ โปรแกรมจะบันทึกผลลัพธ์ที่ได้ โดยจะบันทึกเฉพาะรอบที่สามารถหาคำตอบที่เหมาะสมได้เท่านั้น จากนั้นทำการสับเปลี่ยนรูปแบบของเกรด IT No ในรอบถัดไปจนครบ 3,000 รูปแบบ เมื่อโปรแกรมทำการคำนวณจนครบแล้ว ตรวจสอบว่ามีคำตอบหรือไม่ ถ้ามีคำตอบให้นำคำตอบที่ได้ทั้งหมดมาเปรียบเทียบว่าคำตอบของเกรด IT No รูปแบบใดมีค่าผลรวมของค่าความคลาดเคลื่อนสูงที่สุดโดยจะเลือกคำตอบนั้นเป็นคำตอบที่ดีที่สุด ถ้ายังไม่มีคำตอบโปรแกรมจะเรียกใช้โมดูลปรับปรุงความคลาดเคลื่อนเพื่อทำการสับเปลี่ยนรูปแบบของเกรด IT No ต่อไป

3.1.6 โมดูลปรับปรุงความคลาดเคลื่อน (modification module)

เป็นโมดูลการปรับเปลี่ยนค่าความคลาดเคลื่อนแบบอัตโนมัติ โดยปรับเปลี่ยนรูปแบบของเกรด IT No หลังจากที่มีโมดูลการคำนวณหาค่าที่เหมาะสมทำการปรับเปลี่ยนรูปแบบของเกรด IT No มาแล้ว 3000 รูปแบบโดยจะทำหน้าที่ส่งข้อมูลรูปแบบของเกรด IT No กลับไปยังโมดูลการคำนวณหาค่าที่เหมาะสมเพื่อทำการคำนวณในรอบถัดไป

3.2 สรุป

องค์ประกอบของโปรแกรม TOLOP มีทั้งหมด 6 ส่วน ในแต่ละส่วนมีหน้าที่และการทำงานที่มีความเชื่อมโยงกัน โดยมีโมดูลหลัก ทำหน้าที่เป็นโปรแกรมหลักสำหรับเรียกใช้งานในทุกส่วนของโปรแกรม โมดูลป้อนข้อมูล เป็นส่วนที่แสดงขึ้นมาเพื่อให้ผู้ใช้ป้อนข้อมูลให้กับโปรแกรมสำหรับนำไปใช้ในการประมวลผลในส่วนสำคัญอีก 4 ส่วนคือ โมดูลกราฟต้นไม้, โมดูลฐานข้อมูลของค่าความคลาดเคลื่อนมาตรฐาน, โมดูลการคำนวณหาค่าที่เหมาะสม และ โมดูลปรับปรุงความคลาดเคลื่อน ซึ่งจะได้อธิบายรายละเอียดเพิ่มเติมของขั้นตอนการทำงานทั้ง 4 ส่วนนี้ในบทต่อไป



บทที่ 5

สรุปผลการวิจัย

5.1 สรุป

งานวิจัยนี้สามารถบรรลุวัตถุประสงค์ที่ตั้งไว้คือ การพัฒนาโปรแกรมเพื่อใช้สำหรับแบ่งค่าความคลาดเคลื่อนให้เหมาะสมกับมิติชุดใหม่เมื่อระนาบอ้างอิงของการผลิตและการตรวจสอบแตกต่างไปจากระนาบอ้างอิงของการบอกขนาดในแบบวิศวกรรม

การใช้เทคนิคกราฟต้นไม้ทำให้ง่ายต่อการสร้างความสัมพันธ์ระหว่างค่าความคลาดเคลื่อนของมิติชุดใหม่กับมิติชุดเก่า โดยเทคนิคนี้ไม่จำเป็นต้องอาศัยกฎเกณฑ์มากนัก เพียงแค่สร้างกราฟของมิติชุดใหม่ ก็สามารถอ่านค่ามิติใช้งานต่าง ๆ ที่สัมพันธ์กับความต้องการใช้งานเดิมได้ทันที จากนั้นก็สามารถนำกราฟที่ได้ไปสร้างสมการความสัมพันธ์สำหรับการหาคำตอบที่เหมาะสมด้วยเทคนิคการโปรแกรมเชิงเส้นตรง เทคนิคดังกล่าวช่วยทำให้เกิดความรวดเร็วในการกระจายความคลาดเคลื่อน โดยเฉพาะอย่างยิ่งในกรณีที่ชิ้นงานที่มีหลายมิติและมีกระบวนการขึ้นรูปที่ซับซ้อน

โปรแกรมที่พัฒนาขึ้นไม่เพียงหาผลลัพธ์ที่เหมาะสมที่สุดให้กับปัญหาการแบ่งค่าความคลาดเคลื่อนได้เท่านั้น แต่ยังสามารถหาผลลัพธ์ที่เหมาะสมอื่น ๆ ที่มีค่าความคลาดเคลื่อนอยู่ในเกณฑ์ที่กำหนดไว้พร้อมกันด้วย ซึ่งทำให้ผู้ใช้สามารถเลือกใช้ได้มากกว่า 1 ค่า โดยที่ไม่ต้องสร้างสมการเชิงเส้นตรงด้วยมือและป้อนค่าให้กับโปรแกรมใหม่ทุกครั้งเมื่อมีการปรับเปลี่ยนค่า IT No ทำให้ช่วยลดขั้นตอนและเวลาในการทำงานลงได้มาก ตลอดจนผู้ใช้ไม่จำเป็นต้องมีความรู้และความชำนาญในทางปฏิบัติมากนัก ก็สามารถกระจายความคลาดเคลื่อนให้กับมิติในการผลิตหรือในการวัดขนาดใหม่เพื่อคงไว้ซึ่งความต้องการใช้งานของชิ้นงานที่ผู้ออกแบบกำหนดไว้

โปรแกรมนี้อย่างมีส่วนได้ตอบกับผู้ใช้เข้าใจง่าย อีกทั้งยังแสดงกราฟและข้อมูลที่เป็นประโยชน์อย่างครบถ้วน เหมาะสำหรับนักศึกษาหรือผู้ที่สนใจสามารถนำรหัสต้นฉบับ (source code) ไปพัฒนาต่อให้มีความยืดหยุ่นและครอบคลุมขอบเขตของงานให้มากขึ้นกว่าเดิมได้

5.2 ข้อจำกัดของโปรแกรม TOLOP

เนื่องจากโปรแกรมที่พัฒนาขึ้นนี้ใช้สำหรับแบ่งค่าความคลาดเคลื่อนให้กับมิติเชิงเส้นตรงธรรมดาเท่านั้น ซึ่งไม่สามารถนำไปใช้ในการวิเคราะห์ค่าความคลาดเคลื่อนเชิงเรขาคณิตได้ นอกจากนี้ยังมีข้อจำกัดอีกประการหนึ่ง คือ ข้อจำกัดในเรื่องของความเร็วในการประมวลผลเมื่อชิ้นงานรูปร่างซับซ้อนและมีมิติจำนวนมาก

5.3 ข้อเสนอแนะในการพัฒนาต่อไป

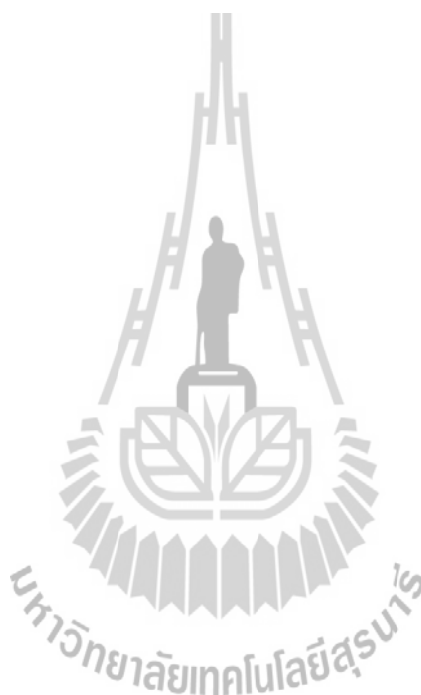
การพัฒนาโปรแกรมถือได้ว่าเป็นไปตามวัตถุประสงค์ แต่ก็ยังมีหลายส่วนที่สามารถนำไปพัฒนาให้ดีขึ้นได้ เช่น ในกรณีที่ชิ้นงานมีมิติจำนวนมากโปรแกรมยังต้องใช้เวลานานในการประมวลผล ดังนั้นจึงควรปรับปรุงให้โปรแกรมมีความเร็วในการประมวลผลมากขึ้น ในส่วนของ การแสดงผลของกราฟควรปรับปรุงให้อยู่ในรูปแบบของกราฟฟิกที่มีลักษณะใกล้เคียงกับกราฟ 3D มากขึ้น เพื่อให้ผู้ใช้สามารถเข้าใจได้ง่ายขึ้นและการแสดงผลลัพธ์ควรมีภาพกราฟฟิกของ ชิ้นงานพร้อมขนาดเพื่อให้เห็นรายละเอียดได้ชัดเจนยิ่งขึ้น ดังนั้นโปรแกรมที่พัฒนาขึ้นนี้จึงเหมาะที่จะเป็นต้นแบบเพื่อให้สามารถพัฒนาไปสู่การนำไปใช้ในอุตสาหกรรมจริงต่อไป

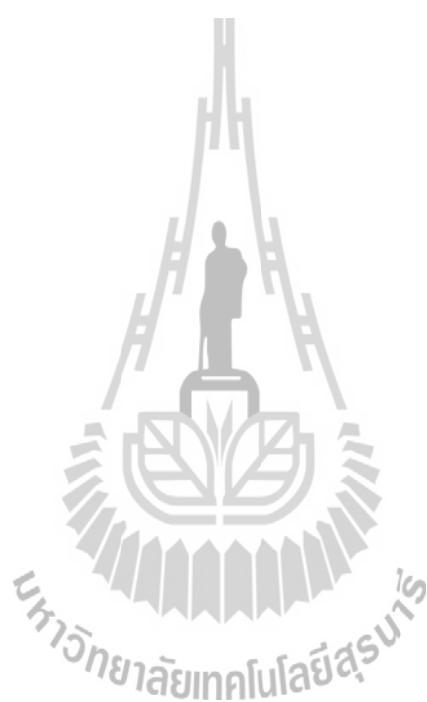


รายการอ้างอิง

- Ahaluwalia, R. S. and Karolin, A. V. (1984). CATC A computer-aided tolerance control system. **Journal of Manufacturing Systems**. 3 (2) : 153-160.
- Drake, P. Jr., (1999). **Dimensioning and tolerancing handbook**. McGraw-Hill.
- Fainguelernt, D., Wiell, R., and Bourdet, P. (1986). Computer aided tolerancing and dimensioning in Process Planning, **Annals of the CIRP**. 35 (1): 381-386
- Farmer, L. E. and Harris, A. G. (1984). Change of datum of the dimensions on engineering Design drawings. **International Journal of Machine Tool Design and Research**. 24 (4): 267-275.
- Gladman, C A. (1966). **Manual for geometric analysis of engineering design**. Australian Trade Publications. Sydney.
- Irani, S. A., Mittal, R. O., and Lehtihet, E. A. (1989) Tolerance chart optimization. **International Journal of Production Research**. 27 (9) : 1531-1552.
- Ji, P., (1993). A tree approach for Tolerance charting **International Journal of Production Research**. 31 (5) : 1023-1033.
- Liggett, J. V., (1993). **Dimensional variations management handbook: A guide for quality design and manufacturing engineers**. Prentice-Hall.
- Ngoi, B. K. A. (1992) Applying linear programming to tolerance chart balancing. **International Journal of Advanced Manufacturing Technology**. 7: 187-192.
- Ngoi, B. K. A., and Cheong, K. C. (1998). The apparent path tracing approach to tolerance charting. **International Journal of Advanced Manufacturing Technology**. 14:580-587
- Sermutsi-anuwat, Y., Whybrew, K., McCallion, H. (1995). CAPPFD – A tolerance-based feature sequencing CAPP system. **Journal of Systems Engineering**. 5 (1): 2-15.
- Sermutsi-anuwat, Y. (2009). Milling fixture design: a tolerance analysis approach. **International Journal of Mechanical Engineering Education**. 37 (2): 111-117.

- Thimm, G., Britton, G. A., and Fok, S. C. (2004). A graph theoretic approach linking Design dimensioning and process planning. **International Journal of Advanced Manufacturing Technology**. 24: 261-271.
- Whybrew, K., Britton, G. A., Robinson, D. F., and Sermsuti-anuwat, Y. (1990). A graph-theoretic approach to tolerance charting. **International Journal of Advanced Manufacturing Technology**. 5: 175-183.
- Oyvind, Bjorke. (1978) **Computer-aided tolerancing**. Tapir Publishers.





ภาคผนวก ก

ตารางเกณฑ์ความคลาดเคลื่อนมาตรฐาน (ISO 286-2)



ตารางที่ ก1 แสดงค่าของเกณฑ์ความคลาดเคลื่อนมาตรฐานสำหรับขนาดรูตั้งแต่ 0-3150 mm¹⁾ (ISO 286-2)

Basic Size		Standard tolerance grades																		
		mm		IT1 ²⁾	IT2 ²⁾	IT3 ²⁾	IT4 ²⁾	IT5 ²⁾	IT6	IT7	IT8	IT9	IT10	IT11	IT12	IT13	IT14 ³⁾	IT15 ³⁾	IT16 ³⁾	IT17 ³⁾
Above	Up to and including	Tolerances																		
		μm												mm						
-	3 ³⁾	0.8	1.2	2	3	4	6	10	14	25	40	60	0.1	0.1	0.3	0.4	0.6	1	1.4	1.4
3	6	1	1.5	2.5	4	5	8	12	18	30	48	75	0.1	0.2	0.3	0.5	0.75	1.2	1.8	1.8
6	10	1	1.5	2.5	4	6	9	15	22	36	58	90	0.2	0.2	0.4	0.6	0.9	1.5	2.2	2.2
10	18	1.2	2	3	5	8	11	18	27	43	70	110	0.2	0.3	0.4	0.7	1.1	1.8	2.7	2.7
18	30	1.5	2.5	4	6	9	13	21	33	52	84	130	0.2	0.3	0.5	0.8	1.3	2.1	3.3	3.3
30	50	1.5	2.5	4	7	11	16	25	39	62	100	160	0.3	0.4	0.6	1	1.6	2.5	3.9	3.9
50	80	2	3	5	8	13	19	30	46	74	120	190	0.3	0.5	0.7	1.2	1.9	3	4.6	4.6
80	120	2.5	4	6	10	15	22	35	54	87	140	220	0.4	0.5	0.9	1.4	2.2	3.5	5.4	5.4
120	180	3.5	5	8	12	18	25	40	63	100	160	250	0.4	0.6	1	1.6	2.5	4	6.3	6.3
180	250	4.5	7	10	14	20	29	46	72	115	185	290	0.5	0.7	1.2	1.9	2.9	4.6	7.2	7.2
250	315	6	8	12	16	23	32	52	81	130	210	320	0.5	0.8	1.3	2.1	3.2	5.2	8.1	8.1
315	400	7	9	13	18	25	36	57	89	140	230	360	0.6	0.9	1.4	2.3	3.6	5.7	8.9	8.9
400	500	8	10	15	20	27	40	63	97	155	250	400	0.6	1	1.6	2.5	4	6.3	9.7	9.7
500	630 ³⁾	9	11	16	22	32	44	70	110	175	280	440	0.7	1.1	1.8	2.8	4.4	7	11	11
630	800 ²⁾	10	13	18	25	36	50	80	125	200	320	500	0.8	1.3	2	3.2	5	8	12.5	12.5
800	100 ³⁾	11	15	21	28	40	56	90	140	230	360	560	0.9	1.4	2.3	3.6	5.6	9	14	14
100	1250 ³⁾	13	18	24	33	47	66	105	165	260	420	660	1.1	1.7	2.6	4.2	6.6	10.5	16.5	16.5
1250	1600 ²⁾	15	21	29	39	55	78	125	195	310	500	780	1.3	2	3.1	5	7.8	12.5	19.5	19.5
1600	2000 ³⁾	18	25	35	46	65	92	150	230	370	600	920	1.5	2.3	3.7	6	9.2	15	23	23
2000	2500 ²⁾	22	30	41	55	78	110	175	280	440	700	1100	1.8	2.8	4.4	7	11	17.5	28	28
2500	3150 ²⁾	26	36	50	68	96	135	210	330	540	860	1350	2.1	3.3	5.4	8.6	13.5	21	33	33

1) Value for standard tolerance grades IT01 and IT0 for basic sizes less than or equal to 500 mm are given in annex A table 5.

2) Value for standard tolerance grades IT1 and IT5 (incl.) for basic sizes over 500 mm are included for experimental use.

3) Standard tolerance grades IT14 and IT18 (incl.) shall not be used for basic sizes less than or equal to 1 mm.



ภาคผนวก ข

คู่มือการใช้งานโปรแกรม TOLOP

คู่มือการใช้งานโปรแกรม

TOLOP V.1



พัฒนาโดย

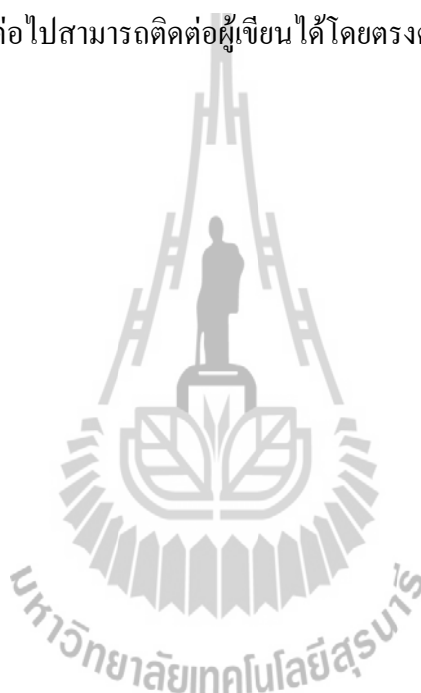
นางสาวอรรณ นงนุช

คำนำ

คู่มือการใช้งานโปรแกรมเล่มนี้จัดทำเพื่อเป็นคู่มือสำหรับช่วยผู้ใช้งานโปรแกรมในการเริ่มต้นใช้งานโปรแกรม TOLOP ประกอบด้วย 5 บท ได้แก่ บทที่ 1 แนะนำโปรแกรม บทที่ 2 การเริ่มต้นใช้งานโปรแกรม บทที่ 3 วิธีการวิเคราะห์ความคลาดเคลื่อนด้วยโปรแกรม TOLOP ซึ่งได้แสดงตัวอย่างการวิเคราะห์และออกแบบที่ละเอียดขั้นตอนอย่างละเอียด

โปรแกรม TOLOP ขณะนี้ คือ รุ่นที่ 1 ผู้เขียนหวังว่าคู่มือการใช้งานโปรแกรมฉบับนี้จะช่วยให้ผู้อ่านสามารถใช้งานโปรแกรม TOLOP ได้อย่างเต็มประสิทธิภาพ หากมีข้อเสนอแนะติชมเพื่อการพัฒนาโปรแกรมต่อไปสามารถติดต่อผู้เขียนได้โดยตรงตามที่แสดงไว้ในคู่มือฉบับนี้

อรรณณ นงนุช



บทที่ 1

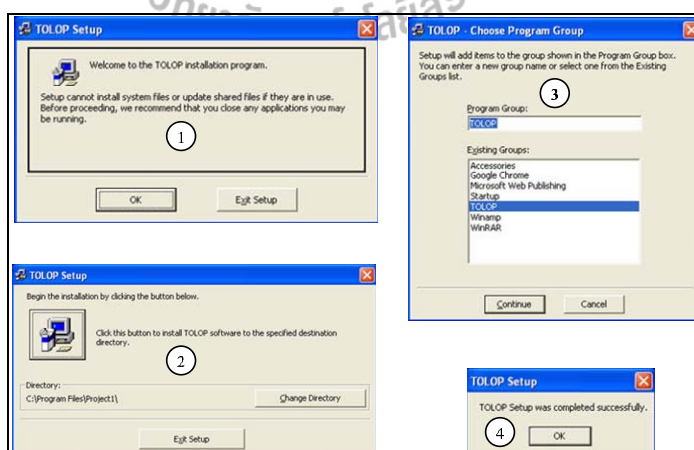
แนะนำโปรแกรม

1.1 ระบบปฏิบัติการที่ต้องการ

- 1) ระบบปฏิบัติการ Windows 98 ขึ้นไป (แนะนำ Windows XP)
- 2) เครื่องคอมพิวเตอร์ที่มี CPU ที่เป็น Pentium 100 MHz ขึ้นไป
- 3) หน่วยความจำ 32 MB ขึ้นไป
- 4) ไดรฟ์ซีดีรอม (CD-ROM) สำหรับติดตั้งโปรแกรม
- 5) การ์ดแสดงผลที่สามารถแสดงผลที่ความละเอียดตั้งแต่ 800x600 ขึ้นไป (แนะนำ 1024x768)
- 6) เม้าส์และคีย์บอร์ดที่สนับสนุนการทำงานของวินโดวส์

1.2 ขั้นตอนการติดตั้งโปรแกรม

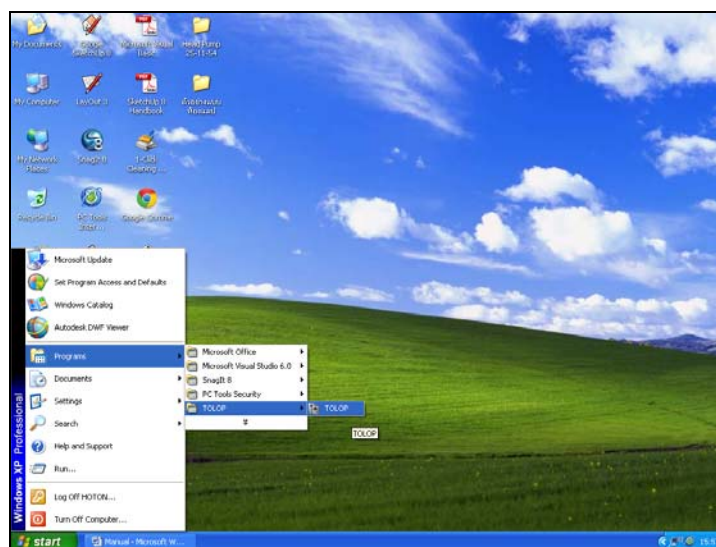
- 1) เปิดหน้าต่าง My Computer และเลือกไปยังโฟลเดอร์ที่เก็บไฟล์ไว้และดับเบิลคลิกที่ไฟล์ Setup จะแสดงหน้าต่างดังรูปที่ M1.1
- 2) ทำตามขั้นตอนที่แสดงในตัวติดตั้งโปรแกรมทีละขั้นตามลำดับ



รูปที่ M1.1 หน้าต่างการติดตั้งโปรแกรม

1.3 การเข้าสู่โปรแกรม

การเข้าสู่โปรแกรม TOLOP สามารถทำได้โดยไปที่ Window Start menu > Program > TOLOP ดังรูปที่ M1.2



รูปที่ M1.2



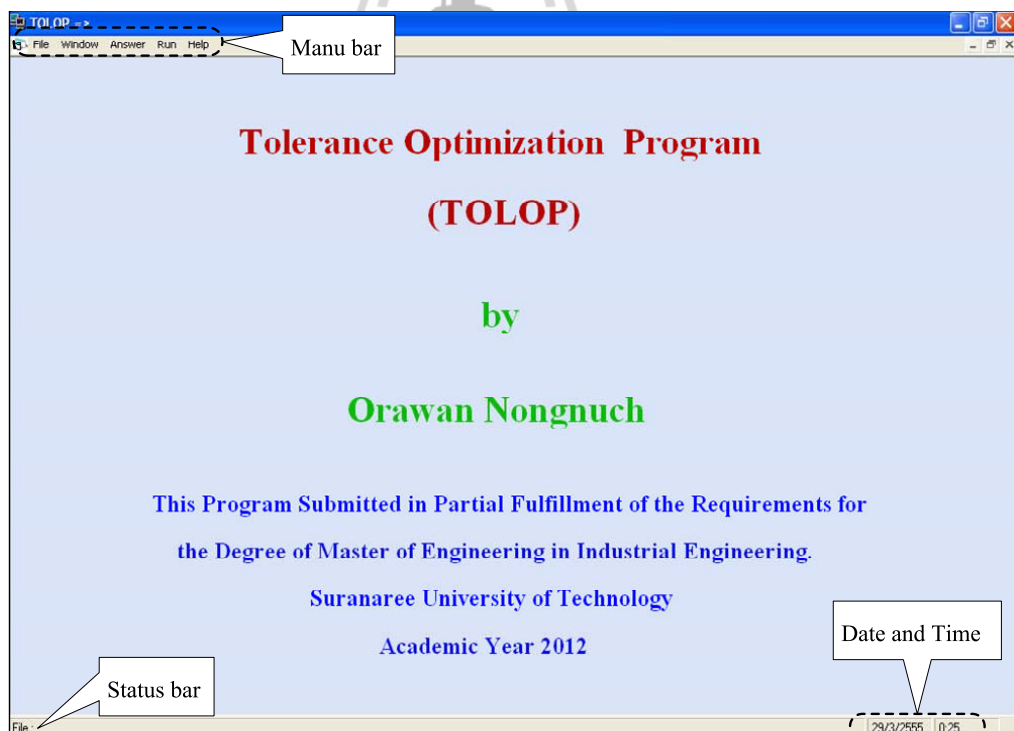
บทที่ 2

การเริ่มต้นใช้งานโปรแกรม

2.1 การเริ่มต้นใช้งานโปรแกรม TOLOP

หลังจากเข้าสู่โปรแกรม TOLOP ตามขั้นตอนในหัวข้อ 1.3 แล้ว โปรแกรมจะแสดงหน้าจอเริ่มต้นดังรูปที่ M2.1 เช่นเดียวกับโปรแกรมส่วนใหญ่ที่ทำงานภายใต้ระบบปฏิบัติการวินโดวส์ โดยมีส่วนประกอบของหน้าจอดังต่อไปนี้

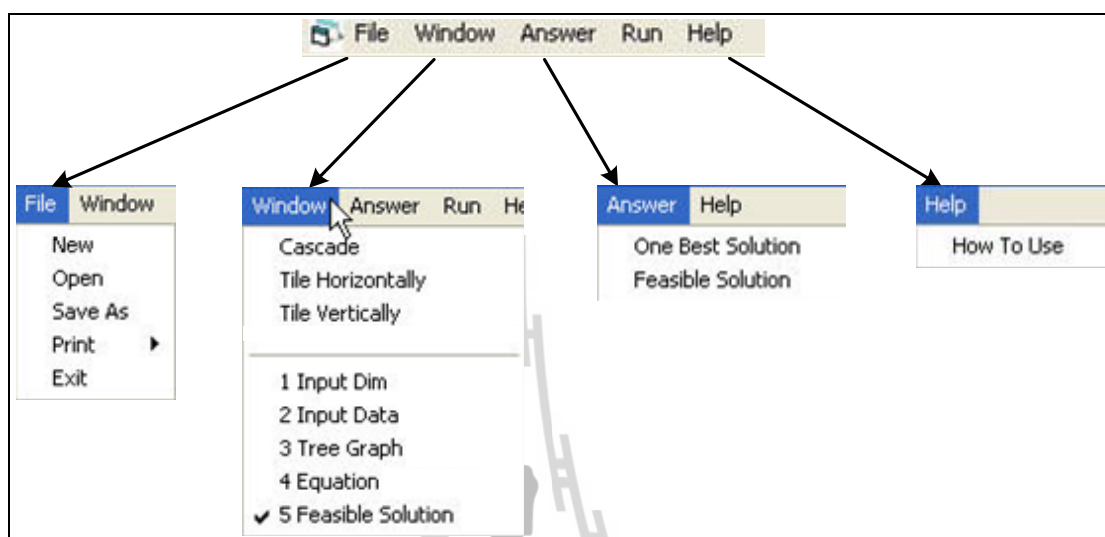
- 1) แถบเมนู (Menu bar) มีไว้ให้ผู้ใช้งานติดต่อและสั่งงานโปรแกรม
- 2) ชื่อโปรแกรมและชื่อผู้พัฒนาโปรแกรม
- 3) แถบสถานะ (Status bar) สำหรับแสดงสถานะของไฟล์ที่เก็บบันทึกข้อมูลพร้อมทั้งแสดงวันที่และเวลาขณะที่ใช้โปรแกรม



รูปที่ M2.1 ส่วนประกอบของหน้าจอเริ่มต้น

2.2 แถบเมนู (Menu bar)

แถบเมนูที่แสดงอยู่ด้านบนของหน้าจอเป็นคำสั่งที่มีไว้ให้ผู้ใช้งานติดต่อสั่งงานโปรแกรม ซึ่งมีส่วนประกอบดังรูปที่ M2.2



รูปที่ M2.2 ส่วนประกอบของแถบเมนู

คำอธิบายเพิ่มเติมสำหรับแถบเมนู

File>New

สร้างงานใหม่

File>Open

เปิดงานเก่า

File>Save As

บันทึกงานในชื่อใหม่

File>Print

พิมพ์รายงาน

File>Exit

ออกจากโปรแกรม

Window>Input Dim

หน้าต่างแสดงหน้าต่างป้อนข้อมูลจำนวนมิติ

Window>Input Data

หน้าต่างแสดงหน้าต่างป้อนข้อมูล

Window>Tree Graph Expand

หน้าต่างสำหรับแสดงการขยายขนาดของกราฟต้นไม้

Window>Equation

หน้าต่างแสดงสมการเริ่มต้น

Window> Feasible Solution

หน้าต่างแสดงคำตอบที่เหมาะสม

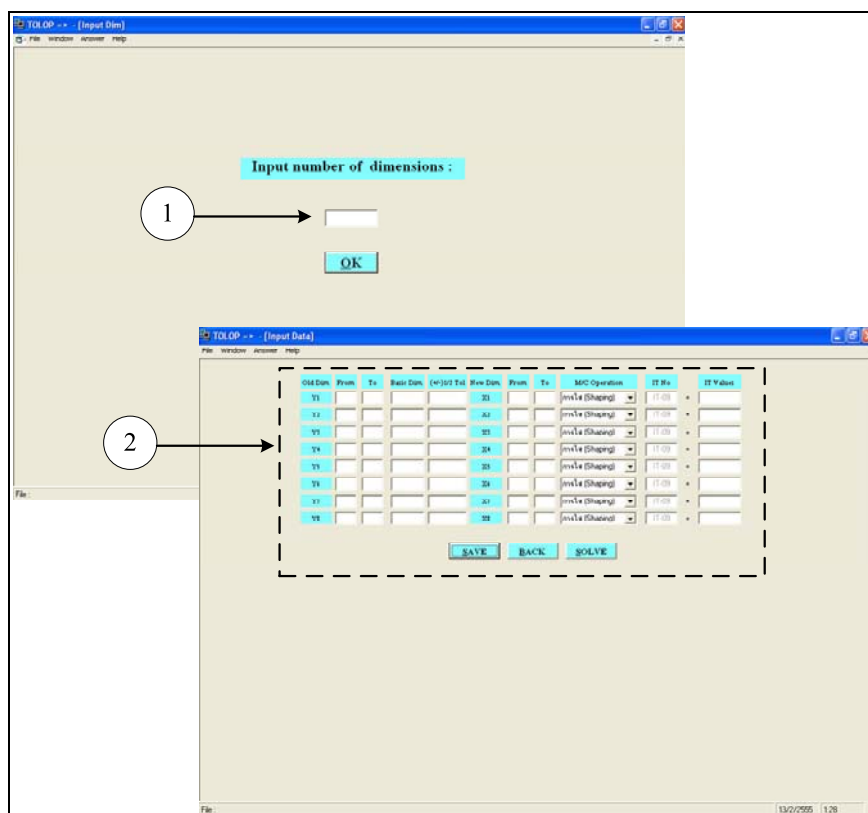
Answer>One Best Solution	แสดงคำตอบที่ดีที่สุด
Answer>Feasible Solution	แสดงคำตอบที่เหมาะสม
Run	เริ่มต้นการทำงานของโปรแกรม
Help>How To Use	วิธีใช้งานโปรแกรม

2.3 การป้อนข้อมูล (Input Data)

เมื่อผู้ใช้งานกดที่แถบเมนู Run (รูปที่ M2.3) โปรแกรมจะแสดงหน้าต่างสำหรับป้อนข้อมูล (รูปที่ M2.4) ในช่องหมายเลข 1 ให้ป้อนจำนวนมิติที่ต้องการ โดยจำนวนมิติที่ป้อนจะต้องมีจำนวนตั้งแต่ 3 มิติขึ้นไป เมื่อป้อนตัวเลขจำนวนมิติแล้วให้กดที่ปุ่ม OK หรือ กดที่ปุ่ม Enter บนคีย์บอร์ด จากนั้นโปรแกรมจะแสดงหน้าจอสำหรับป้อนรายละเอียดของข้อมูลในกรอบหมายเลข 2



รูปที่ M2.4 หน้าต่างสำหรับป้อนข้อมูล



รูปที่ M2.4 หน้าต่างสำหรับป้อนข้อมูล

การป้อนข้อมูลผู้ใช้งานจะต้องป้อนข้อมูลให้ครบทุกช่อง โดยรายละเอียดของข้อมูลที่ต้องป้อนให้กับโปรแกรมมีดังนี้ (รูปที่ M2.5)

- 1) ป้อนตัวอักษรที่ระบุผิวเริ่มต้นของมิติเก่าในช่องข้อความ From
- 2) ป้อนตัวอักษรที่ระบุผิวปลายของมิติเก่าในช่องข้อความ To
- 3) ป้อนขนาดมูลฐานของมิติเก่าในช่องข้อความ Basic Dim
- 4) ป้อนค่าความคลาดเคลื่อนของมิติเก่าลงในช่องข้อความ (+/-) 1/2 Tol
- 5) ป้อนตัวอักษรที่ระบุผิวเริ่มต้นของมิติใหม่ในช่องข้อความ From
- 6) ป้อนตัวอักษรที่ระบุผิวปลายของมิติใหม่ในช่องข้อความ To
- 7) ป้อนข้อมูลกระบวนการผลิตโดยคลิกเลือกที่ลูกศรในช่องข้อความ M/C Operation

TOLOP => [Input Data]

File Window Answer Run Help

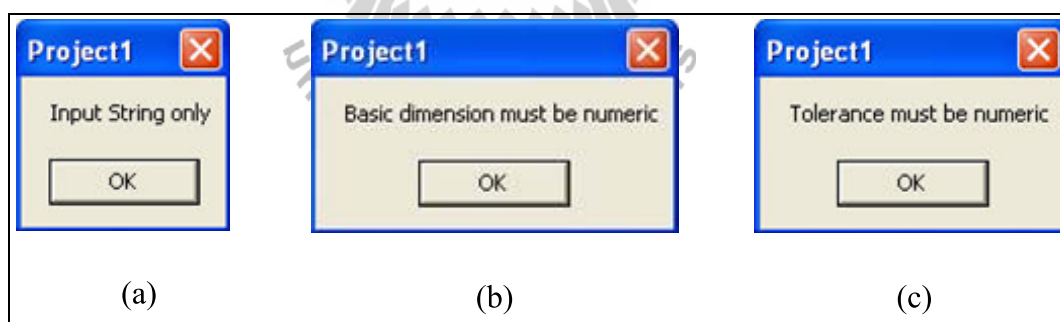
Old Dim	From	To	Basic Dim	(+/-)1/2 Tol	New Dim	From	To	M/C Operation	IT No	IT Values
Y1					X1			การใส่ (Shaping)		=

SAVE BACK SOLVE

1 2 3 4 5 6 7

รูปที่ M2.5 การป้อนข้อมูลให้กับโปรแกรม

การป้อนข้อมูลให้กับโปรแกรมจะต้องป้อนเฉพาะข้อมูลที่เป็นตัวอักษรให้กับช่องหมายเลข 1, 2, 5 และ 6 เท่านั้น ถ้าผู้ใช้ป้อนข้อมูลที่เป็นตัวเลข เครื่องหมาย หรือ สัญลักษณ์อื่นใด โปรแกรมจะแสดงกล่องข้อความดังรูปที่ M2.6 (a) การป้อนข้อมูลให้กับช่องหมายเลข 3 จะต้องป้อนเฉพาะข้อมูลที่เป็นตัวเลขเท่านั้น ถ้าผู้ใช้ป้อนข้อมูลที่เป็นตัวอักษร เครื่องหมาย หรือ สัญลักษณ์อื่นใด โปรแกรมจะแสดงกล่องข้อความดังรูปที่ M2.6 (b) และการป้อนข้อมูลให้กับช่องหมายเลข 4 จะต้องป้อนเฉพาะข้อมูลที่เป็นตัวเลขเท่านั้น ถ้าผู้ใช้ป้อนข้อมูลที่เป็นตัวอักษร เครื่องหมาย หรือ สัญลักษณ์อื่นใด โปรแกรมจะแสดงกล่องข้อความดังรูปที่ M2.6 (c)

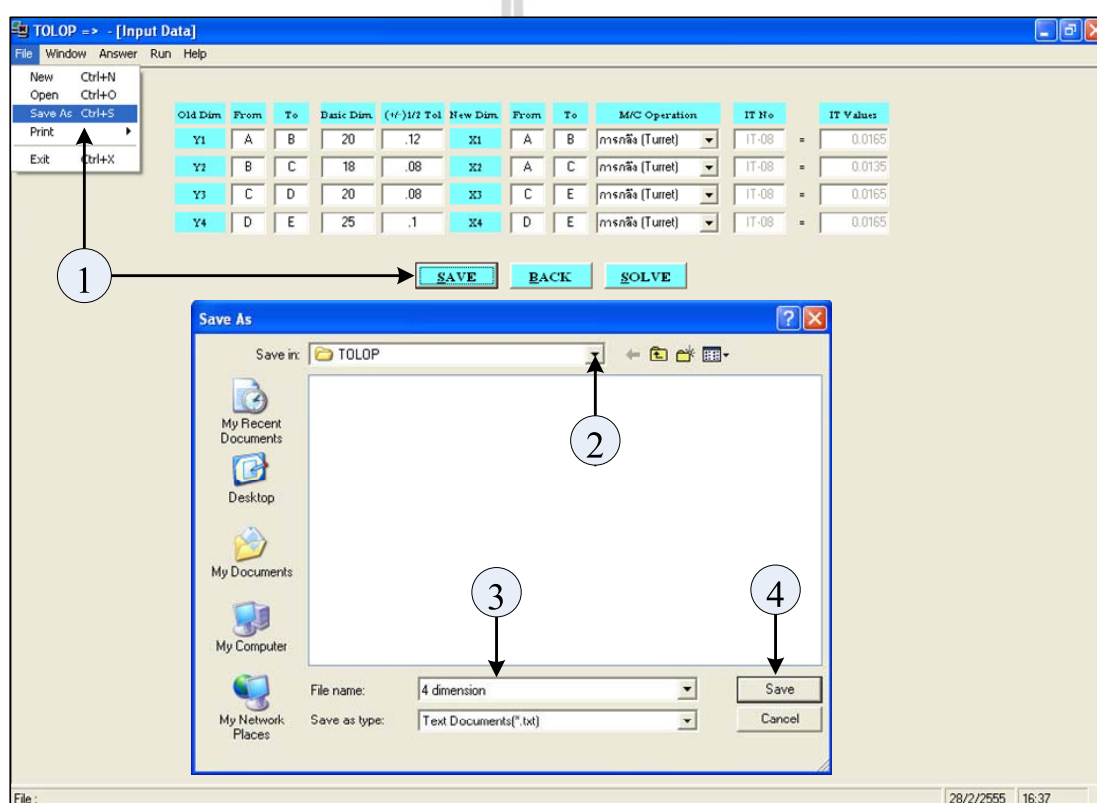


รูปที่ M2.6

2.4 การบันทึกข้อมูล (Save Data)

หลังจากที่ผู้ใช้ป้อนข้อมูลเสร็จเรียบร้อยแล้ว ผู้ใช้สามารถทำการการบันทึกข้อมูลไว้ในหน่วยความจำได้โดยมีขั้นตอนการบันทึกดังนี้

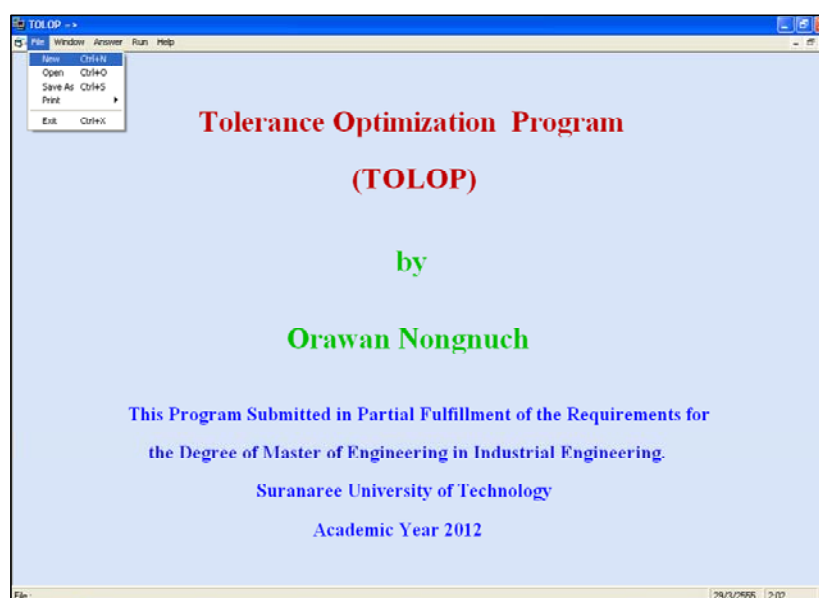
- 1) คลิกที่คำสั่ง File > SaveAs บนแถบเมนู หรือคลิกที่ปุ่ม SAVE จะมีหน้าต่าง Save As แสดงขึ้นมามีดังรูปที่ M2.7
- 2) เลือกโฟลเดอร์ที่ต้องการบันทึกข้อมูล
- 3) ป้อนชื่อไฟล์ที่ต้องการบันทึกข้อมูลลงในช่อง File name
- 4) คลิกที่ปุ่ม Save ข้อมูลจะถูกบันทึกเป็นไฟล์นามสกุล .txt เช่น 4 dimension.txt



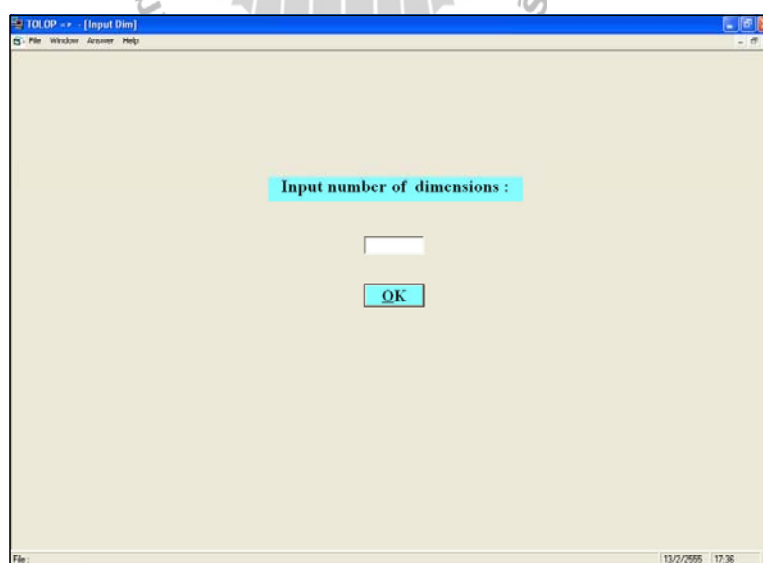
รูปที่ M2.7 หน้าต่างบันทึกข้อมูล

2.5 การเริ่มงานใหม่ (New)

เมื่อต้องการเริ่มแผ่นงานใหม่ให้ไปที่คำสั่งบนแถบเมนู คลิกที่ File > New รูปที่ M2.8 จากนั้นโปรแกรมจะแสดงหน้าจอให้ใส่จำนวนมิติใหม่ ดังรูปที่ M2.9



รูปที่ M2.8

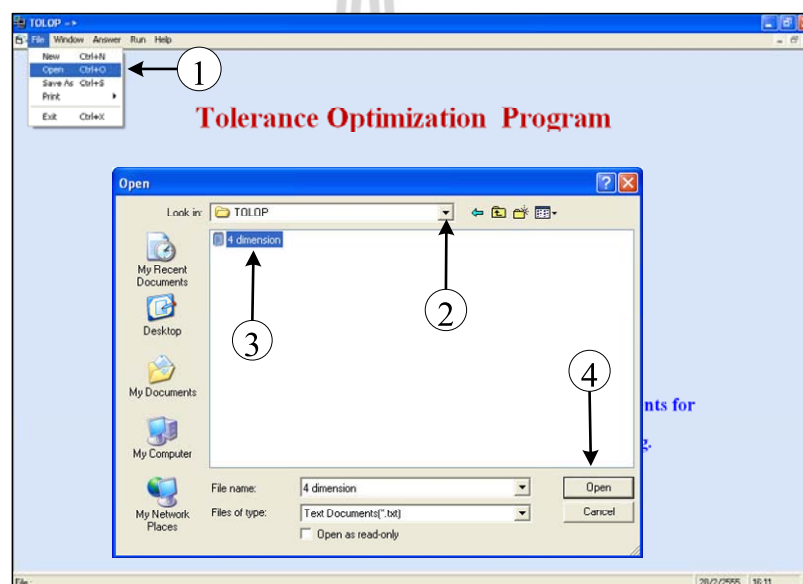


รูปที่ M2.9

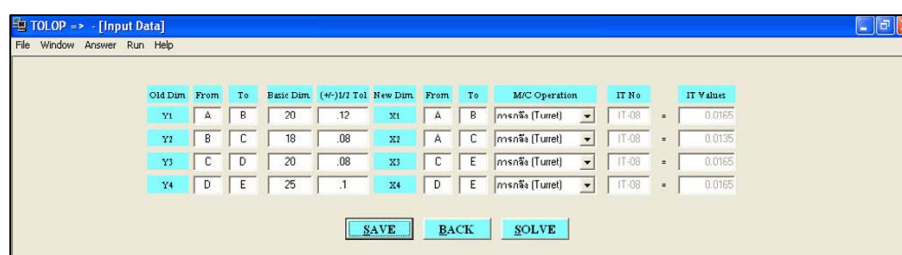
2.6 การเปิดข้อมูล (Open data)

การเปิดข้อมูล คือ การนำข้อมูลที่บันทึกไว้ในหน่วยความจำมาใส่ลงในช่องว่างของหน้าจอป้อนข้อมูลโดยมีขั้นตอนดังนี้

- 1) คลิกที่ File > Open บนแถบเมนูจะมีหน้าต่าง Open แสดงขึ้นมาดังรูปที่ M2.10
- 2) ไปยังโฟลเดอร์ที่เก็บไฟล์ข้อมูลไว้
- 3) เลือกไฟล์ข้อมูลที่ต้องการ
- 4) คลิกที่ปุ่ม Open ข้อมูลจะถูกป้อนลงในช่องว่าง (ข้อมูลที่สามารถเปิดได้ต้องมีนามสกุลเป็น .txt) จากนั้นข้อมูลก็จะถูกนำไปใส่ไว้ในช่องว่างของหน้าจอป้อนข้อมูล ดังรูปที่ M2.11



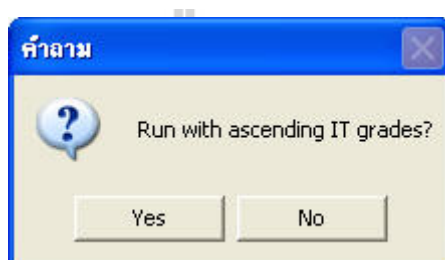
รูปที่ M2.10 หน้าต่างคำสั่งเปิดไฟล์ข้อมูล



รูปที่ M2.11

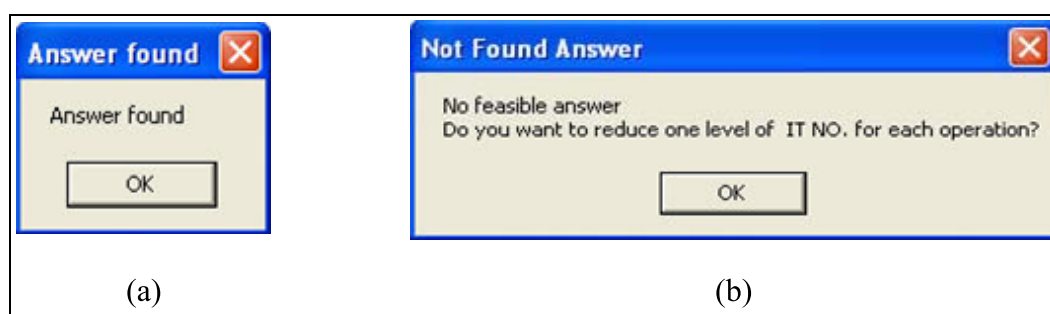
2.7 การหาผลลัพธ์

เมื่อใส่ข้อมูลและบันทึกข้อมูลเรียบร้อยแล้วขั้นตอนต่อไปจะเป็นการหาผลลัพธ์ โดยกดที่ปุ่ม SOLVE โปรแกรมจะแสดงกล่องข้อความว่าต้องการ RUN เกรด IT No จากน้อยไปมากหรือไม่” ดังรูปที่ M2.12 ถ้าต้องการให้โปรแกรมรันค่า IT No จากค่าน้อยไปค่ามากให้กดปุ่ม Yes โปรแกรมจะทำการประมวลผลโดยเริ่มคำนวณจากค่า IT No น้อยที่สุดก่อน แต่ถ้าต้องการให้โปรแกรมประมวลผลจาก IT No มากไปน้อย ให้กดปุ่ม No โปรแกรมก็จะทำการคำนวณจากค่า IT No มากที่สุดก่อน



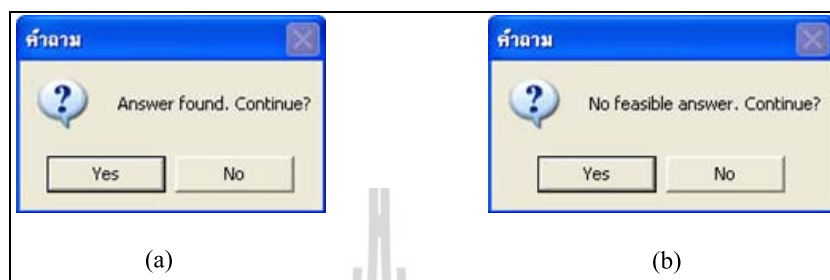
รูปที่ M2.12

ในกรณีที่จำนวนรูปแบบของ IT No ไม่ถึง 3000 รูปแบบ โปรแกรมจะทำการประมวลผลโดยสามารถหาคำตอบได้ในคราวเดียว ถ้าพบคำตอบโปรแกรมจะแสดงกล่องข้อความดังรูปที่ M 2.13 (a) แต่ถ้าไม่มีคำตอบ โปรแกรมจะแสดงกล่องข้อความดังรูปที่ 2.13 (b)



รูปที่ M2.13

สำหรับกรณีที่มีจำนวนรูปแบบมากกว่า 3000 รูปแบบ โปรแกรมจะทำการประมวลผลในรอบแรกโดยใช้ IT No จำนวน 3000 รูปแบบก่อน เมื่อทำการประมวลผลครบ 3000 รูปแบบแล้ว ถ้าพบคำตอบ โปรแกรมจะแสดงกล่องข้อความ ดังรูปที่ M2.14 (a) แต่ถ้าประมวลผลครบ 3000 รูปแบบแล้วยังไม่พบคำตอบ โปรแกรมจะแสดงกล่องข้อความ ดังรูปที่ M2.14 (b)

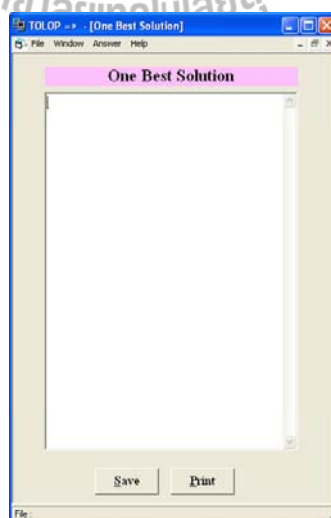


รูปที่ M2.14

2.7.1 หน้าจอแสดงผลลัพธ์

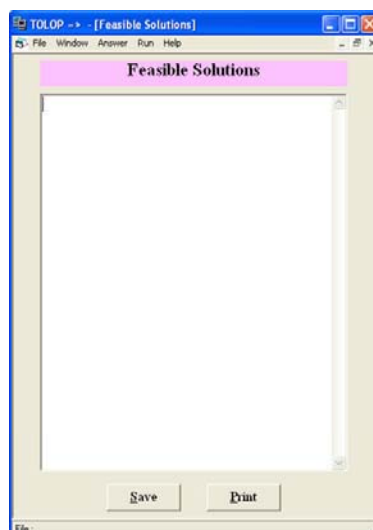
หน้าจอแสดงผลลัพธ์สามารถเลือกให้แสดงผลได้ 2 หน้าจอ ดังนี้

- 1) หน้าจอแสดงคำตอบที่ดีที่สุด ซึ่งเป็นคำตอบที่มีผลรวมของค่าความคลาดเคลื่อนมากที่สุดแสดงดังรูปที่ M2.15 สามารถบันทึกคำตอบโดยกดปุ่ม Save และพิมพ์คำตอบโดยกดปุ่ม Print



รูปที่ M2.15 หน้าจอแสดงคำตอบที่ดีที่สุด

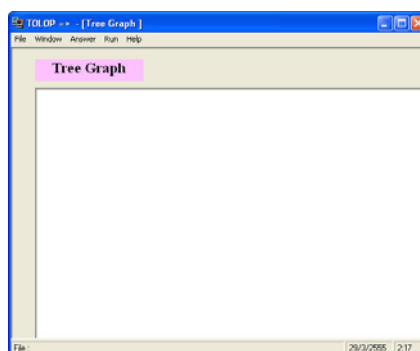
- 2) หน้าจอแสดงคำตอบที่เหมาะสม ซึ่งเป็นคำตอบที่สามารถนำไปเลือกใช้ได้แต่ไม่ใช่คำตอบที่มีผลรวมของค่าความคลาดเคลื่อนมากที่สุด แสดงดังรูปที่ M2.16 สามารถบันทึกคำตอบโดยกดปุ่ม Save และพิมพ์คำตอบโดยกดปุ่ม Print



รูปที่ M2.16 หน้าจอแสดงคำตอบที่เหมาะสม

2.7.2 หน้าจอแสดงกราฟต้นไม้

หน้าจอนี้จะแสดงกราฟต้นไม้ที่สร้างขึ้นจากความสัมพันธ์ของมิติใหม่โดยมีลักษณะดังรูปที่ M2.17

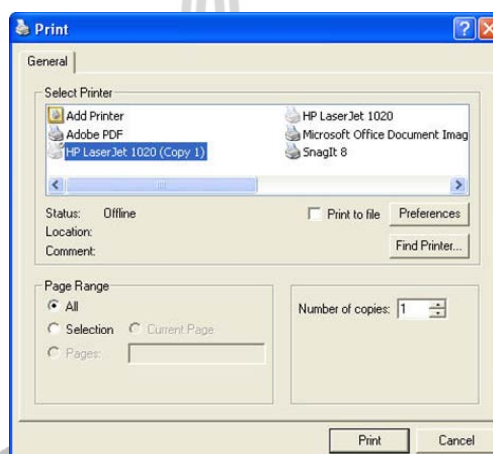


รูปที่ M2.17 หน้าจอแสดงกราฟต้นไม้

2.8 การพิมพ์รายงาน (Print)

การสั่งพิมพ์ข้อมูลจะต้องทำการบันทึกข้อมูลลงไฟล์ก่อนทุกครั้ง ไม่เช่นนั้นโปรแกรมจะไม่สามารถสั่งพิมพ์ข้อมูลได้ เมื่อเลือกคำสั่ง Print แล้ว โปรแกรมจะแสดงหน้าต่าง Print ดังรูปที่ M2.17 ซึ่งการสั่งพิมพ์ข้อมูลสามารถสั่งพิมพ์ได้จากคำสั่งที่อยู่ในแถบเมนู ต่าง ๆ ดังนี้

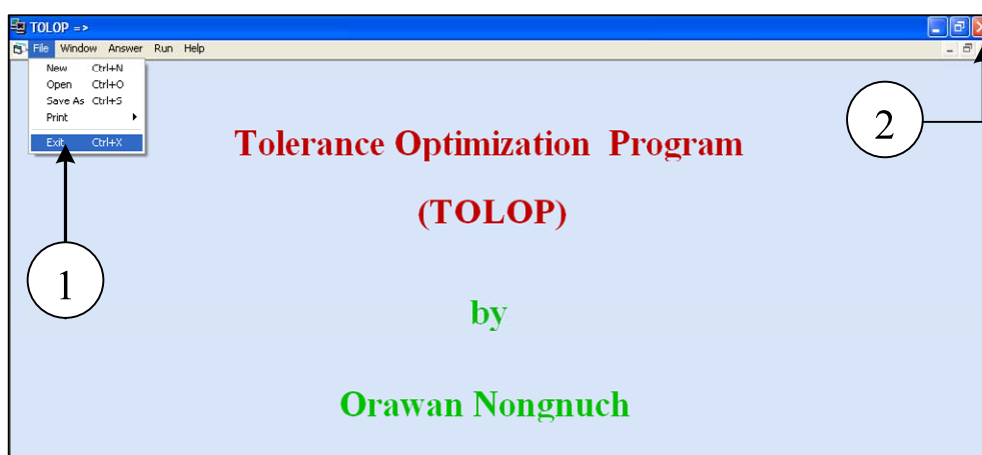
- 1) คลิกที่คำสั่ง File > Print > Form main เพื่อพิมพ์ข้อมูลของหน้าต่างหลัก
- 2) คลิกที่คำสั่ง File > Print > Equation เพื่อพิมพ์สมการ
- 3) คลิกที่คำสั่ง File > Print > One Best Solution เพื่อพิมพ์คำตอบที่ดีที่สุด
- 4) คลิกที่คำสั่ง File > Print > Feasible Solution เพื่อพิมพ์คำตอบที่เหมาะสมทั้งหมดที่สามารถนำมาเลือกใช้ได้



รูปที่ M2.17 หน้าต่างคำสั่งพิมพ์

2.9 การออกจากโปรแกรม (Exit program)

เมื่อต้องการออกจากโปรแกรมให้คลิกที่ Files > Exit หรือคลิกที่เครื่องหมาย × ที่ด้านขวาบนสุดของหน้าต่างหลัก ดังรูปที่ M2.18



รูปที่ M2.18



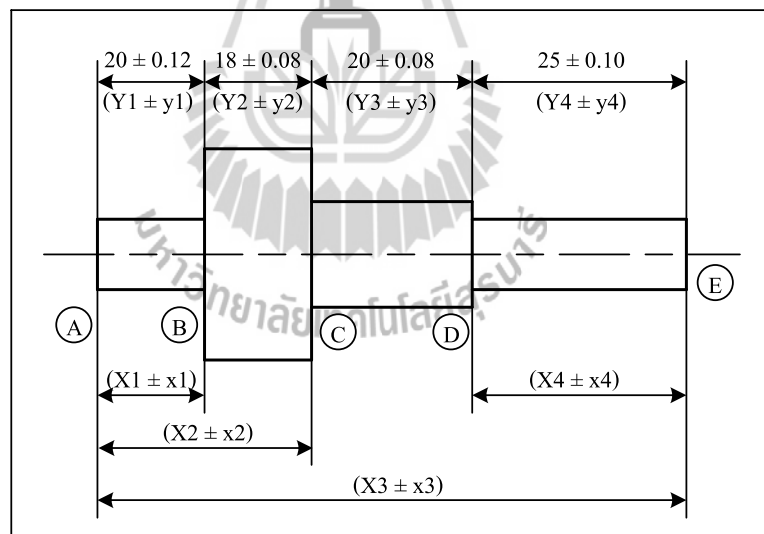
บทที่ 3

วิธีการวิเคราะห์ความคลาดเคลื่อนด้วยโปรแกรม TOLOP

เนื้อหาในบทนี้จะเป็นการนำตัวอย่างโจทย์มาสาธิตการใช้งานโปรแกรม TOLOP แบบเป็นขั้นตอนเพื่อให้เข้าใจวิธีการใช้งานของโปรแกรม TOLOP ยิ่งขึ้น

ตัวอย่างที่ 1

โจทย์ตัวอย่างนี้มีมิติของชิ้นงานจำนวน 4 มิติ (รูปที่ M3.1) ซึ่งระบุขนาดความคลาดเคลื่อนของมิติความยาวเป็นค่าเบี่ยงเบนเท่า ๆ กัน โดยกำหนดให้ $(Y1 \pm y1)$ ถึง $(Y4 \pm y4)$ เป็นมิติชุดเก่า และ $(X1 \pm x1)$ ถึง $(X4 \pm x4)$ เป็นมิติชุดใหม่ที่ต้องการทราบค่า การใช้โปรแกรมเพื่อหาค่าความคลาดเคลื่อนชุดใหม่ มีขั้นตอนดังนี้



รูปที่ M3.1 ตัวอย่างชิ้นงาน 4 มิติ

- 1) ป้อนจำนวนมิติเท่ากับ 4 ลงในช่อง Input Number of dimensions ดังรูปที่ M3.2 จากนั้นกดปุ่ม OK หรือกดปุ่ม Enter ที่คีย์บอร์ด
- 2) โปรแกรมจะแสดงหน้าจอสำหรับป้อนข้อมูลจำนวน 4 แถว จากนั้นให้ป้อนข้อมูลลงในช่องว่างให้ครบทุกช่อง ดังรูปที่ M3.3

- 3) กดปุ่ม SAVE เพื่อทำการบันทึกข้อมูล
- 4) กดปุ่ม SOLVE เพื่อหาผลลัพธ์
- 5) เมื่อโปรแกรมประมวลผลเสร็จแล้วจะแสดงผลลัพธ์ ดังรูปที่ M3.4

Input number of dimensions :

4

OK

รูปที่ M3.2

Old Dim	From	To	Basic Dim.	(+/-)1/2 Tol	New Dim.	From	To	M/C Operation	IT No	IT Values
Y1					X1			การไส (Shaping)	IT-09	=
Y2					X2			การไส (Shaping)	IT-09	=
Y3					X3			การไส (Shaping)	IT-09	=
Y4					X4			การไส (Shaping)	IT-09	=

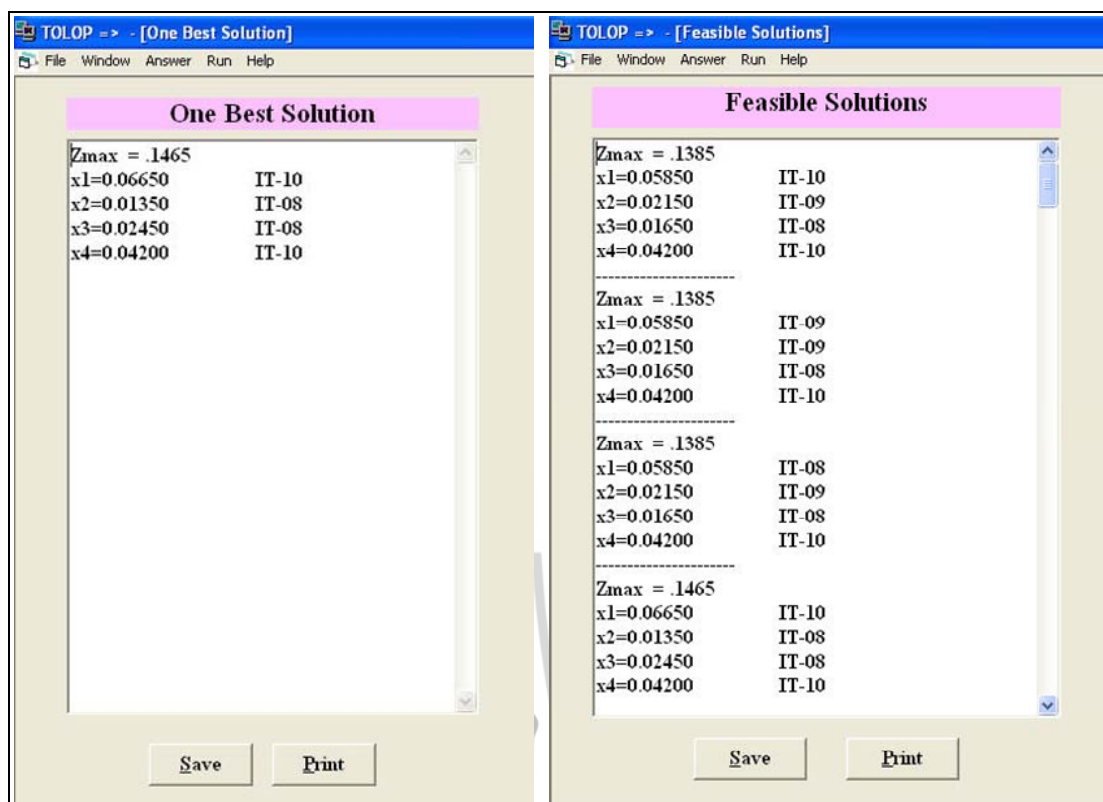
SAVE BACK SOLVE

↓

Old Dim	From	To	Basic Dim.	(+/-)1/2 Tol	New Dim.	From	To	M/C Operation	IT No	IT Values
Y1	A	B	20	.12	X1	A	B	การกลึง (Turret)	IT-08	= 0.0165
Y2	B	C	18	.08	X2	A	C	การกลึง (Turret)	IT-08	= 0.0135
Y3	C	D	20	.08	X3	C	E	การกลึง (Turret)	IT-08	= 0.0165
Y4	D	E	25	.1	X4	D	E	การกลึง (Turret)	IT-08	= 0.0165

SAVE BACK SOLVE

รูปที่ M3.3



รูปที่ M3.4

คำตอบที่ดีที่สุดของโจทย์ตัวอย่างนี้ คือ คำตอบที่มีค่าผลรวมของค่าความคลาดเคลื่อนมากที่สุด เท่ากับ 0.1465 ซึ่งอยู่ในขอบเขตของความคลาดเคลื่อนมาตรฐานที่มากที่สุด โดยมีค่าความคลาดเคลื่อนที่ได้ดังนี้

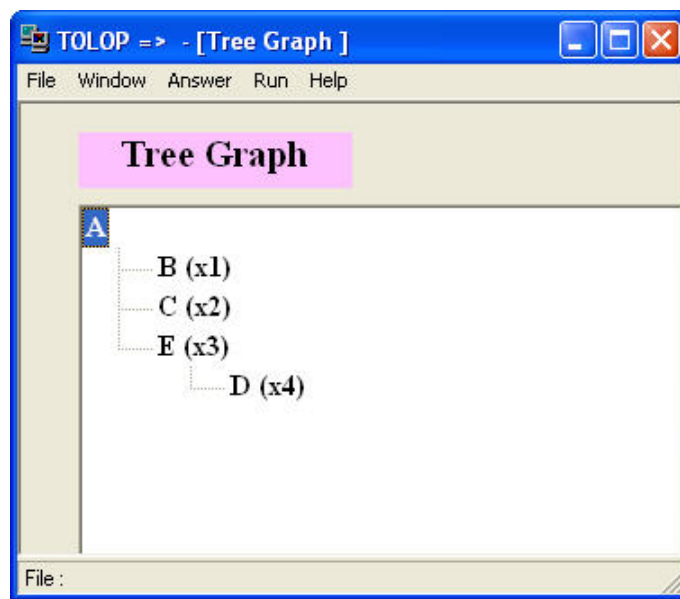
ค่าความคลาดเคลื่อนของมิติ $\pm x_1 = \pm 0.0665$ เลือกใช้เกรดความคลาดเคลื่อน IT 10

ค่าความคลาดเคลื่อนของมิติ $\pm x_2 = \pm 0.0135$ เลือกใช้เกรดความคลาดเคลื่อน IT 8

ค่าความคลาดเคลื่อนของมิติ $\pm x_3 = \pm 0.0245$ เลือกใช้เกรดความคลาดเคลื่อน IT 8

ค่าความคลาดเคลื่อนของมิติ $\pm x_4 = \pm 0.0420$ เลือกใช้เกรดความคลาดเคลื่อน IT 10

นอกจากคำตอบที่ดีที่สุดแล้วยังสามารถเลือกใช้คำตอบอื่น ๆ ในหน้าจอของ Feasible Solution ได้อีกด้วย และยังสามารถเลือกดูข้อมูลอื่น ๆ ที่ได้จากการประมวลผลของโปรแกรม เช่น หน้าจอแสดงกราฟต้นไม้ (รูปที่ M3.5) และหน้าจอแสดงสมการเริ่มต้น (รูปที่ M3.6)



รูปที่ M3.5

Equation

Dim.	Path	New Dim.
Y1	AB	x1
Y2	BC	x1+x2
Y3	CD	x2+x3+x4
Y4	DE	x4

Objective : Max Z = $x1+x2+x3+x4$

Subject to :

$x1$	$\leq$	0.1200
$x1+x2$	$\leq$	0.0800
$x2+x3+x4$	$\leq$	0.0800
$x4$	$\leq$	0.1000
$x1$	$\geq$	0.0165
$x2$	$\geq$	0.0135
$x3$	$\geq$	0.0165
$x4$	$\geq$	0.0165

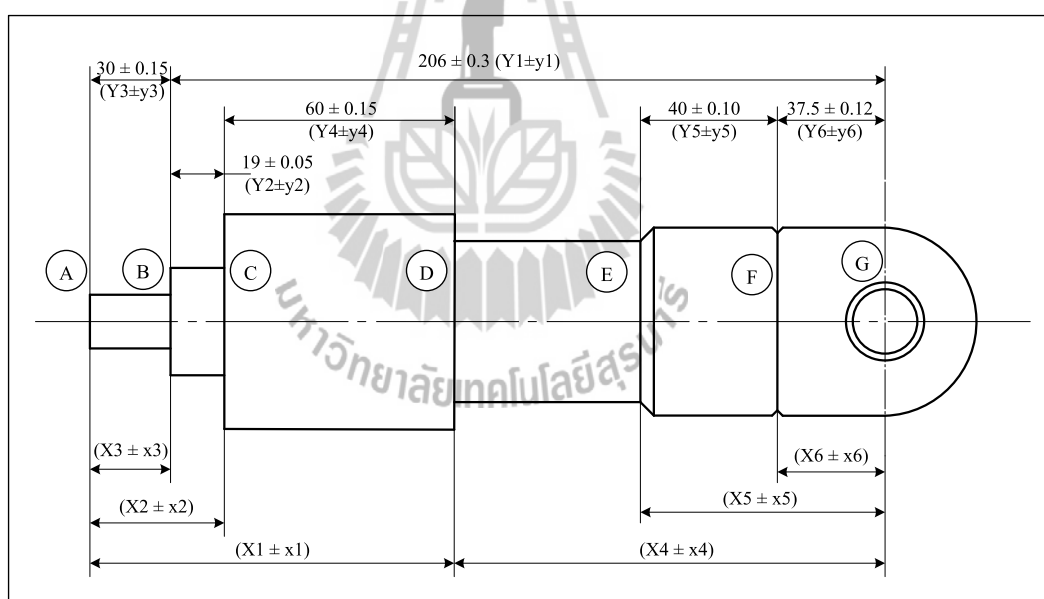
Save Print

รูปที่ M3.6

ตัวอย่างที่ 2

โจทย์ตัวอย่างนี้มีมิติของชิ้นงานจำนวน 6 มิติ (รูปที่ M3.7) ซึ่งระบุขนาดความคลาดเคลื่อนของมิติความยาวเป็นค่าเบี่ยงเบนเท่า ๆ กัน โดยกำหนดให้ $(Y1 \pm y1)$ ถึง $(Y6 \pm y6)$ เป็นมิติชุดเก่า และ $(X1 \pm x1)$ ถึง $(X6 \pm x6)$ เป็นมิติชุดใหม่ที่ต้องการทราบค่า การใช้โปรแกรมเพื่อหาค่าความคลาดเคลื่อนชุดใหม่ มีขั้นตอนดังนี้

- 1) ป้อนจำนวนมิติเท่ากับ 6 ลงในช่อง Input Number of dimensions แล้วกดปุ่ม OK หรือ กดปุ่ม Enter ที่คีย์บอร์ด
- 2) โปรแกรมจะแสดงหน้าต่างสำหรับป้อนข้อมูลจำนวน 6 แถว จากนั้นทำการป้อนข้อมูลลงในช่องว่างให้ครบทุกช่อง ดังรูปที่ M3.9
- 3) กดปุ่ม SAVE เพื่อทำการบันทึกข้อมูล
- 4) กดปุ่ม SOLVE เพื่อหาผลลัพธ์
- 5) เมื่อโปรแกรมประมวลผลเสร็จแล้วจะแสดงผลดังรูปที่ M3.10



รูปที่ M3.7 ตัวอย่างชิ้นงาน 6 มิติ

Input number of dimensions :

๕

OK

รูปที่ M3.8

Old Dim	From	To	Basic Dim	(+/-)1/2 Tol	New Dim	From	To	M/C Operation	IT No	IT Values	
Y1					X1			การไส (Shaping)	IT-09	=	
Y2					X2			การไส (Shaping)	IT-09	=	
Y3					X3			การไส (Shaping)	IT-09	=	
Y4					X4			การไส (Shaping)	IT-09	=	
Y5					X5			การไส (Shaping)	IT-09	=	
Y6					X6			การไส (Shaping)	IT-09	=	

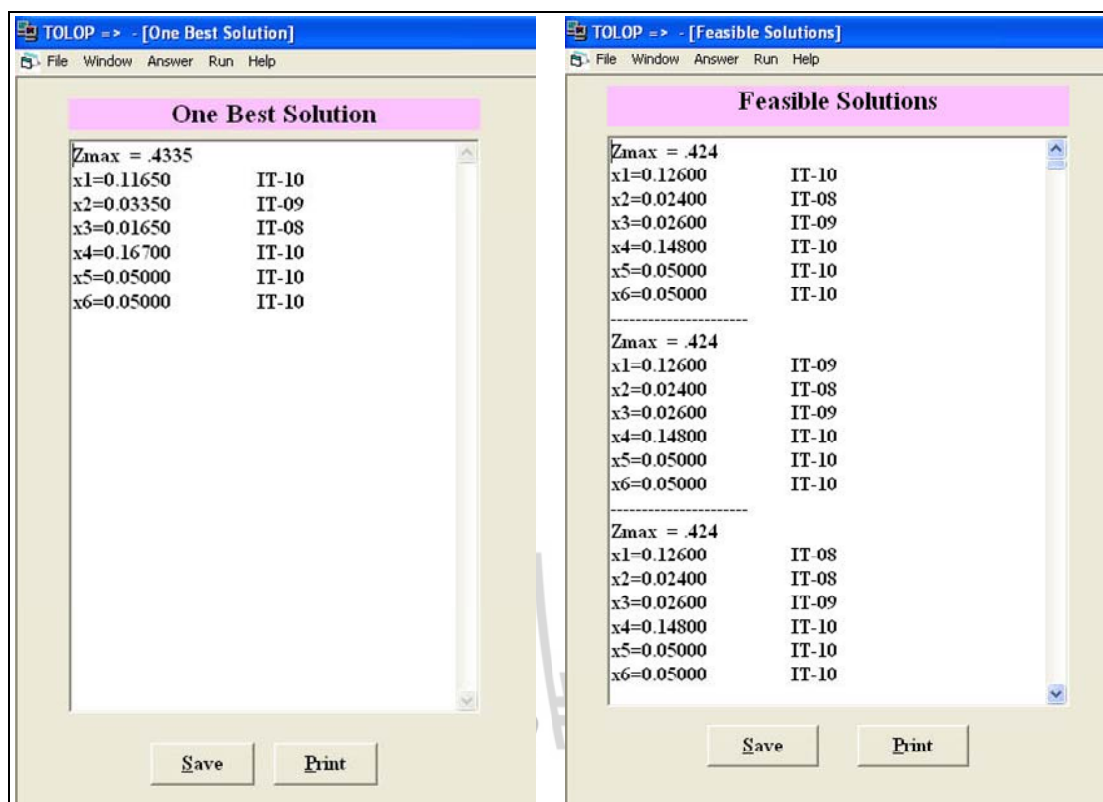
SAVE BACK SOLVE

มหาวิทยาลัยเทคโนโลยีสุรนารี

Old Dim	From	To	Basic Dim	(+/-)1/2 Tol	New Dim	From	To	M/C Operation	IT No	IT Values
Y1	B	G	206	.3	X1	A	D	การกลึง (Turret)	IT-08	= 0.0360
Y2	B	C	19	.05	X2	A	C	การกลึง (Turret)	IT-08	= 0.0165
Y3	A	B	30	.15	X3	A	B	การกลึง (Turret)	IT-08	= 0.0165
Y4	C	D	60	.15	X4	D	G	การกลึง (Turret)	IT-08	= 0.0230
Y5	E	F	40	.1	X5	E	G	การกลึง (Turret)	IT-08	= 0.0195
Y6	F	G	37.5	.12	X6	F	G	การกลึง (Turret)	IT-08	= 0.0195

SAVE BACK SOLVE

รูปที่ M3.9



รูปที่ M3.10

คำตอบที่ดีที่สุดของ โจทย์ตัวอย่างนี้ คือ คำตอบที่มีค่าผลรวมของค่าความคลาดเคลื่อนมากที่สุด เท่ากับ 0.4335 ซึ่งอยู่ในขอบเขตของความคลาดเคลื่อนมาตรฐานที่มากที่สุด โดยมีค่าความคลาดเคลื่อนที่ได้ดังนี้

ค่าความคลาดเคลื่อนของมิติ $\pm x_1 = \pm 0.1165$ เลือกใช้เกรดความคลาดเคลื่อน IT 10

ค่าความคลาดเคลื่อนของมิติ $\pm x_2 = \pm 0.0335$ เลือกใช้เกรดความคลาดเคลื่อน IT 9

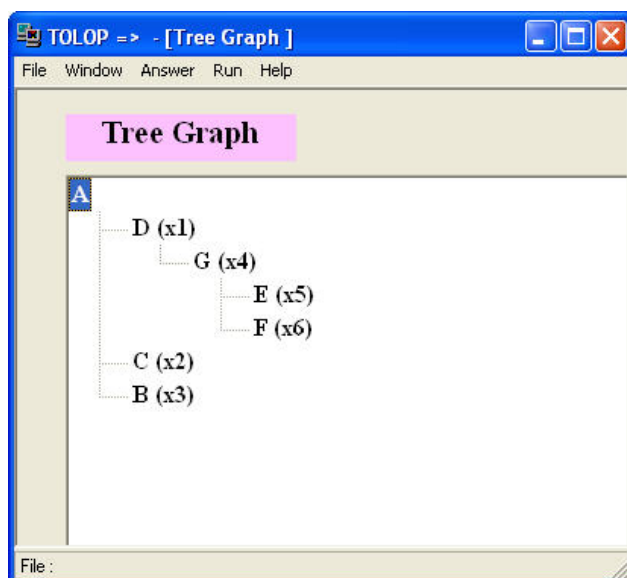
ค่าความคลาดเคลื่อนของมิติ $\pm x_3 = \pm 0.0165$ เลือกใช้เกรดความคลาดเคลื่อน IT 8

ค่าความคลาดเคลื่อนของมิติ $\pm x_4 = \pm 0.1670$ เลือกใช้เกรดความคลาดเคลื่อน IT 10

ค่าความคลาดเคลื่อนของมิติ $\pm x_5 = \pm 0.0500$ เลือกใช้เกรดความคลาดเคลื่อน IT 10

ค่าความคลาดเคลื่อนของมิติ $\pm x_6 = \pm 0.0500$ เลือกใช้เกรดความคลาดเคลื่อน IT 10

นอกจากคำตอบที่ดีที่สุดแล้วยังสามารถเลือกใช้คำตอบอื่น ๆ ในหน้าจอของ Feasible Solution ได้อีกด้วย และยังสามารถเลือกดูข้อมูลอื่น ๆ ที่ได้จากการประมวลผลของโปรแกรม เช่น หน้าจอแสดงกราฟต้นไม้ (รูปที่ M3.11) และหน้าจอแสดงสมการเริ่มต้น (รูปที่ M3.12)



រូបភាព M3.11

Equation

Dim.	Path	New Dim.
Y1	BG	$x1+x3+x4$
Y2	BC	$x2+x3$
Y3	AB	$x3$
Y4	CD	$x1+x2$
Y5	EF	$x5+x6$
Y6	FG	$x6$

Objective : Max Z = $x1+x2+x3+x4+x5+x6$

Subject to :

$x1+x3+x4$	$\leq$	0.3000
$x2+x3$	$\leq$	0.0500
$x3$	$\leq$	0.1500
$x1+x2$	$\leq$	0.1500
$x5+x6$	$\leq$	0.1000
$x6$	$\leq$	0.1200
$x1$	$\geq$	0.0360
$x2$	$\geq$	0.0165
$x3$	$\geq$	0.0165
$x4$	$\geq$	0.0230
$x5$	$\geq$	0.0195
$x6$	$\geq$	0.0195

Buttons: Save, Print

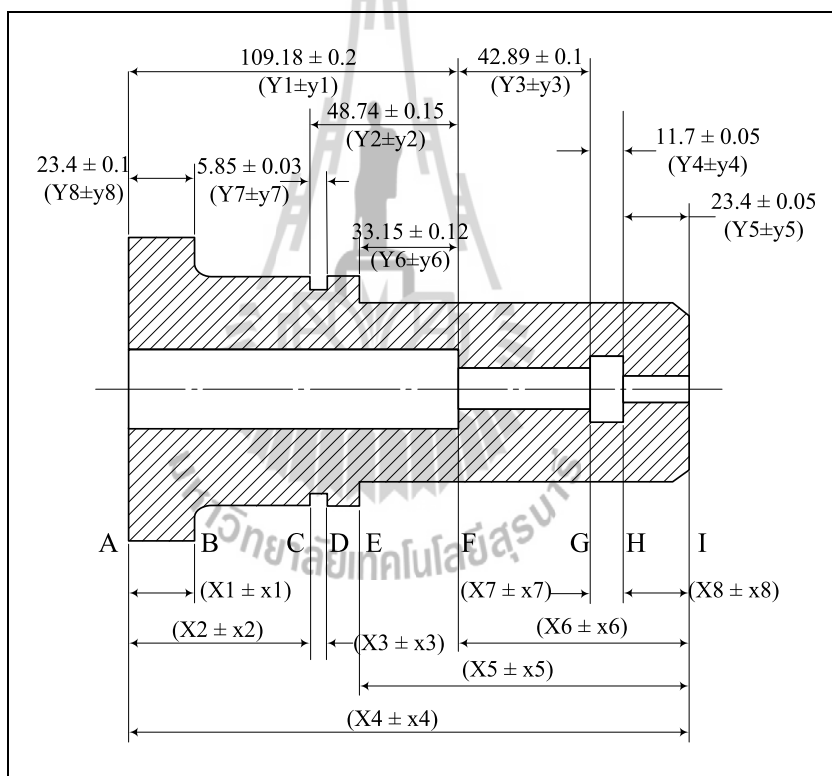
រូបភាព M3.12

ตัวอย่างที่ 3

โจทย์ตัวอย่างนี้มีมิติของชิ้นงานจำนวน 8 มิติ (รูปที่ M3.13) ซึ่งระบุขนาดความคลาดเคลื่อนของมิติความยาวเป็นค่าเบี่ยงเบนเท่า ๆ กัน โดยกำหนดให้ $(Y1 \pm y1)$ ถึง $(Y8 \pm y8)$ เป็นมิติชุดเก่า และ $(X1 \pm x1)$ ถึง $(X8 \pm x8)$ เป็นมิติชุดใหม่ที่ต้องการทราบค่า ขั้นตอนการใช้โปรแกรมเพื่อหาค่าความคลาดเคลื่อนชุดใหม่ มีขั้นตอนดังนี้

1) ป้อนจำนวนมิติเท่ากับ 8 ลงในช่อง Input Number of dimensions จากนั้นกด

ปุ่ม OK หรือกดปุ่ม Enter ที่คีย์บอร์ด โปรแกรมจะแสดงหน้าต่างสำหรับป้อนข้อมูลจำนวน 8 แถว เช่นเดียวกับทั้งสองตัวอย่างที่กล่าวมาข้างต้น จากนั้นป้อนข้อมูลลงในช่องว่างให้ครบทุกช่อง ดังรูปที่ M3.14



รูปที่ M3.13 ตัวอย่างชิ้นงาน 8 มิติ (Farmer and Harris, 1984)

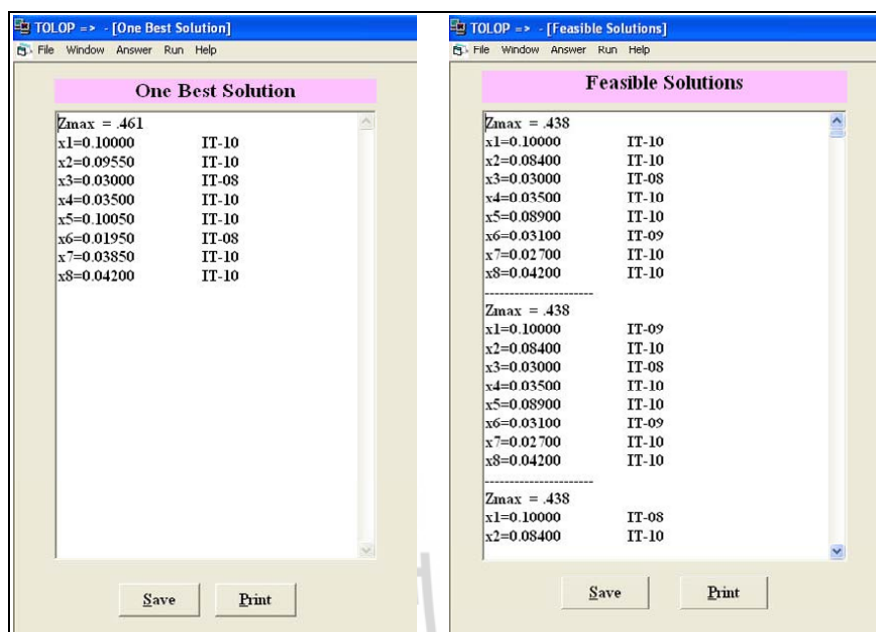
Old Dim	From	To	Basic Dim	(+/-)1/2 Tol	New Dim	From	To	M/C Operation	IT No	=	IT Values
Y1	A	F	109.18	.2	X1	A	B	การกัด (Milling)	IT-08	=	0.0270
Y2	C	F	48.74	.15	X2	A	C	การกัด (Milling)	IT-08	=	0.0195
Y3	F	G	42.89	.1	X3	C	D	การกัด (Milling)	IT-08	=	0.0195
Y4	G	H	11.7	.05	X4	A	I	การกัด (Milling)	IT-08	=	0.0135
Y5	H	I	23.4	.05	X5	E	I	การกัด (Milling)	IT-08	=	0.0165
Y6	E	F	33.15	.12	X6	F	I	การกัด (Milling)	IT-08	=	0.0195
Y7	C	D	5.85	.03	X7	G	H	การกัด (Milling)	IT-08	=	0.0090
Y8	A	B	23.4	.1	X8	H	I	การกัด (Milling)	IT-08	=	0.0165

SAVE BACK SOLVE

รูปที่ M3.14

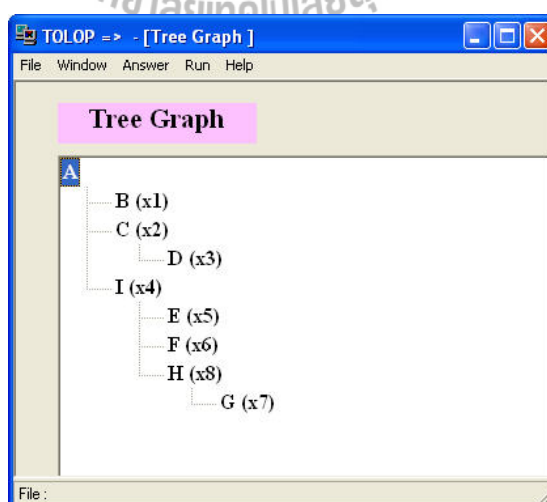
- 3) กดปุ่ม SAVE เพื่อทำการบันทึกข้อมูล
- 4) เมื่อกดปุ่ม SOLVE เพื่อหาผลลัพธ์ การประมวลผลของโปรแกรมในตัวอย่างนี้มีความแตกต่างจากตัวอย่างทั้ง 2 ตัวอย่างข้างต้น เนื่องจากรูปแบบของเกรด IT No ของตัวอย่างนี้มีมากกว่า 3,000 รูปแบบ โดยจะมีจำนวนรูปแบบอยู่ทั้งหมด 6,561 รูปแบบ ซึ่งมาจาก เกรด IT No ที่ใช้ในแต่ละมิติ คือ 3 เกรด ซึ่งมีมิติทั้งหมด 8 มิติ ดังนั้นจึงมีรูปแบบของเกรด IT No ทั้งหมดที่ต้องใช้ในการคำนวณ คือ 3^8 นั่นเอง เมื่อโปรแกรมทำการประมวลผลโดยใช้รูปแบบของเกรด IT No ครบ 3,000 รูปแบบแล้วจะแสดงหน้าจอจดังรูปที่ M3.15 ซึ่งในตัวอย่างนี้สามารถหาผลลัพธ์ได้แล้ว แต่อาจจะยังไม่ใช้คำตอบที่ดีที่สุด ถ้าต้องการที่จะให้โปรแกรมทำการประมวลผลต่อให้กดปุ่ม Yes โปรแกรมจะแสดงกล่องข้อความให้ใส่จำนวนรูปแบบที่ต้องการให้ทำต่อ ในตัวอย่างนี้เราจะให้โปรแกรมทำต่อไปอีก 3,561 รูปแบบ ดังรูปที่ M3.16
- 5) เมื่อโปรแกรมประมวลผลเสร็จแล้วจะแสดงผลดังรูปที่ M3.17 คำตอบที่ดีที่สุดของโจทย์ตัวอย่างนี้ คือ คำตอบที่มีค่าผลรวมของค่าความคลาดเคลื่อนมากที่สุด เท่ากับ 0.3835 ซึ่งอยู่ในขอบเขตของความคลาดเคลื่อนมาตรฐานที่มากที่สุด โดยมีค่าความคลาดเคลื่อนที่ได้ดังนี้

ค่าความคลาดเคลื่อนของมิติ $\pm x_1 = \pm 0.1000$ เลือกใช้เกรดความคลาดเคลื่อน IT 10
 ค่าความคลาดเคลื่อนของมิติ $\pm x_2 = \pm 0.0485$ เลือกใช้เกรดความคลาดเคลื่อน IT 9
 ค่าความคลาดเคลื่อนของมิติ $\pm x_3 = \pm 0.0300$ เลือกใช้เกรดความคลาดเคลื่อน IT 8
 ค่าความคลาดเคลื่อนของมิติ $\pm x_4 = \pm 0.0350$ เลือกใช้เกรดความคลาดเคลื่อน IT 10
 ค่าความคลาดเคลื่อนของมิติ $\pm x_5 = \pm 0.0780$ เลือกใช้เกรดความคลาดเคลื่อน IT 10
 ค่าความคลาดเคลื่อนของมิติ $\pm x_6 = \pm 0.0420$ เลือกใช้เกรดความคลาดเคลื่อน IT 10



รูปที่ M3.17

นอกจากคำตอบที่ดีที่สุดแล้วยังสามารถเลือกใช้คำตอบอื่น ๆ ในหน้าจอของ Feasible Solution ได้อีกด้วย และยังสามารถเลือกดูข้อมูลอื่น ๆ ที่ได้จากการประมวลผลของ โปรแกรม เช่น หน้าจอแสดงกราฟต้นไม้ (รูปที่ M3.18) และหน้าจอแสดงสมการเริ่มต้น (รูปที่ M3.19)



รูปที่ M3.18

Equation		
Dim.	Path	New Dim.
Y1	AF	$x_4 + x_6$
Y2	CF	$x_2 + x_4 + x_6$
Y3	FG	$x_6 + x_7 + x_8$
Y4	GH	x_7
Y5	HI	x_8
Y6	EF	$x_5 + x_6$
Y7	CD	x_3
Y8	AB	x_1
Objective : Max Z = $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$		
Subject to :		
	$x_4 + x_6$	≤ 0.2000
	$x_2 + x_4 + x_6$	≤ 0.1500
	$x_6 + x_7 + x_8$	≤ 0.1000
	x_7	≤ 0.0500
	x_8	≤ 0.0500
	$x_5 + x_6$	≤ 0.1200
	x_3	≤ 0.0300
	x_1	≤ 0.1000
	x_1	≥ 0.0270
	x_2	≥ 0.0195
	x_3	≥ 0.0195
	x_4	≥ 0.0135
	x_5	≥ 0.0165
	x_6	≥ 0.0195
	x_7	≥ 0.0090
	x_8	≥ 0.0165

รูปที่ M3.19



ภาควิชาวิศวกรรมเครื่องกล

รายการโปรแกรม (Program Listings)

Variable

```
Public Tstart As Date
Public Tstop As Date

Public TypeOfUpDn As Integer
Public CurrentDim As Integer
Public total_loop_count As Double
Public Create_Loop_of_IT As Integer
Public Max_It_change As Long
Public One_loop_Sdata As String

Public Zmax_value As String
Public X_before, X_After As String

Public XTree() As TreeNode
Public YTree() As TreeNode
Public Type TreeNode
    Fr As String
    Name As String
    To As String
    Dim As Single
    Tol As Single
End Type

Public Array_Y() As String
Public Array_All_Loop() As String
Public Array_String_Loop() As Integer
Public ArrayCal() As String
Public ArrayRatio() As String
Public ArrayAnswer() As String

Public Num_of_Basic As Integer
Public Num_of_Slack As Integer
Public Num_of_Nonbasic As Integer
```

Main Module

```
Private Sub MDIForm_Load()
    frmScreen.Show
    frmScreen.WindowState = 2
    Me.WindowState = 2
    frmMain.Caption = "TOLOP => "
    CurrentDim = 2
    Call frmTableIT.show_it_table
    Call frmTableIT.Hide
End Sub
```

```

Private Sub mnuEquation_Click()
    frmEquation.Show
    frmEquation.WindowState = 2
End Sub

Private Sub mnuExit_Click()
    Unload Me
End Sub

Private Sub mnuFeasibleSolution_Click()
    frmFeasibleSolution.Show
    frmFeasibleSolution.WindowState = 2
End Sub

Private Sub mnuNew_Click()
    Call frmInputData.clear_dia_to_initial
    frmInputDim.WindowState = 2
End Sub

Private Sub mnuOneBestSolution_Click()
    frmOneBestSol.Show
    frmOneBestSol.WindowState = 2
End Sub

Private Sub mnuOpen_Click()
    frmInputData.TV1.Nodes.Clear
    frmInputData.TV2.Nodes.Clear
    frmEquation.txtEquation.Text = ""
    frmFeasibleSolution.txtFeasibleSol.Text = ""
    frmOneBestSol.txtOneBest.Text = ""
    Call frmInputData.clear_dia_to_initial
    Call frmInputData.read_from_text_file
    frmInputData.WindowState = 2
End Sub

Private Sub mnuPrintEquation_Click()
    Call frmEquation.PrintEQ
End Sub

Private Sub mnuPrintForm_Click()
    PrintForm frmMain.ActiveForm
End Sub

Private Sub mnuPrintFS_Click()
    Call frmFeasibleSolution.PrintFS
End Sub

Private Sub mnuPrintOB_Click()
    Call frmOneBestSol.PrintOB

```

```

End Sub

Private Sub mnuRun_Click()
    Call frmInputDim.Show
    frmScreen.Hide
End Sub

Private Sub mnuSaveAs_Click()
    Call frmInputData.show_save_data
End Sub

Sub PrintForm(x As Form)
    Dim BeginPage, EndPage, NumCopies, i
    On Error Resume Next

    With cdPrintAll
        .CancelError = True
        .ShowPrinter
        If Not cancel Then
            If Err.Number = 0 Then
                BeginPage = .FromPage
                EndPage = .ToPage
                NumCopies = .Copies
                For i = 1 To NumCopies
                    Printer.FontBold = .FontBold
                    Printer.FontItalic = .FontItalic
                    Printer.FontSize = .FontSize
                    Printer.FontStrikethru = .FontStrikethru
                    Printer.FontUnderline = .FontUnderline
                    Printer.Orientation = vbPRORLandscape 'Wider than tall
                    x.PrintForm
                    Printer.EndDoc
                Next i
            End If
        End If
    End With
End Sub

```

Input Module

```

Private Sub cmdOK_Click()
    If (Val(txtDimension.Text) >= 3) And (Val(txtDimension.Text) <= 20) Then
        Call frmInputData.clear_dia_to_initial
        CurrentDim = Val(txtDimension.Text)
        frmInputData.WindowState = 2
        Call frmInputData.DrawForm
        Call frmInputData.show_it_initial
    Else
        MsgBox "Your number must between 3 To 20, Please try again."
    End If
End Sub

```

```

End If
End Sub

Sub DrawForm()
    Dim iCount As Integer
    Dim lineLevel As Integer
    Dim LineStart As Integer
    Dim LineHight As Integer
    LineStart = lblY(0).Top ' 1020
    LineHight = 390
    cbProcess(0).ListIndex = 0
    For iCount = 1 To CurrentDim - 1
        Load lblY(iCount)
        Load YFrom(iCount)
        Load YTo(iCount)
        Load YBasic(iCount)
        Load YTol(iCount)
        Load lblX(iCount)
        Load XFrom(iCount)
        Load XTo(iCount)
        Load cbProcess(iCount)
        cbProcess(iCount).AddItem "ขึ้น (Shaping)"
        cbProcess(iCount).AddItem "เจียร (Dill)"
        cbProcess(iCount).AddItem "ream (Reaming)"
        cbProcess(iCount).AddItem "กลึง (Milling)"
        cbProcess(iCount).AddItem "ปืนกล (Turret)"
        cbProcess(iCount).AddItem "เจียรน้อย (Grinding)"
        cbProcess(iCount).ListIndex = 0

        Load Data_I(iCount)
        Load lblEqua(iCount)
        Load It_No(iCount)
        lineLevel = LineStart + (LineHight * iCount)

        lblY(iCount).Top = lineLevel
        lblY(iCount).Caption = "Y" & iCount + 1
        lblY(iCount).Visible = True
        YFrom(iCount).Top = lineLevel
        YFrom(iCount).Visible = True
        YTo(iCount).Top = lineLevel
        YTo(iCount).Visible = True
        YBasic(iCount).Top = lineLevel
        YBasic(iCount).Visible = True
        YTol(iCount).Top = lineLevel
        YTol(iCount).Visible = True
        cbProcess(iCount).Top = lineLevel
        cbProcess(iCount).Visible = True
    
```

```

        lblX(iCount).Top = lineLevel
        lblX(iCount).Caption = "X" & iCount + 1
        lblX(iCount).Visible = True
        XFrom(iCount).Top = lineLevel
        XFrom(iCount).Visible = True
        XTo(iCount).Top = lineLevel
        XTo(iCount).Visible = True

        Data_It(iCount).Top = lineLevel
        Data_It(iCount).Visible = True

        lblEqua(iCount).Top = lineLevel
        lblEqua(iCount).Visible = True

        It_No(iCount).Top = lineLevel
        It_No(iCount).Visible = True
    Next iCount
    lineLevel = LineStart + (LineHight * CurrentDim) + XTo(0).Height
    Picture1.Top = lineLevel
    frmInputData.TV1.Nodes.Clear
    frmInputData.TV2.Nodes.Clear
    frmInputData.TV1.Nodes.Add , , "A", "A"
    frmInputData.TV2.Nodes.Add , , "A", "A"
    List1.Clear
End Sub

Sub SaveFile(fname As String)
    Dim iCount As Integer
    Open fname For Output As #1
    Print #1, CurrentDim
    For iCount = 0 To CurrentDim - 1
        Print #1, "Y" & iCount + 1 & "," & YFrom(iCount).Text & "," &
        YTo(iCount).Text & "," & YBasic(iCount).Text & "," & YTo(iCount).Text
        & "," & "X" & iCount + 1 & "," & XFrom(iCount).Text & "," &
        XTo(iCount).Text & "," & cbProcess(iCount).ListIndex
    Next iCount
    MsgBox "Data save to " & fname
    Close #1
End Sub

Sub read_from_text_file()
    Dim cancel As Boolean
    On Error GoTo ErrHandler
    cancel = False
    cdFilePath.FileName = App.Path & "*.txt"
    cdFilePath.Filter = "Text Documents (*.txt)*.txt"
    cdFilePath.ShowOpen
    cdFilePath.CancelError = True
    txtAnswer.Text = ""
    If Not cancel Then

```

```

        Call load_old_data
        Call DrawForm
        Call show_old_data
    End If
    Exit Sub
    ErrHandler:
    If Err.Number = cdlCancel Then
        cancel = True
        Resume Next
    End If
End Sub

Sub load_old_data()
    Dim iCount As Integer
    Dim datline As String
    Dim fname As String
    fname = cdFilePath.FileName
    iCount = 0
    Open fname For Input As #1
        Line Input #1, datline
        CurrentDim = Val(datline)
    Close #1
End Sub

Sub show_old_data()
    Dim iCount As Integer
    Dim datline As String
    Dim txtArr() As String
    Dim fname As String
    fname = cdFilePath.FileName
    iCount = 0
    Open fname For Input As #1
        Line Input #1, datline
        CurrentDim = Val(datline)
        frmInputDim.txtDimension = CurrentDim

        Do While Not EOF(1)
            Line Input #1, datline
            txtArr = Split(datline, ",")
            XFrom(iCount) = txtArr(6)
            XTo(iCount) = txtArr(7)
            YFrom(iCount) = txtArr(1)
            YTo(iCount) = txtArr(2)
            YBasic(iCount) = txtArr(3)
            YTol(iCount) = Val(txtArr(4))
            cbProcess(iCount).ListIndex = Val(txtArr(8))
            iCount = iCount + 1
        Loop
    Close #1
End Sub

```

```

Private Sub Form_Load()
    frmShowTree.TV2.Width = frmShowTree.Width - 1000
    frmShowTree.TV2.Height = frmShowTree.Height - 1000
    Me.Move (frmShowTree.Width - frmShowTree.TV2.Width) / 2, (frmShowTree.Height -
    frmShowTree.TV2.Height) / 2
End Sub

Sub show_it_initial()
    Dim i, Col_it, Row_it As Integer
    For i = 0 To frmInputData.It_No.Count - 1
        If cbProcess(i).ListIndex = 0 Then frmInputData.It_No(i) = "IT-09"
        If cbProcess(i).ListIndex = 1 Then frmInputData.It_No(i) = "IT-10"
        If cbProcess(i).ListIndex = 2 Then frmInputData.It_No(i) = "IT-06"
        If cbProcess(i).ListIndex = 3 Then frmInputData.It_No(i) = "IT-08"
        If cbProcess(i).ListIndex = 4 Then frmInputData.It_No(i) = "IT-08"
        If cbProcess(i).ListIndex = 5 Then frmInputData.It_No(i) = "IT-04"
        Col_it = Val(Mid(frmInputData.It_No(i), 4)) + 2
        Row_it = frmSimplexMethod.getIT_Row(Val(frmInputData.YBasic(i).Text))
        If Row_it <> 0 Then
            frmInputData.Data_It(i).Text = frmTableIT.MSH_tableIT.TextMatrix(Row_it, Val(Col_it))
        End If
    Next i
End Sub

Private Sub cbProcess_Click(Index As Integer)
    Call show_it_initial
End Sub

Function check_blank_table() As String
    Dim i As Integer
    Dim fB As Boolean
    Dim cancel As Boolean

    fB = True
    For i = 0 To YFrom.Count - 1
        If (IsNumeric(YFrom(i).Text) Or YFrom(i).Text = "") Then
            MsgBox "Input String only"
            fB = False
            YFrom(i).SetFocus
            GoTo exit_loop
        End If
        If (IsNumeric(YTo(i).Text) Or YTo(i).Text = "") Then
            MsgBox "Input String only"
            fB = False
            YTo(i).SetFocus
            GoTo exit_loop
        End If
    Next i
    If (Not IsNumeric(YBasic(i).Text)) Then
        MsgBox "Data All Must is Numeric only"
    End If
End Function

```

```

        fB = False
        YBasic(i).SetFocus
        GoTo exit_loop
    End If
    If (Not IsNumeric(YTol(i).Text)) Then
        MsgBox "Data All Must is Numeric only"
        fB = False
        YTol(i).SetFocus
        GoTo exit_loop
    End If
    If (IsNumeric(XFrom(i).Text) Or XFrom(i).Text = "") Then
        MsgBox "Input String only"
        fB = False
        XFrom(i).SetFocus
        GoTo exit_loop
    End If
    If (IsNumeric(XTo(i).Text) Or XTo(i).Text = "") Then
        MsgBox "Input String only"
        fB = False
        XTo(i).SetFocus
        GoTo exit_loop
    End If
Next i

exit_loop:
    If fB = True Then
        check_blank_table = "1"
    Else
        check_blank_table = "-1"
    End If
End Function

Private Sub cmdBack_Click()
    Call clear_dia_to_initial
    frmInputDim.WindowState = 2
End Sub

Private Sub cmdSolve_Click()
    CurrentDim = Val(frmInputDim.txtDimension)
    txtAnswer.Text = ""
    frmFeasibleSolution.Hide
    frmOneBestSol.Hide
    X_After = 0
    Zmax_value = ""
    frmProcess.Move (Screen.Width - frmProcess.Width) / 2, (Screen.Height - frmProcess.Height) / 2
    Tstart = Now
    frmInputData.TV1.Nodes.Clear
    frmInputData.TV2.Nodes.Clear
    If check_blank_table = 1 Then
        frmInputData.TV1.Nodes.Clear
    End If

```

```

frmInputData.TV2.Nodes.Clear
frmInputData.TV1.Nodes.Add , "A", "A"
frmInputData.TV2.Nodes.Add , "A", "A"
frmInputData.List1.Clear
frmShowTree.TV2.Nodes.Clear
frmShowTree.TV2.Nodes.Add , "A", "A"
Call frmInputData.create_tree_file
Create_Loop_of_IT = 0
frmProcess.ProgressBar1.Value = 0
frmProcess.Caption = "Process"
frmProcess.Show
Call frmEquation.create_initial_equation
Call frmSimplexMethod.create_msgrid
Call frmTableIT.save_total_loop
Call frmTableIT.cal_save_total_loop
Tstart = Now
End If
End Sub

Sub show_save_data()
Dim cancel As Boolean
On Error GoTo ErrHandler
cancel = False
If check_blank_table = "1" Then
cdFilePath.Filter = "Text Documents(*.txt)|*.txt"
cdFilePath.ShowSave
cdFilePath.CancelError = True
If Not cancel Then
frmMain.sbrStatus.Panels("Status").Text = cdFilePath.FileName
SaveFile cdFilePath.FileName
End If
End If
Exit Sub
ErrHandler:
If Err.Number = cdICancel Then
cancel = True
Resume Next
End If
End Sub

Private Sub cmdSave_Click()
Call show_save_data
End Sub

Private Sub cmdShowTree_Click()
frmShowTree.Show
frmShowTree.WindowState = 2
End Sub

Sub clear_dia_to_initial()

```

```

Dim iCount As Integer
If (frmInputData.lblY.Count > 1) Then
    For iCount = 1 To CurrentDim - 1
        Unload frmInputData.lblY(iCount)
        Unload frmInputData.YFrom(iCount)
        Unload frmInputData.YTo(iCount)
        Unload frmInputData.YBasic(iCount)
        Unload frmInputData.YToI(iCount)
        Unload frmInputData.lblX(iCount)
        Unload frmInputData.XFrom(iCount)
        Unload frmInputData.XTo(iCount)
        Unload frmInputData.cbProcess(iCount)
        Unload frmInputData.Data_It(iCount)
        Unload frmInputData.lblEqa(iCount)
        Unload frmInputData.It_No(iCount)
    Next iCount
End If
iCount = 0
frmInputData.YFrom(iCount).Text = ""
frmInputData.YTo(iCount).Text = ""
frmInputData.YBasic(iCount).Text = ""
frmInputData.YToI(iCount).Text = ""
frmInputData.XFrom(iCount).Text = ""
frmInputData.XTo(iCount).Text = ""
frmInputData.cbProcess(iCount).ListIndex = 0
frmInputData.Data_It(iCount).Text = ""
frmInputData.It_No(iCount).Text = ""
End Sub

```

Tree graph Module

```

Sub create_tree_file()
    Dim iCount As Integer
    Dim datline As String
    Dim txtArr() As String

    iCount = 0
    Open cdFilePath.FileName For Input As #1
        Line Input #1, datline
        CurrentDim = Val(datline)
        Do While Not EOF(1)
            Line Input #1, datline
            txtArr = Split(datline, ",")
            List1.AddItem txtArr(6) & txtArr(7) & txtArr(5)
            ReDim Preserve YTree(iCount)
            ReDim Preserve XTree(iCount)

            XTree(iCount).Name = txtArr(5)
        Loop
    Close #1
End Sub

```

```

        XTree(iCount).Fr = txtArr(6)
        XTree(iCount).To = txtArr(7)

        YTree(iCount).Name = txtArr(0)
        YTree(iCount).Fr = txtArr(1)
        YTree(iCount).To = txtArr(2)
        YTree(iCount).Dim = txtArr(3)
        YTree(iCount).Tol = Val(txtArr(4))
        iCount = iCount + 1
    Loop
Close #1
ProcessTV
Expand _All TV1
Expand _All TV2
Expand _All frmShowTree.TV2
End Sub

Private Sub ProcessTV()
    Dim i As Integer
    List3.Clear
    Do While List1.ListCount > 0
        For i = 0 To List1.ListCount - 1
            If AddTree(Left(List1.List(i), 1), Mid(List1.List(i), 2, 1), LCase(Trim(Mid(List1.List(i), 3)))) = False Then List3.AddItem List1.List(i)
        Next i
        List1.Clear
        NodeCheck
    Loop
End Sub

Private Function AddTree(myParent As String, myChild As String, myName As String) As Boolean
    On Error GoTo ErrAddTree
    Dim tmp As String
    Dim i As Integer

    If TV1.Nodes.Count > 1 Then
        List2.Clear
        For i = 1 To TV1.Nodes.Count
            List2.AddItem TV1.Nodes(i).Key
        Next i
        List2.Text = myParent
        If List2.ListIndex = -1 Then
            tmp = myParent
            myParent = myChild
            myChild = tmp
        End If
    End If

    TV1.Nodes.Add myParent, tvwChild, myChild, myName
    TV2.Nodes.Add myParent, tvwChild, myChild, myChild & " (" & myName & ")"

```

```

frmShowTree.TV2.Nodes.Add myParent, twwChild, myChild, myChild & " (" &
myName & ")"
AddTree = True
Exit Function

ErrAddTree:
AddTree = False
Exit Function
End Function

Private Sub NodeCheck()
Dim i As Integer
If List3.ListCount > 0 Then
For i = 0 To List3.ListCount - 1
List1.AddItem List3.List(i)
Next i
List3.Clear
End If
End Sub

Function GetRoot(FromNode As String, ToNode As String) As String
Dim FromStr As String
Dim ToStr As String
Dim SumStr As String
Dim CutPos As Integer

FromStr = TV1.Nodes(FromNode).FullPath
ToStr = TV1.Nodes(ToNode).FullPath
CutPos = InStr(FromStr, "+")
If CutPos <> 0 Then
FromStr = Mid(FromStr, CutPos)
Else
FromStr = ""
End If
CutPos = InStr(ToStr, "+")
If CutPos <> 0 Then
ToStr = Mid(ToStr, CutPos)
Else
ToStr = ""
End If

List4.Clear
AddTraceRoot FromStr
AddTraceRoot ToStr
SumStr = ""
For CutPos = 0 To List4.ListCount - 1
SumStr = SumStr & "+" & List4.List(CutPos)
Next CutPos
SumStr = Mid(SumStr, 2)
GetRoot = SumStr

```

End Function

```
Private Sub AddTraceRoot(txtINP As String)
    Dim i As Integer
    Dim txtSplit() As String
    txtSplit = Split(txtINP, "+")
    If UBound(txtSplit) > 0 Then
        For i = 0 To UBound(txtSplit)
            List4.Text = txtSplit(i)
            If List4.ListIndex >= 0 Then
                List4.RemoveItem (List4.ListIndex)
            Else
                If Trim(txtSplit(i)) <> "" Then List4.AddItem txtSplit(i)
            End If
        Next i
    End If
End Sub
```

```
Private Sub Expand_All(myTV As TreeView)
    Dim i As Integer
    For i = 0 To myTV.Nodes.Count - 1
        myTV.Nodes.Item(i + 1).Expanded = True
    Next i
End Sub
```

IT No data base Module

```
Private Sub Modify_grid_MSH_tableIT()
    Dim i As Integer
    With frmTableIT.MSH_tableIT
        .ColWidth(0) = 600
        .ColWidth(1) = 600
        .ColWidth(2) = 600
        For i = 1 To 18
            .ColWidth(i + 2) = 650
            .TextMatrix(0, i + 2) = "IT-" & i
        Next i
        .TextMatrix(0, 1) = "From"
        .TextMatrix(0, 2) = "To"
    End With
End Sub
```

```
Sub show_it_table()
    Dim i
    Modify_grid_MSH_tableIT
    i = 1
    Open App.Path & "\itnumber.tab" For Input As #1
    Do While Not EOF(1)
        Line Input #1, datline
```

```

datline = Trim(datline)
If Left(datline, 2) <> "/" Then
txtArr = Split(datline, ",")
frmTableIT.MSH_tableIT.TextMatrix(i, 0) = i 'MSHFlexGrid1.Rows - 1
frmTableIT.MSH_tableIT.TextMatrix(i, 1) = Val(txtArr(0))
frmTableIT.MSH_tableIT.TextMatrix(i, 2) = Val(txtArr(1))
frmTableIT.MSH_tableIT.TextMatrix(i, 3) = Format(Val(txtArr(2)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 4) = Format(Val(txtArr(3)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 5) = Format(Val(txtArr(4)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 6) = Format(Val(txtArr(5)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 7) = Format(Val(txtArr(6)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 8) = Format(Val(txtArr(7)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 9) = Format(Val(txtArr(8)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 10) = Format(Val(txtArr(9)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 11) = Format(Val(txtArr(10)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 12) = Format(Val(txtArr(11)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 13) = Format(Val(txtArr(12)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 14) = Format(Val(txtArr(13)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 15) = Format(Val(txtArr(14)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 16) = Format(Val(txtArr(15)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 17) = Format(Val(txtArr(16)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 18) = Format(Val(txtArr(17)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 19) = Format(Val(txtArr(18)), "#0.0000")
frmTableIT.MSH_tableIT.TextMatrix(i, 20) = Format(Val(txtArr(19)), "#0.0000")
i = i + 1
End If
Loop
Close #1
End Sub

```

Optimization Module

```

Sub create_msgrid()
Dim i, j, m, n, k As Integer
Dim tmpString As String
Num_of_Basic = 0
Num_of_Slack = 0
For i = 0 To UBound(YTree)
    Num_of_Basic = Num_of_Basic + 1
    Num_of_Slack = Num_of_Slack + 1
Next i

For i = 0 To UBound(YTree)
    Num_of_Basic = Num_of_Basic + 1
    Num_of_Slack = Num_of_Slack + 2
Next i

Num_of_Nonbasic = i
ReDim ArrayCal(Num_of_Basic + 2, Num_of_Nonbasic + Num_of_Slack + 2)

```

```

ReDim ArrayAnswer(Num_of_Basic + 2, 2)
ReDim ArrayRatio(Num_of_Basic + 2)

frmSimplexMethod.MSH_Solution.Clear
frmSimplexMethod.MSH_Solution.Rows = Num_of_Basic + 2
frmSimplexMethod.MSH_Solution.Cols = Num_of_Nonbasic + Num_of_Slack + 2 + 1
frmSimplexMethod.MSH_Solution.Height = frmSimplexMethod.Height - 100
frmSimplexMethod.MSHFlexAnswer.Rows = Val(frmInputDim.txtDimension.Text) + 1
frmSimplexMethod.MSHFlexAnswer.Cols = 3

For i = 1 To Val(frmInputDim.txtDimension.Text)
    frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 0) = "x" & i '0
Next i

With frmSimplexMethod.MSH_Solution
    .ColWidth(0) = 500
    For i = 1 To frmSimplexMethod.MSH_Solution.Cols - 3
        .ColWidth(i) = (.Width - 3000) / (frmSimplexMethod.MSH_Solution.Cols)
    Next i
End With

With frmSimplexMethod.MSH_Solution
    For i = 1 To frmSimplexMethod.MSH_Solution.Cols - 1
        .ColAlignment(i) = 4
    Next i
End With

For i = 0 To Num_of_Basic + 1
    For j = 0 To Num_of_Nonbasic + Num_of_Slack + 1
        frmSimplexMethod.MSH_Solution.TextMatrix(i, j) = ""
    Next j
Next i

'-----Add Equation in row0-----

frmSimplexMethod.MSH_Solution.TextMatrix(0, 0) = ""
frmSimplexMethod.MSH_Solution.TextMatrix(1, 0) = "Z"
frmSimplexMethod.MSH_Solution.TextMatrix(1, Num_of_Nonbasic + Num_of_Slack + 1) = 0

For i = 0 To UBound(XTree)
    frmSimplexMethod.MSH_Solution.TextMatrix(0, i + 1) = LCase(XTree(i).Name)
Next i
    j = i + 1
For i = 0 To Num_of_Slack - 1
    frmSimplexMethod.MSH_Solution.TextMatrix(0, j) = "S" & i + 1
    j = j + 1
Next i
frmSimplexMethod.MSH_Solution.TextMatrix(0, Num_of_Nonbasic + Num_of_Slack + 1) = "Solution"
frmSimplexMethod.MSH_Solution.TextMatrix(0, Num_of_Nonbasic + Num_of_Slack + 2) = "Ratio"

'-----Add Equation in row1-----

```

```

For i = 0 To UBound(XTree)
    frmSimplexMethod.MSH_Solution.TextMatrix(1, i + 1) = -1
Next i
'-----Add Equation <-----
For i = 0 To UBound(YTree)
    tmpString = frmInputData.GetRoot(YTree(i).Fr, YTree(i).To)
    frmSimplexMethod.MSH_Solution.TextMatrix(i + 2, Num_of_Nonbasic + Num_of_Slack
    + 1) = Format(YTree(i).Tol, "0.0000")
    frmSimplexMethod.MSH_Solution.TextMatrix(i + 2, Num_of_Nonbasic + i + 1) = "1"
    frmSimplexMethod.MSH_Solution.TextMatrix(i + 2, 0) = "S" & i + 1
    m = 0
    Do
        m = InStr(m + 1, tmpString, "x")
        n = InStr(m + 2, tmpString, "+")
        If m And n Then
            k = Val(Mid$(tmpString, m + 1, n - m - 1))
            frmSimplexMethod.MSH_Solution.TextMatrix(i + 2, k) = 1
            m = n
        Else
            If m Then
                k = Val(Mid$(tmpString, m + 1, Len(tmpString) - m))
                frmSimplexMethod.MSH_Solution.TextMatrix(i + 2, k) = 1
            End If
        End If
    Loop While m
Next i
'-----end unit1 ----->-----
j = i
For i = 0 To UBound(YTree)
    tmpString = LCase(XTree(i).Name)
    frmSimplexMethod.MSH_Solution.TextMatrix(i + 2 + j, Num_of_Nonbasic + i + 1 + j) = "-1"
    frmSimplexMethod.MSH_Solution.TextMatrix(i + 2 + j, Num_of_Nonbasic + i + 1 + j +
    Num_of_Nonbasic) = "1"
    frmSimplexMethod.MSH_Solution.TextMatrix(i + 2 + j, 0) = "S" & i + 1 + j
    m = 0
    Do
        m = InStr(m + 1, tmpString, "x")
        n = InStr(m + 2, tmpString, "+")
        If m And n Then
            k = Val(Mid$(tmpString, m + 1, n - m - 1))
            frmSimplexMethod.MSH_Solution.TextMatrix(i + 2 + j, k) = 1
            m = n
        Else
            If m Then
                k = Val(Mid$(tmpString, m + 1, Len(tmpString) - m))
                frmSimplexMethod.MSH_Solution.TextMatrix(i + 2 + j, k) = 1
            End If
        End If
    Loop While m
Next i

```

```
Call add_IT_No  
End Sub
```

```
Sub add_IT_No()  
Dim i, Col_it, Row_it As Integer  
For i = 0 To CurrentDim - 1  
    Col_it = Val(Mid(frmInputData.It_No(i), 4)) + 2  
    Row_it = getIT_Row(Val(frmInputData.YBasic(i).Text))  
    frmInputData.Data_It(i).Text = frmTableIT.MSH_tableIT.TextMatrix(Row_it, Val(Col_it))  
  
    frmSimplexMethod.MSH_Solution.TextMatrix(i + 2 + Num_of_Nonbasic,  
    Num_of_Nonbasic + Num_of_Slack + 1) = frmInputData.Data_It(i).Text  
  
    frmSimplexMethod.MSHFlexAnswer.TextMatrix(i + 1, 1) = "0"  
Next i  
End Sub
```

```
Function getIT_Row(BasicSizes As Integer) As String  
Dim ix As Integer  
Dim row_no As Integer  
row_no = 0  
For ix = 0 To frmTableIT.MSH_tableIT.Rows - 2  
    If Val(frmTableIT.MSH_tableIT.TextMatrix(ix + 1, 2)) < BasicSizes Then  
        row_no = ix + 2  
    End If  
Next ix  
getIT_Row = row_no  
End Function
```

```
Function check_x() As String  
Dim data_value, data_x, data_table, tmpString As String  
Dim n, m As Integer  
Dim ixx As Integer  
Dim xcount As Integer  
check_x = "1"  
xcount = 0
```

```
For n = 2 To (Num_of_Basic + 1)  
    tmpString = frmSimplexMethod.MSH_Solution.TextMatrix(n, 0)  
    m = 0  
    m = InStr(tmpString, "x")  
    If m Then  
        xcount = xcount + 1  
        data_value = Val(Mid(tmpString, 2))  
        ixx = Val(data_value)  
        data_x = Val(frmInputData.Data_It(Val(data_value) - 1))  
  
        data_table = Val(frmSimplexMethod.MSH_Solution.TextMatrix(n,  
        Num_of_Nonbasic + Num_of_Slack + 1))
```

```

frmSimplexMethod.MSHFlexAnswer.TextMatrix(ixx, 1) = data_table

frmSimplexMethod.MSHFlexAnswer.TextMatrix(0, 1) =
frmSimplexMethod.MSH_Solution.TextMatrix(1, Num_of_Nonbasic + Num_of_Slack + 1)

If (Val(data_table) < Val(data_x)) Then
    check_x = "-1"
    Exit Function
End If
End If
Next n
X_After = xcount
check_x = "1"
End Function

Function find_solution_step() As String
Dim i, j, m, k, c, col_min, row_min, mul_1, pivot_num As Integer
If (check_x = "-1") Then
    find_solution_step = "-2"
    Exit Function
End If

c = 0
For j = 1 To (Num_of_Nonbasic + Num_of_Slack)
    k = Val(frmSimplexMethod.MSH_Solution.TextMatrix(1, j))
    If (k < c) Then
        c = k
        col_min = j
    End If
Next j

If Val(c) >= 0 Then
    find_solution_step = "-1"
    Exit Function
End If

For i = 1 To (Num_of_Basic + 1)
    frmSimplexMethod.MSH_Solution.TextMatrix(i, Num_of_Nonbasic + Num_of_Slack + 2) = "?"
Next i

For i = 2 To (Num_of_Basic + 1)
    k = Val(frmSimplexMethod.MSH_Solution.TextMatrix(i, col_min))
    If k > 0 Then
        frmSimplexMethod.MSH_Solution.TextMatrix(i, Num_of_Nonbasic + Num_of_Slack + 2) =
Val(frmSimplexMethod.MSH_Solution.TextMatrix(i, Num_of_Nonbasic + Num_of_Slack + 1)) / k
    Else
        frmSimplexMethod.MSH_Solution.TextMatrix(i, Num_of_Nonbasic + Num_of_Slack + 2) = "?"
    End If
Next i
k = 0

```

```

m = 999999
row_min = 0

For i = 2 To (Num_of_Basic + 1)
    If frmSimplexMethod.MSH_Solution.TextMatrix(i, Num_of_Nonbasic + Num_of_Slack + 2)
    <> "?" Then
        k = Val(frmSimplexMethod.MSH_Solution.TextMatrix(i, Num_of_Nonbasic +
            Num_of_Slack + 2))
        If k < m Then
            m = k
            row_min = i
        End If
    End If
Next i

If m <> 0 Then
    pivot_num = Val(frmSimplexMethod.MSH_Solution.TextMatrix(row_min, col_min))
    For j = 1 To (Num_of_Nonbasic + Num_of_Slack + 1)
        frmSimplexMethod.MSH_Solution.TextMatrix(row_min, j) =
            Val(frmSimplexMethod.MSH_Solution.TextMatrix(row_min, j)) / pivot_num
    Next j
Else
End If

For i = 1 To (Num_of_Basic + 1)
    If i <> row_min Then
        If Val(frmSimplexMethod.MSH_Solution.TextMatrix(i, col_min)) < 0 Then
            mul_1 = Val(frmSimplexMethod.MSH_Solution.TextMatrix(i, col_min)) * -1
            For j = 1 To (Num_of_Nonbasic + Num_of_Slack + 1)
                frmSimplexMethod.MSH_Solution.TextMatrix(i, j)
                =Val(frmSimplexMethod.MSH_Solution.TextMatrix(i, j))
                +Val(frmSimplexMethod.MSH_Solution.TextMatrix(row_min, j)) * mul_1
            Next j
        Else:
            If Val(frmSimplexMethod.MSH_Solution.TextMatrix(i, col_min)) > 0 Then
                mul_1 = Val(frmSimplexMethod.MSH_Solution.TextMatrix(i, col_min))
                For j = 1 To (Num_of_Nonbasic + Num_of_Slack + 1)
                    frmSimplexMethod.MSH_Solution.TextMatrix(i, j)
                    =Val(frmSimplexMethod.MSH_Solution.TextMatrix(i, j))
                    -Val(frmSimplexMethod.MSH_Solution.TextMatrix(row_min, j)) * mul_1
                Next j
            End If
        End If
    End If
Next i

frmSimplexMethod.MSH_Solution.TextMatrix(row_min, 0) =
frmSimplexMethod.MSH_Solution.TextMatrix(0, col_min)
find_solution_step = "1"
End Function

```

```

Sub show_answer()
Dim str_ans, str_ans1 As String
Dim Max_Z As String
Dim i As Integer
str_ans = ""
str_ans1 = ""
str_ans = str_ans & "Zmax = " & frmSimplexMethod.MSHFlexAnswer.TextMatrix(0, 1) & vbCrLf
Max_Z = (frmSimplexMethod.MSHFlexAnswer.TextMatrix(0, 1))

For i = 1 To CurrentDim
    str_ans1 = str_ans1 & frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 0) & "=" &
    Format(frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 1), "#0.00000") & vbTab &
    vbTab & frmInputData.It_No(i - 1) & vbCrLf
Next i

frmFeasibleSolution.txtFeasibleSol.Text = frmFeasibleSolution.txtFeasibleSol.Text & str_ans & str_ans1 &
"-----" & vbCrLf

If Val(Max_Z) > Val(Zmax_value) Then
    Zmax_value = Max_Z
    frmInputData.txtAnswer = str_ans & str_ans1
    frmOneBestSol.txtOneBest = frmInputData.txtAnswer.Text
End If
End Sub

Sub show_answer_before1()
Dim str_ans, str_ans1 As String
Dim Max_Z As String
Dim i As Integer
str_ans = ""
str_ans1 = ""
str_ans = str_ans & "Zmax = " & frmSimplexMethod.MSHFlexAnswer.TextMatrix(0, 2) & vbCrLf
Max_Z = (frmSimplexMethod.MSHFlexAnswer.TextMatrix(0, 2))
For i = 1 To CurrentDim
    str_ans1 = str_ans1 & frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 0) & "=" &
    Format(frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 2), "#0.00000") & vbTab &
    frmInputData.It_No(i - 1) & vbCrLf
Next i

frmFeasibleSolution.txtFeasibleSol.Text = frmFeasibleSolution.txtFeasibleSol.Text & str_ans & str_ans1 &
"-----" & vbCrLf

If Val(Max_Z) > Val(Zmax_value) Then
    Zmax_value = Max_Z
    frmInputData.txtAnswer = str_ans & str_ans1
    frmOneBestSol.txtOneBest = frmInputData.txtAnswer.Text
End If
End Sub

```

```

Function check_all_x_zero_ornot() As String
Dim xString As String
xString = "1"
For i = 1 To Val(frmInputDim.txtDimension.Text)
    If Val(frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 1)) = 0 Then
        sString = "0"
    End If
Next i
check_all_x_zero_ornot = xString
End Function

```

```

Function check_answer_before() As Integer
Dim i, j, jtest, jIt As Integer
jtest = 1

For i = 1 To Val(frmInputDim.txtDimension.Text)
    j = Val(frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 2))
    If j <= 0 Then
        jtest = 0
    End If
Next i

If jtest > 0 Then
    jtest = 1
    For i = 1 To Val(frmInputDim.txtDimension.Text)
        j = Val(frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 2))
        jIt = Val(frmInputData.Data_It(i - 1))
        If j < jIt Then
            jtest = 0
        End If
    Next i
    If jtest > 0 Then
        check_answer_before = 1
        Exit Function
    End If
End If

check_answer_before = 0
End Function

```

```

Function move_answer_before() As Integer
Dim i, j As Integer
For i = 1 To Val(frmInputDim.txtDimension.Text)
    j = Val(frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 1))
    frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 2) =
    frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 1)
    frmSimplexMethod.MSHFlexAnswer.TextMatrix(0, 2) =
    frmSimplexMethod.MSHFlexAnswer.TextMatrix(0, 1)
    frmSimplexMethod.MSHFlexAnswer.TextMatrix(i, 1) = "0"

```

```

Next i
move_answer_before = 0
End Function

```

Modification Module

```

Sub save_total_loop()
Dim i As Integer
ReDim Array_All_Loop(CurrentDim, 4)
ReDim Array_String_Loop(CurrentDim)
For i = 0 To frmInputData.YFrom.Count - 1
    If frmInputData.cbProcess(i).ListIndex = 0 Then    '"การใส่โลหะ"'
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 9
        Array_All_Loop(i, 2) = 10
        Array_All_Loop(i, 3) = 11
    End If
    If frmInputData.cbProcess(i).ListIndex = 1 Then    '"การเจาะ"'
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 10
        Array_All_Loop(i, 2) = 11
        Array_All_Loop(i, 3) = 12
    End If
    If frmInputData.cbProcess(i).ListIndex = 2 Then    '"ฐาน"'
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 6
        Array_All_Loop(i, 2) = 7
        Array_All_Loop(i, 3) = 8
    End If
    If frmInputData.cbProcess(i).ListIndex = 3 Then    '"ก้น"'
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 8
        Array_All_Loop(i, 2) = 9
        Array_All_Loop(i, 3) = 10
    End If
    If frmInputData.cbProcess(i).ListIndex = 4 Then    '"กลิ้ง"'
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 8
        Array_All_Loop(i, 2) = 9
        Array_All_Loop(i, 3) = 10
    End If
    If frmInputData.cbProcess(i).ListIndex = 5 Then    '"ฉนวนน้อย"'
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 4
        Array_All_Loop(i, 2) = 5
        Array_All_Loop(i, 3) = 6
    End If

```

```

        Array_String_Loop(i) = Array_All_Loop(i, 0)
    Next i
End Sub
Sub save_total_loop2()
Dim i As Integer
ReDim Array_All_Loop(CurrentDim, 4)
ReDim Array_String_Loop(CurrentDim)
For i = 0 To frmInputData.YFrom.Count - 1
    If frmInputData.cbProcess(i).ListIndex = 0 Then    ' "การใส่โลหะ"
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 11
        Array_All_Loop(i, 2) = 10
        Array_All_Loop(i, 3) = 9
    End If
    If frmInputData.cbProcess(i).ListIndex = 1 Then    ' "การเจาะ"
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 12
        Array_All_Loop(i, 2) = 11
        Array_All_Loop(i, 3) = 10
    End If
    If frmInputData.cbProcess(i).ListIndex = 2 Then    ' "ฉาบ"
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 8
        Array_All_Loop(i, 2) = 7
        Array_All_Loop(i, 3) = 6
    End If
    If frmInputData.cbProcess(i).ListIndex = 3 Then    ' "ทาสี"
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 10
        Array_All_Loop(i, 2) = 9
        Array_All_Loop(i, 3) = 8
    End If
    If frmInputData.cbProcess(i).ListIndex = 4 Then    ' "ทาสีเงา"
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 10
        Array_All_Loop(i, 2) = 9
        Array_All_Loop(i, 3) = 8
    End If
    If frmInputData.cbProcess(i).ListIndex = 5 Then    ' "เฟอร์นิเจอร์"
        Array_All_Loop(i, 0) = 3
        Array_All_Loop(i, 1) = 6
        Array_All_Loop(i, 2) = 5
        Array_All_Loop(i, 3) = 4
    End If
    Array_String_Loop(i) = Array_All_Loop(i, 0)
Next i
End Sub

```

```

Sub change_IT()
Dim j As Double
j = 0
Do While j < Val(frmInputData.YFrom.Count - 1)
    Array_Y(j + 1) = Array_Y(j + 1) + 1
    If Array_Y(j + 1) >= 3 Then
        If Array_Y(frmInputData.YFrom.Count - 1) >= 3 Then
            j = frmInputData.YFrom.Count + 3 'Exit loop
        Else
            Array_Y(j + 1) = 0
            j = j + 1
        End If
    Else
        j = frmInputData.YFrom.Count + 3 'Exit loop
    End If
Loop
End Sub

Sub cal_save_total_loop()
Dim i As Double
Dim j As Double
Dim resp_data As String
Dim countxx, old_countxx As Long
Dim total_loop As Double
Dim Max_next_loop As Double
Dim sum_total_loop As Double
Dim strInput As String
Max_next_loop = 3000 'initial data
sum_total_loop = 0
ReDim Array_Y(frmInputData.YFrom.Count)

For i = 0 To frmInputData.YFrom.Count - 1
    Array_Y(i) = 0
Next i

Max_It_change = 3 ^ frmInputData.YFrom.Count
frmProcess.ProgressBar1.Max = Max_It_change + 1 'cal max Number
frmProcess.ProgressBar1.Value = 0
frmProcess.Show
countxx = 0
old_countxx = old_countxx
total_loop = 0

Do While Array_Y(frmInputData.YFrom.Count - 1) < 3
    If TypeOfUpDn = 1 Then
        old_countxx = old_countxx + 10
        TypeOfUpDn = 0
    End If

    If old_countxx <> countxx Then

```

```

old_countxx = countxx
X_After = 0
X_before = 0

For i = 0 To frmInputData.YFrom.Count - 1
    frmInputData.It_No(i).Text = "IT-" & Format(Array_All_Loop(i, Array_Y(i) +
1), "#00") 'Update data IT change
Next i

Call frmSimplexMethod.create_msgrid

End If

Call frmSimplexMethod.move_answer_before
resp_data = frmSimplexMethod.find_solution_step

If Val(resp_data) > 0 Then
    frmSimplexMethod.Text1.Text = Val(frmSimplexMethod.Text1.Text) + 1
Else
    If Val(resp_data) = -2 Then 'IT not OK
        If Val(frmSimplexMethod.check_all_x_zero_ornot) > 0 Then
            If frmSimplexMethod.check_answer_before > 0 Then
                Call frmSimplexMethod.show_answer_before1
                countxx = countxx + 1
                Array_Y(0) = Array_Y(0) + 1
                If Array_Y(0) >= 3 Then
                    Array_Y(0) = 0
                    Call change_IT
                End If
                total_loop = total_loop + 1
                sum_total_loop = sum_total_loop + 1
                frmProcess.ProgressBar1.Value = frmProcess.ProgressBar1.Value + 1
            Else:
                Array_Y(0) = Array_Y(0) + 1
                If Array_Y(0) >= 3 Then
                    Array_Y(0) = 0
                    countxx = countxx + 1
                    Call change_IT
                End If
                total_loop = total_loop + 1
                sum_total_loop = sum_total_loop + 1
                frmProcess.ProgressBar1.Value = frmProcess.ProgressBar1.Value + 1
            End If
        End If
    Else
        If Val(resp_data) = -1 Then
            If X_After < Val(frmInputDim.txtDimension.Text) Then
                Call frmSimplexMethod.show_answer_before1
                countxx = countxx + 1
                Array_Y(0) = Array_Y(0) + 1
            End If
        End If
    End If
End If

```

```

        If Array_Y(0) >= 3 Then
            Array_Y(0) = 0
            Call change_IT
        End If
        total_loop = total_loop + 1
        sum_total_loop = sum_total_loop + 1
        frmProcess.ProgressBar1.Value = frmProcess.ProgressBar1.Value + 1
    Else
        Call frmSimplexMethod.show_answer
        countxx = countxx + 1
        Array_Y(0) = Array_Y(0) + 1
        If Array_Y(0) >= 3 Then
            Array_Y(0) = 0
            Call change_IT
        End If
        total_loop = total_loop + 1
        sum_total_loop = sum_total_loop + 1
        frmProcess.ProgressBar1.Value = frmProcess.ProgressBar1.Value + 1
    End If
End If
End If
End If

If total_loop >= Max_next_loop Then
    total_loop = 0
    If Val(Zmax_value) = 0 Then
        If MsgBox("No feasible answer ต้องการทำต่อหรือไม่", vbYesNo +
vbQuestion, "คำถาม") = vbYes Then
            strInput = InputBox("ใส่จำนวน Loop ใหม่ที่ต้องการคำนวณ", "Loop
ปัจจุบัน" & sum_total_loop & "/" & Max_It_change, Max_next_loop)
            If strInput = vbNullString Then
                Exit Sub
            Else
                Max_next_loop = strInput
                frmProcess.Show
            End If
        Else
            Exit Sub
        End If
        MsgBox "No feasible answer at Loop"
    Else
        frmFeasibleSolution.Show
        frmFeasibleSolution.WindowState = 2
        If MsgBox("Found answer at ต้องการทำต่อหรือไม่", vbYesNo + vbQuestion, "คำถาม") =
vbYes Then
            strInput = InputBox("ใส่จำนวน Loop ใหม่ที่ต้องการคำนวณ", "Loop ปัจจุบัน" &
sum_total_loop & "/" & Max_It_change, Max_next_loop)
            If strInput = vbNullString Then
                Exit Sub
            End If
        End If
    End If
End If

```

```

        Else
            Max_next_loop = strInput
            frmProcess.Show
        End If
    Else
        Exit Sub
    End If
End If
End If

Loop

Tstop = Now

If Val(Zmax_value) = 0 Then
    MsgBox "No feasible answer", vbOKOnly, "Not Found Answer"
Else
    MsgBox "Found answer", vbOKOnly, "Found Answer"
End If

frmFeasibleSolution.Show
frmFeasibleSolution.WindowState = 2
frmProcess.Caption = "Time in process = " & Format(Tstart - Tstop, "hh:mm:ss")
frmProcess.Show

End Sub

```

ภาคผนวก ง
บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่



บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่

อรรรรณ นงนุช และ ขงยุทธ เสริมสุธีอนุวัฒน์ (2552). เทคนิคการกระจายค่าความคลาดเคลื่อนที่เหมาะสมเมื่อเดตัมของการบอกขนาดเปลี่ยนแปลง. การประชุมวิชาการช่างงานวิศวกรรมอุตสาหกรรม, 21 – 22 ตุลาคม 2552, มหาวิทยาลัยขอนแก่น, ประเทศไทย.



เทคนิคการกระจายค่าความคลาดเคลื่อนที่เหมาะสมเมื่อเดดไทมของการบอกขนาด เปลี่ยนแปลง

อรรวรรณ นงนุช และ ยงยุทธ เสริมสุขธนวัฒน์
สาขาวิชาวิศวกรรมอุตสาหกรรม คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จ.นครราชสีมา 30000
E-mail: orawan_nongnuch@hotmail.com

บทคัดย่อ

ในงานตัดโลหะ ถ้าเดดไทมของการจับยึดชิ้นงานแตกต่างไปจากเดดไทมของการบอกขนาดจำเป็นจะต้องมีการเปลี่ยนแปลงมิติบอกขนาดใหม่เพื่อให้ครอบคลุมถึงความคลาดเคลื่อนสะสม (tolerance stacks) ที่เกิดขึ้น และมิติชุดใหม่นี้จะต้องยังคงเป้าหมายเดิมของการออกแบบ บทความนี้อธิบายวิธีการใช้กราฟ (tree graph) สร้างความสัมพันธ์ระหว่างมิติ และใช้โปรแกรมเชิงเส้นตรงใน Excel หาคำตอบ และมีตัวอย่างแสดงวิธีการใช้งาน

คำสำคัญ tolerance allocation, tolerance optimization, linear programming

1. บทนำ

ในงานตัดโลหะ การเปลี่ยนแปลงเดดไทมสำหรับบอกขนาดมักเกิดขึ้นเสมอ โดยเฉพาะอย่างยิ่งเมื่อจำเป็นต้องใช้เดดไทมในการผลิตที่แตกต่างไปจากเดดไทมสำหรับบอกขนาดในแบบวิศวกรรมของชิ้นงาน ผลก็คือจำเป็นต้องปรับมิติชุดใหม่ที่อ้างอิงเดดไทมใหม่เพื่อให้ยังคงไว้ซึ่งเป้าหมายของการออกแบบเดิม ปัญหานี้เป็นปัญหาเกี่ยวกับการกระจายหรือการแบ่งความคลาดเคลื่อนให้กับมิติต่าง ๆ (tolerance allocation) ซึ่งในทางปฏิบัติบนพื้นโรงงาน จะอาศัยผู้ที่มีประสบการณ์สูงในงานตัดโลหะทำหน้าที่ตัดสินใจกำหนดเดดไทมในการผลิต และการกระจายความคลาดเคลื่อนให้กับมิติชุดใหม่ ถึงแม้ว่าในปัจจุบัน

เครื่องมือกลที่ใช้ในการผลิตมีความเที่ยงตรงสูงก็ตาม การกระจายค่าความคลาดเคลื่อนยังคงเป็นเรื่องจำเป็น ทั้งในด้าน การออกแบบ, การวางแผนการผลิต, การผลิต, การตรวจสอบ และการประกอบชิ้นงาน ทั้งนี้เพราะงานตัดโลหะชนิดหนึ่งสามารถทำได้บนเครื่องมือกลที่ต่างกัน และการตรวจสอบมิติเดียวกันก็สามารถทำได้หลายวิธี นอกจากนี้ มีชิ้นงานจำนวนมากที่ต้องเปลี่ยนตำแหน่งการจับชิ้นงานหลายครั้งกว่าจะเสร็จสมบูรณ์ เพราะฉะนั้นการวิเคราะห์ความคลาดเคลื่อน (tolerance analysis) ชนิดนี้ จึงยังคงมีความสำคัญต่ออุตสาหกรรมการผลิตโลหะ

การใช้คอมพิวเตอร์วิเคราะห์ความคลาดเคลื่อนอย่างจริงจังมีมานานกว่า 30 ปี ผลงานของ Bjørke ในปี 1978 [1] แสดงให้เห็นถึงความสำเร็จในการพัฒนาโปรแกรมสำหรับคำนวณค่าความคลาดเคลื่อนสะสม (tolerance stacks), และทำ tolerance optimization ในเครื่องคอมพิวเตอร์ขนาดใหญ่ (mainframe) แต่ในเวลานั้นงานวิจัยด้านนี้ยังอยู่ในวงแคบ ซึ่งอาจเป็นเพราะว่าขีดความสามารถที่จำกัดของเครื่องคอมพิวเตอร์เองหรือราคาคอมพิวเตอร์ที่ยังสูงอยู่ บทบาทของคอมพิวเตอร์เริ่มชัดเจนขึ้นในราว 15 ปีที่ผ่านมา เมื่อ Ahaluwalia และ Karolin รายงานการใช้คอมพิวเตอร์สร้าง tolerance chart ในปี 1984 [2] และในเวลาใกล้เคียงกัน Farmer กับ Harris [3] พัฒนาโปรแกรมคอมพิวเตอร์สำหรับคำนวณค่าคลาดเคลื่อนให้กับมิติต่าง ๆ ที่ถูกกำหนดขึ้นเมื่อเกิดการเปลี่ยนแปลงเดดไทมของการบอกขนาดในแบบวิศวกรรม หลังจากบทความทั้งสองนี้ งานวิจัยทางด้าน

การประยุกต์ใช้คอมพิวเตอร์สำหรับวิเคราะห์ความคลาดเคลื่อนก็แพร่หลายไปอย่างกว้างขวาง

สำหรับเรื่องของกระจายค่าความคลาดเคลื่อนนั้น จะต้องมีการสร้างความสัมพันธ์ระหว่างมิติต่าง ๆ ซึ่งในงานวิจัยส่วนใหญ่ยังคงต้องอาศัยประสบการณ์และความชำนาญเฉพาะตน เพื่อกำหนดว่ามีมิติใดบ้างที่เกี่ยวข้องหรือมีความสัมพันธ์กับความต้องการใช้งาน (functional requirement)

อันที่จริง มีวิธีการสร้างความสัมพันธ์ระหว่างมิติของชิ้นส่วนประกอบ (parts assembly) อยู่หลายวิธี เช่น เทคนิคการใช้แผนภาพวงรอบ (loop diagram) ที่กล่าวไว้ในหนังสือของ Gladman [4], Drake [5], และ Liggett [6] แต่วิธีการเหล่านี้ จะใช้สำหรับวิเคราะห์มิติใช้งาน (functional dimensions) ในชิ้นส่วนต่างชิ้นกัน และถึงแม้ว่าจะสามารถนำวิธีการดังกล่าวไปประยุกต์ให้เข้ากับปัญหาการสร้างความสัมพันธ์ของมิติใช้งานในชิ้นส่วนเดียวกันได้ก็ตาม ก็ยังจำเป็นต้องอาศัยกฎเกณฑ์ [7] หรือความชำนาญในการพิจารณา

ส่วนมาก เรื่องความสัมพันธ์ของมิติใช้งานในชิ้นงานเดียวกัน จะถูกกล่าวถึงในบทความที่เกี่ยวข้องกับการสร้างความสัมพันธ์ระหว่างรอยตัด (machining cut) ในการวางแผนกระบวนการผลิต (process planning) หรือใช้ประกอบการสร้าง tolerance charts เช่น Wiell [8]; Irani และคณะ [9]; และ Whybrew และคณะ [10] เป็นต้น ในสองบทความหลังต่างก็ใช้กราฟ (graphs) เป็นเครื่องมือสร้างความสัมพันธ์ระหว่างรอยตัดในแผนกระบวนการผลิต (process plan) แต่ต่างกันว่า Irani ใช้ connected graph ชนิดหนึ่ง ส่วน Whybrew ใช้ tree graph หรือกราฟต้นไม้ ซึ่งเป็นวิธีการที่ง่ายต่อการทำความเข้าใจและไม่ต้องอาศัยกฎเกณฑ์ที่ย่างยากซับซ้อน

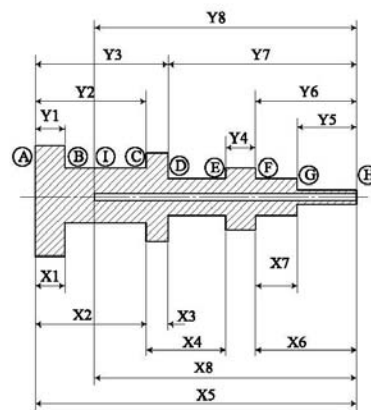
บทความนี้อธิบายการนำวิธีการของ Whybrew ดังกล่าวมาใช้ในการสร้างเงื่อนไขสำหรับการหาคำตอบที่เหมาะสมให้กับการกระจายความคลาดเคลื่อนด้วยเทคนิคโปรแกรมเชิงเส้นตรง (linear programming technique) ที่มีอยู่ใน Excel

2. การสร้างและใช้กราฟต้นไม้

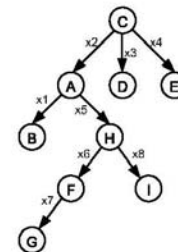
ในรูปที่ 1 แสดงภาพของชิ้นงานกลึง สมมุติว่ามีมิติ Y1 ถึง Y8 เป็นมิติเก่า และมิติ X1 ถึง X8 เป็นมิติชุดใหม่

เมื่อเด็คิมของการบอกขนาดเปลี่ยนแปลง

การสร้างความสัมพันธ์ของความคลาดเคลื่อนของมิติชุดเก่าและมิติชุดใหม่อาศัยการสร้างกราฟต้นไม้สำหรับมิติใหม่เป็นหลักในการกำหนดความสัมพันธ์ โดยมีขั้นตอนดังนี้



รูปที่ 1 แสดงมิติชุดเก่า (Y1,...,Y8) และมิติชุดใหม่ (X1,...,X8)



รูปที่ 2 กราฟต้นไม้สำหรับมิติใหม่

- (1) สเก็ทซ์ภาพฉายตั้งฉากของชิ้นงานที่แสดงมิติซึ่งเกี่ยวข้องกับการเปลี่ยนแปลงเด็คิม และกำหนดชื่อเป็นตัวอักษรให้กับผิวงานที่เป็นขอบเขตของขนาดทั้งหมด (ผิว A, B,.....I) ในวงกลมของรูปที่ 1)

- (2) สร้างกราฟสำหรับมิติใหม่ในรูปที่ 1 โดยการเขียนตัวอักษรของผิวเริ่มต้นไว้ตอนบนสุด (จะเริ่มจากผิวใดของชิ้นงานก็ได้) ในรูปที่ 2 เริ่มจาก C และเรียก C ว่าผิวหรือโหนด (node) C
- (3) ตรวจดูว่ามีมิติอะไรบ้างที่เกี่ยวข้องกับโหนดเริ่มต้นนี้ ให้ลากลูกศรออกจากโหนดเริ่มต้นไปยังอักษรที่ระบุผิวปลายของมิติเหล่านั้น ผลของขั้นตอนนี้ คือแขนง (links) CA, CD, และ CE ของมิติ X2, X3, และ X4 ตามลำดับ และมีโหนดที่เกิดขึ้นใหม่คือ A, D, และ E
- (4) เขียนลูกศรของมิติที่ประกอบด้วยผิวซึ่งเป็นโหนดที่เขียนลงในกราฟก่อนหน้านี้
- (5) ดำเนินขั้นตอนที่ (4) จนครบทุกมิติ จะได้กราฟต้นไม้ของมิติใหม่ที่สมบูรณ์ดังรูปที่ 2

การสร้างสมการแสดงความสัมพันธ์ระหว่างมิติทั้งสองชุด ใช้หลักการที่ว่า ความคลาดเคลื่อนของมิติแต่ละมิติคือความต้องการใช้งานของการออกแบบ (functional requirement) และมิติใหม่ทุกมิติคือ มิติใช้งาน (หรือ function dimension ซึ่งหมายถึงมิติที่มีผลต่อความต้องการใช้งาน) เพราะฉะนั้น ผลรวมของความคลาดเคลื่อนสะสมของมิติใหม่จะต้องไม่มากกว่าความคลาดเคลื่อนของมิติเก่าที่เกี่ยวข้อง

จากหลักการนี้ ทำให้สามารถสร้างสมการแสดงความสัมพันธ์ระหว่างมิติชุดเก่าและมิติชุดใหม่ได้ ตัวอย่างเช่น มิติเก่า Y7 ระหว่างผิว D และ H ในแบบจะเกิดจากเส้นทาง (path) ในกราฟ (รูปที่2) จากโหนด D ถึง H ซึ่งประกอบด้วยมิติ X2, X3, และ X5 ดังนั้นจึงเขียนเป็นสมการแสดงความสัมพันธ์ของความคลาดเคลื่อนของมิติได้เป็น: $x_2 + x_3 + x_5 \leq y_7$ และในทำนองเดียวกันก็สามารถหาสมการอื่น ๆ ได้ดังนี้

$$x_1 \leq y_1$$

$$x_2 \leq y_2$$

$$x_2 + x_3 \leq y_3$$

$$x_2 + x_4 + x_5 + x_6 \leq y_4$$

$$x_6 + x_7 \leq y_5$$

$$x_6 \leq y_6$$

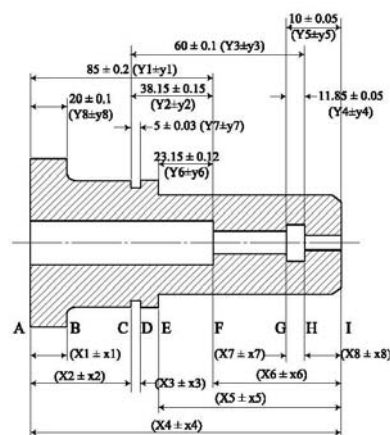
$$x_8 \leq y_8$$

3. ตัวอย่าง

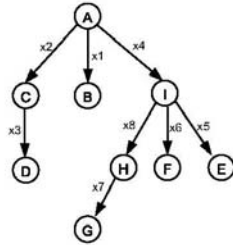
ในรูปที่ 3 เป็นภาพตัดของชิ้นงานหล่ออะลูมิเนียม ซึ่งมีการบอกขนาดความคลาดเคลื่อนของมิติความยาวเป็นค่าเบี่ยงเบนเท่า ๆ กันจากขนาดมูลฐาน (equally bilateral tolerancing) เพื่อให้เข้าใจได้ง่ายขึ้น ในรูปนี้ได้เขียนกำกับมิติเก่าทุกมิติด้วยสัญลักษณ์ของการบอกขนาดเป็น (Y1±y1) ถึง (Y8±y8) ซึ่งเป็นมิติที่อยู่ทางด้านบนทั้งหมดของรูป; สมมติว่ามิติใหม่คือ (X1±x1) ถึง (X8±x8) ซึ่งอยู่ทางด้านล่างทั้งหมด และใช้ตัวอักษร A ถึง I แทนผิวชิ้นงานที่เกี่ยวข้องกับมิติของขนาด

การที่จะกำหนดว่ามิติชุดใหม่ประกอบด้วยมิติใดบ้างจะขึ้นอยู่กับขั้นตอนของการตัดสินใจเป็นสำคัญ แต่ในที่นี้สมมติว่ามิติ X ทั้งหมดที่แสดงในรูปที่ 3 คือมิติชุดใหม่เมื่อเติมเต็มของการบอกขนาดเปลี่ยนแปลง

จากนั้นสร้างกราฟต้นไม้ของมิติชุดใหม่เพื่อกำหนดความสัมพันธ์ของความคลาดเคลื่อนตามขั้นตอนที่ได้อธิบายไว้ในหัวข้อที่ 2 ซึ่งจะได้อกราฟต้นไม้ของมิติใหม่ดังรูปที่ 4



รูปที่ 3 มิติของชิ้นงานตัวอย่าง



รูปที่ 4 กราฟต้นไม้สำหรับมิติใหม่

จากกราฟและมิติเก่าทำให้สามารถ สรุปได้เป็น ตารางที่ 1 ซึ่งแสดงความสัมพันธ์ระหว่างมิติชุดเก่าและ เส้นทางกราฟ (path) ที่เกี่ยวข้องกับมิติชุดใหม่

ตารางที่ 1 ความสัมพันธ์ของความคลาดเคลื่อนของมิติ

มิติเก่า	แขนง	มิติใหม่
y1	AF	x4, x6
y2	CF	x2, x4, x6
y3	CH	x2, x4, x8
y4	GH	x7
y5	GI	x7, x8
y6	EF	x5, x6
y7	CD	x3
y8	AB	x1

และจากนี้ก็สามารถเขียนเป็นปัญหาของโปรแกรมเชิงเส้นตรงได้ดังนี้:-

สมการเป้าหมาย:

$$\text{Max: } x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 \quad (1)$$

สมการข้อจำกัด:

$$x4 + x6 \leq 0.20 \quad (2)$$

$$x2 + x4 + x6 \leq 0.15 \quad (3)$$

$$x2 + x4 + x8 \leq 0.10 \quad (4)$$

$$x7 \leq 0.05 \quad (5)$$

$$x7 + x8 \leq 0.05 \quad (6)$$

$$x5 + x6 \leq 0.12 \quad (7)$$

$$x3 \leq 0.03 \quad (8)$$

$$x1 \leq 0.10 \quad (9)$$

$$x1 \geq 0.027 \quad (10)$$

$$x2 \geq 0.020 \quad (11)$$

$$x3 \geq 0.023 \quad (12)$$

$$x4 \geq 0.014 \quad (13)$$

$$x5 \geq 0.011 \quad (14)$$

$$x6 \geq 0.017 \quad (15)$$

$$x7 \geq 0.009 \quad (16)$$

$$x8 \geq 0.017 \quad (17)$$

สมการที่ (10) ถึง (17) เป็นข้อจำกัดเกี่ยวกับกระบวนการผลิต และในที่นี้ได้กำหนดไว้ว่ากระบวนการผลิตมีความสามารถในการผลิตไม่น้อยกว่า ค่าเกรดความคลาดเคลื่อนสากล (International Tolerance Grade) 8 หรือ IT8

และเมื่อนำชุดสมการข้างบนเป็นข้อมูลป้อนให้กับสมุดงาน Microsoft Excel โดยใช้คำสั่ง Solver ซึ่งเป็นการคำนวณของ Microsoft Excel จะได้ผลลัพธ์เป็นตารางที่ 2 ซึ่งแสดงค่าความคลาดเคลื่อนที่เหมาะสมสำหรับมิติใหม่คือ:-

$$\pm x1 = \pm 0.100; \pm x2 = \pm 0.070; \pm x3 = \pm 0.030;$$

$$\pm x4 = \pm 0.014; \pm x5 = \pm 0.104; \pm x6 = \pm 0.017;$$

$$\pm x7 = \pm 0.034 \text{ และ } \pm x8 = \pm 0.017$$

และมีค่าสูงสุดของฟังก์ชัน = 0.384.

ตารางที่ 2 แสดงการคำนวณหาค่าความคลาดเคลื่อน

Max	X1	X2	X3	X4	X5	X6	X7	X8	1/2Tol
0.384	0.100	0.070	0.030	0.014	0.104	0.017	0.034	0.017	Y _{xy}
				1		1			≤ 0.200
		1		1		1			≤ 0.150
		1		1				1	≤ 0.100
							1		≤ 0.050
							1	1	≤ 0.050
					1	1			≤ 0.120
				1					≤ 0.030
	1								≤ 0.100
	1								≤ 0.027
		1							≤ 0.020
			1						≤ 0.023
				1					≤ 0.014
					1				≤ 0.011
						1			≤ 0.017
							1		≤ 0.009
								1	≤ 0.017

4. สรุป

เทคนิคการใช้กราฟต้นไม้ทำให้ง่ายต่อการสร้างความสัมพันธ์ระหว่างค่าความคลาดเคลื่อนของมิติชุดใหม่กับมิติชุดเก่า เทคนิคชนิดนี้ไม่จำเป็นต้องอาศัยกฎเกณฑ์

มากนัก เพียงแต่สร้างกราฟของมิติชุดใหม่ ก็สามารถอ่านค่ามิติใช้งานต่าง ๆ ที่สัมพันธ์กับความต้องการใช้งานเดิมได้ทันที เพราะฉะนั้นจึงช่วยให้เกิดความรวดเร็วในการกระจายความคลาดเคลื่อน โดยเฉพาะอย่างยิ่งในกรณีที่ต้องทดลองเดิมนของการบอกขนาดหลาย ๆ ชุด เพื่อหาชุดที่เหมาะสมที่สุด และในปัจจุบันเทคนิคการแก้ปัญหาด้วยโปรแกรมเชิงเส้นตรงมีอยู่ในโปรแกรมสมุดงาน (spread-sheet programs) ทั่วไป ทำให้สะดวกต่อการแก้ปัญหาการกระจายค่าความคลาดเคลื่อนเป็นอย่างยิ่ง

บทความนี้เป็นผลของการศึกษาเบื้องต้นของการพัฒนาโปรแกรมคอมพิวเตอร์สำหรับใช้ในการกระจายค่าความคลาดเคลื่อนที่เหมาะสมให้มีมิติต่าง ๆ อย่างอัตโนมัติซึ่งกำลังดำเนินการอยู่ในขณะนี้

เอกสารอ้างอิง

- [1] Øyvind Bjerke, Computer-Aided Tolerancing, Tapir Publishers, 1978.
- [2] Ahluwalia, R.S. and Karolin, A.V., 1984 CATC-A Computer-Aided Tolerance Control System, Journal of Manufacturing System, 3(2), pp.,153-160.
- [3] Farmer, L.E. and Harris, A.G., 1984, Change of Datum of the Dimensions on Engineering Design Drawings, International Machine Tool Design and Research, Vol.24, No.4, pp 267-275.
- [4] Gladman, C.A., Geometric Analysis of Engineering Designs, Australian Trade Publishers , Sydney, 1972.
- [5] Drake, P.Jr., Dimensioning and Tolerancing Handbook, McGraw-Hill, 1999.
- [6] Liggett, J.V., Dimensional Variations Management Handbook: A Guide for Quality, Design, and Manufacturing Engineers, Prentice-Hall, 1993.
- [7] Sermuti-anuwat, Y.,2009, Milling Fixture Design: a Tolerance Analysis Approach (accepted for publication in the International Journal of Mechanical Engineering Education).
- [8] Fainguelemt, D., Wiell, R., and Bourdet, P., 1986, Computer Aided Tolerancing and Dimensioning in Process Planning, Annals of the CIRP Vol.35/1/ 1986.
- [9] Irani, S.A., Mittal, R.O., and Lehtihet, E.A., 1989, Tolerance Chart Optimization, International Journal of Production Research, 26(2).
- [10] Whybrew, K., Britton, G.A., Robinson, D.F., and Sermuti-anuwat, Y., 1990, A Graph-Theoretic Approach to Tolerance Charting, International Journal of Advanced Manufacturing Technology. 5:175-183.

ประวัติผู้เขียน

นางสาวอรรณ นงนุช เกิดเมื่อวันที่ 5 ตุลาคม พ.ศ. 2526 ภูมิลำเนาอยู่ที่บ้านเลขที่ 35/388 หมู่ 6 ตำบลเขาสามยอด อำเภอเมืองลพบุรี จังหวัดลพบุรี เริ่มเข้าศึกษาระดับประถมศึกษาที่ โรงเรียนวรพิทยา ระดับมัธยมต้นที่ โรงเรียนวินิตศึกษาในพระราชูปถัมภ์ฯ ระดับมัธยมปลายที่ โรงเรียนพระนารายณ์ และเริ่มเข้าศึกษาระดับปริญญาตรีที่ สาขาวิชาวิศวกรรมโลหการ สำนักวิชา วิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อปี พ.ศ. 2545 และสำเร็จ การศึกษาเมื่อปี พ.ศ. 2548 หลังจากสำเร็จการศึกษาได้เข้าทำงานกับ บริษัท สหวิริยาเพลทมิล จำกัด จังหวัดชลบุรี เป็นเวลา 1 ปี

จากนั้นได้เข้าศึกษาต่อในสาขาวิชาวิศวกรรมอุตสาหการ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปี พ.ศ. 2550 ในขณะที่ศึกษาต่อในระดับปริญญาโท ได้เป็น ผู้ช่วยสอนรายวิชาปฏิบัติการกรรมวิธีการผลิต ของสาขาวิชาวิศวกรรมอุตสาหการ

ผลงานวิจัย : ได้เสนอบทความเข้าร่วมในการประชุมวิชาการข่ายวิศวกรรมอุตสาหการ ประจำปี 2552 เรื่องเทคนิคการกระจายค่าความคลาดเคลื่อนที่เหมาะสมเมื่อเด็คซ์ของการบอกขนาด เปลี่ยนแปลง

มหาวิทยาลัยเทคโนโลยีสุรนารี