หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. B.G. Watters,R.B. Coleman,G.L. Duckworth and E.F. Berkman " A Perspective On Active Machinery Isolation " IEEE Proceedings of the 27^{th} Conference on Decision and Control Austin, 1988
2. Colin F Ross " Active Isolation of Ship Machinery Platforms " IEEE Proceedings of the 27^{th} Conference on Decision and Control Austin, 1988
3. Z. Jason Geng and Leonard S. Haynes " Six Degree-of-Freedom Active Vibration Control Using the Stewart Platforms " IEEE, VOL.2, NO.1, MARCH 1994
4. Eric H. Anderson and Jonathan P. How " Active Vibration Isolation Using Adaptive Feedforward Control " IEEE, Proceedings of the American Control Conference, 1997
5. JK Baker " Vibration Isolation " Oxford University Press

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บรรณานุกรม

1. William T. Thomson " Vibration Theory and Applications " 4th edition Prentice-Hall. Inc. 1993
2. Austin H. Church " Mechanical Vibrations " 2nd edition John Wiley and Sons. Inc. 1963
3. J.P. Den Hartog " Mechanical Vibrations " 4th edition McGraw-Hill Book Company. Inc. 1956
4. Robert K. Vierck " Vibration Analysis " 5th edition International Textbook Company.1967
5. รศ.ดร.วิบูลย์ แสงวีระพันธุ์ศิริ " การควบคุมระบบพลศาสตร์ " สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย พ.ศ. 2538
6. Curtis D. Johnson " Process Control Instrumentation Technology " 2nd edition John Wiley & Sons. Inc. 1982
7. Kutsuhiko Ogata " Modern Control Engineering " 1st edition Prentice-Hall International. Inc.1990
8. William F. Riley and Loren Zachary " Introduction to Mechanic of Materials " 1st edition John Wiley & Sons. Inc. 1989
9. Neil Storey " Electronics A Systems Approach " Addison-Wesley Publishing Company 1992
10. ปานเพชร ชินินทร และ ขวัญชัย สินทิพย์สมบูรณ์ " นิวแมติก อุตสาหกรรม " บริษัท เอช.เอ็น. กรุ๊ป จำกัด พ.ศ. 2539
11. วันชัย คุณากรวงศ์,บัณฑิต บัวบูชา,รัฐวุฒิ ประทุมราช และ รุ่งแสง เครือไวศยวรรณ " การคำนวณวงจรทรานซิสเตอร์ " หจก.สำนักพิมพ์ฟิสิกส์เซ็นเตอร์
12. J.L. Meriam and L.G. Kraige " Engineering Mechanics Volume Two Dynamics " 3rd edition John Wiley & Sons. Inc. 1993

ภาคผนวก

ภาคผนวก ก

โปรแกรมที่ใช้ในการควบคุมระบบ

โปรแกรมนี้ใช้สำหรับบอร์ดควบคุม (Controller Board) CP-S8252 V2.0 ซึ่งอยู่ในกลุ่ม MCS-51 มีข้อมูลดังนี้

```
ORG     0000H

;-----------------------------------------------------------
;       Define keypad
;-----------------------------------------------------------

KPAD_ROW        BIT     P2.0
KPAD_ROW1       BIT     P2.1
KPAD_ROW2       BIT     P2.2
KPAD_ROW3       BIT     P2.3
KPAD_COL2       BIT     P2.4
KPAD_COL1       BIT     P2.5
KPAD_COL0       BIT     P2.6

KPAD_DATA       EQU     030H    ; For keep keypad data
DATA1           EQU     031H    ; For keep input data
DATA2           EQU     032H
DATA3           EQU     033H
DATA4           EQU     034H
DATA5           EQU     035H

;-----------------------------------------------------------
;                       Define variable
;-----------------------------------------------------------
```

```
FWD_SIG             BIT   P1.0
BWD_SIG             BIT   P1.1
HOME_SIGNAL         BIT   P0.0
PULSE1              BIT   P3.4
PULSE2              BIT   P2.7

I         EQU   036H
J         EQU   037H

OVER        EQU   038H

MUL1H               EQU   039H
MUL1L               EQU   03AH
MUL2                EQU   03BH
MUL_RESULTL         EQU   03CH
MUL_RESULTH         EQU   03DH
MUL_RESULTHH        EQU   03EH

COMPARE_V           EQU   03FH
COMPARE1H           EQU   040H
COMPARE1L           EQU   041H
COMPARE2H           EQU   042H
COMPARE2L           EQU   043H

HEXH        EQU   044H
HEXL        EQU   045H

KEY_DATAH           EQU   046H
KEY_DATAL           EQU   047H
```

```
POSH            EQU   048H
POSL            EQU   049H
POS2H           EQU   04AH
POS2L           EQU   04BH
ROTATE          EQU   04CH
LAST_ROTATE     EQU   04DH
POSSTOPH        EQU   04EH
POSSTOPL        EQU   04FH
HZH             EQU   050H
HZL             EQU   051H
POSDH           EQU   054H
POSDL           EQU   055H
CHECK_HZ        EQU   056H
OLD_HZ          EQU   057H
LIMIT           EQU   058H


;------------------------------------------------------------
;                     Define LCD pins
;------------------------------------------------------------

CS_LCD   EQU   P1.3     ; E  LCD (PIN INT1)
RS_LCD   EQU   P1.2     ; RS LCD (Pin T0)

                AJMP  MAIN

                ORG   00BH                       ; TF0 vector interrupt
                SETB  OVER
                RETI

MAIN:            MOV   SP,#256-32   ; Initial Stack 32 Byte
```

```
            MOV   P3,#00111111B
            MOV   P2,#11111111B
            MOV   P0,#00000000B
            CLR   FWD_SIG
            CLR   BWD_SIG

;-----------------------------------------------------------
;     Initialize LCD
;-----------------------------------------------------------
    SETB   CS_LCD      ; initial LCD
    LCALL  DELAY       ; initial delay
    LCALL  INIT_LCD    ; initial LCD

;-----------------------------------------------------------
;     Main loop
;-----------------------------------------------------------
            MOV   TMOD,#55H          ; counter0 16 bit , counter1 16 bit
            SETB  EA
            MOV   POSH,#003H
            MOV   POSL,#08CH

SET_HOME:
            JB    HOME_SIGNAL,STOP_TO_HOME
            SETB  BWD_SIG
            CLR   FWD_SIG
            LJMP  SET_HOME
STOP_TO_HOME:
            MOV   P0,#11111111B
```

```
                CLR    FWD_SIG
                CLR    BWD_SIG

MAIN_LOOP:

                MOV    OLD_HZ,#0
                LJMP   INITIAL_HZ

RECORD_POSITION:

                MOV    A,#040H
                LCALL  GOTO_LCD
                MOV    LIMIT,#16
                LCALL  WRLINE_LCD
                MOV    POSSTOPH,HEXH
                MOV    POSSTOPL,HEXL
                MOV    A,#000H
                LCALL  GOTO_LCD
                MOV    DPTR,#MOVE
                MOV    LIMIT,#16
                LCALL  WRLINE_LCD
                LCALL  DELAY_1s
CHECK:
                CLR    C
                MOV    A,POSSTOPH
                SUBB   A,POSH
                JNC    COND1
                LCALL  RUN_BACKWARD
                LJMP   MAIN_LOOP
COND1:          JZ     MAIN_LOOP
                LCALL  RUN_FORWARD
```

```
COND2:          LJMP   MAIN_LOOP

;-----------------------------------------------------------------------
;               DETECT SPEED
;-----------------------------------------------------------------------
INITIAL_HZ:
                MOV    CHECK_HZ,#3
                MOV    OLD_HZ,HZL
DETECT_HZ:
                MOV    TH1,#000H
                MOV    TL1,#000H
                SETB   ET1
                SETB   TR1
                LCALL  DELAY_1s
                MOV    HEXH,TH1
                MOV    HEXL,TL1
                MOV    HZL,HEXL
                MOV    A,HEXL
                MOV    B,#60
                MUL    AB
                MOV    HEXL,A
                MOV    HEXH,B
                CLR    TR1
                CLR    ET1
                MOV    A,#000H
                LCALL  GOTO_LCD
                MOV    DPTR,#DETECT
                MOV    LIMIT,#7
                LCALL  WRLINE_LCD
                LCALL  SHOW_DEC                          ;show freq in rpm
                MOV    DPTR,#RPM
```

```
            MOV    LIMIT,#4
            LCALL  WRLINE_LCD

HZ_TO_POSITION:
            MOV    HEXL,HZL
            MOV    A,HEXL
            CJNE   A,OLD_HZ,INITIAL_HZ
            MOV    OLD_HZ,HEXL
            DJNZ   CHECK_HZ,DETECT_HZ

            CJNE   A,#13,NOT13
            MOV    HEXH,#020H
            MOV    HEXL,#030H
            MOV    DPTR,#POSITION13
            LJMP   RECORD_POSITION
NOT13:
            CJNE   A,#15,NOT15
            MOV    HEXH,#018H
            MOV    HEXL,#0E0H
            MOV    DPTR,#POSITION15
            LJMP   RECORD_POSITION
NOT15:
            CJNE   A,#17,NOT17
            MOV    HEXH,#014H
            MOV    HEXL,#000H
            MOV    DPTR,#POSITION17
            LJMP   RECORD_POSITION
NOT17:
            CJNE   A,#18,NOT18
            MOV    HEXH,#011H
            MOV    HEXL,#090H
```

```
        MOV   DPTR,#POSITION18
        LJMP  RECORD_POSITION
NOT18:
        CJNE  A,#20,NOT20
        MOV   HEXH,#00FH
        MOV   HEXL,#020H
        MOV   DPTR,#POSITION20
        LJMP  RECORD_POSITION
NOT20:
        CJNE  A,#22,NOT22
        MOV   HEXH,#00CH
        MOV   HEXL,#0B0H
        MOV   DPTR,#POSITION22
        LJMP  RECORD_POSITION
NOT22:
        CJNE  A,#23,NOT23
        MOV   HEXH,#00AH
        MOV   HEXL,#040H
        MOV   DPTR,#POSITION23
        LJMP  RECORD_POSITION
NOT23:
        CJNE  A,#25,NOT25
        MOV   HEXH,#007H
        MOV   HEXL,#0D0H
        MOV   DPTR,#POSITION25
        LJMP  RECORD_POSITION
NOT25:
        LJMP  INITIAL_HZ
```

```
;-----------------------------------------------------------------
;                         Run Forward
; (POSH,POSL)=RUN_FORWARD(POSH,POSL,POSSTOPH,POSSTOPL)
;-----------------------------------------------------------------
RUN_FORWARD:
                          PUSH  ACC
                          MOV   TH0,#000H
                          MOV   TL0,#000H
                          SETB  ET0
                          SETB  TR0
                          CLR   C
                          MOV   A,POSSTOPL
                          SUBB  A,POSL
                          MOV   POSDL,A
                          MOV   A,POSSTOPH
                          SUBB  A,POSH
                          MOV   POSDH,A
                          CLR   C
                          MOV   A,POSDL
                          SUBB  A,#20
                          MOV   POSDL,A
                          MOV   A,POSDH
                          SUBB  A,#0
                          MOV   POSDH,A

FORWARD10:
                          CLR   C
                          MOV   POS2H,TH0
                          MOV   POS2L,TL0
                          MOV   A,POS2H
                          SUBB  A,POSDH
```

```
                JC      SET_FW_SIGNAL
                JNZ     STOP_FW
                MOV     A,POSDL
                SUBB    A,POS2L
                JC      STOP_FW
                LJMP    SET_FW_SIGNAL
STOP_FW:
                CLR     FWD_SIG
                CLR     BWD_SIG
                LCALL   DELAY_1s
                LCALL   DELAY_1s
                MOV     HEXH,TH0
                MOV     HEXL,TL0
                CLR     C
                MOV     A,POSL
                ADD     A,HEXL
                MOV     POSL,A
                MOV     A,POSH
                ADDC    A,HEXH
                MOV     POSH,A
                MOV     HEXH,POSH
                MOV     HEXL,POSL
                MOV     A,#000H
                LCALL   GOTO_LCD
                MOV     DPTR,#STOP
                MOV     LIMIT,#16
                LCALL   WRLINE_LCD
                LCALL   DELAY_1s
                CLR     TR0
                CLR     ET0
                POP     ACC
```

```
            RET
SET_FW_SIGNAL:
            MOV   P0,#00000000B
            SETB  FWD_SIG
            CLR   BWD_SIG
            LJMP  FORWARD10

;------------------------------------------------------------
;           Run Backward
; (POSH,POSL)=RUN_BACKWARD(POSH,POSL,POSSTOPH,POSSTOPL)
;------------------------------------------------------------
RUN_BACKWARD:
            PUSH  ACC
            MOV   TH0,#000H
            MOV   TL0,#000H
            SETB  ET0
            SETB  TR0
            CLR   C
            MOV   A,POSL
            SUBB  A,POSSTOPL
            MOV   POSDL,A
            MOV   A,POSH
            SUBB  A,POSSTOPH
            MOV   POSDH,A
            CLR   C
            MOV   A,POSDL
            SUBB  A,#20
            MOV   POSDL,A
            MOV   A,POSDH
            SUBB  A,#0
            MOV   POSDH,A
```

```
BACKWARD10:
            CLR   C
            MOV   POS2H,TH0
            MOV   POS2L,TL0
            MOV   A,POS2H
            SUBB  A,POSDH
            JC    SET_BW_SIGNAL
            JNZ   STOP_BW
            MOV   A,POSDL
            SUBB  A,POS2L
            JC    STOP_BW
            LJMP  SET_BW_SIGNAL
STOP_BW:
            CLR   BWD_SIG
            CLR   FWD_SIG
            LCALL DELAY_1s
            LCALL DELAY_1s
            MOV   HEXH,TH0
            MOV   HEXL,TL0
            CLR   C
            MOV   A,POSL
            SUBB  A,HEXL
            MOV   POSL,A
            MOV   A,POSH
            SUBB  A,HEXH
            MOV   POSH,A
            MOV   HEXH,POSH
            MOV   HEXL,POSL
            MOV   A,#000H
            LCALL GOTO_LCD
```

```
                MOV   DPTR,#STOP
                MOV   LIMIT,#16
                LCALL WRLINE_LCD
                LCALL DELAY_1s
                CLR   TR0
                CLR   ET0
                POP   ACC
                RET
SET_BW_SIGNAL:
                MOV   P0,#00000000B
                SETB  BWD_SIG
                CLR   FWD_SIG
                LJMP  BACKWARD10


;------------------------------------------------------------------
;       Show decimal number to LCD
;------------------------------------------------------------------
SHOW_DEC:
                PUSH  ACC
                PUSH  B
                PUSH  1
                PUSH  2
                MOV   I,#5
SHOW_DEC1:
                MOV   A,HEXH
                MOV   B,#10
                DIV   AB
                MOV   HEXH,A
                MOV   A,HEXL
                ANL   A,#0F0H
                ORL   A,B
```

```
SWAP  A
MOV   B,#10
DIV   AB
MOV   R1,A
MOV   A,B
SWAP  A
MOV   B,A
MOV   A,HEXL
ANL   A,#0FH
ORL   A,B
MOV   B,#10
DIV   AB
MOV   R2,A
MOV   DATA1,DATA2
MOV   DATA2,DATA3
MOV   DATA3,DATA4
MOV   DATA4,DATA5
MOV   DATA5,B
MOV   A,R1
SWAP  A
ORL   A,R2
MOV   HEXL,A
DJNZ  I,SHOW_DEC1
LCALL SHOW_LCD
LCALL DELAY
POP   2
POP   1
POP   B
POP   ACC
RET
```

```
;------------------------------------------------------------
;*                show LCD               *
;------------------------------------------------------------
SHOW_LCD:
                PUSH   ACC
                MOV    A,DATA5
                ADD    A,#030H
                ACALL  WR_LCD
                MOV    A,DATA4
                ADD    A,#030H
                ACALL  WR_LCD
                MOV    A,DATA3
                ADD    A,#030H
                ACALL  WR_LCD
                MOV    A,DATA2
                ADD    A,#030H
                ACALL  WR_LCD
                MOV    A,DATA1
                ADD    A,#030H
                ACALL  WR_LCD
                POP    ACC
                RET


;*****************************************************
;*            Input  : ACC  (ASCII)            *
;*            Output : Data bus LCD             *
;*****************************************************
WR_LCD:
                PUSH   B
                PUSH   2
```

```
                    MOV    B,A
                    ANL    A,#0F0H
                    ORL    A,#00001100B    ;RS/CS HI
                    MOV    R2,A
                    MOV    A,P1
                    ANL    A,#00000011B
                    ORL    A,R2
                    MOV    P1,A
                    LCALL  EN_LCD
                    MOV    A,B             ;Low byte
                    SWAP   A
                    ANL    A,#0F0H
                    ORL    A,#00001100B
                    MOV    R2,A
                    MOV    A,P1
                    ANL    A,#00000011B
                    ORL    A,R2
                    MOV    P1,A
                    LCALL  EN_LCD
                    POP    2
                    POP    B
                    RET

;------------------------------------------------------------
;       Write Line to LCD
;------------------------------------------------------------
WRLINE_LCD:
                    PUSH   ACC
                    PUSH   0
                    MOV    R0,#0
WRLINE_LCD1:
```

```
        MOV   A,#0
        MOVC  A,@A+DPTR
        LCALL WR_LCD
        INC   DPTR
        INC   R0
        MOV   A,R0
        CJNE  A,LIMIT,WRLINE_LCD1
        POP   0
        POP   ACC
        RET
;*****************************************************
;*        Write Instruction LCD          *
;*        Input  : ACC (Command)          *
;*        Output : Data bus LCD           *
;=====================================================
WR_INS:
        PUSH  ACC
        PUSH  B
        MOV   B,A
        ANL   A,#0F0H
        SETB  ACC.3          ;CS HI
        MOV   R2,A
        MOV   A,P1
        ANL   A,#00000011B   ;else bit
        ORL   A,R2
        MOV   P1,A           ; High byte
        LCALL EN_LCD
        MOV   A,B            ; Low byte
        SWAP  A
        ANL   A,#0F0H
        SETB  ACC.3
```

```
                MOV    R2,A
                MOV    A,P1
                ANL    A,#00000011B
                ORL    A,R2
                MOV    P1,A
                LCALL  EN_LCD
                POP    B
                POP    ACC
                RET

;****************************************************
;*          Goto position of LCD          *
;*          Input : ACC (addr.)          *
;****************************************************
GOTO_LCD:
                PUSH   ACC
                SETB   ACC.7
                LCALL  WR_INS
                POP    ACC
                RET

;----------------------------------------------------------
;       Clear LCD Screen
;----------------------------------------------------------
CLR_SCREEN:
                PUSH   ACC
                MOV    A,#01H
                LCALL WR_INS
                POP    ACC
                RET
```

```
;**************************************************
;*              Initial LCD              *
;*              4-Bit Interface              *
;**************************************************
INIT_LCD:
                PUSH  ACC
                CLR   RS_LCD
                MOV   A,#33H      ; Set DL = 1 3-time
                LCALL WR_INS
                MOV   A,#32H      ; Clear DL = 0 1-time
                LCALL WR_INS
                MOV   A,#28H      ; Function set
                LCALL WR_INS      ; DL=0 4Bit,N=1 2Line,F=0 5X7
                MOV   A,#0CH
                LCALL WR_INS      ; Entry display,cursor off,cursor not blink
                MOV   A,#06H      ; Entry mode set
                LCALL WR_INS      ; I/D=1 Increment,S=0 cursor shift
                MOV   A,#01H      ; Clear display
                LCALL WR_INS      ; Clear display,set DD RAM address=0
                POP   ACC
                RET
;**************************************************
;*              Enable Pin E LCD              *
;*              Active Chip select              *
;**************************************************
EN_LCD:         CLR   CS_LCD      ; Enable LCD
                LCALL BUSY        ; Busy delay time
                SETB  CS_LCD      ; Disable LCD
                RET
```

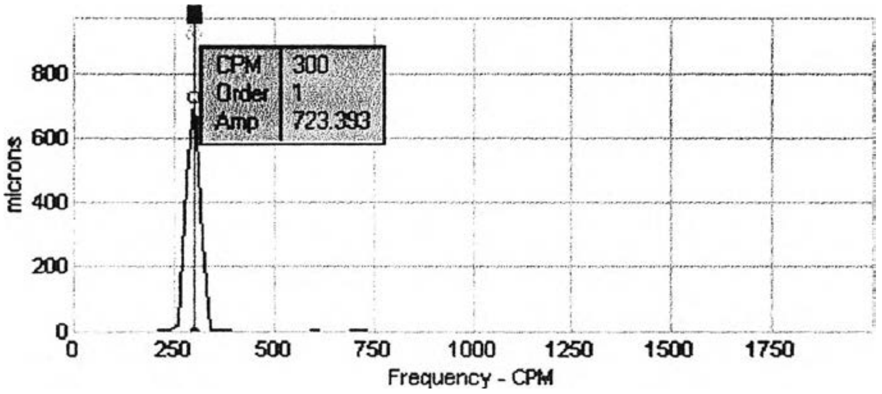
```
;*****************************************************
;*              Delay time for Busy             *
;*               Wait LCD Ready                 *
;*****************************************************
BUSY:     PUSH   07H
          PUSH   06H
          MOV    R6,#05H
BUSY1:    MOV    R7,#0FFH
          DJNZ   R7,$
          DJNZ   R6,BUSY1
          POP    06H
          POP    07H
          RET

;*****************************************************
;*                 Delay Time                   *
;*****************************************************
DELAY:    PUSH   07H
          PUSH   06H
DEL1:     MOV    R6,#0FFH
DEL2:     MOV    R7,#0FFH
          DJNZ   R7,$
          DJNZ   R6,DEL2
          POP    06H
          POP    07H
          RET

;*****************************************************
;*              Delay Time 1 second                  *
;*****************************************************
DELAY_1s:           PUSH   07H
```

```
                 PUSH  06H
                 PUSH  05H
                 MOV   5,#100
DELAY_10ms:      MOV   7,#10                  ;Each loop = 10 ms
DELAY_2:         MOV   6,#0BFH                ;Each loop = 1 ms
DELAY_3:         NOP
                 NOP
                 NOP
                 NOP
                 NOP
                 NOP
                 DJNZ  R6,DELAY_3
                 DJNZ  R7,DELAY_2
                 DJNZ  R5,DELAY_10ms
                 POP   05H
                 POP   06H
                 POP   07H
                 RET


;                      '0123456789ABCDEF'

FWD_RUN:         DB    'FWD RUN        '
BWD_RUN:         DB    'BWD RUN        '
STOP:            DB    'STOP AT        '
HZ:              DB    ' Hz'
DETECT:          DB    'DETECT '
MOVE:            DB    'MOVE TO        '
RPM:             DB    ' RPM'
POSITION13:      DB    'POSITION 42 cm. '
POSITION15:      DB    'POSITION 39 cm. '
```

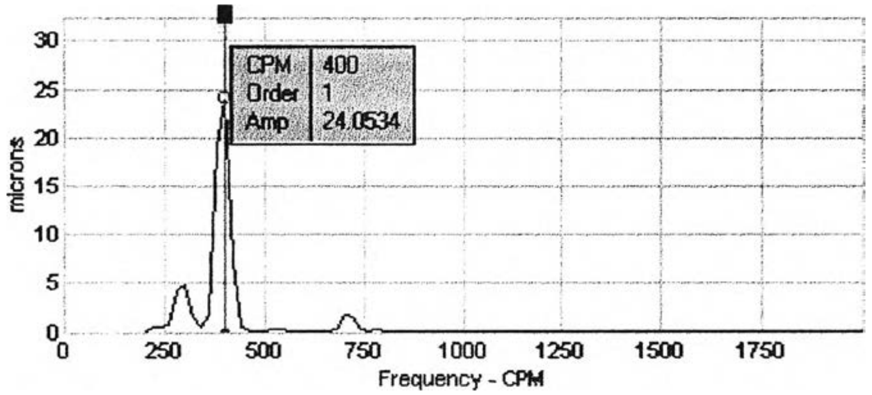
```
POSITION17:     DB    'POSITION 37 cm. '
POSITION18:     DB    'POSITION 36 cm. '
POSITION20:     DB    'POSITION 35 cm. '
POSITION22:     DB    'POSITION 34 cm. '
POSITION23:     DB    'POSITION 33 cm. '
POSITION25:     DB    'POSITION 32 cm. '

END
```

ภาคผนวก ข

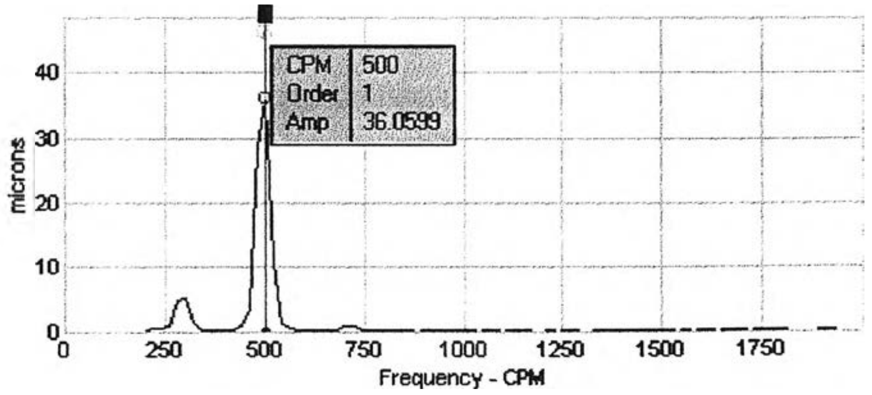
ตัวอย่างภาพสัญญาณโดเมนความถี่ที่ได้จากเครื่องมือวัด MICROLOG



เครื่องจักรทำงานที่ 300 rpm

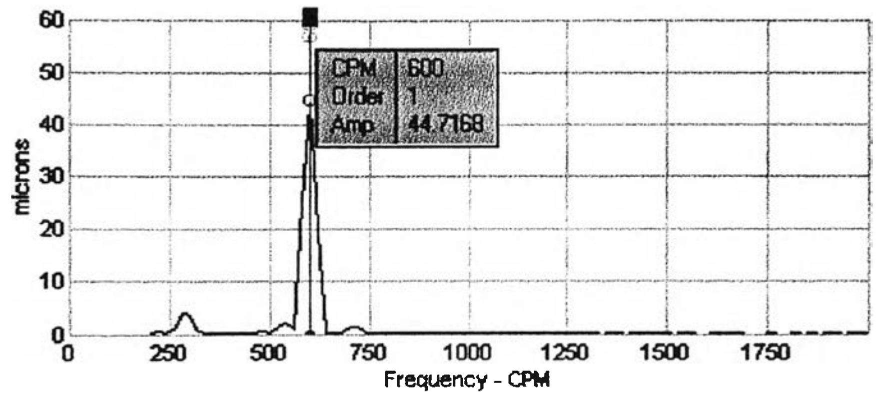


เครื่องจักรทำงานที่ 400 rpm

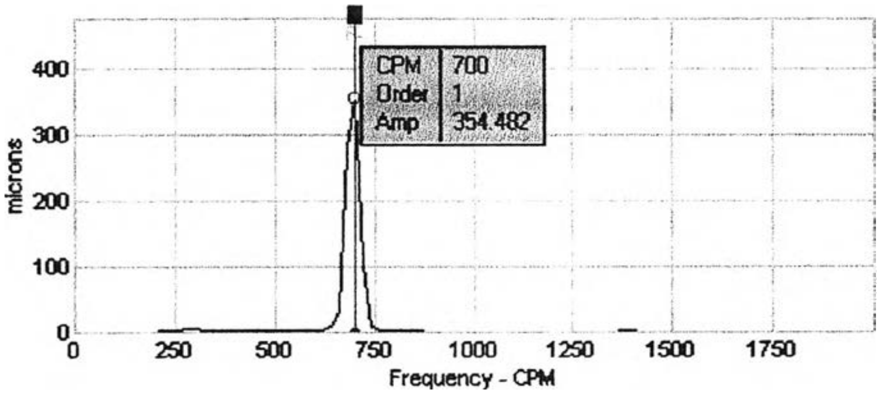


เครื่องจักรทำงานที่ 500 rpm

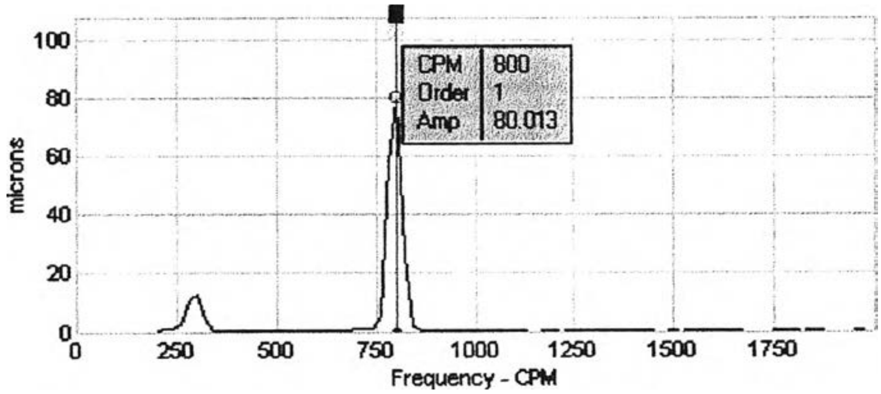
รูปที่ ข-1 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน x จากข้อมูลดิบในตาราง 5-1



เครื่องจักรทำงานที่ 600 rpm

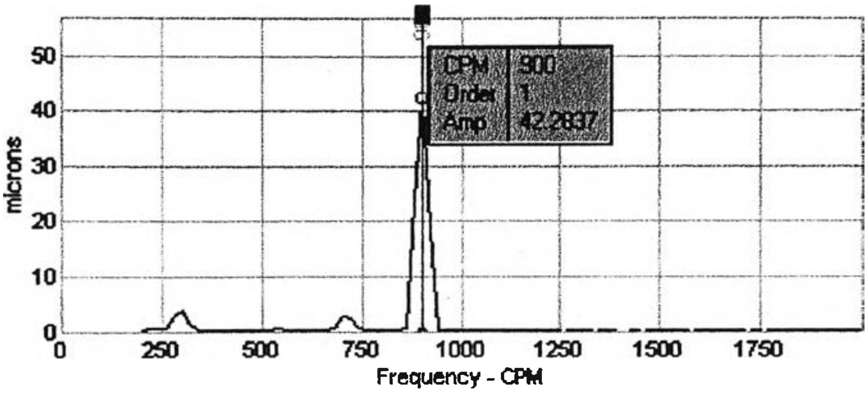


เครื่องจักรทำงานที่ 700 rpm

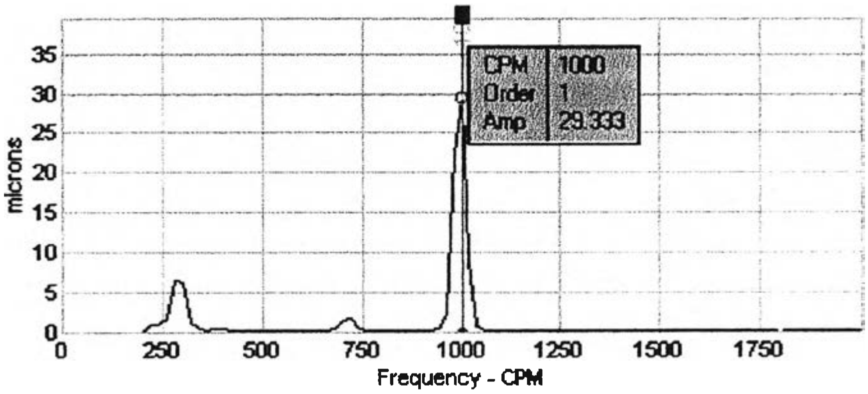


เครื่องจักรทำงานที่ 800 rpm

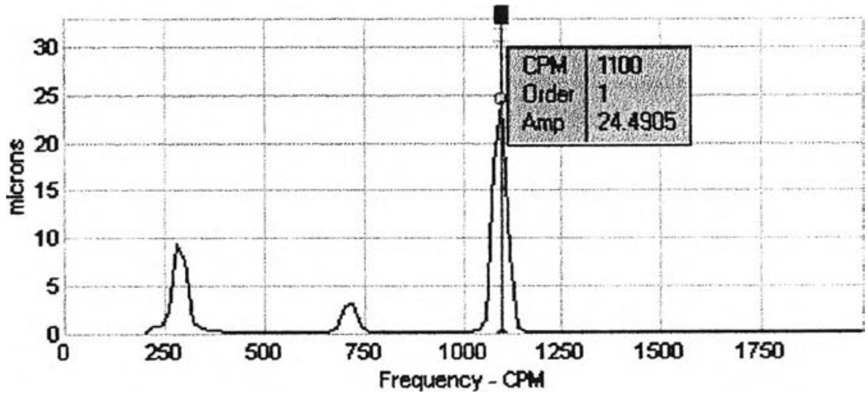
รูปที่ ข-1 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน x จากข้อมูลดิบในตาราง 5-1 (ต่อ)



เครื่องจักรทำงานที่ 900 rpm

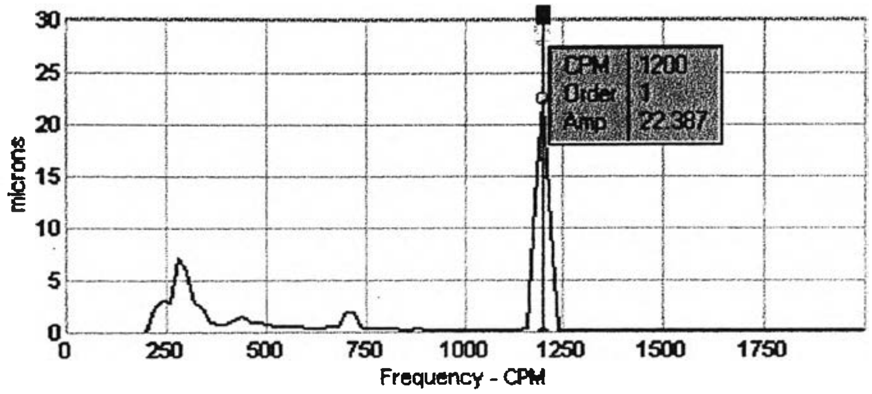


เครื่องจักรทำงานที่ 1000 rpm

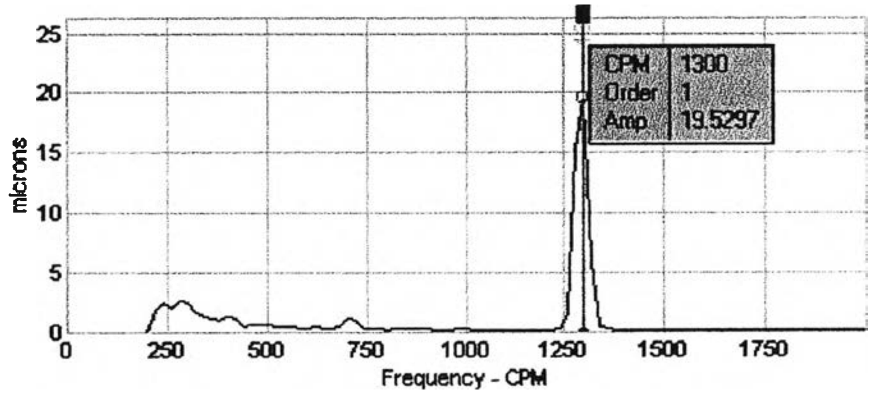


เครื่องจักรทำงานที่ 1100 rpm

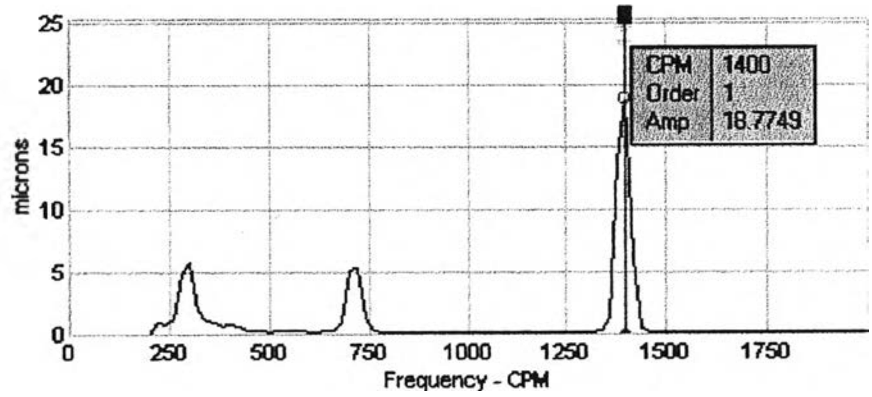
รูปที่ ข-1 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน x จากข้อมูลดิบในตาราง 5-1 (ต่อ)



เครื่องจักรทำงานที่ 1200 rpm

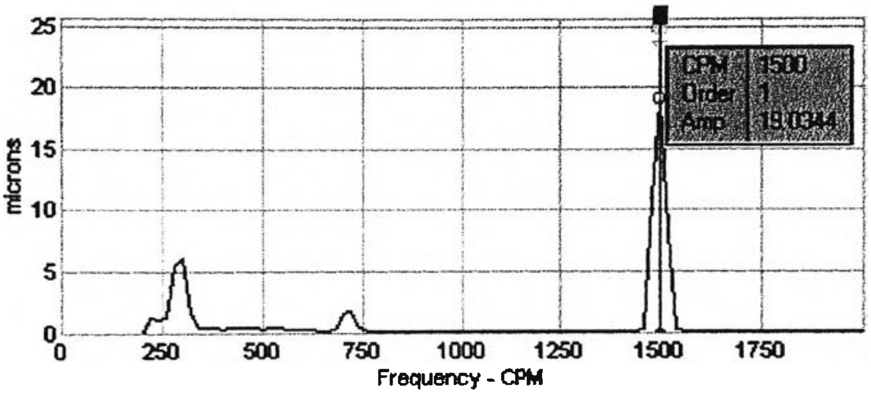


เครื่องจักรทำงานที่ 1300 rpm



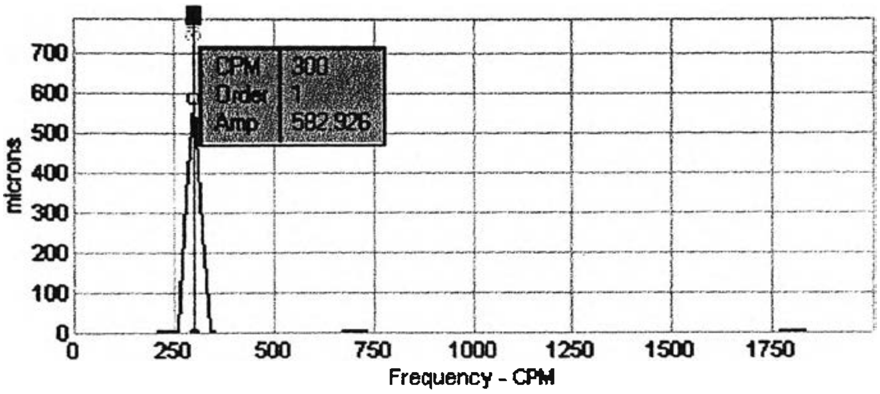
เครื่องจักรทำงานที่ 1400 rpm

รูปที่ ข-1 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน x จากข้อมูลดิบในตาราง 5-1 (ต่อ)

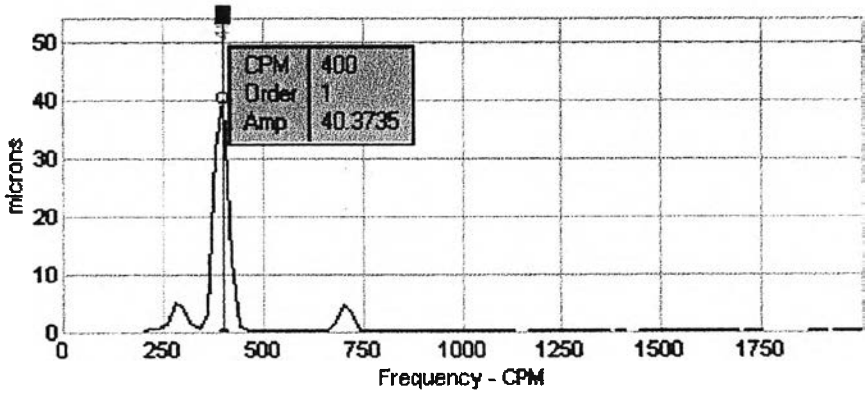


เครื่องจักรทำงานที่ 1500 rpm

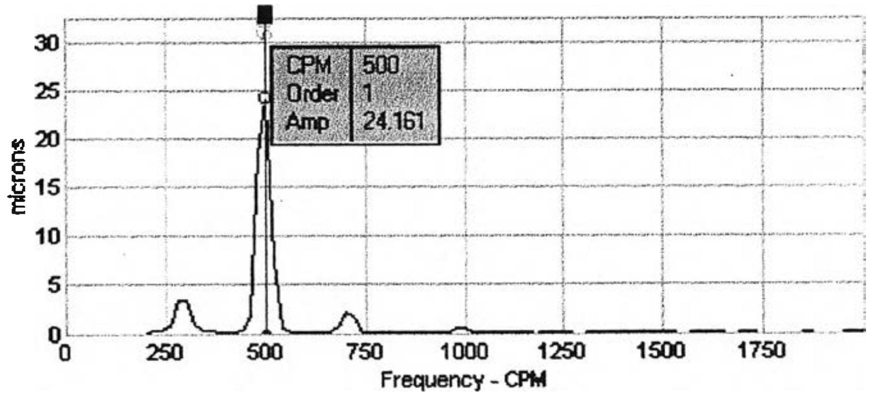
รูปที่ ข-1 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน x จากข้อมูลดิบในตาราง 5-1 (ต่อ)



เครื่องจักรทำงานที่ 300 rpm

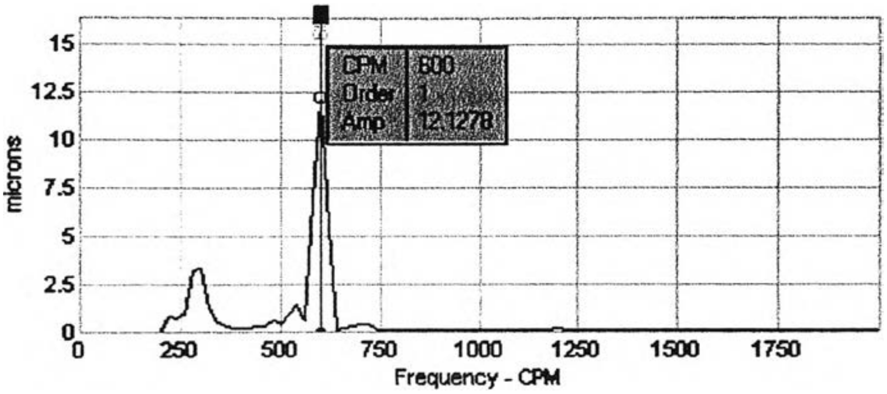


เครื่องจักรทำงานที่ 400 rpm

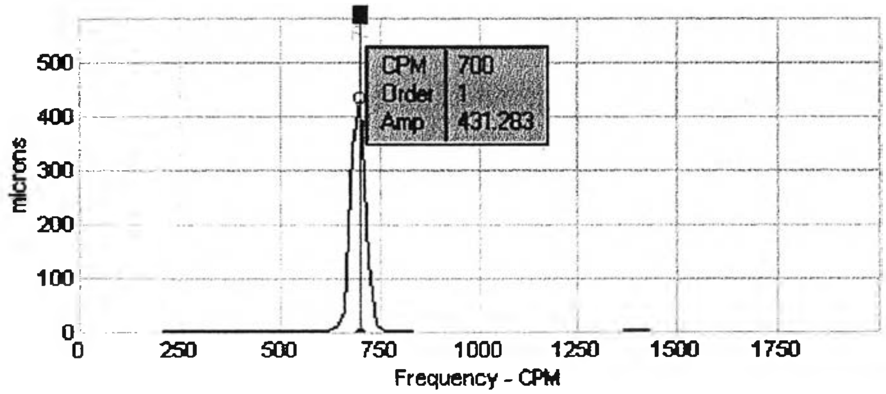


เครื่องจักรทำงานที่ 500 rpm

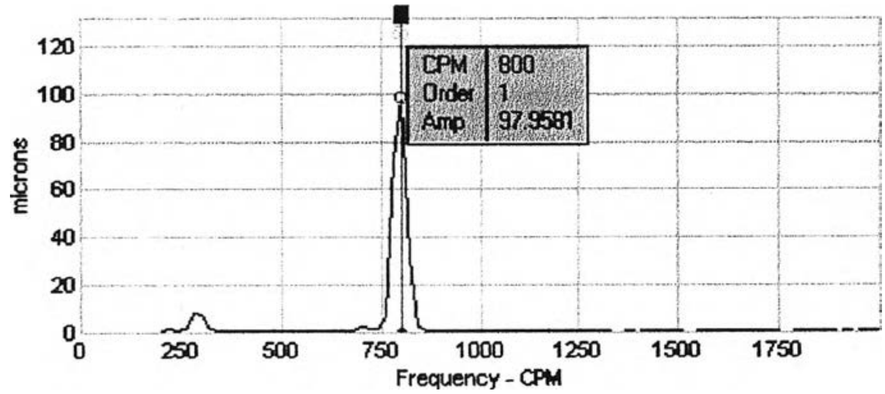
รูปที่ ข-2 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน y จากข้อมูลดิบในตาราง 5-1



เครื่องจักรทำงานที่ 600 rpm

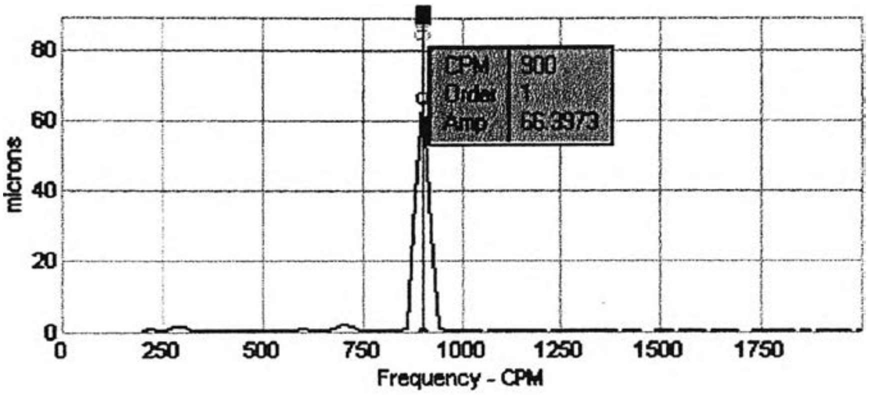


เครื่องจักรทำงานที่ 700 rpm

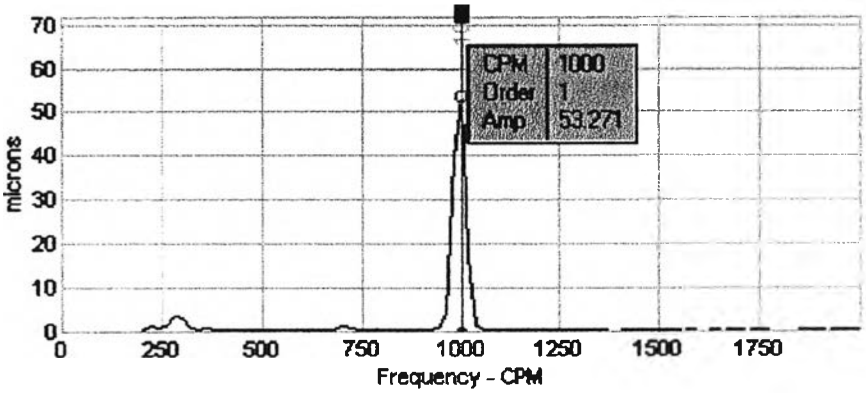


เครื่องจักรทำงานที่ 800 rpm

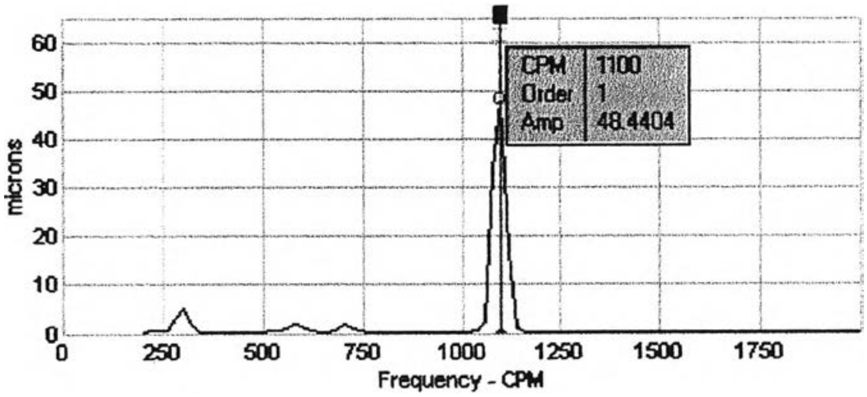
รูปที่ ข-2 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน y จากข้อมูลดิบในตาราง 5-1 (ต่อ)



เครื่องจักรทำงานที่ 900 rpm

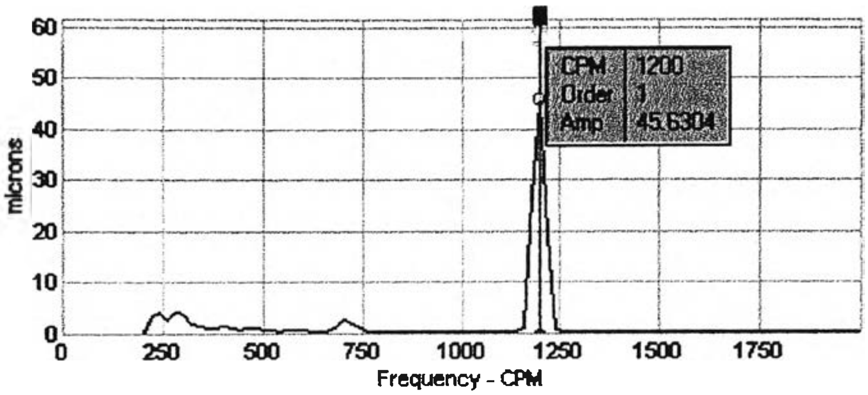


เครื่องจักรทำงานที่ 1000 rpm

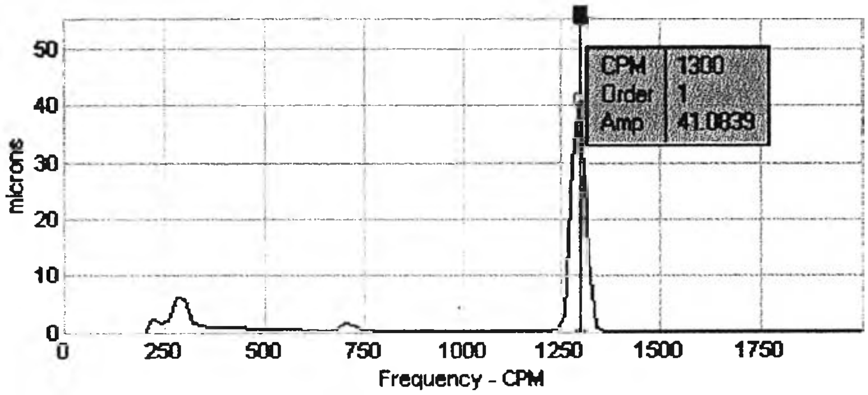


เครื่องจักรทำงานที่ 1100 rpm

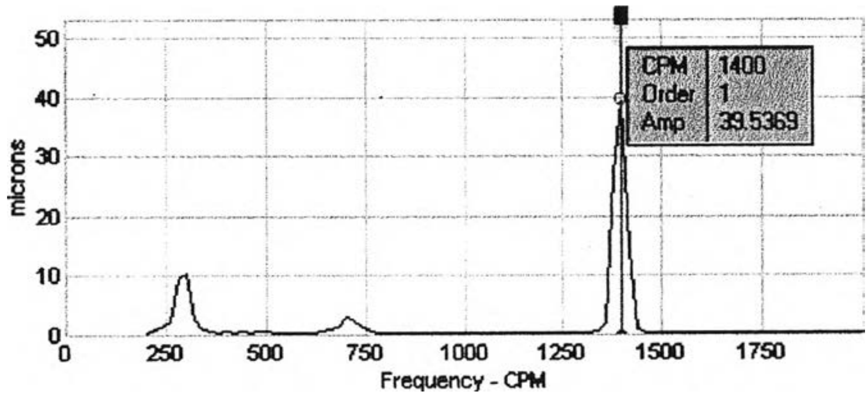
รูปที่ ข-2 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน y จากข้อมูลดิบในตาราง 5-1 (ต่อ)



เครื่องจักรทำงานที่ 1200 rpm

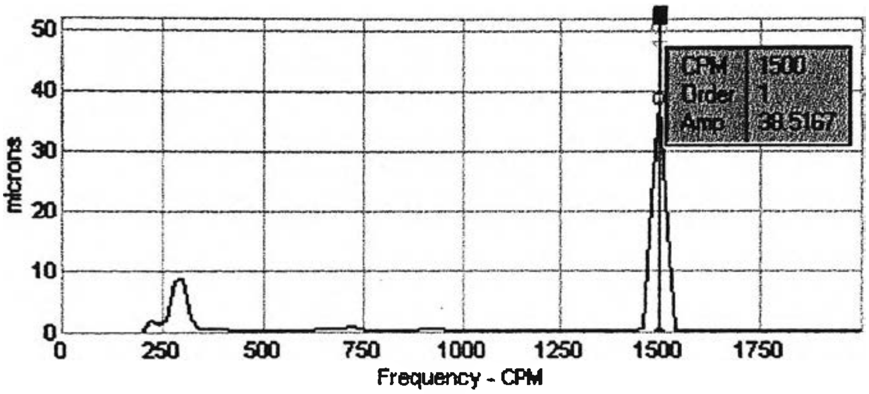


เครื่องจักรทำงานที่ 1300 rpm



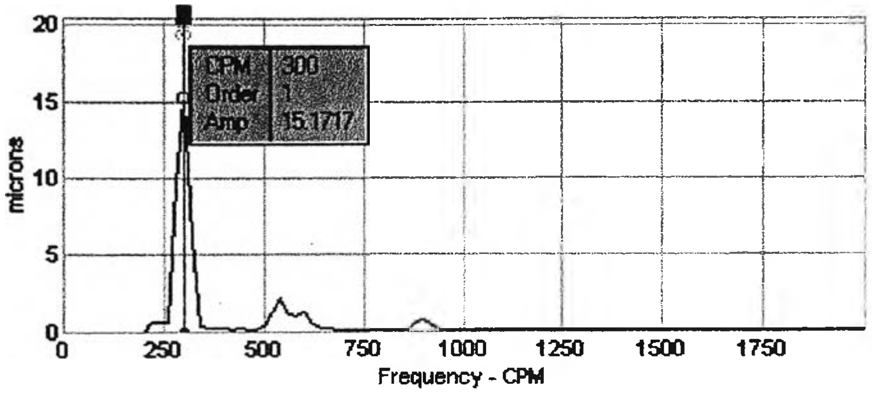
เครื่องจักรทำงานที่ 1400 rpm

รูปที่ ข-2 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน y จากข้อมูลดิบในตาราง 5-1 (ต่อ)

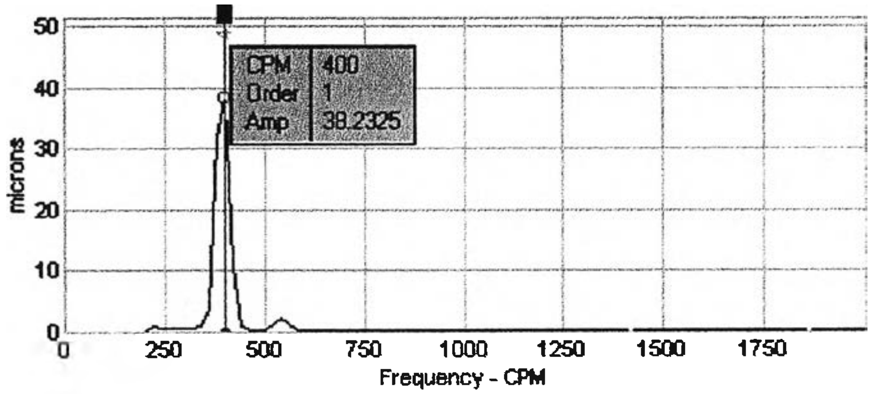


เครื่องจักรทำงานที่ 1500 rpm

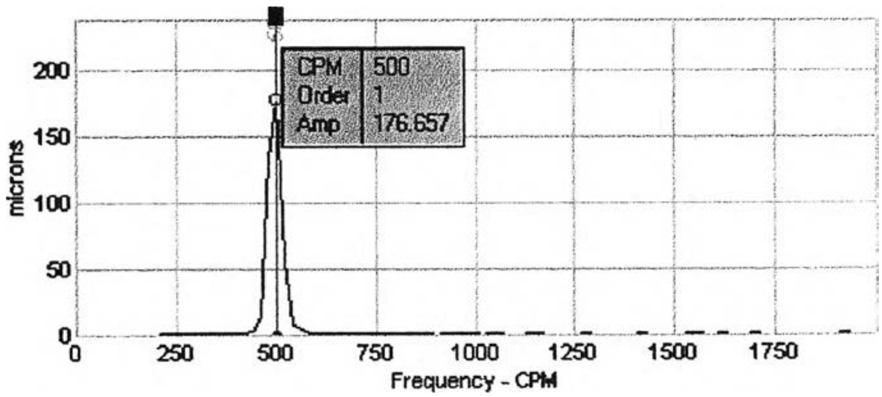
รูปที่ ข-2 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน y จากข้อมูลดิบในตาราง 5-1 (ต่อ)



เครื่องจักรทำงานที่ 300 rpm

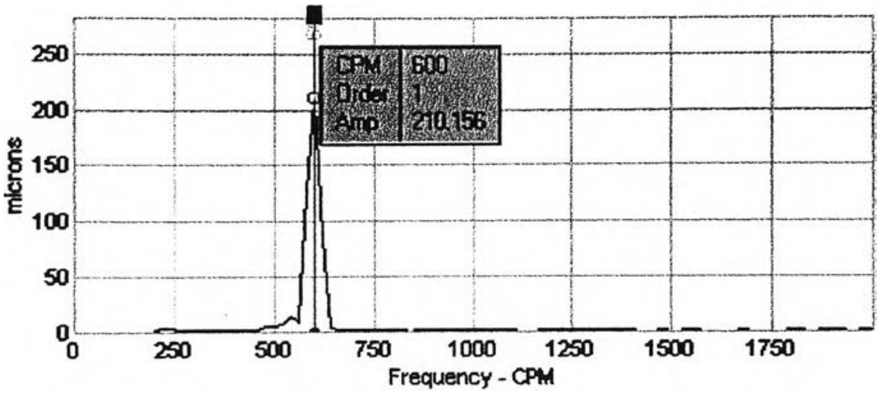


เครื่องจักรทำงานที่ 400 rpm

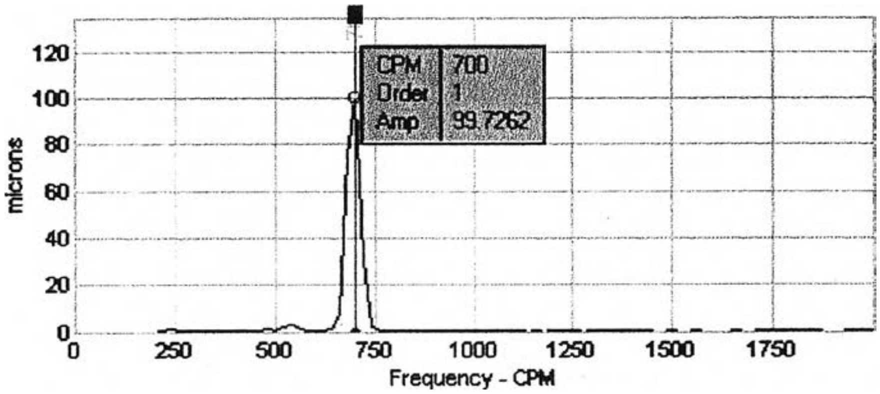


เครื่องจักรทำงานที่ 500 rpm

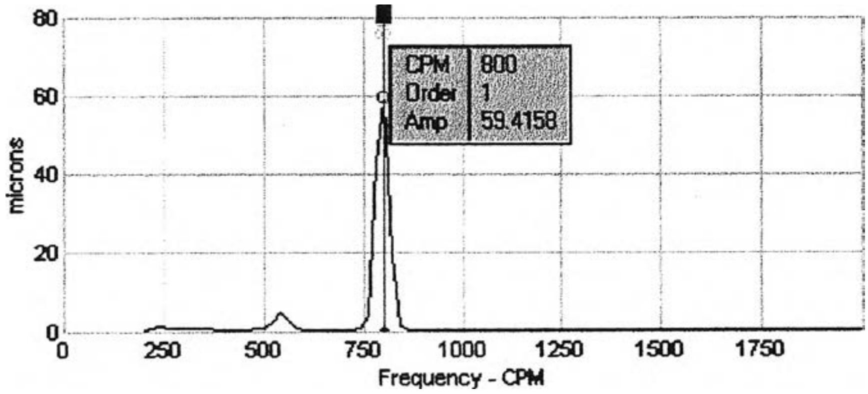
รูปที่ ข-3 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน z จากข้อมูลดิบในตาราง 5-1



เครื่องจักรทำงานที่ 600 rpm

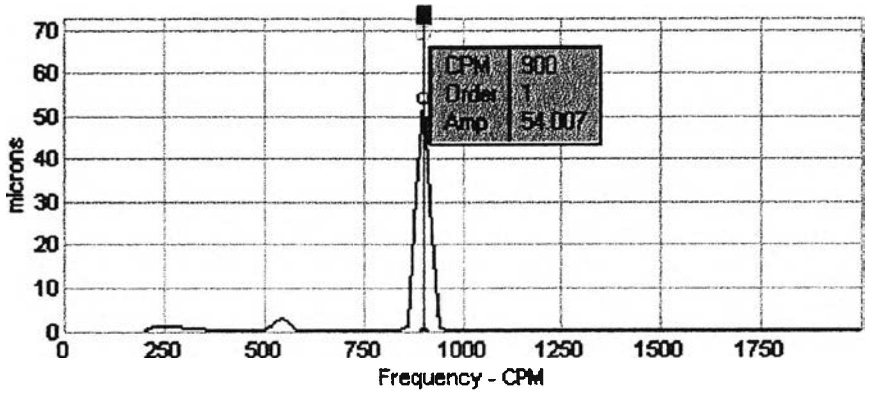


เครื่องจักรทำงานที่ 700 rpm

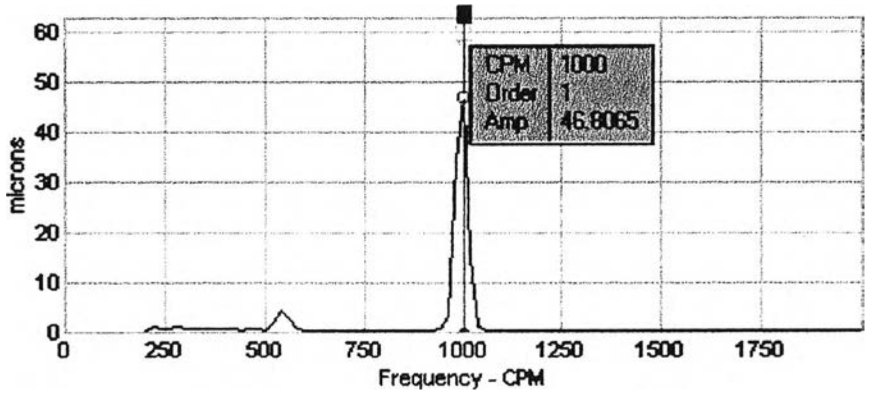


เครื่องจักรทำงานที่ 800 rpm

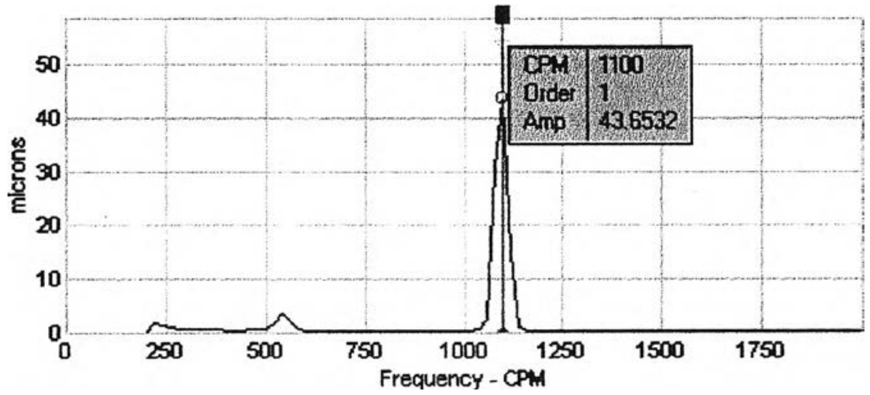
รูปที่ ข-3 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน z จากข้อมูลดิบในตาราง 5-1 (ต่อ)



เครื่องจักรทำงานที่ 900 rpm

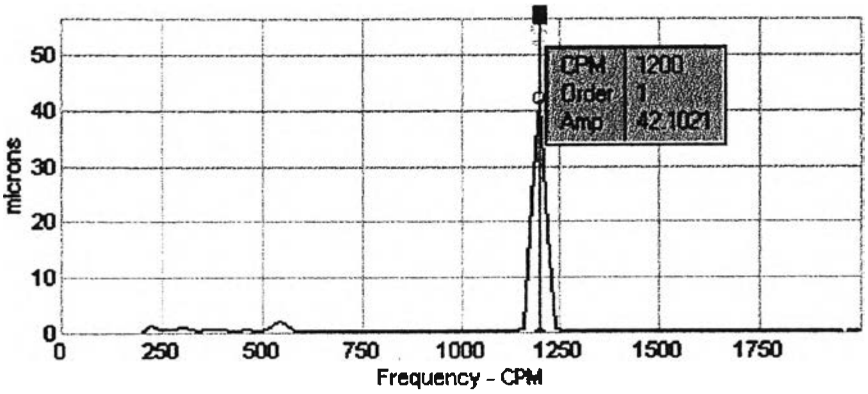


เครื่องจักรทำงานที่ 1000 rpm

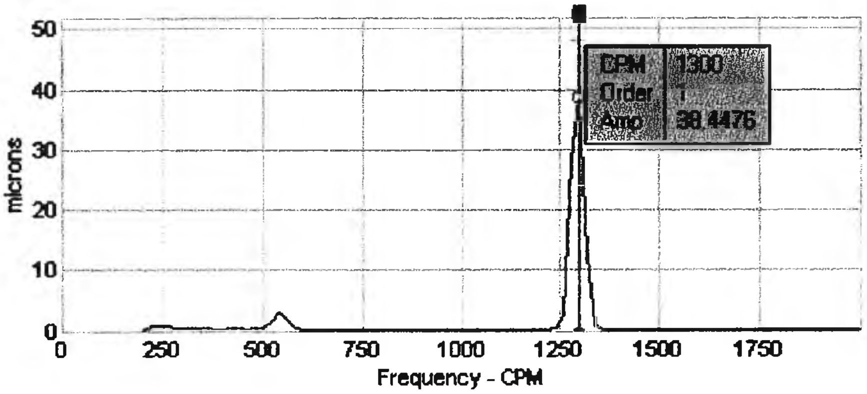


เครื่องจักรทำงานที่ 1100 rpm

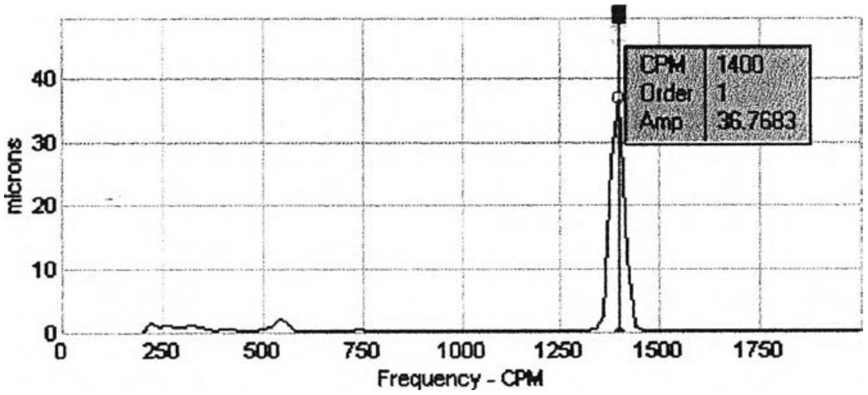
รูปที่ ข-3 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน z จากข้อมูลดิบในตาราง 5-1 (ต่อ)



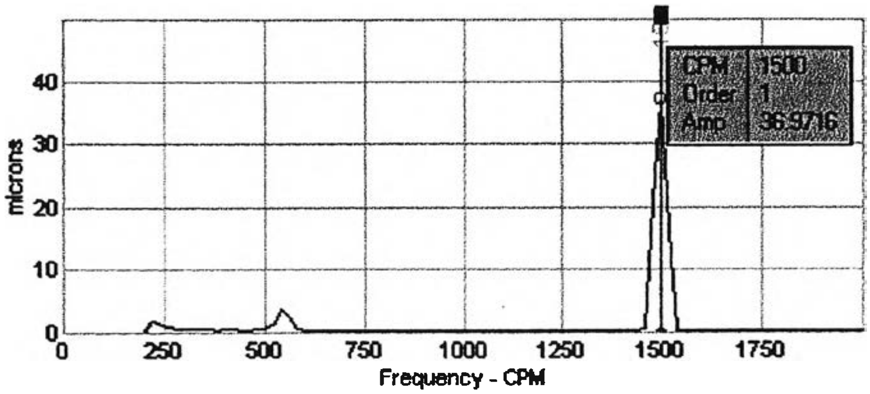
เครื่องจักรทำงานที่ 1200 rpm



เครื่องจักรทำงานที่ 1300 rpm



เครื่องจักรทำงานที่ 1400 rpm



เครื่องจักรทำงานที่ 1500 rpm

รูปที่ ข-3 ภาพสัญญาณโดเมนความถี่ของการสั่นสะเทือน z จากข้อมูลดิบในตาราง 5-1 (ต่อ)

ภาคผนวก ค

ค่าอัตราส่วนความหน่วง (Damping Ratio) โดยประมาณ สำหรับสปริงที่ใช้วัสดุต่าง ๆ กัน

Material	Approximate damping ratio
Steel spring	0.005
Natural rubber	0.05
Neoprene	0.05
Friction-damped springs	0.33
Metal mesh	0.12
Air damping	0.17
Felt and cork	0.06

ตารางที่ ค-1 ค่าอัตราส่วนความหน่วง (Damping Ratio) โดยประมาณ สำหรับสปริงที่ใช้วัสดุต่าง ๆ กัน [5]

ภาคผนวก ง

การคำนวณหาค่าความถี่ธรรมชาติของระบบในบทที่ 5

จากสมการ 5-3 ถึง 5-8 ซึ่งเป็นสมการแสดงขนาดการกระจัดทั้ง 6 พิกัดของระบบ ดังนี้

$$X = -\frac{-4asdmk_x\omega^2F_0 + 16asdk_x^2F_0 - 16s^2dk_x^2F_0}{4sk_x(-m\omega^4I_y + 4s^2m\omega^2k_x + 4\omega^2I_yk_x + 4b^2m\omega^2k_z - 16b^2k_xk_z)} \quad (ง\text{-}1)$$

$$Y = \frac{eF_0}{4sk_y} + \frac{(-me\omega^2F_0 - 4sk_yF_0 + 4ek_yF_0)(\omega^2I_x + 4s(s-a)k_y + 4b^2k_z)}{4sk_y(-m\omega^4I_x + 4s^2m\omega^2k_y + 4\omega^2I_xk_y + 4b^2m\omega^2k_z - 16b^2k_yk_z)} \quad (ง\text{-}2)$$

$$Z = \frac{F_0}{4\omega^2 - 4k_z} \quad (ง\text{-}3)$$

$$\beta_0 = -\frac{-me\omega^2F_0 - 4sk_yF_0 + 4ek_yF_0}{-m\omega^4I_x + 4s^2m\omega^2k_y + 4\omega^2I_xk_y + 4b^2m\omega^2k_z - 16b^2k_yk_z} \quad (ง\text{-}4)$$

$$\theta_0 = \frac{dF_0(m\omega^2 - 4k_x)}{-m\omega^4I_y + 4s^2m\omega^2k_x + 4\omega^2I_yk_x + 4b^2m\omega^2k_z - 16b^2k_xk_z} \quad (ง\text{-}5)$$

$$\gamma_0 = \frac{dF_0}{\omega^2I_z - 4b^2k_x - 4b^2k_y} \quad (ง\text{-}6)$$

สามารถคำนวณความถี่ธรรมชาติได้โดยกำหนดพจน์ที่เป็นตัวหารในแต่ละสมการให้มีค่าเป็น 0 ดังนั้น คำนวณค่าความถี่ธรรมชาติของพิกัด x และ θ ซึ่งมีการควบของพิกัดได้จากค่า ω ในสมการ

$$-m\omega^4I_y + 4s^2m\omega^2k_x + 4\omega^2I_yk_x + 4b^2m\omega^2k_z - 16b^2k_xk_z = 0 \quad (ง\text{-}7)$$

คำนวณค่าความถี่ธรรมชาติของพิกัด y และ β ซึ่งมีการควบของพิกัดได้จากค่า ω ในสมการ

$$-m\omega^4I_x + 4s^2m\omega^2k_y + 4\omega^2I_xk_y + 4b^2m\omega^2k_z - 16b^2k_yk = 0 \quad (ง\text{-}8)$$

คำนวณค่าความถี่ธรรมชาติของพิกัด z ได้จากค่า ω ในสมการ

$$4\omega^2 - 4k_z = 0 \quad (ง\text{-}9)$$

และคำนวณค่าความถี่ธรรมชาติของพิกัด γ ได้จากค่า ω ในสมการ

$$\omega^2I_z - 4b^2k_x - 4b^2k_y = 0 \quad (ง\text{-}10)$$

โดยพารามิเตอร์ต่าง ๆ มีค่าดังต่อไปนี้

s = 10.2 cm, a = 13.8 cm, b = 20 cm, d = 14 cm, e = 4.2 cm,

m = 53.85 kg,

$k_x = k_y$ = 4709 N/m, k_z = 47.2 kN/m,

I_x = 2.11 kg-m^2, I_y = 2.15 kg-m^2 และ I_z=2.50 kg-m^2

ดังนั้น คำนวณค่าความถี่ธรรมชาติใน 6 พิกัด โดยแทนค่าพารามิเตอร์ในสมการ ง-7, ง-8, ง-9 และ ง-10 คำนวณหาค่าความถี่ธรรมชาติจากค่า ω ได้ดังต่อไปนี้

-ความถี่ธรรมชาติของการเคลื่อนที่ในพิกัด x และ θ (มีการควบของพิกัด) อยู่ที่ 174 และ 581 rpm

-ความถี่ธรรมชาติของการเคลื่อนที่ในพิกัด z อยู่ที่ 565 rpm

-ความถี่ธรรมชาติของการเคลื่อนที่ในพิกัด y และ β (มีการควบของพิกัด) อยู่ที่ 174 และ 586 rpm

-ความถี่ธรรมชาติของการเคลื่อนที่ในพิกัด γ อยู่ที่ 234 rpm

ภาคผนวก จ

คำตอบของระบบสมการ 5-18 ในบทที่ 5

$$X=-((am\omega^2-4ak_x+4sk_x)(dm\omega^2F_0+lm\omega^2F_0-4dF_0k_z))/$$
$$(m^2\omega^6 i_y-4m^2s^2\omega^4k_x-4m\omega^4 i_y k_x-4b^2m^2\omega^4k_z-4l^2m^2\omega^4k_z-4m\omega^4 i_y k_z+16b^2m\omega^2k_xk_z+$$
$$16l^2m\omega^2k_xk_z+16ms^2\omega^2k_xk_z+16\omega^2 i_y k_xk_z+16b^2m\omega^2k_z^2-64b^2k_xk_z^2)$$

$$Y=$$
$$(-am\epsilon\omega^4 i_z F_0-\omega^4 i_x i_z F_0+4ab^2m\epsilon\omega^2F_0k_x+4b^2\omega^2 i_x F_0k_x+4ab^2m\epsilon\omega^2F_0k_y+4al^2m\epsilon\omega^2F_0k_y+$$
$$4b^2\omega^2 i_x F_0k_y-4dl\omega^2 i_x F_0k_y-4as\omega^2 i_z F_0k_y+4s^2\omega^2 i_z F_0k_y+4a\epsilon\omega^2 i_z F_0k_y-4s\epsilon\omega^2 i_z F_0k_y+$$
$$16ab^2sF_0k_xk_y-16b^2s^2F_0k_xk_y-16ab^2\epsilon F_0k_xk_y+16b^2s\epsilon F_0k_xk_y+16ab^2sF_0k_y^2-$$
$$16adlsF_0k_y^2-16b^2s^2F_0k_y^2+16dls^2F_0k_y^2-16ab^2\epsilon F_0k_y^2-16al^2\epsilon F_0k_y^2+16b^2s\epsilon F_0k_y^2-$$
$$4adlms\omega^2F_0k_z-4al^2ms\omega^2F_0k_z+4b^2\omega^2 i_z F_0k_z-16b^4F_0k_xk_z-16b^4F_0k_yk_z+$$
$$16b^2dlF_0k_yk_z+16adlsF_0k_yk_z+16al^2sF_0k_yk_z-16dls^2F_0k_yk_z)/$$
$$(m\omega^6 i_x i_z-4b^2m\omega^4 i_x k_x-4b^2m\omega^4 i_x k_y-4l^2m\omega^4 i_x k_y-4ms^2\omega^4 i_z k_y-4\omega^4 i_x i_z k_y+$$
$$16b^2ms^2\omega^2k_xk_y+16b^2\omega^2 i_x k_xk_y+16b^2ms^2\omega^2k_y^2+16l^2ms^2\omega^2k_y^2+16b^2\omega^2 i_x k_y^2-$$
$$64al^2sk_y^3-4b^2m\omega^4 i_z k_z+16b^4m\omega^2k_xk_z+16b^4m\omega^2k_yk_z+16b^2l^2m\omega^2k_yk_z-$$
$$16al^2ms\omega^2k_yk_z-16l^2ms^2\omega^2k_yk_z+16b^2\omega^2 i_z k_yk_z-64b^4k_xk_yk_z-64b^4k_y^2k_z+64al^2sk_y^2k_z)$$

$$Z=$$
$$(-dlm^2\omega^4F_0-l^2m^2\omega^4F_0-m\omega^4 i_y F_0+4dlm\omega^2F_0k_x+4l^2m\omega^2F_0k_x+4ms^2\omega^2F_0k_x+4\omega^2 i_y F_0k_x+$$
$$4b^2m\omega^2F_0k_z-16b^2F_0k_xk_z)/$$
$$(m^2\omega^6 i_y-4m^2s^2\omega^4k_x-4m\omega^4 i_y k_x-4b^2m^2\omega^4k_z-4l^2m^2\omega^4k_z-4m\omega^4 i_y k_z+16b^2m\omega^2k_xk_z+$$
$$16l^2m\omega^2k_xk_z+16ms^2\omega^2k_xk_z+16\omega^2 i_y k_xk_z+16b^2m\omega^2k_z^2-64b^2k_xk_z^2)$$

$$\beta_0=$$
$$(-m\epsilon\omega^4 i_z F_0+4b^2m\epsilon\omega^2F_0k_x+4b^2m\epsilon\omega^2F_0k_y+4l^2m\epsilon\omega^2F_0k_y-4s\omega^2 i_z F_0k_y+4\epsilon\omega^2 i_z F_0k_y+$$
$$16b^2sF_0k_xk_y-16b^2\epsilon F_0k_xk_y+16b^2sF_0k_y^2-16dlsF_0k_y^2-16b^2\epsilon F_0k_y^2-4dlms\omega^2F_0k_z-$$
$$4l^2ms\omega^2F_0k_z+16dlsF_0k_yk_z)/$$
$$(m\omega^6 i_x i_z-4b^2m\omega^4 i_x k_x-4b^2m\omega^4 i_x k_y-4l^2m\omega^4 i_x k_y-4ms^2\omega^4 i_z k_y-4\omega^4 i_x i_z k_y+$$
$$16b^2ms^2\omega^2k_xk_y+16b^2\omega^2 i_x k_xk_y+16b^2ms^2\omega^2k_y^2+16l^2ms^2\omega^2k_y^2+16b^2\omega^2 i_x k_y^2-$$
$$64al^2sk_y^3-4b^2m\omega^4 i_z k_z+16b^4m\omega^2k_xk_z+16b^4m\omega^2k_yk_z+16b^2l^2m\omega^2k_yk_z-$$
$$16al^2ms\omega^2k_yk_z-16l^2ms^2\omega^2k_yk_z+16b^2\omega^2 i_z k_yk_z-64b^4k_xk_yk_z-64b^4k_y^2k_z+64al^2sk_y^2k_z)$$

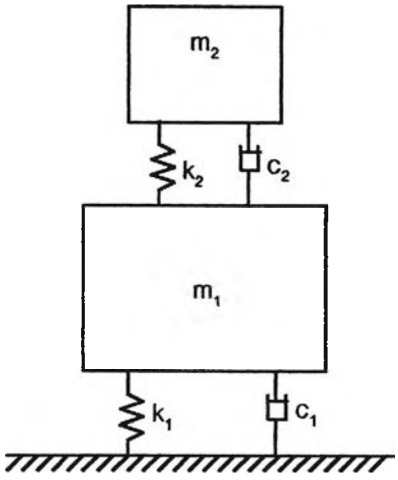
$$\theta_0=(-dm^2\omega^4F_0-lm^2\omega^4F_0+4dm\omega^2F_0k_x+4lm\omega^2F_0k_x+4dm\omega^2F_0k_z-16dF_0k_xk_z)/$$
$$(-m^2\omega^6 i_y+4m^2s^2\omega^4k_x+4m\omega^4 i_y k_x+4b^2m^2\omega^4k_z+4l^2m^2\omega^4k_z+4m\omega^4 i_y k_z-16b^2m\omega^2k_xk_z-$$
$$16l^2m\omega^2k_xk_z-16ms^2\omega^2k_xk_z-16\omega^2 i_y k_xk_z-16b^2m\omega^2k_z^2+64b^2k_xk_z^2)$$

$$\gamma_0=$$
$$(-dm\omega^4 i_x F_0-lm\omega^4 i_x F_0+4dms^2\omega^2F_0k_y+4lms^2\omega^2F_0k_y-4alm\epsilon\omega^2F_0k_y-4lms\epsilon\omega^2F_0k_y+$$
$$4d\omega^2 i_x F_0k_y-16alsF_0k_y^2+16al\epsilon F_0k_y^2+4b^2dm\omega^2F_0k_z+4b^2lm\omega^2F_0k_z-16b^2dF_0k_yk_z)/$$
$$(m\omega^6 i_x i_z-4b^2m\omega^4 i_x k_x-4b^2m\omega^4 i_x k_y-4l^2m\omega^4 i_x k_y-4ms^2\omega^4 i_z k_y-4\omega^4 i_x i_z k_y+$$
$$16b^2ms^2\omega^2k_xk_y+16b^2\omega^2 i_x k_xk_y+16b^2ms^2\omega^2k_y^2+16l^2ms^2\omega^2k_y^2+16b^2\omega^2 i_x k_y^2-$$
$$64al^2sk_y^3-4b^2m\omega^4 i_z k_z+16b^4m\omega^2k_xk_z+16b^4m\omega^2k_yk_z+16b^2l^2m\omega^2k_yk_z-$$
$$16al^2ms\omega^2k_yk_z-16l^2ms^2\omega^2k_yk_z+16b^2\omega^2 i_z k_yk_z-64b^4k_xk_yk_z-64b^4k_y^2k_z+64al^2sk_y^2k_z)$$

ภาคผนวก ฉ

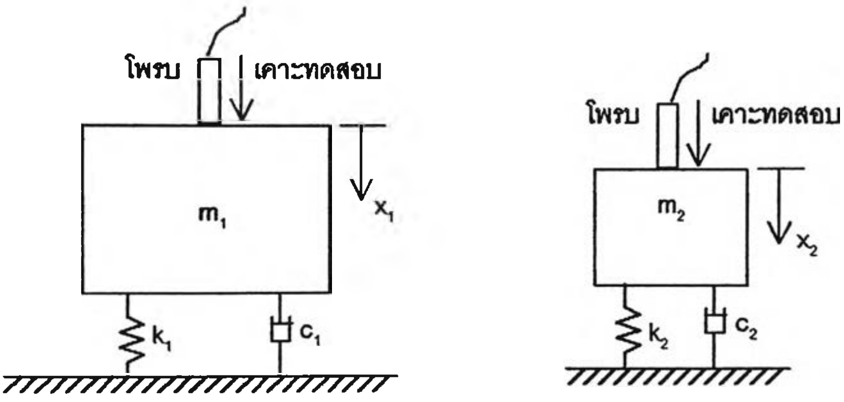
การเคาะทดสอบเพื่อหาค่าอัตราส่วนความหน่วงของระบบ ในหัวข้อ 7-1-4 ผลของค่าความหน่วงที่มีต่อระบบในการทดลอง

เนื่องจากระบบในการทดลองมีค่าความหน่วงแฝงอยู่ จึงวิเคราะห์ระบบที่มีการติดตั้งตัวดูดซับการสั่นสะเทือนและมีค่าความหน่วง โดยทดลองหาค่าความหน่วงจากการทดลองและเปรียบเทียบผลของความหน่วงที่มีต่อระบบ พิจารณาระบบในการทดลองที่ 7-1 ดังรูปที่ ฉ-1



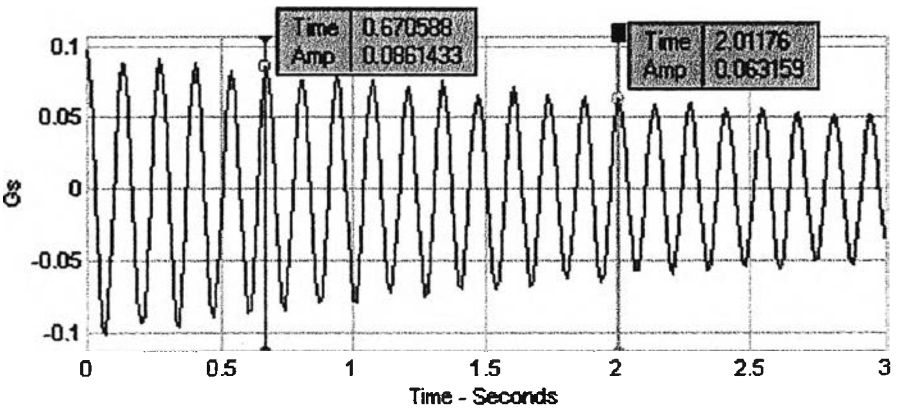
รูปที่ ฉ-1 ระบบในการทดลองเมื่อมีความหน่วง

จากรูปที่ ฉ-1 ระบบมีค่าความหน่วง c_1 และ c_2 ค่าความหน่วงทั้ง 2 ค่านี้เป็นส่วนที่เกิดจากพลังงานสูญเสียไปในสปริงหรือคานในแต่ละรอบของความเค้นที่เกิดขึ้นในสปริงหรือคาน เรียกว่า ความหน่วงเชิงโครงสร้าง (Structural Damping) ทดลองหาค่าความหน่วง c_1 และ c_2 โดยแยกระบบออกเป็น 2 ส่วนและทำการเคาะทดสอบ (Bump Test) ดังรูปที่ ฉ-2 (สปริง k_1 คือ ตัวกันการสั่นสะเทือน และสปริง k_2 คือคานที่เลื่อนจุดรองรับได้จากการทดลองที่ 7-1)



รูปที่ ฉ-2 ระบบในการทดลองที่ทำการเคาะทดสอบเพื่อหาค่าความหน่วง

ใช้ค้อนยางเคาะทดสอบ และใช้อุปกรณ์วัดสัญญาณการสั่นสะเทือนไมโครล็อกวัดการสั่นสะเทือนโดยติดตั้งโพรบไว้ด้านบนที่บริเวณกึ่งกลางของก้อนมวล วัดค่าความเร่ง (Acceleration) ของ $\ddot{x}_1$ ในรูปแบบของโดเมนเวลาในช่วงทรานเซียน (Transient) ได้ดังรูปที่ ฉ-3



รูปที่ ฉ-3 สัญญาณความเร่ง $\ddot{x}_1$ ในช่วงทรานเซียนในรูปแบบโดเมนเวลา

วิเคราะห์สัญญาณในรูปที่ ฉ-3 เพื่อหาค่าความหน่วง c_1 โดยคำนวณจากการลดลงแบบลอการิทึม (Logarithmic Decrement) ของเส้นกราฟจากความสัมพันธ์

$$\delta = \frac{2\pi\zeta}{\sqrt{1-\zeta^2}} = \frac{1}{n}\ln\frac{x_0}{x_n} \qquad (ฉ\text{-}1)$$

โดย δ คือ ค่าการลดลงแบบลอการิทึม (Logarithmic Decrement)

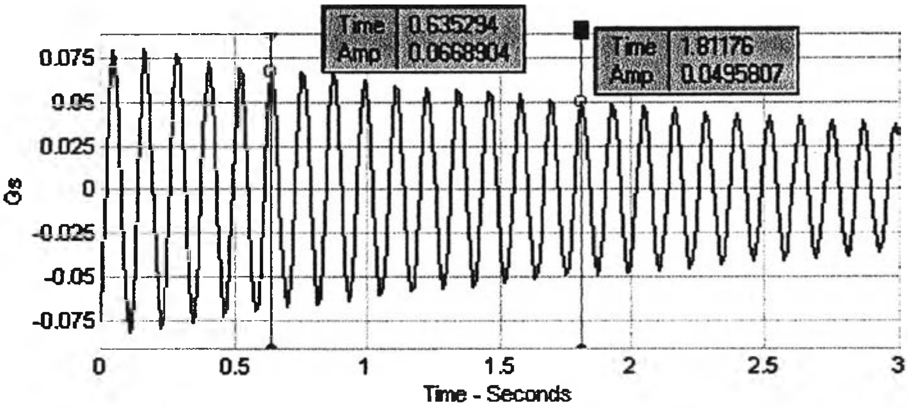
ζ คือ อัตราส่วนความหน่วง (Damping Ratio)

x_0 คือ แอมพลิจูดของสัญญาณความเร่งยอดแรกที่นำมาวิเคราะห์

x_n คือ แอมพลิจูดของสัญญาณความเร่งถัดไปอีก n ยอดที่นำมาวิเคราะห์

ดังนั้น จากรูปที่ ฉ-3 ค่า n = 10 , x_0 = 0.0861 Gs และ x_n = 0.0632 Gs คำนวณได้ค่าอัตราส่วนความหน่วง $\zeta_1 = 0.0049$

ใช้ค้อนยางเคาะทดสอบครั้งเดียว และใช้อุปกรณ์วัดสัญญาณการสั่นสะเทือนไมโครล็อกวัดการสั่นสะเทือนโดยติดตั้งโพรบไว้ด้านบนที่บริเวณกึ่งกลางของก้อนมวล วัดค่าความเร่ง (Acceleration) ของ $\ddot{X}_2$ ในรูปแบบของโดเมนเวลาในช่วงทรานเซียน (Transient) ได้ดังรูปที่ ฉ-4



รูปที่ ฉ-4 สัญญาณความเร่ง $\ddot{X}_2$ ในช่วงทรานเซียนในรูปแบบโดเมนเวลา

สามารถหาค่าความหน่วง c_2 โดยคำนวณจากการลดลงแบบลอการิทึม (Logarithmic Decrement) ของเส้นกราฟ เช่นเดียวกับการหาค่า c_1 ดังนั้น จากรูปที่ ฉ-4 ค่า n = 10, x_0 = 0.0669 Gs และ x_n = 0.0496 Gs คำนวณจากสมการ ฉ-1 ได้ค่าอัตราส่วนความหน่วง $\zeta_2 = 0.0047$

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นาย เอกพงษ์ ตั้งพัฒนเสรี เกิดเมื่อวันที่ 22 พฤศจิกายน พ.ศ. 2520 ที่กรุงเทพมหานคร ศึกษาในระดับมัธยมศึกษาตอนต้นและมัธยมศึกษาตอนปลายที่โรงเรียนบดินทรเดชา (สิงห์ สิงหเสนี) จากนั้นเข้าศึกษาในคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2537 สำเร็จปริญญาวิศวกรรมศาสตรบัณฑิต จากภาควิชาวิศวกรรมเครื่องกลในปีการศึกษา 2540 และได้เข้าศึกษาต่อในหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2541