



ปีการศึกษา 2535

การควบคุมระดับของน้ำในท่อโดยใช้อุปกรณ์คอมพิวเตอร์

Computer Control of Water Level in a Stand pipe

โดย

นาย วชิรพงษ์

สาส์สิงห์

นาย เชิดชัย

พุทธทอง

นางสาว สุนทรี

ทองปากน้ำ

อาจารย์ที่ปรึกษา

ผศ.ดร.มงคล มงคลวงศ์โรจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032741

ปริญญานิพนธ์ปีการศึกษา 2535

ภาควิชา วิศวกรรมเครื่องกล

คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง การควบคุมระดับของน้ำในท่อโดยใช้คอมพิวเตอร์

ผู้จัดทำ

- | | | | |
|-----------------|-----------|-------------|--------|
| 1, นายเชิดชัย | พุทธทอง | เลขประจำตัว | 321084 |
| 2. นายวชิรพงษ์ | สาส์สิงห์ | " | 321272 |
| 3. นางสาวสุนทรี | ทองปากน้ำ | " | 321379 |


.....อาจารย์ที่ปรึกษา

(ดร.มงคล มงคลวงศ์โรจน์)

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ เป็นเรื่องเกี่ยวกับการควบคุมระดับน้ำโดยใช้คอมพิวเตอร์ ซึ่งตัวควบคุมที่ใช้ในกระบวนการเป็นแบบ PID และโมเดลอ้างอิง หลักในการทำงานของระบบ คือ ค่าความดันของระดับของเหลว ในท่อพลาสติกใสจะถูกแปลงให้อยู่ในรูปของกระแสไฟฟ้า โดยตัว Pressure Sensor ซึ่งค่ากระแสที่ได้จะอยู่ในช่วง 4 - 20 mA โดยมีช่วงความดันอยู่ในช่วง 1-6 bar จากนั้นกระแสไฟฟ้าที่ได้จะถูกส่งเข้าไปในวงจร i/v เพื่อแปลงค่ากระแสที่ได้ให้อยู่ในรูปของความต่างศักย์ไฟฟ้า ที่อยู่ในช่วง -5 ถึง +5 volt จากค่าความต่างศักย์ที่ได้นี้จะถูกส่งต่อไปยัง PCL - 812 ซึ่งเป็นอุปกรณ์ที่ทำหน้าที่แปลงสัญญาณ analog (ศักย์ไฟฟ้า) ให้เป็นสัญญาณ digital เพื่อส่งเข้าตัวโปรแกรม ให้ทำการโดยตัวควบคุมแบบ PID หรือ IMC แล้วส่งสัญญาณควบคุมออกมาผ่าน PCL - 812 อีกครั้งเพื่อแปลงสัญญาณกลับจากสัญญาณ digital ให้เป็นสัญญาณ analog ที่มีค่าระดับอยู่ระหว่าง 0 ถึง +5 volt แล้วทำการแปลงค่าความต่างศักย์ที่ได้นี้กลับไปอยู่ในรูปของกระแสไฟฟ้าในช่วง 4-20 mA ตามเดิมโดยวงจร v/i ต่อจากนั้นค่ากระแสจะถูกเปลี่ยนเป็นความดันลมที่ป้อนเข้าสู่ Pneumatic Control valve เพื่อทำการควบคุมปริมาณน้ำที่ไหลเข้าท่อพลาสติกใสโดยตัว i/p ซึ่งที่กล่าวมาทั้งหมดนี้เป็นการควบคุมในหนึ่งรอบเท่านั้น

ในการควบคุมแบบ PID ตัวแปรที่มีความสำคัญในการควบคุม คือ ค่า Control Parameter ซึ่งได้แก่ค่า k_p , T_i และ T_d ในการหาค่าเหล่านี้ เราทำได้โดยการอาศัยทฤษฎีของ Ziegler และ Nichol ซึ่งตามทฤษฎีดังกล่าว เราจำเป็นต้องทราบ ค่าคุณลักษณะของระบบอื่น ได้แก่ ค่า Time Constant และ Delay time หรือ Dead time เสียก่อนจึงจะนำไปหาค่า Control Parameter ต่อไปได้

ค่า Control Parameter ที่หาได้จากทฤษฎีของ Ziegler และ Nichol นั้นเป็นเพียงค่าเสนอแนะเริ่มต้นเท่านั้น ดังนั้นในทางปฏิบัติ เราจึงจำเป็นต้องลองทำการทดลองหาค่า Control Parameter เหล่านั้นเพื่อหาค่าที่ดีที่สุดของการควบคุมในระบบ โดยอาศัยค่าที่หาได้จากทฤษฎีของ Ziegler และ Nichol เป็นค่าเริ่มต้น

สำหรับในการควบคุมแบบโมเดลอ้างอิง สมการการควบคุมของตัว Controller หาได้จาก Mathematical model ของระบบ แต่เนื่องจาก ระบบของเราไม่ได้เป็นระบบที่สมบูรณ์แบบ คือ มีสิ่งรบกวนภายนอกเข้ามาอยู่ด้วย ดังนั้น จึงต้องมีตัว filter เข้ามาช่วยให้ระบบของเรามีความเป็นระบบที่สมบูรณ์แบบมากขึ้น โดยที่ Gain ขยายของตัว filter หาได้จากการทดลอง โดยการลองปรับค่าที่ค่าระดับต่าง ๆ กันแล้วพิจารณาค่าที่ดีที่สุดกับระบบของเรา.

ABSTRACT:

This project is the study about Water Level Control in the Stand pipe by Computer. The controller in the process is PID type or Model Reference Control type. The principle of the process work in one - loop control is that pressure of water in the tube which is 1 to 6 bar in range is converted to current in range of 4 to 20 mA according to the pressure. After being converted into current, the current is sent to the i/v circuit in order to transform it into voltage, -5 to +5 volt in range, and then we can get the analog signal. This analog signal is further sent to the PCL-812 which is the device for changing the analog signal to the digital signal. After that the digital signal is moved to the program in the computer for analyzing by the PID or Model Reference Control controller and then the computer sent back the control signal through PCL-812 again for converting the control signal (which is now in the digital signal) into analog signal in range of 0 to +5 volt. The analog signal is then further converted back into current in range of 4-20 mA by v/i circuit. After that, the current from v/i circuit is further changed into pressure of air supplying to regulate the pneumatic control valve for controlling the flowrate of water into the stand pipe.

The most important parameter in PID controller are the control parameter which are K_p , T_d and T_i . The method in estimating these value is based on the theory of Ziegler and Nichol. In such theory, it is necessary to know the characteristic value of the system. These characteristic value are the time constant and delay time. After finding all these value, the control parameter can be estimated by Ziegler and Nichol method.

The control parameter estimated by Ziegler and Nichol method, is just only the first guess value. So that, in practice, tuning for these control parameter becomes an important method to obtain the best value of the control parameter in controlling our process. In that tuning, it must base on the first guess value from Ziegler and Nichol method.

For IMC controller, the equation of the controller is obtained from mathematical model of the process. Since the model is not a perfect model that has an external disturbances, so that it must be important to introduce filter in order that the controller become practical realization. The filter gain is obtained by experiment by tuning from experiment its value and justifying for the best value for the model.

สารบัญ

	หน้า
บทคัดย่อฉบับภาษาไทย	(a)
บทคัดย่อฉบับภาษาอังกฤษ	(b)
บทที่ 1 ทฤษฎี	1
บทที่ 2 อุปกรณ์(Hardware) และการ Calibration	26
บทที่ 3 Program and Flow chart (Software)	60
บทที่ 4 ผลการทดลอง	161
บทที่ 5 บทวิจารณ์ และสรุป	195
กิตติกรรมประกาศ	202
เอกสารอ้างอิง	203

บทที่ 1 ทฤษฎี

Mathematical Model

คือการนำสมการทางคณิตศาสตร์ มาวิเคราะห์ เพื่อแก้ปัญหาในระบบ Dynamic ซึ่งเราจะสามารถสมมุติสมการต่าง ๆ ได้ นั่น ต้องขึ้นอยู่กับ ระบบ และ สภาพแวดล้อม และจะเห็นได้ว่า วิธี Mathematical Model นี้ สามารถใช้ได้ดีกว่าวิธีอื่น ๆ เช่น ปัญหาการควบคุม Optimal, transient - response หรือ frequency - response ที่ใช้กับ single - input, single - output, linear, time - invariant system, ซึ่ง transfer function ของระบบสามารถทำได้โดยใช้วิธีนี้ และเป็นที่น่าสังเกตว่า สมการทางคณิตศาสตร์ สามารถแก้ปัญหาที่มีตัวแปร และ กรณีต้องการคำตอบหลาย ๆ ค่าได้ แต่วิธีนี้เหมาะสำหรับใช้กับคอมพิวเตอร์เท่านั้น

Simplicity versus Accuracy

เราสามารถทำให้ Mathematical Model ดีขึ้นได้โดยการเพิ่มความสัมพันธ์ของตัวแปรเข้าไป วิธีหนึ่งที่ใช้กัน คือ การเพิ่มสมการความสัมพันธ์ของระบบเข้าไป ถึงแม้ว่า accuracy จะไม่ใช่สิ่งสุดท้ายที่ต้องการมากนัก แต่ก็นับว่าเป็นสิ่งที่ดีที่ควรทำในโมเดลตัวอย่าง โดยทั่วไปสิ่งที่เราต้องการคือ การใช้ Mathematical Model มาแก้ปัญหาของระบบที่เรา กำลังพิจารณาอยู่ แต่อย่างไรก็ตาม สิ่งที่เป็นไปได้ที่สุดสำหรับตอนนี้คือ การที่ Mathematical Model สามารถประมาณค่า ของ Dynamic system จริง ๆ เท่านั้นเอง

เพื่อที่จะทำให้การพิจารณาระบบง่ายขึ้น เราจำเป็นต้องตัด Physical property บางตัวทิ้งไป แต่โดยทั่วไป ถ้าหากเป็น lumped parameter เราจำเป็นต้องคงเอาไว้ ส่วนที่เป็น nonlinearities และ distributed parameter เราจะตัดทิ้งไป แต่การที่จะตัด parameter เหล่านี้ทิ้งได้จะต้องเป็น low frequency model แต่ถ้าเป็น high frequency model เหตุการณ์จะกลับกัน

Linear System

ระบบใด ๆ จะถือว่าเป็น linear system ก็ต่อเมื่อ ผลรวมของแรงสองแรงที่ทำในระบบในลักษณะเป็น simultaneous จะเท่ากับผลที่คิดจากการกระทำของแต่ละแรงแยกกัน ดังนั้น ในระบบที่เป็น linear system เราสามารถที่จะทำการทดลองผลของ input ต่าง ๆ ที่ทำกับระบบได้ โดยแยกกันทำทีละครั้ง แล้วเอาผลของ input ทุกตัวมารวมกัน ก็จะได้ผลรวม

(response) ของระบบที่แท้จริง ทำให้เราสามารถที่จะหาผลเฉลยของระบบได้ง่าย

Liquid Level System

ในการวิเคราะห์ที่มีความเกี่ยวข้องกับการไหลของของเหลว จำเป็นอย่างยิ่งที่จะต้องแบ่งแยกชนิดของการไหลว่าเป็นแบบ laminar หรือ turbulent โดยพิจารณาจากค่า Renold number สำหรับในกรณีทั่วไปมักพิจารณาว่า ในการไหลที่มีค่า Renold number มากกว่า 3,000 - 4,000 มักเป็นการไหลแบบ turbulent และในการไหลที่มีค่า Renold number ต่ำกว่า 2,000 เป็นการไหลแบบ laminar ซึ่งในทางปฏิบัติแล้ว เรามักจะพบการไหลที่เป็นแบบ turbulent มากกว่า laminar

ในการวิเคราะห์ แบบจำลองทางคณิตศาสตร์ของระบบควบคุมระดับของเหลวเพื่อเป็นการทำให้ระบบง่ายขึ้น เราจึงนิยามตัวแปรที่นำมาใหม่ 2 ตัว คือ ความต้านทาน (resistance) และความจุ (capacitance)

ความต้านทาน (resistance) พิจารณาการไหลผ่านวาล์วใด ๆ พบว่า ค่าความต้านทาน คือ อัตราการเปลี่ยนแปลงของระดับของเหลว ต่อ อัตราการเปลี่ยนแปลงของอัตราการไหล

$$R = \frac{dH}{dQ}$$

ความจุ (capacitance) ค่าของความจุของถังถูกนิยามเป็น อัตราการเปลี่ยนแปลงปริมาตรของของเหลวที่บรรจุอยู่ในถัง ต่ออัตราการเปลี่ยนแปลงของระดับของเหลว

$$C = \frac{dV}{dQ}$$

ถ้าถังมีลักษณะสมมาตร เช่นรูปทรงกระบอกค่า C จะมีค่าคงที่ซึ่งเท่ากับพื้นที่ก้นถังนั้น พิจารณาระบบควบคุมระดับของเหลวตามที่แสดงดังในรูปที่ 1 ในระบบดังกล่าว ของเหลวจะไหลผ่านวาล์วออกลงสู่ท่อ ถ้าหากการไหลผ่านวาล์วเป็นแบบ laminar ความสัมพันธ์ระหว่าง อัตราการไหลที่สภาวะคงตัว กับ

ค่าระดับความสูงของของเหลวที่สภาวะคงตัว ณ. ค่าระดับความต้านทานหนึ่ง จะเป็นดังต่อไปนี้

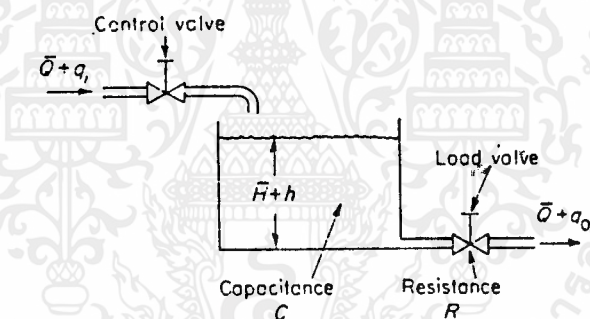
$$Q = KH$$

เมื่อ

Q = อัตราการไหลที่สภาวะคงตัว

K = ค่าสัมประสิทธิ์

H = ค่าระดับของของเหลวที่สภาวะคงตัว



รูปที่ 1 liquid level

และสำหรับการไหลแบบ laminar ค่าความต้านทาน : R , หาได้จาก

$$R_1 = \frac{dH}{dQ} = \frac{H}{Q}$$

จะเห็นว่าในการไหลแบบ laminar ค่าความต้านทานจะมีค่าคงที่ แต่ถ้าในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไหลแบบ turbulent ความสัมพันธ์จะกลายเป็น

$$Q = K\sqrt{H}$$

และค่าความต้านทาน R_L ในการไหลแบบ turbulent จะเป็น

$$R_L = \frac{dH}{dQ}$$

$$\text{จาก } Q = K\sqrt{H}$$

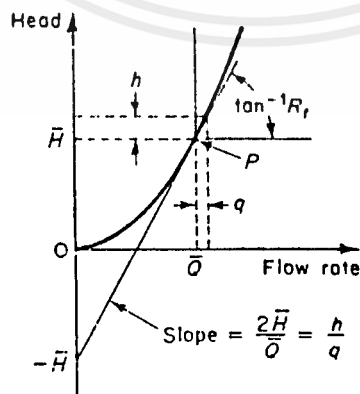
$$dQ = \frac{K}{2\sqrt{H}} dH$$

$$\frac{dH}{dQ} = \frac{2\sqrt{H}}{K} = \frac{2\sqrt{H}}{Q\sqrt{H}} = \frac{2H}{Q}$$

ดังนั้น

$$R_L = \frac{2H}{Q}$$

โดยที่ค่าของ R_L จะไม่คงที่แต่จะขึ้นอยู่กับอัตราการไหลและความสูงของระดับของเหลวที่ตำแหน่งต่าง ๆ แต่ถึงอย่างไรก็ตามเราอาจจะพิจารณาได้ว่าค่า R_L คงที่ได้ก็ต่อเมื่อการเปลี่ยนแปลงของระดับของเหลวและอัตราการไหลมีค่าน้อย ๆ



รูปที่ 2 head versus flowrate curve

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมระดับของเหลว กลับไปพิจารณารูปที่ 1 อีกครั้ง ค่าตัวแปรต่าง ๆ นิยามได้ดังต่อไปนี้

Q = อัตราการไหล ณ จุดคงตัว

q_1 = อัตราการเปลี่ยนแปลงน้อย ๆ ของการไหลเข้าถัง

q_2 = อัตราการเปลี่ยนแปลงน้อย ๆ ของการไหลออกถัง

H = ระดับความสูงของของเหลว ณ จุดคงตัวก่อนที่จะมีการเปลี่ยนแปลงเกิดขึ้น

h = อัตราการเปลี่ยนแปลงน้อย ๆ ของระดับความสูงของของเหลวเนื่องจากการเปลี่ยนแปลงน้อย ๆ ของอัตราการไหลเข้าและออกของของเหลวจากถัง

เนื่องจากอัตราการไหลเข้าลบกับอัตราการไหลออกในขณะช่วงเวลาเล็ก ๆ ใด ๆ จะเท่ากับจำนวน ปริมาตรของของเหลวที่เพิ่มขึ้นในถัง

$$CdH = (q_1 - q_2)dt \dots\dots\dots (1)$$

จากค่านิยามของค่าความต้านทาน

$$q_2 = \frac{h}{R}$$

สมการ (1) กลายเป็น

$$RC \frac{dH}{dt} + h = Rq_1 \dots\dots\dots (2)$$

พิจารณาด้วยว่า RC คือค่า time constant ของระบบ และโดยการใส่ การแปลง Laplace เข้าไปในสมการ (2) โดยให้ค่า initial เป็นศูนย์ ได้เป็น

$$(RCs + 1)H(s) = RQ_1(s)$$

ดังนั้น trasfer function ของระบบคือ

$\frac{H(s)}{Q_1(s)} = \frac{R}{RCs + 1}$

$H(s)$ คือ output ของระบบ

$Q(s)$ คือ input ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Basic Control Action

ในที่นี้จะอธิบายถึงเฉพาะการควบคุมพื้นฐานที่ใช้ในตัวควบคุม (controllers) แบบ analog เท่านั้น ซึ่งสามารถแบ่งออกเป็น 6 ประเภทด้วยกัน คือ

1. ตัวควบคุมแบบ ปิด - เปิด หรือ แบบ 2 ตำแหน่ง
2. ตัวควบคุมแบบ Proportional (P)
3. ตัวควบคุมแบบ Integral (I)
4. ตัวควบคุมแบบ Proportional Integral (PI)
5. ตัวควบคุมแบบ Proportional Derivative (PD)
6. ตัวควบคุมแบบ Proportional Integral and Derivative (PID)

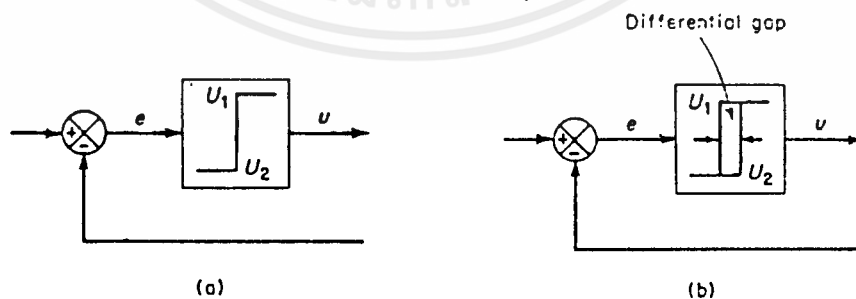
การควบคุมแบบ ปิด - เปิด หรือ two - position

ในการควบคุมชนิดนี้จะมีอุปกรณ์ที่อยู่กับที่ 2 ตำแหน่ง เช่น ในกรณีแบบ ปิด-เปิด และมีราคาถูก สามารถใช้ได้กับงานอุตสาหกรรมหลายชนิด เมื่อเราได้สัญญาณที่ออกจากตัวควบคุมเป็น $U(t)$ และค่าผิดพลาดที่เกิดขึ้นเป็น $e(t)$ ในกรณีการควบคุมแบบ 2 ตำแหน่งนี้ สัญญาณ $U(t)$ จะมีค่าสูงสุด และ ค่าต่ำสุดเท่านั้น และขึ้นอยู่กับค่าผิดพลาด $e(t)$ ว่ามีค่าเป็นบวกหรือลบ จึงได้ว่า

$$U(t) = U_1 \text{ เมื่อ } e(t) > 0$$

$$U(t) = U_0 \text{ เมื่อ } e(t) < 0$$

ค่าของ U_1 และ U_2 เป็นค่าคงที่ ค่าต่ำสุดของ U_2 อาจเป็นได้ทั้ง 0 และ $-U_1$



รูปที่ 3 block diagram of an On-Off Controller and liquid level control system

Proportional Control Action

สมการความสัมพันธ์ระหว่าง $U(t)$ กับ $e(t)$ คือ

$$U(t) = K_p e(t)$$

หรือในรูปของ laplace - transform คือ

$$\frac{U(s)}{E(s)} = K_p$$

เมื่อ K_p คือ proportional gain



รูปที่ 4 block diagram of Proportional Controller

Intigral Control Action

ค่าของสัญญาณจากตัวควบคุม $U(t)$ จะเปลี่ยนแปลงตามค่า $e(t)$ ดังนี้

$$\frac{d[U(t)]}{dt} = K_i e(t)$$

$$U(t) = K_i \int_0^t e(t) dt$$

เมื่อค่า K_i คือ adjustable constant

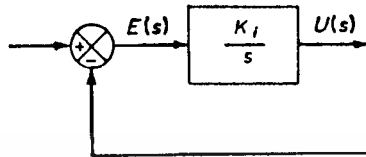
จะได้ transfer function ของระบบควบคุมแบบ Integral คือ

$$\frac{U(s)}{E(s)} = \frac{K_i}{s}$$

ถ้าค่า $e(t)$ เพิ่มขึ้นเป็นสองเท่าแล้ว ค่า $U(t)$ จะเปลี่ยนเป็นสองเท่าอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวดเร็ว ถ้า $e(t) = 0$ จะยังมีค่าเท่าเดิม ในระบบนี้บางครั้งเรียกว่า การควบคุมแบบ reset



รูปที่ 5 block diagram of an Integral Controller

Proportional plus Integral Control Action

สมการความสัมพันธ์ คือ

$$U(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt$$

หรือ Transfer function ของระบบคือ

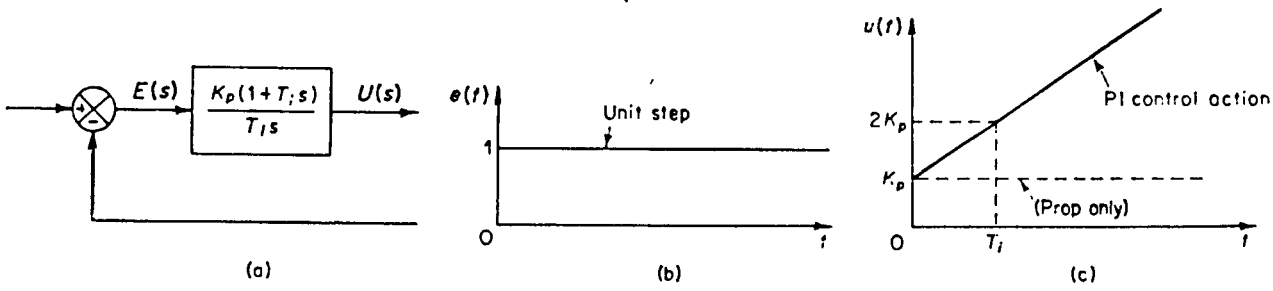
$$\frac{U(s)}{E(s)} = K_p \left[1 + \frac{1}{T_i s} \right]$$

เมื่อ

K_p คือ proportional gain

T_i คือ Integral time

ค่า K_p , T_i สามารถปรับได้ โดยค่า T_i จะมีผลต่อ integral control แต่ค่า K_p มีผลต่อทั้ง Proportional และ Integral control ส่วน กลับของค่า T_i เรียกว่า "reset rate"



รูปที่ 6 block diagram of a Proportional-plus-Integral Controller and diagrams depicting a unit-step input and the controller output.

Proportional plus Derivative Control Action

สมการความสัมพันธ์ คือ

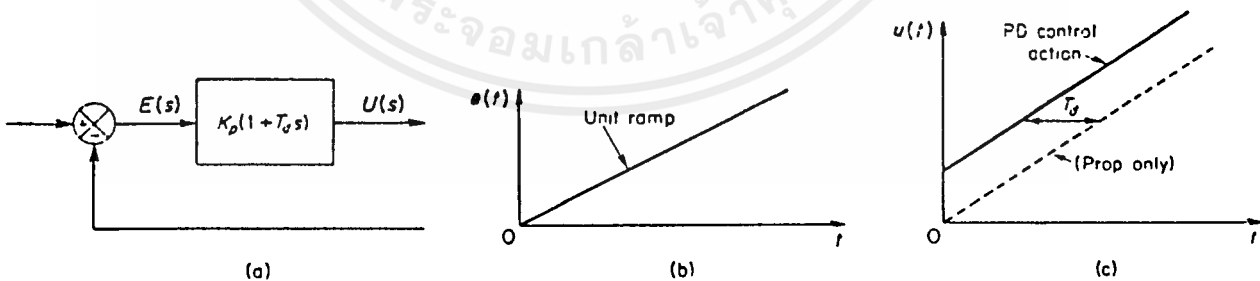
$$U(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

และ transfer function คือ

$$\frac{U(s)}{E(s)} = K_p [1 + T_d s]$$

เมื่อ

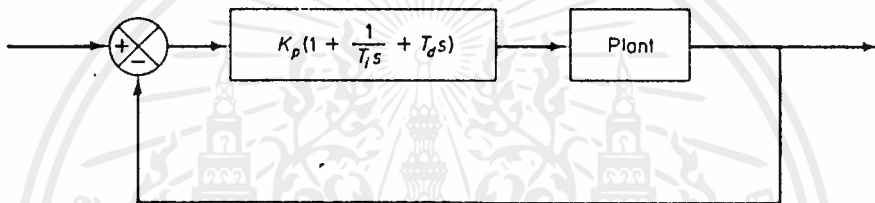
T_d คือ derivation time



รูปที่ 7 block diagram of a Proportional-plus-Derivative controller and diagrams depicting a unit-ramp input and controller output.

PID Control Action

จะเป็นการควบคุมแบบป้อนกลับ (feedback control) โดยทำหน้าที่เปรียบเทียบค่าระดับที่แท้จริง (output) กับค่าที่ตั้งไว้ (set - point value) ความแตกต่างระหว่างค่าที่ตั้งไว้กับค่าที่แท้จริง คือ ค่าที่ผิดพลาด (error) จากนั้น ตัวควบคุมจะส่งสัญญาณควบคุมออกมา เพื่อลด error นั้นลง ถ้าให้ $U(t)$ คือ output ที่ออกมาจากตัวควบคุม และ $e(t)$ คือสัญญาณหาค่าผิดพลาด (error signal) เราสามารถเขียน Block diagram และระบบการควบคุมระดับของเหลวโดยใช้ตัวควบคุมได้โดย



รูปที่ 8 PID controller of a plant

การควบคุมแบบPID สมการทางคณิตศาสตร์ของตัวควบคุมแบบ PID คือ

$$U(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}$$

ซึ่ง Transfer function ของระบบ คือ

$$\frac{U(s)}{E(s)} = K_p \left[1 + \frac{1}{T_i s} + T_d s \right]$$

K_p = proportional gain

T_i = Integral time

T_d = derivativse time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ziegler และ Nichol method Ziegler และ Nichol ได้ทำการศึกษา ค้นคว้าวิธีการในการประมาณค่า parameter ของตัวควบคุมต่าง ๆ ดังที่ได้กล่าวไปแล้วอันได้แก่ Proportional Control time: K_p , Integral time: T_i และ Derivative time : T_d การวิเคราะห์ค่า parameter ต่าง ๆ เหล่านี้ ตามวิธีของ Ziegler และ Nichol อาศัยการสังเกตคุณลักษณะเฉพาะของผลตอบสนองแบบชั่วขณะ (Transient Response) ของระบบโดยวิธีการวิเคราะห์ค่า parameter ดังกล่าวสามารถหาได้จากการทดลองเท่านั้น

Ziegler และ Nichol ได้เสนอวิธีในการทดลองหาค่า controller parameter ทั้งหมด 2 วิธี ด้วยกัน คือ

1. วิธีแรก หาได้โดยการทำการทดลองหาผลตอบสนองของระบบต่อ input แบบ unit step ซึ่งถ้าหาก Transfer function ของระบบ ไม่มีลักษณะของ integrator(s) หรือ Dominant complex- conjugate pole แล้ว ผลตอบสนองต่อ input แบบ unit step จะมีลักษณะเป็นคล้าย ๆ S - curve ดังแสดงในรูป

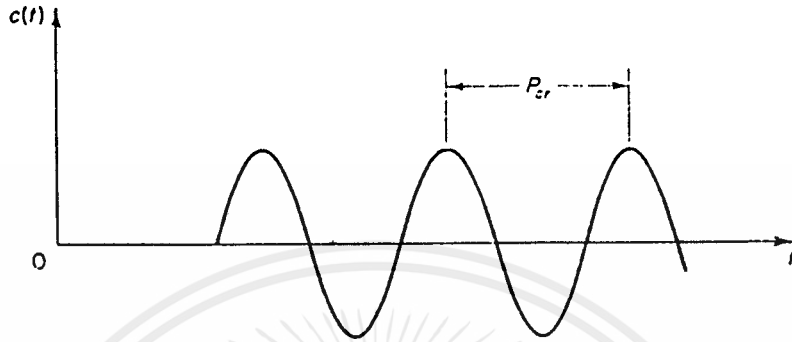
S - curve ที่ได้จะประกอบไปด้วยตัวแปรที่สำคัญ 2 ตัว คือ delay time : L และ Time constant โดยค่าของ delay time และ time constant หาได้จากการลากเส้นตรง ณ. จุดเปลี่ยนเว้าของกราฟ และพิจารณาจุดตัดของเส้นสัมผัสกราฟ กับแกนเวลา และเส้น $c(t) = k$ ดังแสดงในรูป

หลังจากทราบค่า delay time และ time constant แล้วเราสามารถหาค่า controller parameter ได้จาก table ข้างล่าง ซึ่งคิดขึ้นโดย Ziegler และ Nichol

Ziegler - Nichol Tuning Rule Based on Step Response of Plant (First method)

Type of controller	K_p	T_i	T_d
P	T/L	-	-
PI	$0.9T/L$	$L/0.3$	-
PID	$1.2T/L$	$2L$	$0.5L$

2. วิธีที่สอง Ziegler และ Nichol ได้แนะนำวิธีไว้ว่าให้ set ค่า $T_i = T_d = 0$ ก่อนโดยให้ใช้แต่เฉพาะค่า proportional gain : K_p จากนั้นให้ทำการเพิ่มค่า K_p ขึ้นเรื่อย ๆ ตั้งแต่ศูนย์จนถึงค่าวิกฤต (K_{cr}) ซึ่ง ณ.ค่าวิกฤตนี้ output ของระบบจะเกิดการสั่นโดยไม่หยุดดังรูป



รูปที่ 9 Sustained oscillation with period P_{cr}
 ณ.ค่าวิกฤตของ K_{cr} อ่านค่าวิกฤตของ P_{cr} ได้ เมื่อทราบค่าทั้งสองค่าแล้ว
 นำไปคำนวณหาค่าcontroller parameter ได้ตามตาราง
 การประมาณค่าแบบ Ziegler-Nichol เป็นที่นิยมใช้กันอย่างแพร่
 หลายในการหาค่า PID Control Parameter ซึ่งจะเป็นประโยชน์อย่างยิ่ง
 ในระบบที่เราไม่อาจทราบ Transfer function ได้อย่างแน่ชัด แต่ถึงกระ
 นั้นก็ตามการประมาณค่าแบบ Ziegler-Nichol ก็สามารถที่จะนำไปใช้กับ
 ระบบที่รู้ Transfer function ได้ด้วย

Ziegler - Nichol Tuning Rule Based on
Step Response of Plant (Second method)

Type of controller	K_p	T_i	T_d
P	$0.5K_{cr}$	-	-
PI	$0.45K_{cr}$	$P_{cr}/1.2$	-
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

The PID algorithm

ตัวควบคุมระบบแบบ PID สามารถแสดงในรูปของสมการได้เป็น

$$G_c(s) = \frac{M(s)}{E(s)} = k_c \left(1 + \frac{1}{T_i s} + T_d s \right)$$

โดย k_c = Controller gain
 T_i = Integral action time
 T_d = derivative

ตัวควบคุมที่แสดงดังในสมการข้างบนดังกล่าว เป็นลักษณะตัวควบคุมในอุดมคติ คือ แต่ละเทอมในสมการจะไม่ส่งผลกระทบซึ่งกันและกันหรือแต่ละเทอมแยกเป็นอิสระต่างหากจากกันและกัน

ตัวควบคุมแบบ PID ชนิดที่เป็นอุดมคตินี้จะไม่สามารถใช้ได้ทางปฏิบัติ เมื่อเราทำการควบคุมระบบด้วยคอมพิวเตอร์ โดยที่ Transfer function ของการควบคุมแบบ PID โดยใช้คอมพิวเตอร์ (หรืออุปกรณ์ analog) จะอยู่ในรูปของ

$$G_c(s) = \frac{M(s)}{E(s)} = \frac{k_c (1/T_i s) (T_d s + 1)}{T_i s (1 + \alpha T_d s)}$$

โดยที่ T_i = Integral time
 T_d = Derivative action time
 α = ค่าคงที่ซึ่งจะมีค่าอยู่ในช่วง 1/6 ถึง 1/20

แต่อย่างไรก็ตามตัวควบคุมชนิด PID แบบในทางอุดมคติก็นำมาซึ่งการเกิด roll-off ซึ่งโดยทั่วไปแล้ว มักจะนำมาใช้ร่วมกับ filter โดยที่ตัว filter นี้ จะช่วยทำให้ปรากฏการณ์ที่เรียกว่า roll-off เมื่อใช้กับความถี่สูง ๆ ซึ่งจะเกิดขึ้นกับเทอมของ derivative action ในสมการการควบคุมดังกล่าว การเกิด roll-off ขึ้นนี้สามารถทำได้โดย

(1) ทำการ filter สัญญาณ input ก่อนเข้า PID แบบอุดมคติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- (2) ทำการ filter สัญญาณ output ที่ออกมาจากตัวควบคุมแบบ PID ในทางอุดมคติ
- (3) ใช้ตัวควบคุมที่มีองค์ประกอบของ roll - off เช่น ใช้ตัวควบคุมที่มีองค์ประกอบแบบ lead - lag มากกว่าที่จะใช้แบบ derivative action ทั้งหมด

การนำเอาตัวควบคุมแบบ PID ในทางอุดมคติมาใช้งานในทางปฏิบัติ (Implementing the Ideal PID Controller)

จากสมการการควบคุมแบบ PID ในทางอุดมคติ

$$G_c(s) = \frac{M(s)}{E(s)} = k_c \left(1 + \frac{1}{T_i s} + T_d s \right)$$

สามารถเขียนให้อยู่ในรูปของ Time domain ได้เป็น (ใส่ Inverse Laplace เข้าไป)

$$m(t) = k_c T_d \frac{de(t)}{dt} + k_c e(t) + \frac{k_c}{T_i} \int e(t) dt$$

เพื่อที่จะนำสมการข้างบนดังกล่าวไปใช้กับการควบคุมที่เป็น analog ดังนั้น จึงจำเป็นต้องแปลงสมการดังกล่าวให้อยู่ในรูป Discrete form โดยการใช้ finite difference อันดับ 1

$$\left. \frac{df}{dt} \right|_k = \frac{f_k - f_{k-1}}{\Delta t}$$

$$\int_0^n e(t) dt = \sum_{k=c}^n e_k \Delta t$$

ดังนั้นสมการดังกล่าวสามารถเขียนได้เป็น

$$m_n = k_c \left(T_d \frac{e_n - e_{n-1}}{\Delta t} \right) + e_n + \frac{1}{T_i} \sum_{k=0}^n e_k \Delta t$$

โดยการเปลี่ยนตัวแปรใหม่

$$k_p = k_c$$

$$k_i = k_c T_i / T_s$$

$$k_d = k_c T_d / T_s$$

ซึ่ง $T_s = \Delta t$ = ช่วงเวลาในการสุ่ม (Sampling interval)
 ดังนั้นสมการการควบคุมแบบ PID จึงเขียนใหม่ได้เป็น

$$m_n = k_p * e_n + k_i s_n + k_d (e_n - e_{n-1})$$

โดยที่ $s_n = s_{n-1} + e_n$ คือเป็นผลรวมบางค่าผิดพลาด

การควบคุมแบบ PID อย่างง่าย ๆ โดยใช้แนวคิดของสมการดังกล่าวข้างต้น สามารถที่จะเขียนเป็น Program ในภาษา PASCAL ได้ดังนี้

Ex 4.1 PID controller written in Pascal

PROGRAM PIDCroller;

CONST

kpvalue = 1.0;

kivalue = 0.8

kdvalue = 0.8;

VAR

s, kp, ki, ke, en, enoln, mn : REAL;

stop : BOOLEAN;

FUNCTION ADC:REAL;EXTERNAL;

BEGIN (* main program *)

stop := FALSE;

s := 0.0;

kp := kpvalue;

ki := kivalue;

kd := kdvalue;

enoil := ADC;

(* control loop *)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REPEAT

en : = ADC; (* ADC returns value of error en *)

s : = s+en; (* integral summation *)

mn : = kp*en + ki*s + kd*(en-enold);

DAC (mn);

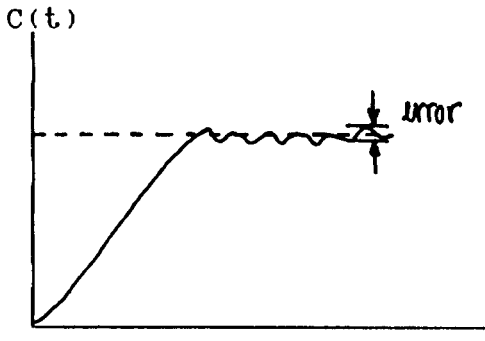
enold : = en;

UNTIL stop;

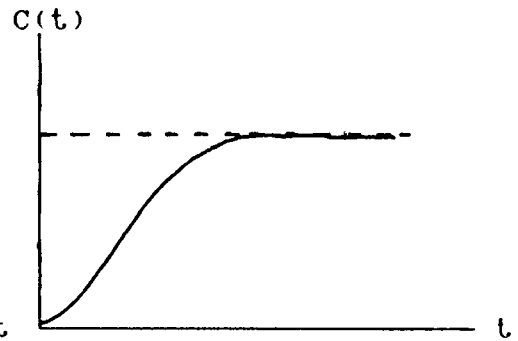
END.

จะเห็นได้ว่า โปรแกรมที่แสดงไว้ข้างต้นนั้น เป็นโปรแกรมที่ไม่ถูกต้องและไม่สามารถนำไปใช้ได้ทางปฏิบัติ ที่กล่าวว่าจะไม่ถูกต้องก็เพราะว่าโปรแกรมหดงกล่าวไม่ได้คำนึงถึงการที่จะต้องคำนวณค่าของตัวแปรที่ปรากฏในตัวโปรแกรมไปพร้อม ๆ กันทีเดียว จากโปรแกรมจะเห็นได้ว่า ค่าของช่วงเวลาในการสุ่ม T_s (Sample interval or Δt) จะขึ้นอยู่กับความเร็วในการคำนวณของตัวคอมพิวเตอร์เอง โดยจะมีค่าแตกต่างกันไปขึ้นอยู่กับชนิดของเครื่องที่ใช้ ดังนั้นเพื่อที่จะแก้ไขโปรแกรมให้ถูกต้องจำเป็นต้องที่จะต้องกำหนดช่วงเวลาในการสุ่มขึ้นมา

สำหรับที่กล่าวว่าจะไม่สามารถนำไปใช้ได้ทางปฏิบัติได้ก็เพราะรูปแบบของการควบคุมแบบ PID ที่ใช้ในตัวโปรแกรมนี้ไม่ได้มีการคำนึงถึงข้อจำกัดในระบบตามความเป็นจริงหรือระบบไม่ได้เป็น model ตามอุดมคติอย่างสมบูรณ์แบบนั้นเอง ดังนั้นถ้านำโปรแกรมดังตัวอย่าง ไปใช้งานจริงแล้วจะก่อให้เกิดปัญหาค่าของ integral จะมีค่ามากขึ้นเรื่อย ๆ เพราะในระบบตามจริงแล้ว ณ จุดที่ขอบรับได้ว่าเป็นจุดเสถียรของระบบ จะมี error เกิดขึ้นอยู่ โดยที่ค่า error นี้จะมีค่าเป็นศูนย์ ที่จุดเสถียรของระบบ ถ้าหากเป็น model ตามอุดมคติ ดังนั้นในทางปฏิบัติ ณ จุดเสถียรค่าของ integral ในตัวควบคุมจะไม่คงที่แต่จะบวกทบขึ้นไปเรื่อย ๆ ซึ่งจะทำให้เกิด Integral wind - up ได้ในที่สุด นอกจากนี้ตัวโปรแกรมยังจะต้องคำนึงถึงสิ่งรบกวน (noise) ในระบบอีกด้วย ดังนั้นโปรแกรมที่ใช้ได้ตามความเป็นจริงจึงจะต้องมีการใส่ค่าคงที่ค่าหนึ่งเข้าไปด้วย ซึ่งจะกล่าวถึงโดยละเอียดภายหลัง



โมเดลตามความจริง



โมเดลในอุดมคติ

พิจารณา

$$S_n = S_{n-1} + e_n$$

ไม่คงที่ ณ จุด steady state

เพราะ e_n ไม่เท่ากับ 0

$$S_n = S_{n-1} + e_n$$

จะคงที่ ณ จุด steady state

เพราะ e_n เท่ากับ 0

TIMING

สิ่งสำคัญในโปรแกรมควบคุมระบบที่มีลักษณะเป็นแบบ real time นั่นคือ จะต้องทำงาน ไปอย่างต่อเนื่องดังนั้นลักษณะโดยทั่วไปของโปรแกรมจะมีโครงสร้างแบบการวนหลูปไม่รู้จบ (infinite loop) เขียนออกมาเป็น algorithm ได้เป็น

EXAMPLE 4.3 General form of a control program

PROGRAM Real TimeControl ;

(* declarations *)

BEGIN

LOOP

Control Task;

END (* loop *);

END Real TimeControl.

สิ่งสำคัญภายใน Control task จะต้องประกอบไปด้วย ช่วงเวลาที่โปรแกรมใช้สุมข้อมูลจาก model เข้ามาทำการวิเคราะห์ ซึ่งวิธีการสุมค่าข้อมูลของโปรแกรมสามารถกระทำได้หลายวิธีดังต่อไปนี้

- polling
- external interrupt.
- ballast coding
- real time clock signals

สำหรับในสองวิธีแรกนั้น (polling and external interrupt) จะอาศัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งสัญญาณมาจากกระบวนการ เมื่อมันพร้อมที่จะรับการควบคุมแล้ว โดยสัญญาณดังกล่าวนี้ จะถูกกำหนด ให้ส่งมาด้วยอัตราเท่ากับ ช่วงเวลาในการสุ่ม (Sampling interval, T_s) โดยการกำหนดขนาดของช่วงเวลาในการสุ่ม และถูกกำหนดขึ้นในตอนต้นของกระบวนการควบคุม และการควบคุม จะมีความถูกต้องมากขึ้นเพียงใดก็ขึ้นอยู่กับกาหนดช่วงเวลาในการสุ่ม

สำหรับความแตกต่างของสองวิธีนี้คือ polling จะอาศัย interrupt จากภายในตัว computer แต่ external interrupt สัญญาณจะมาจากภายนอกระบบ

EXAMPLE 4.4 Use of polling for timing

PROCEDURE ControlTask;

BEGIN

LOOP

WHILE NOT(Digin(SampleTime)) DO

(* wait until time *)

END (* while *);

(* control action *)

END (* loop *);

END ControlTask;

จะเห็นได้ว่าวิธี polling เป็นวิธีที่จำกัดเฉพาะระบบการควบคุมที่เล็ก ๆ ไม่สลับซับซ้อน แต่วิธีนี้มีข้อได้เปรียบในความง่ายและสามารถนำมาประยุกต์ใช้ได้กับตัวโปรแกรมได้ทันทีเลย

สำหรับวิธี Ballst Coding และ Real Time Clock Signal จะไม่กล่าวถึงเพราะ ในการทำโครงการนี้ ไม่ได้ใช้วิธีดังกล่าว สำหรับผู้ที่สนใจสามารถหาศึกษาอ่านได้จากเอกสารอ้างอิงหมายเลข

การควบคุมระบบ PID ในทางปฏิบัติ

ในการใช้งานจริงระบบการควบคุมที่ได้กล่าวมาแล้วนั้นไม่สามารถทำงานได้ถูกต้องตามความต้องการได้เพราะในระบบจริงจะมีสิ่งรบกวนอยู่มาก(noise) ซึ่งเราสามารถทำการปรับปรุงระบบการควบคุมระบบได้โดยวิธีการดังต่อไปนี้

- bumpless transfer
- saturation (integral wind-up)
- noise
- modified integral and derivative calculation

bumpless transfer

จากสมการ

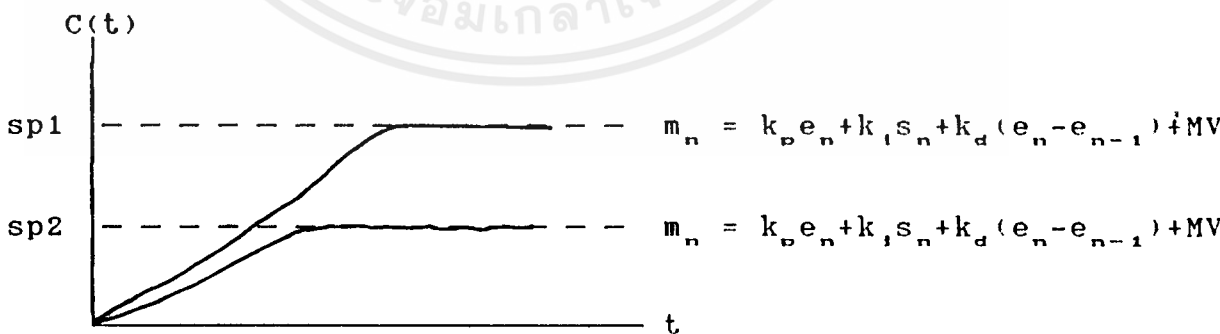
$$G_c(s) = M(s)/E(s) = k_c (1 + 1/T_i s + T_d s)$$

หมายความว่าที่สภาวะ steady state จะไม่มี error เกิดขึ้นค่า Controlled variable จะมีค่าเท่ากับศูนย์ คือค่า M เท่ากับศูนย์นั่นเอง แต่ในความเป็นจริงไม่ได้เป็นเช่นนั้น ดังนั้นในทางปฏิบัติโดยทั่วไป จึงทำการปรับปรุงสมการดังกล่าวโดยการบวกค่าคงที่ค่าหนึ่งเข้าไป (MV) ซึ่งเราเรียกว่า Manipulated Variable ดังนั้น สมการการควบคุมดังกล่าวจึงกลายเป็น

$$m_n = k_p e_n + k_i s_n + k_d (e_n - e_{n-1}) + MV$$

ค่า MV นี้ เปรียบเทียบได้กับการเป็นค่า Setting ของระบบ ถ้าหากในระบบใดที่ไม่มีค่า MV จะทำให้เกิดค่า Integral action ขึ้น ซึ่งค่า Integral นี้จะทำการชดเชยค่า MV ที่ถูกละทิ้งไป แต่มันเป็นการยากที่จะทำให้ระบบเกิดความราบเรียบที่สภาวะ steady state ได้ โดยปราศจากสิ่งรบกวนต่อใดเดลที่เราทำการควบคุมอยู่

ปัญหาในการพิจารณาว่าเราจะ set ค่า MV เป็นเท่าใด ในระบบของนั้นเป็นสิ่งที่ยากมากซึ่งค่า MV นี้จะต้องหามาจากวิธีลองผิดลองถูก (Trial and error) โดยการลองแทนค่า MV ค่าต่าง ๆ กัน ลงสมการการควบคุม แล้วลองเปลี่ยนค่าไปเรื่อย ๆ ซึ่งค่าที่หามาได้นี้ จะใช้ได้เฉพาะกับที่สภาวะ Steady state หนึ่ง ๆ เท่านั้น ถ้าหากเราเปลี่ยนจุด Steady state ของระบบไป ค่า MV ก็จะต้องเปลี่ยนไปด้วย



saturation

เป็นวิธีที่ใช้แก้ปัญหาการเกิด Integral wind-up ในระบบการควบคุม วิธีดังกล่าวนี้สามารถแบ่งแยกได้อีกหลายวิธีดังต่อไปนี้

- Fixed limit on integral term
- Stop summation on saturation
- Integral subtraction

Fixed limit on integral term

ทำโดยการกำหนดค่าต่ำสุดและมากที่สุดของค่า Integral summation เมื่อค่า Integral summation ในระบบมีมากกว่าค่าที่ตั้งไว้มันก็จะถูก reset กลับไปที่ค่าที่ตั้งไว้ ณ ค่ามากหรือน้อยสุดแล้วแต่ความเหมาะสม ตามตัวอย่างต่อไปนี้

EXAMPLE 4.6

REPEAT

```
en := ADC; (* ADC returns value of error en *)
s := s+en; (* integral summation *)
IF s>smax THEN s:=smin
ELSE IF s<smin THEN s:=smin;
mn := kp*en + ki*s + kd*(en-enod);
DAC(mn);
```

```
enold := en;
```

```
UNTIL stop;
```

Stop Summation

ในวิธีนี้ค่าของ Integral Summation จะถูกทำการ "แช่แข็ง" ไว้ เมื่อตัว Control actuator อยู่ในสถานะ Saturation โดยวิธีการดังกล่าวค่า Integral Summation จะคงมีค่าคงที่ในระหว่างที่ตัว Control actuator อยู่ในสถานะ Saturation

สำหรับวิธี noise และ modified integral and derivative calculation จะไม่กล่าวถึงเพราะ ไม่ได้ใช้ในโครงงานนี้ สำหรับผู้ที่สนใจหาศึกษาอ่านได้จากเอกสารอ้างอิง หมายเลข ๔๕

Model Reference Control

ทฤษฎีระบบควบคุมอัตโนมัติ มีบทบาทสำคัญเป็นอย่างมากต่อวงการอุตสาหกรรม โดยเฉพาะอย่างยิ่งในยุคที่คอมพิวเตอร์ได้ถูกนำมาใช้อย่างแพร่หลาย ทฤษฎีระบบควบคุมอัตโนมัติปัจจุบันเมื่ออยู่หลายทฤษฎีด้วยกัน อาทิเช่น Optimal Control , Smith Predictor, Inferential Control, Model Algorithmic Control , Dynamics Matrix Control และ IMC ระบบควบคุมอัตโนมัติ โดยทั่วไปนิยมใช้ระบบควบคุมแบบป้อนกลับ แทนที่ตัดสินใจในการวัดคุณภาพ ของตัวควบคุมสามารถพิจารณาได้ดังนี้ ประการแรก ความสามารถในการรักษาให้ output หรือ ที่ต้องการควบคุมไม่เปลี่ยนแปลง เมื่อมีสิ่งรบกวนที่สามารถวัดได้ หรือ สิ่งรบกวนที่ไม่สามารถวัดได้

(Measureable and Unmeasureable disturbance) กระทบต่อระบบหรือขบวนการนั้น ๆ ประการที่สอง เมื่อมีการเปลี่ยนค่า setpoint output จะต้องเปลี่ยนตามได้อย่างเรียบและรวดเร็ว ประการที่สาม ความเสถียร และสมรรถนะของระบบจะต้องอยู่ในเกณฑ์ที่ยอมรับได้

บทความนี้จะได้กล่าวถึงทฤษฎี Internal Model Control (IMC) ซึ่งได้ถูกพัฒนามาตั้งแต่ปี ค.ศ 1982 โดย Gacia, C.E. และ Morari ,M.[1] สามารถเขียนเป็น Block diagram ในรูปของ Laplace ตามรูปข้างล่าง

$$m(s) = \frac{g_c(s)}{(1 + g_c(s)[g(s) - \tilde{g}(s)])} [y_u(s) - d(s)] \quad \dots (1)$$

$$y(s) = \frac{g(s)g_c(s)}{(1 + g_c(s)[g(s) - \tilde{g}(s)])} [y_u(s) - d(s)] + d(s) \dots (2)$$

$m(s)$ = สัญญาณควบคุม (Manipulated Variable)

$y(s)$ = output หรือ ตัวแปรที่ต้องการควบคุม

$d(s)$ = สิ่งรบกวน (disturbance)

$g(s)$ = transfer function ของระบบ

$\tilde{g}(s)$ = โมเดลคณิตศาสตร์ของระบบ

$g_c(s)$ = ตัวควบคุมระบบ (controller)

$y_u(s)$ = ค่าที่ตั้ง (set point)

คุณสมบัติของ Model Reference Control

พิจารณาถึงผลที่ได้จากสมการที่ (2) เป็นดังนี้

(1) สำหรับ Perfect Model : $\tilde{g}(s) = g(s)$ แทนค่าลงในสมการที่ (2) จะได้

$$y(s) = g(s)g_c(s)[y_u(s) - d(s)] + d(s) \dots \dots \dots (3)$$

สมการที่ (3) แสดงให้เห็นว่าระบบควบคุมแบบโอเนกัลบจะเสถียรก็ต่อเมื่อ $g(s) \cdot g_c(s)$ เสถียร

(2) เมื่อเลือกให้ $g_c(s) = \tilde{g}^{-1}(s)$ แล้วแทนในสมการที่ (2) ก็จะได้ $y(t) = y_u(t) \dots \dots \dots (4)$

สมการที่ (4) แสดงให้เห็นว่าเป็นกรณี Infinte gain

(3) เมื่อสมมติ $\lim_{t \rightarrow \infty} d(t) = d(t)_{\text{คงที่}} =$ ค่าคงที่

แล้วเลือก $\lim_{t \rightarrow \infty} y_u(t) = y_u =$ ค่าคงที่

และ $g_c(0) = \tilde{g}^{-1}(0)$
(15)

โดยอาศัย Final Value Theorem จะได้

$$\lim_{t \rightarrow \infty} y(t) = y_u \dots \dots \dots (5)$$

สมการที่ (5) แสดงให้เห็นว่าไม่เกิดค่าผิดพลาด (offset) .

การออกแบบตัวควบคุมของระบบควบคุมด้วย Model Reference Control

จากคุณสมบัติของ Model Reference Control ทั้งสามข้อในสมการที่ (3) (4) และ (5) แสดงให้เห็นว่าสามารถออกแบบตัวควบคุมโดยการแพดเตอร์โมเดลทางคณิตศาสตร์ออกเป็นสองส่วน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\tilde{g}(s) = g_+^{\sim}(s) \cdot g_-^{\sim}(s) \dots \dots \dots (6)$$

โดยที่ $\tilde{g}(s)$ = โมเดลทางคณิตศาสตร์ของระบบ
 $g_+^{\sim}(s)$ = โมเดลคณิตศาสตร์ซึ่งประกอบด้วย Time delay ทวีคูณส่วนที่ไม่เสถียร
 $g_-^{\sim}(s)$ = โมเดลทางคณิตศาสตร์ที่สามารถเป็นจริงได้ และเสถียร

เพราะฉะนั้น ตัวควบคุมของระบบควบคุมด้วย Model Reference Control คือ

$$g(s) = g_-^{\sim -1}(s) \dots \dots \dots (7)$$

เพื่อให้เป็นไปตามสมการที่ (4) คือ $g_c(0) = g_-^{\sim -1}(0)$ ดังนั้นจำเป็นต้อง

ให้ $|g_+^{\sim}(0)| = 1$ และเนื่องจากว่าในทางปฏิบัติ ไม่สามารถได้โมเดลคณิตศาสตร์ ที่สมบูรณ์เหมือนระบบจริง (Perfect Model) ดังนั้น

$$g(s) = \tilde{g}(s)[1 + L(s)] \dots \dots \dots (8)$$

และ $|L(i\omega)| \leq 1(\omega)$

โดยที่ $L(s)$ = Uncertainty ของระบบคณิตศาสตร์ (16)

$$\text{ให้ } g_c(s) = g_-^{\sim -1}(s) \cdot h(s) \dots \dots \dots (9)$$

โดยที่ $h(s)$ = Filter ทำหน้าที่ช่วยเมื่อโมเดลคณิตศาสตร์ไม่ perfect

ในระบบควบคุมอัตโนมัติที่ควบคุมด้วย Model Reference Control ทางปฏิบัติได้แสดงในรูป ซึ่งเป็น Block diagram เมื่อโมเดลคณิตศาสตร์ไม่ perfect จากรูป ดังกล่าวพอสรุปได้ว่า ระบบควบคุมแบบป้อนกลับจะเสถียรก็ต่อเมื่อ

$$|h(i\omega) \cdot g_+^{\sim}(i\omega) \cdot L(i\omega)| < 1 \dots \dots \dots (10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก $\left| \tilde{g}_+(iw) \right| = 1$ จากสมการที่ (10) จะได้ขนาดของ Filter เป็นดังนี้

$$\left| h(iw) \right| < 1^{-1}(w) \dots \dots \dots (11)$$

การออกแบบระบบควบคุมด้วย Model Reference Control ขนาดของ Filter ในทางปฏิบัติจะให้ค่าดังนี้

$$\left| h(iw) \right| = [1(1+\beta_r) + \beta_r]^{-1} \dots \dots \dots (12)$$

โดยที่ β_r เป็นค่าเสมือน safety factor ซึ่งมากกว่าหรือเท่ากับ ประมาณ -3db หรือเท่ากับ $1/\sqrt{2}$

การควบคุมระบบด้วย Model Reference Control

สัญญาณควบคุมที่ส่งจากตัวควบคุมจะเขียนอยู่ในรูปลาปลาซ ได้ดังนี้

$$M(s) = \frac{K_1(1 + T_1s)(1 + T_2s)}{K_2(1 - K_1)} \dots \dots \dots (13)$$

เขียนอยู่ในรูป difference ได้ดังนี้

$$M_n = \frac{K_1}{K_2(1-K_1)} (e_n + (T_1+T_2)(e_n - e_{n-1})/T + T_1T_2(e_n - 2e_{n-1} + e_{n-2})/T^2)$$

โดยที่ T = sampling time

e_n = ค่าผิดพลาดเมื่อทดสอบครั้งที่ n ใด ๆ

M_n = สัญญาณควบคุมทดสอบครั้งที่ n ใด ๆ

บทที่ 2

อุปกรณ์การทดลอง

1. Pressure Transmitter (P\I Converter) จะเป็นอุปกรณ์ตรวจจับความดันของน้ำ ซึ่งสามารถอ่านค่าได้ตั้งแต่ 0 ถึง 6 bar และ จะแปลงความดันนี้เป็น กระแส DC. ในช่วง 4 ถึง 20 mA แต่ทั้งนี้จะต้องมีไฟเลี้ยงให้แก่ Pressure Transmitter โดยใช้ไฟ DC. ขนาด 20 V.

2. วงจร I\V Converter จะทำหน้าที่รับกระแสจาก Pressure Transmitter มาทำการเปลี่ยน ให้เป็น DC.Voltage ขนาด -5 ถึง +5 V. ซึ่งเป็นสัญญาณ Analog จากนั้นสัญญาณนี้จะถูกส่งเข้า A\D Card เพื่อแปลงสัญญาณ จาก Analog เป็น Digital เพื่อใช้สำหรับการประมวลผลโดย Computer ต่อไป

3. วงจร V\I Converter จะทำหน้าที่รับสัญญาณที่ประมวลผลโดย Computer แล้ว ซึ่งจะส่งออกมาเป็น Analog Voltage ขนาด 0 ถึง +5 V. มาทำการแปลงเป็น กระแสไฟ DC. ขนาด 4 ถึง 20 mA ที่จะใช้ในการควบคุม I\I Converter อีกที

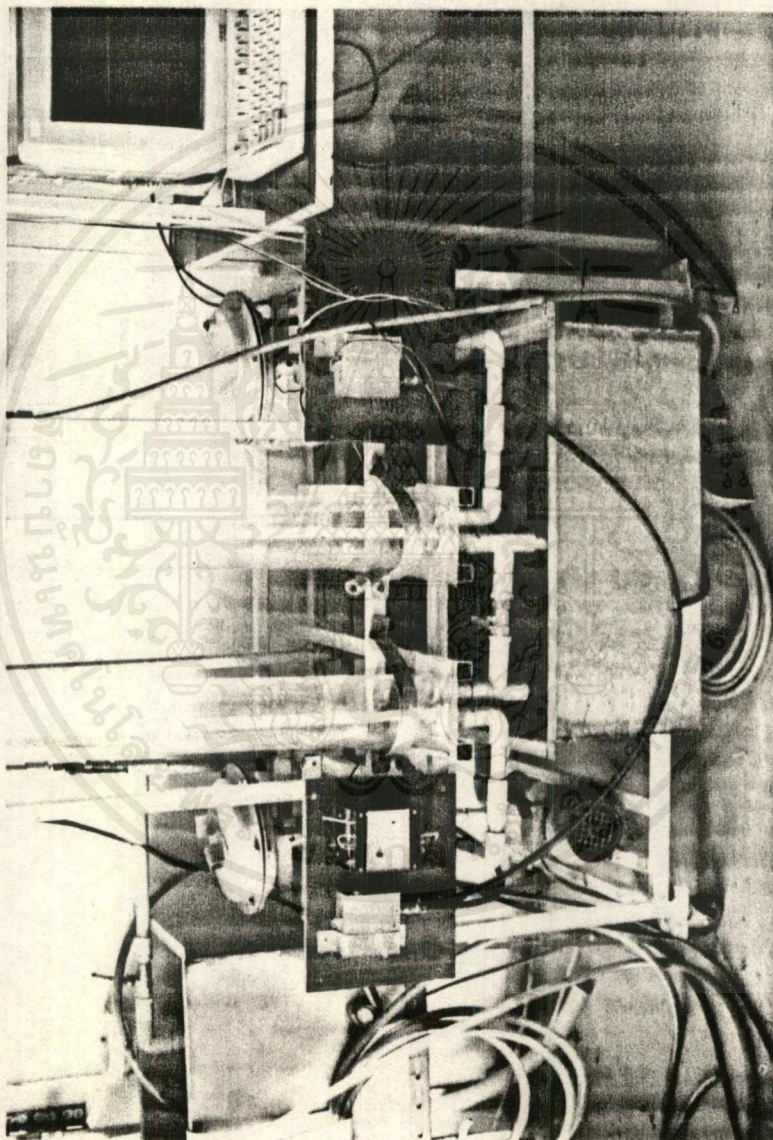
4. I\I Converter เป็นอุปกรณ์รับกระแสขนาด 4 ถึง 20 mA จาก วงจร V\I Converter มาทำการปิด - เปิดลม เพื่อให้ลมไปดัน Pressure Control Valve อีกทีหนึ่ง

5. Pressure Control Valve จะเป็น Pneumatic Control Valve ชนิดหนึ่ง จะต้องอาศัยลมจาก บีมลมมาทำการปิด - เปิด เพื่อปล่อยให้น้ำจากถังบนไหลเข้าสู่ท่อพลาสติก

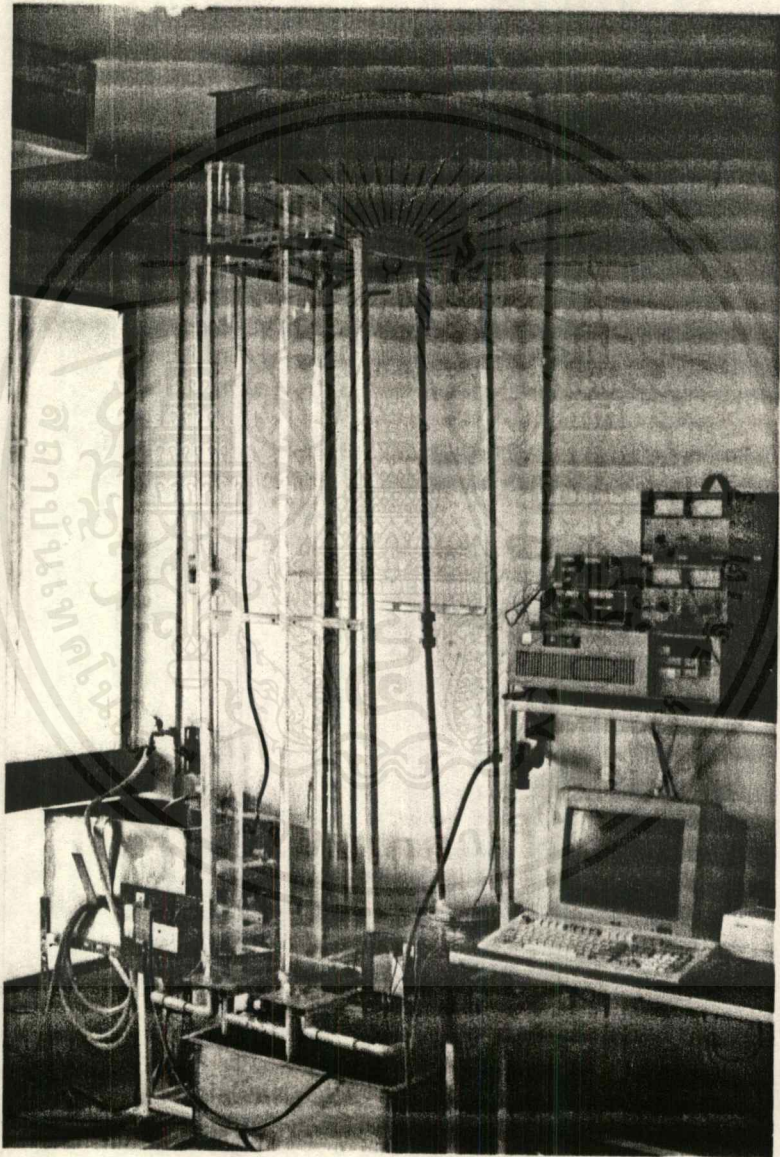
6. ท่อพลาสติก จะทำหน้าที่แสดงผลการควบคุมระดับน้ำของ Feedback Control System ที่เราได้จำลองขึ้นมา

7. Software หรือ ส่วน Program ทาง Computer ที่ใช้ในการเก็บข้อมูล และใช้ในการประมวลผล ประกอบด้วย program ที่สำคัญ ๆ ส่วน คือ

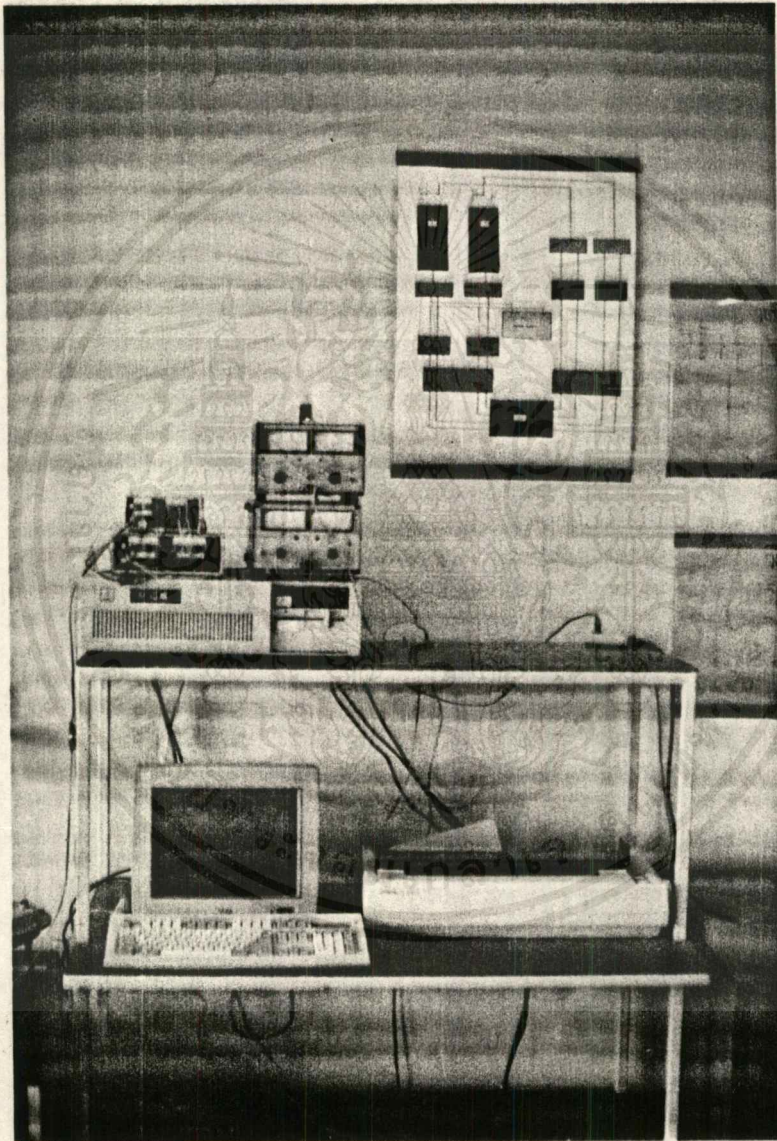
- V-H & H-V เป็นโปรแกรมสำหรับเก็บข้อมูล regression ของความสัมพันธ์ ระหว่าง $V = f(H)$ และ $H = f(V)$
- Regression เป็นโปรแกรมสำหรับให้ประมาณค่าพหุนามที่ผ่านจุดของกลุ่มข้อมูลโดยใช้วิธี least square
- Ziegler เป็นโปรแกรมที่ใช้ในการหาค่า Control Parameter ต่าง ๆ ที่ใช้ในการ ควบคุมระดับน้ำ
- single เป็นโปรแกรมที่ใช้ในการควบคุมระดับน้ำแบบ ท่อเดียว
- simutanous เป็นโปรแกรมที่ใช้ในการควบคุมระดับน้ำแบบ สองท่อพร้อมกัน
- Model เป็นโปรแกรมที่ใช้ในการควบคุมระดับน้ำแบบ Model Reference Control



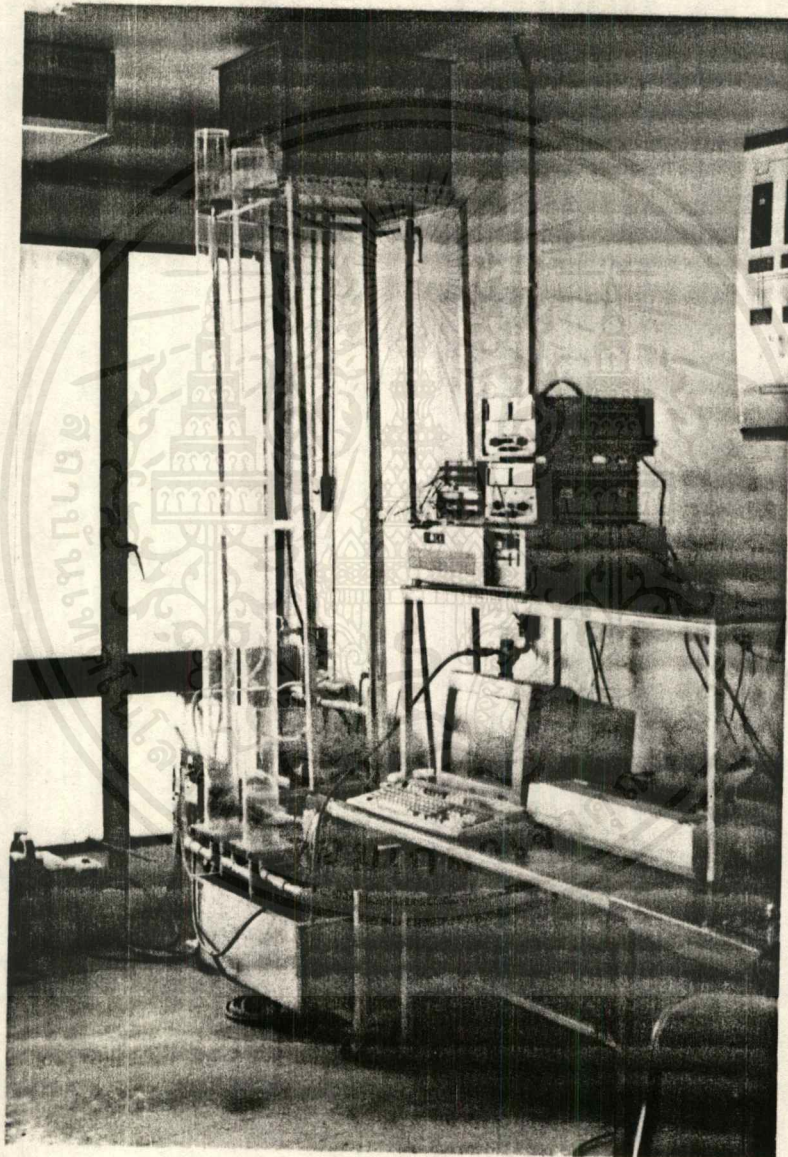
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



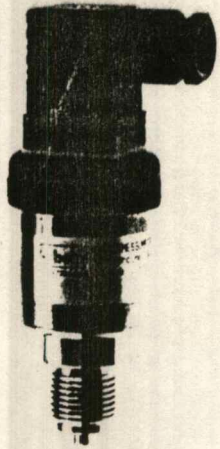
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Design Features

- Piezo-resistive measuring element
- Stainless steel housing, protection per IP65
- Overlapping range balancing, reduction to the next size (accessible from the exterior)
- Independent of shock or vibration influence
- Load-independent output signal 4...20 mA, 2-wire circuitry
- Measuring ranges 0...0.6 bar up to 0...400 bar
- Individual compensation
- Accuracy 0.3 % f.s.
- Meas. system vacuum tight, overload protected, position independent

Options

- explosion proof design

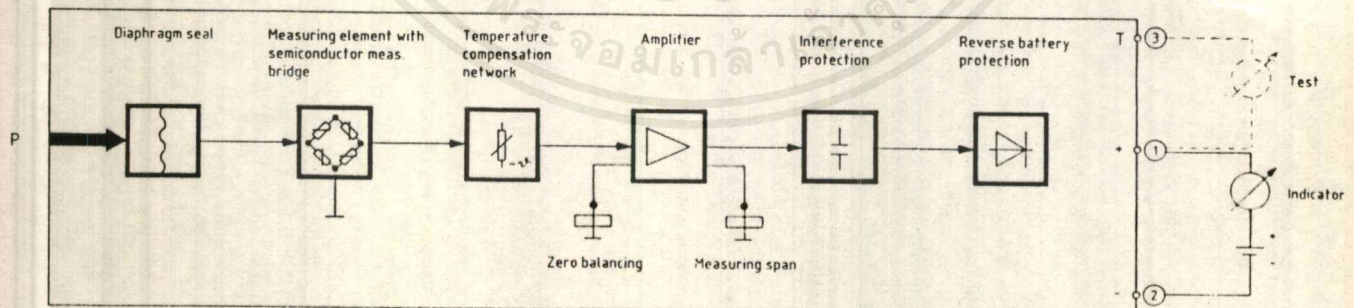
General Description

The E 2313 pressure transmitter is accommodated in a stainless steel housing of the IP 65 protection type. The electrical connection is effected with reverse battery protection in 2-wire circuitry via a bent plug-type connection element. The balancing elements - accessible from the exterior - are protected by a sealing cap. The sensor system is protected by means of an internal stainless steel diaphragm. The integrated electronics is manufactured using the SMD technology and enclosed in plastics. Pressure compensation for ranges up to 10 bar is effected by a sinter washer under the socket base. Gauges with ranges above 10 bar do not require pressure compensation.

The adaptation of the transmitter to changing operating conditions is ensured by servomechanisms located beneath a sealing cap. All intermediate values can be adjusted up to the next lower measuring range (e.g. 0...16 bar to 0...10 bar).

The devices come up to the state of the art under normal operating conditions. In case of special requirements, please let us have further specifications.

Circuit Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Technical Data

Standard Pressure Ranges

Gauge Pressure (bar)

-1...+3	0...0,6	0...6*	0...60*
-1...+5	0...1*	0...10	0...100*
-1...+9	0...1,6	0...16	0...160
-1...+15	0...2,5*	0...25*	0...250
0...1	0...4	0...40	0...400

Absolute Pressure (bar abs)

0...1	0...2,5	0...4	0...6
0...1,6			

Balancing

Zero $\pm 10\%$ Measuring range + 5 % and up to
next lower meas. range

Overload Limit

Up to 100 bar - 2 x the measuring span
Above 160 bar - 1,5 x the meas. span

Overload Influence

 ≤ 25 bar: $\leq 0.1\%$ f.s. > 25 bar: $\leq 0.2\%$ f.s.

Materials Stainless Steel

Diaphragm: 1.4404

Socket: 1.4571

Housing: 1.4301

Options

Transmitter explosion proof, PTB-No.: Ex 89.C.2112

E1 EEx ib II C T5

supply current circuit: $U \leq 30$ V, $I \leq 150$ mA, $P \leq 1$ Wtransmitter: $C_i \leq 12$ nF, $L_i = 0$ ambient temperature: $-25...+40^\circ\text{C}$

E2 EEx ia II C T5

supply current circuit: $U \leq 20$ V, $I \leq 100$ mA, $P \leq 0,6$ Wtransmitter: $C_i \leq 12$ nF, $L_i = 0$ ambient temperature: $-25...+40^\circ\text{C}$

Filling media

Silicone oil

Output Signal

4...20 mA, 2-wire circuitry

Linearity and Hysteresis

Up to 100 bar - $\leq 0.3\%$ f.s.Above 160 bar - $\leq 0.4\%$ f.s.

Burden

24 V supply: $R \leq 600$ Ohm

Burden Influence

with full burden change

 $dR = 600$ Ohm: $\leq 0.1\%$ f.s.

Response Time

 ≤ 20 ms

Supply Voltage

24 V DC, operational range

12...30 V DC

Influence of Operating Voltage

 $\leq 0.2\%$ / 10 V f.s.

Admissible Storage Temperature

 $-25...+80^\circ\text{C}$

Temperature-compensated Range

 $-10...+60^\circ\text{C}$

Temperature Influence

at zero point and meas. span
typical $\pm 0.5\%$ (max. $\leq \pm 0.8\%$) for the
entire compensated range, at nominal
range and 23°C

Type of Protection (housing)

IP 65 (with connection plug mounted
conforming to specifications)

Electrical Connection

Plug-type connecting element
acc. to DIN 43650

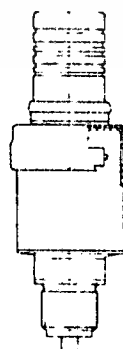
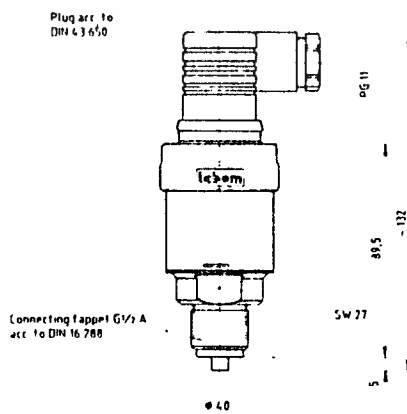
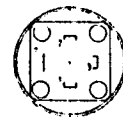
Process connection

G 1/2 A

Weight

appr. 300 g

Dimensions

The adjusting potentiometer is accessible after
lifting of the sealing capTerminal identification:
Terminal 1 - Supply
Terminal 2 - Supply
Terminal 3 - Test against -

Data for Ordering

Type

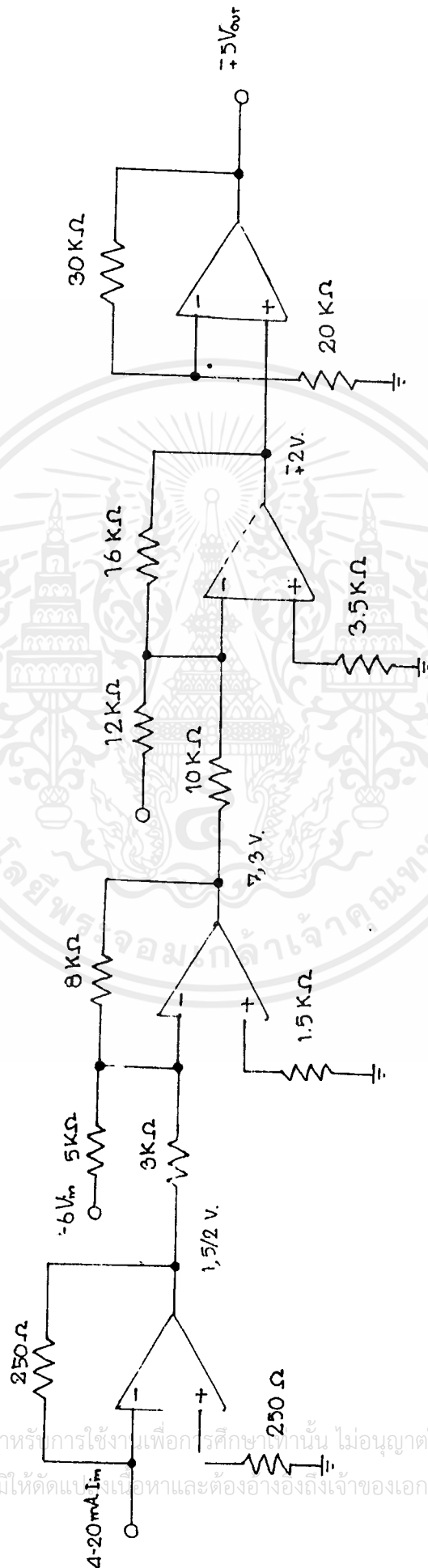
Meas. range

Option

Data for ordering

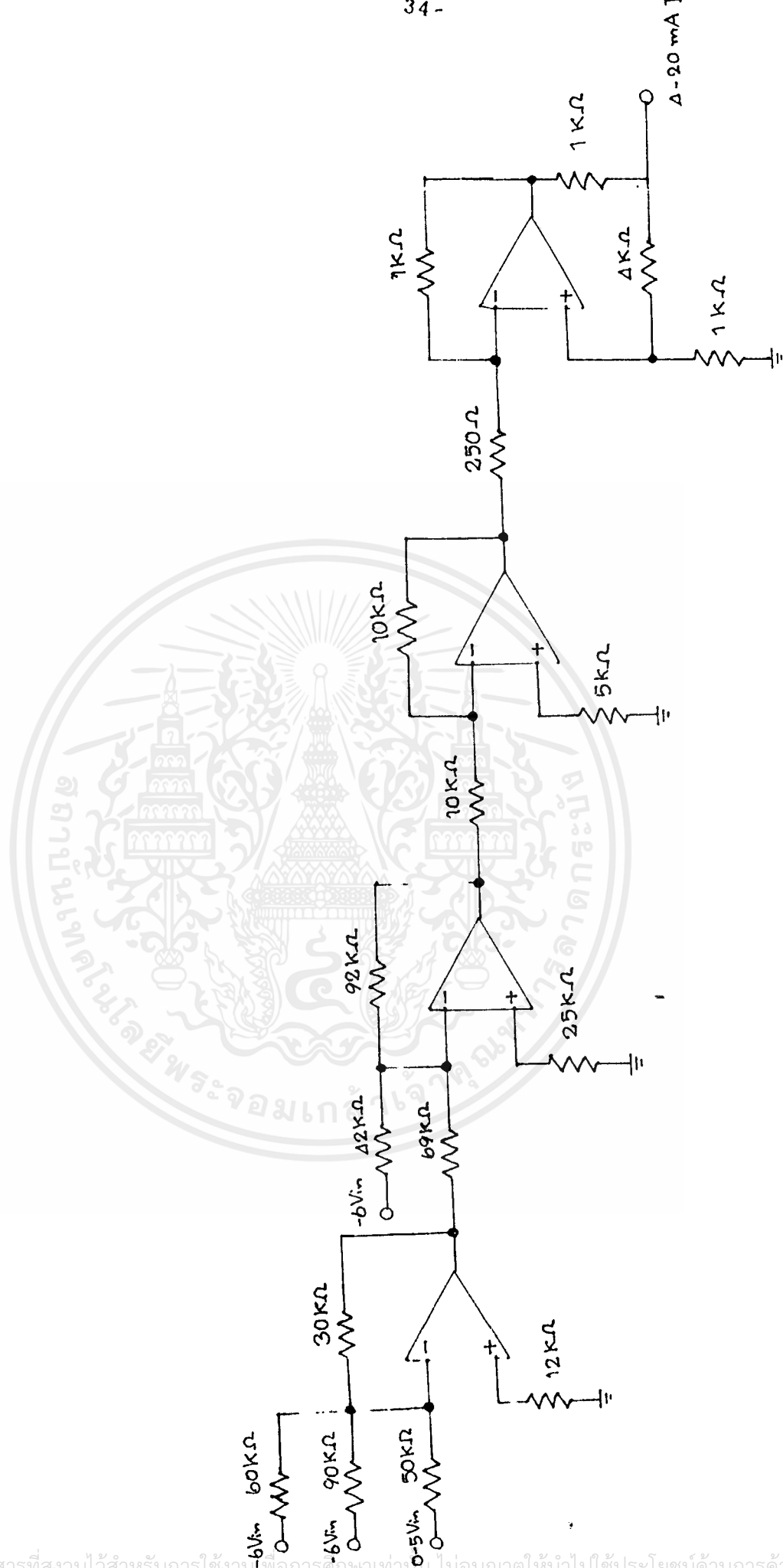
E 2313 /

* ex-stock delivery



I / V CONVERTER CIRCUIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า -
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Introduction to The PCL-812

PCL - 812 เป็น card ที่มีความสามารถสูง ความเร็วสูง และมี function การทำงานให้เลือกมากมาย ซึ่งสามารถใช้ร่วมกับเครื่อง IBM PC/XT/AT และ compatible computer ซึ่งเป็น card ที่มีความสามารถสูง เหมาะกับการใช้ในห้องทดลอง และในอุตสาหกรรม ซึ่งการใช้งานนี้ รวมไปถึงการเก็บข้อมูล, กระบวนการควบคุม, การทดสอบแบบอัตโนมัติ และยังเป็นเครื่องมือที่ใช้อำนวยความสะดวกในโรงงานอีกด้วย

1-2. คุณสมบัติของ PCL-812

- *.สามารถรับข้อมูลเข้าได้ 16 ช่อง
- *.ในงานด้านอุตสาหกรรมใช้ converter ขนาด 12-bit (HADC574Z) อัตราการแปลงสัญญาณสูงสุด คือ 30 KHz ใน DMA mode
- *.ช่วงของสัญญาณเข้า มีทั้งขั้วบวกและขั้วลบ คือ ± 1 V, ± 2 V, ± 5 V, ± 10 V.
- *.มี A/d trigger mode 3 ตัว คือ : software trigger
: programable pacer trigger
: external trigger pulse trigger
- *.มีความสามารถที่จะแปลงสัญญาณ A/D ด้วยโปรแกรมระบบควบคุม หรือ DMA
- *.มี INTEL 8253-5 เป็นตัวรับสัญญาณ trigger ซึ่งรับด้วยอัตรา 0.5 MHz ถึง 30 นาที ต่อหนึ่งจังหวะ timer time base มีความถี่ 2 MHz สำหรับ 16-bit counter channel จะใช้สำหรับการทำงานอย่างอื่น
- *.สำหรับด้าน output เป็นแบบ 12-bit ออกจาก D/A card ซึ่งมีช่วงสัญญาณตั้งแต่ 0 - 5 โวลต์ แต่สามารถเพิ่ม range ได้อีกโดยการ ใช้ -5 เป็นจุดอ้างอิง ทั้งนี้ความถูกต้องของสัญญาณก็ขึ้นอยู่กับ สัญญาณที่เข้ามาทาง A/D ด้วยสัญญาณ AC หรือ DC จากภายนอก ก็สามารถนำมาสร้าง D/A range ได้
- *.สามารถประยุกต์ใช้กับสัญญาณ input และ output ขนาด 16-bit ได้

1-5 Product Specification

1-5-1 Analog Input (A/D Converter)

- Channels : 16 Single-ended.
Resolution : 12 bits.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input Range : Bipolar : $\pm 10\text{V}$, $\pm 5\text{V}$, $\pm 2\text{V}$, $\pm 1\text{V}$.
 Overvoltage : Continuous $\pm 30\text{V}$ Max.
 Conversion type : Successive Approximation.
 Converter : HADC574Z.
 Conversion speed : 30 KHz Max.
 Accuracy : 0.015 % of reading ± 1 bit.
 Linearity : ± 1 bit.
 Trigger mode : Software trigger, on-board programmable time or external trigger
 Data transfer : Program control, Interrupt control or DMA.
 External trigger : (TTL compatible, load 0.4 mA max. at 0.5V (low) or 0.05mA max. at 2.7V (high)).

1-5-2 Analog Output (D/A converter)

Channels : 2 channels.
 Resolution : 12 Bits.
 Output Range : 0 to $+5\text{V}$ with fixed -5V reference. $\pm 10\text{V}$ with external DC or AC reference.
 Reference voltage : Internal : -5V ($\pm 0.05\text{V}$).
 External : DC or AC, $\pm 10\text{V}$ Max.
 Conversion type : 12 bit monolithic multiplying.
 Analog devices : AD754/AKN or equivalent.
 Linearity : $\pm 1/2$ bit.
 Output drive : $\pm 5\text{mA}$ max.
 Settling time : 30 microseconds.

1-5-3 Digital Input

Channel : 16 bits.
 Level : TTL compatible.
 Input voltage : Low - 0.8V max. at 0.5V.
 High - 0.05mA max. at 2.7V.

1-5-4 Digital Output

Channel : 16 bits.
 Level : TTL compatible.
 Output voltage : Low - Sink 8mA at 0.5V max.
 High - Source -0.4mA at 2.4V min.

1-5-5 Programmable Timer/Counter.

Device : INTEL 8253-5
 Counter : 3 channels, 16 bit. 2 channels permanently connected to 2 MHz clock as programmable pacer, 1 channel free for user application.
 Input, gate : TTL/DTL/CMOS compatible.
 Time base : 2 MHz.
 Pacer Output : 35 minutes/pulse to 0.015 MHz.

1-5-6 Interrupt Channel

Level : IRQ 2 to 7, jumper selectable.
 Enable : Via S0, S1 and S2 of CONTROL register.

1-5-7 DMA Channel

Level : 1 or 3, jumper selectable.
 Enable : Via S0, S1 and S2 of CONTROL register.

1-5-8 General Specifications

Power consumption : +5V : typ. 500mA, max. 1A
 +12V : typ. 50mA, max. 100mA
 -12V : typ. 14mA, max. 20mA.
 I/O connector : 20-pin post header for Analog/Digital I/O ports.
 Adapter available to convert to 37 pin D-type connector.

I/O base address : Require a 16 consecutive address locations. Base address definable by the DIP switches for address line A8- A4. (Factory setting is &H220).

Operating Temp : 0 to +50 deg.C.

Storage Temp : -20 to +65 deg.C.

Weight : 8.6 oz (242.89 gm).

INSTALLATION

2-2 Jumper Settings

PCL-812 card ถูกออกแบบเพื่อให้ใช้งานได้ง่าย ประกอบด้วย dip switches 2 ตัว และมี jumper 7 ตัว ซึ่งมีการทำงานที่สัมพันธ์กัน ตำแหน่งของ switch แต่ละตัวสามารถเปิดดูได้ใน Appendix.B

2-2-1 Base Address Selection

: Switch name : SW1

อุปกรณ์โดยส่วนมากที่ต่อติดภายนอกอยู่กับคอมพิวเตอร์และ Interface Card จะถูกควบคุมโดยการใช้ I/O Port โดยที่ I/O Port ที่ทำการควบคุมอุปกรณ์ต่าง ๆ เหล่านี้จะใช้ช่วง address ที่ว่างอยู่ ของ PC I/O port ในการส่งสัญญาณควบคุม Interface card และอุปกรณ์ที่ต่อภายนอกอยู่กับคอมพิวเตอร์ จากตารางข้างล่างจะแสดงถึงตำแหน่ง address ต่าง ๆ ของ PC I/O Port ซึ่งจะมีประโยชน์ในการเลือกตำแหน่งของ Base address ของ PCL - 812 Card

APPENDIX C PC I/O PORT ADDRESS MAP

I/O Address Range (Hex)	Function
000-1FF	Base system
200	Reserved
201	Game control
202-277	Reserved
278-27F	Second printer port.
280-2F7	Reserved
2F8-2FF	COM2
300-377	Reserved
378-37F	First printer port.

380-3AF	Reserved
3B0-3BF	Mono Disp/Print adapter
3C0-3CF	Reserved
3D0-3DF	Color/Graphics
3E0-3EF	Reserved
3F0-3F7	Floppy disk drive
3F8-3FF	COM1

I/O Port base address ของ PCL-812 สามารถเลือกได้โดยการ set dip Switch ทั้ง 8 ตัว และช่วง address ที่ใช้งานของ PCL-812 คือ 16 address นับจาก Base address ขึ้นมา ดังนั้นในการเลือกช่องว่างของ PC I/O Port address นั้นจะต้องแน่ใจว่าในช่วงนั้นมีช่วง address ที่ว่างมากพอกับช่วง address ใช้งานของ PCL - 812 คือ 16 address ขึ้นไป

สำหรับ Base addresss ที่ใช้งานได้ดีของ PCL-812 จะมีค่าตั้งแต่ Hex 200 จนถึง Hex 3F0 แต่ถึงอย่างไรก็ตามในช่วงดังกล่าวนี้ PC บางเครื่องอาจจะใช้ติดต่อกับอุปกรณ์บางตัวอยู่ก็เป็นได้ ดังนั้นจึงควรตรวจสอบให้ดีเสียก่อนในการเลือก Base address ของ PCL-812 โดยปกติค่า Base address ของ PCL-812 จะถูก set ที่ตำแหน่ง Hex 220 มาเรียบร้อยแล้วจากโรงงานผลิต แต่ถ้าผู้ใช้มีความประสงค์จะเปลี่ยนก็ย่อมทำได้โดยการปรับตำแหน่ง dip Switch ซึ่งสามารถทำได้ตามตารางต่อไปนี้

I/O Address Range (Hex)	Switch Position						
	1	2	3	4	5	6	7
	A9	A8	A7	A7	A5	A4	A3
(Fixed)							
200-20XF	1	0	0	0	0	0	x
210-21XF	1	0	0	0	0	1	x
220-22XF ^m	1	0	0	0	1	0	x
220-23XF	1	0	0	0	1	1	x
300-30XF	1	1	0	0	0	0	x
3F0-3FXF	1	1	1	1	1	1	x

Note :- ON = 0 , OFF = 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- A4...A9 correspond to PC bus address lines.
- * means factory setting.

2-2-2: Wait State Selection

ในเครื่อง PC ที่มีความเร็วสูง ๆ ค่า wait state จะมีความจำเป็นอย่างยิ่งสำหรับ PCL-812 เพื่อที่จะทำให้ได้รับข้อมูลที่มีความเสถียร PCL-812 สามารถเลือกค่า wait state ได้เป็น 0, 2, 4 และ 6 ตามลำดับโดยการปรับ dip Switch ตำแหน่งที่ 7 และ 8 ตามตารางข้างล่างนี้

Switch Position		Time
7	8	Delay
0	0	0
1	0	2
0	1	4
1	1	6

2-2-3 : Bipolar Input Range Selection

Switch name : SW2

ในการกำหนดช่วงสัญญาณ analog ที่เข้า PCL-812 สามารถทำได้โดยการปรับ dip Switch 5 ตัว ตามตารางข้างล่างดังต่อไปนี้

Switch Position					Bipolar
1	2	3	4	5	Range
ON	OFF	ON	OFF	X	+/- 10 V
OFF	ON	ON	OFF	X	+/- 5 V
ON	OFF	OFF	ON	X	+/- 2 V
OFF	ON	OFF	ON	X	+/- 1 V

Note : See Chapter 5-2 for the user definition analog input range

2-3 Connector Pin Assingment

การเชื่อมต่อ PCL - 812 กับอุปกรณ์ภายนอกทำได้โดยการเชื่อมต่อ Connector บน PCL-812 ซึ่งมี 20 pin ต่อ 1 Connector กับอุปกรณ์ภายนอกนั้นโดยใช้สายแถบ (flat cables) ตำแหน่งต่าง ๆ ของ Connector บน PCL-812 และความหมายของ pin ขาต่าง ๆ ในแต่ละ Connector ดูได้จาก diagram ใน APPENDIX B และดังต่อไปนี้

Legend :

A/D	-	Analog input
A.GND	-	Analog ground
D/A	-	Analog output
D/O	-	Digital output
D/I	-	Digital input
D.GND	-	Digital and supply ground
CLK	-	Clock input for the 8253
GATE	-	Gate input for the 8253
OUT	-	Signal output for the 8253
VREF	-	Voltage reference

Connector 1 (CN1) - Analog Input (Single-ended channels)

A/D	0	1	2	A.GND
A/D	1	3	4	A.GND
A/D	2	5	6	A.GND
A/D	3	7	8	A.GND
A/D	4	9	10	A.GND
A/D	5	11	12	A.GND
A/D	6	13	14	A.GND
A/D	7	15	16	A.GND
A/D	8	17	18	A.GND
A/D	9	19	20	A.GND

Connector 2 (CN2) - Analog Output

A/D	10	1	2	A.GND
A/D	11	3	4	A.GND
A/D	12	5	6	A.GND
A/D	13	7	8	A.GND
A/D	14	9	10	A.GND
A/D	15	11	12	A.GND
D/A	1	13	14	A.GND
D/A	2	15	16	A.GND
V.REF	1	17	18	A.GND
V.REF	2	19	20	A.GND

Connector 3 (CN3) Output

D/O	0	1	2	D/O	1
D/O	2	3	4	D/O	3
D/O	4	5	6	D/O	5
D/O	6	7	8	D/O	7
D/O	8	9	10	D/O	9
D/O	10	11	12	D/O	11
D/O	12	13	14	D/O	13
D/O	14	15	16	D/O	15
D.GND		17	18	D.GND	
+ 5V		19	20	+ 12V	

Conector 4 (CN4) - Digital Input

D/I	0	1	2	D/I	1
D/I	2	3	4	D/I	3
D/I	4	5	6	D/I	5
D/I	6	7	8	D/I	7
D/I	8	9	10	D/I	9
D/I	10	11	12	D/I	11
D/I	12	13	14	D/I	13
D/I	14	15	16	D/I	15
D.GND		17	18	D.GND	
+ 5V		19	20	+ 12V	

Connector 5 (CN5) - Counter

EX. TRG	1	2	CTR1	GATE
DRDY	3	4	CTR0	CLK
2 MHz	5	6	CTR0	OUT
	7	8	CTR0	GATE
	9	10	CTR1	OUT
	11	12		
	13	14		
	15	16		
D.GND	17	18	D.GND	
+ 5V	19	20		

Register Structer and Format

ช่วง address ใช้งานของ PCL-812 คือ 16 address ดังที่ได้กล่าวมาแล้ว ดังนั้นในการที่จะทำโปรแกรม PCL-812 นั้น จำเป็นอย่างยิ่งที่จะต้องทำความเข้าใจความหมาย และหน้าที่ของ address ต่าง ๆ ตั้งแต่ address ที่ศูนย์นับจาก PCL-812 Base addresss ไปจนถึง address ที่

16 นับจาก PCL-812 Base address ซึ่งสรุปออกมาได้เป็นดังนี้

4-1 I/O Port Address Map

The following table shows the location of each register and driver relative to the base address , and its usage.

Location	Read	Write
Base +0	Counter 0	Counter 0
+1	Counter 1	Counter 1
+2	Counter 2	Counter 2
+3	N/U	Counter control
+4	A/D low byte	CH1 D/A low byte
+5	A/D high byte	CH1 D/A high byte
+6	D/I low byte	CH2 D/A low byte
+7	D/I high byte	CH2 D/A high byte
+8	N/U	Clear interrupt request
+9	N/U	N/U
+10	N/U	MUX scan channel
+11	N/U	PCL - 812 Control
+12	N/U	Software A/D trigger
+13	N/U	D/O low byte
+14	N/U	D/O high byte
+15	N/U	

*N/U = Not Used

4-2 Data Register

The A/D data registers use address Base +4 and +5.

Data format :

1. A/D low byte and Channel number

BASE	+4	D7	D6	D5	D4	D3	D2	D1	D0
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

2. A/D high byte

BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	DRDY	AD11	AD10	AD9	AD8

LEGEND :

AD11 to AD0 - Analog to digital data. AD0 is the least significant byte (LSB) and AD11 is the most significant byte (MSB) of the D/A data.

DRDY - Data ready signal.

4-3 Mux Scan Register

The MUX scan register is a write only register using address BASE +10. The low nybble provides the scan channel number. The MUX multiplexer switches to the new channel when writing to this register.

Data Format :

BASE +10	D7	D6	D5	D4	D3	D2	D1	D0
Scan channel	X	x	x	x	CL3	CL2	CL1	CL0

LEGEND :

CL3 to CL0 - Scan channel number.

4-4 Digital I/O Registers

The PCL-812 offers 16 digital input channels and 16 digital output channels. These I/O channels use the input address BASE +6 and BASE +7. The output port are at BASE +13 and BASE +14. The data format of each port is as follow :

Data Format :

BASE +6	D7	D6	D5	D4	D3	D2	D1	D0
(read port)								
D/I low byte	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
BASE +13	D7	D6	D5	D4	D3	D2	D1	D0
(write port)								
D/O low byte	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

BASE +7	D7	D6	D5	D4	D3	D2	D1	D0
(read port)								
D/I high byte	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8
BASE +14	D7	D6	D5	D4	D3	D2	D1	D0
(write port)								
D/O high byte	DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8

4-5 D/A Output Register

The D/A output registers are write registers using address BASE +4, +5, +6, +7.

Data Format :

BASE +4	D7	D6	D5	D4	D3	D2	D1	D0
D/A #1	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
low byte								
BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
D/A #1	X	X	X	X	DA11	DA10	DA9	DA8
high byte								
BASE +6	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
low byte								
BASE +7	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2	X	X	X	X	DA11	DA10	DA9	DA8
high byte								

LEGEND :

DA11 to DA10 - Digital to analog data. DA0 is the least significant byte (LSB) and DA11 is the most sinificant byte (MSB)of the D/A data.

4-6 PCL-812 Control Register

The PCL-812 control register is a write-only register using address BASE+11.This register provides information

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

on the operating modes of the PCL-812.

DATA Format :

BASE +11	D7	D6	D5	D4	D3	D2	D1	D0
PCL-812	X	X	X	X	X	S2	S1	S0

CNTL status

A. Under internal trigger condetion

S2 S1 S0

0 0 0 : Disable Software & Pacer trigger.

0 0 1 : Enable Software trigger only.

0 1 0 : Enable Pacer trigger using DMA transfer only

1 1 0 : Enable Pacer trigger using program transfer or interrupt transfer. If using pacer trigger and program transfer, the jumper JP4 must set to the "X" position, as below:

IRQ	2	3	4	5	6	7	x
JP4	0	0	0	0	0	0	0
	0	0	0	0	0	0	0

A/D Converter

PCL-812 จะทำการแปลงสัญญาณ จากสัญญาณ analog ที่มีขนาด 12 bit เนื่องจาก register 1 ตัว สามารถเก็บข้อมูลได้แค่ 8 bit เพราะฉะนั้น จึงจำเป็นต้องใช้ register 2 ตัว ในการเก็บข้อมูล ดังนั้น ข้อมูล A/D จึงถูกเก็บไว้ใน register 2 ตัว ซึ่งอยู่ ณ. ตำแหน่ง address ที่ BASE +4 และ +5 โดยที่ข้อมูล A/D low byte จะอยู่ในตำแหน่ง D0 ถึง D7 ณ. address ที่ BASE +4 และข้อมูล A/D high byte จะอยู่ที่ตำแหน่ง D0 ถึง D3 ณ. address ที่ BASE +5 โดยที่ข้อมูลนัยสำคัญที่น้อยที่สุด (Least significant data) คือ D0 ใน BASE +4 และข้อมูลนัยสำคัญมากที่สุด (Most significant data คือ D3 ใน BASE +5 และการเลือกช่องสำหรับรับสัญญาณ analog สามารถเลือกได้ที่ ตำแหน่ง BASE +10

The data formats of the A/D data registers are:

A/D low byte and Channel number.

BASE +4	D7	D6	D5	D4	D3	D2	D1	D0
	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

A/D high byte.

BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	DRDY	AD11	AD10	AD9	AD8

Data format :

BASE +10	D7	D6	D5	D4	D3	D2	D1	D0
	X	X	X	X	CL3	CL2	CL1	CL0

Trigger Mode.

การ Trigger คือการส่งสัญญาณไปยัง PCL - 812 ให้เริ่มทำการรับข้อมูล และทำการแปลงข้อมูลนั้น จาก analog ให้เป็น digital หรือ digital ให้เป็น analog ในการแปลงข้อมูลทุกครั้งจะต้องทำการส่งสัญญาณ Trigger ไปยัง PCL-812 ทุกครั้งก่อนเสมอ

การส่งสัญญาณ Trigger สามารถทำได้โดยใช้วิธีใด วิธีหนึ่งต่อไปนี้.

1. Software Trigger สัญญาณ Trigger จะถูกควบคุมโดยตัวโปรแกรมจากคำสั่งในโปรแกรมนั้นสำหรับการส่งสัญญาณ Trigger โดยวิธีนี้จะไม่เป็นที่นิยมใช้สำหรับการแปลงสัญญาณ A/D ที่มีความเร็วในการแปลงสูง เนื่องจากข้อจำกัดในการ execute program นั้น

2. On-board programmable pacer สำหรับ PCL-812 จะใช้ INTEL 8253 programmable interval timer/counter ในการส่ง

สัญญาณแบบนี้ โดยที่ counter 1 และ 2 ของ INTEL 8253 จะให้สัญญาณ Trigger ที่มีคาบเวลาที่เที่ยงตรงสูง ในการแปลงสัญญาณแบบนี้ มักใช้ในงานที่ต้องการความเร็วสูง

3. External pulse trigger การควบคุมการส่งสัญญาณ Trigger แบบนี้ จะทำการผ่าน EXT.TRG (connector 5 pin 1) การส่งสัญญาณ Trigger แบบนี้ มักใช้ในงานที่มีคาบเวลาในการแปลงที่ไม่คงที่ แต่การแปลงสัญญาณขึ้นอยู่กับสภาวะเงื่อนไข เช่นการควบคุมอุณหภูมิของ Thermocouple

A/D Datatransfer

กระบวนการแปลงสัญญาณจาก analog เป็น digital นั้น สามารถทำได้ 3 วิธี เมื่อใช้ PCL-812 แต่ในที่นี้จะขอกล่าวเพียงวิธีเดียวซึ่งเป็นวิธีที่ง่ายที่สุด และ ใช้ในการทำโครงการนี้ คือ วิธี "program control data transfer" วิธีการนี้ จะใช้หลักของ Polling Method คือหลังจากที่สัญญาณ Trigger ถูกส่งไปยังตัวแปลงสัญญาณ A/D program ที่ผู้ใช้เขียนขึ้น จะทำการตรวจสอบจุดสิ้นสุดของการแปลงข้อมูล (End of Conversion : EOC) โดยตรวจดูที่ EOC bit ของตัวแปลงสัญญาณ A/D ถ้าหาก EOC bit ของตัวแปลงสัญญาณ A/D ถูกตรวจจับได้แสดงว่าข้อมูลได้ทำการแปลงเสร็จสิ้นลงแล้ว ต่อจากนั้น ข้อมูลที่ถูกแปลงแล้ว (สัญญาณ digital) ก็จะถูกเคลื่อนย้ายจาก A/D data register ไปยัง หน่วยความจำของ PC โดยโปรแกรมที่ผู้ใช้เขียนขึ้นมา

How to Excecute an A/D Conversion

algorithm ในการแปลงสัญญาณ analog เป็น digital สำหรับวิธี Program Control data Transfer ที่ใช้ PCL -812 มีขั้นตอนดังนี้

- ขั้นที่ 1. เลือกช่องรับสัญญาณ analog โดยการระบุ MUX Scan range (BASE + 10)
- ขั้นที่ 2. ส่งสัญญาณ Triggerไปที่ Control register (BASE+11)
- ขั้นที่ 3. คอยสัญญาณ EOC bit โดยการตรวจสอบ A/D high byte register (BASE +5) EOC bit
- ขั้นที่ 4. ผ่านข้อมูลจาก A/D Converter โดยอ่านจาก A/D data regidter (BASE +5 และ BASE +4) ต้องทำการอ่านค่า high byte ก่อน
- ขั้นที่ 5. ทำการแปลงรูปข้อมูล จาก binary เป็น รูปจำนวนเต็ม

D/A Conversion

สำหรับ PCL-812 ออกแบบมาให้มีช่องส่งสัญญาณ analog ได้ 2 ช่อง โดยการใช้บัพเฟอร์ เป็น 2 เท่าของขนาด 12 bit register ที่ทำหน้าที่แปลงสัญญาณ D/A นี้ จะเป็น register แบบเขียนได้อย่างเดียว (write only) โดยใช้ address ที่ BASE +4, +5, +6 และ +7 ซึ่งมี DA0 เป็น least significant byte และ DA11 เป็น Most significant byte ของข้อมูล

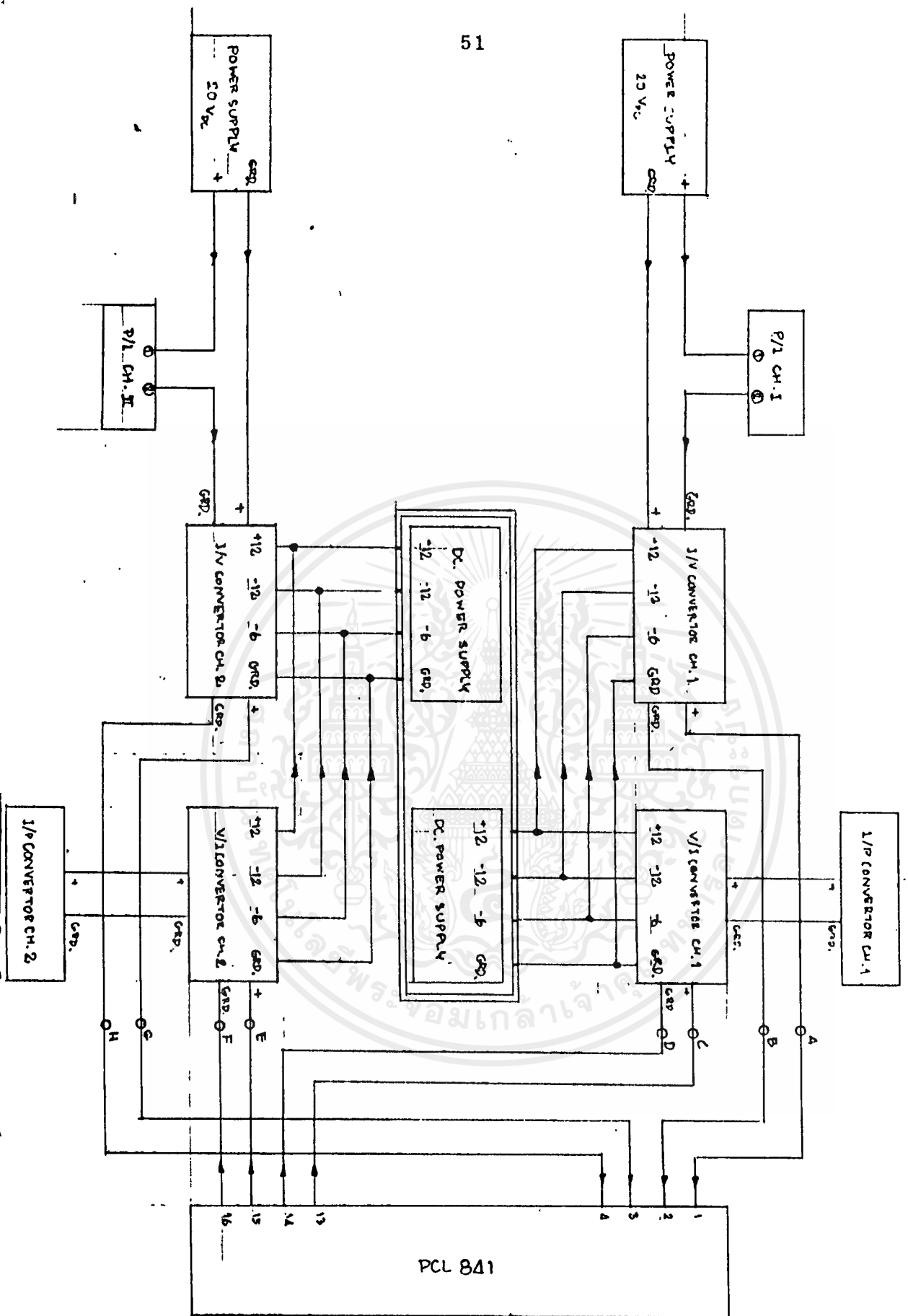
BASE +4	D7	D6	D5	D4	D3	D2	D1	D0
D/A #1	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
low byte								

BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
D/A #1	X	X	X	X	DA11	DA10	DA9	DA8
high byte								

BASE +6	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
low byte								

BASE +7	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2	X	X	X	X	DA11	DA10	DA9	DA8
high byte								

ข้อระลึกรในการส่งสัญญาณ digital เพื่อเข้าไปแปลงให้เป็นสัญญาณ analog นั้น จะต้องส่งค่าที่มีนัยสำคัญน้อยที่สุด (least significant byte) เข้าไปก่อนเสมอ ซึ่งค่าที่ส่งเข้าไปนี้จะถูกเก็บไว้ใน (D/A) register ที่วาระยะหนึ่ง หลังจากที่มีนัยสำคัญมากที่สุด (Most significant byte) ถูกส่งไปเข้าแล้ว ค่า low byte และ high byte ก็จะถูกนำมารวมกัน และส่งไปยังตัวแปลง D/A พร้อม ๆ กันทีเดียว สัญญาณ analog ที่ได้จะมีค่าอยู่ในช่วง 0 ถึง +5 โวลต์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการทดลอง

-การเก็บข้อมูลเพื่อหาความสัมพันธ์ระหว่าง voltage กับ ความสูงของน้ำในท่อ และ ความสัมพันธ์ระหว่างความสูงของน้ำในท่อ กับ voltage การหาความสัมพันธ์ระหว่างความสูงของน้ำในท่อกับvoltageที่เข้า PCL-812

; program ที่ใช้ "V-H & H-V .EXE

ขั้นตอนในการเก็บข้อมูล

1. ตัดสายไฟที่เข้าระหว่าง PCL -812 ที่จุด A,B สำหรับ CH1 และ G,H สำหรับCH2 ในรูป circuit ของระบบออก แล้วนำมาต่อเข้ากับ VOM เพื่อที่จะทำการวัด voltage

2. เรียกโปรแกรม V-H & H-V.EXE จากแผ่น Project disk

3. เลือกทำรายการในข้อ 2 คือ "Getting H-V data"

4. ใส่รายละเอียดต่างๆ ที่โปรแกรมต้องการลงไป ข้อควรระวังคือค่า Sample Interval ไม่ควรทำให้ มีค่าช่วงที่ห่างมากเกินไป เพราะจะทำให้ได้ ข้อมูลที่หายขาด ค่าที่แนะนำคือ Sample Interval = 5

5. เมื่อโปรแกรมเริ่มทำงาน ให้ใส่น้ำลงในท่อตามค่าระดับที่ระบุไว้บนจอแล้วอ่านค่า voltage ที่สอดคล้องกับความสูงของระดับน้ำในขณะนั้นจาก VOM บันทึกค่า voltage นั้นลงในคอมพิวเตอร์

6. ทำซ้ำไปเรื่อยๆจนกว่าค่าความสูงของระดับน้ำในท่อจะถึงค่าที่ตั้งไว้ การวิเคราะห์หาความสัมพันธ์ของข้อมูล

1. เรียกโปรแกรม Regress.EXE จากแผ่น Project disk

2. ใส่รายละเอียดต่างๆที่โปรแกรมต้องการลงไป แล้วเลือก degree ของ Polynomial ที่จะใช้ในการประมาณความสัมพันธ์ของกลุ่มข้อมูลเหล่านั้น ให้เหมาะสมโดยพิจารณาจาก Scattering diagram

3. สมการของ fitting curve จะอยู่ในรูป

$$y = a_0 + a_1x + a_2x^2 + \dots + a_nx^{n-1} \text{ โดยค่า } a_0, a_1, a_2, \dots, a_n$$

คือคำตอบของ Program Regress .EXE

การหาความสัมพันธ์ระหว่าง voltage ที่ออกจาก PCL -812

: Program ที่ใช้ "V-H & H-V .EXE

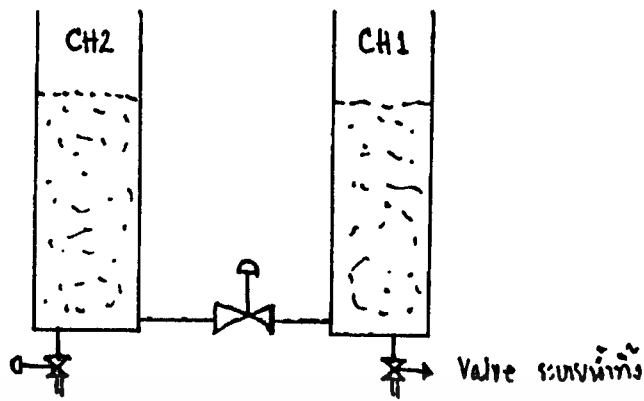
ขั้นตอนในการเก็บข้อมูล

1. เรียกโปรแกรม V-H & H-V .EXE จากแผ่น Project disk
 2. เลือกทำรายการในข้อ1คือ "Getting V-H data"
 3. เมื่อปรากฏ รายการย่อย ให้เลือกต่อไปอีกให้เลือกข้อ1 คือ "Run model Singlery" สำหรับการเก็บข้อมูลของท่อเดียว หรือเลือกข้อ2 คือ "Run model Simultaneously" สำหรับการเก็บข้อมูลของสองท่อ
 4. เมื่อบันทึกรายการตามข้อ3 แล้วก็ทำสไลด์เปลี่ยนที่โปรแกรมต้องการลงไป
 5. เมื่อโปรแกรมทำงาน PCL - 812 จะส่งค่า voltage ออกไป ทำให้ Pnuematic Control valveเปิดมีน้ำไหลเข้าท่อได้ จากนั้นคอยสักครู่จนกว่าระดับน้ำจะคงที่ (steady) แล้วทำการบันทึกค่าความสูงของระดับน้ำที่สอดคล้องกับค่า voltage ที่ออกจาก PCL - 812 ในขณะนั้นลงบนคอมพิวเตอร์
 6. ทำซ้ำข้อ 5 เรื่อยๆ จนกว่าค่า voltage ที่ออกจาก PCL - 812 มีค่าเพิ่มขึ้นจนถึงค่าสุดท้ายที่ตั้งไว้
 7. ข้อควรระวังนอกจากค่า SampleInterval ที่ต้องเลือกให้เหมาะสมแล้วว่าการบันทึกค่าความสูงของระดับน้ำต้องรอจนกว่าระดับนั้นเข้าสู่ภาวะ steady จริง ๆ เสียก่อน
- การวิเคราะห์ความสัมพันธ์ของข้อมูล

เรียก Program Regress.EXE เพื่อทำการวิเคราะห์ความสัมพันธ์ของ voltage ที่ออกจาก PCL - 812 กับความสูงของระดับน้ำในท่อ

หมายเหตุ

การเก็บข้อมูลของ ความสัมพันธ์ดังกล่าวจะต้องทำทุกครั้ง เมื่อมีการปรับหรือเปลี่ยน valve ที่ทำหน้าที่ระบายน้ำถึงที่กันท่อทั้ง 2 ความรูป

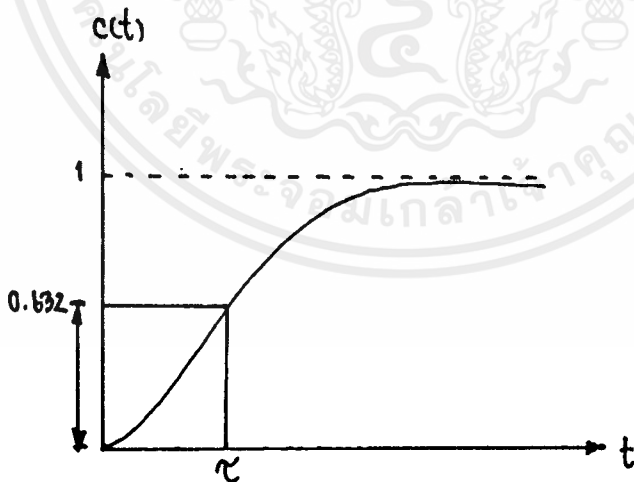


การหาคุณสมบัติของระบบ

ค่าคุณสมบัติของระบบ (System Characteristic) ได้แก่ ค่า Time Constant และ ค่า Delay time ที่เราต้องทำการกำหนดค่าคุณสมบัติเหล่านี้ ก็เพื่อนำไปคำนวณหาค่า Control parameter (ซึ่งได้แก่ k_p , T_i , T_d) โดยอาศัยทฤษฎีของ Ziegler and Nichol

ในการหาค่า Time Constant ของระบบนั้น เราทำได้โดยวิธีการดังต่อไปนี้

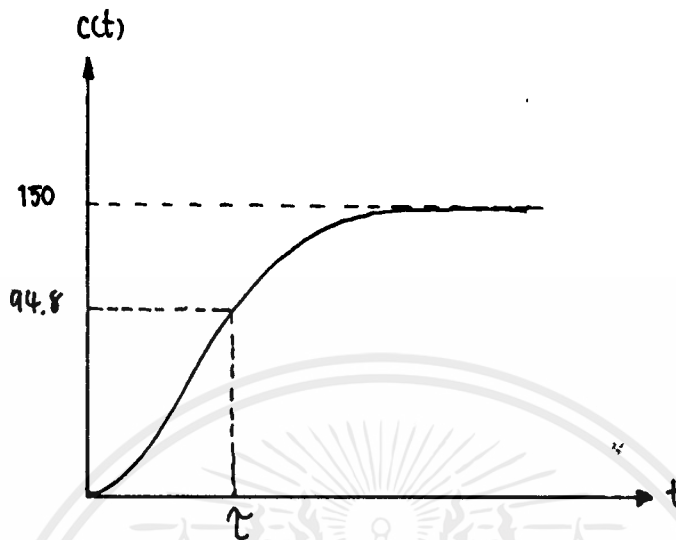
1. เราทราบว่าค่า Time Constant คือ เวลาที่ระบบใช้ตั้งแต่เริ่มต้น จนถึง 63.2 % ของค่าที่สภาวะคงที่ (steady) ดังรูป



2. ดังนั้นถ้าเราทำการจับเวลาตั้งแต่ที่ $t=0$ จนถึง 63.2% ของค่าที่สภาวะคงที่ เราก็สามารถหาค่า Time Constant ของระบบได้ เช่น ถ้าเราตั้งจุด set point ของระบบไว้ที่ 150 cm. ดังนั้น Time Constant ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือเวลาที่ใช้ตั้งแต่เริ่มต้นจนถึง 63.2 % ของค่า 150cm. หรือเท่ากับ 94.8cm.



3. ในการหาค่า Time Constant ของระบบเราจะใช้ Program Zieg2Sin.EXE สำหรับการควบคุมแบบ 1 ท่อ หรือ Zieg2Sim.EXE สำหรับการควบคุมแบบ 2 ท่อ เมื่อทำการเรียกโปรแกรมจากแผ่น Project disk แล้วก็ทำการใส่รายละเอียดต่าง ๆ ที่โปรแกรมต้องการลงไป

4. เมื่อโปรแกรมเริ่ม run และ Pnuematic Control valve เปิดให้น้ำไหลเข้าท่อ ก็ให้ทำการจับเวลาทันที และหยุดจับเวลาตำแหน่งที่ 63.2% ของค่า Set point เวลาที่จับได้คือ (Time Constant) ของระบบ (τ)

5. ทำซ้ำข้อ 3-4 ประมาณ 3 ถึง 4 ครั้ง โดยคงค่า set point ไว้ที่ค่าเดิม แล้วนำค่า Time Constant ที่วัดได้ในแต่ละครั้งมาเฉลี่ยเป็นค่า Time Constant ของระบบ

ในการหาค่า Delay time เราทำได้โดยวิธีการดังต่อไปนี้

1. ค่า delay time หรือ dead time หมายถึงช่วงเวลาที่ระบบเสียไป เนื่องจากความล่าช้าในการสังเกตควบคุมของ Controller เพราะในความเป็นจริง model ของเราประกอบไปด้วยอุปกรณ์ทางกลเสียเป็นส่วนใหญ่ เช่น Pnuematic Control valve ซึ่งเป็นอุปกรณ์ที่เคลื่อนที่และมีการตอบสนองต่อสัญญาณควบคุมได้ช้า ดังนั้นเมื่อ model ได้รับสัญญาณควบคุมมาจากตัว

Controller จะเห็นได้ว่าระบบของเราจะไม่ตอบสนองต่อสัญญาณควบคุมนั้นในทันทีทันใดเพราะมีการสูญเสียเวลาส่วนหนึ่งไปกับการเคลื่อนที่ของ valve ในการ ปิด-เปิด ตามสัญญาณควบคุมนั้นและนอกจากนี้ยังมีการสูญเสียเวลาส่วนหนึ่งในการไหลของน้ำจาก Pnuematic Control valve ไปเข้าสู่ท่อพลาสติกใส เวลาที่สูญเสียไปทั้งนี้เราเรียกว่า delay time หรือ dead time ของระบบ

2. การหาค่า delay time ทำได้โดยเรียกใช้ Program

"Zieg2Sin.EXE" สำหรับการควบคุมแบบท่อนเดียว หรือ "Zieg2Sim.EXE" สำหรับการควบคุมแบบ 2 ท่อ ข้อสำคัญก่อนที่จะทำโปรแกรมต้องทำการระบายน้ำที่ค้างอยู่ในท่อน้ำระหว่าง Pnuematic Control valve กับท่อพลาสติกใสออกให้หมดเสียก่อน มิฉะนั้นจะทำให้ค่า delay time ที่ได้มีความคลาดเคลื่อนสูง

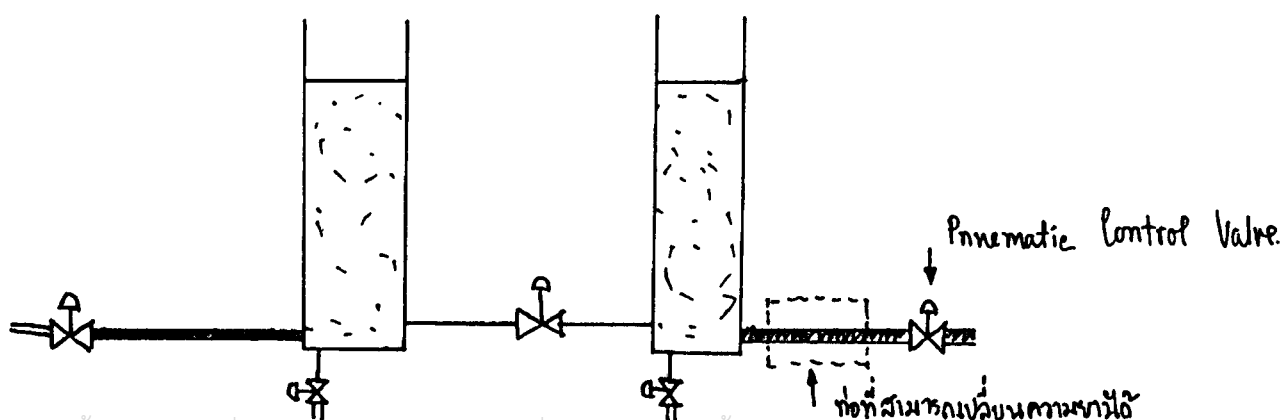
3. เมื่อ load โปรแกรมออกมาเรียบร้อยแล้ว ก็ใส่รายละเอียดต่าง ๆ ที่โปรแกรมต้องการลงไป

4. ทำการจับเวลาตั้งแต่ก้าน valve เริ่มขยับและหยุดจับเวลาเมื่อมีน้ำไหลเข้าท่อพลาสติกใส ค่าเวลาที่จับได้คือค่า delay time

5. ทำซ้ำข้อ 2-4 หลาย ๆ ครั้ง แล้วนำค่าที่ได้มาเฉลี่ยเป็นค่า delay time ของระบบ

หมายเหตุ

การหาค่า Time Constant และ Delay time จะต้องทำทุกครั้งเมื่อมีการเปลี่ยนความยาวของท่อน้ำระหว่าง Pnuematic Control valve กับ ท่อพลาสติกใสตามรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองควบคุมระดับน้ำในท่อเดียว

: Program ที่ใช้ "CongrSin.EXE"

ขั้นตอนการทดลอง

1.ต่อชุดการทดลองตามรูป ที่แสดงไว้ และ เปิดไฟเลี้ยง Pressure Sensor,เปิดไฟเลี้ยงวงจร i/v-v/i และ เปิดลมมาเข้า Pneumatic Control valve (ป้อนลมเข้าที่ความดัน 1.4 bar)

2.ในกรณีที่มีการปรับวาล์วน้ำทั้งที่กั้นท่อพลาสติก ซึ่งเราจะได้ผลการแสดงความสัมพันธ์ระหว่างความสูงของน้ำในท่อกับ voltage ที่เข้า PCL-812 และ voltage ที่ออกจาก PCL-812 กับความสูงของน้ำในท่อเป็นสมการใหม่ ดังนั้นจึงจำเป็นต้องใส่ ความสัมพันธ์ดังกล่าวเข้าไปใน โปรแกรมและทำการ compile program "CongSin.C" ใหม่ โดยมีขั้นตอนดังนี้

2.1ทำการ load editor TURBO C

2.2เมื่อ load TURBO C ขึ้นจอภาพได้แล้ว ให้ทำการ load file "FUNC1.C"

2.3ทำการแก้ไข file ดังกล่าวโดยใส่สมการแสดงความสัมพันธ์ที่หาได้ใหม่ลงไป ซึ่งมีรูปแบบในการใส่สมการดังนี้

#define ที่ฟังก์ชัน (สมการ...)

และโดยที่ $V(H)$ คือ ชื่อของสมการแสดงความสัมพันธ์ระหว่างความสูงของน้ำในท่อกับ voltage ที่ออกจาก PCL-812

$H(V)$ คือ ชื่อของสมการแสดงความสัมพันธ์ระหว่าง voltage ที่เข้า PCL-812 กับความสูงของน้ำในท่อ

2.4เมื่อทำการแก้ไข file ดังกล่าวแล้วก็ กดปุ่ม F2 เพื่อทำการ SAVE file "FUNC .C" ไว้

2.5load file "CONGRSIN.C" ขึ้นมา

2.6อย่าไปทำการ แก้ไขตัวโปรแกรมนี้เป็นอันขาด เพราะอาจทำให้โปรแกรมเกิดerror ขึ้นได้

2.7กดปุ่มF9เพื่อทำการcompileโปรแกรมเพื่อให้กลายเป็นสกุล ".EXE"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.เรียก program "CONGRSIN.EXE" ขึ้นมาแล้วใส่รายละเอียดต่าง ๆ ที่ตัวโปรแกรมต้องการลงไป ได้แก่ค่า Time Constant, Delay time, Sample Interval เป็นต้น

4.ในขณะที่ RUN program ถ้าเราต้องการจะ พิมพ์กราฟผลตอบสนองของระบบ ก็ให้กดปุ่ม Shift+Print Screen โปรแกรมก็จะทำการพิมพ์ภาพ Graphic ออกมาให้

5.ถ้าต้องการจะสิ้นสุดการควบคุมระดับน้ำ ก็ให้กดปุ่มใด ๆ ก็ได้บน key board โปรแกรมจะหยุดการทำงานเอง และ valve จะปิดตัวเองโดยอัตโนมัติ

การทดลองการควบคุมระดับน้ำแบบ 2 ท่อ

: Program ที่ใช้ "CongrSim.EXE"

ขั้นตอนการทดลอง

ดำเนินการทดลองเหมือนกับการทดลองแบบท่อเดียวทุกประการ แต่ถ้ามีการปรับ valve ระบายน้ำทั้งตรงกันท่อพลาสติกใสหุ้ม ซึ่งตรงกับกรณีข้อ 2 ของการควบคุมระดับน้ำในท่อเดียว ก็ให้ดำเนินการแก้ไข แบบเดียวกัน แต่ให้ทำการ load file "FUNC2.C"แทน file "FUNC1.C" และชื่อของสมการแสดงความสัมพันธ์มีดังนี้

$v(H)$ คือ สมการแสดงความสัมพันธ์ระหว่างความสูงของน้ำในท่อ กับ voltage ที่ส่งออกไปจาก PCL-812 ของท่อที่ 1

$v_v(H)$ คือ "_____"
voltage ที่ส่งออกไปจาก PCL-812 ของท่อที่ 2

$H1(V)$ คือ สมการแสดงความสัมพันธ์ระหว่าง voltageที่เข้าPCL-812 กับความสูงของน้ำในท่อที่ 1

$H2(v)$ คือ "_____"
กับความสูงของน้ำในท่อที่ 2

การทดลองการควบคุมระดับน้ำโดยระบบโมเดลอ้างอิง

: program ที่ใช้ "IMCGR.EXE"

ขั้นตอนการทดลอง

1. ต่อชุดการทดลองเหมือนกับการควบคุมระดับน้ำในท่อเดียว
2. เรียกโปรแกรม "IMCGR.EXE" แล้วทำการป้อนรายละเอียดต่างๆ ที่โปรแกรมต้องการลงไป
3. ในขณะที่ทำการ RUN Program ถ้าต้องการพิมพ์ภาพ Graphics ของผลตอบสนองของระบบก็ให้กดปุ่ม Shift + Print Screen
4. ถ้าต้องการจะสิ้นสุด การควบคุมระดับน้ำ ก็ให้กดปุ่มใด ๆ ก็ได้บน key board โปรแกรมจะหยุดทำงานเองและ valve จะปิดตัวเองโดยอัตโนมัติ



บทที่ 3

Program และ Flowchart

ส่วนของโปรแกรมที่ใช้ในการควบคุมระบบทั้งหมดได้แก่

1. โปรแกรม regress ใช้ในการหา สมการ Polynomial และหาค่า coefficient ต่าง ๆ ของ สมการ นอกจากนี้ ยังใช้ในการ plot curve เพื่อดูกราฟของสมการ ซึ่งสามารถทำได้ ตั้งแต่ polynomial degree 1 ถึง degree 9 ซึ่งใช้ร่วมกับ
2. โปรแกรม Zieg2Sin ใช้ในการหา Control Parameter ของ PID Control แบบท่อเดียว
3. โปรแกรม Zieg2sim ใช้ในการหา Control Parameter ของ PID Control แบบสองท่อ (Simulate)
4. โปรแกรม Congrsin ใช้ในการ Run ระบบแบบท่อเดียว
5. โปรแกรม Congrsim ใช้ในการ Run ระบบแบบสองท่อ
6. โปรแกรม IMCGRAPH ใช้ในการ Run ระบบแบบ Model Reference Control
7. โปรแกรม V-H&H-V ใช้ในการเก็บข้อมูลของ model ก่อนนำไปเข้า โปรแกรม Regress



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Program polynomial regression. */
```

```
#include <graphics.h>
```

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include <print.h>
```

```
#include "box.c"
```

```
#define No 0
```

```
#define Yes 1
```

```
main(){
```

```
int m,n,i,j,loop,choose;
```

```
int dat_trans[2];
```

```
double X[100],Y[100];
```

```
char Char_[2];
```

```
title();
```

```
new: clrscr();
```

```
textbackground(YELLOW);
```

```
textcolor(WHITE);
```

```
box(10,3,70,5);
```

```
cprintf("    How many data do you have ?    : ");
```

```
scanf("%d",&n);
```

```
box(20,7,35,23);
```

```
j=1;
```

```
for(i=1;i<=n;i++){
```

```
    gotoxy(2,j);
```

```
    cprintf("X[%d] = ",i);
```

```
    scanf("%lf",&X[i]);
```

```

    if( (j==15) & (i!=n) ){

        clrscr();

        j=0;

    }

    j++;

}

box(45,7,60,23);

j=1;

for(i=1;i<=n;i++){

    gotoxy(2,j);

    cprintf("Y[%d] = ",i);

    scanf("%lf",&Y[i]);

    if( (j==15) & (i!=n) ){

        clrscr();

        j=0;

    }

    j++;

}

old:dat_trans[1] = n;

scatter_diagram(X,Y,dat_trans);

n = dat_trans[2];

regression(n,n,X,Y);

textbackground(CYAN);

textcolor(MAGENTA);

box(10,12,70,14);

cprintf("Would you like to terminate this program ? (y/n):");

Char_[0] = getch();

switch(Char_[0]){

    case 'y': loop = Yes; break;

    case 'Y': loop = Yes; break;

    default: loop = No; break;

}

}

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 'n': loop = No; break;
case 'N': loop = No; break;
}

if(loop == Yes){
clrscr();

cprintf("Would you like to print to printer ? (y/n) : ");
Char_[0]=getche();
switch(Char_[0]){
    case 'y' : print_sol(X,Y,n); break;
    case 'Y' : print_sol(X,Y,n); break;
}

    box(10,12,70,14);
gotoxy(15,1);
cprintf("Press any key to stop.");
getch();
window(1,1,80,25);
textbackground(BLACK);
textcolor(WHITE);
clrscr();
exit(0);
}

if(loop == No){
box(10,16,70,21);

gotoxy(5,1);

printf("1.Start with new data.");

gotoxy(5,2);

printf("2.Start with old data.");

gotoxy(5,4);

printf("SELECT ==> ");

```

```

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

switch(choose){

case 1: goto new;

case 2: goto old;

default:window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

textbackground(RED);

textcolor(WHITE);

box(10,12,70,14);

gotoxy(10,1);

printf("Ellegal entry, stop program terminate!");

getch();

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

exit(1);

}

}

}

title(){

    clrscr();

    textbackground(BLUE);

    textcolor(WHITE);

```

```

box(25,2,55,4);

cprintf("    Polynomial Regression.");

box(1,6,80,21);

gotoxy(5,1);

cprintf("Polynomial regression function : ");

gotoxy(8,3);

cprintf("y = a0 + a1*x + a2*x^2 + ... + an*x^n ");

gotoxy(5,6);

cprintf("The normal equation : ");

gotoxy(5,8);

cprintf("a0*n      + a1*x      + a2*x^2      + . . . + an*x^n      = y");

gotoxy(5,9);

cprintf("a0*x      + a1*x^2      + a2*x^3      + . . . + an*x^(n+1) = y*x");

gotoxy(5,10);

cprintf("a0*x^2 + a1*x^3      + a2*x^4      + . . . + an*x^(n+2) = y*x^2");

gotoxy(5,11);

cprintf(" . . . . . ");

gotoxy(5,12);

cprintf(" . . . . . ");

gotoxy(5,13);

cprintf(" . . . . . ");

gotoxy(5,14);

cprintf("a0*x^n + a1*x^(n+1) + a2*x^(n+2) + . . . + an*x^(2*n) = y*x^n");

box(15,23,65,25);

cprintf("          press any key to continue. ");

getch();

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
```

```

#define xx getmaxx()
```

```

#define yy getmaxy()
```

```

scatter_diagram(X,Y,dat_mat)
```

```

double X[20],Y[20];
```

```

int dat_mat[5];
```

```

{
```

```

int graphdriver=DETECT,graphmode,i;
```

```

double x,y;
```

```

char word[5],c;
```

```

char graphpath[]="";
```

```

    initgraph(&graphdriver,&graphmode,graphpath);
```

```

    if(graphresult()==grOk) {
```

```

        setlinestyle(SOLID_LINE,0,THICK_WIDTH);
```

```

        settextstyle(SANS_SERIF_FONT,HORIZ_DIR,2);
```

```

        settextjustify(LEFT_TEXT,BOTTOM_TEXT);
```

```

        setlinestyle(SOLID_LINE,0,NORM_WIDTH);
```

```

        setbkcolor(BLUE);
```

```

        rectangle(xx/2+50,30,xx/2+310,65);
```

```

        moveto(xx/2+120,60);
```

```

        setcolor(WHITE);
```

```

        outtext("SCATTER DIAGRAM.");
```

```

        moveto(65,65);
```

```

        outtext("Y");
```

```

        moveto(xx-65,yy-65);
```

```

        outtext("X");
```

```

        setlinestyle(SOLID_LINE,0,NORM_WIDTH);
```

```

        line(60,60,60,yy);
```

```

line(0,yy-60,xx-60,yy-60);
setcolor(YELLOW);
for(i=1;i<=dat_mat[1];i++){
    x=X[i];
    y=Y[i];
    delay(100);
    pieslice(60+x*5,(yy-60)-y,0,360,2);
}
moveto(xx/2+60,45);
setcolor(WHITE);
outtext("Choose degree of polynomial :");
for(i=0;(c=getch()) != '\r';i++){
    word[i]=c;
    switch(c) {
        case '0' : outtext("0"); break;
        case '1' : outtext("1"); break;
        case '2' : outtext("2"); break;
        case '3' : outtext("3"); break;
        case '4' : outtext("4"); break;
        case '5' : outtext("5"); break;
        case '6' : outtext("6"); break;
        case '7' : outtext("7"); break;
        case '8' : outtext("8"); break;
        case '9' : outtext("9"); break;
    }
}
dat_mat[2]=0;
for(i=0;word[i] >= '0' && word[i] <= '9';i++)
    dat_mat[2]=dat_mat[2]*10+word[i]-'0';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 closegraph();
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    else {

gotoxy(26,14);

printf("GRAPHICS NOT ALLOWED ON YOUR PC.");

getch();

}

}

```

```

regression(M,N,X,Y)

int M,N;

double X[20],Y[20];

{

double coff[41],K[21],P[400],ans[21];

double B[22][22],Q[22][22];

int i,j;

    get_sumX(coff,X,N,M);

    arrange_sumX(coff,P,M);

    get_sumY(X,Y,K,M,N);

    make_matrix(B,Q,P,K,M);

    gauss_seidel(B,ans,M);

    print_solution(ans,Q,M);

    graph(ans,X,Y,M,N);

}

```

```

get_sumX(coefficent,x,n,m)

double coefficent[41],x[20];

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i,j;
double sum;
coefficient[1]=n;
for(i=1;i<=2*n;i++){
    sum=0;
    for(j=1;j<=n;j++){
        sum = sum + pow(x[j],(double) i);
        coefficient[i+1] = sum;
    }
}

```

```

arrange_sumX(coefficient,p,n)
double coefficient[41],p[400];
int n;
{
    int i,r=1,first_value=0,index=1;
    for(i=1;i<=pow(n+1, 2);i++){
        if(i-r == n+1){
            r=i;
            index=1;
            first_value=first_value + 1;
        }
        p[i]=coefficient[index+first_value];
        index = index+1;
    }
}

```

```

get_sumY(x,y,k,n,n)

```

```

double x[20],y[20],k[21];

```

```

int m,n;

{

double sum=0;

int i,j;

for(i=1;i<=n;i++)

sum = sum + y[i];

k[1]=sum;

for(i=1;i<=m;i++){

sum=0;

for(j=1;j<=n;j++)

sum=sum+y[j]*pow(x[j],(double) i);

k[i+1]=sum;

}

}

make_matrix(b,q,p,k,m)

double b[22][22],q[22][22],p[400],k[21];

int m;

{

int i,j,next=1;

for(i=1;i<=m+1;i++){

for(j=1;j<=m+2;j++){

if(j==m+2){

b[i][j]=k[i];

q[i][j]=b[i][j];

}

else{

b[i][j]=p[next];

q[i][j]=b[i][j];

next=next+1;

```

```

}
}
}
}

```

```
gauss_seidel(b,ans,n)
```

```
double b[22][22],ans[21];
```

```
int n;
```

```
{
```

```
int index,i,j,k;
```

```
int n; /* dimension of augmented matrix */
```

```
double M;
```

```
n=n+1;
```

```
index=0;
```

```
for(k=1;k<=n-1;k++){
```

```
for(i=2+index;i<=n;i++){
```

```
M=b[i][k]/b[k][k];
```

```
for(j=1+index;j<=n+1;j++){
```

```
b[i][j]=b[i][j]-M*b[k][j];
```

```
}
```

```
}
```

```
index = index+1;
```

```
}
```

```
back_substitute(b,ans,n);
```

```
}
```

```
back_substitute(b,ans,n)
```

```
double b[22][22],ans[21];
```

```
int n;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int shift_left,k,c;

double sum;

ans[n]=b[n][n+1]/b[n][n];

shift_left=0;

for(k=n-1;k>=1;k--){

sum=0;

for(c=n-shift_left;c<=n;c++){

sum = sum + b[k][c]*ans[c];

}

shift_left = shift_left + 1;

ans[k] = (b[k][n+1]-sum)/b[k][k];

}

}

print_solution(ans,q,m)

double ans[1,q[22][22];

int m;

{

int k,i,j,z=1,tab=1;

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

textbackground(MAGENTA);

textcolor(WHITE);

box(30,1,50,3);

cprintf("      SOLUTION  ");

textbackground(YELLOW);

```

```

box(1,5,80,23);

gotoxy(30,1);

cprintf("COEFFICIENT MATRIX.");

for(i=1;i<=m+1;i++){
for(j=1;j<=m+2;j++){
if(j==m+2){
gotoxy(tab,i+3);

cprintf("%s\t10.4g", " : ",q[i][j]);

tab=1;
}
else{
gotoxy(tab,i+3);
cprintf("%10.4g",q[i][j]);
tab = tab+11;
}
}
}

getch();

textbackground(YELLOW);

textcolor(WHITE);

clrscr();

box(1,5,80,23);

gotoxy(30,1);

cprintf("COEFFICIENT OF REGRESSION.");

for(k=1;k<=m+1;k++){
if(k<=13){
if( fabs(ans[k]) < 1E-10 ) ans[k]=0;

gotoxy(5,2+k);

cprintf("a[%2d] = %5.10f",k-1,ans[k]);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else {

if( fabs(ans[k]) < 1E-10 ) ans[k]=0;

gotoxy(45,2+z);

cprintf("a[%2d] = % 5.10f",k-1,ans[k]);

i++;

}

}

getch();

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

}

```

```

graph(ans,X,Y,M,n)
double ans[21],X[20],Y[20];
int M,n;
{
int graphdriver=DETECT,graphmode,i;
double x,y,y1,y2;
double Xscale,Yscale;
char Q[2],graphpath[]="";

```

AGAIN:

```

textbackground(BLACK);

clrscr();

textbackground(BLUE);

textcolor(YELLOW);

box(10,5,70,8);

textcolor(WHITE);

gotoxy(5,1);

```

```

gotoxy(5,2);

printf("Enter y-scale : "); scanf("%lf",&Yscale);

initgraph(&graphdriver,&graphmode,graphpath);

if(graphresult()==grOk) {

setlinestyle(SOLID_LINE,0,THICK_WIDTH);

settextstyle(SANS_SERIF_FONT,HORIZ_DIR,5);

settextjustify(LEFT_TEXT,BOTTOM_TEXT);

setlinestyle(SOLID_LINE,0,NORM_WIDTH);

setbkcolor(BLUE);

rectangle(xx/2+10,20,xx/2+310,50);

moveto(xx/2+30,40);

outtext("SCATTER DIAGRAM AND FITTING CURVE.");

moveto(65,65);

outtext("Y");

moveto(xx-65,yy-65);

outtext("X");

setlinestyle(SOLID_LINE,0,NORM_WIDTH);

line(60,60,60,yy);

line(0,yy-60,xx-60,yy-60);

setcolor(YELLOW);

for(i=1;i<=n;i++){

    x=X[i];

    y=Y[i];

    delay(100);

    pieslice(60+x*Xscale,(yy-60)-y*Yscale,0,360,2);

}

setcolor(MAGENTA);

for(x=-10;x<=60;x+=0.1){

    y1=0;y2=0;

    for(i=1;i<=M+1;i++){

```

```

y1 = y1 + ans[i]*pow(x,(double) i-1);
y2 = y2 + ans[i]*pow(x+0.1,(double) i-1);
}

line(60+x*Xscale,(yy-60)-y1*Yscale,60+(x+0.1)*Xscale,(yy-60)-y2*Yscale);
}

getch();
closegraph();
}

else {
gotoxy(26,14);
printf("GRAPHICS NOT ALLOWED ON YOUR PC.");
getch();
}

textcolor(YELLOW);
box(10,5,70,8);
textcolor(WHITE);
gotoxy(5,1);
cprintf("Redraw graph again ? (y/n) :");
Q[0]=getche();
switch(Q[0]){
    case 'y': goto AGAIN;
    case 'Y': goto AGAIN;
    case 'n': break;
    case 'N': break;
}
}

print_sol(x,y,n)

double x[100],y[100];
int n;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    int i,j,k;
    int skip,index;
    double least;

    window(1,1,80,25);

    clrscr();

    open_prn();

    skip=0;

    while(skip<n){
        least=x[1];
        for(i=1;i<=n-skip;i++){
            if(least>=x[i]){
                least=x[i];
                index=i;
            }
        }

        PRINT(PRN,"%s% 5.3f%s% 5.3f\n", "      ",least,"      ",y[index]);

        x[index]=x[n-skip];
        y[index]=y[n-skip];

        skip++;

    }

    clos_prn();
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Program Ziegler-Nichol (Single control) */
```

```
#include <graphics.h>
```

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
#include <print.h>
```

```
#include <bios.h>
```

```
#include <math.h>
```

```
#include <dos.h>
```

```
#include "box.c"
```

```
#define Base_add 0x220
```

```
#define ENABLE 1
```

```
#define SOFTTRIG 1
```

```
#define h(V) (V) /* h-v regression */
```

```
#define v(H) (H) /* v-h regression */
```

```
double get_time(void); /*Time function */
```

```
double how_long(double,double); /* Time function */
```

```
double X[500],Y[500],ERC[500]; /* co-ordinate matrix */
```

```
double SP_high; /* set point height */
```

```
double high_er;
```

```
double time=0, SampleInterval, NextSampleTime, finish, start, elapse_time;
```

```
union REGS inreg,outreg;
```

```
int CH,i=0;
```

```
main(){
```

```
Ziegler();
```

```
print_sol();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

```

```
Ziegler(){
    long int h_byte, l_byte, AD_value, DA_value, voltage_range;
    double high,v_error,volt;
    int MUX,lport,hport;
    int j=0;

    textbackground(BLUE);

    clrscr();

    textbackground(GREEN);
    textcolor(WHITE);
    box(10,5,70,10);
    gotoxy(5,1);
    cprintf("Enter set-point height (cm): "); scanf("%lf",&SP_high);
    gotoxy(5,2);
    cprintf("Enter sample interval (sec): "); scanf("%lf",&SampleInterval);
    gotoxy(5,3);
    cprintf("Enter channel number (1-2): "); scanf("%d",&CH);
    switch(CH){
        case 1: MUX=10; lport=4; hport=5; break;
        case 2: MUX=11; lport=6; hport=7; break;
        default: gotoxy(5,4);

        sound(450);

        cprintf("Fatal error, program terminate !");

        getch();

        textbackground(BLACK);

        window(1,1,80,25);

        clrscr();

        nosound();

        exit(1);
    }
}
```

```

}

window(1,1,80,25);

textbackground(BLUE);

textcolor(YELLOW);

clrscr();

gotoxy(10,3);

cprintf(" Procedure of evaluation of the controller parameter by");

gotoxy(23,4);

cprintf("Ziegler and Nichol method");

textbackground(MAGENTA);

textcolor(WHITE);

box(5,6,75,24);

gotoxy(15,1); cprintf("Time");
gotoxy(35,1); cprintf("Height");
gotoxy(55,1); cprintf("Error");
gotoxy(10,2); cprintf("-----");

time = get_time();

start = time;

NextSampleTime = time + SampleInterval;

gotoxy(14,j+3); cprintf("%5.2f",0);
gotoxy(34,j+3); cprintf("%5.2f",0);
gotoxy(54,j+3); cprintf("%5.2f",SP_high);

X[i]=0;

Y[i]=0;

ERR[i]=SP_high;

i++;

j++;

window(10,9,70,23);

```

```

/* ----- A/D conversion ----- */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do{

    /* Use of polling for timing */
    while(time < NextSampleTime){
time = get_time();
    }

    NextSampleTime = time + SampleInterval;

    finish = time;
    elapse_time = how_long(start,finish);

    /* Start interface with PCL-812 Card */
    /* Step 1: Set the input channel by specifying the MUX scan range */

outportb(Base_add+10,MUX);

    /* Step 2: Write the pacer trigger mode to the controll register */

        /* Choose mode of enabling software trigger only */
outportb(Base_add+11,ENABLE);

    /* Step 3: Wait for the EOC-End of Conversion, by reading the A/D
        high byte register EOC bit */

        /* start reading data */
outportb(Base_add+12,SOFTTRIG);

do{

    h_byte = inportb(Base_add+5);

```

```
/* Step 4: Read data from A/D converter by reading the A/D register.
```

```
Must first read high byte*/
```

```
/* high byte was success read already in step 3 */
```

```
l_byte = inportb(Base_add+4);
```

```
/* Step 5: Data coversion */
```

```
AD_value = h_byte*256 + l_byte;
```

```
/* input voltage range, -5 to +5 volt */
```

```
voltage_range = 10;
```

```
/* volt from PCL-812 */
```

```
volt = ((double) AD_value * (double) voltage_range)/4095.0 + (-5.0);
```

```
/* call regression function to calculate height in tank */
```

```
high = h(volt);
```

```
/* error is the difference in height between set point
```

```
height and presenting height in tank */
```

```
high_er = SP_high - high;
```

```
/* ----- D/A conversion ----- */
```

```
/* call regression function to convert error height to voltage */
```

```
/*V_error = v(high_er); Old program */
```

```
V_error = v(SP_high); /* New program */
```

```

/* output voltage range, 0 to +5 volt */
    voltage_range = 5;

    DA_value = (long int)( V_error*4095/voltage_range );
    h_byte = DA_value/256;
    l_byte = DA_value%256;
    outportb(Base_add+l1,ENABLE);
    outportb(Base_add+lport,l_byte); /* send low byte to the card */
    outportb(Base_add+hport,h_byte); /* send high byte to the card */

/* show data on screen */
    gotoxy(10 ,j+1); cprintf("%5.2f",elapsed_time);
    gotoxy(30,j+1); cprintf("%5.2f",high);
    gotoxy(50,j+1); cprintf("%5.2f",high_er);
    X[i]=elapsed_time;
    Y[i]=high;
    ER[i]=high_er;
    i++;
    j++;
    if( (j%14) == 0){
        j=1;
        clrscr();
    }
}while(bioskey(1) == NULL);
window(1,1,80,25);
textbackground(BLACK);
textcolor(WHITE);
clrscr();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define xx getmaxx()

#define yy getmaxy()

print_sol(){
int graphdriver=DETECT,graphmode;
double x,y;
char graphpath[]="";
int k;
initgraph(&graphdriver,&graphmode,graphpath);
if(graphresult()==grOk) {
setlinestyle(SOLID_LINE,0,NORM_WIDTH);
line(10,15,10,yy-5);
line( 5,yy-10,xx-15,yy-10);
moveto(15,20);
setcolor(CYAN);
outtext("Height (cm)");
moveto(xx-80,yy-18);
outtext("Time (sec)");
moveto(10,yy-10); /* goto origin */
setcolor(MAGENTA);
line(10,yy-10-SP_high*3,xx-15,yy-10-SP_high*3);
setcolor(YELLOW);
for(k=0;k<i;k++){
x=X[k];
y=Y[k];
lineto(10+x*10,yy-10-y*3);
}

setcolor(WHITE);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 getch(); getch();
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

closegraph();
}
else {
window(1,1,80,25);

clrscr();

textbackground(BLUE);

textcolor(YELLOW);

box(10,8,70,10);

gotoxy(3,1);

printf("GRAPHICS NOT ALLOWED ON YOUR PC OR MISSING *.BGI FILE.");

getch();getch();

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();
}

init_prn();

open_prn();

PRINT(PRN,"%s% 5.2f\n", "Set-point height : ",SP_high);

PRINT(PRN,"%s% 5.2f\n", "Sample interval : ",SampleInterval);

PRINT(PRN,"%s% d\n\n", "Channel number : ",CH);

PRINT(PRN,"          Time(sec)          Height(cm)          Error(cm)\n");

PRINT(PRN,"          -----\n\n");

for(k=0;k<i;k++){

PRINT(PRN,"%s%5.2f%s%5.2f%s%5.2f\n", "          ",X[k], "          ",Y[k], "

}

PRINT(PRN,"          -----\n");

clos_prn();

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Function to get the number of seconds elapsed since midnight on the
   system clock.*/
```

```
double get_time(){
```

```
    unsigned int hrs,mins,secs,hundredths;
```

```
    /* request DOS service 0x2c */
```

```
    inreg.h.ah = 0x2c;
```

```
    /* call interrupt 0x21 with the appropriate unions */
```

```
    intdos(&inreg,&outreg);
```

```
    hrs = outreg.h.ch;          /* hours elapsed returned in CH */
```

```
    mins = outreg.h.cl;        /* minutes elapsed returned in CL */
```

```
    secs = outreg.h.dh;        /* seconds elapsed returned in DH */
```

```
    hundredths = outreg.h.dl;  /* 1/100's seconds returned in DL */
```

```
    /* total elapsed time (in seconds) */
```

```
    return ( (double) hrs*3600.0 + (double) mins*60.0 + (double) secs + (double) hundredths/100.0)
```

```
}
```

```
/* Function to compute the number of seconds elapsed between start and finish */
```

```
double how_long(double start,double finish)
```

```
{
```

```
    return (finish-start);
```

```
}
```

```
/* Function to initial printer */
```

```
init_prn(){
```

```
    inreg.h.ah=1;
```

```
    inreg.x.dx=0;
```

```
    int86(0x17,&inreg,&outreg);
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Program Ziegler-Nichol (Simultaneous control) */
```

```
#include <conio.h>
```

```
#include <math.h>
```

```
#include <stdio.h>
```

```
#include <print.h>
```

```
#include <bios.h>
```

```
#include <graphics.h>
```

```
#include <dos.h>
```

```
#include "box.c"
```

```
#define Base_add 0x220
```

```
#define ENABLE 1
```

```
#define SOFTTRIG 1
```

```
#define b1(V) (V)
```

```
#define v1(H) (H)
```

```
#define b2(V) (V)
```

```
#define v2(H) (H)
```

```
double get_time(void);
```

```
double how_long(double,double);
```

```
double X[500],Y1[500],ER1[500],Y2[500],ER2[500];
```

```
double SP1_high,SP2_high;
```

```
double time=0,SampleInterval,NextSampleTime,finish,start,elapsed_time;
```

```
union REGS inreg,outreg;
```

```
int i=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าการ main() ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Ziegler();

print_sol();

}

```

```

Ziegler(){

long int h1_byte,l1_byte,AD1_value,DA1_value;

long int h2_byte,l2_byte,AD2_value,DA2_value;

double   high1,high1_er,volt1,V1_error;

double   high2,high2_er,volt2,V2_error;

long int voltage_range;

int      k,j=2;

    textbackground(BLUE);

    clrscr();

    textbackground(MAGENTA);

    textcolor(YELLOW);

    box(5,2,75,8);

    gotoxy(5,1);

    cprintf("Enter CH1 set-point height : "); scanf("%lf",&SP1_high);

    gotoxy(5,3);

    cprintf("Enter CH2 set-point height : "); scanf("%lf",&SP2_high);

    gotoxy(5,5);

    cprintf("Enter sample interval      : "); scanf("%lf",&SampleInterval);

    window(1,1,80,25);

    clrscr();

    gotoxy(17,2);

    cprintf("Procedure of evaluating the controller parameter by");

    gotoxy(33,3);

    cprintf("Ziegler-Nichol method");

    textbackground(CYAN);

```

```

textcolor(WHITE);
box(5,5,75,24);
gotoxy(2,1); cprintf("Time (sec)");
gotoxy(16,1); cprintf("Height1 (cm)");
gotoxy(30,1); cprintf("Error1 (cm)");
gotoxy(44,1); cprintf("Height2 (cm)");
gotoxy(58,1); cprintf("Error2 (cm)");
gotoxy(1,2);
for(k=1;k<=69;k++)
cprintf("-");

gotoxy(2,3); cprintf("%5.2f",0.0);
gotoxy(15,3); cprintf("%5.2f",0.0);
gotoxy(30,3); cprintf("%5.2f",SP1_high);
gotoxy(43,3); cprintf("%5.2f",0.0);
gotoxy(58,3); cprintf("%5.2f",SP2_high);
X[i]=0;
Y1[i]=0;
ER1[i]=SP1_high;
Y2[i]=0;
ER2[i]=SP2_high;
i++;

window(6,8,74,23);

time = get_time();

start = time;

NextSampleTime = time+SampleInterval;

do{

while(time < NextSampleTime){

time = get_time();

}

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 finish = time;
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

elapsed_time = how_long(start,finish);

NextSampleTime = time+SampleInterval;

/*----- CH1_A/D -----*/
outportb(Base_add+10,10);
outportb(Base_add+11,ENABLE);
outportb(Base_add+12,SOFTTRIG);
do{
h1_byte = inportb(Base_add+5);
}while((h1_byte & 0x10) != 0);
l1_byte = inportb(Base_add+4);
AD1_value = h1_byte*256 + l1_byte;

/*----- CH2_A/D -----*/
outportb(Base_add+10,11);
outportb(Base_add+11,ENABLE);
outportb(Base_add+12,SOFTTRIG);
do{
h2_byte = inportb(Base_add+5);
}while((h2_byte & 0x10) != 0);
l2_byte = inportb(Base_add+4);
AD2_value = h2_byte*256 + l2_byte;

voltage_range = 10;
volt1 = ((double) AD1_value * (double) voltage_range)/4095.0 + (-5);
volt2 = ((double) AD2_value * (double) voltage_range)/4095.0 + (-5);

high1 = h1(volt1);
high2 = h2(volt2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

high1_er = SP1_high-high1;
high2_er = SP2_high-high2;

V1_error = v1(high1_er);
V2_error = v2(high2_er);

/*----- CH1_D/A -----*/
voltage_range = 5;
DA1_value = (long int) (V1_error*4095/voltage_range);
h1_byte = DA1_value/256;
l1_byte = DA1_value%256;
outportb(Base_add+11,ENABLE);
outportb(Base_add+4,l1_byte);
outportb(Base_add+5,h1_byte);

/*----- CH2_D/A -----*/
voltage_range = 5;
DA2_value = (long int) (V2_error*4095/voltage_range);
h2_byte = DA2_value/256;
l2_byte = DA2_value%256;
outportb(Base_add+11,ENABLE);
outportb(Base_add+6,l2_byte);
outportb(Base_add+7,h2_byte);

gotoxy(2, j); cprintf("%5.2f",elapsed_time);
gotoxy(15,j); cprintf("%5.2f",high1);
gotoxy(30,j); cprintf("%5.2f",high1_er);
gotoxy(43,j); cprintf("%5.2f",high2);
gotoxy(58,j); cprintf("%5.2f",high2_er);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 j++;
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((j%16) == 0){
    clrscr();
    j=1;
}

X[i] = elapse_time;
Y1[i] = high1;
ER1[i] = high1_er;
Y2[i] = high2;
ER2[i] = high2_er;
i++;

}while(bioskey() == NULL);
textbackground(BLACK);
textcolor(WHITE);
}

#define xx getmaxx()
#define yy getmaxy()

print_sol(){
int graphdriver=DETECT,graphmode;
double x1,y1,x2,y2;
char graphpath[]="";
int k;

initgraph(&graphdriver,&graphmode,graphpath);
if(graphresult()==grOk) {
setlinestyle(SOLID_LINE,0,NORM_WIDTH);

line(10,15,10,yy-5);

line( 5,yy-10,xx-15,yy-10);

moveto(15,20);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outtext("Height (cm)");

moveto(xx-80,yy-18);

outtext("Time (sec)");

moveto(10,yy-10); /* goto origin */

setcolor(YELLOW);

line(10,yy-10-SP1_high*3,xx-15,yy-10-SP1_high*3);

setcolor(GREEN);

line(10,yy-10-SP2_high*3,xx-15,yy-10-SP2_high*3);

for(k=0;k<i-1;k++){

setcolor(YELLOW);

x1=X[k];

y1=Y1[k];

line(10+x1*10,yy-10-y1*3,10+X[k+1]*10,yy-10-Y1[k+1]*3);

setcolor(GREEN);

x2=X[k];

y2=Y2[k];

line(10+x2*10,yy-10-y2*3,10+X[k+1]*10,yy-10-Y2[k+1]*3);

}

setcolor(WHITE);

getch(); getch();

closegraph();

}

else {

window(1,1,80,25);

clrscr();

textbackground(BLUE);

textcolor(YELLOW);

box(10,8,70,10);

gotoxy(3,1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่า `printf("GRAPHICS NOT ALLOWED ON YOUR PC OR MISSING *.BGI FILE.");` การทุกครั้งที่มีการนำไปใช้

```

getch();getch();

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

}

init_prn();

open_prn();

PRINT(PRN,"%s% 5.2f\n", "          CH1 Set-point height : ",SP1_high);

PRINT(PRN,"%s% 5.2f\n", "          CH2 Set-point height : ",SP2_high);

PRINT(PRN,"%s% 5.2f\n\n", "          Sample interval : ",SampleInterval);

PRINT(PRN,"          Time(sec)          Height1(cm)          Error1(cm)\n");

PRINT(PRN,"          -----\n\n");

for(k=0;k<i;k++){

PRINT(PRN,"%s%5.2f%s%5.2f%s%5.2f\n", "          ",X[k], "          ",Y1[k], "

PRINT(PRN,"          -----\n");

PRINT(PRN,"          \n\n          Time(sec)          Height2(cm)          Error2(cm)\n");

PRINT(PRN,"          -----\n");

for(k=0;k<i;k++){

PRINT(PRN,"%s%5.2f%s%5.2f%s%5.2f\n", "          ",X[k], "          ",Y2[k], "

PRINT(PRN,"          -----\n\n");

clos_prn();

}

```

```

/* Function to get the number of seconds elapsed since midnight on the

```

```

system clock.*/

```

```

double get_time(){

```

```

    unsigned int hrs,mins,secs,hundredths;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* request DOS service 0x2c */
inreg.h.ah = 0x2c;

/* call interrupt 0x21 with the appropriate unions */
intdos(&inreg,&outreg);

hrs = outreg.h.ch;      /* hours   elapsed returned in CH */
mins = outreg.h.cl;     /* minutes elapsed returned in CL */
secs = outreg.h.dh;     /* seconds elapsed returned in DH */
hundredths = outreg.h.dl; /* 1/100's seconds returned in DL */

/* total elapsed time (in seconds) */
return ( (double) hrs*3600.0 + (double) mins*60.0 + (double) secs + (double) hundredths/100.0);
}

/* Function to compute the number of seconds elapsed between start and finish */
double how_long(double start,double finish)
{
return (finish-start);
}

/* Function to initial printer */
init_prn(){
inreg.h.ah=1;
inreg.x.dx=0;
int86(0x17,&inreg,&outreg);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Program single control (GRAPHICS SCREEN) by PID controller */
```

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include <bios.h>
```

```
#include <dos.h>
```

```
#include <graphics.h>
```

```
#include <print.h>
```

```
#include "box.c"
```

```
#include "func1.c"
```

```
#define Base_add 0x220
```

```
#define ENABLE 1
```

```
#define SOFTTRIG 1
```

```
#define Kp(T,L) (1.20*T/L)
```

```
#define Ti(T,L) (2.0*L)
```

```
#define Td(T,L) (0.5*L)
```

```
double get_time(void);
```

```
double how_long(double,double);
```

```
double X[1000],Y[1000],ERC[1000];
```

```
double SP_high;
```

```
double time=0,SampleInterval,NextSampleTime,finish,start,elapsed_time,delta_time,LoopTime;
```

```
double TimeConstant,DelayTime,GainAdjust;
```

```
double x,y;
```

```
char graphpath[]="";
```

```
int k;
```

```
int CH,i=0;
```

```
union REGS inreg,outreg;
```

```
main(){
```

```
    PID();
```

```
}
```

```
PID(){
```

```
long int  h_byte,l_byte,AD_value,DA_value,voltage_range;
```

```
double    kp,ti,td;
```

```
double    high,V_control,volt,high_er,V_SP,V,volt_er;
```

```
double    s=0,en,enold;
```

```
double    mn;
```

```
int        MUX,lport,hport;
```

```
int        k;
```

```
textbackground(BLUE);
```

```
clrscr();
```

```
textbackground(MAGENTA);
```

```
textcolor(WHITE);
```

```
box(10,4,70,12);
```

```
gotoxy(5,1);
```

```
cprintf("Enter channel number    : "); scanf("%d",&CH);
```

```
gotoxy(5,2);
```

```
cprintf("Enter set-point height : "); scanf("%lf",&SP_high);
```

```

gotoxy(5,3);

printf("Enter time constant      : "); scanf("%lf",&TimeConstant);

gotoxy(5,4);

printf("Enter delay time        : "); scanf("%lf",&DelayTime);

gotoxy(5,5);

printf("Enter sample interval    : "); scanf("%lf",&SampleInterval);

gotoxy(5,6);

printf("Enter gain adjust factor: "); scanf("%lf",&GainAdjust);

switch(CH){

case 1: MUX=10; lport=4; hport=5; break;

case 2: MUX=11; lport=6; hport=7; break;

        default: gotoxy(5,7);

textcolor(GREEN);

printf("Fatal error, program terminate itself! ");

sound(450);

getch();

nosound();

textbackground(BLACK);

textcolor(WHITE);

clrscr();

exit(1);

}

```

```

/*---- evaluate the controller parameter-Kp,Ti and Td ----*/

```

```

kp=Kp(TimeConstant,DelayTime)*GainAdjust;

```

```

ti=Ti(TimeConstant,DelayTime)*GainAdjust;

```

```

td=Td(TimeConstant,DelayTime)*GainAdjust;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 /*Initial error equals to set-point height because there is no water in tank at the begining*/
 ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งถ้ามีเหตุเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

enold = 1; /* 100% error */

X[i]=0;

Y[i]=0;

ERC[i]=SP_high;

time=get_time();

start=time;

NextSampleTime=time+SampleInterval;

/*----- A/D conversion -----*/

#define xx getmaxx()
#define yy getmaxy()

initgraph(&graphdriver,&graphmode,graphpath);

setbkcolor(BLUE);

setlinestyle(SOLID_LINE,0,NORM_WIDTH);

line(50,15,50,yy-5);

line(15,yy-50,xx-15,yy-50);

setcolor(WHITE);

settextstyle(SMALL_FONT,HORIZ_DIR,5);

moveto(60,20);

outtext("Height (cm)");

moveto(xx-100,yy-65);

outtext("Time (sec)");

```

```

moveto(5,yy-55-20);
outtext(" 20 -");
moveto(5,yy-55-40);
outtext(" 40 -");
moveto(5,yy-55-60);
outtext(" 60 -");
moveto(5,yy-55-80);
outtext(" 80 -");
moveto(5,yy-55-100);
outtext(" 100 -");
moveto(5,yy-55-120);
outtext(" 120 -");
moveto(5,yy-55-140);
outtext(" 140 -");
moveto(5,yy-55-160);
outtext(" 160 -");
moveto(5,yy-55-180);
outtext(" 180 -");
moveto(5,yy-55-200);
outtext(" 200 -");

```

```

moveto(50+30,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");
moveto(50+25,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("30");
moveto(50+60,yy-50);

```

```

settextstyle(SMALL_FONT,HORIZ_DIR,4);

```

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

moveto(50+55,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("60");
moveto(50+90,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(":");
moveto(50+85,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("90");
moveto(50+120,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(":");
moveto(50+115,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("120");
moveto(50+150,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(":");
moveto(50+145,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("150");
moveto(50+180,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(":");
moveto(50+175,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("180");
moveto(50+210,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(":");

```

```

moveto(50+205,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("210");

moveto(50+240,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+235,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("240");

moveto(50+270,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+265,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("270");

moveto(50+300,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+295,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("300");

moveto(50+330,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+325,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("330");

moveto(50+360,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

moveto(50+355,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("360");

moveto(50+390,yy-50);

settextstyle(SMALL_FONT,HORIZ_DIR,4);

outtext("!");

moveto(50+385,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("390");

moveto(50+420,yy-50);

settextstyle(SMALL_FONT,HORIZ_DIR,4);

outtext("!");

moveto(50+415,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("420");

moveto(50+450,yy-50);

settextstyle(SMALL_FONT,HORIZ_DIR,4);

outtext("!");

moveto(50+445,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("450");

moveto(50+480,yy-50);

settextstyle(SMALL_FONT,HORIZ_DIR,4);

outtext("!");

moveto(50+475,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("480");

moveto(50+510,yy-50);

settextstyle(SMALL_FONT,HORIZ_DIR,4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

moveto(50+505,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("510");

moveto(50,yy-50); /* goto origin */

setcolor(WHITE);

line(50,yy-50-SP_high,xx-15,yy-50-SP_high);

setcolor(WHITE);

do{

    x=X[i];

    y=Y[i];

    i++;

    lineto(50+x,yy-50-y);

    if(graphresult()!= grOk){

window(1,1,80,25);

clrscr();

textbackground(BLUE);

textcolor(YELLOW);

box(10,8,70,10);

gotoxy(3,1);

printf("GRAPHICS NOT ALLOWED ON YOUR PC OR MISSING *.BGI FILE.");

getch();getch();

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

}

```

LoopTime=time;

```
while(time<NextSampleTime){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```
finish=time;
```

```
NextSampleTime=time+SampleInterval;
```

```
elapsed_time=how_long(start,finish);
```

```
/* start interface with PCL-812 */
```

```
/* Step 1: set the input channel by specifying the MUX scan range */
```

```
outportb(Base_add+10,MUX);
```

```
/* Step 2: write the pacer trigger mode to control register, chose mode of software trigger only
```

```
outportb(Base_add+11,ENABLE);
```

```
/* Step 3: wait for EOC bit by reading A/D high byte */
```

```
outportb(Base_add+12,SOFTTRIG);
```

```
do{
```

```
    h_byte = inportb(Base_add+5);
```

```
}while((h_byte & 0x10) != 0);
```

```
/* Step 4: read low byte data */
```

```
l_byte = inportb(Base_add+4);
```

```
/* Step 5: data conversion */
```

```
AD_value = h_byte*256+l_byte;
```

```
/* input voltage , -5 to +5 volt */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
voltage_range=10;
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* volt from PCL-812 */

volt = ((double) AD_value* (double) voltage_range)/4095.0 + (-5);

/* call regerssion function to calculate height in tank */

high = h(volt);

high_er = SP_high - high;

/* control signal to model */

if(high <= lim1) V = v1(high);
if(high > lim1 && high <= lim2) V = v2(high);
if(high > lim2) V = v3(high);

/* call regression function to convert set point height to voltage */
if(SP_high <= lim1) V_SP = v1(SP_high);
if(SP_high > lim1 && SP_high <= lim2) V_SP = v2(SP_high);
if(SP_high > lim2) V_SP = v3(SP_high);

/* error is the difference in VOLTAGE between set-point height and
presenting height in tank */

volt_er = V_SP - V;

if(high_er < 5) mn = V_SP;

if(high_er < -5) mn = 0; /* valve shut off */

/*----- PID controller -----*/

if(high_er >= 5) {

    en = volt_er/V_SP;

    s = s+en; /* integral summation */

    delta_time = how_long(LoopTime,finish);

    mn = kp*( td*(en-enold)/delta_time + en + s*delta_time/ti );

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}

/*----- D/A conversion -----*/

voltage_range = 5; /* output voltage, 0 to 15 volt */

DA_value = (long int) (mn*4095/voltage_range);

h_byte = DA_value/256;

l_byte = DA_value%256;

outportb(Base_add+11,ENABLE);

outportb(Base_add+lport,l_byte);

outportb(Base_add+hport,h_byte);

X[i]=elapsed_time;

Y[i]=high;

ER[i]=high_er;

}while(bioskey(1) == NULL);

outportb(Base_add+lport,0);

outportb(Base_add+hport,0);

closegraph();

textcolor(WHITE);

textbackground(BLACK);

init_prn();

open_prn();

PRINT(PRN,"%s% d\n", "Channel number : ",CH);

PRINT(PRN,"%s% 5.2f\n", "Set-point height : ",SP_high);

PRINT(PRN,"%s% 5.2f\n", "Sample interval : ",SampleInterval);

PRINT(PRN,"%s% 5.2f\n", "Time constant : ",TimeConstant);

PRINT(PRN,"%s% 5.2f\n", "Delay time : ",DelayTime);

PRINT(PRN,"%s% 5.5f\n", "Kp : ",Kp(TimeConstant,DelayTime)*GainAdjust);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PRINT(PRN,"%s% 5.5f\n", "      Ti      : ",Ti(TimeConstant,DelayTime)*GainAdjust);
PRINT(PRN,"%s% 5.5f\n\n", "      Td      : ",Td
(TimeConstant,DelayTime)*GainAdjust);
/*PRINT(PRN,"      Time(sec)      Height(cm)      Error(cm)\n");
PRINT(PRN,"      -----\n\n");
for(k=0;k<i;k++){
PRINT(PRN,"%s%5.2f%s%5.2f%s%5.2f\n", "      ",X[k],"      ",Y[k],"
}
PRINT(PRN,"      -----\n");
*/
clos_prn();
}

/* Function to get the number of seconds elapsed since midnight on the
system clock.*/
double get_time(){
unsigned int hrs,mins,secs,hundredths;
/* request DOS service 0x2c */
inreg.h.ah = 0x2c;

/* call interrupt 0x21 with the appropriate unions */
intdos(&inreg,&outreg);

hrs = outreg.h.ch;      /* hours   elapsed returned in CH */
mins = outreg.h.cl;     /* minutes elapsed returned in CL */
secs = outreg.h.dh;     /* seconds elapsed returned in DH */
hundredths = outreg.h.dl; /* 1/100's seconds returned in DL */

/* total elapsed time (in seconds) */
return ( (double) hrs*3600.0 + (double) mins*60.0 + (double) secs + (double) hundredths/100.0);

```

```

}
```

```

/* Function to compute the number of seconds elapsed between start and finish */
```

```

double how_long(double start,double finish)
```

```

{
```

```

    return (finish-start);
```

```

}
```

```

/* Function to initial printer */
```

```

init_prn(){
```

```

    inreg.h.ah=1;
```

```

    inreg.x.dx=0;
```

```

    int86(0x17,&inreg,&outreg);
```

```

}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/**** NEW VERSION TYPE 2 *****/
```

```
/* program simultaneous control (GRAPHICS SCREEN) by PID control */
```

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
#include <bios.h>
```

```
#include <dos.h>
```

```
#include <graphics.h>
```

```
#include <print.h>
```

```
#include <math.h>
```

```
#include "box.c"
```

```
#include "func2.c"
```

```
#define Base_add 0x220
```

```
#define SOFTTRIG 1
```

```
#define ENABLE 1
```

```
#define xx getmaxx()
```

```
#define yy getmaxy()
```

```
#define Kp(T,L) (1.20*T/L)
```

```
#define Ti(T,L) (2.0*L)
```

```
#define Td(T,L) (0.5*L)
```

```
double get_time(void);
```

```
double how_long(double,double);
```

```
double X[1000],Y1[1000],ER1[1000],Y2[1000],ER2[1000];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 double SP1_high,SP2_high;
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double time=0,SampleInterval,NextSampleTime,finish,start,elapsed_time,delta_time,LoopTime;

double TimeConstant1,DelayTime1,TimeConstant2,DelayTime2;

int CH,i=0;


int graphdriver=DETECT,graphmode;
char graphpath[]="";


union REGS inreg,outreg;


main(){
    SimultaneousPID();
}


SimultaneousPID(){
    long int h_byte,l_byte,AD1_value,DA1_value,AD2_value,DA2_value,voltage_range;
    double kp1,ti1,td1;
    double kp2,ti2,td2;
    double high1,V1_SP,volt1,high1_er,V1,V1_er;
    double high2,V2_SP,volt2,high2_er,V2,V2_er;
    double s1=0,en1,enold1;
    double s2=0,en2,enold2;
    double mn1,mn2;
    int k;


    textbackground(BLUE);

    clrscr();

    textbackground(MAGENTA);

    textcolor(WHITE);

    box(10,5,70,20);

```

```

printf("Enter CH1 set-point height : "); scanf("%lf",&SP1_high);
gotoxy(5,2);
printf("Enter CH2 set-point height : "); scanf("%lf",&SP2_high);
gotoxy(5,3);
printf("Enter CH1 time constant      : "); scanf("%lf",&TimeConstant1);
gotoxy(5,4);
printf("Enter CH2 time constant      : "); scanf("%lf",&TimeConstant2);
gotoxy(5,5);
printf("Enter CH1 delay time         : "); scanf("%lf",&DelayTime1);
gotoxy(5,6);
printf("Enter CH2 delay time         : "); scanf("%lf",&DelayTime2);
gotoxy(5,7);
printf("Enter sample interval       : "); scanf("%lf",&SampleInterval);

/* Evaluate the controll parameter-Kp,Ti and Td */

kp1=Kp(TimeConstant1,DelayTime1);
ti1=Ti(TimeConstant1,DelayTime1);
td1=Td(TimeConstant1,DelayTime1);

kp2=Kp(TimeConstant2,DelayTime2);
ti2=Ti(TimeConstant2,DelayTime2);
td2=Td(TimeConstant2,DelayTime2);

/* initial error */

enold1 1;

enold2=1;    /* 100% error */

```

```

Y1[i]=0;

ER1[i]=enold1;

Y2[i]=0;

ER2[i]=enold2;


time=get_time();

start=time;

NextSampleTime=time+SampleInterval;


initgraph(&graphdriver,&graphmode,graphpath);
if(graphresult() != grOk){
window(1,1,80,25);
clrscr();
textbackground(BLUE);
textcolor(YELLOW);
box(10,8,70,10);
gotoxy(3,1);
printf("GRAPHICS NOT ALLOWED ON YOUR PC OR MISSING *.BGI FILE.");
getch();getch();
window(1,1,80,25);
textbackground(BLACK);
textcolor(WHITE);
clrscr();
exit(0);
}

setlinestyle(SOLID_LINE,0,NORM_WIDTH);

setbkcolor(BLUE);

setlinestyle(SOLID_LINE,0,NORM_WIDTH);

line(50,15,50,yy-5);
line(15,yy-50,xx-15,yy-50);

```

```

setcolor(WHITE);

settextstyle(SMALL_FONT,HORIZ_DIR,5);

moveto(60,20);

outtext("Height (cm)");

moveto(xx-100,yy-65);

outtext("Time (sec)");

settextstyle(SMALL_FONT,HORIZ_DIR,4);

moveto(5,yy-55-20);

outtext(" 20 -");

moveto(5,yy-55-40);

outtext(" 40 -");

moveto(5,yy-55-60);

outtext(" 60 -");

moveto(5,yy-55-80);

outtext(" 80 -");

moveto(5,yy-55-100);

outtext(" 100 -");

moveto(5,yy-55-120);

outtext(" 120 -");

moveto(5,yy-55-140);

outtext(" 140 -");

moveto(5,yy-55-160);

outtext(" 160 -");

moveto(5,yy-55-180);

outtext(" 180 -");

moveto(5,yy-55-200);

outtext(" 200 -");

```

```

moveto(50+30,yy-50);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 settextstyle(SMALL_FONT,HORIZ_DIR,4);
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outtext("!");
moveto(50+25,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("30");
moveto(50+60,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");
moveto(50+55,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("60");
moveto(50+90,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");
moveto(50+85,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("90");
moveto(50+120,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");
moveto(50+115,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("120");
moveto(50+150,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");
moveto(50+145,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("150");
moveto(50+180,yy-50);

```

```

outtext(":");
moveto(50+175,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("180");
moveto(50+210,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(":");
moveto(50+205,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("210");
moveto(50+240,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(":");
moveto(50+235,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("240");
moveto(50+270,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(":");
moveto(50+265,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("270");
moveto(50+300,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(":");
moveto(50+295,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("300");
moveto(50+330,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);

```

```

outtext("!");

moveto(50+325,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("330");

moveto(50+360,yy-50);

settextstyle(SMALL_FONT,HORIZ_DIR,4);

outtext("!");

moveto(50+355,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("360");

moveto(50+390,yy-50);

settextstyle(SMALL_FONT,HORIZ_DIR,4);

outtext("!");

moveto(50+385,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("390");

moveto(50+420,yy-50);

settextstyle(SMALL_FONT,HORIZ_DIR,4);

outtext("!");

moveto(50+415,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("420");

moveto(50+450,yy-50);

settextstyle(SMALL_FONT,HORIZ_DIR,4);

outtext("!");

moveto(50+445,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("450");

moveto(50+480,yy-50);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outtext("!");

moveto(50+475,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("480");

moveto(50+510,yy-50);

settextstyle(SMALL_FONT,HORIZ_DIR,4);

outtext("!");

moveto(50+505,yy-40);

settextstyle(SMALL_FONT,VERT_DIR,4);

outtext("510");

setcolor(WHITE);
line(50,yy-50-SP1_high,xx-15,yy-50-SP1_high);
setcolor(GREEN);
line(50,yy-50-SP2_high,xx-15,yy-50-SP2_high);

/*----- A/D conversion -----*/

do{
    if(i>1){
setcolor(WHITE);

line(50+X[i-1],yy-50-Y1[i-1],50+X[i],yy-50-Y1[i]);

setcolor(GREEN);

line(50+X[i-1],yy-50-Y2[i-1],50+X[i],yy-50-Y2[i]);

    }

    i++;

LoopTime=time;

```

```

time=get_time();
}

finish=time;

NextSampleTime = time+SampleInterval;

elapsed_time = how_long(start,finish);

```

```

/* start interface with PCL-812 */

```

```

/* read CH1 data */

```

```

outportb(Base_add+10,10);
outportb(Base_add+11,ENABLE);
outportb(Base_add+12,SOFTTRIG);
do{
h_byte = inportb(Base_add+5);
}while((h_byte & 0x10) != 0);
l_byte = inportb(Base_add+4);
AD1_value = h_byte*256 + l_byte;

```

```

/* read CH2 data */

```

```

outportb(Base_add+10,11);
outportb(Base_add+11,ENABLE);
outportb(Base_add+12,SOFTTRIG);
do{
h_byte=inportb(Base_add+5);
}while((h_byte & 0x10) != 0);
l_byte = inportb(Base_add+4);
AD2_value = h_byte*256 + l_byte;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

voltage_range = 10;

volt1 = ((double) AD1_value * (double) voltage_range)/4095.0 + (-5);
volt2 = ((double) AD2_value * (double) voltage_range)/4095.0 + (-5);

high1 = h1(volt1);
high2 = h2(volt2);

high1_er = SP1_high - high1;
high2_er = SP2_high - high2;

/* control signal to model */

if(high1 <= lim1) V1 = v1(high1);
if(high1 > lim1 && high1 <= lim2) V1 = v2(high1);
if(high1 > lim2) V1 = v3(high1);

if(high2 <= lim1) V2 = vv1(high2);
if(high2 > lim1 && high2 <= lim2) V2 = vv2(high2);
if(high2 > lim2) V2 = vv3(high2);

/* call regression to convert sp_high to voltage */

if(SP1_high <= lim1) V1_SP = v1(SP1_high);
if(SP1_high > lim1 && SP1_high <= lim2) V1_SP = v2(SP1_high);
if(SP1_high > lim2) V1_SP = v3(SP1_high);

if(SP2_high <= lim1) V2_SP = vv1(SP2_high);
if(SP2_high > lim1 && SP2_high <= lim2) V2_SP = vv2(SP2_high);
if(SP2_high > lim2) V2_SP = vv3(SP2_high);

delta_time = how_long(LoopTime,finish);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* error is the difference in VOLTAGE between SP-high and
   presenting height in the tube */
```

```
V1_er = V1_SP - V1;
```

```
V2_er = V2_SP - V2;
```

```
if(high1_er < 10) mn1 = V1_SP;
```

```
if(high1_er < -10) mn1 = 0;    /* valve shut off */
```

```
/* PID controller */
```

```
if(high1_er >= 10){
```

```
    en1 = V1_er/V1_SP;
```

```
    s1 = s1 + en1;
```

```
    mn1 = kp1*( td1*(en1-enold1)/delta_time + en1 + s1*delta_time/ti1 );
```

```
enold1 = en1;
```

```
}
```

```
if(high2_er < 10) mn2 = V2_SP;
```

```
if(high2_er < -10) mn2 = 0;    /* valve shut off */
```

```
if(high2_er >= 10){
```

```
    en2 = V2_er/V2_SP;
```

```
    s2 = s2 + en2;
```

```
    mn2 = kp2*( td2*(en2-enold2)/delta_time + en2 + s2*delta_time/ti2 );
```

```
enold2 = en2;
```

```
}
```

```
/*----- D/A conversion -----*/
```

```
voltage_range = 5;
```

```
/* CH1 */
```

```

h_byte = DA1_value/256;
l_byte = DA1_value%256;
outportb(Base_add+11,ENABLE);
outportb(Base_add+4,l_byte);
outportb(Base_add+5,h_byte);

/* CH2 */

DA2_value = (long int) ( mn2*4095/voltage_range );
h_byte = DA2_value/256;
l_byte = DA2_value%256;
outportb(Base_add+11,ENABLE);
outportb(Base_add+6,l_byte);
outportb(Base_add+7,h_byte);

X[i]=elapsed_time;
Y1[i]=high1;
ER1[i]=high1_er;
Y2[i]=high2;
ER2[i]=high2_er;

}while(bioskey(1)==NULL);

closegraph();

outportb(Base_add+4,0);
outportb(Base_add+5,0);
outportb(Base_add+6,0);
outportb(Base_add+7,0);

textbackground(BLACK);

textcolor(WHITE);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 init_prn();
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

open_prn();

PRINT(PRN,"%s% 5.2f\n", "          CH1 Set-point height : ",SP1_high);
PRINT(PRN,"%s% 5.5f\n", "          Kp1           : ",Kp(TimeConstant1,DelayTime1));
PRINT(PRN,"%s% 5.5f\n", "          Ti1           : ",Ti(TimeConstant1,DelayTime1));
PRINT(PRN,"%s% 5.5f\n\n", "          Td1           : ",Td(TimeConstant1,DelayTime1));
PRINT(PRN,"%s% 5.2f\n", "          CH2 Set-point height : ",SP2_high);
PRINT(PRN,"%s% 5.5f\n", "          Kp2           : ",Kp(TimeConstant2,DelayTime2));
PRINT(PRN,"%s% 5.5f\n", "          Ti2           : ",Ti(TimeConstant2,DelayTime2));
PRINT(PRN,"%s% 5.5f\n", "          Td2           : ",Td(TimeConstant2,DelayTime2));
PRINT(PRN,"%s% 5.2f\n\n", "          Sample interval : ",SampleInterval);

/*PRINT(PRN,"          Time(sec)          Height1(cm)          Error1(cm)\n");
PRINT(PRN,"          -----\n\n");
for(k=0;k<i;k++){
PRINT(PRN,"%s%5.2f%s%5.2f%s%5.2f\n", "          ",X[k], "          ",Y1[k], "
}
PRINT(PRN,"          -----\n\n");
PRINT(PRN,"          Time(sec)          Height2(cm)          Error2(cm)\n");
for(k=0;k<i;k++){
PRINT(PRN,"%s%5.2f%s%5.2f%s%5.2f\n", "          ",X[k], "          ",Y2[k], "
}
*/
PRINT(PRN,"          -----\n\n");
clos_prn();

}

/* Function to get the number of seconds elapsed since midnight on the
system clock.*/

double get_time(){
unsigned int hrs,mins,secs,hundredths;

```

```
/* request DOS service 0x2c */
```

```
inreg.h.ah = 0x2c;
```

```
/* call interrupt 0x21 with the appropriate unions */
```

```
intdos(&inreg,&outreg);
```

```
hrs = outreg.h.ch;      /* hours elapsed returned in CH */
```

```
mins = outreg.h.cl;     /* minutes elapsed returned in CL */
```

```
secs = outreg.h.dh;     /* seconds elapsed returned in DH */
```

```
hundredths = outreg.h.dl; /* 1/100's seconds returned in DL */
```

```
/* total elapsed time (in seconds) */
```

```
return ( (double) hrs*3600.0 + (double) mins*60.0 + (double) secs + (double) hundredths/100.0);
```

```
}
```

```
/* Function to compute the number of seconds elapsed between start and finish */
```

```
double how_long(double start,double finish)
```

```
{
```

```
return (finish-start);
```

```
}
```

```
/* Function to initial printer */
```

```
init_prn(){
```

```
inreg.h.ah=1;
```

```
inreg.x.dx=0;
```

```
int86(0x17,&inreg,&outreg);
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Program single control (GRAPHICS SCREEN) by PID controller */
```

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include <bios.h>
```

```
#include <dos.h>
```

```
#include <graphics.h>
```

```
#include <print.h>
```

```
#include "box.c"
```

```
#include "func1.c"
```

```
/* ----- math model equation ----- */
```

```
#define G(T,t) (1-exp(-(t/T)))
```

```
/* ----- */
```

```
#define Base_add 0x220
```

```
#define ENABLE 1
```

```
#define SOFTTRIG 1
```

```
double get_time(void);
```

```
double how_long(double,double);
```

```
double X[1000],Y[1000],ER[1000];
```

```
double SP_high;
```

```
double time=0,SampleInterval,NextSampleTime,finish,start,elapsed_time,delta_time,LoopTime;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
double TimeConstant;

double FilterGain,MathHigh,MathVolt,disturbance;
```

```
int graphdriver=DETECT,graphmode;
```

```
double x,y;
```

```
char graphpath[]="";
```

```
int k;
```

```
int CH,i=0;
```

```
union REGS inreg,outreg;
```

```
main(){
    PID();
}
```

```
PID(){
    long int  h_byte,l_byte,AD_value,DA_value,voltage_range;
    double    high,V_control,volt,high_er,V_SP,V;
    double    en,mn;
    int        MUX,lport,hport;
    int        k;
```

```
textbackground(BLUE);
```

```
clrscr();
```

```
textbackground(MAGENTA);
```

```
textcolor(WHITE);
```

```
box(10,4,70,12);
```

```
gotoxy(5,1);
```

```
cprintf("Enter channel number :"); scanf("%d",&CH);
```

```

gotoxy(5,2);

printf("Enter set-point height      : "); scanf("%lf",&SP_high);

gotoxy(5,3);

printf("Enter time constant        : "); scanf("%lf",&TimeConstant);

gotoxy(5,4);

printf("Enter filter gain (0.0-1.0) : "); scanf("%lf",&FilterGain);

gotoxy(5,5);

printf("Enter sample interval      : "); scanf("%lf",&SampleInterval);


switch(CH){
case 1: MUX=10; lport=4; hport=5; break;
case 2: MUX=11; lport=6; hport=7; break;
default: gotoxy(5,7);
textcolor(GREEN);
printf("Fatal error, program terminate itself! ");
sound(450);
getch();
nosound();
textbackground(BLACK);
textcolor(WHITE);
clrscr();
exit(1);
}

Xfil=0;

Yfil=0;

ERfil=SP_high;

time=get_time();

```

```
NextSampleTime=time+SampleInterval;
```

```
/*----- A/D conversion -----*/
```

```
#define xx getmaxx()
```

```
#define yy getmaxy()
```

```
initgraph(&graphdriver,&graphmode,graphpath);
```

```
setbkcolor(BLUE);
```

```
setlinestyle(SOLID_LINE,0,NORM_WIDTH);
```

```
line(50,15,50,yy-5);
```

```
line(15,yy-50,xx-15,yy-50);
```

```
setcolor(WHITE);
```

```
settextstyle(SMALL_FONT,HORIZ_DIR,5);
```

```
moveto(60,20);
```

```
outtext("Height (cm)");
```

```
moveto(xx-100,yy-65);
```

```
outtext("Time (sec)");
```

```
settextstyle(SMALL_FONT,HORIZ_DIR,4);
```

```
moveto(5,yy-55-20);
```

```
outtext(" 20 -");
```

```
moveto(5,yy-55-40);
```

```
outtext(" 40 -");
```

```
moveto(5,yy-55-60);
```

```
outtext(" 60 -");
```

```
moveto(5,yy-55-80);
```

```
outtext(" 80 -");
```

```
moveto(5,yy-55-100);
```

```

outtext(" 100 -");
moveto(5,yy-55-120);
outtext(" 120 -");
moveto(5,yy-55-140);
outtext(" 140 -");
moveto(5,yy-55-160);
outtext(" 160 -");
moveto(5,yy-55-180);
outtext(" 180 -");
moveto(5,yy-55-200);
outtext(" 200 -");

moveto(50+30,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(";");
moveto(50+25,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("30");
moveto(50+60,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(";");
moveto(50+55,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("60");
moveto(50+90,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext(";");
moveto(50+85,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("90");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

moveto(50+120,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+115,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("120");

moveto(50+150,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+145,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("150");

moveto(50+180,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+175,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("180");

moveto(50+210,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+205,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("210");

moveto(50+240,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+235,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("240");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

moveto(50+270,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+265,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("270");

moveto(50+300,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+295,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("300");

moveto(50+330,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+325,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("330");

moveto(50+360,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+355,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("360");

moveto(50+390,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+385,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("390");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

moveto(50+420,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+415,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("420");

moveto(50+450,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+445,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("450");

moveto(50+480,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+475,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("480");

moveto(50+510,yy-50);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtext("!");

moveto(50+505,yy-40);
settextstyle(SMALL_FONT,VERT_DIR,4);
outtext("510");

moveto(50,yy-50); /* goto origin */
setcolor(WHITE);
line(50,yy-50-SP_high,xx-15,yy-50-SP_high);
setcolor(WHITE);

```

```
do{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        y=Y[i];

        i++;

        lineto(50+x,yy-50-y);

        if(graphresult()!= grOk){

window(1,1,80,25);

clrscr();

textbackground(BLUE);

textcolor(YELLOW);

box(10,8,70,10);

gotoxy(3,1);

printf("GRAPHICS NOT ALLOWED ON YOUR PC OR MISSING *.BGI FILE.");

getch();getch();

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

}

LoopTime=time;

while(time<NextSampleTime){

time=get_time();

}

finish=time;

NextSampleTime=time+SampleInterval;

elapsed_time=how_long(start,finish);

/* start interface with PCL-812 */

/* Step 1: set the input channel by specifying the MUX scan range */

```

เอกสารนี้ `outportb(Base_add+10,MUX);` ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Step 2: write the pacer trigger mode to control register, chose mode of software trigger only */
```

```
outportb(Base_add+11,ENABLE);
```

```
/* Step 3: wait for EOC bit by reading A/D high byte */
```

```
outportb(Base_add+12,SOFTTRIG);
```

```
do{
```

```
    h_byte = inportb(Base_add+5);
```

```
}while((h_byte & 0x10) != 0);
```

```
/* Step 4: read low byte data */
```

```
l_byte = inportb(Base_add+4);
```

```
/* Step 5: data conversion */
```

```
AD_value = h_byte*256+l_byte;
```

```
/* input voltage , -5 to +5 volt */
```

```
voltage_range=10;
```

```
/* volt from PCL-812 */
```

```
volt = ((double) AD_value* (double) voltage_range)/4095.0 + (-5);
```

```
/* call regerssion function to calculate height in tank */
```

```
high = h(volt);
```

```
if(i>5){
```

```
    if(high < 10)
```

```
        high = Y[i-1];
```

```
    if(high > 170)
```

```
        high = Y[i-1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if( fabs(high-Y[i-1]) > 20 ) high = Y[i-1];
}

high_er = SP_high - high;

/* control signal to model */
if(high <= lim1) V = v1(high);
if(high > lim1 && high <= lim2) V = v2(high);
if(high > lim2) V = v3(high);

/* call regression function to convert set point height to voltage */
if(SP_high <= lim1) V_SP = v1(SP_high);
if(SP_high > lim1 && SP_high <= lim2) V_SP = v2(SP_high);
if(SP_high > lim2) V_SP = v3(SP_high);

if(high_er < 5) mn = V_SP;
if(high_er < -5) mn = 0; /* valve shut off */

/*----- IMC controller -----*/
if(high_er >= 5) {
    MathHigh = G(TimeConstant,elapsed_time)*SP_high;

    if(MathHigh <= lim1) MathVolt = v1(high);
    if(MathHigh > lim1 && high <= lim2) MathVolt = v2(high);
    if(MathHigh > lim2) MathVolt = v3(high);

    disturbance = V-MathVolt;

    en = (V_SP-disturbance);

    mn = en*FilterGain*(1/G(TimeConstant,elapsed_time));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 /*----- D/A conversion -----*/
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
voltage_range = 5; /* output voltage, 0 to +5 volt */
```

```
DA_value = (long int) (mn*4095/voltage_range);
```

```
h_byte = DA_value/256;
```

```
l_byte = DA_value%256;
```

```
outportb(Base_add+lport,ENABLE);
```

```
outportb(Base_add+lport,l_byte);
```

```
outportb(Base_add+hport,h_byte);
```

```
X[i]=elapsed_time;
```

```
Y[i]=high;
```

```
ER[i]=high_er;
```

```
}while(bioskey(1) == NULL);
```

```
outportb(Base_add+lport,0);
```

```
outportb(Base_add+hport,0);
```

```
closegraph();
```

```
textcolor(WHITE);
```

```
textbackground(BLACK);
```

```
init_prn();
```

```
open_prn();
```

```
PRINT(PRN,"%s% d\n", "          Channel number   : ",CH);
```

```
PRINT(PRN,"%s% 5.2f\n", "          Set-point height : ",SP_high);
```

```
PRINT(PRN,"%s% 5.2f\n", "          Sample interval : ",SampleInterval);
```

```
PRINT(PRN,"%s% 5.2f\n", "          Time constant   : ",TimeConstant);
```

```
PRINT(PRN,"%s% 5.2f\n", "          Filter gain     : ",FilterGain);
```

```
/*PRINT(PRN,"          Time(sec)          Height(cm)          Error(cm)\n");
```

```
PRINT(PRN,"          \n\n");
```

```
for(k=0;k<i;k++){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้เพื่อการค้า
ไม่ว่าในรูปแบบใดก็ตาม หากมีข้อผิดพลาดหรือต้องการแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PRINT(PRN,"%s%5.2f%s%5.2f%s%5.2f\n",
      "X[k],",
      "Y[k],",
      "\n");
}

PRINT(PRN,"
-----\n");
*/

clos_prn();
}

/* Function to get the number of seconds elapsed since midnight on the
   system clock.*/

double get_time(){
  unsigned int hrs,mins,secs,hundredths;
  /* request DOS service 0x2c */
  inreg.h.ah = 0x2c;

  /* call interrupt 0x21 with the appropriate unions */
  intdos(&inreg,&outreg);

  hrs = outreg.h.ch;    /* hours   elapsed returned in CH */
  mins = outreg.h.cl;   /* minutes elapsed returned in CL */
  secs = outreg.h.dh;   /* seconds elapsed returned in DH */
  hundredths = outreg.h.dl; /* 1/100's seconds returned in DL */

  /* total elapsed time (in seconds) */
  return ( (double) hrs*3600.0 + (double) mins*60.0 + (double) secs + (double) hundredths/100.0);
}

/* Function to compute the number of seconds elapsed between start and finish */

double how_long(double start,double finish)
{
  เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
  ไม่ว่าในรูปแบบใดๆ ทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
  return (finish-start);
}

```

```

}

/* Function to initial printer */
init_prn(){
inreg.h.ab=1;
inreg.x.dx=0;
int86(0x17,&inreg,&outreg);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Program to get data for V-h & h-V regression function */
```

```
#include <conio.h>
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include <dos.h>
```

```
#include <print.h>
```

```
#include "box.c"
```

```
#define Base_add 0x220
```

```
#define ENABLE 1
```

```
#define SoftTrig 1
```

```
double high,volt;
```

```
double start_point,stop_point;
```

```
double SampleInterval;
```

```
long int low_byte,high_byte,AD_value,DA_value;
```

```
long int voltage_range;
```

```
int i=0,j,choose,type;
```

```
main(){
```

```
do{
```

```
window(1,1,80,25);
```

```
textbackground(BLUE);
```

```
textcolor(YELLOW);
```

```
clrscr();
```

```
gotoxy(7,3);cprintf("Procedure of getting data for V-h & h-V regression function");
```

```
textbackground(MAGENTA);
```

```
error1:textcolor(WHITE);
```

```
box(20,6,60,16);
```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(5,2); cprintf("1. Getting V-h data");
gotoxy(5,4); cprintf("2. Getting h-V data");
gotoxy(5,6); cprintf("3. Return to main menu");
textcolor(GREEN+BLINK);
gotoxy(8,8); cprintf("SELECT ==> ");
scanf("%d",&choose);
if(choose == 1){
error2:      textcolor(WHITE);
box(20,6,60,16);
gotoxy(5,2); cprintf("1. Run model singlely.");
gotoxy(5,4); cprintf("2. Run model simultaneously.");
gotoxy(5,6); cprintf("3. Return to main menu");
textcolor(GREEN+BLINK);
gotoxy(8,8); cprintf("SELECT ==> ");
scanf("%d",&type);
if(type==1) Single_V_h();
if(type==2) Simultaneous_V_h();
if(type==3){
window(1,1,80,25);
textbackground(BLACK);
textcolor(WHITE);
clrscr();
break;
}
if((type < 1) || (type > 3)){
sound(450);
delay(200);
nosound();
goto error2;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

if(choose == 2) h_V();

if((choose < 1) || (choose > 3)){

sound(450);

delay(200);

nosound();

goto error1;

}

}while(choose !=3);

window(1,1,80,25);

textbackground(BLACK);

textcolor(WHITE);

clrscr();

}

/* Function to generate voltage and measure height */
Single_V_h(){

int CH,lport,hport,MUX,l=1;

window(1,1,80,25);

textbackground(BLUE);

clrscr();

textbackground(YELLOW);

textcolor(WHITE);

box(10,5,70,20);

gotoxy(5,1);

printf("Enter channel number (1-2)          : "); scanf("%d",&CH);

gotoxy(5,2);

printf("Enter start voltage  (-5 to +5 volt) : "); scanf("%lf",&start_point);

gotoxy(5,3);

printf("Enter stop voltage   (-5 to +5 volt) : "); scanf("%lf",&stop_point);

```

```

gotoxy(5,4);

printf("Enter sample interval          : "); scanf("%lf",&SampleInterval);
switch(CH){
case 1: lport=4; hport=5; MUX=10; break;
case 2: lport=6; hport=7; MUX=11; break;
        default: gotoxy(5,5);
printf("Fatal error, program terminate !");
sound(450);
getch();
nosound();
textbackground(BLACK);
textcolor(WHITE);
clrscr();
exit(1);
}
window(1,1,80,25);
textbackground(BLUE);
textcolor(YELLOW);
clrscr();
gotoxy(10,8);
printf("Procedure of getting data for V-h regression function.");
textbackground(MAGENTA);
textcolor(WHITE);
open_prn();
PRINT(PRN,"%s%d%s", "    CH : ",CH,"          V-h data\n\n");
PRINT(PRN,"          ----- \n");
PRINT(PRN,"          VOLT          HEIGHT\n");
PRINT(PRN,"          ----- \n\n");
do{

```

```

gotoxy(5,1);

cprintf("%s%5.2f","Now,setting voltage at : ",start_point);

gotoxy(5,3);

textcolor(GREEN+BLINK);

cprintf("Press any key if you are ready.");

getch();

textcolor(WHITE);

outportb(Base_add+10,MUX);

outportb(Base_add+11,ENABLE);

outportb(Base_add+12,SoflTrig);

do{

    high_byte = inportb(Base_add+5);

}while(( high_byte & 0x10) != 0);

low_byte = inportb(Base_add+4);

AD_value = high_byte*256 + low_byte;

DA_value = AD_value;

high_byte= DA_value/256;

low_byte = DA_value%256;

outportb(Base_add+11,ENABLE);

outportb(Base_add+lport,low_byte);

outportb(Base_add+hport,high_byte);

box(5,15,75,18);

gotoxy(5,1);

cprintf("Coresponding hieght (at steady-state) = ");

scanf("%lf",&high);

PRINT(PRN,"%d%s% 5.3f%s% 5.5f\n",l++,".          ",start_point,"          ",high);

outportb(Base_add+lport,0);

outportb(Base_add+hport,0);

textbackground(WHITE);

```

```

box(5,15,75,18);

gotoxy(1,1);

cprintf("If water is drained ");

gotoxy(20,1);

textcolor(YELLOW+BLINK);

cprintf(" TOTALLY ");

gotoxy(28,1);

textcolor(RED);

cprintf(" from tank then press any key to continue.");

getch();

window(5,15,75,18);

textbackground(BLUE);

clrscr();

textbackground(MAGENTA);

textcolor(WHITE);

start_point = start_point + SampleInterval;

    while(start_point <= stop_point);

    PRINT(PRN," -----\n\n");

    clos_prn();

}

```

```

Simultaneous_V_h(){

double  high1,high2;

int      l=1;

window(1,1,80,25);

textbackground(BLUE);

clrscr();

textbackground(YELLOW);

textcolor(WHITE);

```

```

box(10,5,70,20);

```

```

gotoxy(5,1);

printf("Enter start voltage (-5 to +5 volt) : "); scanf("%lf",&start_point);

gotoxy(5,2);

printf("Enter stop voltage (-5 to +5 volt) : "); scanf("%lf",&stop_point);

gotoxy(5,3);

printf("Enter sample interval : "); scanf("%lf",&SampleInterval);

window(1,1,80,25);

textbackground(BLUE);

textcolor(YELLOW);

clrscr();

gotoxy(10,8);

printf("Procedure of getting data for V-h regression function.");

textbackground(MAGENTA);

textcolor(WHITE);

open_prn();

PRINT(PRN,"%s","Simultaneous: V-h data\n\n");

PRINT(PRN,"-----\n");

PRINT(PRN," VOLT CH1-HEIGHT CH2-HEIGHT \n");

PRINT(PRN,"-----\n\n");

do{

box(5,9,75,13);

gotoxy(5,1);

printf("%s%5.2f","Now,setting CH1 voltage at : ",start_point);

gotoxy(5,3);

textcolor(GREEN+BLINK);

printf("Press any key if you are ready.");

getch();

textcolor(WHITE);

outportb(Base_add+10,10);

```

```

outportb(Base_add+12,SoftTrig);

do{

    high_byte = inportb(Base_add+5);

}while(( high_byte & 0x10) != 0);

low_byte = inportb(Base_add+4);

AD_value = high_byte*256 + low_byte;

DA_value = AD_value;

high_byte = DA_value/256;

low_byte = DA_value%256;

/*----- CH1 -----*/

    outportb(Base_add+11,ENABLE);

outportb(Base_add+4,low_byte);

outportb(Base_add+5,high_byte);

/*----- CH2 -----*/

outportb(Base_add+11,ENABLE);

outportb(Base_add+6,low_byte);

outportb(Base_add+7,high_byte);


box(5,15,75,18);

gotoxy(5,1);

cprintf("Coresponding CH1-hieght (at steady-state) = ");

scanf("%lf",&high1);

gotoxy(5,2);

cprintf("Coresponding CH2-hieght (at steady-state) = ");

scanf("%lf",&high2);

PRINT(PRN,"%d%s% 5.3f%s% 5.5f%s% 5.5f\n",l++,",",start_point,",high1,"

outportb(Base_add+4,0);

outportb(Base_add+5,0);

```

```

outportb(Base_add+7,0);

textbackground(WHITE);

textcolor(RED);

box(5,15,75,18);

gotoxy(1,1);

printf("If water is drained ");

gotoxy(20,1);

textcolor(YELLOW+BLINK);

printf(" TOTALLY ");

gotoxy(28,1);

textcolor(RED);

printf(" from tank then press any key to continue.");

getch();

    window(5,15,75,18);

textbackground(BLUE);

clrscr();

textbackground(MAGENTA);

textcolor(WHITE);

start_point = start_point + SampleInterval;

    }while(start_point <= stop_point);

    PRINT(PRN," -----\\n\\n");

    clos_prn();

}

```

```

/* Function to generate height and measure voltage */

```

```

h_V(){

int CH,hport,lport,MUX,l=1;

window(1,1,80,25);

textbackground(BLUE);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

textbackground(YELLOW);

textcolor(WHITE);

box(10,5,70,20);

gotoxy(5,1);

cprintf("Enter channel number (1-2): "); scanf("%d",&CH);

gotoxy(5,2);

cprintf("Enter start hieght      : "); scanf("%lf",&start_point);

gotoxy(5,3);

cprintf("Enter stop hieght      : "); scanf("%lf",&stop_point);

gotoxy(5,4);

cprintf("Enter sample interval   : "); scanf("%lf",&SampleInterval);

switch(CH){

case 1: lport=4; hport=5; MUX=10; break;

case 2: lport=6; hport=7; MUX=11; break;

default: gotoxy(5,5);

cprintf("Fatal error, program terminate !");

sound(450);

getch();

nosound();

textbackground(BLACK);

textcolor(WHITE);

clrscr();

exit(1);

}

window(1,1,80,25);

textbackground(BLUE);

textcolor(YELLOW);

clrscr();

gotoxy(10,8);

```

```

textbackground(MAGENTA);

textcolor(WHITE);

open_prn();

PRINT(PRN,"%s%d%s",      CH : ",CH,"          h-v data\n\n");

PRINT(PRN,"      -----\\n");

PRINT(PRN,"          HEIGHT          VOLT\\n");

PRINT(PRN,"      -----\\n\\n");

do{

box(5,9,75,13);

gotoxy(5,1);

cprintf("%s%5.2f","Now,setting height at : ",start_point);

gotoxy(5,3);

textcolor(GREEN+BLINK);

cprintf("Press any key if you are ready.");

getch();

textcolor(WHITE);

    outputb(Base_add+10,MUX);

outputb(Base_add+11,ENABLE);

outputb(Base_add+12,SoftTrig);

do{

    high_byte = inportb(Base_add+5);

}while(( high_byte & 0x10) != 0);

low_byte = inportb(Base_add+4);

AD_value = high_byte*256 + low_byte;

DA_value = AD_value;

high_byte= DA_value/256;

low_byte = DA_value%256;

outputb(Base_add+11,ENABLE);

outputb(Base_add+lport,low_byte);

outputb(Base_add+hport,high_byte);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

box(5,15,75,18);

gotoxy(5,1);

cprintf("Coresponding voltage : ");

scanf("%lf",&volt);

PRINT(PRN,"%d%s% 5.3f%s% 5.5f\n",l++,".",start_point,"",volt);

clrscr();

start_point = start_point+SampleInterval;

}while(start_point<=stop_point);

PRINT(PRN," -----\n\n");

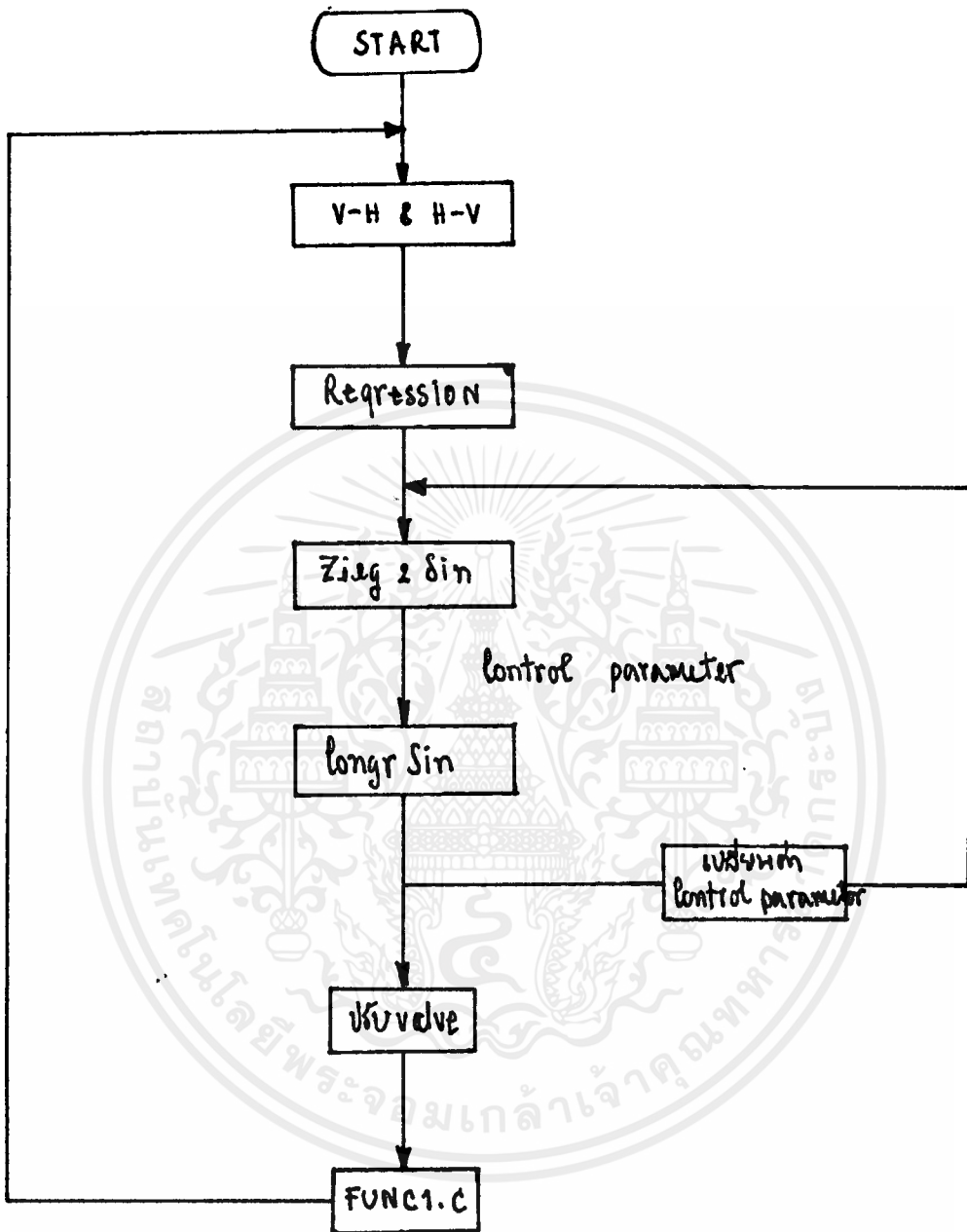
clos_prn();

}

```

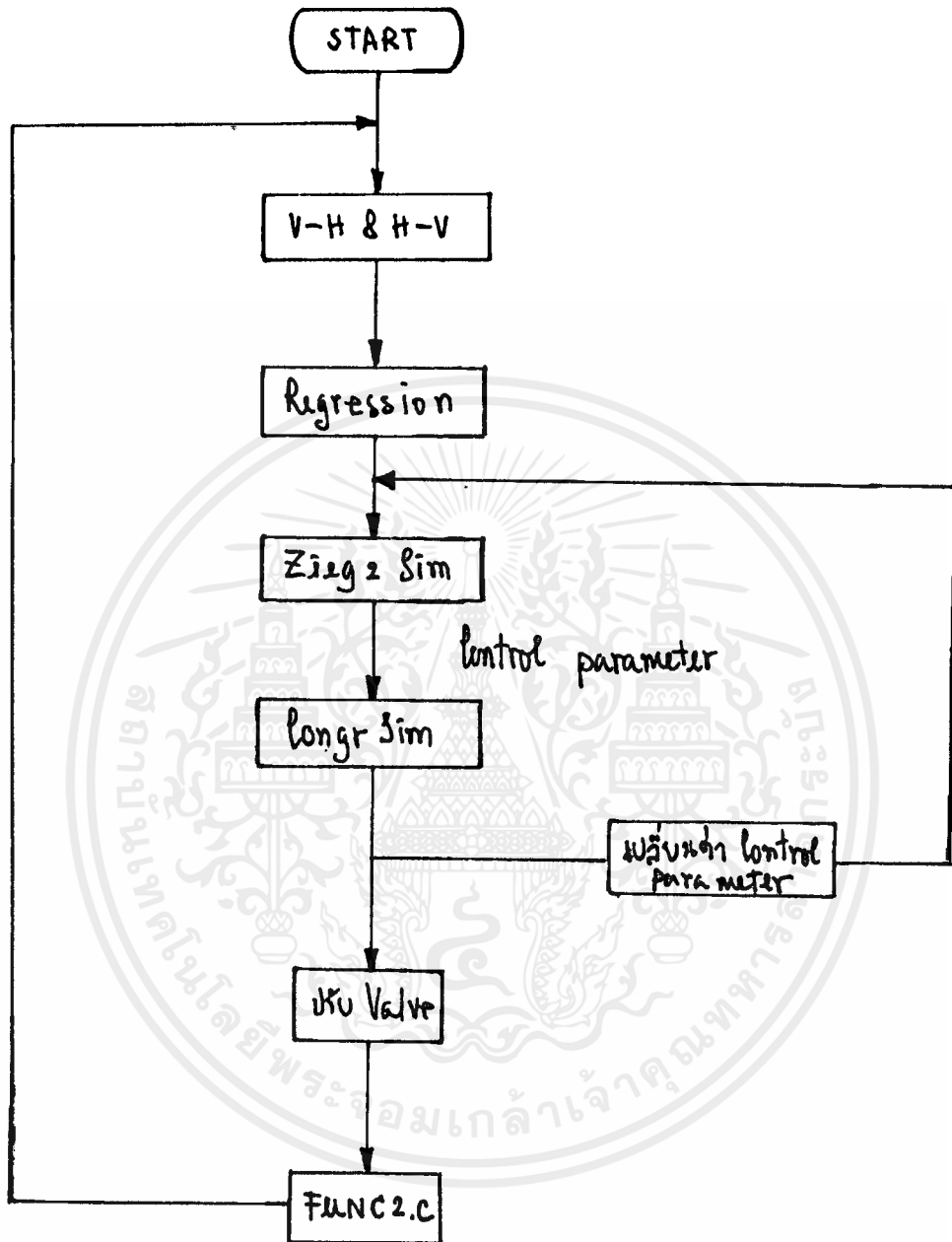


การควบคุมแบบ 1 หอ



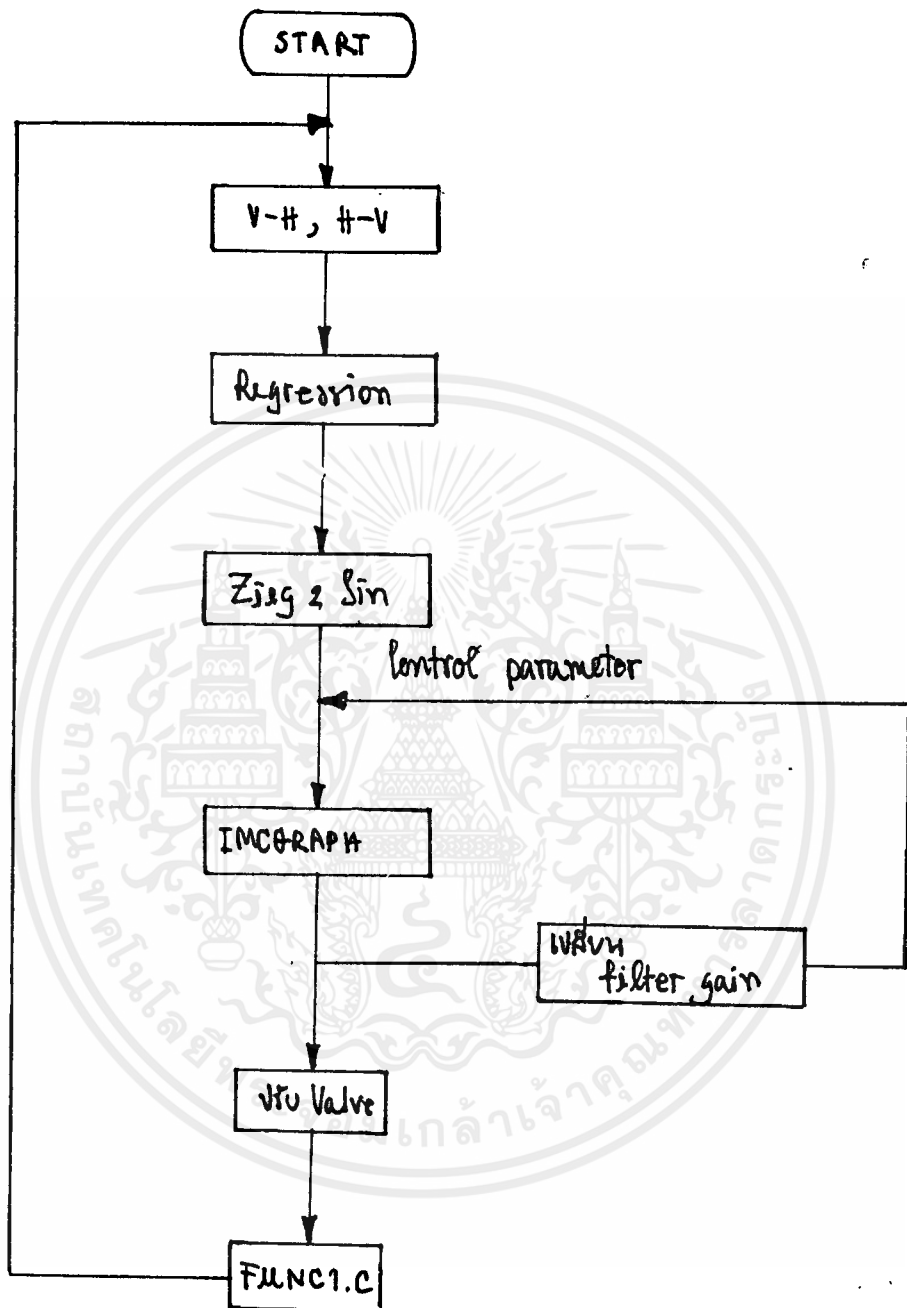
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบบ 2 ท่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบบไม่เดลอ้าริง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

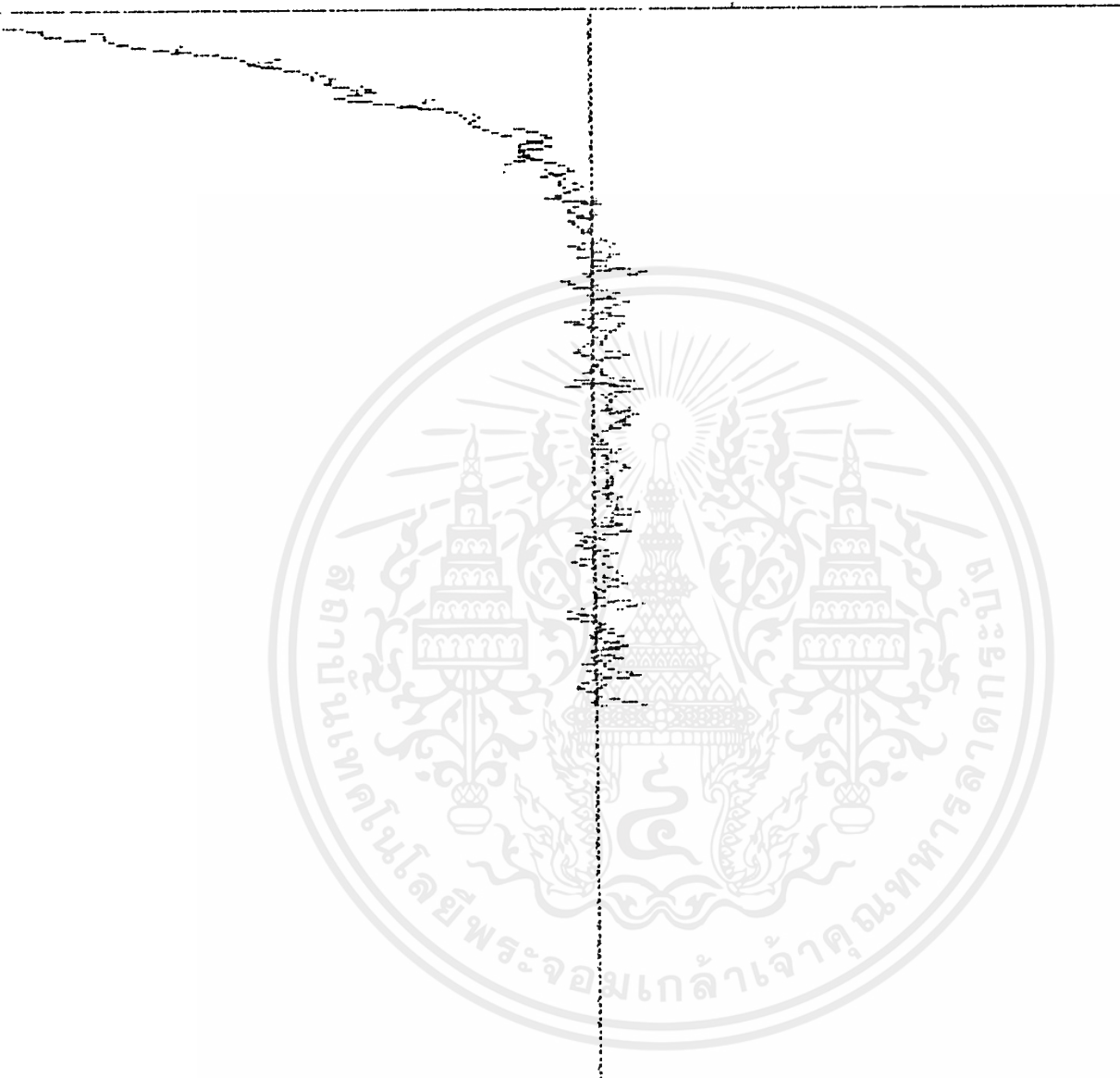
ผลการทดลองแบ่งออกเป็นตอน ๆ ดังนี้

1. เป็นการทดลอง การควบคุมน้ำในท่อเดียวแบบ PID ซึ่งได้แก่กราฟในรูปที่ 1 ถึง 12 แต่ในกราฟรูปที่ 9 ถึง 12 นั้น เป็นการทดลอง disturbance โดยมีการใส่น้ำจากนอกระบบเข้าไป เพื่อดูผลการควบคุม
2. เป็นการทดลอง การควบคุมน้ำสองท่อพร้อมกัน แบบ PID โดยใช้ระบบควบคุมเดียวกัน โดยแบ่ง 2 ช่วง ใหญ่ ๆ คือ ช่วงแรกเป็นการเปลี่ยน control parameter ซึ่งได้แก่กราฟในรูปที่ 13 ถึง 19 ส่วนอีกช่วงหนึ่งเป็นการทดลอง non-linear effect โดยมีการเปิดวาล์วที่เชื่อมระหว่างท่อทั้งสองให้น้ำในท่อทั้งสองไหลถึงกัน ซึ่งได้แก่ กราฟในรูปที่ 20 และ 21
3. เป็นการทดลองการควบคุมระบบแบบ Model Reference Control โดยมีการทดลองเปลี่ยน control parameter และ time constants ของระบบ ซึ่งได้แก่กราฟในรูปที่ 22 ถึง 30

ผลการทดลอง : การควบคุมระดับน้ำแบบ 1 ท่อ

Graph No.	Set point (cm.)	ch. No.	Time Const. (sec.)	Delay time (sec.)	K _p	T _i	T _d	Disturbance (cm ³ /sec)
1	150	1	74.0	0.81	109.6296	1.62	0.405	-
2	150	1	74.0	0.81	219.2592	3.24	0.810	-
3	150	1	74.0	0.81	438.5185	6.48	1.620	-
4	150	1	74.0	0.81	877.0370	12.96	3.240	-
.....								
5	185	2	46.22	0.81	68.4740	1.62	0.405	-
6	185	2	46.22	0.81	136.9481	3.24	0.810	-
.....								
7	150	1	98.13	1.53	77.1664	3.05	0.763	-
8	150	1	115.81	2.14	64.9401	4.28	1.07	-
.....								
9	150	1	74.0	1.38	64.3478	2.76	0.69	13.88
10	150	1	74.0	1.38	64.3478	2.76	0.69	118.02
11	150	1	74.0	1.38	64.3478	2.76	0.69	278.15
12	150	1	74.0	1.38	64.3478	2.78	0.69	518.86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



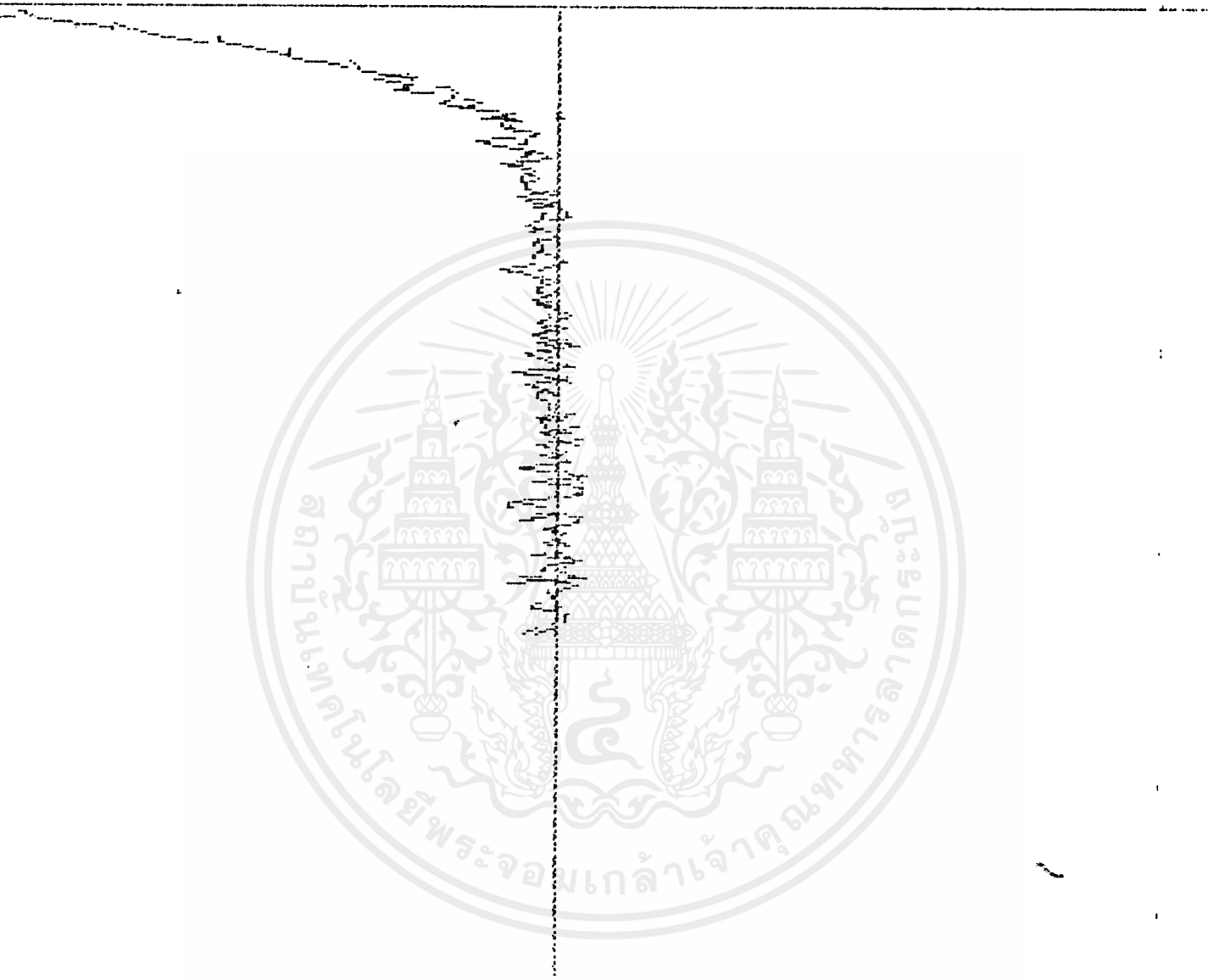
channel number : 1
st-point height : 150.00
ample interval : 2.00
me constant : 74.00
elay time : 0.81
 : 109.62963
 : 1.62000
 : 0.40500

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Channel number : 1
 Antenna-point height : 150.00
 Sample interval : 2.00
 Time constant : 74.00
 Delay time : 0.81
 : 219.25926
 : 3.24000
 : 0.81000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



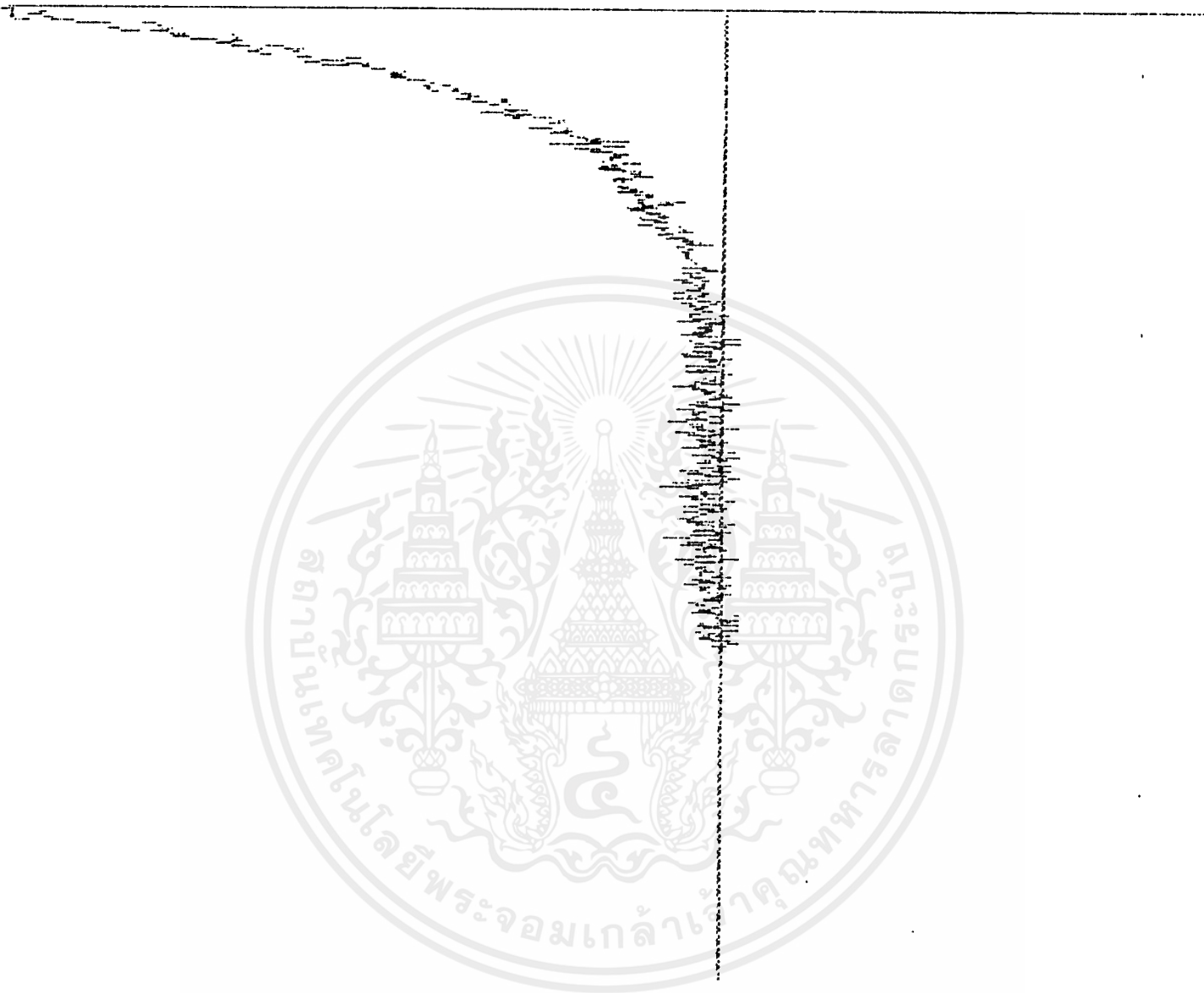
channel number : 1
set-point height : 150.00
sample interval : 2.00
time constant : 74.00
delay time : 0.81
 : 438.51852
 : 6.48000
 : 1.62000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Channel number : 1
 Set-point height : 150.00
 Sample interval : 2.00
 Time constant : 74.00
 Delay time : 0.81
 kp : 877.03704
 Ti : 12.96000
 Td : 3.24000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



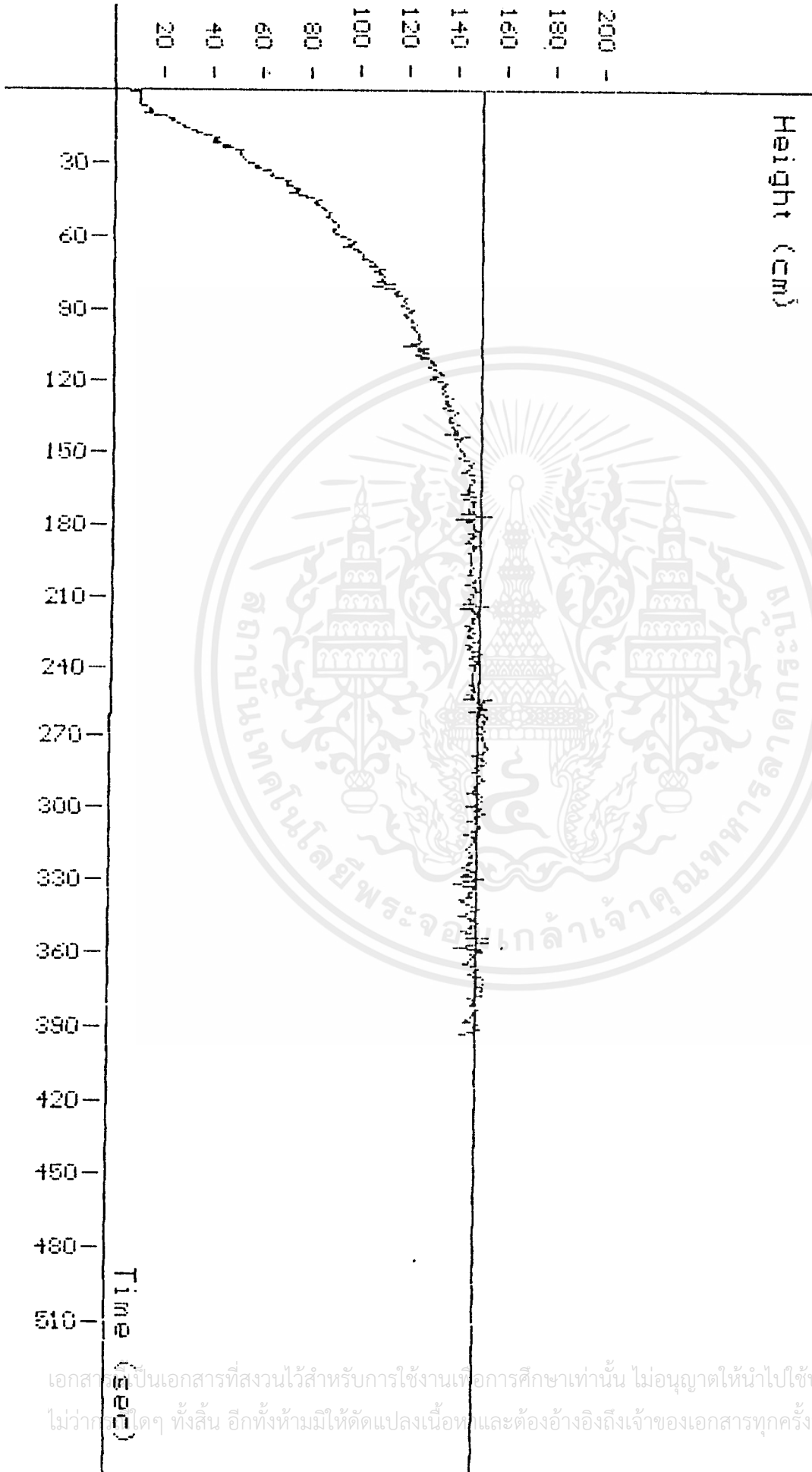
Channel number : 2
 Set-point height : 185.00
 Sample interval : 1.00
 Time constant : 46.22
 Delay time : 0.81
 Kp : 68.47407
 Ti : 1.62000
 Td : 0.40500

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Channel number : 2
 Set-point height : 185.00
 Sample interval : 1.00
 Time constant : 46.22
 Delay time : 0.81
 Kp : 136.94815
 Ti : 3.24000
 Td : 0.81000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

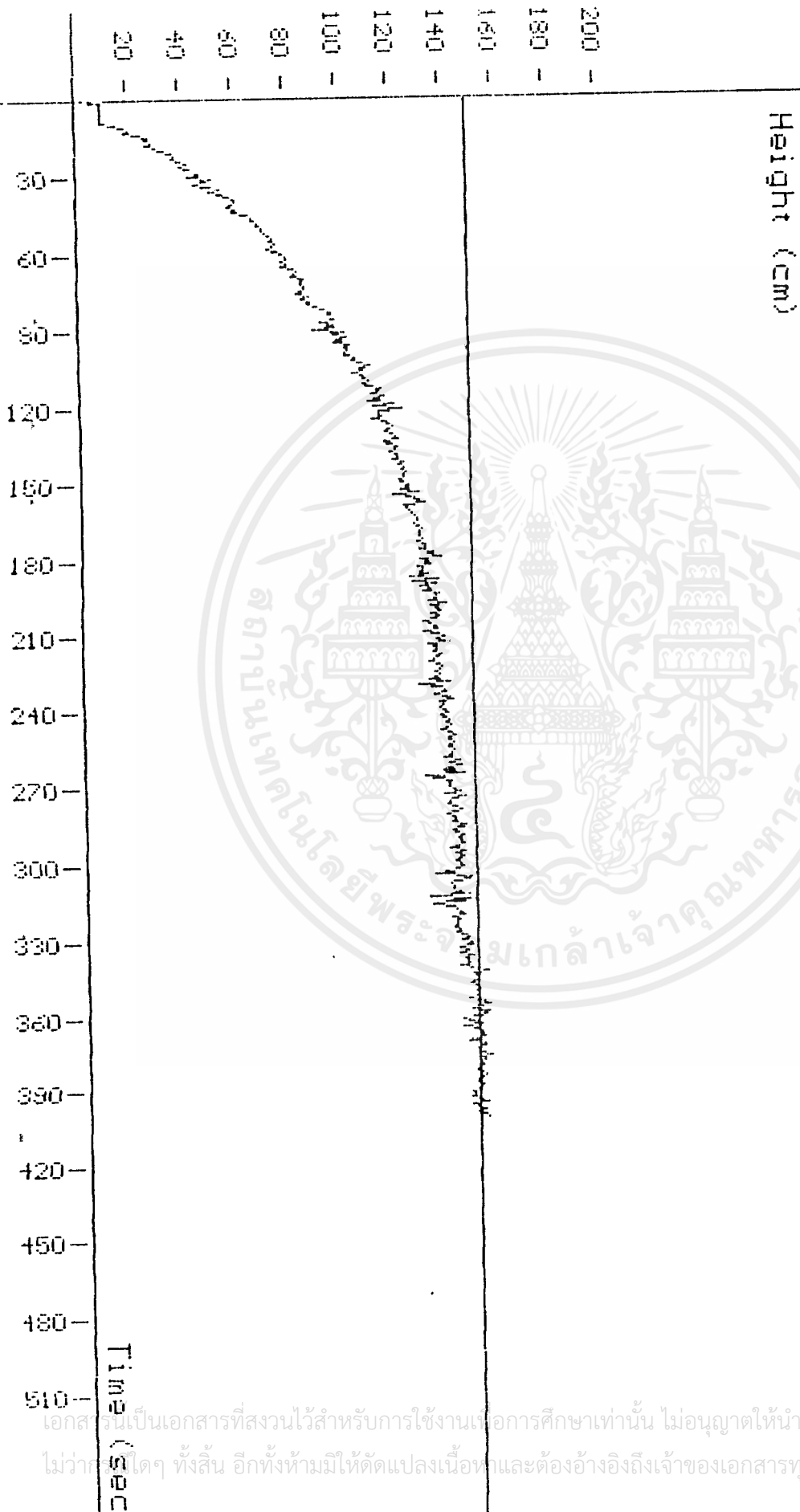


Channel number : 1
Set-point height : 150.00
Sample interval : 1.00
Time constant : 98.13
Delay time : 1.53
Kp : 77.16645
Ti : 3.05200
Td : 0.76300

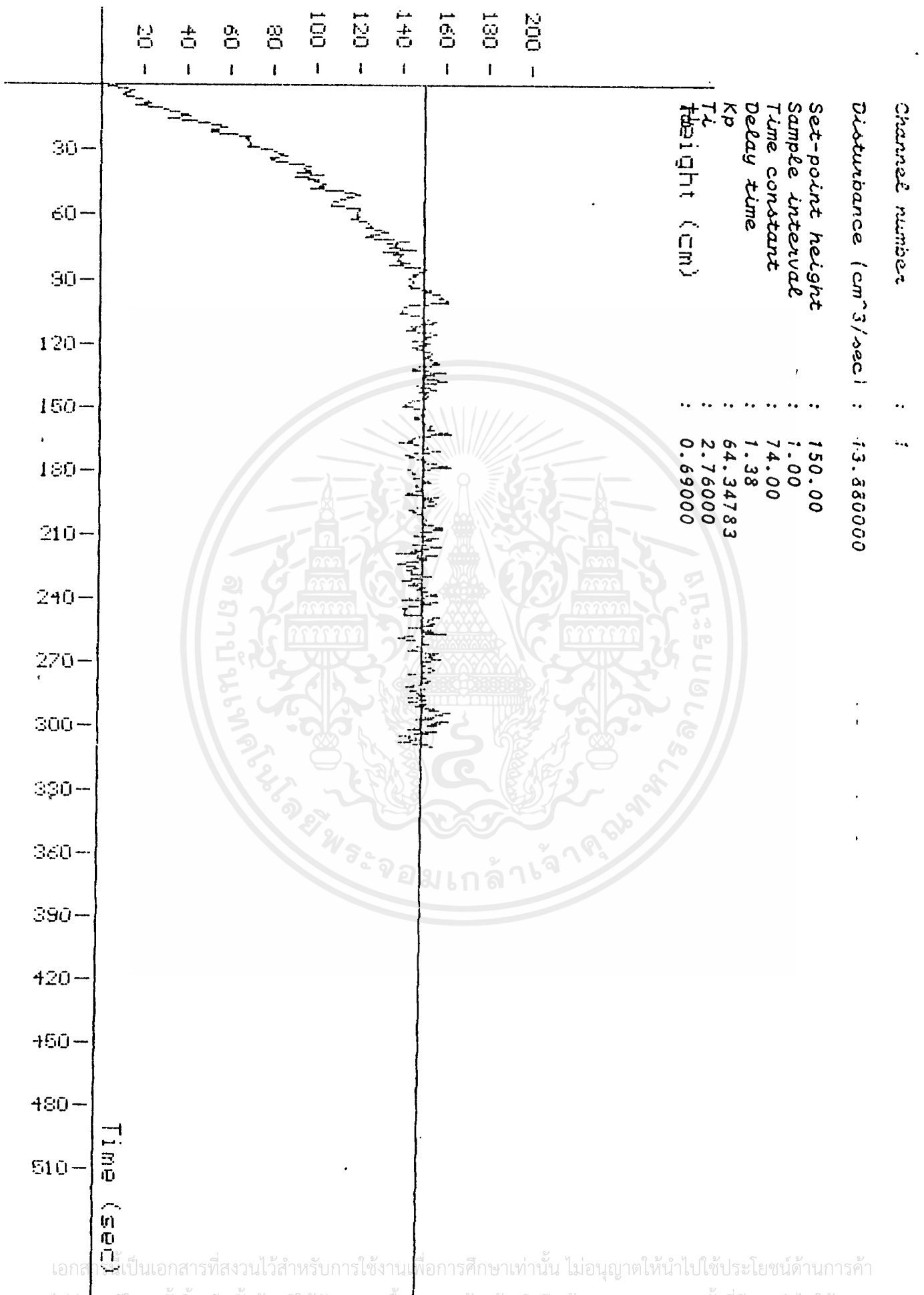
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

channel number : 1
 Set-point height : 150.00
 Sample interval : 1.00
 Time constant : 115.81
 Delay time : 2.14
 KP : 64.94018
 TI : 4.28
 TD : 1.07

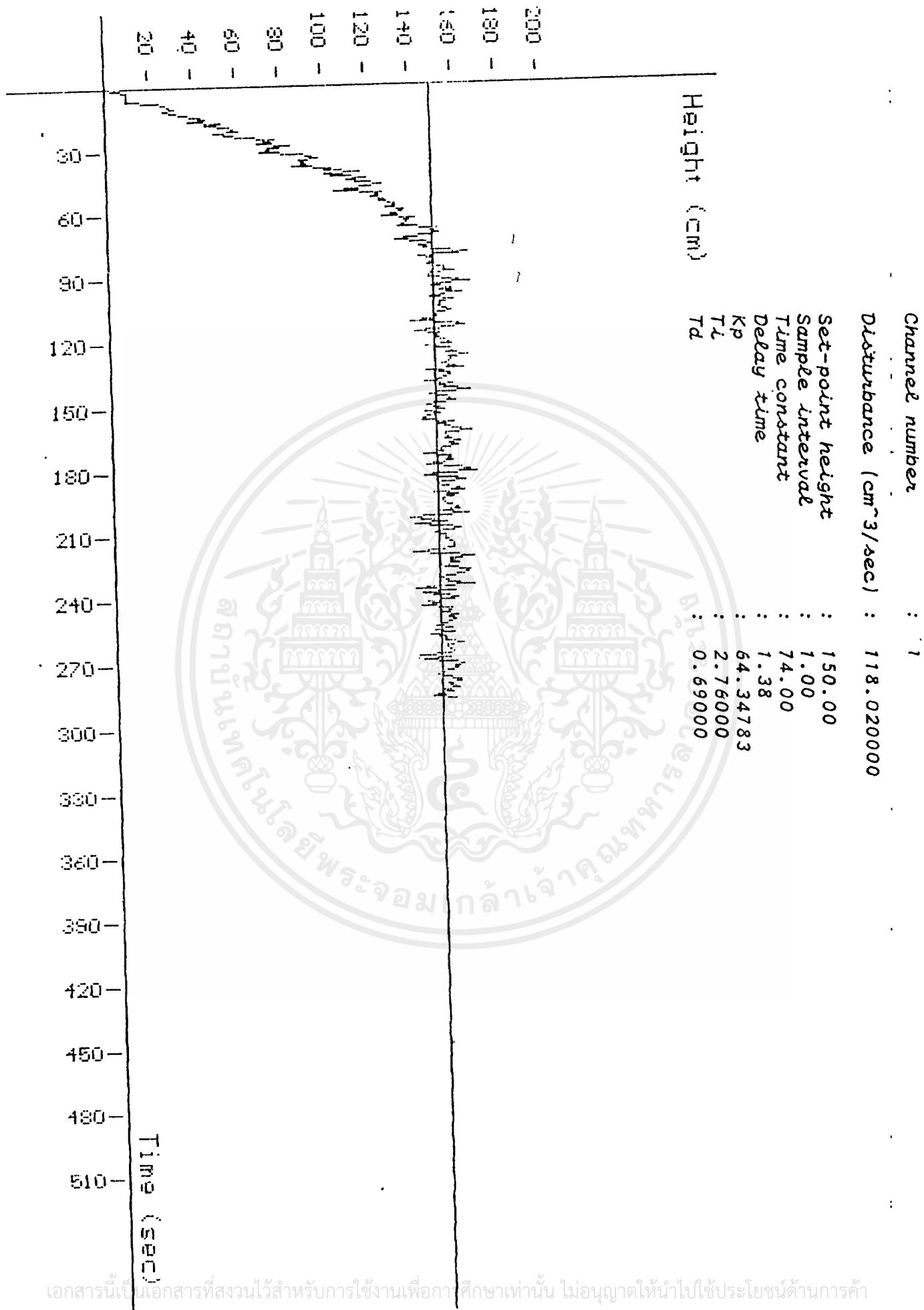
DELAYTIME = 2.14



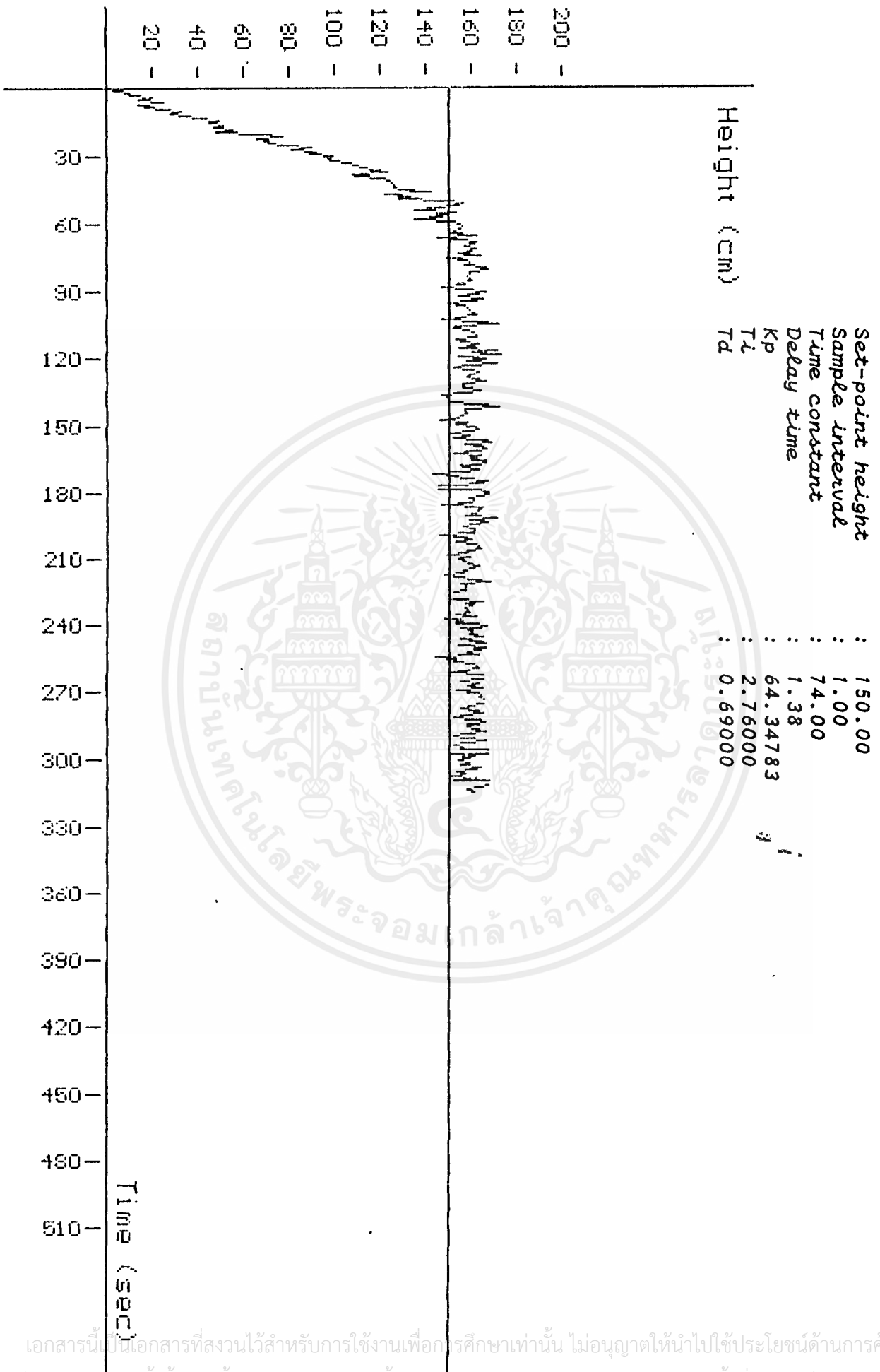
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าในรูปแบบใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

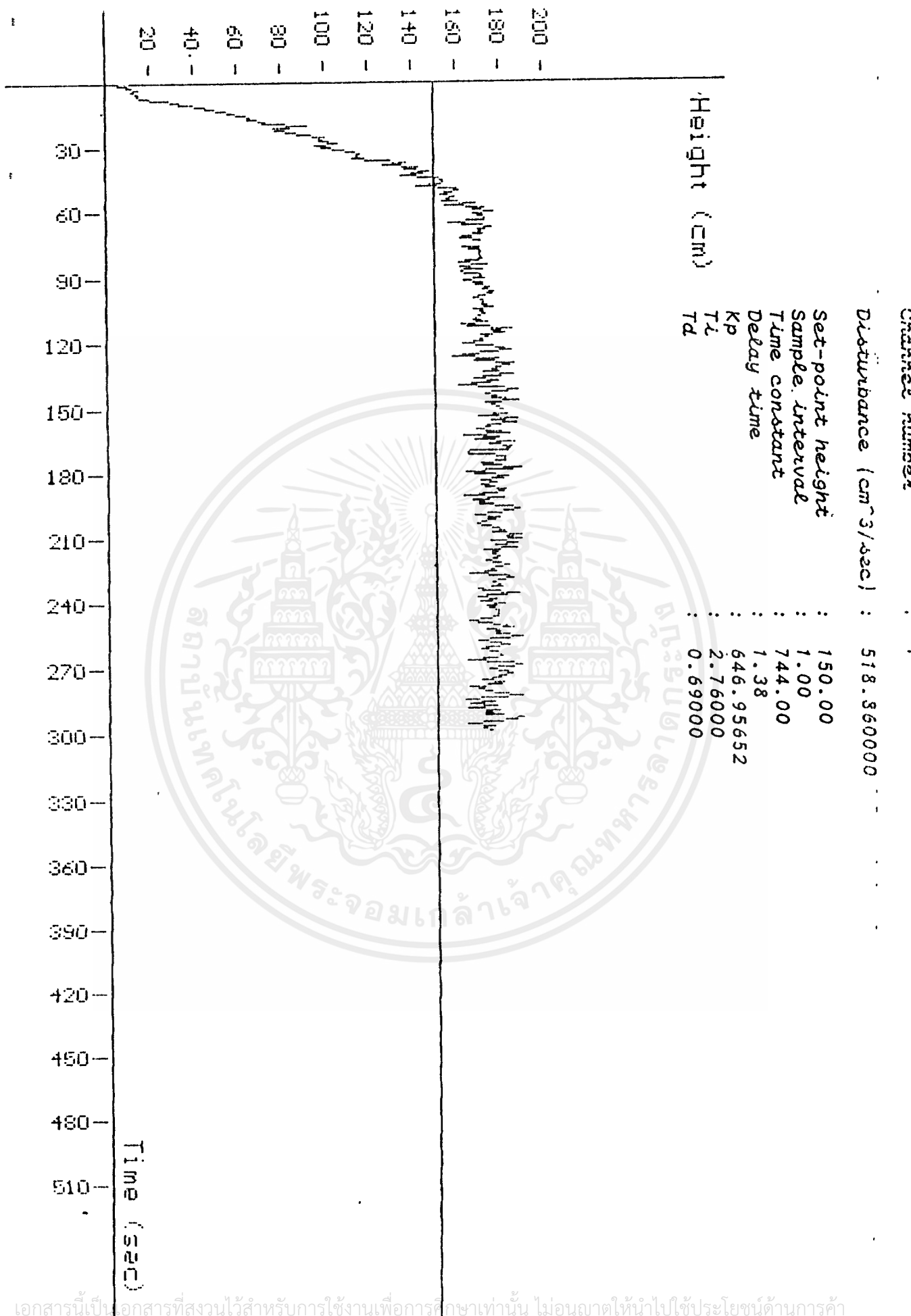


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



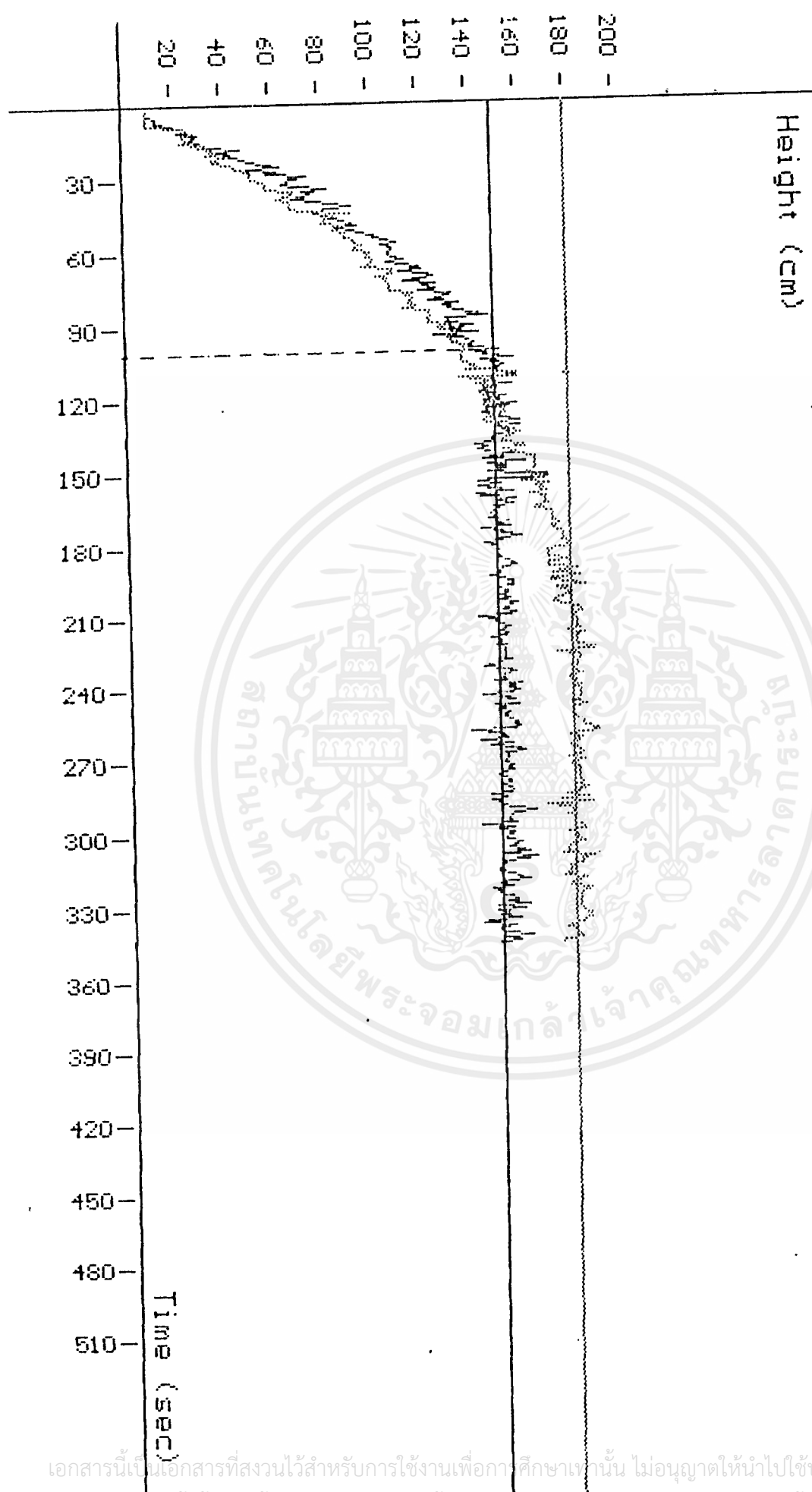
Channel number : 1
 Distance (cm³/sec) : 278.150000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

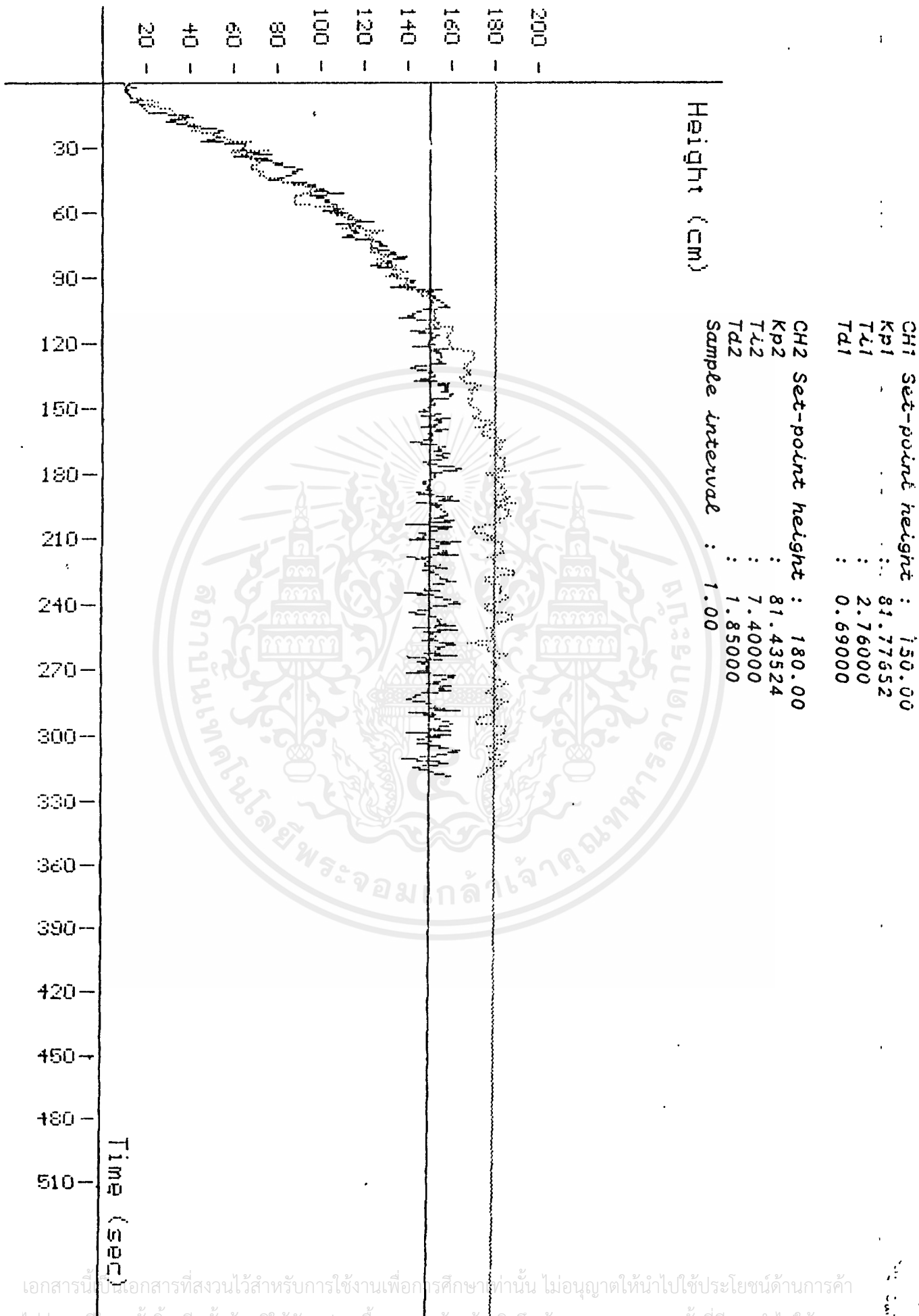
Graph No.	channel # 1					channel # 2					หมายเหตุ			
	set point (cm.)	time const. (sec.)	delay time (sec.)	K_p	T_i	T_d	det point (cm.)	Time const. (sec.)	delay time (sec.)	K_p		T_i	T_d	
กลุ่มที่ 1	13	150	94.043	1.38	81.77652	2.76	0.69	180	62.773	1.85	40.71762	3.7	0.925	
	14	150	94.043	1.38	81.77652	2.76	0.69	180	62.773	1.85	81.43524	7.4	1.85	
	15	150	94.043	1.38	81.77652	2.76	0.69	180	62.773	1.85	162.87049	14.8	3.70	
														
กลุ่มที่ 2	16	150	94.043	1.38	163.55304	5.52	1.38	180	62.773	1.85	40.71762	3.7	0.925	
	17	150	94.043	1.38	327.10609	11.04	2.76	180	62.773	1.85	40.71762	3.7	0.925	
กลุ่มที่ 3	18	150	103.35	1.526	81.2713	3.052	0.76	180	62.773	1.85	40.71762	3.7	0.925	
	19	150	110.95	2.14	62.21495	4.280	1.07	180	62.773	1.85	40.71762	3.7	0.925	
														
กลุ่มที่ 4	20	150	94.043	1.38	81.77652	2.76	0.69	180	62.773	1.85	40.71762	3.7	0.925	เปิดวาล์วเต็ม ระหว่างท่อไว้ ครึ่งหนึ่งของค่า ที่ตั้งเปิด
														
21	150	94.043	1.38	81.77652	2.76	0.69	180	62.773	1.85	40.71762	3.7	0.925	เปิดวาล์วเต็ม ระหว่างท่อไว้	



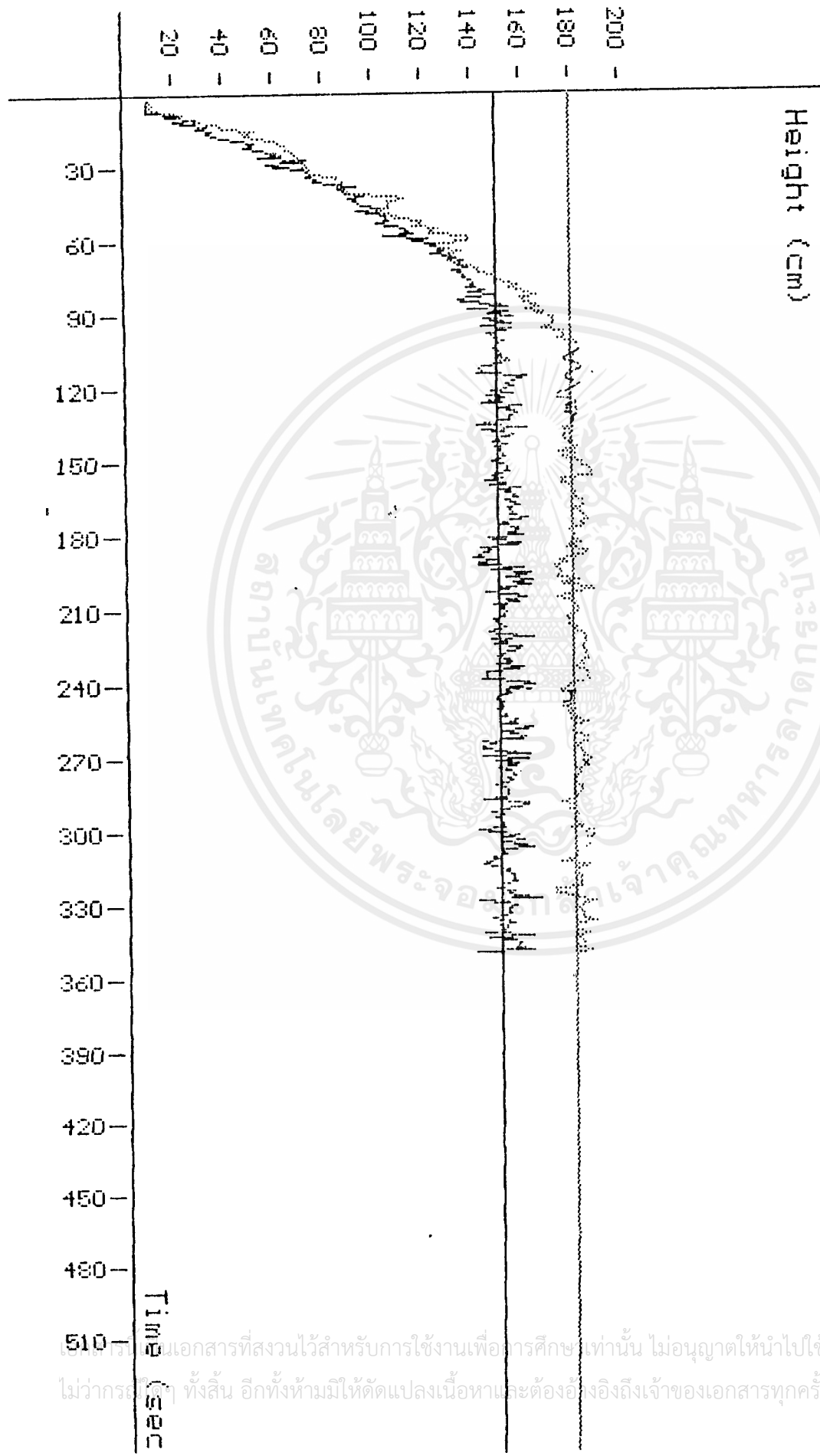
CH1 Set-point height : 150.00
Kp1 : 81.717652
Td1 : 2.76000
Td1 : 0.69000

CH2 Set-point height : 180.00
Kp2 : 40.71762
Td2 : 3.70000
Td2 : 0.92500
Sample interval : 1.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



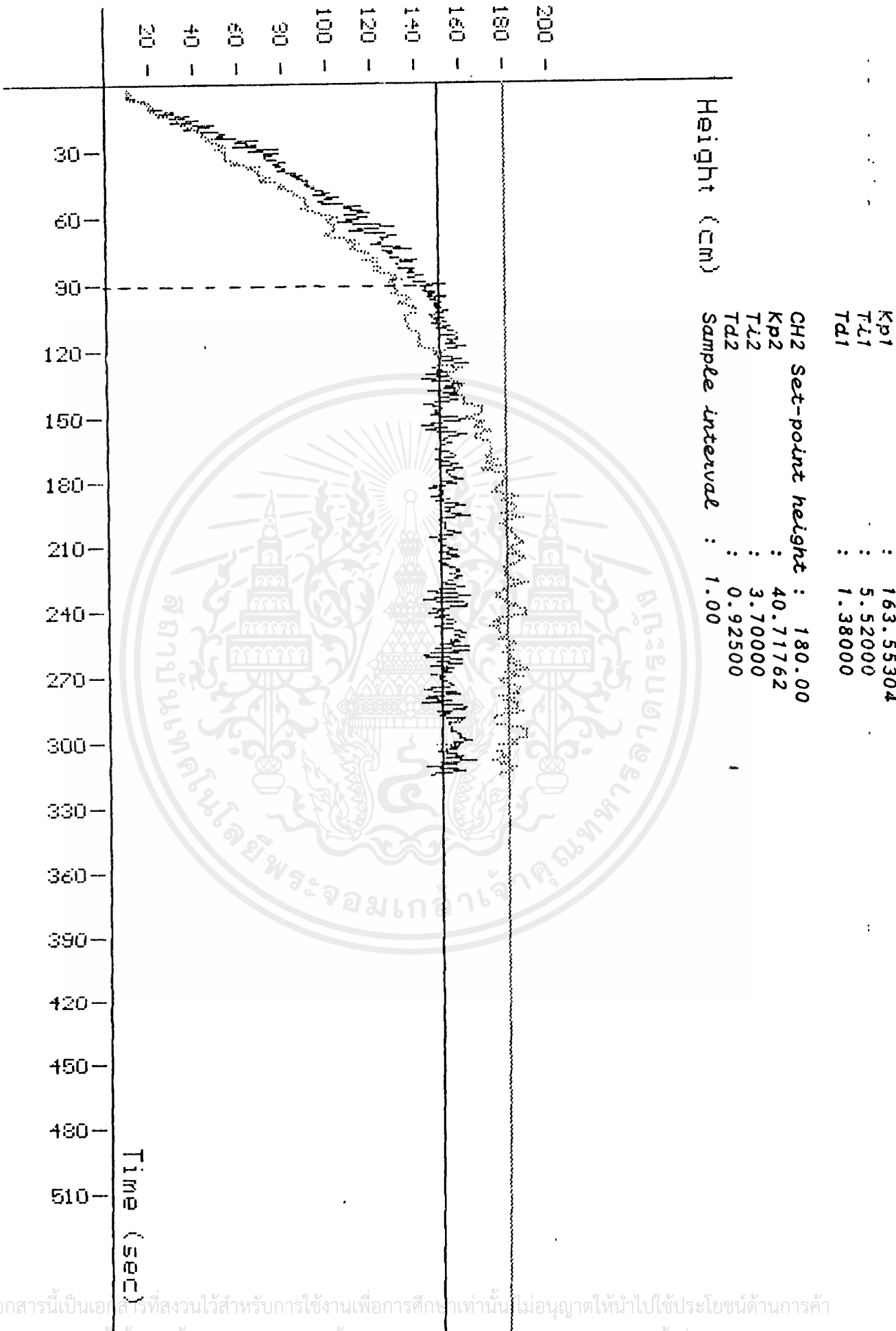
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



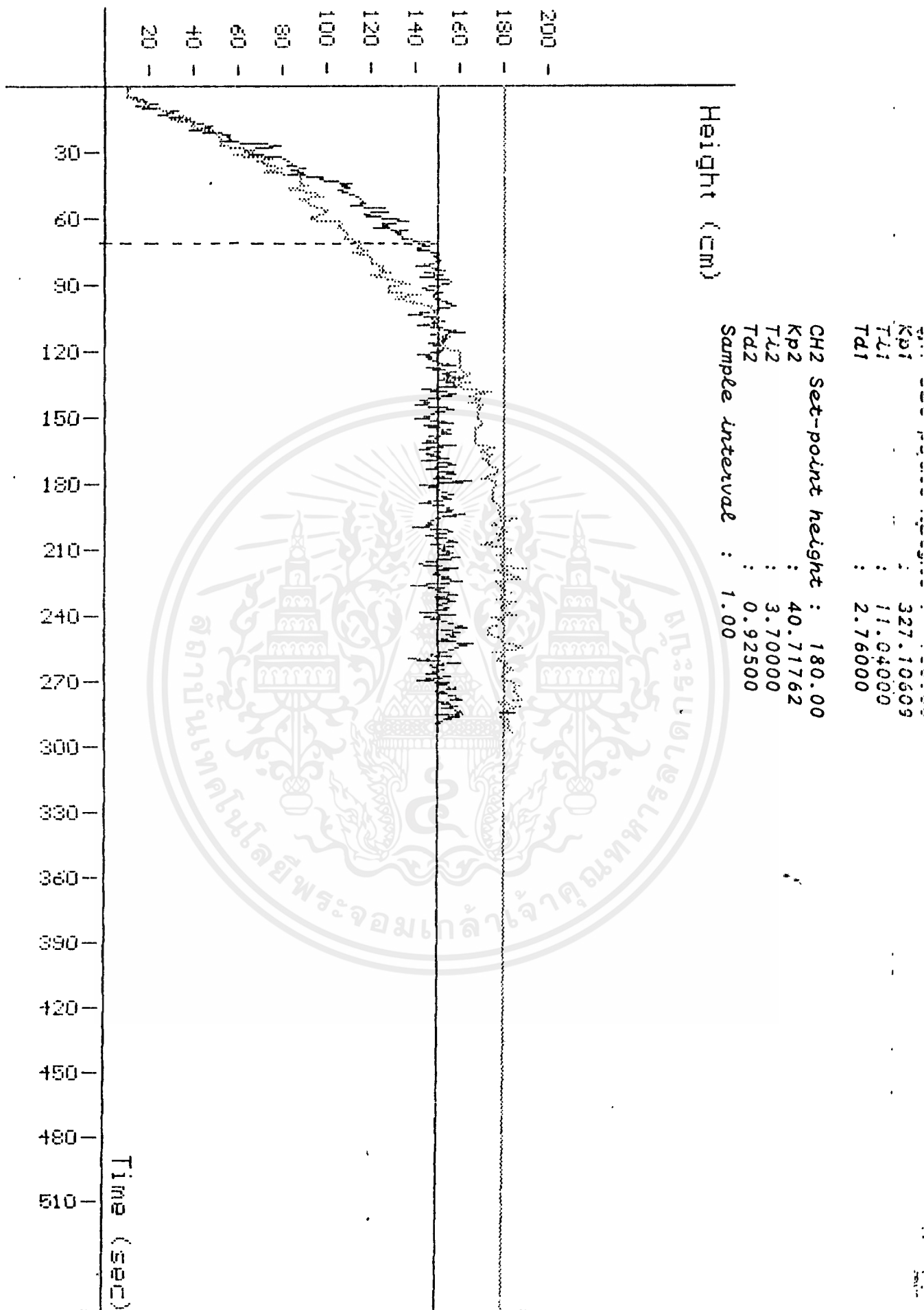
Kp1 : 31.77652
Td1 : 2.76000
Td1 : 0.69000

CH2 Set-point height : 180.00
Kp2 : 162.87049
Td2 : 14.80000
Td2 : 3.70000
Sample interval : 1.00

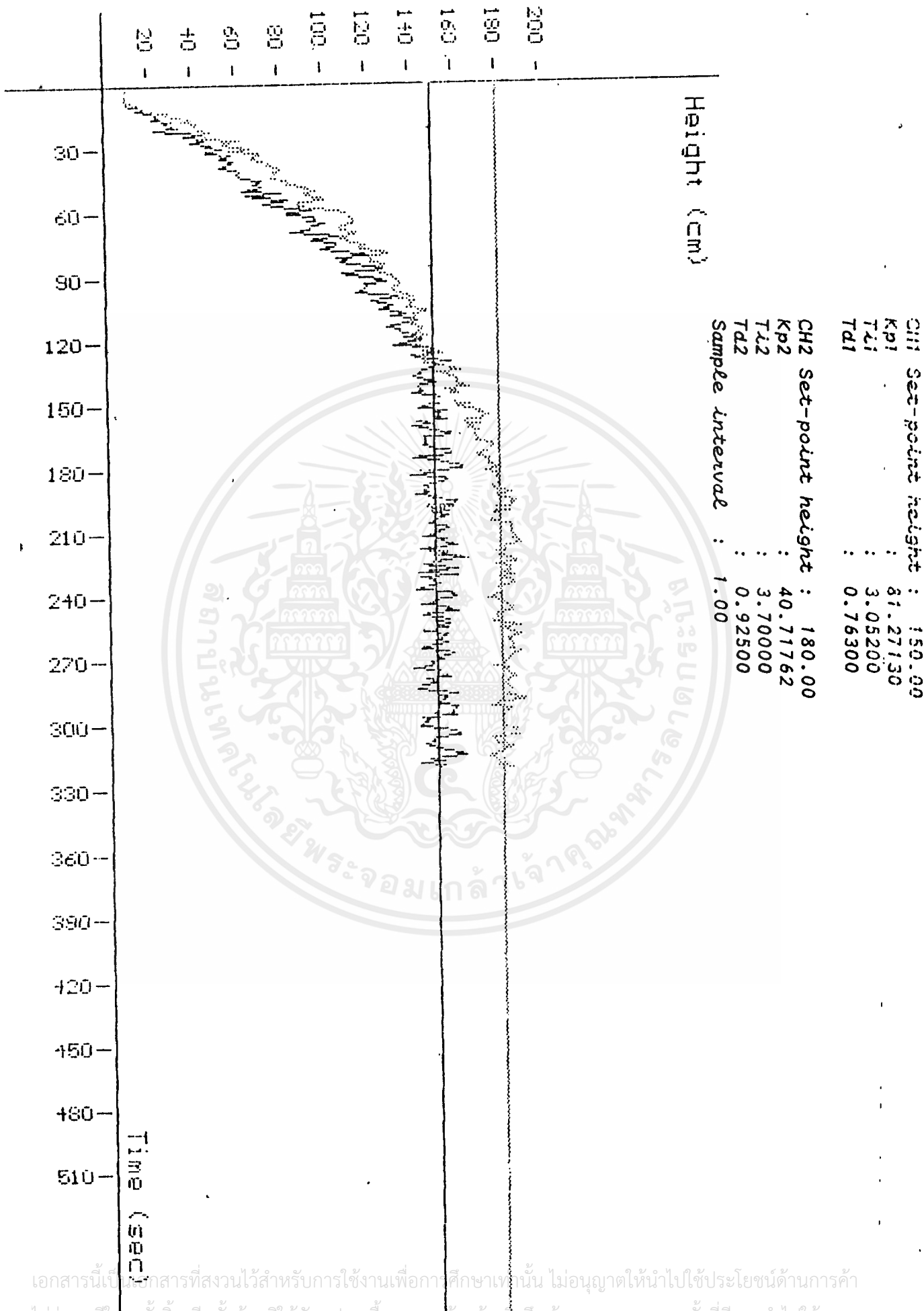
เอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



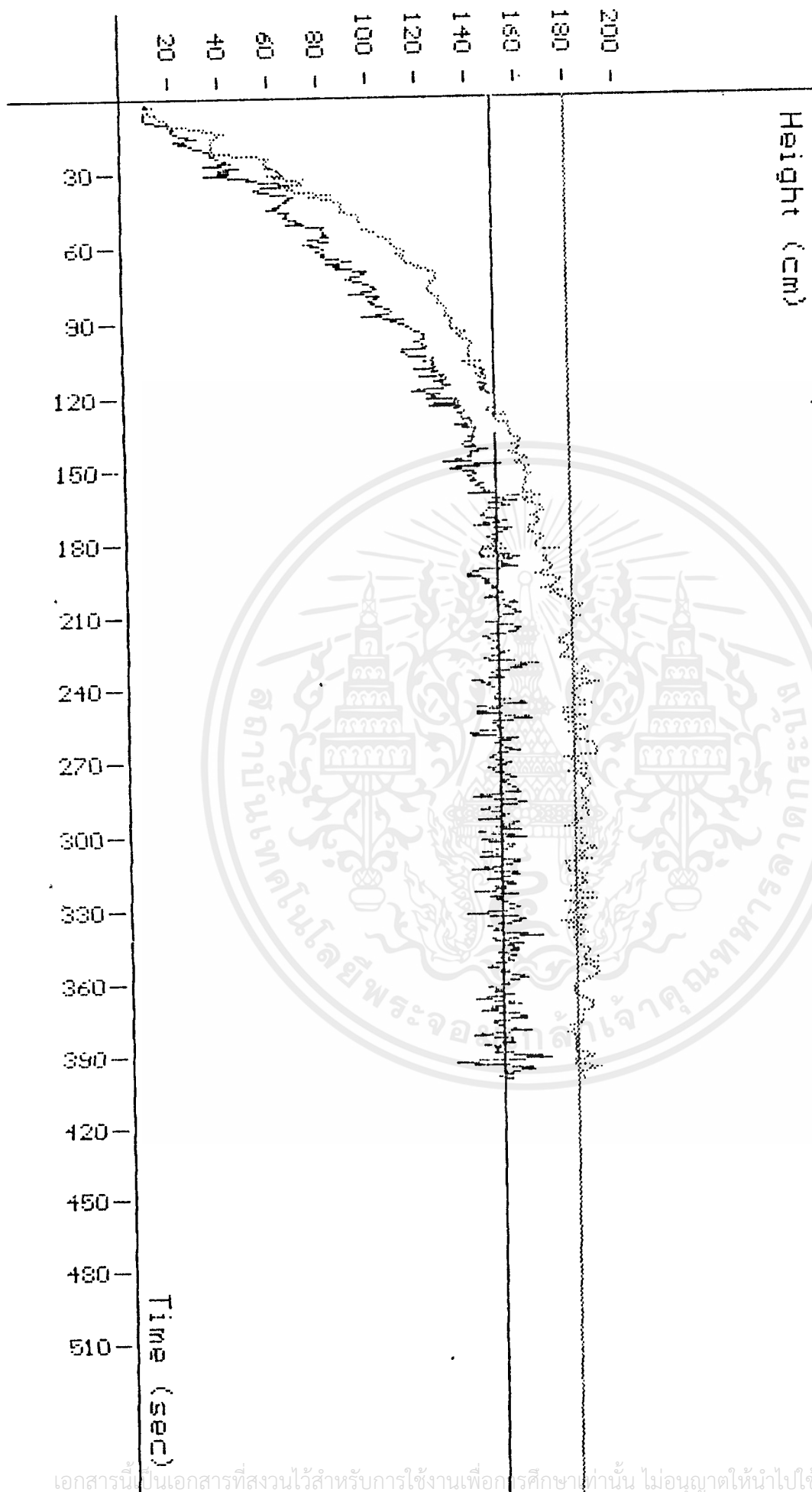
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

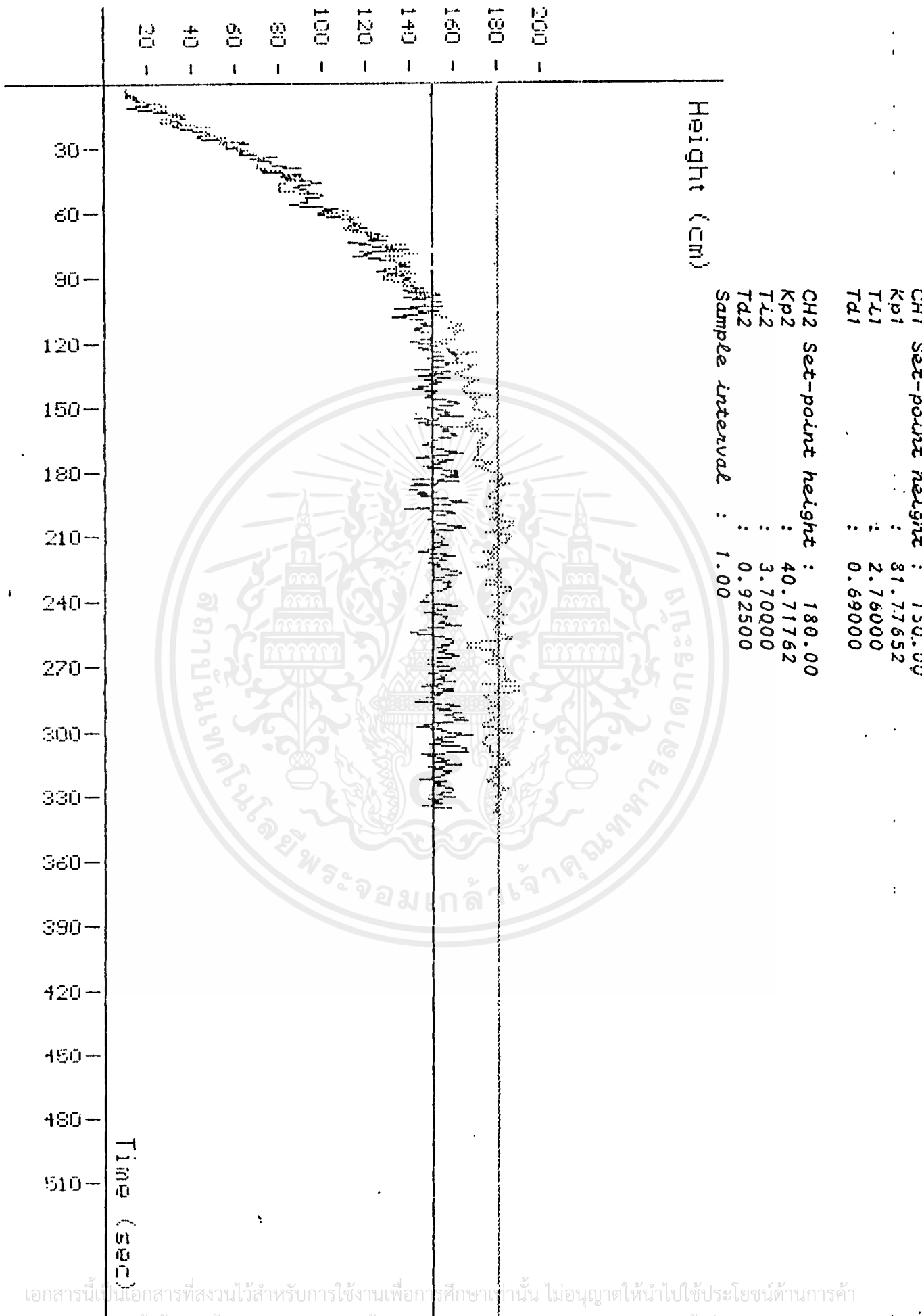


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CH1 Set-point height : 150.00
 KP1 : 62.21495
 Td1 : 4.28000
 Td1 : 1.07000

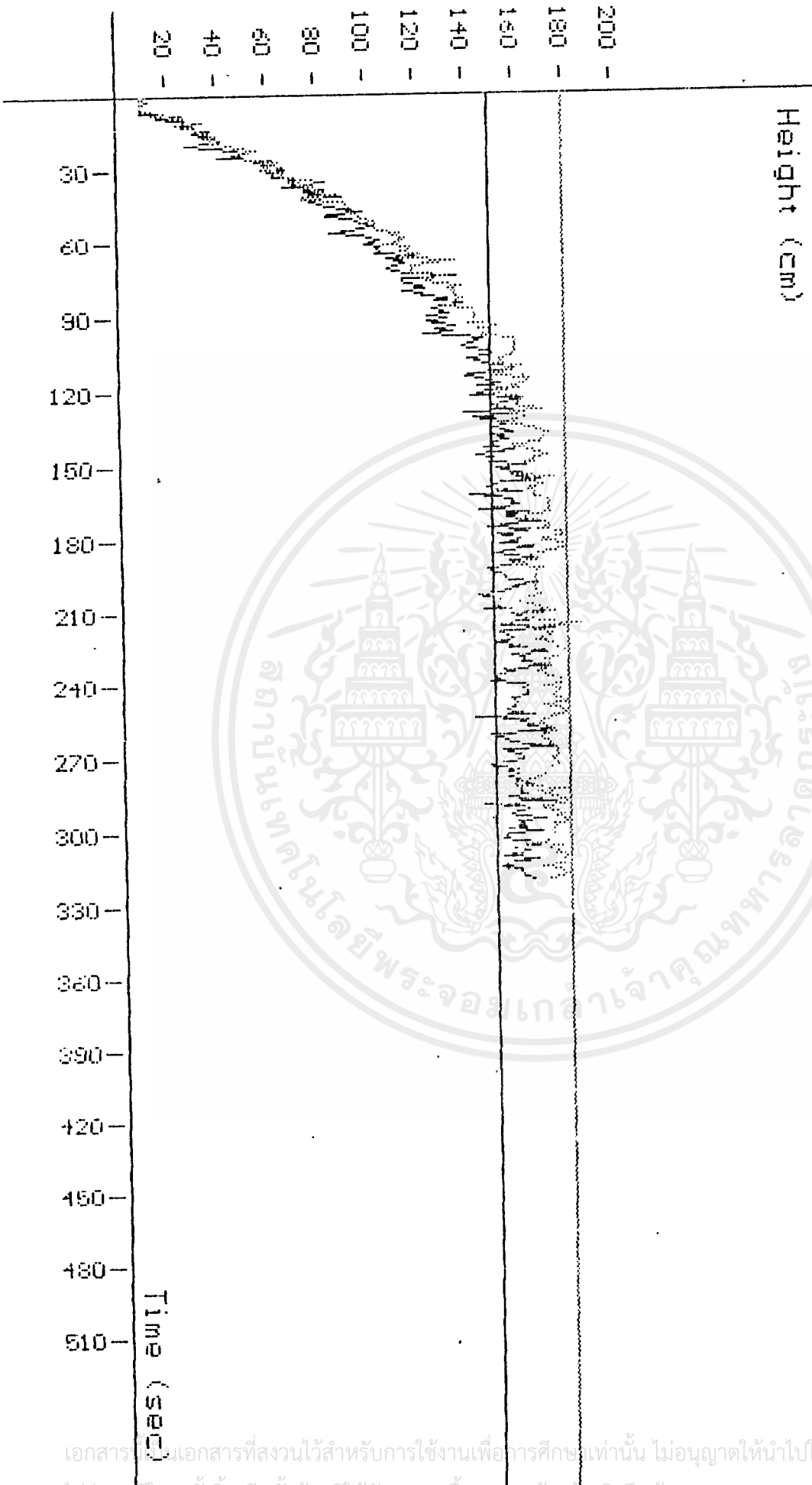
CH2 Set-point height : 180.00
 KP2 : 40.71762
 Td2 : 3.70000
 Td2 : 0.92500
 Sample interval : 1.00



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CH1 Set-point height : 150.00
 KP1 : 84.76522
 TL1 : 2.76000
 Td1 : 0.69000

CH2 Set-point height : 180.00
 KP2 : 40.71762
 TL2 : 3.70000
 Td2 : 0.92500
 Sample interval : 1.00

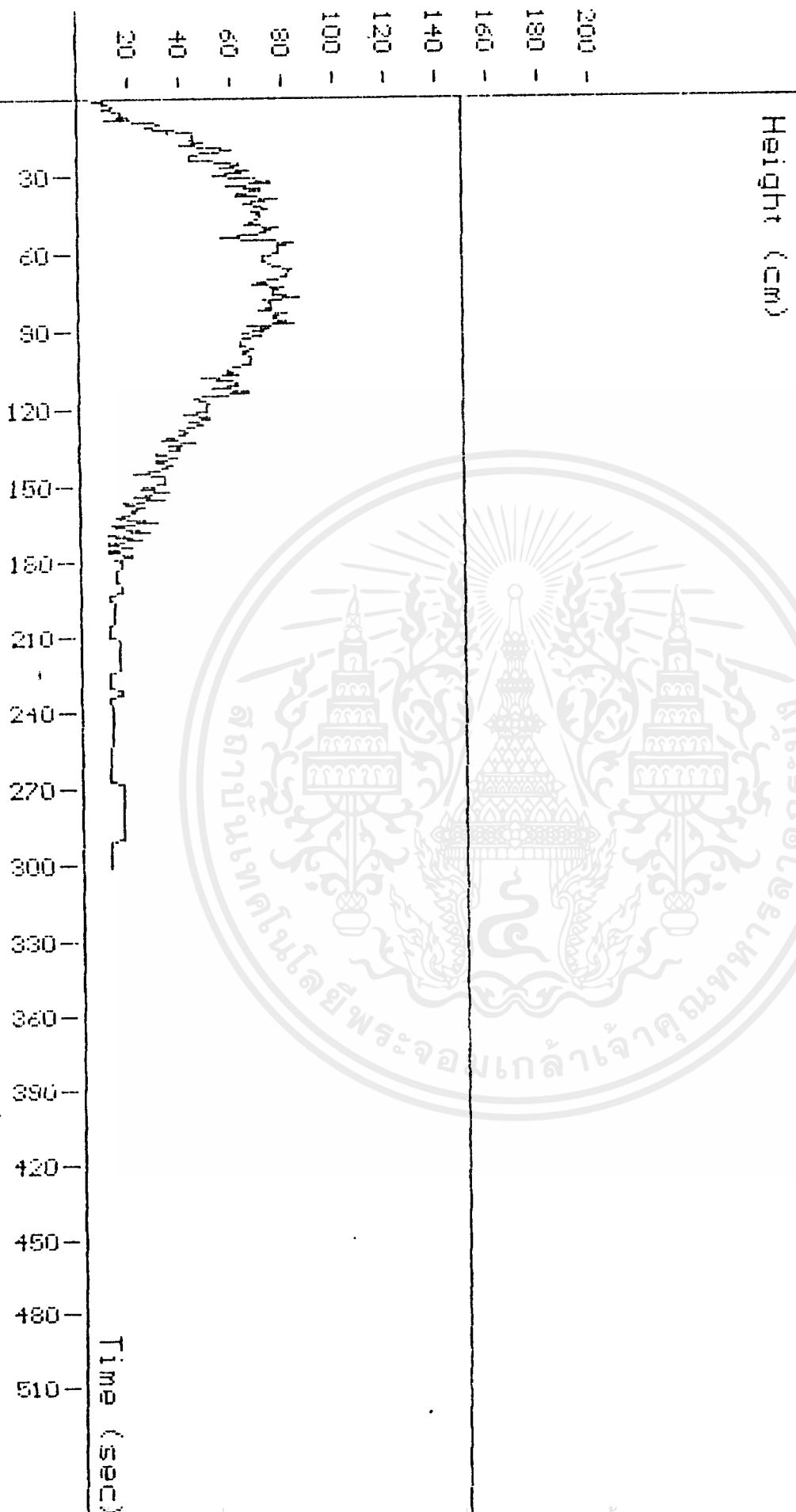


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง : Model Reference Control

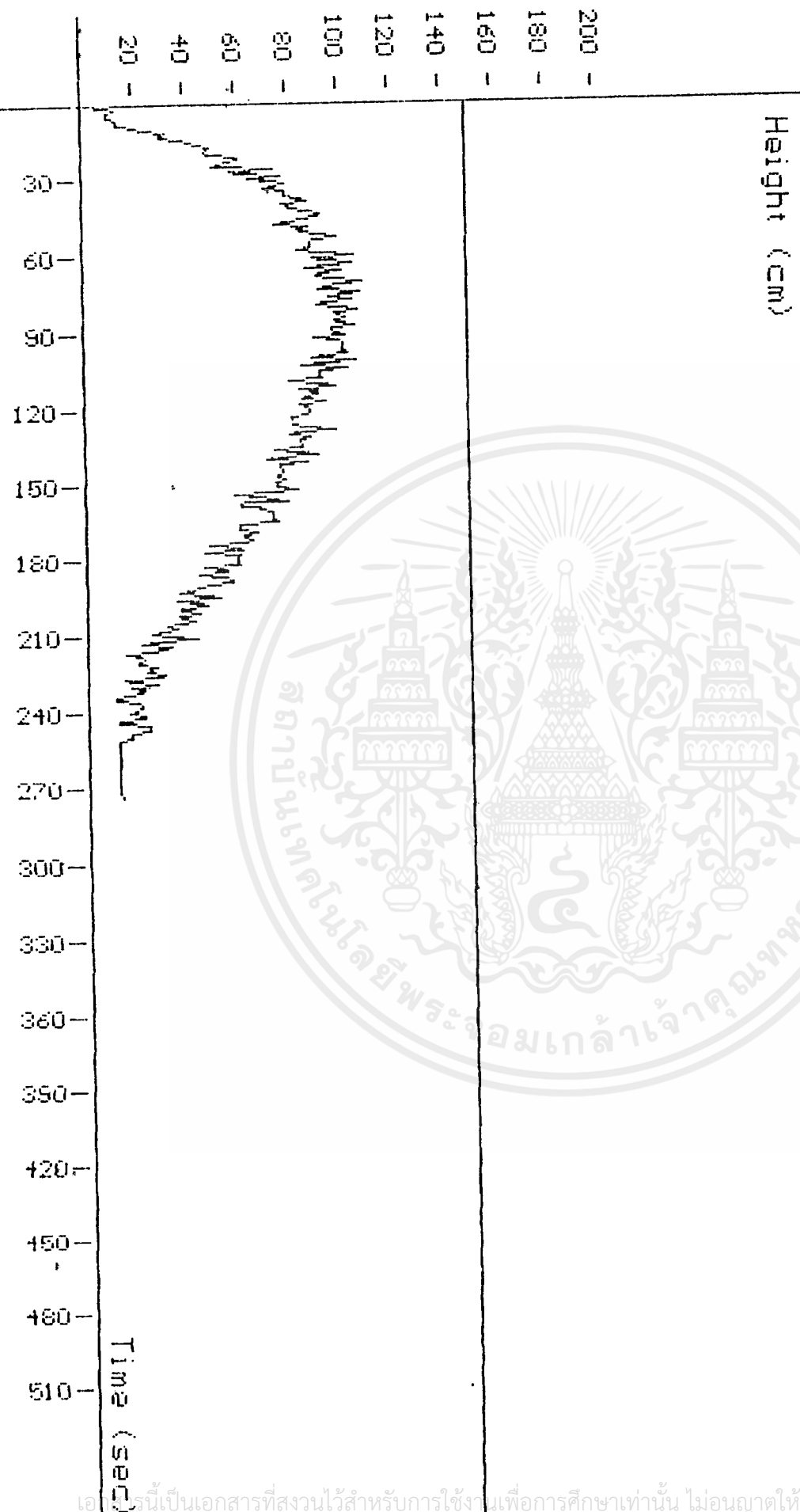
Graph No.	Set Point (cm.)	Time Constant (sec.)	Filter Gain
22	150	74.0	0.20
23	150	74.0	0.30
24	150	74.0	0.40
25	150	74.0	0.50
26	150	74.0	0.60
27	150	74.0	0.70
28	150	74.0	0.80
29	150	74.0	0.90
30	150	74.0	1.00

Set-point height : 150.00
 Sample interval : 1.00
 Time constant : 74.00
 Filter gain : 0.20



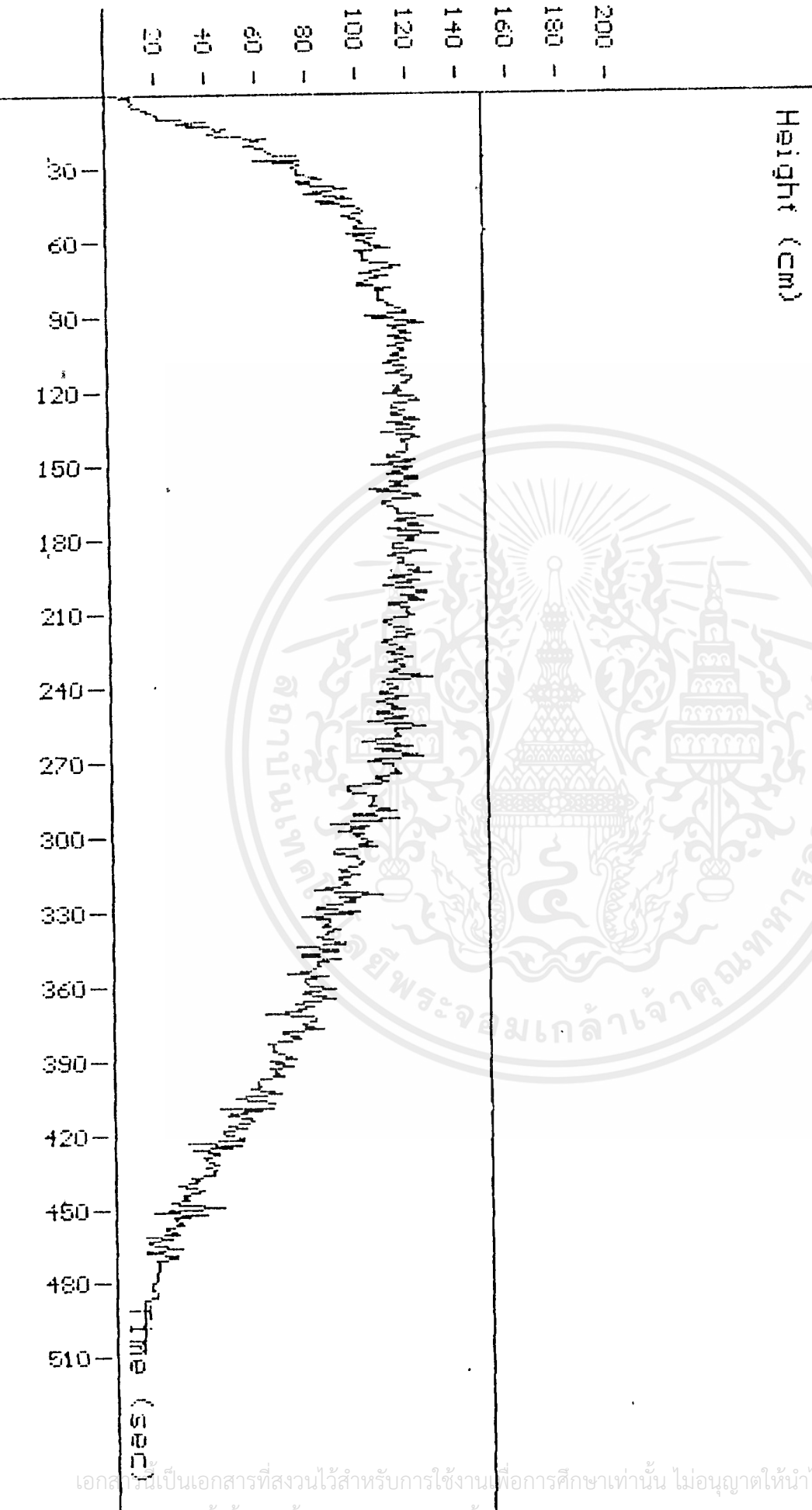
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Channel number : 1
 Set-point height : 150.00
 Sample interval : 1.00
 Time constant : 74.00
 Filter gain : 0.30



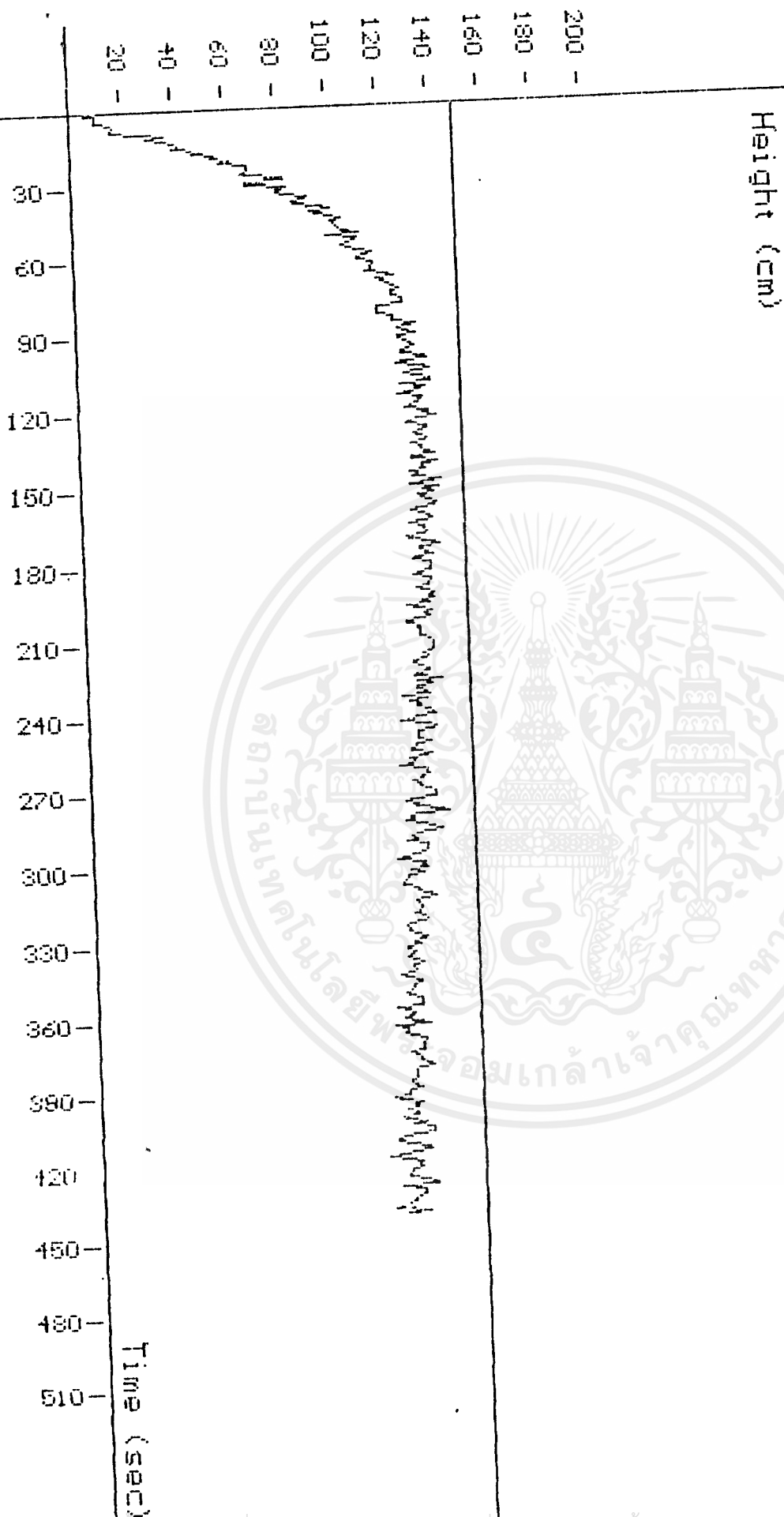
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ทางกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Channel number : 1
 Set-point height : 150.30
 Sample interval : 1.00
 Time constant : 74.00
 Filter gain : 0.40



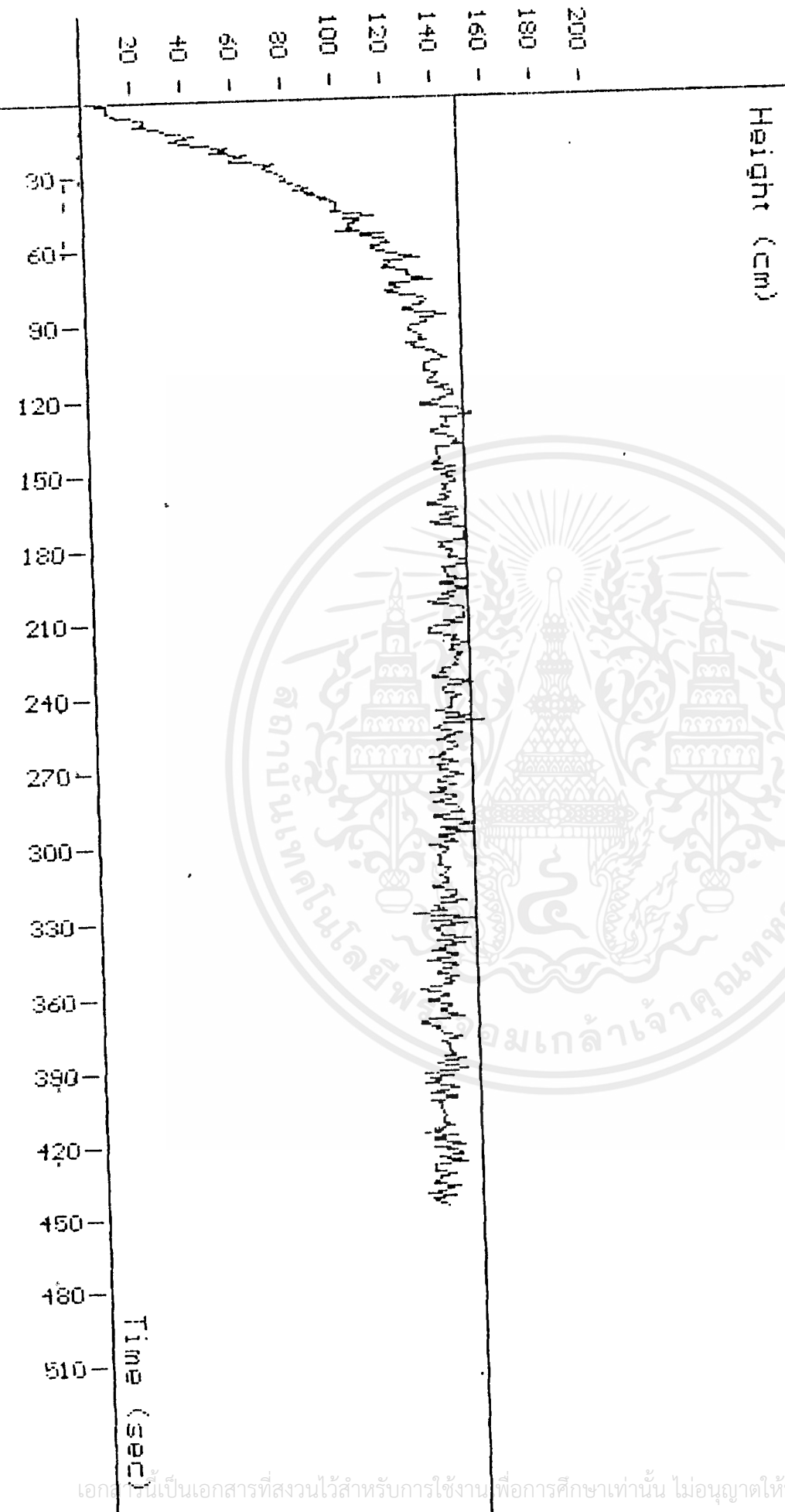
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Channel number : 1
 Set-point height : 150.00
 Sample interval : 1.00
 Time constant : 74.00
 Filter gain : 0.50



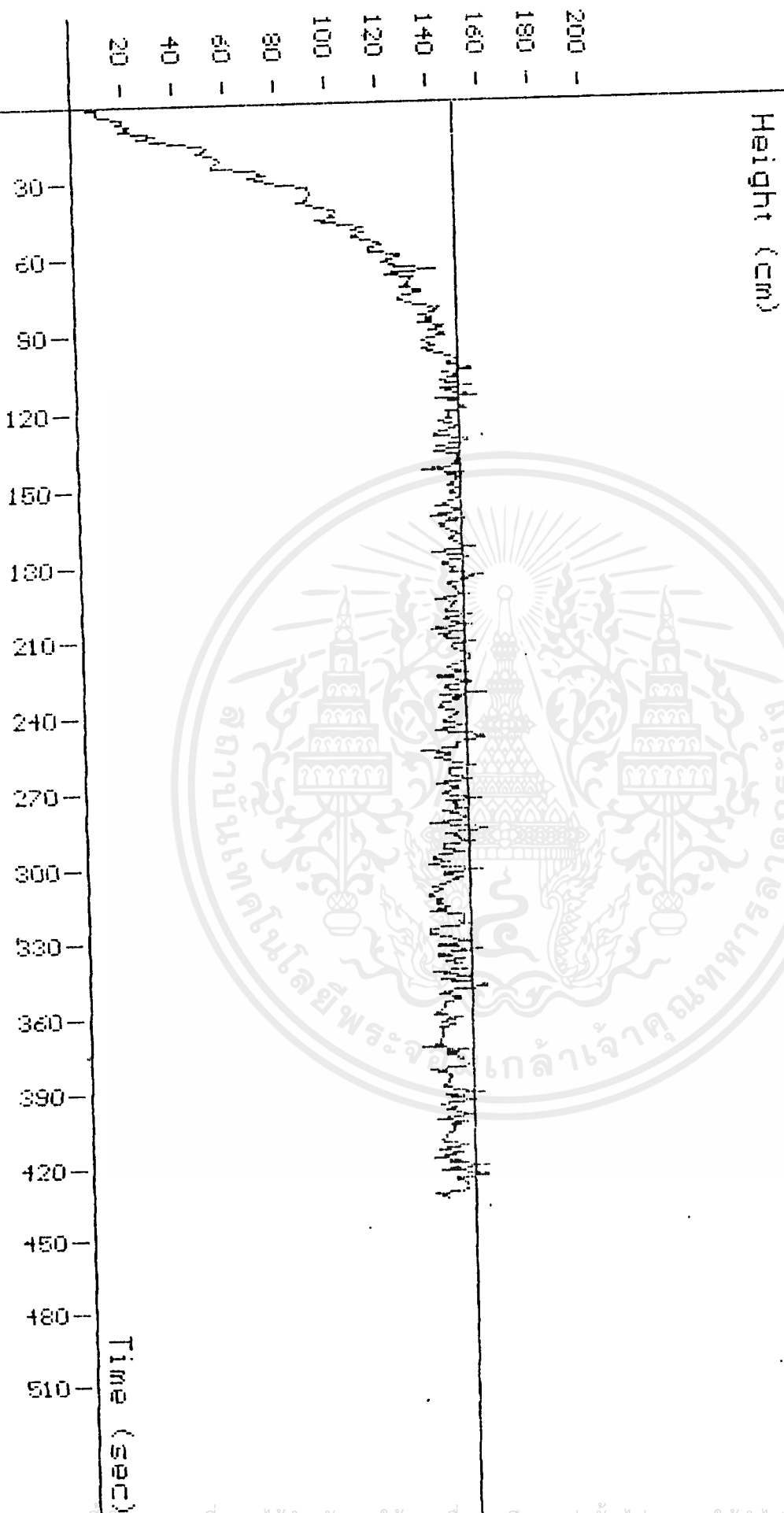
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Channel number : 1
Set-point height : 150.00
Sample interval : 1.00
Time constant : 74.00
Filter gain : 0.60



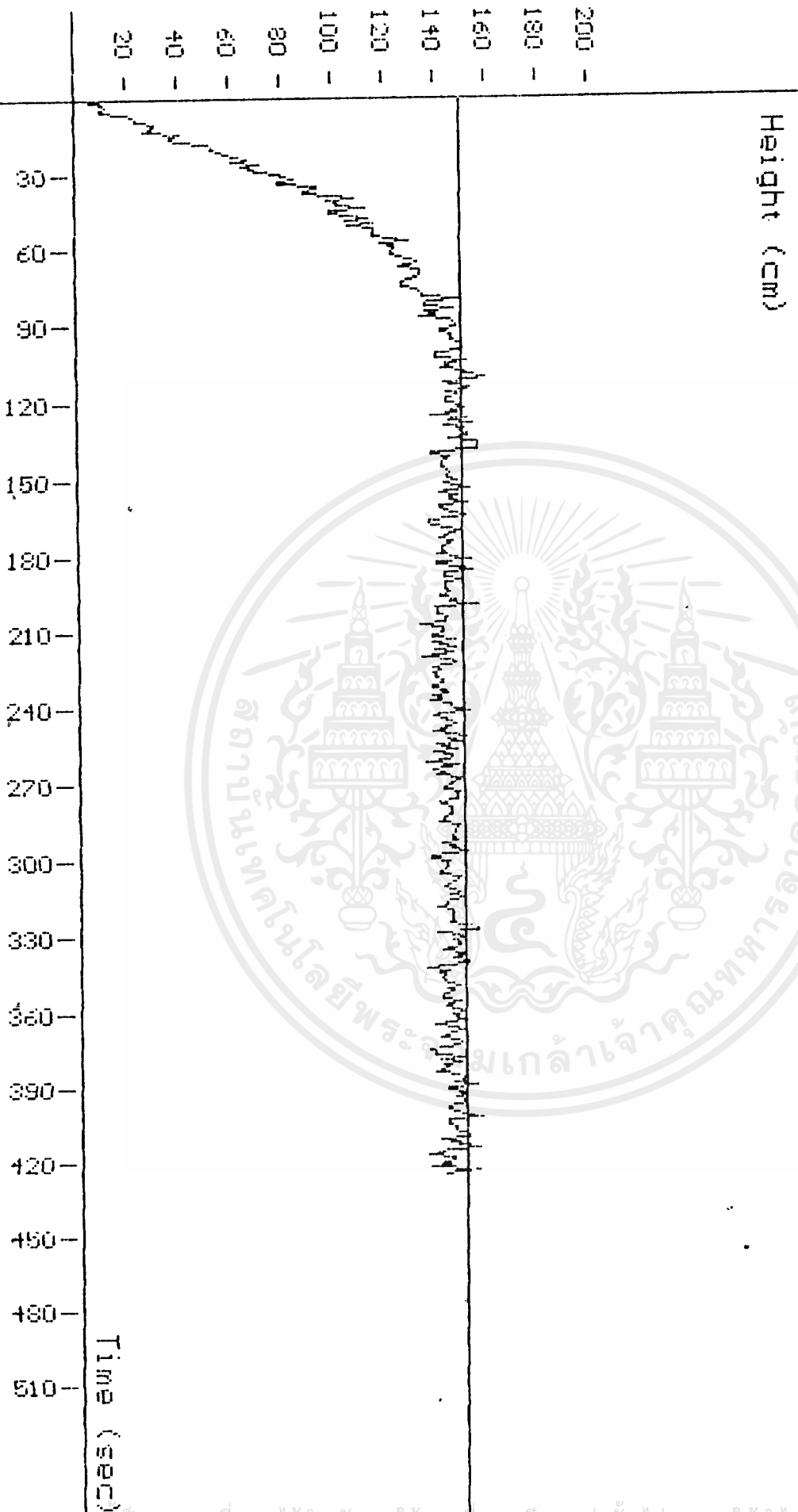
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Station number : 150.00
 Set-point height : 1.00
 Sample interval : 74.00
 Time constant : 0.70
 Filter gain : 0.70

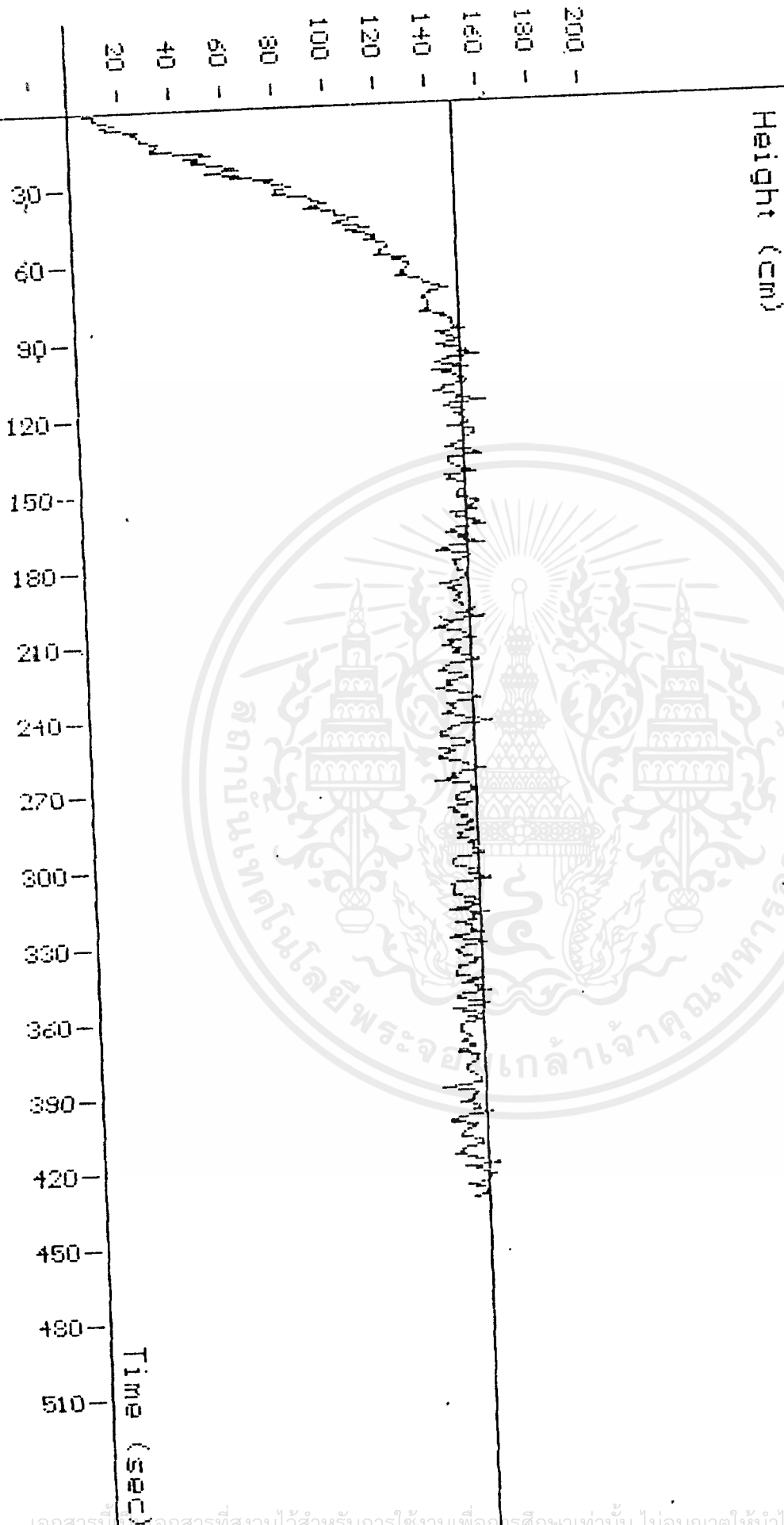


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Channel number : 1
 Set-point height : 150.00
 Sample interval : 1.00
 Time constant : 74.00
 Filter gain : 0.80



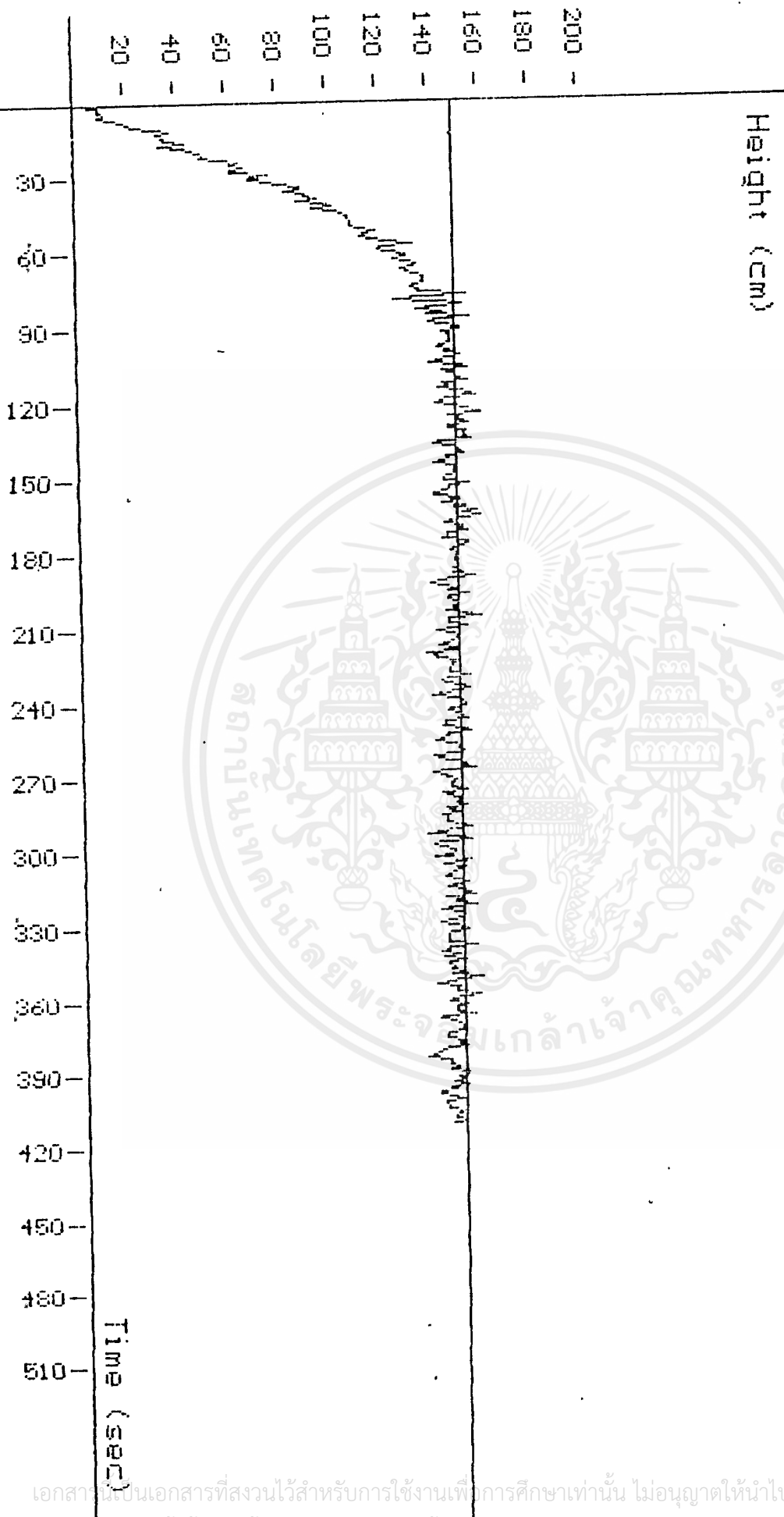
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Set-point height : 150.00
 Sample interval : 1.00
 Time constant : 74.00
 Filter gain : 0.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Checked name :
Set-point height : 150.00
Sample interval : 1.00
Time constant : 74.00
Filter gain : 1.00



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองการควบคุมระดับน้ำแบบท่อนเดียว

ตอนที่ 1 เป็นการควบคุมระดับน้ำที่ 150 cm. และ ทดลองปรับค่า Control parameter (K_p , T_i และ T_d) ดู โดยปรับเป็น 2, 4 และ 8 เท่าของเดิม สามารถสรุปผลได้เป็นดังนี้

- (1.1) เมื่อเปรียบเทียบผลของการปรับเป็น 2 เท่าค่าเดิม จะเห็นได้ว่ากราฟ No.2 ซึ่งมีค่า control parameter เป็น 2 เท่าของกราฟ No.1 จะเข้าสู่สภาวะ Steady State ได้เร็วกว่า
- (1.2) เมื่อเปรียบเทียบผลของการปรับเป็น 4 เท่าของค่าเดิมจะเห็นได้ว่า กราฟ No.3 ซึ่งมีค่า control parameter เป็น 4 เท่าของกราฟ No.1 จะแทบไม่มีความแตกต่างจาก กราฟ No.1 เลย หรืออาจกล่าวได้ว่า ผลของการปรับ Gain เป็น 4 เท่าของค่าเดิม ไม่มีผลทำให้ระบบเข้าสู่สภาวะ Steady State เร็วขึ้น
- (1.3) ถ้าเราลองปรับค่า control parameter เป็น 8 เท่าของค่าเดิม คือ กราฟ No.4 จะเห็นว่าระบบจะเกิดการไม่เสถียร และ ไม่สามารถควบคุมระดับน้ำที่ตำแหน่ง 150 cm. ได้

ตอนที่ 2 เป็นการควบคุมระดับน้ำที่ 185 cm. และทดลองปรับค่า control parameter ให้เป็น 2 เท่าของค่าเดิมดู โดยสามารถสรุปผลได้ เป็นดังนี้

- (2.1) เมื่อเปรียบเทียบกราฟ No.6 ซึ่งมีค่า control

parameter เป็น 2 เท่าของกราฟ No.5 จะพบว่าระบบจะเข้าสู่สภาวะ

Steady State ได้เร็วกว่า เหมือนกับการควบคุมระดับน้ำที่ตำแหน่ง 150 cm. แต่ผลของการปรับค่า control parameter เป็น 2 เท่าของค่าเดิม ณ. การควบคุมที่ระดับ 185 cm. นี้จะเห็นผลไม่ชัด หรือมีความแตกต่างน้อย เมื่อเทียบกับการไม่ปรับเลย

ตอนที่ 3 เป็นการทดลองเปลี่ยนค่า delay time ของระบบ โดยเปลี่ยนจาก 1.53 sec. เป็น 2.14 sec. สามารถสรุปผลได้ดังนี้

(3.1) กราฟ No.7 ซึ่งมีค่า delay time เท่ากับ 1.53 sec. เมื่อเปรียบเทียบกับกราฟ No.8 ซึ่งมีค่า delay time 2.14 sec. จะเห็นได้ว่า กราฟ No.7 จะเข้าสู่สภาวะ Steady State ได้เร็วกว่ากราฟ No.8 อย่างเห็นได้ชัด

ตอนที่ 4 เป็นการทดลองเพื่อศึกษาถึงอิทธิพลของสิ่งรบกวนภายนอก (disturbance) โดยการใส่สิ่งรบกวนภายนอกเข้าไปในระบบ และเพิ่มขนาดสิ่งรบกวนภายนอกให้มากขึ้นเรื่อย ๆ สามารถสรุปผลได้ว่า

(4.1) จากทฤษฎีของการควบคุมแบบ PID เราทราบว่า ตัวควบคุมแบบ I (Integral Control) จะเป็นตัวกำจัด error ของระบบให้หมดไปเมื่อระบบเข้าสู่สภาวะ Steady State ดังนั้นในการทดลองตอนนี้ จึงเป็นการศึกษาดูถึงประสิทธิภาพ ของตัวควบคุมแบบ I ในระบบของเรา

(4.2) จากกราฟ No.9 เป็นการควบคุมระดับน้ำที่ 150 cm. โดยมี disturbance เข้าสู่ระบบขนาด $13.88 \text{ cm}^3/\text{sec}$. จะเห็นได้ว่า ตัวควบคุมแบบ I สามารถจัด

disturbance โดยระดับน้ำในท่อโดยเฉลี่ยก็ยังคงอยู่ที่ 150 cm. ตามค่าที่ตั้งไว้

- (4.3) จากกราฟ No.10 ถึง No.12 เมื่อเราเพิ่มขนาด disturbance ขึ้นเรื่อย ๆ เราพบว่าความสามารถในการขจัด disturbance ของระบบ ของตัวควบคุมแบบ I จะลดลง ซึ่งดูได้จากระดับน้ำในท่อ โดยเฉลี่ยที่ไม่สามารถควบคุมไว้ ในระดับที่ต้องการได้
- (4.4) เป็นที่น่าสังเกตว่า ณ. ที่ระดับการควบคุมเดียวกัน และค่า control parameter ที่เท่ากัน ระบบที่มี disturbance ในปริมาณที่เหมาะสม จะเป็นผลดีต่อระบบมากกว่าผลเสีย คือ ระบบจะเข้าสู่สภาวะ Steady State ได้รวดเร็วขึ้น (เปรียบเทียบกราฟ No.1 กับกราฟ No.9)

สรุปผลการทดลอง

การควบคุมระดับน้ำแบบ 2 ท่อ

ตอนที่ 1 เป็นการควบคุมระดับน้ำที่ 150 cm. และ 180 cm. และทดลองปรับค่า control parameter ของ chanel #2 โดยปรับเป็น 2, 4 เท่าของค่าเดิม สามารถสรุปได้ดังนี้

(1.1) จากกราฟ No.13 สำหรับการควบคุมที่ 150 cm. ของ chanel #1 และ 180 cm. ของ chanel #2 เราพบว่า กราฟของการควบคุมที่ระดับ 150 cm. จะเข้าสู่สภาวะ Steady State ก่อนกราฟของการควบคุมที่ระดับ 180 cm.

(1.2) เมื่อเราทำการปรับ ค่า control parameter ของ chanel #2 เป็น 2 เท่า และ 4 เท่าของค่าเดิม เราพบว่ากราฟของการควบคุมระดับที่ 180 cm. จะเข้าสู่สภาวะ Steady State ได้เร็วขึ้น โดยดูได้จากกราฟ No.14 และ No.15 ตามลำดับ

ตอนที่ 2 เป็นการควบคุมระดับน้ำที่ 150 cm. และ 180 cm. และทดลองปรับค่า control parameter ของ chanel #1 ดู โดยปรับเป็น 2, 4 เท่าของค่าเดิม สามารถสรุปผลได้ดังนี้

(2.1) กราฟของการควบคุมระดับที่ 150 cm. ของกราฟ No.16 และ 17 ซึ่งมีค่า control parameter เป็น 2 และ 4 เท่าของกราฟ No.13 ตามลำดับจะเข้าสู่สภาวะ Steady State ได้เร็วขึ้น โดยกราฟของการควบคุมระดับน้ำที่ 150 cm. ของกราฟ No.16 จะเข้าสู่สภาวะ Steady State ได้เร็วกว่ากราฟ No. 13 และกราฟ No.17 จะเข้าสู่สภาวะ Steady State เร็วกว่ากราฟ No.16

ตอนที่ 3 เป็นการศึกษาถึงการเปลี่ยนค่า delay time ของ chanel

#1 ในการควบคุมแบบ 2 ท่อ สามารถสรุปผลได้เป็นดังนี้

(3.1) กราฟ No.18 เป็นการควบคุมที่ระดับ 150 cm. และ 180 cm. โดย chanel #1 มีค่า delay time เท่ากับ 1.53 sec. และในกราฟ No.19 เป็นการที่ระดับเดียวกัน แต่ chanel #1 มี delay time เท่ากับ 2.14 sec. จะพบว่าในระบบที่มีค่า delay time มากจะมีผลทำให้การเข้าสู่สภาวะ Steady State ช้าลงตามลำดับ

ตอนที่ 4 เป็นการศึกษาถึงผลของ non-linearity ในระบบ การทำให้เกิด non-linearity สามารถทำได้ โดยเปิดวาล์ว ที่เชื่อมระหว่างท่อพลาสติกทั้งสอง เพื่อให้ น้ำในท่อพลาสติกทั้งสองสามารถไหลไป-มาระหว่างกันได้ ซึ่งได้ผลทดลองดังนี้

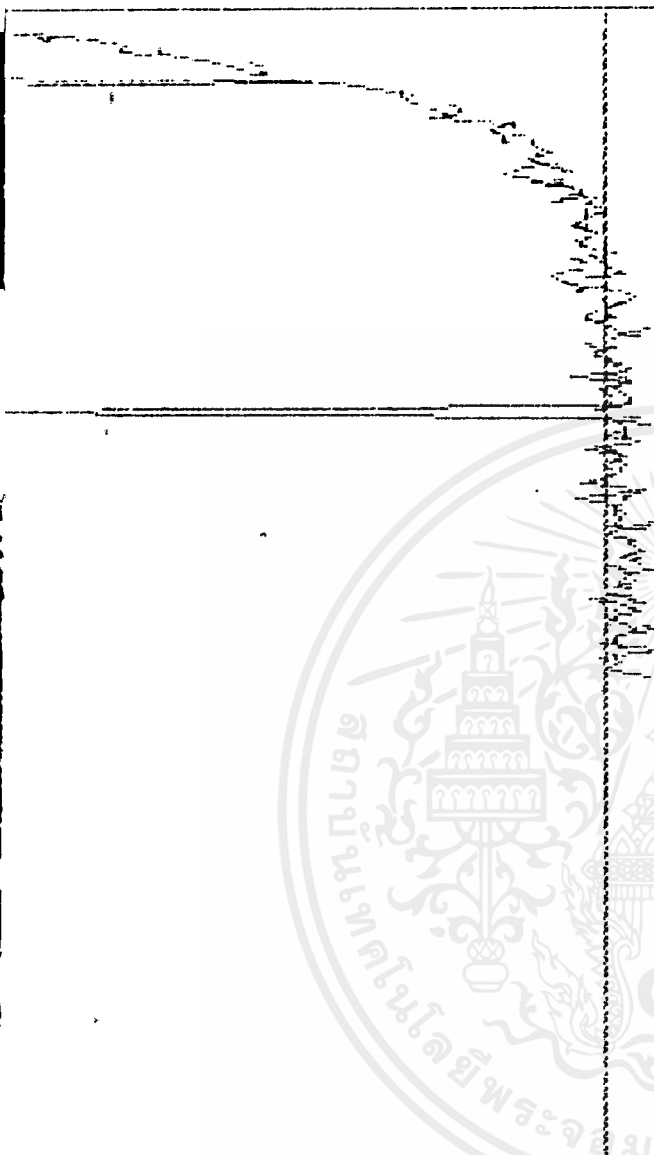
(4.1) ในกราฟ N.20 เป็นการเปิดวาล์วที่เชื่อมระหว่างท่อทั้งสองออกครึ่งหนึ่ง ของตำแหน่งเปิดสุด และในกราฟ No. 21 เป็นการเปิดวาล์วดังกล่าวออกที่ตำแหน่งเปิดสุด จะเห็นว่า ผลของ non-linearity จะมีแนวโน้มทำให้การควบคุมระดับน้ำในท่อทั้งสองเปลี่ยนจากระดับที่ตั้งไว้มาเป็นระดับเดียวกัน

วิจารณ์ผลการทดลอง

ในการทดลองบางครั้ง อาจเกิดเหตุการณ์ที่กราฟผลตอบแทนของระบบเกิดตกлубลงมาจากค่าที่ใกล้เคียงกันในขณะนั้น โดยที่ระบบจริง ๆ (หรือระดับน้ำในท่อพลาสติกใส) ไม่ได้ตกลงมาดังกราฟนั้น การเกิดปรากฏการณ์เช่นนี้เรียกว่า "การเกิด glitch" ดังแสดงไว้ในกราฟ No.31

การเกิด glitch เกิดขึ้นเนื่องจากตัวแปลงสัญญาณ A/D เกิดการแปลงสัญญาณไม่ทันในบางช่วง เช่น สมมุติว่า มีการแปลงสัญญาณจาก $0111\ 1111_2$ (127_{10}) ไปเป็น $1000\ 0000_2$ (128_{10}) เราทราบว่าค่า 0 ในทาง digital หมายถึง ปิดค่า 1 ในทาง digital หมายถึง เปิด ซึ่งตำแหน่ง เปิด หรือ ปิด ในที่นี้ คือ ค่าที่วงจร A/D ส่งสัญญาณ pulse ออกไป ดังนั้นในการแปลงสัญญาณจาก $0111\ 1111_2$ ไปเป็น $1000\ 0000_2$ 7 bit แรกของสัญญาณ $0111\ 1111_2$ จะต้องถูก set กลับไปที่ตำแหน่ง เปิด แต่ในทางปฏิบัติการ set ให้ เปิด หรือ ปิด จะทำโดยพร้อม ๆ กันไม่ได้ จะต้องทำไล่ไปตั้งแต่ bit ต่ำสุดไปจนถึง bit สูงสุด ซึ่งในบางครั้งวงจรยังแปลงสัญญาณไม่ทันเสร็จเรียบร้อย ก็ต้องรับส่งสัญญาณออกไป เพราะได้รับสัญญาณ trigger มาจากตัวโปรแกรม อย่างเช่นในกรณีนี้ เมื่อวงจร A/D กำลังแปลงสัญญาณ มาถึง bit ที่ 7 คือ set สัญญาณ 7 bit กลับลงมาเป็น 0 หมดแล้ว และกำลังจะแปลงสัญญาณ bit ที่ 8 กลับขึ้นไปเป็น 1 แต่เมื่อได้รับสัญญาณ trigger จากตัว โปรแกรมเข้ามา จึงต้องส่งสัญญาณที่แปลงได้ในขณะนั้น คือ $0000\ 0000_2$ ออกไปโดยยังไม่ทันได้แปลง bit ที่ 8 ให้เป็น 1 เสียก่อน ผลคือ การส่งค่า ศูนย์ เข้าไปในตัวโปรแกรม ทำให้กราฟตกлубลงมา เกิด glitch ขึ้นได้

สาเหตุของการเกิด glitch อาจเกิดจากรา set ค่า wait state ของตัว PCL-812 ให้น้อยเกินไป แต่สาเหตุสำคัญ คือ ตัววงจรร้อนมากเกินไป เพราะอาจจะเปิดใช้งานมานาน การแก้ไขทำได้โดยเพิ่มค่า wait state ให้มากขึ้น หรือ ปิด computer เพื่อพักเครื่องสักระยะหนึ่ง แล้วเปิดใช้งานใหม่ ก็สามารถแก้การเกิด glitch ได้



nel number : 1
 point height : 150.00
 le interval : 2.00
 constant : 74.00
 y time : 0.81
 : 109.62963
 : 1.62000
 : 0.40500

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

- *ขอขอบพระคุณ อาจารย์ ดร.มงคล มงคลวิโรจน์ ที่ให้คำปรึกษามาโดยตลอด
- *ขอขอบคุณ พี่ มณฑา เทียมเมือง ที่ให้คำแนะนำและความสะดวกในการทำ
อุปกรณ์ต่าง ๆ
- *ขอขอบคุณ เพื่อนนักศึกษารุ่นที่ 28 ที่เป็นกำลังใจมาตลอด



เอกสารอ้างอิง

1. ดร.ดวงแก้ว สวามิภักดิ์, "การโปรแกรมภาษา C", ซีเอ็ดบุ๊คเซ็นเตอร์, 2521
2. Ernest F. Johnson, "Automatic Process Control", New York: McGraw-Hill Book Company, 1967
3. Katsuhiko Ogatta, "Modern Control Engineering", 2nd edition : Englewood Cliffs N.J: Prentice Hall, 1990
4. Noel M. Moris, "Control Engineering", 3rd New York: McGraw Hill Company, 1983
5. "PCL-812 USER Manual", Printed in Taiwan, 1989
6. Richard C. Dorf, "Modern Control System", Addison Wesley, 1989
7. Stuart Berett, "Real time computer control", Prentice Hall, 1988
8. "Turbo C Reference Guide version 2.0", Borland International, 1988
9. Werner Feibel, "Using Quick C", New York: McGraw Hill Book Company, 1988
10. เอกสารประกอบการสัมมนา ทางวิชาการวิศวกรรมเครื่องกล ครั้งที่ 3 เรื่อง "วิศวกรรมเครื่องกลกับการพัฒนาประเทศ", มหาวิทยาลัยสงขลานครินทร์