รายการอ้างอิง

1. Cassiopeia Technologies Incorporporated . CASSIM CANDU-9 model description . Toronto Canada: (n.p.), 1995.
2. Cassiopeia Technologies Incorporporated . CASSIM User manual. Toronto Canada: (n.p.), (n.d.).
3. Sewoo, C., and Soon-Heung, C. Application of neural networks to a connectionist expert system for transient identification in nuclear power plants. Nuclear Technology 102 (May 1993): 177-191.
4. Myung-Sub, R., Se-Woo, C., and Soon-Heung, C. Power prediction in nuclear power plants using a Back-Propagation Learning neural network. Nuclear Technology 94 (May 1991): 270-278.
5. Reifman, J., and Vitelar, J. E. Accelerating learning of neural networks with conjugate gradients for nuclear power plant applications. Nuclear Technology 106 (May 1994): 225-241.
6. Keehoon, K., and Bartlett, E. B. Error estimation by series association for neural network systems Neural Computation 7 (1995) :799-808.
7. Levine, R. I., Drang, D. E., and Edelson, B. AI and expert system . 2nd ed . Singapore: McGraw-Hill, 1991.
8. ราตรี สุดทรวง. ประสาทสรีรวิทยา . พิมพ์ครั้งที่ 2. คณะแพทยศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย : (ม.ป.ท) , 2528.
9. Russell, S. J., and Norvig, P. Artificial intelligence a modern approach . New Jersey USA : Prentice-Hall ,1995.
10. Lam, K., Cham, W., and Li, C. CASSIM(TM) Simulation database version 1.2[Computer Program]. Toronto Canada : Cassiopeia Technologies Incorporated , 1995.
11. Lam, K., Cham, W., and Li, C. CASSIM(TM) Simulation engine version 1.1[Computer Program].Toronto Canada : Cassiopeia Technologies Incorporated , 1995.
12. National Instruments Corporation . LabVIEW for windows tutorial . Austin Texas : (n.p.), 1994.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บรรณานุกรม

Atomic Energy of Canada Limited. CANDU-9 Overview (Training Document). (n.p.): (n.p.), (n.d.).

Atomic Energy of Canada Limited. CANDU reactor and kinetics (Training Document). (n.p.): (n.p.), (n.d.).

Atomic Energy of Canada Limited. Instrumentation and control (Training Document). (n.p.): (n.p.), (n.d.).

Atomic Energy of Canada Limited. Nuclear power plant control (Training Document). (n.p.): (n.p.), (n.d.).

Atomic Energy of Canada Limited. Reactor control and simulation (Training Document). (n.p.): (n.p.), (n.d.).

IBM Corporation. VS FORTRAN Programming guide. 1st ed. San Jose California: IBM Programming Publishing, 1984.

Keehoon, K., and Bartlett, E. B. Error prediction for a nuclear power plant fault-dianostic advisor using neural networks. Nuclear Technology 108 (November 1994): 283-295.

Levine, R. I., Drang, D. E., and Edelson, B. AI and expert system . 2nd ed . Singapore:McGraw-Hill, 1991.

Naito, N., Sakuma, A., and Shigeno, K. A real-time expert system for nuclear power plant failure dianosis and operational guide .Nuclear Technology 79 (December 1987): 284-296.

National Instruments Corporation . LabVIEW function reference manual . Austin Texas : (n.p.), 1994.

Salamun, I., and Stritar, A. Nuclear power plat transient dianostics using transfer functions as estimators. Nuclear Technology 124 (November 1991): 118-137.

Schalkoff, R. J. Pattern recognition :statistical , structural and neural approaches . USA : John Wiley&Sons , 1992.

Seong-Soo, C., Ki-Sig, K., and Soon-Heung, C. Development of an on-line fuzzy expert system for integrated alarm processing in nuclear power plants. IEEE Transactions on Nuclear Science 42 (August 1995): 1406-1417.

Se-Woo, C., Han-Gon, K., Wan-Joo, K., Bok-Ki, M., and Soon-Heung, C. Development of an expert system for failure dianosis of primary side system . Nuclear Technology 97 (January 1992): 1-15.

Takizawa, Y., Hattori, Y., Itoh, J., and Fukumota, A. An intelligent man-machine system for future nuclear power plants. Nuclear Technology 107 (July 1994): 72-82.

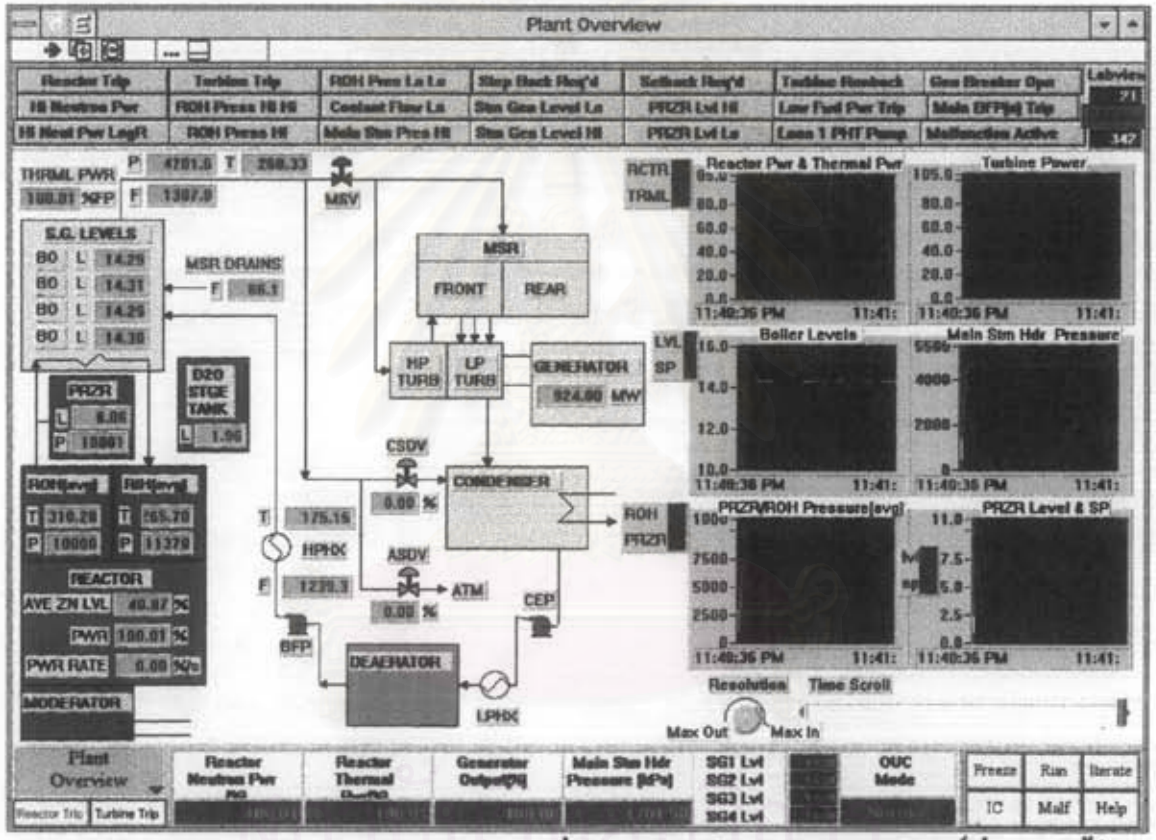
Vo, T. V., and Edwards, D. R, Development of in-service inspection priorities for pressurizes water reactor high-pressure injection system components. Nuclear Technology 106 (April 1994): 110-124.

Wu, W., and Lin, C. Optimal reliable control system design for steam generators in pressurized water reactors. Nuclear Technology 106 (May 1994): 216-224.

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

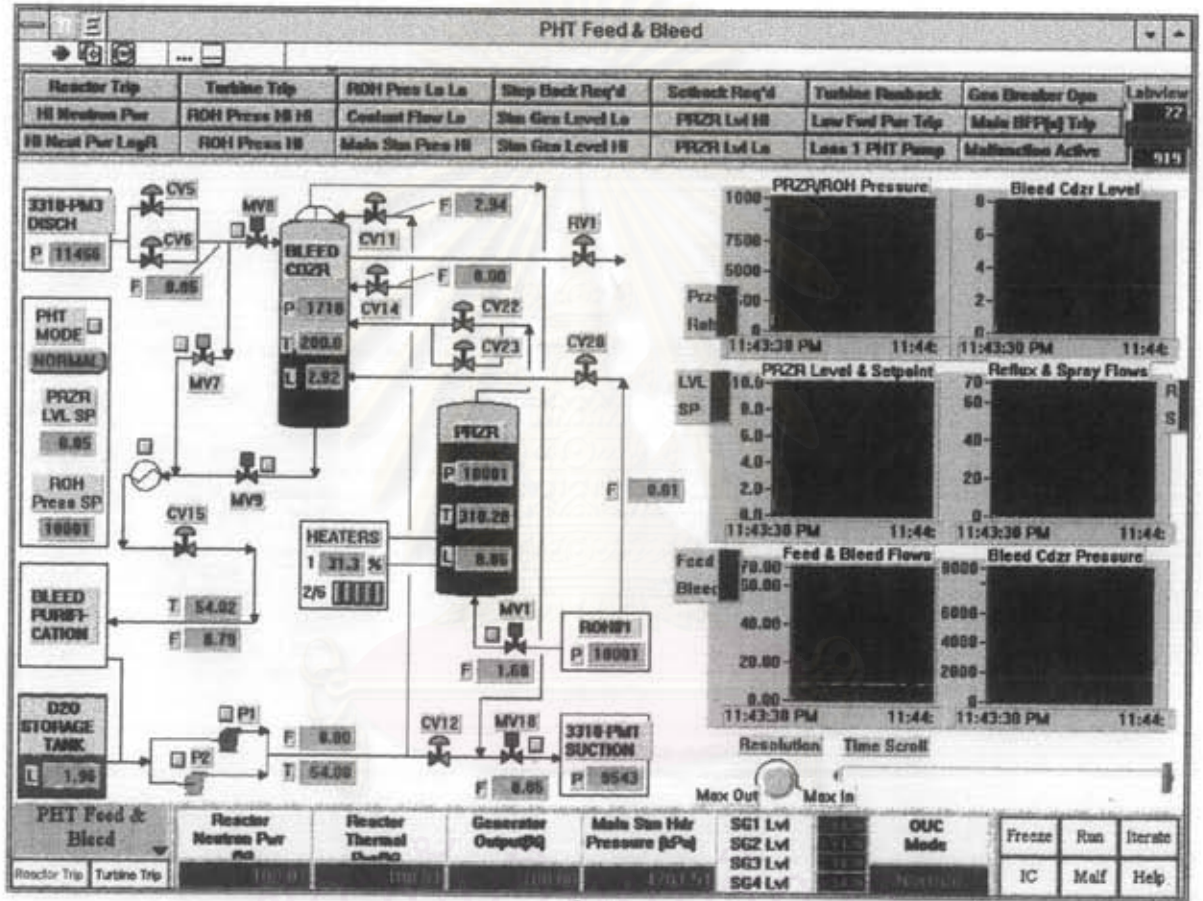
ภาคผนวก ก.



รูปที่ ก.1 แสดงภาพของPlant Overview Screen ซึ่งมีรายละเอียดของพารามิเตอร์ที่นำมาใช้เรียนรู้ทั้งหมด

(Lam, K., Cham, W., and Li, C. โปรแกรมจำลองแบบโรงไฟฟ้านิวเคลียร์แคนดู-9

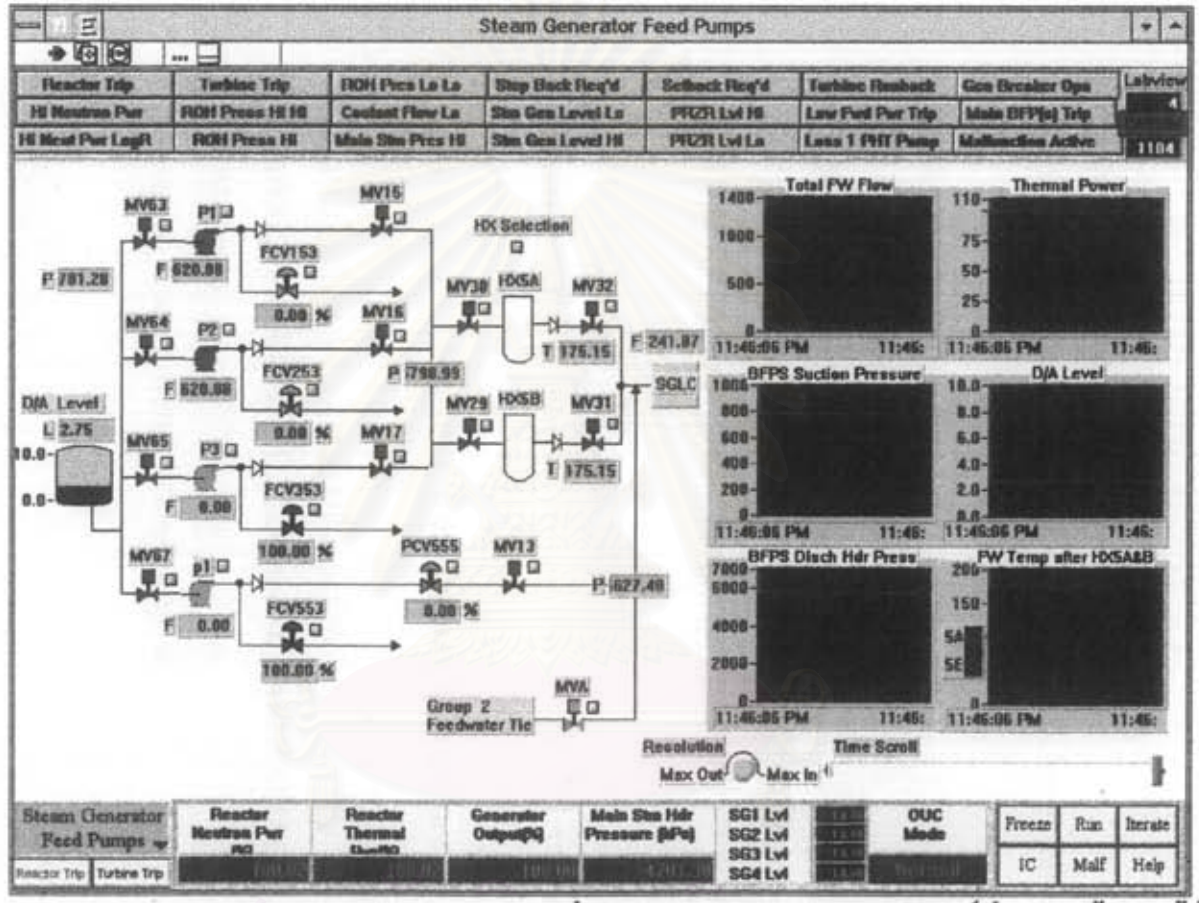
[Computer Program].Toron Ontario ,Canada: Cassiopia Technologies Incorporated ,1995.)



รูปที่ ก.2 แสดงภาพของPHT Feed&Bleed Screen ซึ่งมีรายละเอียดของพารามิเตอร์ที่นำมาใช้เรียนรู้ทั้งหมด

(Lam, K., Cham, W., and Li, C. โปรแกรมจำลองแบบโรงไฟฟ้านิวเคลียร์แคนดู-9

[Computer Program].Toronto ,Ontario ,Canada: Cassiopia Technologies Incorporated ,1995.)



รูปที่ ก.3 แสดงภาพของ SG Feed Pumps Screen ซึ่งมีรายละเอียดของพารามิเตอร์ที่นำมาใช้เรียนรู้ทั้งหมด

(Lam, K., Cham, W., and Li, C. โปรแกรมจำลองแบบโรงไฟฟ้านิวเคลียร์แคนดู-9

[Computer Program].Toronto ,Ontario ,Canada: Cassiopia Technologies Incorporated ,1995.)

ภาคผนวก ข.

ระหัสต้นฉบับ (Source Code) ภาษาฟอร์แทรนของโปรแกรมเครือข่ายนิวรอลที่สร้างขึ้น พร้อมความหมายของตัวแปร

Algorithm 860

ขั้นตอนวิธีของส่วน **Watchdog Block**

หน้าที่ ตรวจสอบสภาวะการเดินเครื่องของโรงไฟฟ้าฯ โดยใช้หลักการของ Competitive Network

หลักการพื้นฐาน หัวข้อ 4.2 , 5.2 , 6.2.2

1.รายละเอียดในส่วนตัวแปรของค่านำเข้า

PARAMETER(1),PARAMETER(2),.....,PARAMETER(90) : ค่าพารามิเตอร์จาก Plant Overview Block ซึ่งนำมาสร้างเป็นพารามิเตอร์เวกเตอร์ใน Watchdog Block

2.รายละเอียดในส่วนตัวแปรของค่านำออก

STATUS : เป็นตัวแปรที่บอกสภาวะการเดินเครื่องของโรงไฟฟ้าฯหลังจากที่ Watchdog Block ได้ตรวจสอบแล้ว

ANGLE_OLD, ANGLE_DIFF : ตัวแปรใช้ในการคำนวณค่ามุม α

3.รายละเอียดในส่วนตัวแปรของค่าสัมประสิทธิ์

NORMAL_PARAMETER(1),.....,NORMAL_PARAMETER(90) : ค่าส่วนประกอบของนอมอลเวกเตอร์

CRITICAL_ANGLE_DIFF : ค่าวิกฤติของมุม α

4.ส่วนของโปรแกรม

```
SUBROUTINE ALG860(INP,OUT,COF,DT)
C
C AUTHOR: PHIPHAT.T
C
C DATE:7-NOV-1997
C
C REVISION NOTES:WATCHDOG_ BLOCK
C
C ALGORITHM NUMBER: 860
C
C ALGORITHM NAME: WATCH_DOG
C
C ALGORITHM DESC: MONITOR STATUS OF NUCLEAR POWER PLANT
```

```
C
C INPUTS: 91
C 1-90:PARAMETER(90),necessory parameter to monitor status
C
C OUTPUTS: 1
C 1:STATUS,normal/abnormal , 1/0
C 2:ANGLE_OLD,angle between vector
C 3:ANGLE_DIFF,angle difference
C 4:COUNT_DOWN,number of abnormal
C 5:COUNTER,number of iteration
C
C COFFICIENT:91
C 1-90:NORMAL_PARAMETER(90),vector at normal operation
C 91  :CRITICAL_ANGLE_DIFF(degree),difference angle of abnormal operation
C  DECLARATION
C
      DOUBLE PRECISION INP(*),OUT(*),COF(*),DT
      DOUBLE PRECISION PARAMETER(90),
      NORMAL_PARAMETER(90)
      DOUBLE PRECISION CRITICAL_ANGLE_DIFF,ABSOLUTE,ANGLE_NEW,STATUS
      DOUBLE PRECISION ANGLE_OLD,ANGLE_DIFF,COUNT_DOWN,COUNTER
      DOUBLE PRECISION DIFF_HIGH,DIFF_LOW
      INTEGER I
C
C   INTERLOCK PROCEDURE
C
      STATUS=OUT(1)
      IF (STATUS.EQ.0) GOTO 200
C
C INPUTS AND COFS SETUP
C
C  INPUT SETUP
      DO 10 I=1,90
      PARAMETER(I)=INP(I)
 10 CONTINUE
      DO 20 I=1,90
      NORMAL_PARAMETER(I)=COF(I)
 20 CONTINUE
      ANGLE_OLD=OUT(2)
      COUNT_DOWN=OUT(4)
      COUNTER=OUT(5)+1
      CRITICAL_ANGLE_DIFF=COF(91)
C
C NORMALIZE NORMAL_PARAMETER_VECTOR
C
      ABSOLUTE=0
      DO 30 I=1,90     ABSOLUTE=ABSOLUTE+NORMAL_PARAMETER(I)*NORMAL_PARAMETER(I)
 30 CONTINUE
      ABSOLUTE=SQRT(ABSOLUTE)
      DO 40 I=1,90
      NORMAL_PARAMETER(I)=NORMAL_PARAMETER(I)/ABSOLUTE
 40 CONTINUE
C
C NORMALIZE INPUT
C
      ABSOLUTE=0
      DO 50 I=1,90
      ABSOLUTE=ABSOLUTE+PARAMETER(I)*PARAMETER(I)
 50 CONTINUE
      ABSOLUTE=SQRT(ABSOLUTE)
      DO 60 I=1,90
      PARAMETER(I)=PARAMETER(I)/ABSOLUTE
 60 CONTINUE
C
C    CALCULATE DEG OF PARAMETER_VECTOR AND NORMAL_PARAMETER
C
      ANGLE_NEW=0
      DO 70 I=1,90  ANGLE_NEW=ANGLE_NEW+NORMAL_PARAMETER(I)*PARAMETER(I)
 70 CONTINUE
      ANGLE_NEW=ACOS(ANGLE_NEW)
      ANGLE_DIFF=ABS(ANGLE_NEW-ANGLE_OLD)
C
C  UPDATE OUTPUT
C
      STATUS=1
      DIFF_HIGH=CRITICAL_ANGLE_DIFF+0.1
```

```
      DIFF_LOW=CRITICAL_ANGLE_DIFF-0.1
      IF((ANGLE_NEW.GT.(DIFF_HIGH)).OR.(ANGLE_NEW.LT.(DIFF_LOW))
     &.OR.(ACCESSORY.EQ.4).OR.(ACCESSORY.EQ.11)
     &.OR.(ACCESSORY.EQ.16)) THEN
      STATUS=0
      ENDIF
      OUT(1)=STATUS
      OUT(2)=ANGLE_NEW
      OUT(3)=ANGLE_DIFF
      OUT(4)=COUNT_DOWN
      OUT(5)=COUNTER
  200 RETURN
      END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Algorithm 861

ขั้นตอนวิธีของส่วน SOFM Block

หน้าที่ สร้าง SOFM Map ขึ้นจากพารามิเตอร์ 9 ตัว

หลักการพื้นฐาน หัวข้อ 4.3 , 5.3 , 6.2.3

1.รายละเอียดในส่วนตัวแปรของค่านำเข้า

PARAMETER_A(1),PARAMETER_A(2),PARAMETER_A(3) : พารามิเตอร์ 3 ค่าแรกที่ใช้ในการสร้าง SOFM ชุดที่ 1 ของบล็อก

PARAMETER_B(1),PARAMETER_B(2),PARAMETER_B(3) : พารามิเตอร์ 3 ค่าแรกที่ใช้ในการสร้าง SOFM ชุดที่ 2 ของบล็อก

PARAMETER_C(1),PARAMETER_C(2),PARAMETER_C(3) : พารามิเตอร์ 3 ค่าแรกที่ใช้ในการสร้าง SOFM ชุดที่ 3 ของบล็อก

2.รายละเอียดในส่วนตัวแปรของค่านำออก

MAP_A(1,1),MAP_A(1,2),.....,MAP_A(10,3) : ค่าน้ำหนักประจำโนดของ SOFM ชุดที่ 1 ซึ่งเกิดจากการคำนวณแต่ละรอบและจะถูกส่งกลับมาคำนวณใหม่ในรอบถัดไป

MAP_B(1,1),MAP_B(1,2),.....,MAP_B(10,3) : ค่าน้ำหนักประจำโนดของ SOFM ชุดที่ 2 ซึ่งเกิดจากการคำนวณแต่ละรอบและจะถูกส่งกลับมาคำนวณใหม่ในรอบถัดไป

MAP_C(1,1),MAP_C(1,2),.....,MAP_C(10,3) : ค่าน้ำหนักประจำโนดของ SOFM ชุดที่ 3 ซึ่งเกิดจากการคำนวณแต่ละรอบและจะถูกส่งกลับมาคำนวณใหม่ในรอบถัดไป

TR_COUNT : ตัวแปรที่ทำหน้าที่นับจำนวนครั้งของการเรียนรู้

LEARN_READY :ตัวแปรที่ทำหน้าที่ส่งสัญญาณให้บล็อกอื่นๆทราบว่าได้สร้าง SOFM เรียบร้อยแล้วตามจำนวนรอบการคำนวณที่กำหนดในโปรแกรม

3.รายละเอียดในส่วนตัวแปรของค่าสัมประสิทธิ์

NUM_LEARNING : จำนวนรอบของการเรียนรู้ (50 รอบ , 100 รอบ , 150 รอบ)

4.ส่วนของโปรแกรม

```
SUBROUTINE ALG861(INP,OUT,COF,DT)
C
C AUTHOR: PHIPHAT.T
C
C DATE:9-June-1997
C
C REVISION NOTES:3_DIMENSION_SOFM_NEURAL_LEARNING_AND_CALCULATE
```

```
C
C
C
C
C ALGORITHM NUMBER: 861
C
C ALGORITHM NAME: SOFM_NEURAL_LEARNING
C
C ALGORITHM DESC: Self-Organizing Feature Maps
C           : 3x(3_Dimension with 10 Node)
C
C INPUTS: 10
C 1-3:PARAMETER_A(3),parameter input of sofm map A
C 4-6:PARAMETER_B(3),parameter input of sofm map B
C 7-9:PARAMETER_C(3),parameter input of sofm map C
C 10:SWITCH,(on/off 0/1)
C
C
C OUTPUTS: 90
C 1-30:MAP_A(10,3),map output of each step AC 31-60:MAP_B(10,3),map output of each step B
C 61-90:MAP_C(10,3),map output of each step C
C 91:TR_COUNT, count of iteration in transient
C 92:LEARN_READY,learning status(1/0 ready/not ready)
C
C COFFICIENT:1
C 1:NUM_LEARNING,number of iteration learning
C
C
C  DECLARATION
C
C
      DOUBLE PRECISION INP(*),OUT(*),COF(*),DT
      DOUBLE PRECISION PARAMETER_A(3),PARAMETER_B(3),PARAMETER_C(3)
      DOUBLE PRECISION LEARNRATE,MAP_A(10,3),MAP_B(10,3),MAP_C(10,3)
      DOUBLE PRECISION MAPINIT_A(10,3),MAPINIT_B(10,3),MAPINIT_C(10,3)
      DOUBLE PRECISION MAGNITUDE,ACTIVE_A,ACTIVE_B,ACTIVE_C
      DOUBLE PRECISION DEG_A(10),DEG_B(10),DEG_C(10),ABSOLUTE,X,Y,Z
      INTEGER I,J,K,SWITCH,TR_COUNT,LEARN_READY,NUM_LEARNING
C
C DEFINE FUNCTION ABSOLUTE
      ABSOLUTE(X,Y,Z)=SQRT(X*X+Y*Y+Z*Z)+.00001
C
C    DEFINE MAPS INITIAL
C
      MAPINIT_A(1,1)=0.185
      MAPINIT_A(1,2)=0.571
      MAPINIT_A(1,3)=-0.8
      MAPINIT_A(2,1)=-0.49
      MAPINIT_A(2,2)=0.35
      MAPINIT_A(2,3)=-0.8
      MAPINIT_A(3,1)=-0.49
      MAPINIT_A(3,2)=-0.35
      MAPINIT_A(3,3)=-0.8
      MAPINIT_A(4,1)=0.18
      MAPINIT_A(4,2)=-0.57
      MAPINIT_A(4,3)=-0.8
      MAPINIT_A(5,1)=0.6
      MAPINIT_A(5,2)=0
      MAPINIT_A(5,3)=-0.8
      MAPINIT_A(6,1)=0.185
      MAPINIT_A(6,2)=0.57
      MAPINIT_A(6,3)=0.8
      MAPINIT_A(7,1)=-0.485
      MAPINIT_A(7,2)=0.35
      MAPINIT_A(7,3)=0.8
      MAPINIT_A(8,1)=-0.485
      MAPINIT_A(8,2)=-0.35
      MAPINIT_A(8,3)=0.8
      MAPINIT_A(9,1)=0.185
      MAPINIT_A(9,2)=-0.57
      MAPINIT_A(9,3)=0.8
      MAPINIT_A(10,1)=0.6
      MAPINIT_A(10,2)=0
      MAPINIT_A(10,3)=0.8
      DO 10 I=1,10
      DO 20 J=1,3
```

```
      MAPINIT_B(I,J)=MAPINIT_A(I,J)
      MAPINIT_C(I,J)=MAPINIT_A(I,J)
   20 CONTINUE
   10 CONTINUE
C
C
C
C INPUTS, OUTPUTS AND COFS SETUP
C
      PARAMETER_A(1)=INP(1)
      PARAMETER_A(2)=INP(2)
      PARAMETER_A(3)=INP(3)
      PARAMETER_B(1)=INP(4)
      PARAMETER_B(2)=INP(5)
      PARAMETER_B(3)=INP(6)
      PARAMETER_C(1)=INP(7)
      PARAMETER_C(2)=INP(8)
      PARAMETER_C(3)=INP(9)
      NUM_LEARNING=COF(1)
      LEARN_READY=0
      SWITCH=INP(10)
      TR_COUNT=OUT(91)
      IF (SWITCH.EQ.1) GOTO 170
      IF (TR_COUNT.EQ.NUM_LEARNING) LEARN_READY=1
      IF (TR_COUNT.EQ.NUM_LEARNING) GOTO 160
      TR_COUNT=TR_COUNT+1
C
C NORMALIZE PARAMETER
C
      MAGNITUDE=ABSOLUTE(PARAMETER_A(1),PARAMETER_A(2),PARAMETER_A(3))
      PARAMETER_A(1)=PARAMETER_A(1)/MAGNITUDE
      PARAMETER_A(2)=PARAMETER_A(2)/MAGNITUDE
      PARAMETER_A(3)=PARAMETER_A(3)/MAGNITUDE
      MAGNITUDE=ABSOLUTE(PARAMETER_B(1),PARAMETER_B(2),PARAMETER_B(3))
      PARAMETER_B(1)=PARAMETER_B(1)/MAGNITUDE
      PARAMETER_B(2)=PARAMETER_B(2)/MAGNITUDE
      PARAMETER_B(3)=PARAMETER_B(3)/MAGNITUDE
      MAGNITUDE=ABSOLUTE(PARAMETER_C(1),PARAMETER_C(2),PARAMETER_C(3))
      PARAMETER_C(1)=PARAMETER_C(1)/MAGNITUDE
      PARAMETER_C(2)=PARAMETER_C(2)/MAGNITUDE
      PARAMETER_C(3)=PARAMETER_C(3)/MAGNITUDE
C
C INITIAL MAP
C
      K=1
      DO 35 I=1,10
      DO 30 J=1,3
      MAP_A(I,J)=OUT(K)
      K=K+1
   30 CONTINUE
   35 CONTINUE
      K=1
      DO 45 I=1,10
      DO 40 J=1,3
      MAP_B(I,J)=OUT(K)
      K=K+1
   40 CONTINUE
   45 CONTINUE
      K=1
      DO 55 I=1,10
      DO 50 J=1,3
      MAP_C(I,J)=OUT(K)
      K=K+1
   50 CONTINUE
   55 CONTINUE
C
C  NORMALIZE MAP_VECTOR
C
      DO 60 I=1,10
      MAGNITUDE=ABSOLUTE(MAP_A(I,1),MAP_A(I,2),MAP_A(I,3))
      MAP_A(I,1)=MAP_A(I,1)/MAGNITUDE
      MAP_A(I,2)=MAP_A(I,2)/MAGNITUDE
      MAP_A(I,3)=MAP_A(I,3)/MAGNITUDE
      MAGNITUDE=ABSOLUTE(MAP_B(I,1),MAP_B(I,2),MAP_B(I,3))
      MAP_B(I,1)=MAP_B(I,1)/MAGNITUDE
      MAP_B(I,2)=MAP_B(I,2)/MAGNITUDE
```

```
      MAP_B(I,3)=MAP_B(I,3)/MAGNITUDE
      MAGNITUDE=ABSOLUTE(MAP_C(I,1),MAP_C(I,2),MAP_C(I,3))
      MAP_C(I,1)=MAP_C(I,1)/MAGNITUDE
      MAP_C(I,2)=MAP_C(I,2)/MAGNITUDE
      MAP_C(I,3)=MAP_C(I,3)/MAGNITUDE
   60 CONTINUE
C
C     CALCULATE DEG OF MAP_VECTOR AND PARAMETER_VECTOR
C
      DO 70 I=1,10
      DEG_A(I)=PARAMETER_A(1)*MAPINIT_A(I,1)
     #+PARAMETER_A(2)*MAPINIT_A(I,2)+PARAMETER_A(3)*MAPINIT_A(I,3)
      DEG_B(I)=PARAMETER_B(1)*MAPINIT_B(I,1)
     #+PARAMETER_B(2)*MAPINIT_B(I,2)+PARAMETER_B(3)*MAPINIT_B(I,3)
      DEG_C(I)=PARAMETER_C(1)*MAPINIT_C(I,1)
     #+PARAMETER_C(2)*MAPINIT_C(I,2)+PARAMETER_C(3)*MAPINIT_C(I,3)
   70 CONTINUE
C
C    FIND NEARLY VECTOR
C

      ACTIVE_A=1
      ACTIVE_B=1
      ACTIVE_C=1
      DO 80 I=2,10
      IF (DEG_A(I).GT.DEG_A(ACTIVE_A)) ACTIVE_A=I
      IF (DEG_B(I).GT.DEG_B(ACTIVE_B)) ACTIVE_B=I
      IF (DEG_C(I).GT.DEG_C(ACTIVE_C)) ACTIVE_C=I
   80 CONTINUE
C
C    DEFINE LEARNRATE
C
      LEARNRATE=1
C
C
C     UPDATE WEIGHT
C
      DO 90 J=1,3
      MAP_A(ACTIVE_A,J)=MAP_A(ACTIVE_A,J)+LEARNRATE*(PARAMETER_A(J)
     #-MAP_A(ACTIVE_A,J))
      MAP_B(ACTIVE_B,J)=MAP_B(ACTIVE_B,J)+LEARNRATE*(PARAMETER_B(J)
     #-MAP_B(ACTIVE_B,J))
      MAP_C(ACTIVE_C,J)=MAP_C(ACTIVE_C,J)+LEARNRATE*(PARAMETER_C(J)
     #-MAP_C(ACTIVE_C,J))
   90 CONTINUE
      MAGNITUDE=ABSOLUTE(MAP_A(ACTIVE_A,1),MAP_A(ACTIVE_A,2),
     #MAP_A(ACTIVE_A,3))
      MAP_A(ACTIVE_A,1)=MAP_A(ACTIVE_A,1)/MAGNITUDE
      MAP_A(ACTIVE_A,2)=MAP_A(ACTIVE_A,2)/MAGNITUDE
      MAP_A(ACTIVE_A,3)=MAP_A(ACTIVE_A,3)/MAGNITUDE
C
      MAGNITUDE=ABSOLUTE(MAP_B(ACTIVE_B,1),MAP_B(ACTIVE_B,2),
     #MAP_B(ACTIVE_B,3))
      MAP_B(ACTIVE_B,1)=MAP_B(ACTIVE_B,1)/MAGNITUDE
      MAP_B(ACTIVE_B,2)=MAP_B(ACTIVE_B,2)/MAGNITUDE
      MAP_B(ACTIVE_B,3)=MAP_B(ACTIVE_B,3)/MAGNITUDE
C
      MAGNITUDE=ABSOLUTE(MAP_C(ACTIVE_C,1),MAP_C(ACTIVE_C,2),
     #MAP_C(ACTIVE_C,3))
      MAP_C(ACTIVE_C,1)=MAP_C(ACTIVE_C,1)/MAGNITUDE
      MAP_C(ACTIVE_C,2)=MAP_C(ACTIVE_C,2)/MAGNITUDE
      MAP_C(ACTIVE_C,3)=MAP_C(ACTIVE_C,3)/MAGNITUDE
C
C
C UPDATE OUTPUT
C
      k=1
      DO 100 I=1,10
      DO 110 J=1,3
      OUT(K)=MAP_A(I,J)
      K=K+1
  110 CONTINUE
  100 CONTINUE
      DO 120 I=1,10
      DO 130 J=1,3
      OUT(K)=MAP_B(I,J)
```

```
      K=K+1
130 CONTINUE
120 CONTINUE
      DO 140 I=1,10
      DO 150 J=1,3
      OUT(K)=MAP_C(I,J)
      K=K+1
150 CONTINUE
140 CONTINUE
160 OUT(91)=TR_COUNT
      OUT(92)=LEARN_READY
170 RETURN
      END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Algorithm 862

ขั้นตอนวิธีของส่วน **Map Distance Block**

หน้าที่ คำนวณค่าผลรวมของระยะทางจากเวกเตอร์ของ SOFM ที่ได้จากการเรียนรู้ใหม่กับ SOFM หลัก

หลักการพื้นฐาน หัวข้อ 4.4 , 5.4

1.รายละเอียดในส่วนตัวแปรของค่านำเข้า

WEIGHT_A(1,1),WEIGHT_A(1,2),.....,WEIGHT_A(10,3) : ค่าน้ำหนักประจำโนดของ SOFM ชุดที่ 1 ซึ่งส่งมาเพื่อคำนวณระยะทาง

WEIGHT_B(1,1),WEIGHT_B(1,2),.....,WEIGHT_B(10,3) : ค่าน้ำหนักประจำโนดของ SOFM ชุดที่ 2 ซึ่งส่งมาเพื่อคำนวณระยะทาง

WEIGHT_C(1,1),WEIGHT_C(1,2),.....,WEIGHT_C(10,3) : ค่าน้ำหนักประจำโนดของ SOFM ชุดที่ 3 ซึ่งส่งมาเพื่อคำนวณระยะทาง

SWITCH : ตัวแปรซึ่งกำหนดการเริ่มต้นทำงานของ Map Distance Block

2.รายละเอียดในส่วนตัวแปรของค่านำออก

DISTANCE_A(1) , ... , DISTANCE_A(10) : ค่าระยะทางของ SOFM ชุดที่ 1 ที่คำนวณได้

DISTANCE_B(1) , ... , DISTANCE_B(10) : ค่าระยะทางของ SOFM ชุดที่ 2 ที่คำนวณได้

DISTANCE_C(1) , ... , DISTANCE_C(10) : ค่าระยะทางของ SOFM ชุดที่ 3 ที่คำนวณได้

DISPLACE_A : ค่าระยะขจัดของ SOFM ชุดที่ 1 ที่คำนวณได้

DISPLACE_B : ค่าระยะขจัดของ SOFM ชุดที่ 2 ที่คำนวณได้

DISPLACE_C : ค่าระยะขจัดของ SOFM ชุดที่ 3 ที่คำนวณได้

3.รายละเอียดในส่วนตัวแปรของค่าสัมประสิทธิ์

BASE_DISTANCE_A(1) , ..., BASE_DISTANCE_A(10) : ค่าระยะทางของ SOFM ชุดที่ 1 ที่ผ่านการเรียนรู้ไว้แล้ว

BASE_DISTANCE_A(1) , ..., BASE_DISTANCE_B(10) : ค่าระยะทางของ SOFM ชุดที่ 2 ที่ผ่านการเรียนรู้ไว้แล้ว

BASE_DISTANCE_A(1) , ..., BASE_DISTANCE_C(10) : ค่าระยะทางของ SOFM ชุดที่ 3 ที่ผ่านการเรียนรู้ไว้แล้ว

4.ส่วนของโปรแกรม

```
      SUBROUTINE ALG862(INP,OUT,COF,DT)
C
C AUTHOR: PHIPHAT.T
C
C DATE:4-OCT-1997
C
C REVISION NOTES:MAP_DISTANCE BLOCK
C
C
C
C
C ALGORITHM NUMBER: 862
C
C ALGORITHM NAME: MAP_DISTANCE
C
C ALGORITHM DESC: CALCULATE DEGREE BETWEEN SOFM
C
C
C INPUTS: 91
C 1-30:WEIGHT_A(10,3),weight of sofm map A
C 31-60:WEIGHT_B(10,3),weight of sofm map B
C 61-90:WEIGHT_C(10,3),weight of sofm map C
C 91:SWITCH,switch to calculate(1/0 calculate/not calculate)
C
C
C OUTPUTS: 33
C 1-10:DISTANCE_A(10), distance between sofm_a_input and sofm_a_learning
C 11-20:DISTANCE_B(10), distance between sofm_b_input and sofm_b_learning
C 21-30:DISTANCE_C(10), distance between sofm_c_input and sofm_c_learning
C 31:DISPLACE_A
C 32:DISPLACE_B
C 33:DISPLACE_C
C
C COFFICIENT:30
C 1-10:BASE_DISTANCE_A(10),base vector distance a
C 11-20:BASE_DISTANCE_B(10),base vector distance b
C 21-30:BASE_DISTANCE_C(10),base vector distance c
C
C  DECLARATION
C
C
      DOUBLE PRECISION INP(*),OUT(*),COF(*),DT
      DOUBLE PRECISION WEIGHT_A(10,3),WEIGHT_B(10,3),WEIGHT_C(10,3)
      DOUBLE PRECISION DISTANCE_A(10),DISTANCE_B(10)
      DOUBLE PRECISION DISTANCE_C(10)
      DOUBLE PRECISION BASE_DISTANCE_A(10),BASE_DISTANCE_B(10)
      DOUBLE PRECISION BASE_DISTANCE_C(10),DISPLACE_A,DISPLACE_B
      DOUBLE PRECISION DISPLACE_C,MAPINIT_A(10,3),MAPINIT_B(10,3)
      DOUBLE PRECISION MAPINIT_C(10,3),SWITCH
      INTEGER I,J,K
C
      MAPINIT_A(1,1)=0.185
      MAPINIT_A(1,2)=0.571
      MAPINIT_A(1,3)=-0.8
      MAPINIT_A(2,1)=-0.49
      MAPINIT_A(2,2)=0.35
      MAPINIT_A(2,3)=-0.8
      MAPINIT_A(3,1)=-0.49
      MAPINIT_A(3,2)=-0.35
      MAPINIT_A(3,3)=-0.8
      MAPINIT_A(4,1)=0.18
      MAPINIT_A(4,2)=-0.57
      MAPINIT_A(4,3)=-0.8
      MAPINIT_A(5,1)=0.6
      MAPINIT_A(5,2)=0
      MAPINIT_A(5,3)=-0.8
      MAPINIT_A(6,1)=0.185
      MAPINIT_A(6,2)=0.57
      MAPINIT_A(6,3)=0.8
```

```
      MAPINIT_A(7,1)=-0.485
      MAPINIT_A(7,2)=0.35
      MAPINIT_A(7,3)=0.8
      MAPINIT_A(8,1)=-0.485
      MAPINIT_A(8,2)=-0.35
      MAPINIT_A(8,3)=0.8
      MAPINIT_A(9,1)=0.185
      MAPINIT_A(9,2)=-0.57
      MAPINIT_A(9,3)=0.8
      MAPINIT_A(10,1)=0.6
      MAPINIT_A(10,2)=0
      MAPINIT_A(10,3)=0.8
      DO 1 I=1,10
      DO 2 J=1,3
      MAPINIT_B(I,J)=MAPINIT_A(I,J)
      MAPINIT_C(I,J)=MAPINIT_A(I,J)
    2 CONTINUE
    1 CONTINUE
C
C
C
C INPUTS, OUTPUTS AND COFS SETUP
C
C
C  INPUT SETUP
      K=1
      DO 20 I=1,10
      DO 10 J=1,3
      WEIGHT_A(I,J)=INP(K)
      K=K+1
   10 CONTINUE
   20 CONTINUE
      DO 40 I=1,10
      DO 30 J=1,3
      WEIGHT_B(I,J)=INP(K)
      K=K+1
   30 CONTINUE
   40 CONTINUE
      DO 60 I=1,10
      DO 50 J=1,3
      WEIGHT_C(I,J)=INP(K)
      K=K+1
   50 CONTINUE
   60 CONTINUE
      SWITCH=INP(91)
      IF (SWITCH.EQ.0) GOTO 220
      DO 65 I=1,10
      BASE_DISTANCE_A(I)=COF(I)
      BASE_DISTANCE_B(I)=COF(I+10)
      BASE_DISTANCE_C(I)=COF(I+20)
   65 CONTINUE
C
C
C
C    CALCULATE DEG OF WEIGHT_MAP AND LEARNING_MAP
C
      DO 70 I=1,10
      DISTANCE_A(I)=WEIGHT_A(I,1)*MAPINIT_A(I,1)+
     #WEIGHT_A(I,2)*MAPINIT_A(I,2)+WEIGHT_A(I,3)*MAPINIT_A(I,3)
      DISTANCE_B(I)=WEIGHT_B(I,1)*MAPINIT_B(I,1)+
     #WEIGHT_B(I,2)*MAPINIT_B(I,2)+WEIGHT_B(I,3)*MAPINIT_B(I,3)
      DISTANCE_C(I)=WEIGHT_C(I,1)*MAPINIT_C(I,1)+
     #WEIGHT_C(I,2)*MAPINIT_C(I,2)+WEIGHT_C(I,3)*MAPINIT_C(I,3)
      IF (DISTANCE_A(I).GE.0.99) DISTANCE_A(I)=0
      IF (DISTANCE_B(I).GE.0.99) DISTANCE_B(I)=0
      IF (DISTANCE_C(I).GE.0.99) DISTANCE_C(I)=0
   70 CONTINUE
C
C  UPDATE OUTPUT
C
      DISPLACE_A=0
      DISPLACE_B=0
      DISPLACE_C=0
      DO 80 I=1,10
      DISPLACE_A=DISPLACE_A+ABS(DISTANCE_A(I)-BASE_DISTANCE_A(I))
      DISPLACE_B=DISPLACE_B+ABS(DISTANCE_B(I)-BASE_DISTANCE_B(I))
```

```
      DISPLACE_C=DISPLACE_C+ABS(DISTANCE_C(I)-BASE_DISTANCE_C(I))
      OUT(I)=DISTANCE_A(I)
      OUT(I+10)=DISTANCE_B(I)
      OUT(I+20)=DISTANCE_C(I)
   80 CONTINUE
      OUT(31)=10-DISPLACE_A
      OUT(32)=10-DISPLACE_B
      OUT(33)=10-DISPLACE_C
  220 RETURN
      END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Algorithm 863

ขั้นตอนวิธีของส่วน Map Identify Block

หน้าที่ คำนวณเปรียบเทียบระยะขจัดเพื่อแสดงลำดับของภาวะทรานเซียนต์ที่วิเคราะห์ได้

หลักการพื้นฐาน หัวข้อ 4.4 , 4.5

1.รายละเอียดในส่วนตัวแปรของค่านำเข้า

VECTOR_DISTANCE(1) , ... ,VECTOR_DISTANCE(12) : ผลรวมทางสถิติของระยะทางจาก SOFM 12 ชุด

SWITCH : ตัวแปรซึ่งกำหนดการเริ่มต้นทำงานของ Map Identify Block

2.รายละเอียดในส่วนตัวแปรของค่านำออก

COMPETITIVE : ตัวแปรที่แทนค่าความเหมือนกันของ SOFM แต่ละชุด

3.รายละเอียดในส่วนตัวแปรของค่าสัมประสิทธิ์

VECTOR_BASE(1) , ..., VECTOR_BASE(12) : ผลรวมทางสถิติของระยะทางจาก SOFM 12 ชุดที่เก็บไว้ในฐานข้อมูล

4.ส่วนของโปรแกรม

```
   SUBROUTINE ALG863(INP,OUT,COF,DT)
C
C AUTHOR: PHIPHAT.T
C
C DATE:5-DEC-1997
C
C REVISION NOTES:MAP_IDENTIFY
C
C
C
C
C ALGORITHM NUMBER: 863
C
C ALGORITHM NAME: IDENTIFY
C
C ALGORITHM DESC:TRANSIENT MAP IDENTIFY
C INPUT:4
C 1-12:VECTOR_DISTANCE(1-12),vector of distance
C     which calculate from map number 1-12
C 13:SWITCH, switch to calculate (ON/OFF 1/0)
C
C OUTPUT:1
C 1:COMPETITIVE,score which identify the nearest vector
C
C
C COEFFICIENTS: 12
C 1-12:VECTOR_BASE(1-12),vector of distance which calculate from real operate
```

```
C
C DECLARATION
C
      DOUBLE PRECISION INP(*),OUT(*),COF(*),DT
C
      DOUBLE PRECISION VECTOR_DISTANCE(12),ABSOLUTE_VECTOR_DISTANCE
      DOUBLE PRECISION VECTOR_BASE(12),ABSOLUTE_VECTOR_BASE
      DOUBLE PRECISION COMPETITIVE
      INTEGER SWITCH,I
C
C
C
C
C INPUTS, OUTPUTS AND COFS SETUP
C
       DO 10 I=1,12
       VECTOR_DISTANCE(I)=INP(I)
       VECTOR_BASE(I)=COF(I)
 10 CONTINUE
       SWITCH=INP(13)
       IF (SWITCH.EQ.0) GOTO 200
C
C CALCULATE NORMALIZE VECTOR
C
       ABSOLUTE_VECTOR_DISTANCE=0
       ABSOLUTE_VECTOR_BASE=0
       DO 20 I=1,12
       ABSOLUTE_VECTOR_DISTANCE=ABSOLUTE_VECTOR_DISTANCE
      #+VECTOR_DISTANCE(I)*VECTOR_DISTANCE(I)
       ABSOLUTE_VECTOR_BASE=ABSOLUTE_VECTOR_BASE
      #+VECTOR_BASE(I)*VECTOR_BASE(I)
 20 CONTINUE
       ABSOLUTE_VECTOR_DISTANCE=SQRT(ABSOLUTE_VECTOR_DISTANCE)
       ABSOLUTE_VECTOR_BASE=SQRT(ABSOLUTE_VECTOR_BASE)
       DO 30 I=1,12
       VECTOR_DISTANCE(I)=VECTOR_DISTANCE(I)/ABSOLUTE_VECTOR_DISTANCE
       VECTOR_BASE(I)=VECTOR_BASE(I)/ABSOLUTE_VECTOR_BASE
 30 CONTINUE
C
C CALCULATE THE COMPETITIVE DATA
C
       COMPETITIVE=0
       DO 40 I=1,12
       COMPETITIVE=COMPETITIVE+VECTOR_DISTANCE(I)*VECTOR_BASE(I)
 40 CONTINUE
C
C UPDATE OUTPUT
C
 200 OUT(1)=COMPETITIVE
       RETURN
       END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Algorithm 864

ขั้นตอนวิธีของส่วน Data to normalize Block

หน้าที่ ทำหน้าที่แปลงค่าพารามิเตอร์จากโรงไฟฟ้าฯทุกค่าให้เป็นค่าบรรทัดฐาน

หลักการพื้นฐาน หัวข้อ 4.1

1.รายละเอียดในส่วนตัวแปรของค่านำเข้า

PARAMETER(1) ,.....,PARAMETER(30) : พารามิเตอร์ 30 ชุด

MIN(1) , ..., MIN(30) : ค่าต่ำสุดของพารามิเตอร์ทั้ง 30 ชุด

MAX(1) , ..., MAX(30) : ค่าสูงสุดของพารามิเตอร์ทั้ง 30 ชุด

2.รายละเอียดในส่วนตัวแปรของค่านำออก

NOR(1) ,..... , NOR(30) : ค่าบรรทัดฐานของพารามิเตอร์ที่โปรแกรมคำนวณได้

3.รายละเอียดในส่วนตัวแปรของค่าสัมประสิทธิ์

NUMBER_PAR : กำหนดจำนวนพารามิเตอร์ที่ใช้ในการคำนวณ

4.ส่วนของโปรแกรม

```
SUBROUTINE ALG864(INP,OUT,COF,DT)
C
C AUTHOR: PHIPHAT.T
C
C DATE:16-MARCH-1997
C
C REVISION NOTES: TRANSFER FUNCTION DATA TO -1....1(NORMALIZE)
C
C 16-03-97 CLI
C
C
C ALGORITHM NUMBER: 864
C
C ALGORITHM NAME: DATA_TO_NORMALIZE  BLOCK
C
C ALGORITHM DESC: GENERAL DATA TO NORMALIZE DATA
C
C INPUTS: 60
C 1-30 :PARAMETER,Parameter to Normalize
C 31-90: MIN-MAX,Minimun Maximum Data 1-60
C OUTPUTS: 30
C 1-30 :NOR1-30 ,Normalize Data Output1-30
C COEFFICIENTS: 1
C 1:NUMBER_PAR , Number of Parameter
C
C DECLARATION
C
```

```
      DOUBLE PRECISION INP(*),OUT(*),COF(*),DT
C
      DOUBLE PRECISION MIN(30),MAX(30)
      DOUBLE PRECISION PARAMETER(30),NOR(30)
      INTEGER I,P,NUMBER_PAR
C
C
C INPUTS, OUTPUTS AND COFS SETUP
C
      NUMBER_PAR=COF(1)
      P=1
      DO 10 I=1,NUMBER_PAR
      PARAMETER(I)=INP(I)
      MIN(I)=INP(P+30)
      MAX(I)=INP((P+1)+30)
      P=P+2
 10 CONTINUE
C  PROG BEGINS
C
C------CALCULATE NORMALIZE DATA
      DO 20 I=1,NUMBER_PAR
      NOR(I)=TAN(0.7854*(2*PARAMETER(I)-MAX(I)-MIN(I))
     &/(MAX(I)-MIN(I)+0.00001))
 20 CONTINUE
C  UPDATE OUTPUT
      DO 50 I=1,NUMBER_PAR
      OUT(I)=NOR(I)
 50 CONTINUE
      RETURN
      END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Algorithm 865

ขั้นตอนวิธีของส่วน SOFM_SUM

หน้าที่ คำนวณค่าเฉลี่ยของระยะขจัดที่เกิดจาก SOFM 12 ชุด

หลักการพื้นฐาน หัวข้อ 4.5 , 5.5

1.รายละเอียดในส่วนตัวแปรของค่านำเข้า

DISTANCE(1) ,....., DISTANCE(12) : ค่าระยะทาง 12 ค่าเพื่อมาหาค่าเฉลี่ย

2.รายละเอียดในส่วนตัวแปรของค่านำออก

RESULT : ผลของค่าเฉลี่ยที่คำนวณได้

3.รายละเอียดในส่วนตัวแปรของค่าสัมประสิทธิ์

NUMBER_DIST : กำหนดจำนวนค่าระยะทางที่ใช้ในการคำนวณ

4.ส่วนของโปรแกรม

```
 SUBROUTINE ALG865(INP,OUT,COF,DT)
C
C AUTHOR: PHIPHAT.T
C
C DATE:7-NOV-1997
C
C REVISION NOTES:SOFM_SUM
C
C
C
C
C ALGORITHM NUMBER: 865
C
C ALGORITHM NAME: SOFM_SUM
C
C ALGORITHM DESC: AVERAGE FOR INPUT SUM
C
C
C INPUTS: 12
C 1-12:DISTANCE(12),input
C
C
C OUTPUTS:1
C 1:RESULT,output parameter for sum calculate
C
C
C
C COFFICIENT:1
C 1:NUM_DIST,number of input
C
C  DECLARATION
C
```

```
C
      DOUBLE PRECISION INP(*),OUT(*),COF(*),DT
      DOUBLE PRECISION DISTANCE(12),RESULT,NUM_DIST
      INTEGER I
C
C INPUTS AND COFS SETUP
      NUM_DIST=COF(1)
C
C  INPUT SETUP
      RESULT=0
      DO 10 I=1,NUM_DIST
      DISTANCE(I)=INP(I)
      RESULT=RESULT+DISTANCE(I)
 10 CONTINUE
C     RESULT=RESULT/NUM_DIST
C
C UPDATE OUTPUT
C
      OUT(1)=RESULT
      RETURN
      END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Algorithm 866

ขั้นตอนวิธีของส่วน INDICATOR FOR TRANSIENT

หน้าที่ คำนวณหาค่าที่สูงสุดของระยะขจัดจากระยะขจัดทั้งหมด

1.รายละเอียดในส่วนตัวแปรของค่านำเข้า

DISPLACE(1) ,....., DISPLACE(23) : ค่าระยะขจัดที่มาคำนวณเพื่อหาค่าสูงสุด

2.รายละเอียดในส่วนตัวแปรของค่านำออก

RESULT : ลำดับที่ของระยะขจัดที่มีค่าสูงสุด

3.รายละเอียดในส่วนตัวแปรของค่าสัมประสิทธิ์

ไม่มีค่าสัมประสิทธิ์

4.ส่วนของโปรแกรม

```
SUBROUTINE ALG866(INP,OUT,COF,DT)
C
C AUTHOR: PHIPHAT.T
C
C DATE:7-NOV-1997
C
C REVISION NOTES:INDICATOR
C
C
C
C
C ALGORITHM NUMBER: 866
C
C ALGORITHM NAME: INDICATOR
C
C ALGORITHM DESC: INDICATOR FOR TRANSIENT
C
C
C INPUTS: 23
C 1-23:DISPLACE(23),Input
C
C
C OUTPUTS:1
C 1:RESULT,output number of transient
C
C
C
C COFFICIENT:0
C
C
C  DECLARATION
C
C
```

```
      DOUBLE PRECISION INP(*),OUT(*),COF(*),DT
      DOUBLE PRECISION DISPLACE(23),RESULT
      INTEGER I
C
C INPUTS AND COFS SETUP
C
C  INPUT SETUP
      RESULT=0
      DO 10 I=1,23
      DISPLACE(I)=INP(I)
      RESULT=DISPLACE(I)+RESULT
 10 CONTINUE
      IF (RESULT.EQ.0) GOTO 200
      RESULT=1
      DO 20 I=2,23
      IF (DISPLACE(I).GT.DISPLACE(RESULT)) RESULT=I
 20 CONTINUE
C
C
C  UPDATE OUTPUT
C
      OUT(1)=RESULT
200 RETURN
      END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Algorithm 867

ขั้นตอนวิธีของส่วน DECISION_BLOCK

หน้าที่ คำนวณตัดสินลำดับของภาวะทรานเซียนต์จริงจากภาวะทรานเซียนต์ที่เกิดขึ้นที่ 50 รอบ , 100 รอบ , 150 รอบ ซึ่งเก็บไว้ในตารางที่ 5.5

หลักการพื้นฐาน หัวข้อ 6.1.1

1.รายละเอียดในส่วนตัวแปรของค่านำเข้า

TRANSIENT(1) ,....., TRANSIENT(3) : ลำดับของภาวะทรานเซียนต์ซึ่งได้จากการเรียนรู้ที่ 50 รอบ , 100 รอบ , 150 รอบ ตามลำดับ

2.รายละเอียดในส่วนตัวแปรของค่านำออก

TRANSIENT_NUM : ชนิดของภาวะทรานเซียนต์ที่เกิดขึ้นในขั้นตอนสุดท้าย

3.รายละเอียดในส่วนตัวแปรของค่าสัมประสิทธิ์

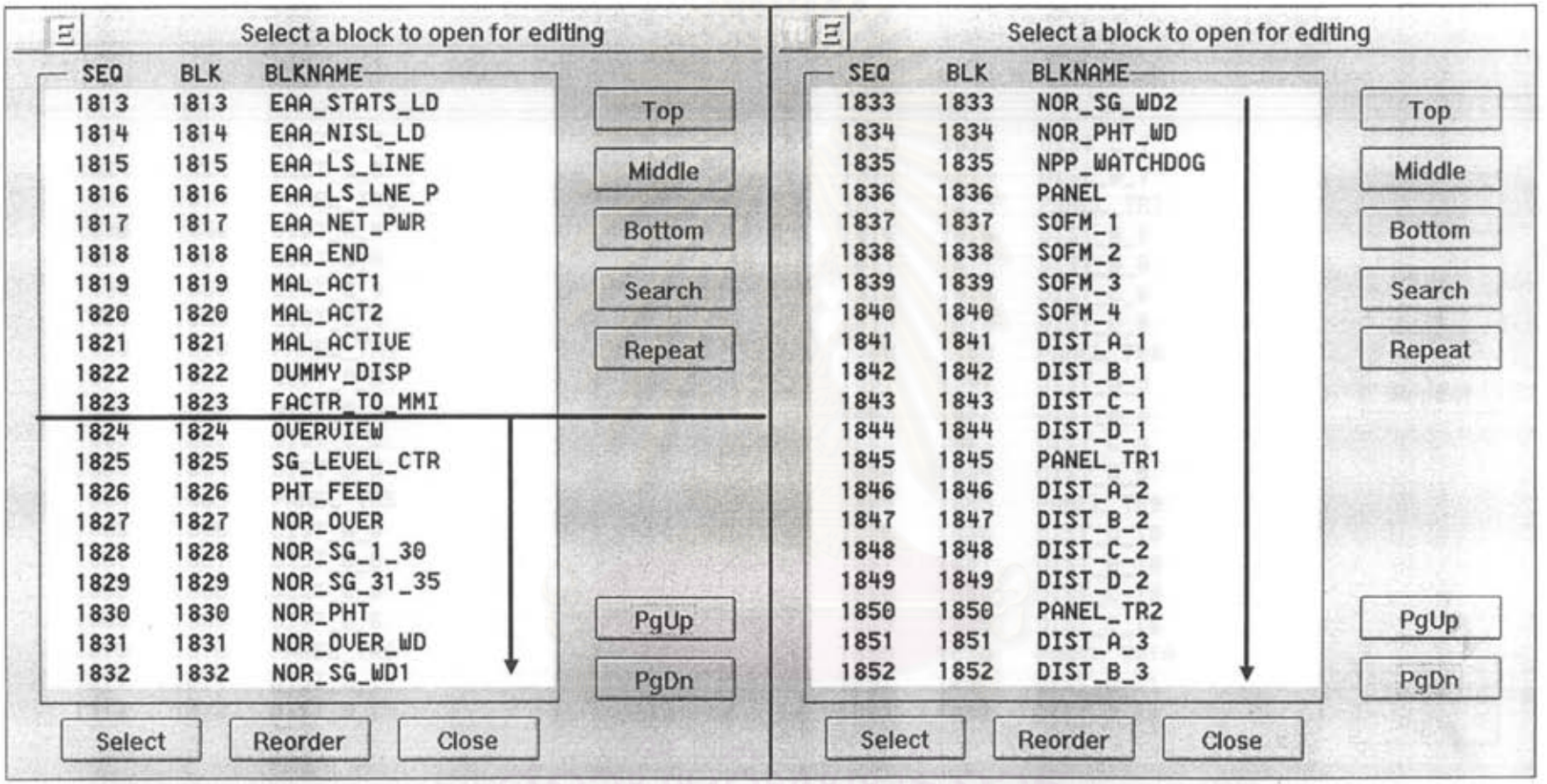
ไม่มีค่าสัมประสิทธิ์

4.ส่วนของโปรแกรม

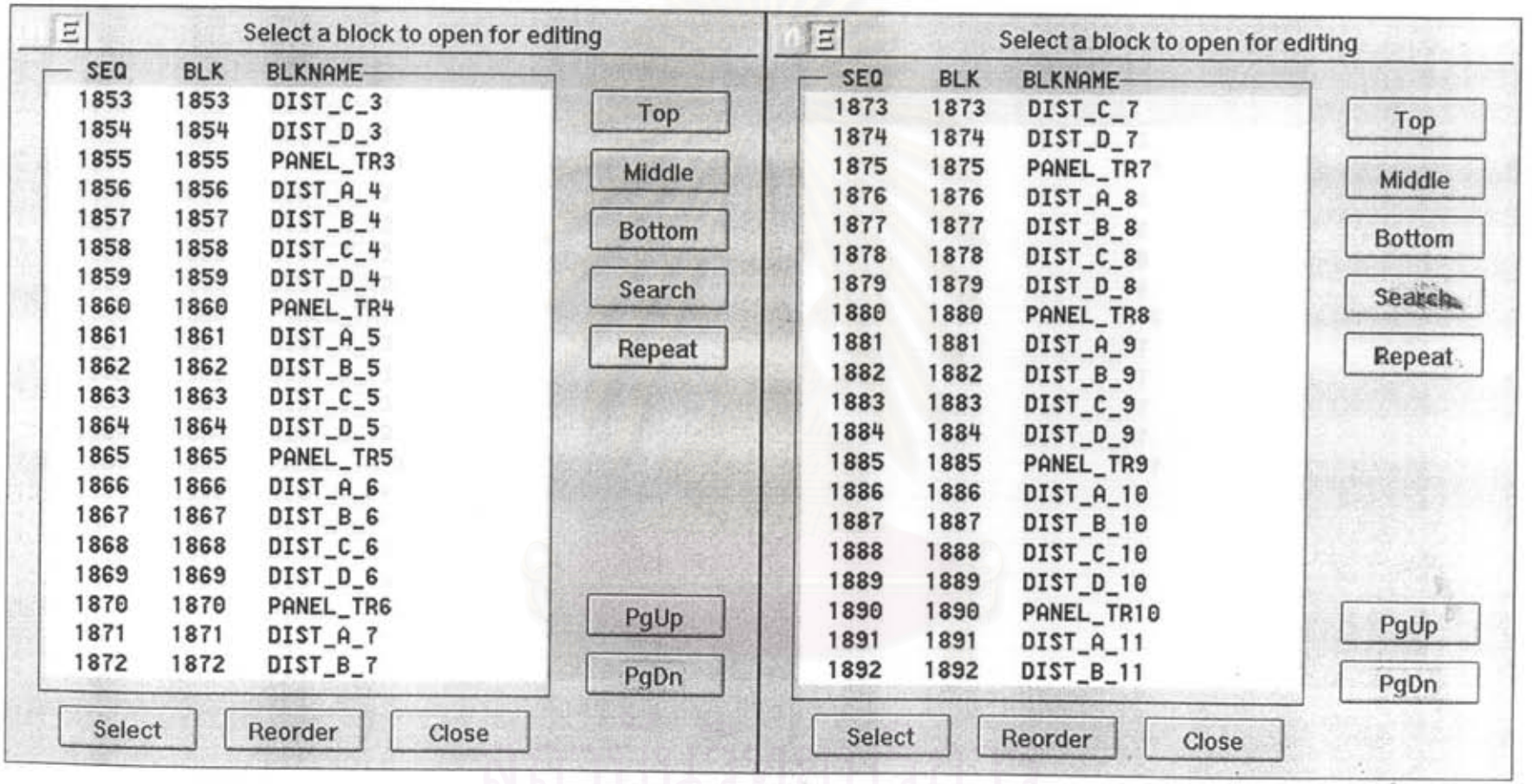
```
SUBROUTINE ALG867(INP,OUT,COF,DT)

C
C AUTHOR: PHIPHAT.T
C
C DATE:7-NOV-1997
C
C REVISION NOTES:DECISION OF TRANSIENT FROM 50,100,150 ITERATION
C
C
C
C
C ALGORITHM NUMBER: 867
C
C ALGORITHM NAME: DECISION_BLOCK
C
C ALGORITHM DESC: IT HAVE 3 TRANSIENTS FROM SOFM TO DECISION BY THIS BLOCK
C
C
C INPUTS: 3
C 1-3:TRANSIENT(3),input of TRANSIENT
C
C OUTPUTS:1
C 1:TRANSIENT_NUM,output of transient number in the final
C
C COFFICIENT:1
```

```
C
C
C  DECLARATION
C
C
      DOUBLE PRECISION INP(*),OUT(*),COF(*),DT
      DOUBLE PRECISION TRANSIENT(3)
      INTEGER TRANSIENT_NUM
C
C INPUTS AND COFS SETUP
C
C INPUT SETUP
      TRANSIENT(1)=INP(1)
      TRANSIENT(2)=INP(2)
      TRANSIENT(3)=INP(3)
C
      TRANSIENT_NUM=TRANSIENT(3)
      IF ((TRANSIENT_NUM.EQ.13).AND.(TRANSIENT(2).EQ.10)) TRANSIENT_NUM=10
      IF ((TRANSIENT(1).EQ.21).OR.(TRANSIENT(2).EQ.21).OR.(TRANSIENT(3).EQ.21))
     &TRANSIENT_NUM=21
C
C  UPDATE OUTPUT
C
      OUT(1)=TRANSIENT_NUM
      RETURN
      END
```

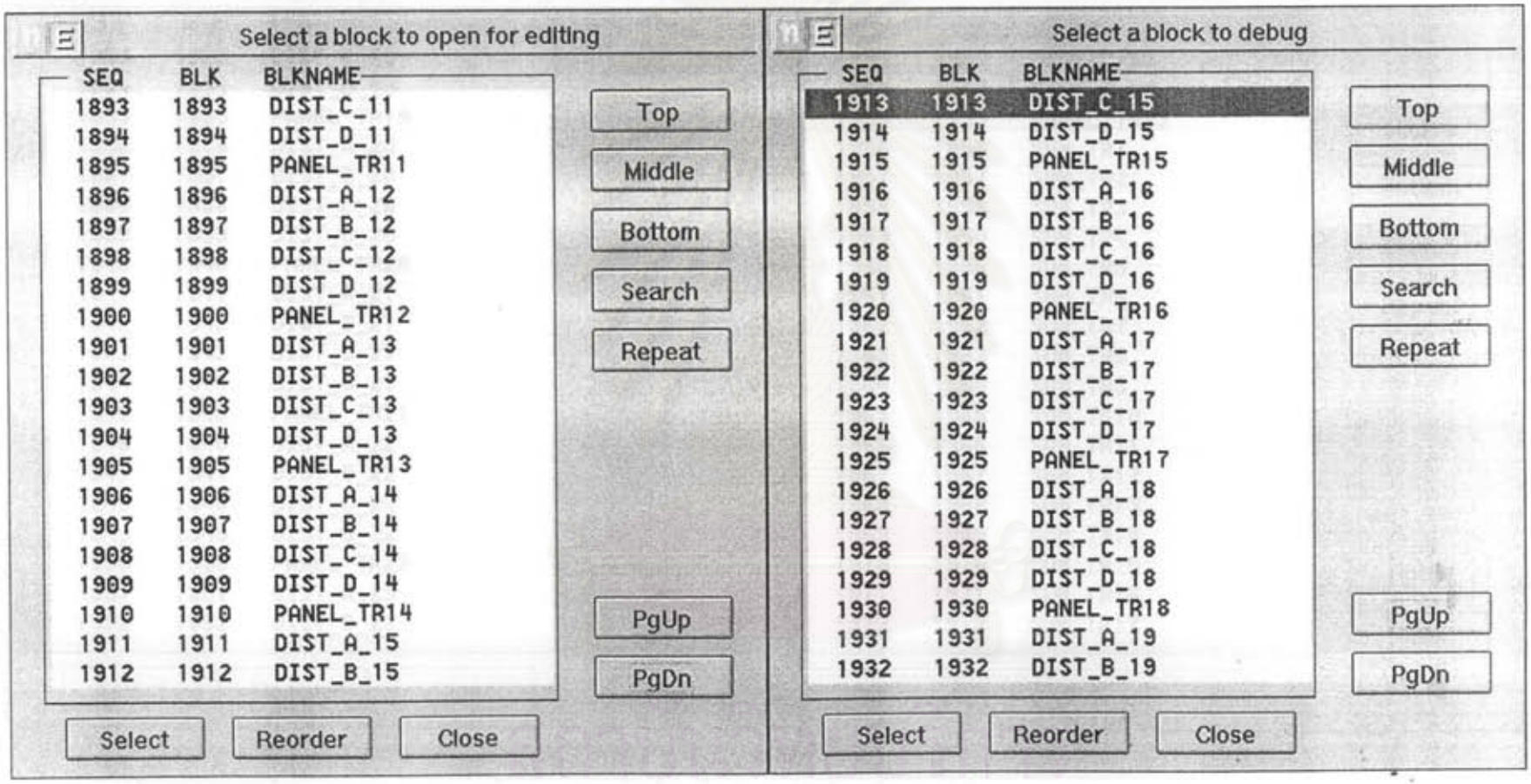
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



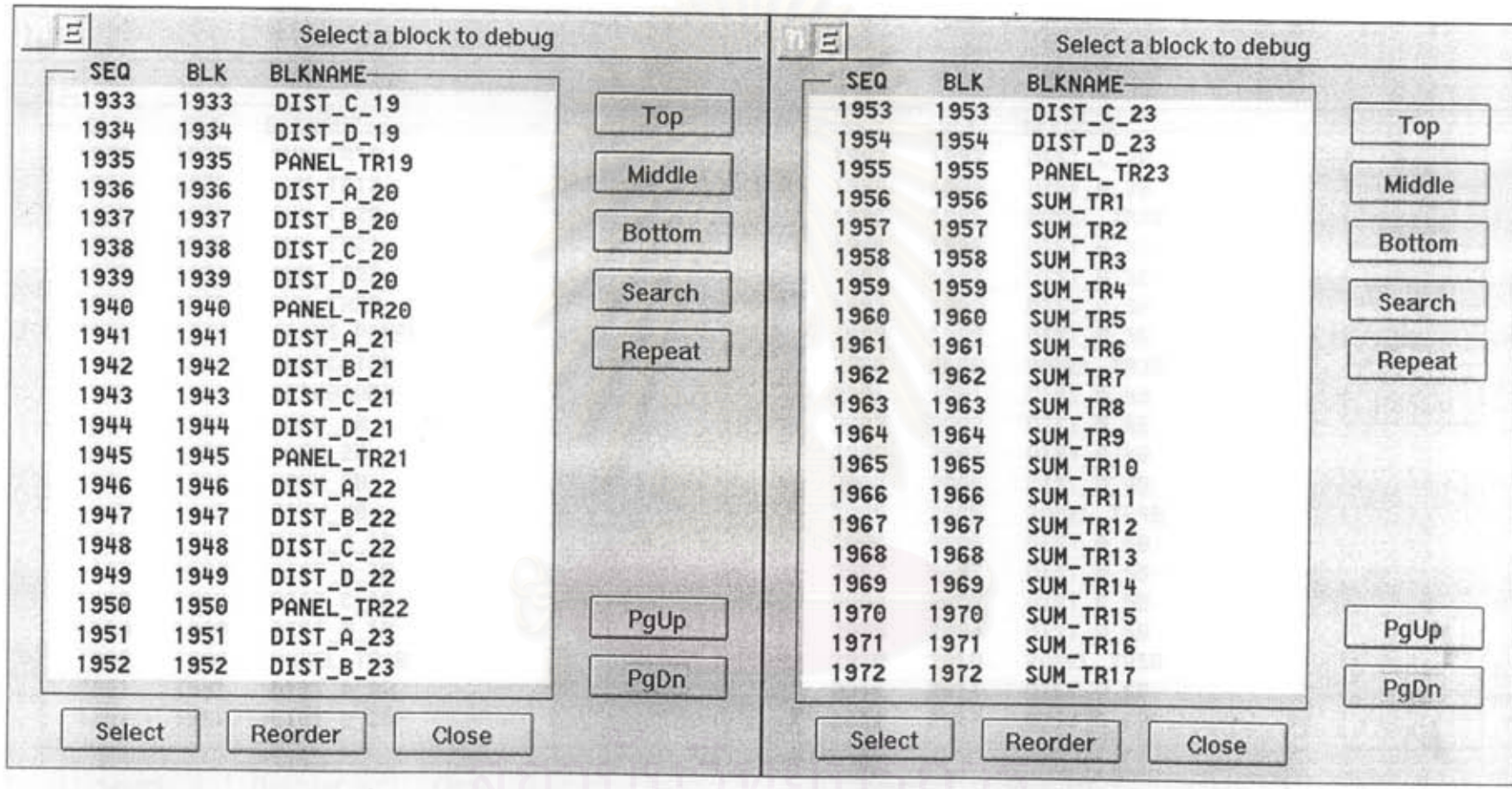
รูปที่ ค.1 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272 [4]



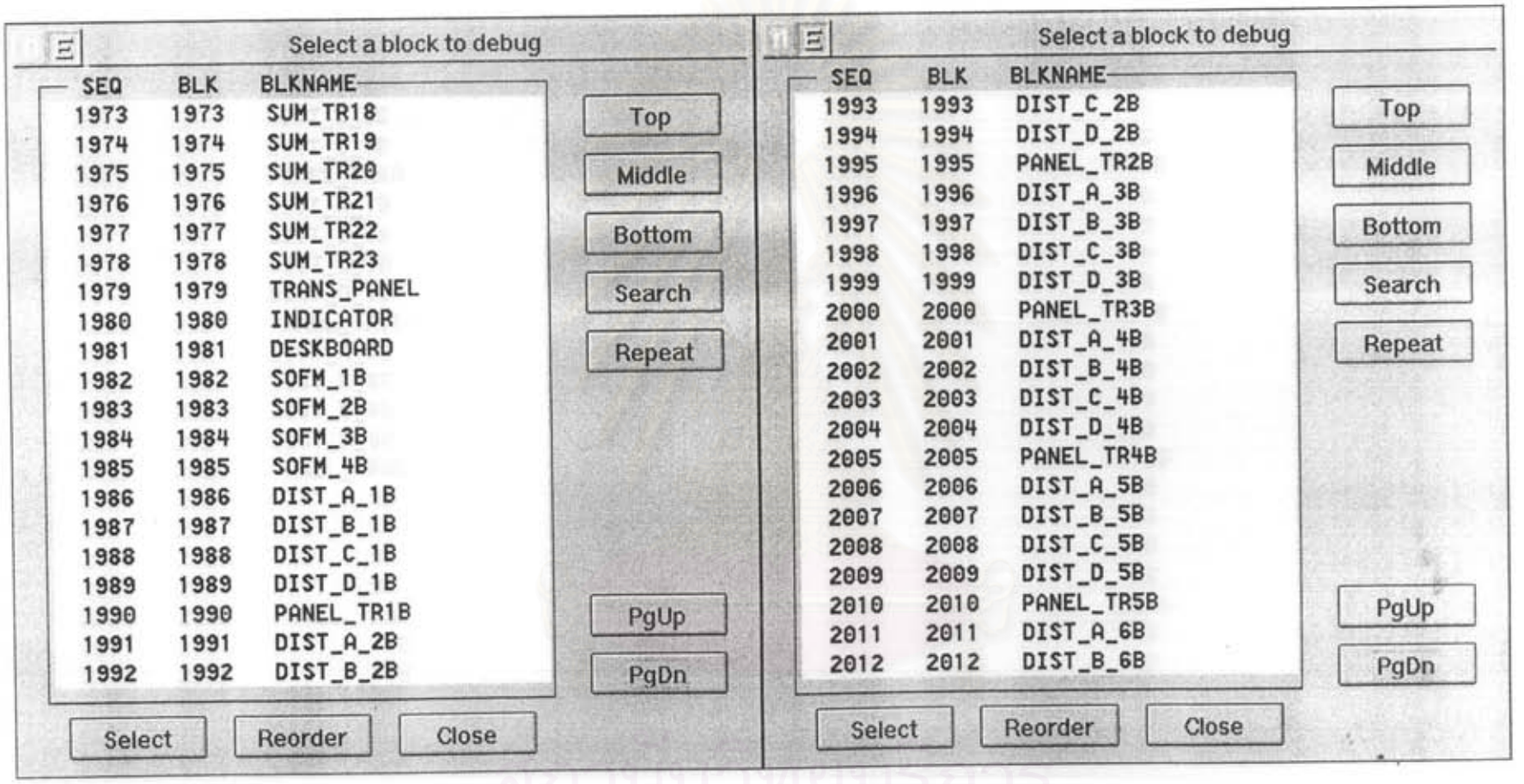
รูปที่ ค.2 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272



รูปที่ ค.3 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272

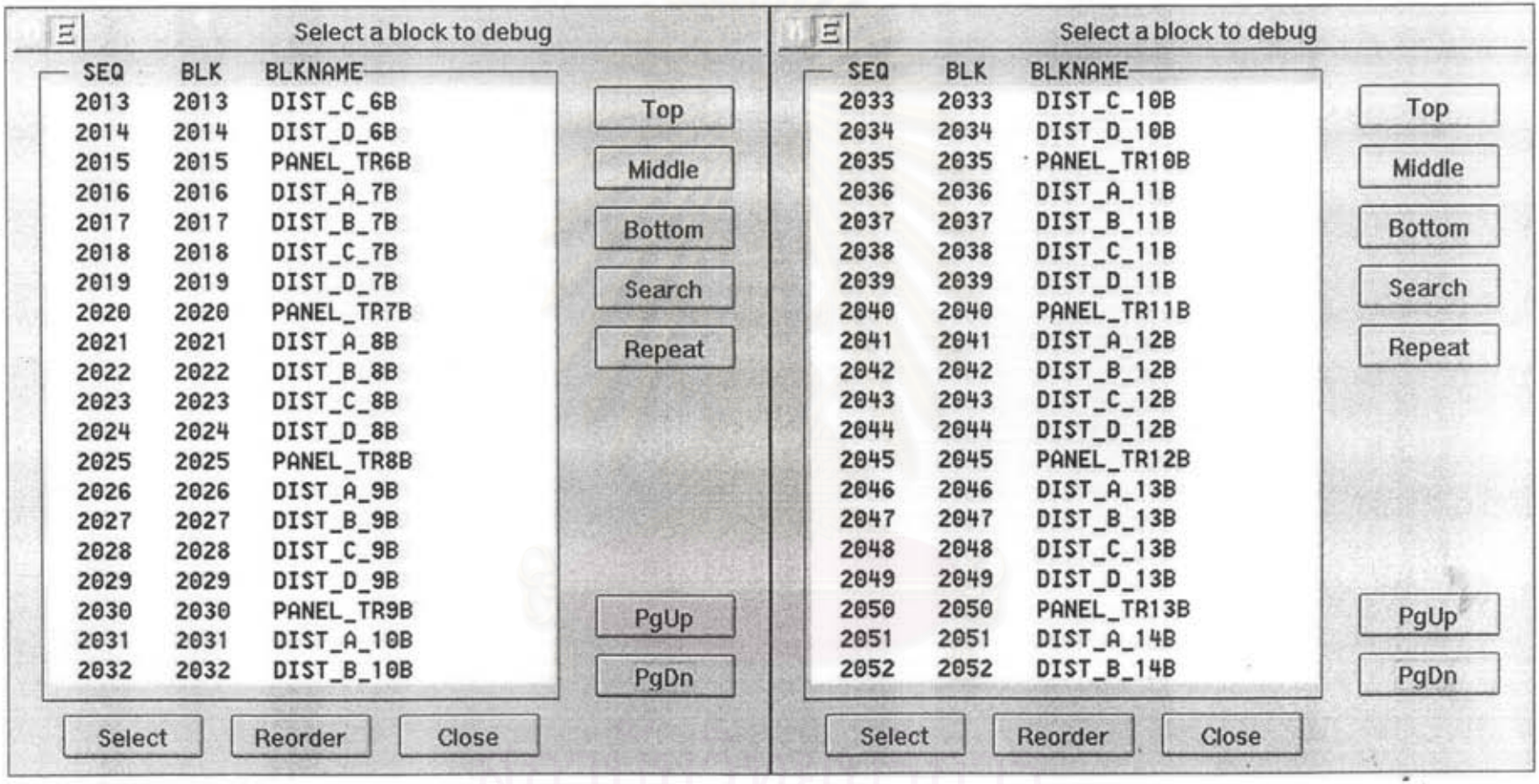


รูปที่ ค.4 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272

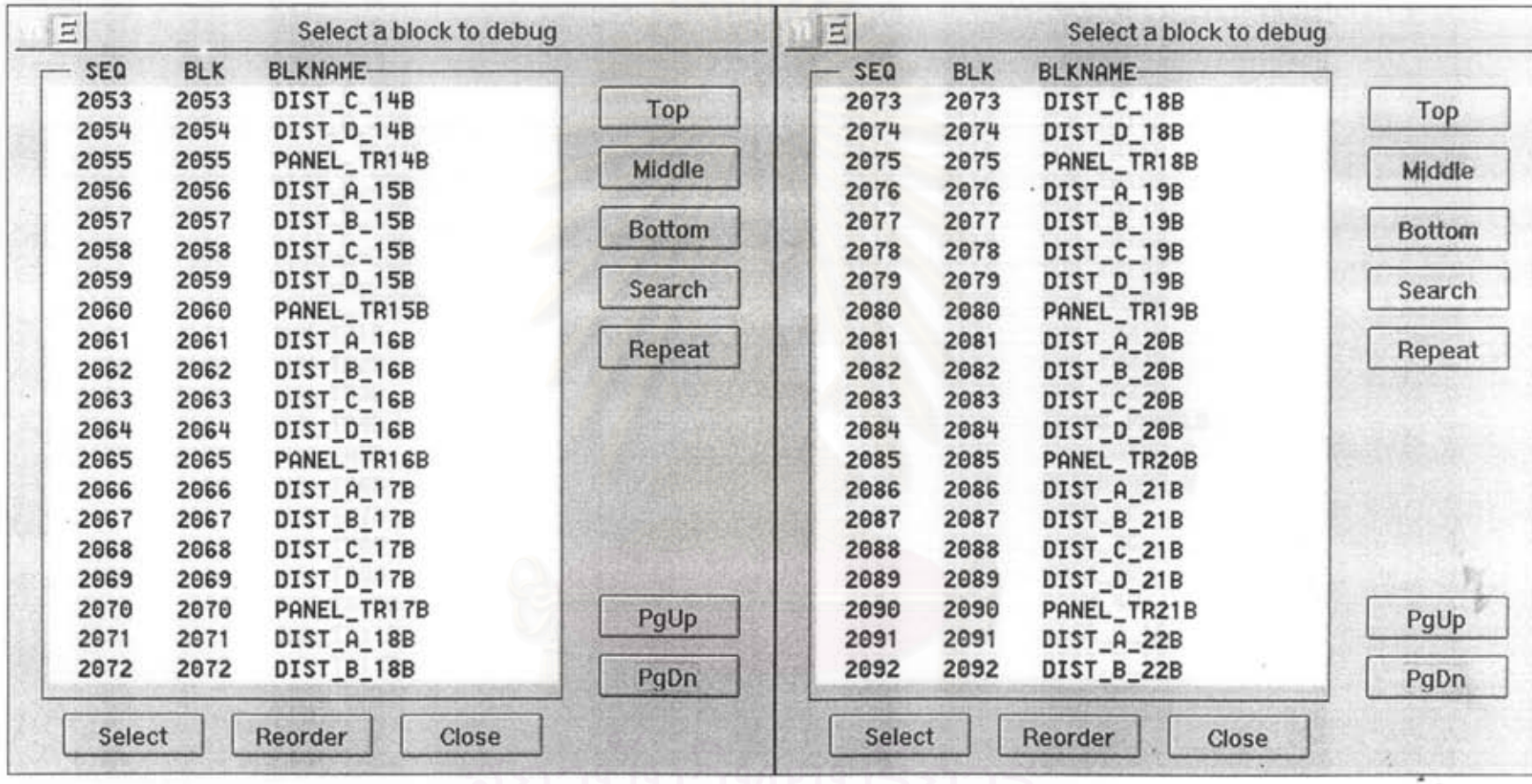


รูปที่ ค.5 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272

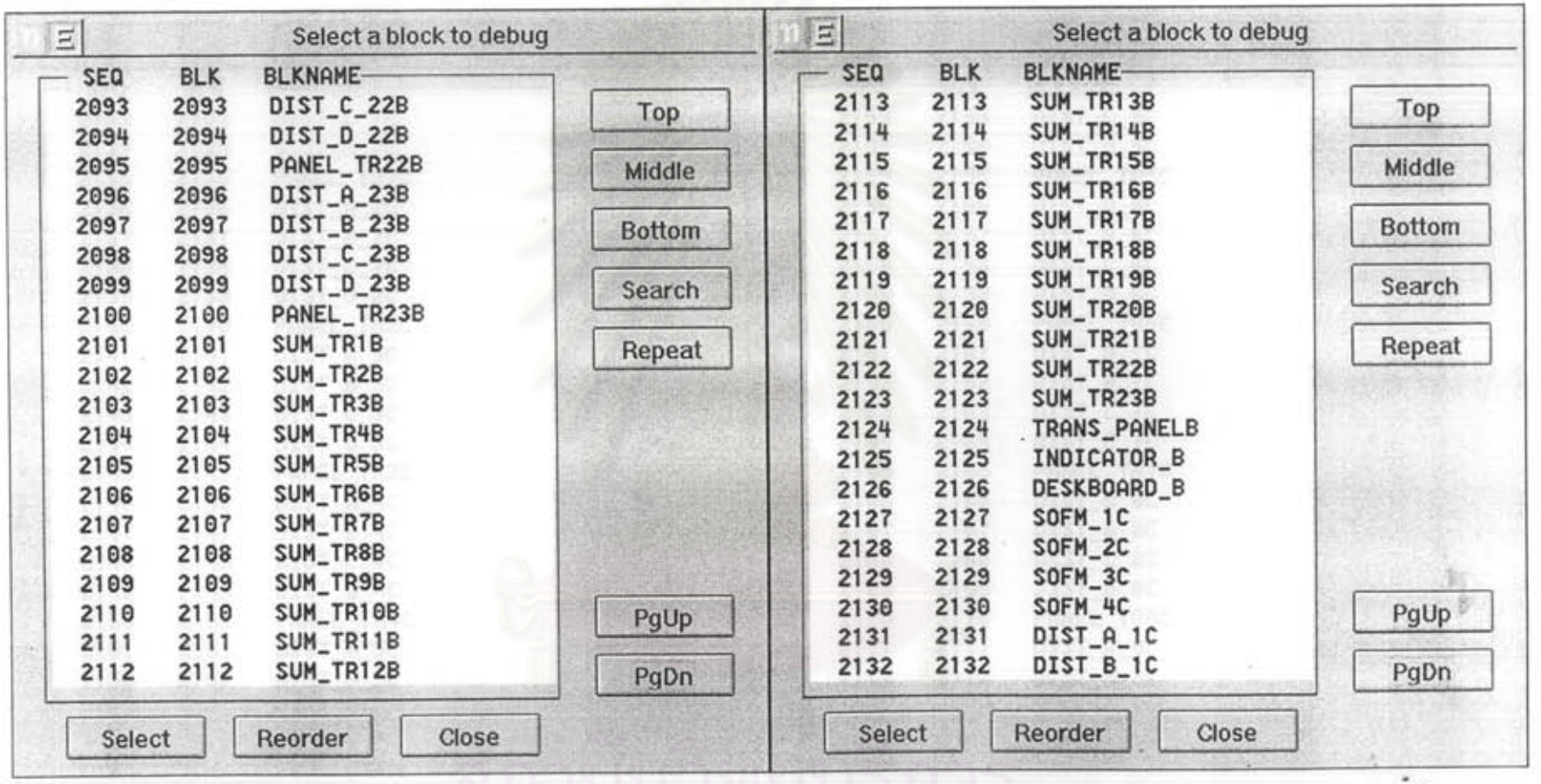
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



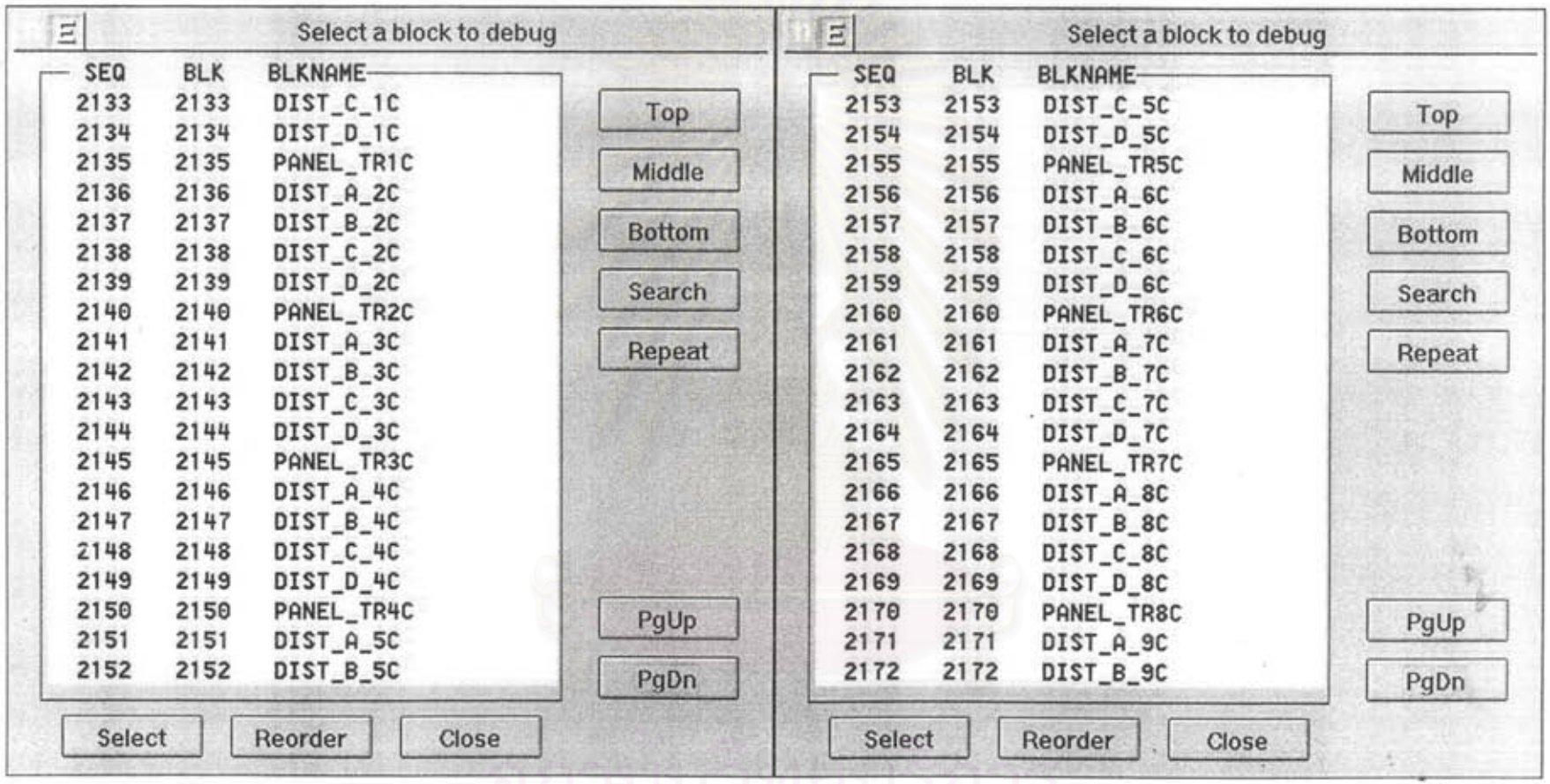
รูปที่ ค.6 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272



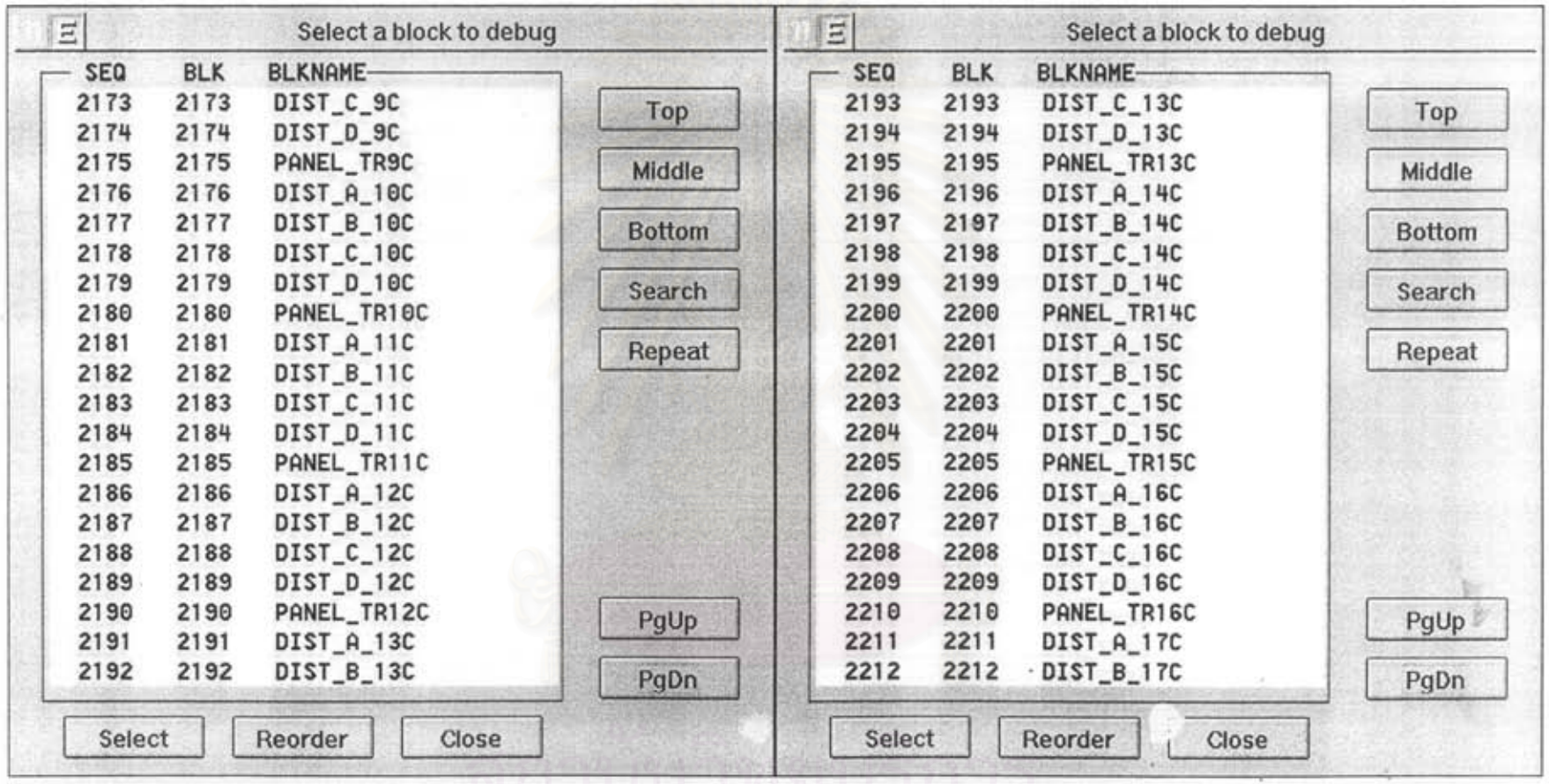
รูปที่ ค.7 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272



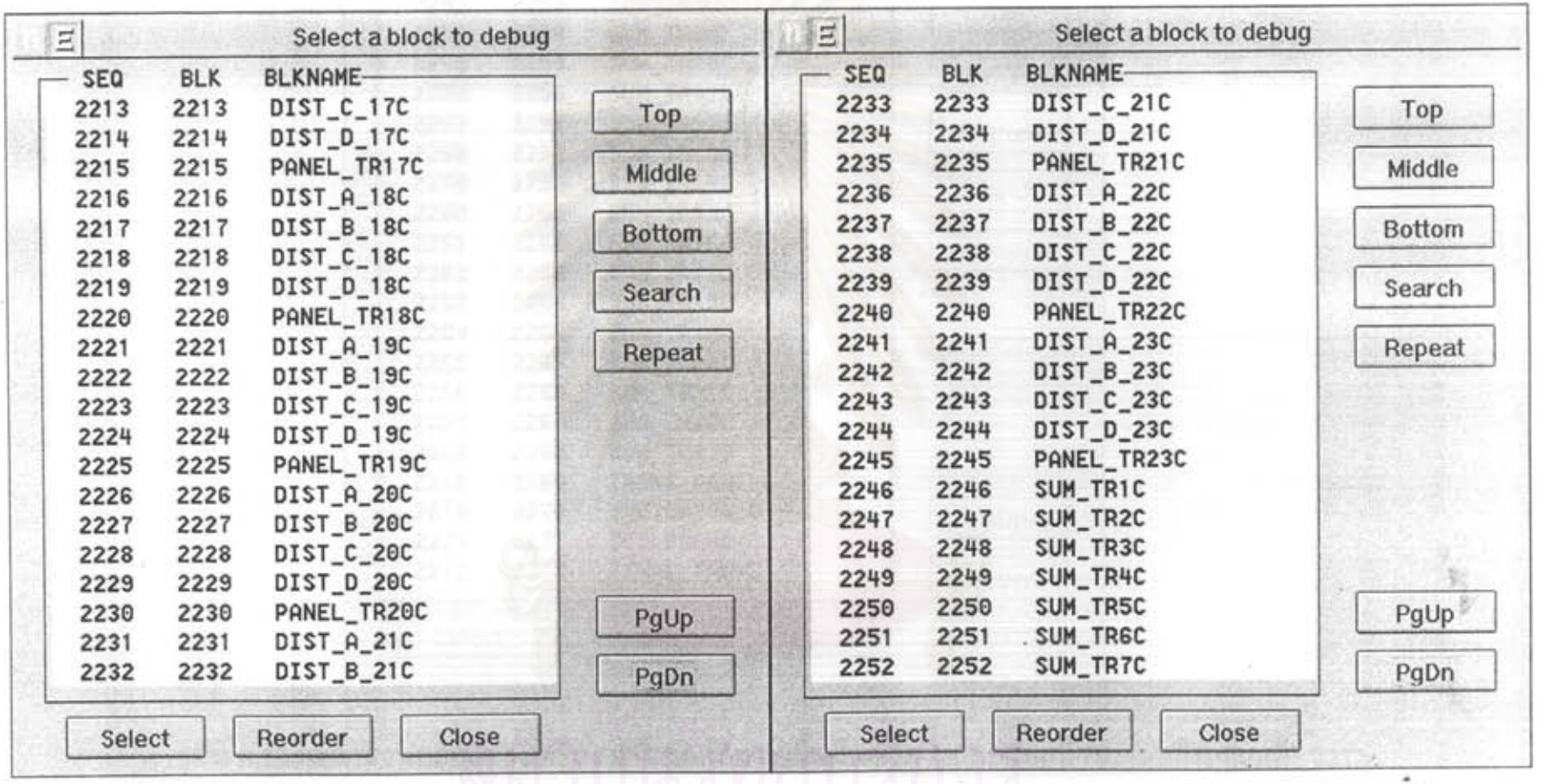
รูปที่ ค.8 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272



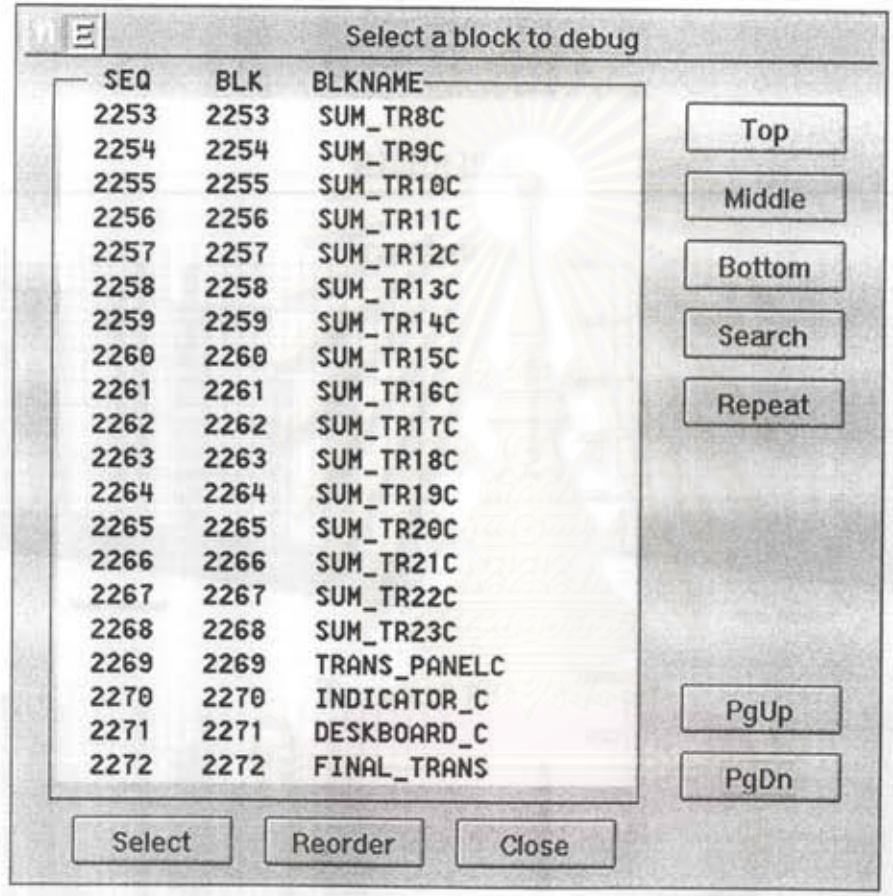
รูปที่ ค.9 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272



รูปที่ ค.10 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272

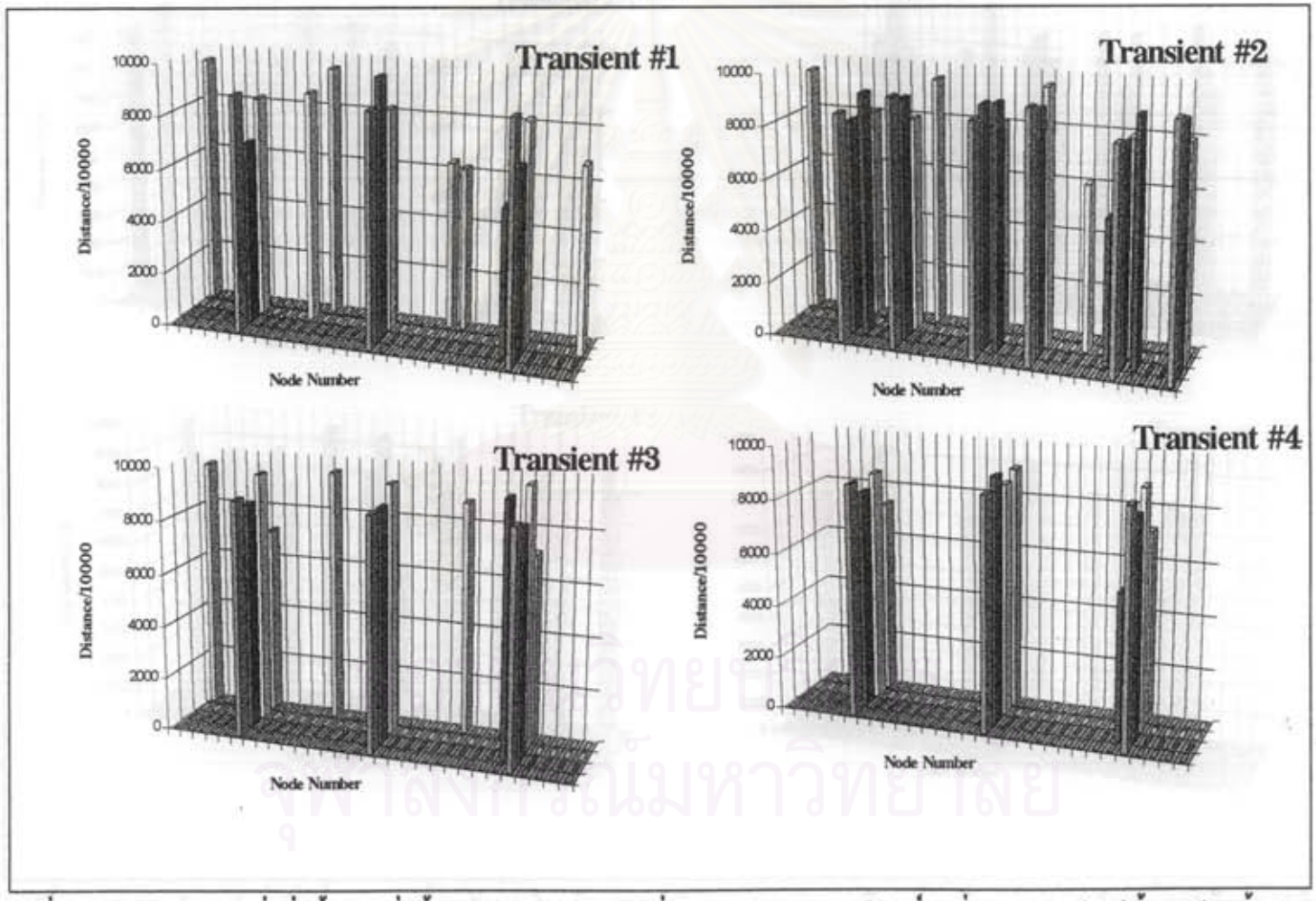


รูปที่ ค.11 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272



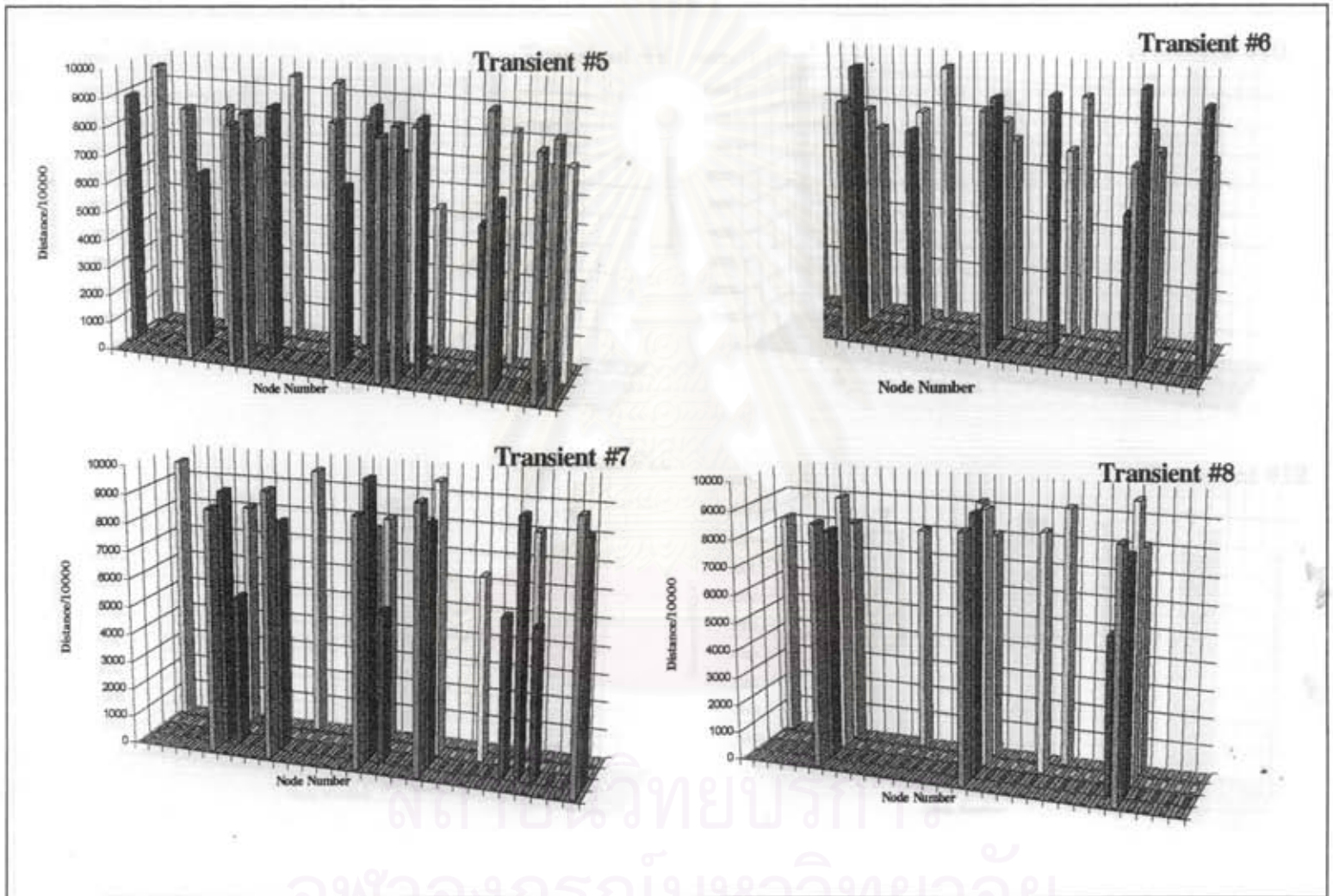
รูปที่ ค.12 แสดงส่วนบล็อกของเครือข่ายนิวรอลที่สร้างขึ้นโดยเริ่มตั้งแต่บล็อกที่ 1824จนถึงบล็อกที่ 2272

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

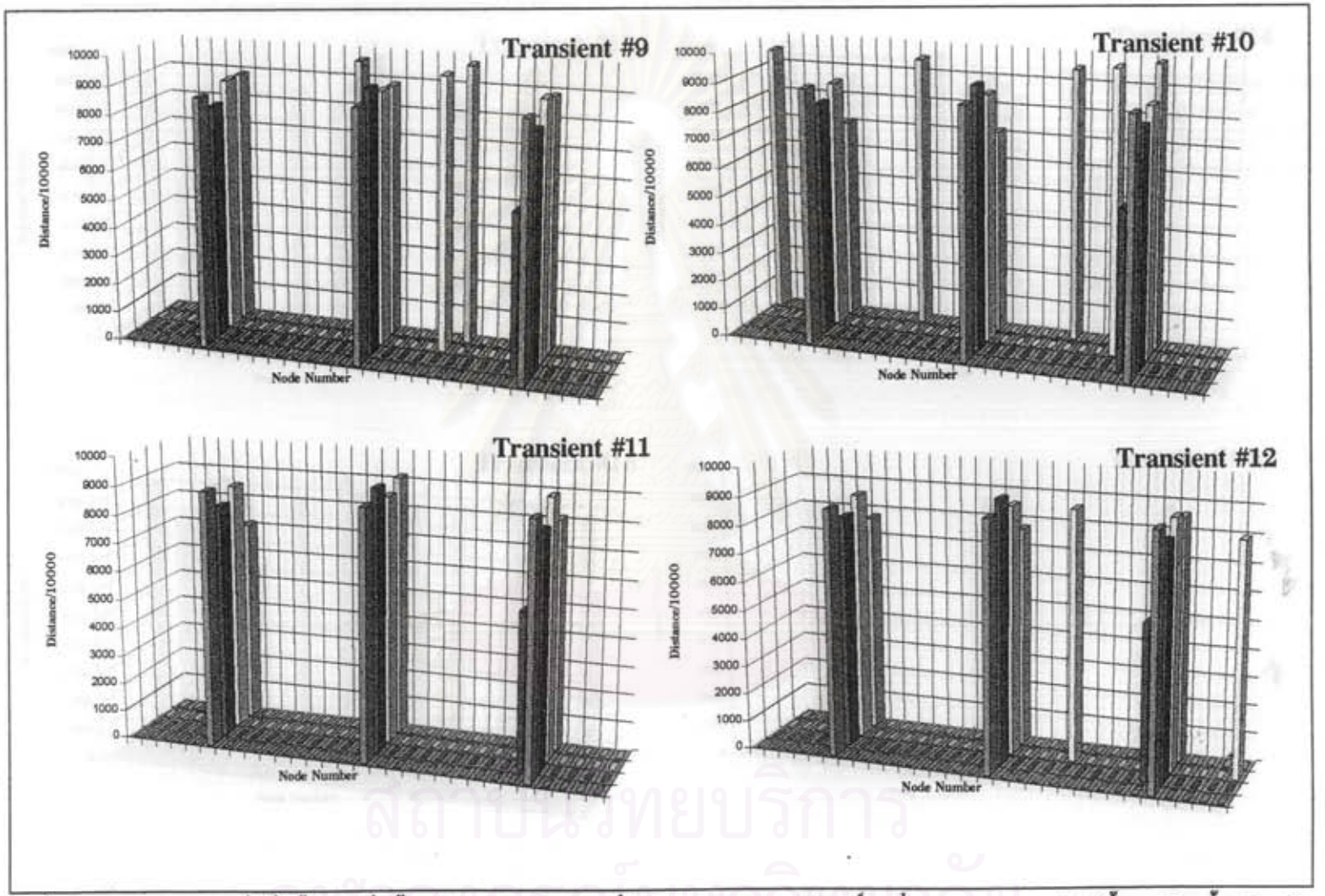


รูปที่ ง.1 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 1 , 2 , 3 ,4 โดยใช้เวลาเรียนรู้ 5 วินาที

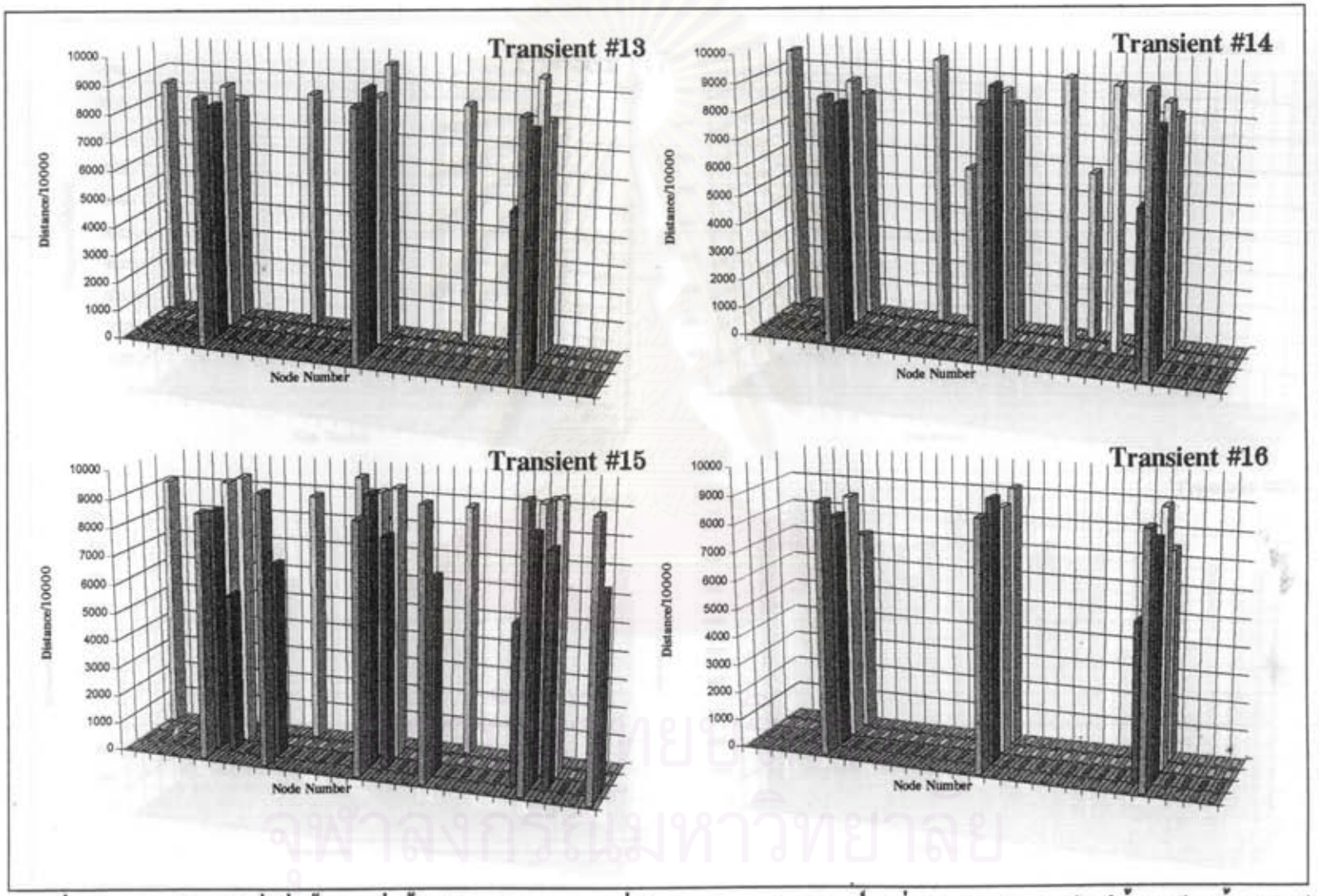
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



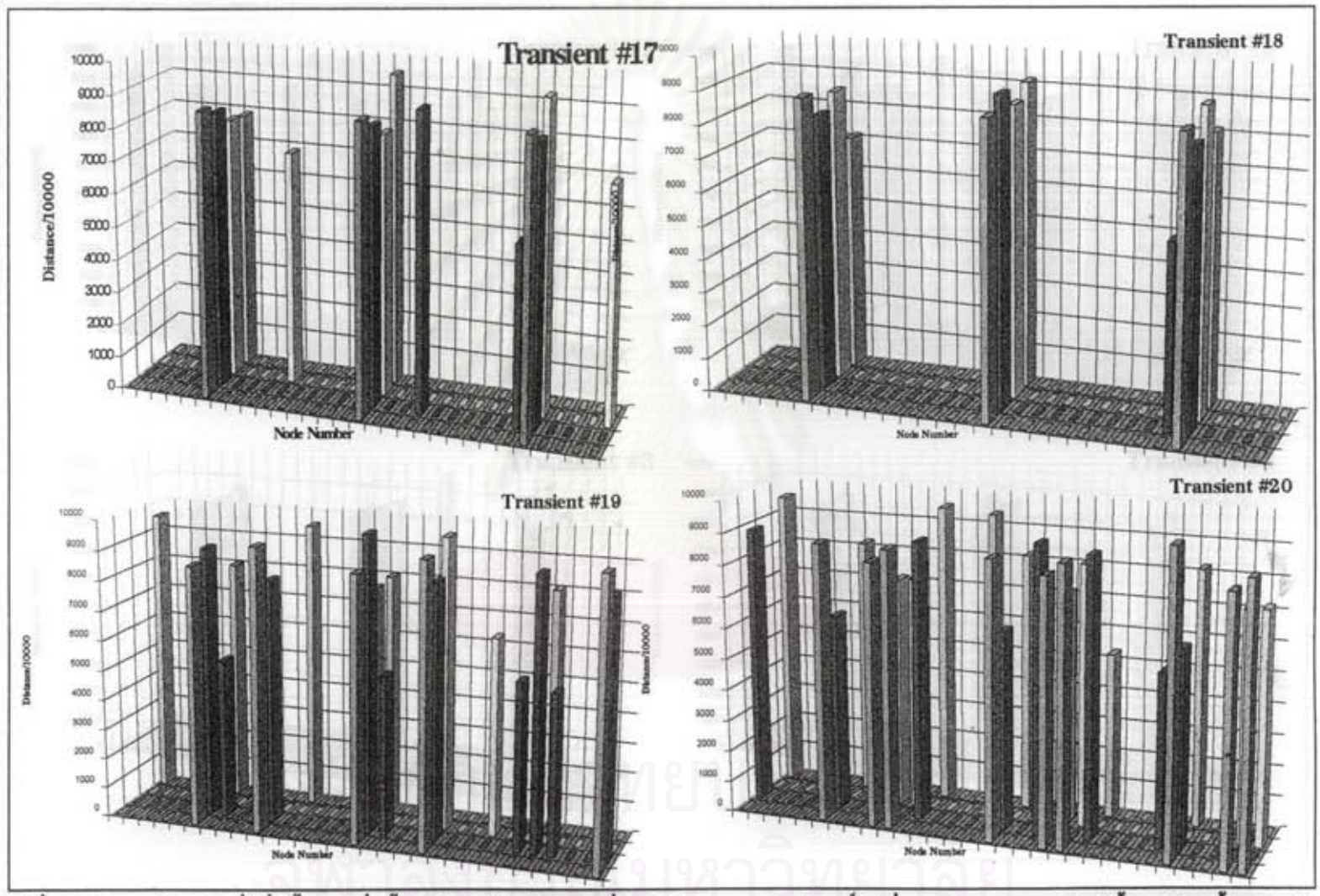
รูปที่ ง.2 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 5 , 6 , 7 , 8 โดยใช้เวลาเรียนรู้ 5 วินาที



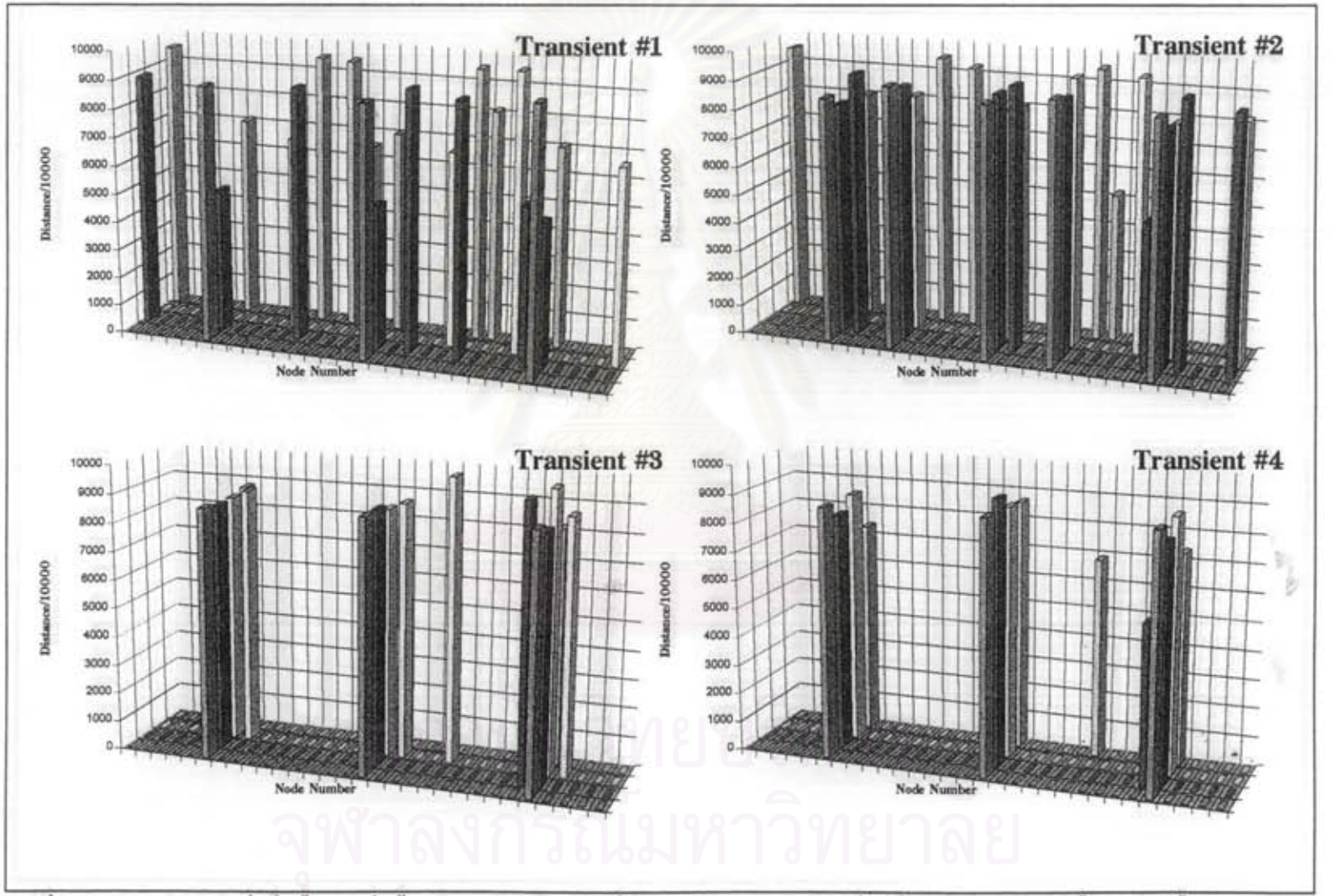
รูปที่ ง.3 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 9 , 10 , 11 , 12 โดยใช้เวลาเรียนรู้ 5 วินาที



รูปที่ ง.4 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 13 , 14 , 15 , 16 โดยใช้เวลาเรียนรู้ 5 วินาที

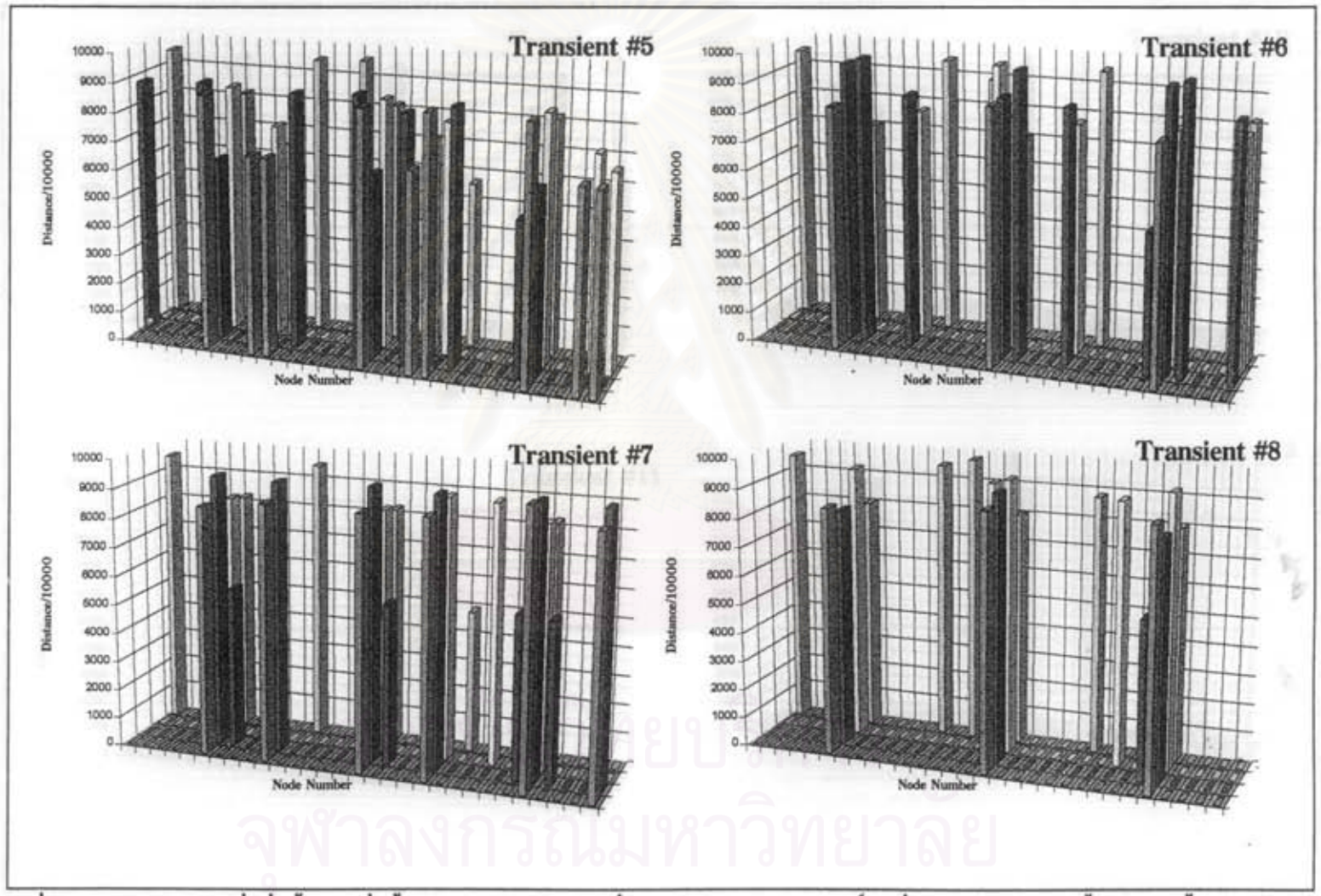


รูปที่ ง.5 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 17 , 18 , 19 , 20 โดยใช้เวลาเรียนรู้ 5 วินาที



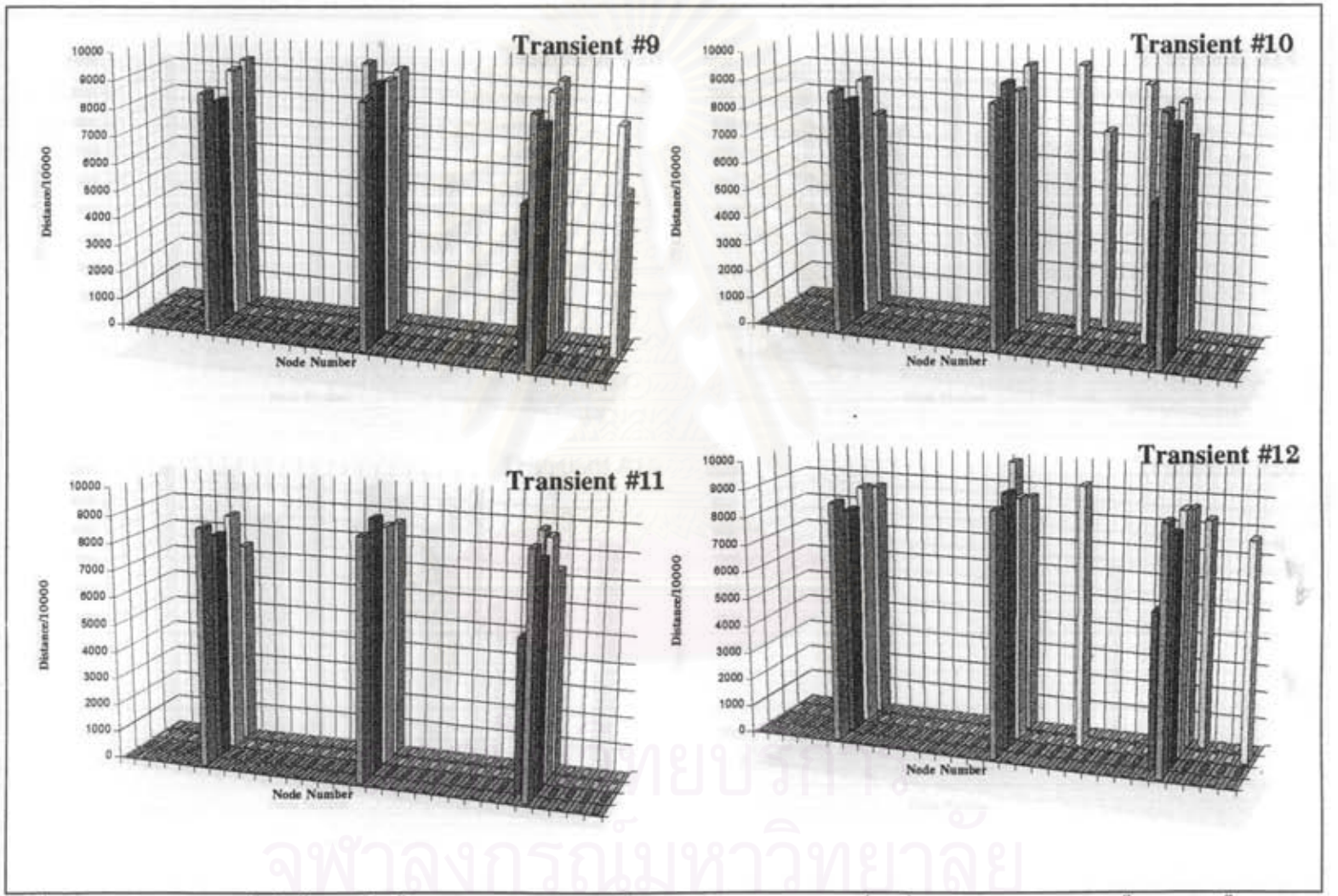
รูปที่ ง.6 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 1 , 2 , 3 , 4 โดยใช้เวลาเรียนรู้ 10 วินาที

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



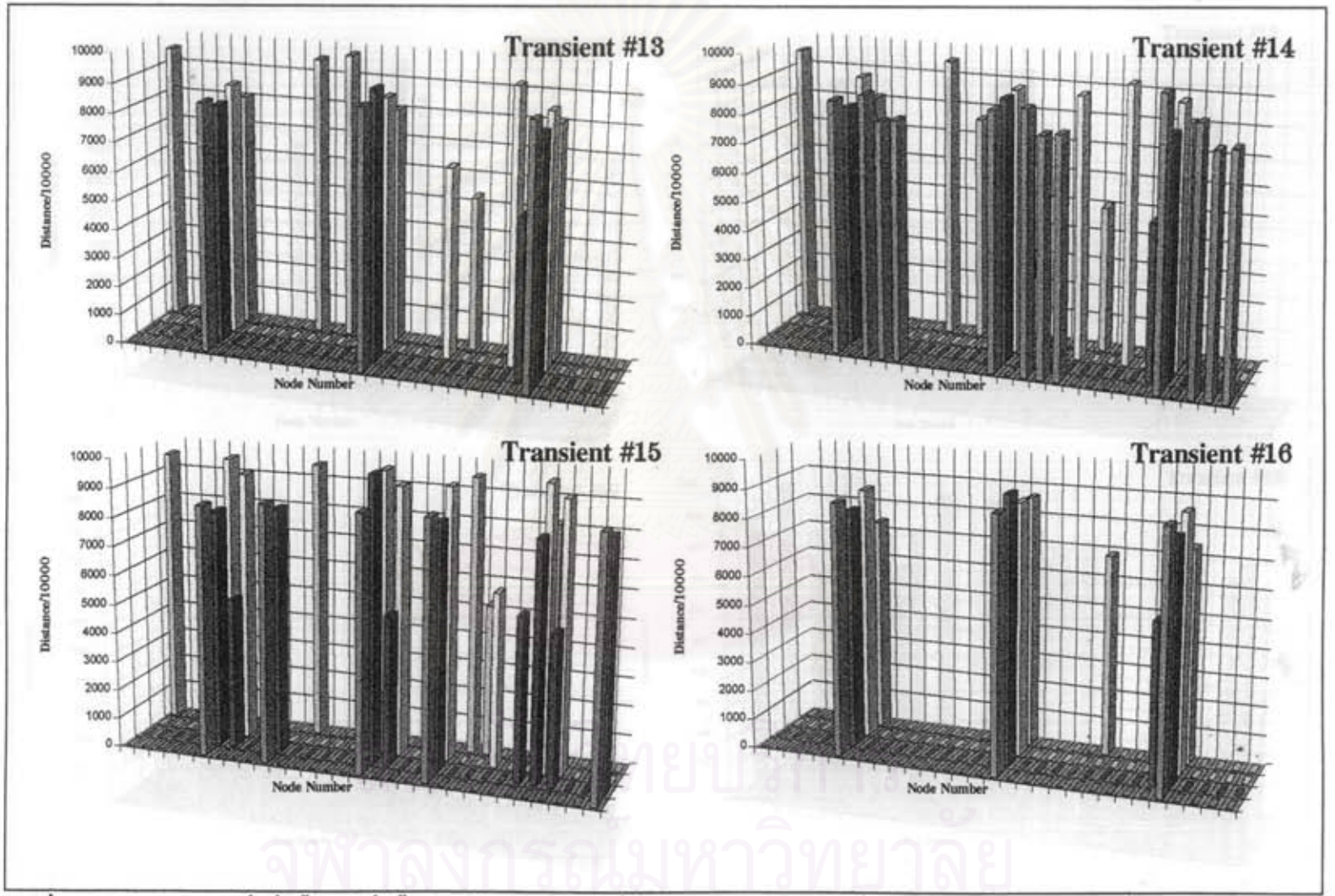
รูปที่ ง.7 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 5 , 6 , 7 , 8 โดยใช้เวลาเรียนรู้ 10 วินาที

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

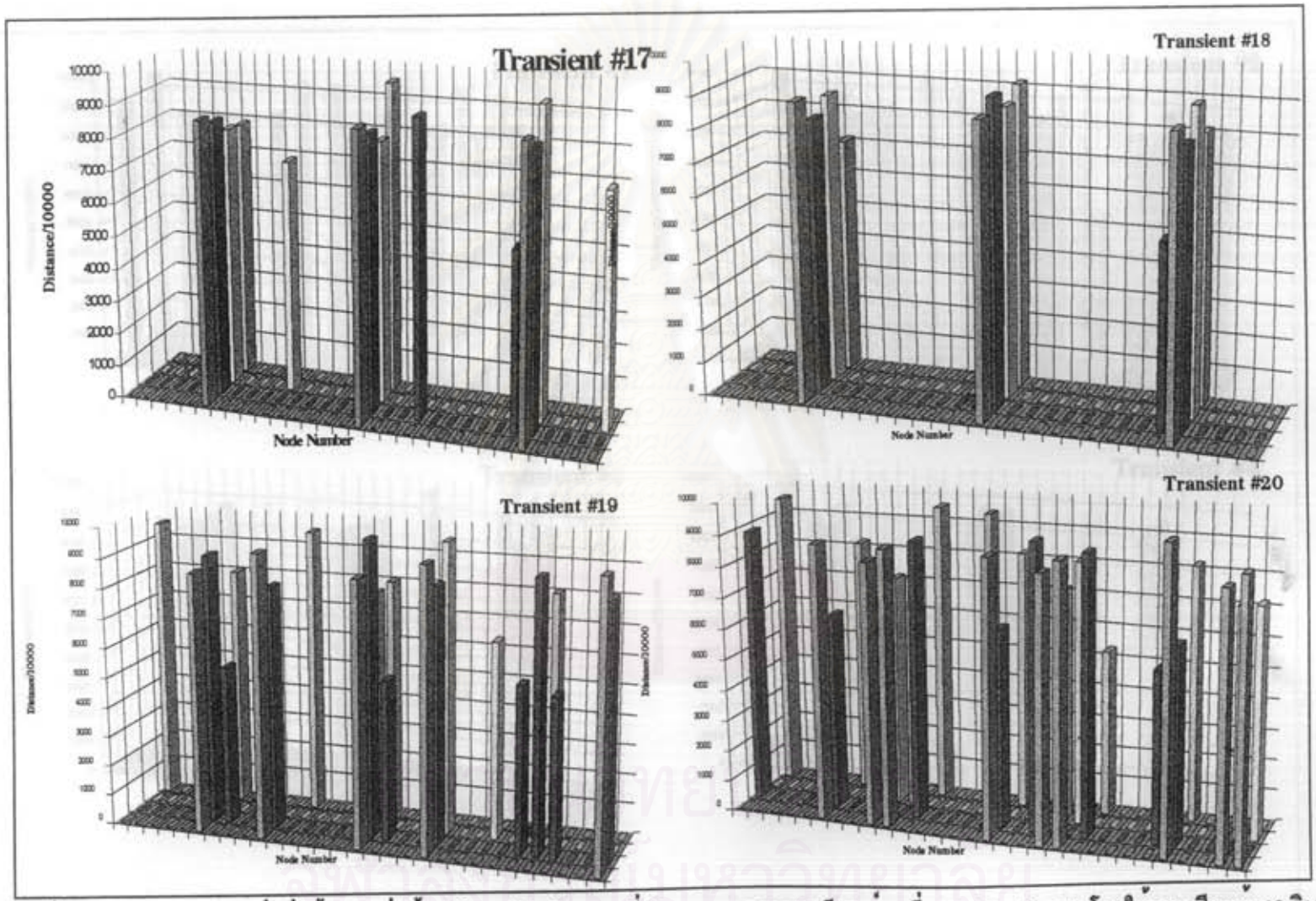


รูปที่ ง.8 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 9 , 10 , 11 , 12 โดยใช้เวลาเรียนรู้ 10 วินาที

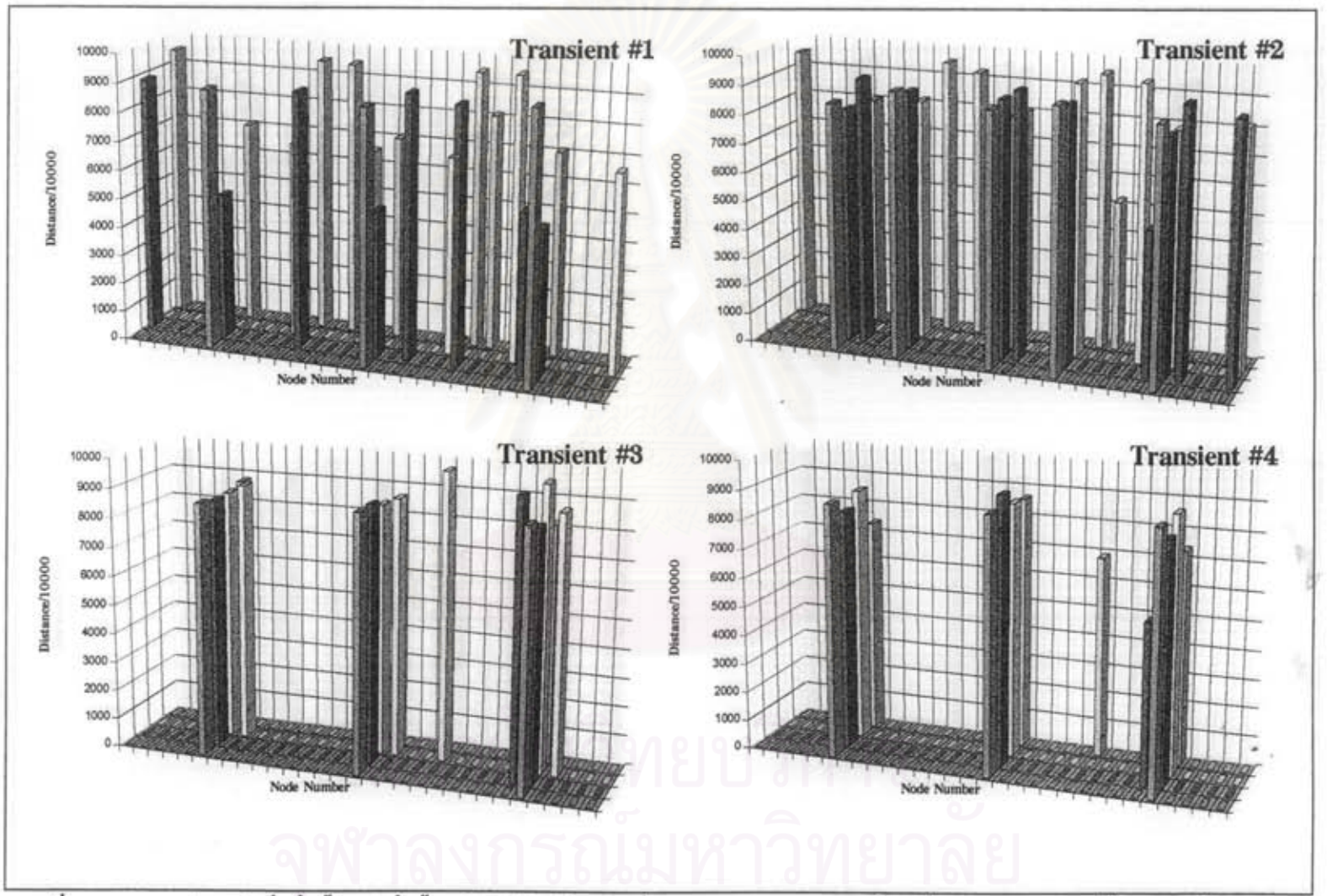
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.9 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 13 , 14 , 15 , 16 โดยใช้เวลาเรียนรู้ 10 วินาที

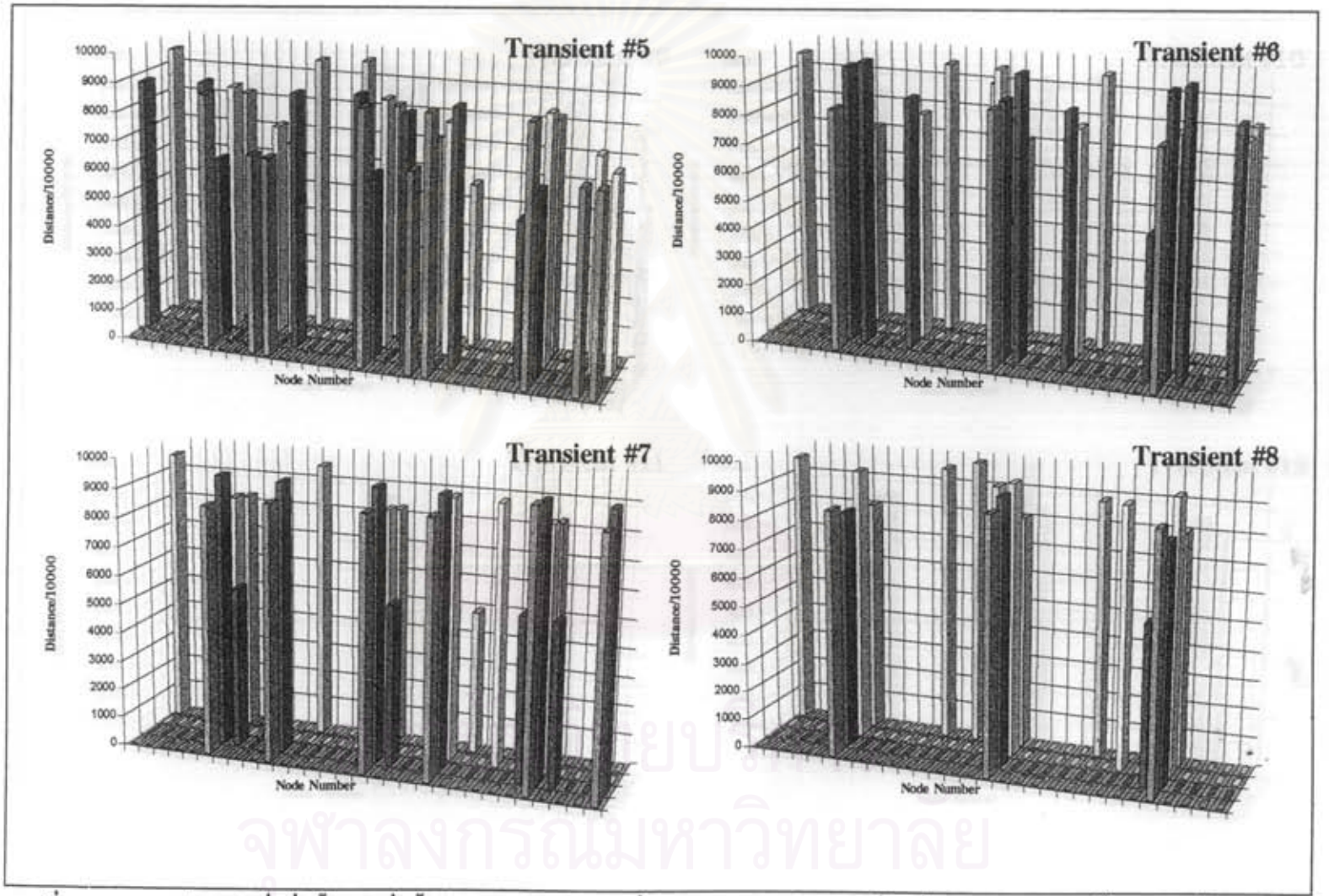


รูปที่ ง.10 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 17 , 18 , 19 , 20 โดยใช้เวลาเรียนรู้ 10 วินาที

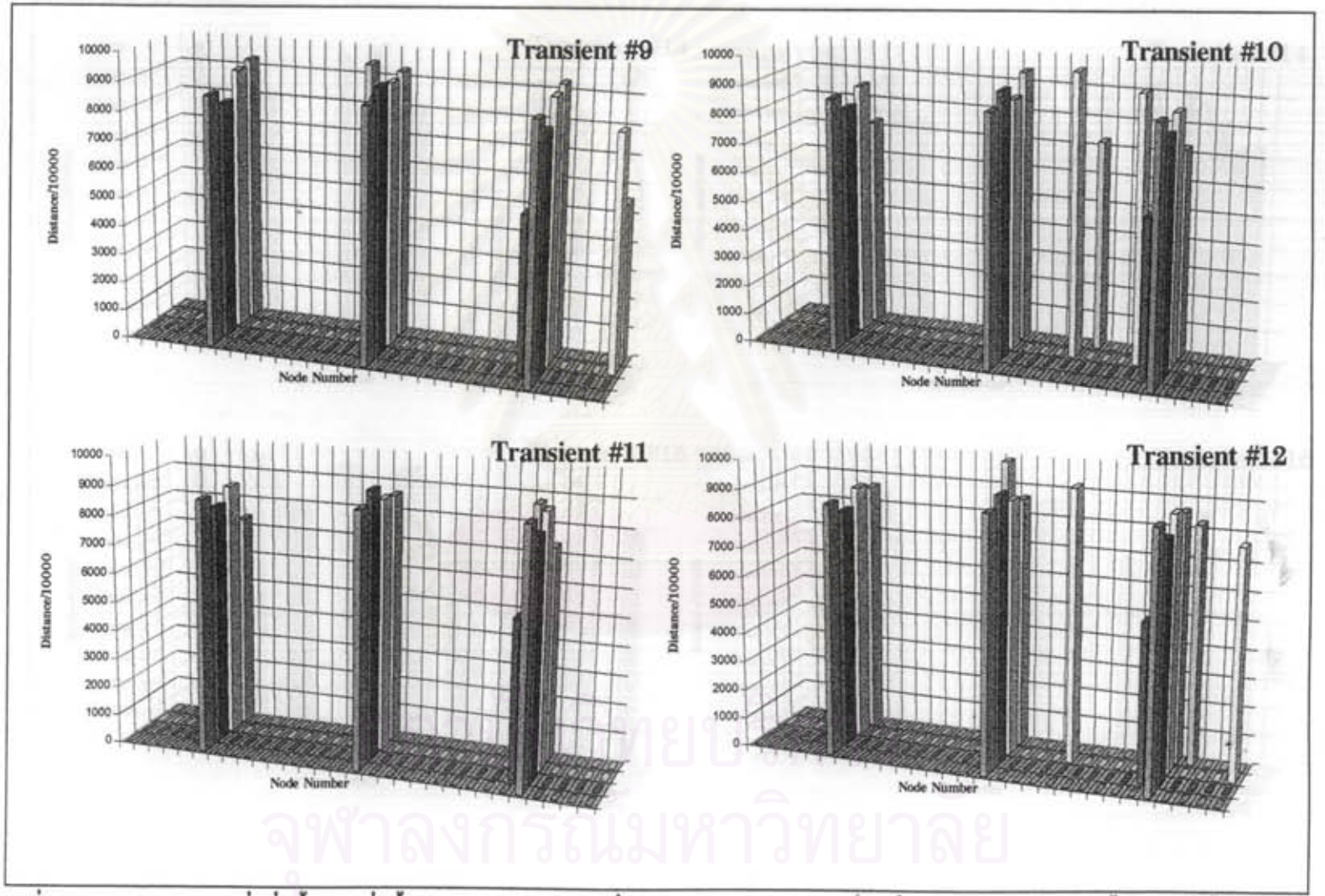


รูปที่ ง.11 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 1 , 2 , 3 , 4 โดยใช้เวลาเรียนรู้ 15 วินาที

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

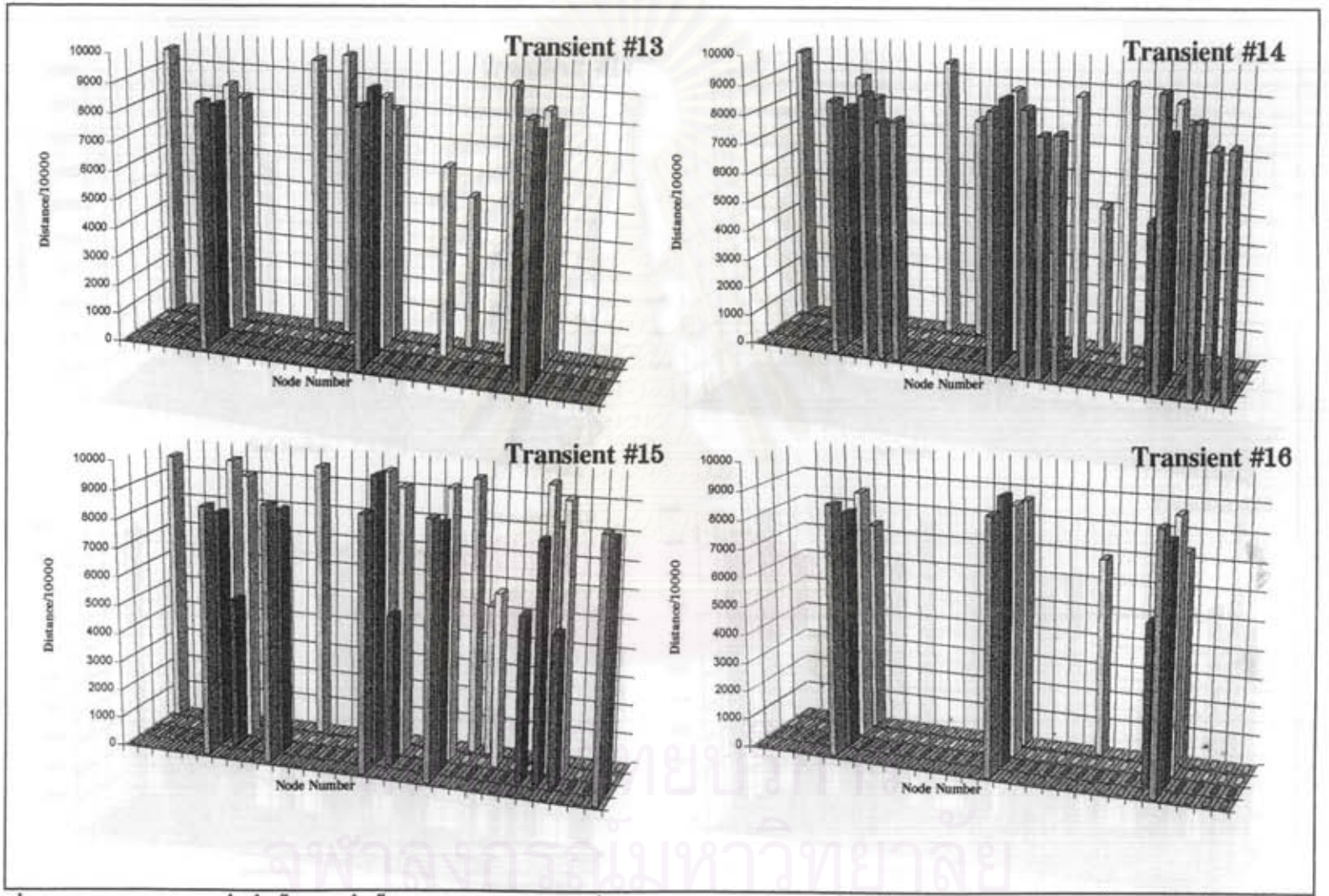


รูปที่ ง.12 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 5 , 6 , 7 , 8 โดยใช้เวลาเรียนรู้ 15 วินาที

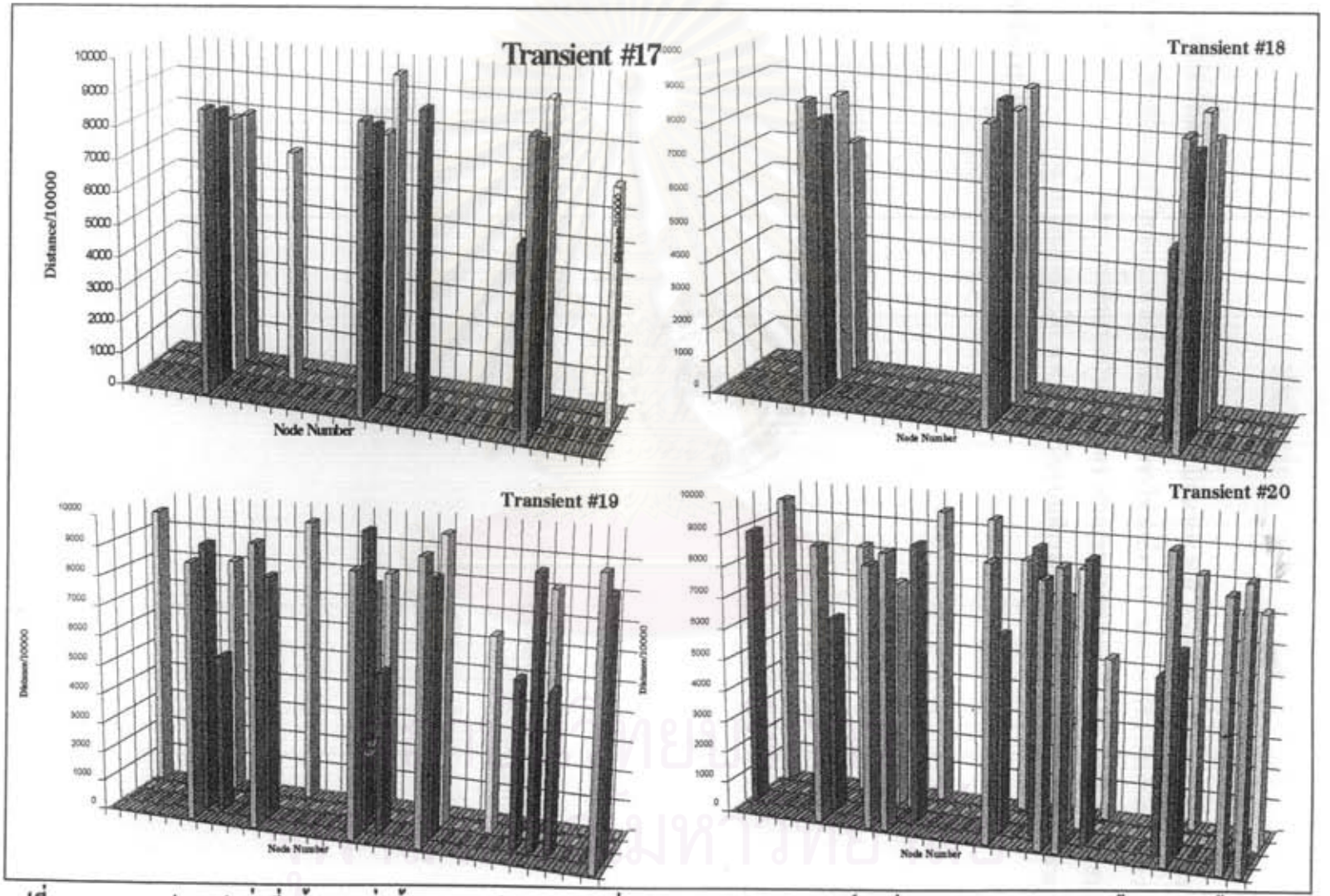


รูปที่ ง.13 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 9 , 10 , 11 , 12 โดยใช้เวลาเรียนรู้ 15 วินาที

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ง.14 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 13 , 14 , 15 , 16 โดยใช้เวลาเรียนรู้ 15 วินาที



รูปที่ ง.15 แสดงรูปกราฟแท่งที่สร้างจากค่าน้ำหนักของโนดใน FM ที่เกิดจากภาวะทรานเซียนต์ชุดที่ 17 , 18 , 19 , 20 โดยใช้เวลาเรียนรู้ 15 วินาที

ประวัติผู้เขียน

นาย พิพัฒน์ ธนาภรณ์ชินพงษ์ เกิดเมื่อวันที่ 22 มิถุนายน พ.ศ. 2508 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตร์บัณฑิต สาขาไฟฟ้าสื่อสาร จากมหาวิทยาลัยเกษตรศาสตร์ ในปีการศึกษา 2531 และ เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต สาขานิวเคลียร์เทคโนโลยี จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2538 ปัจจุบันดำรงตำแหน่ง วิศวกรระดับ 6 ประจำฝ่ายระบบสื่อสาร การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย