รายการอ้างอิง

ภาษาไทย

กิดานันท์ มลิทอง. 2536. เทคโนโลยีการศึกษาร่วมสมัย. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร: บริษัท เอดิสัน เพรส โพรดักส์ จำกัด

บัณฑิต โรจน์อารยานนท์. 2536. หลักการไฟฟ้าสื่อสาร. พิมพ์ครั้งที่ 3. กรุงเทพมหานคร: สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย

ประสิทธิ์ ประพิณมงคลการ. 2533. หลักการระบบสื่อสาร. กรุงเทพมหานคร: บริษัท ซีเอ็ดยูเคชั่น จำกัด

ภาษาอังกฤษ

Burrus, C.S., McClellan, J.H., Oppenheim, A.V., Parks, T.W., Schafer, R.W., and Schuessler, H.W. 1994. Computer-based exercises for signal processing using MATLAB. New Jersey: Prentice-Hall, Inc.

Charls View Software, Inc. 1995. Microsoft Internet Assistant for Word 1.0z [Computer program] Seattle: Microsoft Co.

Couch, L.W. 1993. Digital and analog communication systems. 4th ed. New York: Macmillan Publishing Co.

Gibson, J.D. 1993. Principles of digital and analog communications. 2nd ed. New York: Macmillan Publishing Co.

Little, J.N. and Shure, L. 1993 Signal Processing Toolbox User's Guide. Mass: The MathWorks, Inc.

MATLAB Notebook Suite User's Guide. 1994. Mass: The MathWorks, Inc.

MATLAB Reference Guide. 1992. Mass: The MathWorks, Inc.

Mortensen, R.E. 1987. Random signals and systems. Singapore: John Wiley & Sons, Inc.

Oppenheim, A.V. and Schafer, R.W. 1989. Discrete-time signal processing. New Jersey: Prentice-Hall, Inc.

Proakis, J.G. 1989. Digital Communications. 2nd ed. Singapore: McGraw-Hill Book Co.

Shanmugam, K.S. 1985. Digital and analog communication systems. Singapore: John Wiley & Sons, Inc.

Zeytinoglu, O.M. and Ma, N.w. 1991. Communication System Toolbox [Computer program]. Toronto: Department of Electrical and Computer Engineering, Ryerson Polytechnic University.

ภาคผนวก

ภาคผนวก ก

ทูลบอกซ์ด้านระบบสื่อสาร

ความนำ

ภาคผนวก ก นี้ แสดงบัญชีรายชื่อและความหมายของฟังก์ชันต่างๆ ของทูลบ็อกซ์ด้านระบบสื่อสารที่ได้รับการพัฒนาโดยภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัย Ryerson Polytechnic ประเทศแคนาดา

ทูลบอกซ์ด้านระบบสื่อสาร

1. การสร้างตัวเลขอย่างสุ่ม (Random number generation)

binary	สร้างตัวเลขไบนารีแบบสุ่ม
corr_seq	สร้างตัวแปรสุ่มจากกระบวนการสุ่มแบบออโต้รีเกรสซีฟอันดับหนึ่ง (First order auto-regressive process)
exponent	สร้างตัวแปรสุ่มที่มีการกระจายแบบเอ็กซโปเนนเชียล
gauss	สร้างตัวแปรสุ่มที่มีการกระจายแบบเกาส์เซียน
laplace	สร้างตัวแปรสุ่มที่มีการกระจายแบบลาปลาเซียน
uniform	สร้างตัวแปรสุ่มที่มีการกระจายแบบยูนิฟอร์ม
realize	สร้างกระบวนการสุ่มไซน์โดยสุ่มค่าเฟส
speech	สร้างสัญญาณเสียงแบบสุ่ม

2. การวิเคราะห์ความน่าจะเป็น

cdf	คำนวณฟังก์ชัน cdf ของตัวแปรสุ่ม
exp_cdf	คำนวณและพล็อตฟังก์ชัน cdf ของตัวแปรสุ่มแบบเอ็กซโปเนนเชียล
exp_pdf	คำนวณและพล็อตฟังก์ชัน pdf ของตัวแปรสุ่มแบบเอ็กซโปเนนเชียล
gamma_pdf	คำนวณและพล็อตฟังก์ชัน pdf ของตัวแปรสุ่มแบบแกมมา
gaus_cdf	คำนวณและพล็อตฟังก์ชัน cdf ของตัวแปรสุ่มแบบเกาส์เซียน
gaus_pdf	คำนวณและพล็อตฟังก์ชัน pdf ของตัวแปรสุ่มแบบเกาส์เซียน
lapl_cdf	คำนวณและพล็อตฟังก์ชัน cdf ของตัวแปรสุ่มแบบลาปลาเซียน
lapl_pdf	คำนวณและพล็อตฟังก์ชัน pdf ของตัวแปรสุ่มแบบลาปลาเซียน
meansq	หาค่าเฉลี่ยกำลังสองของกำลังสัญญาณ
pdf	หาค่าและพล็อตฟังก์ชัน pdf ของตัวแปรสุ่ม
q_function	ฟังก์ชัน Q
rayl_cdf	คำนวณและพล็อตฟังก์ชัน cdf ของตัวแปรสุ่มแบบเรเลห์
rayl_pdf	คำนวณและพล็อตฟังก์ชัน pdf ของตัวแปรสุ่มแบบเรเลห์

unif_cdf	คำนวณและพล็อตฟังก์ชัน cdf ของตัวแปรสุ่มแบบยูนิฟอร์ม
unif_pdf	คำนวณและพล็อตฟังก์ชัน cdf ของตัวแปรสุ่มแบบยูนิฟอร์ม
var	หาค่าแวเรียนส์ของตัวแปรสุ่ม

3. เกมที่เกี่ยวข้องกับทฤษฎีความน่าจะเป็นและกระบวนการสุ่ม

dice	การทดลองโยนลูกเต๋า
dart	การทดลองปาเป้า
guess	เกมทายข้อมูลส่วนตัวของบุคคลที่ถูกสุ่มมา
integral	การอินติเกรตด้วยการจำลองสถานการณ์แบบ Monte-Carlo
new_born	ฟังก์ชันแสดงจำนวนการเกิดของทารก
person_data	ฟังก์ชันสร้างข้อมูลส่วนตัวของแต่ละบุคคล
temperature	ฟังก์ชันแสดงอุณหภูมิในแต่ละวัน

4. ฟังก์ชันใช้ในการวิเคราะห์ทั่วไป

acf	หาค่าฟังก์ชันออโตคอร์รีเลชัน (Autocorrelation function)
acf_plot	พล็อตกราฟของฟังก์ชันออโตคอร์รีเลชัน
ecorr	หาค่าฟังก์ชันออนซอมเบิลออโตคอร์รีเลชัน (Ensemble autocorrelation function)
psd	หาค่าฟังก์ชันความหนาแน่นสเปกตรัมกำลัง (Power spectral density function)
psd_plot	พล็อตกราฟของฟังก์ชันความหนาแน่นสเปกตรัมกำลัง

5. ควอนไตเซชัน

a2d	การแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล
d2a	การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก
mu_inv	การขยายสัญญาณโดยใช้กฎ μ - law
mu_law	การอัดสัญญาณโดยใช้กฎ μ - law
quant_ch	แสดงคุณลักษณะของควอนไตเซอร์
quantize	การควอนไตซ์สัญญาณแบบยูนิฟอร์ม

6. การประมวลผลข้อมูลไบนารี

bcd	การเข้ารหัสแบบ BCD (Binary Code Decimal)
bin_enc	การเข้ารหัสเป็นสัญญาณไบนารีแบบ Natural
bin_dec	การถอดรหัสสัญญาณไบนารีแบบ Natural

bin2gray	การแปลงรหัสแบบ Natural เป็นรหัสแบบ Gray
gray2bin	การแปลงรหัสแบบ Gray เป็นรหัสแบบ Natural
bin2pol	การแปลงรูปแบบสัญญาณไบนารีเป็นสัญญาณโพลาร์
bin2bipo	การแปลงรูปแบบสัญญาณไบนารีเป็นสัญญาณไบโพลาร์
diff_dec	การถอดรหัสแบบดิฟเฟอเรนเชียล (Differential Decoding)
diff_enc	การเข้ารหัสแบบดิฟเฟอเรนเชียล
invert	1's complement ของลำดับไบนารี
par2ser	การแปลงสัญญาณอนุกรมเป็นสัญญาณแบบขนาน
xor	Exclusive OR

7. *รูปแบบการส่งสัญญาณไบนารี*

manchester	สร้างไลน์โค้ดแบบ Manchester
rect_nrz	สร้างไลน์โค้ดแบบ rectangular NRZ
rect_rz	สร้างไลน์โค้ดแบบ rectangular RZ
triangle	สร้างไลน์โค้ดแบบ triangle
nyquist	สร้างไลน์โค้ดแบบ Nyquist pulse
nyq_gen	สร้างรูปคลื่นแบบ Nyquist
duob_gen	สร้างรูปคลื่นแบบ duobinary
duobinar	สร้างพัลส์แบบ duobinary หรือ modified duobinary
modulate	การมอดูเลตสัญญาณดิจิตอล (ASK, BPSK, FSK)
osc	ออสซิลเลเตอร์แบบไซน์ (Sinusoidal oscillator)
vco	Voltage Controlled Oscillator
wave_gen	กำเนิดรูปคลื่นสัญญาณไบนารี
waveplot	พล็อตสัญญาณไบนารี

8. *การส่งผ่านข้อมูล*

bpf	ฟิลเตอร์ผ่านแถบความถี่ (Band-pass filter)
channel	จำลองช่องสัญญาณในระบบสื่อสาร
eye_diag	สร้างและแสดงแพทเทินรูปดวงตา
lpf	ฟิลเตอร์ผ่านความถี่ต่ำ (Low-pass filter)
rc	ฟิลเตอร์วงจร RC ออร์เดอร์ 1 (1 st order RC-filter)
detect	การตรวจจับสัญญาณไบนารี
envelope	เครื่องตรวจจับซองสัญญาณ
int_dump	ฟิลเตอร์อินติเกรตและดัมพ์ (Integrate and dump filter)

mixer	2 input mixer
match	แมทซ์ฟิลเตอร์
rx	ฟังก์ชันเครื่องรับสัญญาณ
tx	ฟังก์ชันเครื่องส่งสัญญาณ

9. ฟังก์ชันอรรถประโยชน์

blackbox	ฟิลเตอร์ที่ไม่ทราบค่าออร์เดอร์และแบนด์วิดท์
convert	แปลงอาร์เรย์ของ MATLAB ไปเป็นไฟล์เสียงของ SUN
exp5_c6	คำนวณกำลังสัญญาณและ SQNR
fftsize	หาค่าจำนวนจุดในการทำ FFT
fx_menu	ฟังก์ชันตัวอย่างที่ใช้อินดิเกรต
play	กำเนิดเสียงจากอะเรย์ของ MATLAB (MEX-file)
playback	กำเนิดเสียงจากไฟล์เสียง
limiter	ฟังก์ชันลิมิเตอร์
normalize	นอร์มอไลซ์ข้อมูลอินพุท
rectify	เรกติไฟสัญญาณอินพุท
sinc	ฟังก์ชันซิงค์ (Sinc function)
spec_est	การประมาณค่าสเปกตรัมสัญญาณ
stat_plot	พล็อตไดอะแกรมสแกตเตอร์ (Scatter diagram)

ภาคผนวก ข

รายละเอียดมาโครของชุดทดลองด้านระบบสื่อสารแบบดิจิตอล

ความนำ

ในภาคผนวกนี้ เป็นการเสนอรายละเอียดของมาโครของชุดทดลองด้านระบบสื่อสารแบบดิจิตอล โดยรายละเอียดของมาโครแสดงเพียงส่วนที่ได้ปรับปรุงดัดแปลงจากเทมเพลท m-book.dot และมาโครส่วนที่เขียนขึ้นใหม่เท่านั้น

รายละเอียดมาโครของชุดทดลอง

1. มาโคร AutoOpen

```
' ************************************************************************
'       AutoOpen
'       Macro run whenever an existing Notebook document is opened
' ************************************************************************
Sub MAIN
        MatlabStartup
        RecalcDocument("AutoInit")
        '--- initialize "did we ask if user wants to close MATLAB?"
        SetDocumentVar("AskedForShutdown", "no")
End Sub

'++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
' This part is used for solving the memory problem when using with
' Microsoft Internet Assistance
'++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'       MatlabRecalc
'       Recalc the document for all cells
Declare Function IsAppLoaded Lib "kernel"(name$) As Integer \
  Alias "GetModuleHandle"

Sub RecalcDocument(cellType$)
        '--- Check that MATLAB is running
        If IsAppLoaded("matlab.exe") = 0 Then
                MatlabStartup
        End If

        '--- recalc all input and autoinit cells in document
        '--- don't recalc if there are no cells
        If CountBookmarks() = 0 Then
                Goto fin
        End If

        '--- set types to recalc
        If cellType$ = "all" Then
                type1$ = "INPUT"
                type2$ = "AUTOINIT"
        Else
                type1$ = UCase$(cellType$)
                type2$ = UCase$(cellType$)
        End If

        '--- Save the selection positions
        selStart = GetSelStartPos()
        selEnd = GetSelEndPos()
```

```
        ' loop through all the bookmarks and recalc the input cells
        WaitCursor 1
        markTotal = CountBookmarks() - 1
        Dim marks$(markTotal)
        For i = 0 To markTotal
                marks$(i) = BookmarkName$(i + 1)
        Next i
        For i = 0 To markTotal
                found1 = InStr(1, marks$(i), type1$)
                found2 = InStr(1, marks$(i), type2$)
                If found1 > 0 Or found2 > 0 Then
                        MatlabEval.EvalInputCell(marks$(i))
                        ScreenRefresh
                End If
        Next i
        WaitCursor 0

fin:

'--- Put the selection back
SetSelRange selStart, selEnd

End Sub
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

' *******************************************************************
'                            End of Macro
' *******************************************************************
```

2. มาโคร MatlabEval

```
' ************************************************************************
'       MatlabEval
'       Sends current input chunk to MATLAB for evaluation.
'       Creates new output chunk for result.
' ************************************************************************
Declare Function IsAppLoaded Lib "kernel"(name$) As Integer \
  Alias "GetModuleHandle"

Sub MAIN
'--- Setup error handler
On Error Goto handleError

'--- Check that MATLAB is running
If IsAppLoaded("matlab.exe") = 0 Then
       MatlabStartup
End If

'--- Find the input cell for the current insertion point
'--- Find all input and autoinit cells in the current selection
bFound = 0
inputTotal = 0
markTotal = CountBookmarks() - 1
If markTotal >= 0 Then
       Dim marks$(markTotal)
       For i = 0 To markTotal
              inputMark$ = BookmarkName$(i + 1)
              test = CmpBookmarks("\sel", inputMark$)
              If (InStr(1, inputMark$, "INPUT") > 0 Or \
                     InStr(1, inputMark$, "AUTOINIT") > 0) And \
                     (test = 0 Or (test >= 3 And test <= 8)) Then
                            marks$(inputTotal) = inputMark$
                            inputTotal = inputTotal + 1
                            bFound = 1
              End If
       Next
End If

If bFound = 1 Then
       '--- Evaluate the selected cells
       For i = 0 To inputTotal - 1
              '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
              ' Change Inputcell style afterrun Input -> Input run
               Mark$ = LTrim$(marks$(i))
               EditGoTo Mark$
               FormatStyle .Name = "Input run"
              '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
              EvalInputCell(marks$(i))
       Next
Else
       '--- The selection is not an existing input cell,
       '--- so create one and evaluated it
       currCellNum = InsertInputCell.InputCellFromSelection("Input")
       If currCellNum = - 1 Then
              Goto fin
       End If
       inputMark$ = "INPUT_" + LTrim$(Str$(currCellNum))
       If Not ExistingBookmark(inputMark$) Then
              '--- found an autoinit cell
              inputMark$ = "AUTOINIT_" + LTrim$(Str$(currCellNum))
       End If
       EvalInputCell(inputMark$)
End If

EditGoTo "\StartOfSel"
Goto fin

'--- Error handling needs work
handleError:
Print "Error: Notebook programmer error"
Beep
```

```
fin:
End Sub

' ************************************************************************
'       EvalInputCell
'       Given the name of an input cell, evaluate it.
' ************************************************************************
Sub EvalInputCell(inputMark$)

nl$ = Chr$(13)

currCellNum = Val(Right$(inputMark$, Len(inputMark$) - \
                        InStr(1, inputMark$, "_")))

'--- Send input to MATLAB for evaluation
EvalInput:
DDETerminateAll

chan = DDEInitiate("MATLAB", "Engine")
Print "MATLAB computing..."
WaitCursor 1

'--- Set the format to compact for this routine's purposes
DDEExecute(chan, "format compact;")

'--- Execute input cell
cmd$ = PrepMultilineInput$(GetBookmark$(inputMark$))
DDEExecute(chan, cmd$)

'--- Request results from MATLAB
StrResult$ = DDERequest$(chan, "EngStringResult")
FigResult$ = DDERequest$(chan, "EngFigureResult")
EndOfLine(0)

'--- surpress screen updates while output cell is generated
ScreenUpdating 0

'--- If there is no output from MATLAB, remove
'--- existing output cell if there is one and exit
outputMark$ = "OUTPUT_" + LTrim$(Str$(currCellNum))
If Len(StrResult$) = 0 And FigResult$ = "no" Then
        '--- If there is no output remove output cell
        If ExistingBookmark(outputMark$) Then
                EditGoTo .Destination = outputMark$
                If EmptyBookmark("\sel") = 0 Then
                        EditClear
                Else
                        EditBookmark .Name = outputMark$, .Delete
                End If
        End If
        Goto CloseAndExit
End If

'--- Check for MATLAB error message
If InStr(StrResult$, "??? ") <> 0 Then
        FormatName$ = "Error"
Else
        FormatName$ = "Output"
End If

'--- Strip off return in first line
If Left$(strResult$, 1) = nl$ Then \
        StrResult$ = Right$(strResult$, Len(strResult$) - 1)

'--- And rip all the returns from the last line
While Right$(strResult$, 1) = nl$
        StrResult$ = Left$(StrResult$, Len(StrResult$) - 1)
Wend

'--- Find corresponding output cell
'--- Create a new output cell if there is not one
'--- save the format style to follow the output cell
If Not ExistingBookmark(outputMark$) Then
```

```
oldStyle$ = StyleName$()'save style
'--- if we're at the end of the document or bumping against
'--- another cell, insert a new paragraph
test = CmpBookmarks("\sel", "\EndOfDoc")
If test = 0 Then
        InsertPara
        FormatStyle .Name = "Normal"
End If
EditGoTo .Destination = inputMark$
ParaDown

'--- Put blank lines before and after the output cell
If GetDocumentVar$("MatlabFmtStyle") = "Loose" Then
        If GetText$(GetSelStartPos() - 1, GetSelStartPos()) = nl$ Then
                CharLeft
                Insert nl$
                CharRight
        Else
                Insert nl$
        End If
        CharLeft 1, 1
        ResetChar()
        FormatStyle .Name = "Normal"
        CharRight
        CharRight 1, 1
        If Selection$() <> nl$ Then
                CharLeft
                If GetText$(GetSelStartPos() - 1, \
                                GetSelStartPos()) = nl$ Then
                        CharLeft
                        Insert nl$
                        CharLeft 1, 1
                        ResetChar()
                        FormatStyle .Name = "Normal"
                        CharRight 2
                Else
                        Insert nl$
                        CharLeft 1, 1
                        ResetChar()
                        FormatStyle .Name = "Normal"
                End If
        End If
        CharLeft
End If

If InsertInputCell.IsNestedInput("\sel") > 0 Then
        CharLeft
        InsertPara
        EditGoTo "\para"
        ResetChar()
        GoBack
        InsertedParagraph = 1
End If

'--- Insert the text and figure result to the output cell
FormatStyle .Name = FormatName$
SetStartOfBookmark "\sel", outputMark$
Insert StrResult$
If FigResult$ = "yes" Then
        InsertEmbeddedFigure(StrResult$)
        CharRight
End If
Insert "  "'two spaces
CharLeft
ExtendSelection
EditGoTo outputMark$
CopyBookmark "\Sel", outputMark$

'--- Add an extra space after output cell because it makes
'--- it easier to insert free form text between cells.
EditGoTo "\EndOfSel"
Cancel
CharRight
Insert nl$
```

```
        CharRight
        CharLeft 3, 1
        ResetChar()
        FormatStyle .Name = "Normal"
        EditGoTo "\EndOfSel"
        CharLeft
        If InsertedParagraph = 1 Then
                CharRight 1, 1
                EditClear
        End If
Else
        '--- Replace the existing output cell

        '--- For loose formatting, add a blank line after the input cell
        '--- For compact formatting, delete the blank line, if necessary
        EditGoTo .Destination = outputMark$
        CharLeft
        CharLeft 2, 1
        If GetDocumentVar$("MatlabFmtStyle") = "Loose" And \
                Selection$() <> nl$ + nl$ Then
                CharRight
                CharLeft
                Insert nl$
        ElseIf GetDocumentVar$("matlabFmtStyle") = "Compact" And \
                        Selection$() = nl$ + nl$ Then
                CharRight
                CharLeft 1, 1
                EditClear
        End If

        EditGoTo .Destination = outputMark$
        CharLeft 1, 1
        FormatStyle .Name = FormatName$
        Insert StrResult$
        If FigResult$ = "yes" Then
                InsertEmbeddedFigure(StrResult$)
        End If
        CharRight 1, 1
        If Selection$() = nl$ And FormatName$ <> "Error" Then EditClear
        EndOfLine 0
        CharRight 2, 1
        If GetDocumentVar$("matlabFmtStyle") = "Loose" And \
                        Selection$() <> nl$ + nl$ Then
                CharLeft
                Insert nl$
        End If
End If

'--- restore screen updates and cursor
ScreenUpdating 0
WaitCursor 0

'--- Close connection to MATLAB
CloseAndExit:
Print ""
DDETerminate(chan)
Goto fin

'--- Error handling needs work
handleError:
Print "Error: MATLAB is not running"
Beep

fin:
End Sub

' ********************************************************************
' Function for replacing newlines in a multiline input cell
' to be commas.  The special case of [ ] operators which require
' semicolons, is handled.
' All this because MATLAB's evalString cannot handle newlines.
'
' Function returns the converted string. Called by MatlabEval.
' ********************************************************************
```

```
Function PrepMultilineInput$(cmd$)
        variable$ =
"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_])"
        nl$ = Chr$(13)
        comma$ = ","
        semi$ = ";"
        comments = 0
        quotes = 0

        pos = InStr(1, cmd$, nl$)
        If pos = 0 Then
                PrepMultilineInput$ = cmd$
        Else
                pos = 1

                While pos <= Len(cmd$)
                        curChar$ = Mid$(cmd$, pos, 1)
                        If inVariable > 0 Then
                                If curChar$ = "'" Then
                                        inVariable = 0
                                        out$ = out$ + "'"
                                ElseIf InStr(variable$, curChar$) = 0 Then
                                        inVariable = 0
                                        pos = pos - 1
                                Else
                                        out$ = out$ + curChar$
                                End If
                                If curChar$ = "]" Then brackets = brackets - 1
                        ElseIf comments > 0 Then
                                If curChar$ = nl$ Then
                                        comments = 0
                                        If brackets > 0 Then
                                                out$ = out$ + ";"
                                        Else
                                                out$ = out$ + ","
                                        End If
                                End If
                        ElseIf quotes > 0 Then
                                If curChar$ = nl$ Then
                                        ' Reprocess the newline
                                        quotes = 0
                                        pos = pos - 1
                                ElseIf curChar$ = "'" Then
                                        quotes = 0
                                        out$ = out$ + curChar$
                                Else
                                        out$ = out$ + curChar$
                                End If
                        ElseIf elipsis > 0 Then
                                If elipsis = 3 Then
                                        If curChar$ = nl$ Then
                                                elipsis = 0
                                        End If
                                Else
                                        If curChar$ = "." Then
                                                elipsis = elipsis + 1
                                        Else
                                                If elipsis = 2 Then
                                                        out$ = out$ + ".."
                                                Else
                                                        out$ = out$ + "."
                                                End If
                                                elipsis = 0
                                                '--- Reprocess curChar$
                                                pos = pos - 1
                                        End If
                                End If
                        Else
                                If InStr(variable$, curChar$) <> 0 Then
inVariable = 1

                                If curChar$ = "[" Then
                                        brackets = brackets + 1
                                        out$ = out$ + curChar$
```

```
                        ElseIf curChar$ = "]" Then
                                If brackets <= 0 Then
                                        ' MsgBox "Error: too many ]'s"
                                        brackets = 0
                                Else
                                        brackets = brackets - 1
                                End If
                                out$ = out$ + curChar$
                        ElseIf curChar$ = "'" Then
                                quotes = 1
                                out$ = out$ + curChar$
                        ElseIf curChar$ = "%" Then
                                comments = 1
                        ElseIf curChar$ = "." Then
                                elipsis = elipsis + 1
                        ElseIf curChar$ = nl$ Then
                                If brackets > 0 Then
                                        out$ = out$ + ";"
                                Else
                                        out$ = out$ + ","
                                End If
                        Else
                                out$ = out$ + curChar$
                        End If
                End If
                pos = pos + 1

        Wend
        PrepMultilineInput$ = out$
    End If
End Function

' ************************************************************************
' InsertEmbeddedFigure
'
'       Insert the embedded figure by pasting the contents of the
'       clipboard and then format the picture to the current size set in
'       Notebook Options.
' ************************************************************************
Sub InsertEmbeddedFigure(StrResult$)
    nl$ = Chr$(13)

    If Len(StrResult$) > 0 Then Insert nl$
    EditPaste
    CharLeft 1, 1
    units$ = GetDocumentVar$("MATLABFigUnits")
    '--- get current figure sizes (converted to points)
    sizeXpts = Val(NotebookOptions.InchesToAny$(\
                        NotebookOptions.AnyToInches$(\
                        GetDocumentVar$("MATLABFigWidth"), units$), "
Points"))
    sizeYpts = Val(NotebookOptions.InchesToAny$(\
                        NotebookOptions.AnyToInches$(\
                        GetDocumentVar$("MATLABFigHeight"), units$), "
Points"))
    FormatPicture .SetSize = 1, .SizeX = sizeXpts, .SizeY = sizeYpts

    SetFocusToWord
End Sub

' ************************************************************************
' SetFocusToWord
' ************************************************************************
Sub SetFocusToWord
    '--- set focus back to Word
    WordName$ = "Microsoft Word ÀÒËÒä·Â"
    If DocMaximize() Then
        WordName$ = WordName$ + " - " + WindowName$()
    End If
    AppActivate WordName$, 1
End Sub

' ************************************************************************
' Subroutine for sending a string to MATLAB to be
```

```
' evaluated.
' ************************************************************************
Sub EvalString(command$)
On Error Goto handleError
DDETerminateAll
chan = DDEInitiate("MATLAB", "Engine")
Print "MATLAB computing..."
DDEExecute(chan, command$)
Print ""
DDETerminate(chan)
Goto fin

handleError:
Print "Error: MATLAB is not running"
Beep

fin:
End Sub

' ************************************************************************
' Function for sending a string to MATLAB to be
' evaluated.  Returns the string result from MATLAB.
' ************************************************************************
Function EvalWithResult$(command$)
On Error Goto handleError
DDETerminateAll
chan = DDEInitiate("MATLAB", "Engine")

Print "MATLAB computing..."
DDEExecute(chan, command$)
EvalWithResult$ = DDERequest$(chan, "EngStringResult")
Print ""

DDETerminate(chan)
Goto fin

handleError:
Print "Error: MATLAB is not running"
Beep
EvalWithResult$ = ""

fin:
End Function

' ************************************************************************
'                          End of Macro
' ************************************************************************
```

3. *มาโคร MatlabStartup*

```
' ************************************************************************
'       MatlabStartup
'       Starts matlab.exe and initializes settings for the Notebook
'       in Word and MATLAB.
' ************************************************************************
Declare Function IsAppLoaded Lib "kernel"(name$) As Integer \
  Alias "GetModuleHandle"

Sub MAIN
        rc = ExecuteMatlab
        If (rc = 1) Then
                WaitCursor 1

                '--- initialize Word settings for Notebook
                '--- (MATLAB does not like smart quotes)
                ToolsAutoCorrectSmartQuotes(0)
                ToolsOptionsAutoFormat .ReplaceQuotes = 0
                ToolsOptionsView .BookMarks = 1
                '--- turn off WordPerfect Navigation keys because of
                '--- Ctrl+Enter conflict
                ToolsOptionsGeneral .WPDocNavKeys = 0
                ViewToolbars .Toolbar = "[illegible]", .Show

                '--- initialize Notebook Options settings
                '--- settings are saved in document vars
                '--- sends commands to MATLAB
                 NotebookOptions.InitFromSavedSettings

'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
' This part is used for solving the memory problem when using with
' HYPERLINK (WIA)

' ************************************************************************
'       Initialize the Notebook options from the settings saved with
'       the document.
' ************************************************************************
'Sub InitFromSavedSettings

        '--- surpress screen updates, set cursor to hourglass
        ScreenUpdating 0
        WaitCursor 1

        '--- set settings for showing cell markers in the doc
        oldDirtyFlag = IsTemplateDirty()
        cellMarks = Val(GetDocumentVar$("ShowCellMarkers"))
        If cellMarks > 0 Then
                cellMarkMenuName$ = "Hide &Cell Markers"
        Else
                cellMarkMenuName$ = "Show &Cell Markers"
        End If
        ToolsOptionsView .BookMarks = cellMarks
        ToolsCustomizeMenus .Category = 2, .Name = "ShowHideCellMarks", .
Position = 9, .Menu = "Notebook", .MenuText = cellMarkMenuName$, .Rename, .
Context = 1
        If oldDirtyFlag = 0 Then
                SetTemplateDirty 0
                SetDocumentDirty 0
        End If

        '--- send format settings to MATLAB
        If Len(GetDocumentVar$("MATLABFmt")) > 0 And \
                Len(GetDocumentVar$("MATLABFmtStyle")) > 0 Then
                If LCase$(GetDocumentVar$("MATLABFmt")) = "plus" Then
                  cmd$ = "format +;"
                Else
                  cmd$ = "format " + LCase$(GetDocumentVar$("MATLABFmt")) + ";"
                End If
                EvalString(cmd$)
        End If
```

```
        '--- send figure settings to MATLAB
        If Len(GetDocumentVar$("MATLABFigUnits")) > 0 And \
                Len(GetDocumentVar$("MATLABFigWidth")) > 0 And \
                Len(GetDocumentVar$("MATLABFigHeight")) > 0 Then
                If GetDocumentVar$("MATLABFigEmbed") = "YES" Then
                        cmd$ = "system_dependent(12,'on');"
                        whitebg$ = "whitebg(0,'white');"
                Else
                        cmd$ = "system_dependent(12,'off');"
                        whitebg$ = "whitebg(0,'black');"
                End If
                If GetDocumentVar$("MATLABFig16Color") = "YES" Then
                        cmd$ = cmd$ + "system_dependent(14,'on');"
                Else
                        cmd$ = cmd$ + "system_dependent(14,'off');"
                End If
                cmd$ = cmd$ + "set(0,'defaultfigureinverthardcopy','off');" + \
                             whitebg$ + "close all;"
                EvalString(cmd$)
        End If

        '--- set focus back to Word
        WordName$ = "Microsoft Word ÀÒŽÒä·Â"
        If DocMaximize() Then
                WordName$ = WordName$ + " - " + WindowName$()
        End If
        AppActivate WordName$, 1

        '--- restore screen updates, set cursor to normal
        ScreenUpdating 1
        WaitCursor 0

'End Sub
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                WaitCursor 0
        End If
End Sub

'******************************************************************
'       ExecuteMatlab
'       Starts the MATLAB executable.  Prompt user if there is no
'       startup directory in winword6.ini
'       Returns 1 for success 0 for failure
'******************************************************************
Function ExecuteMatlab
AppName$ = "matlab.exe"

If IsAppLoaded(AppName$) = 0 Then
        AppPath$ = GetPrivateProfileString$("MATLAB Notebook", \
                "MATLAB-PATH", "WINWORD6.INI")
        On Error Goto AbortStartup
        If Len(AppPath$) = 0 Then
                AppPath$ = "c:\matlab\bin"
                AppPath$ = InputBox$("Enter path for MATLAB", \
                        "MATLAB Startup", AppPath$)
                SetPrivateProfileString("MATLAB Notebook", \
                        "MATLAB-PATH", AppPath$, "WINWORD6.INI")
        End If

        WaitCursor 1
        Print "Starting MATLAB..."
        currPath$ = DefaultDir$(14)
        ChDir AppPath$
        Shell AppPath$ + "\" + AppName$, 0
        WordName$ = "Microsoft Word ÀÒŽÒä·Â"
        If DocMaximize() Then
                WordName$ = WordName$ + " - " + WindowName$()
        End If
        AppActivate WordName$, 1       'Switch back to Word.
        ChDir currPath$
        Print ""
        WaitCursor 0
Else
```

```
        Print "MATLAB is running"
End If
ExecuteMatlab = 1
Goto bye
AbortStartup:
ExecuteMatlab = 0
Beep
Print "Cannot start MATLAB"
bye:
End Function

' ************************************************************************
' Subroutine for sending a string to MATLAB to be
' evaluated.
'       ** This function is a duplicate of the one in MATLABEval macro
'       ** It must also be here because the NotebookOptions macro may be
'       ** called from MATLABEval, and Word doesn't allow circular calls.
'       ** (ie MATLABEval -> MATLABStartup ->
'       ** NotebookOptions.InitFromSavedSettings -> MATLABEval.EvalString)
'       ** This is for the pathelogical case where the user shuts down
'       ** MATLAB while the Notebook is running.
'       ** This the one place where DDE code exists outside of MATLABEval
' ************************************************************************
Sub EvalString(command$)
        On Error Goto handleError
        DDETerminateAll
        chan = DDEInitiate("MATLAB", "Engine")
        Print "MATLAB computing..."
        DDEExecute(chan, command$)
        Print ""
        DDETerminate(chan)
        Goto fin
handleError:
        Print "Error: MATLAB is not running"
        Beep
fin:
End Sub

' ************************************************************************
'               End of Macro
' ************************************************************************
```

4. *มาโคร NotebookOptions*

```
' ************************************************************************
'       NotebookOptions
'       Displays the Notebook Options dialog
'       handles callbacks for dialog's controls
' ************************************************************************
Dim Shared OldFigUnits$
Sub MAIN
Dim MLFormats$(7)
Dim FigUnits$(2)
MLFormats$(0) = "Short"
MLFormats$(1) = "Long"
MLFormats$(2) = "Hex"
MLFormats$(3) = "Bank"
MLFormats$(4) = "Plus"
MLFormats$(5) = "Short e"
MLFormats$(6) = "Long e"
MLFormats$(7) = "Rational"
FigUnits$(0) = "Inches"
FigUnits$(1) = "Centimeters"
FigUnits$(2) = "Points"

Begin Dialog UserDialog 286, 280, "Notebook Options", .OptionsCallback
        PushButton 43, 252, 88, 21, "&OK", .OKButt
        PushButton 146, 252, 88, 21, "&Cancel", .CancelButt
        GroupBox 11, 8, 262, 78, "Numeric Format"
        GroupBox 12, 97, 261, 128, "Figure Options"
        CheckBox 23, 111, 239, 16, "Embed Figures in M-book", .FigEmbedFlag
        CheckBox 23, 130, 186, 16, "Use 16-Color Figures", .Fig16ColorFlag
        DropListBox 21, 23, 160, 98, MLFormats$(), .MLFormatList
        OptionGroup  .FormatOptions
                OptionButton 21, 42, 131, 16, "Loose", .LooseFmt
                OptionButton 21, 59, 151, 16, "Compact", .CompactFmt
        Text 22, 154, 40, 13, "Units", .Text3
        Text 21, 178, 45, 13, "Width", .Text1
        Text 21, 202, 51, 13, "Height", .Text2
        DropListBox 88, 152, 160, 108, FigUnits$(), .FigUnits
        TextBox 88, 176, 160, 18, .FigWidth
        TextBox 88, 200, 160, 18, .FigHeight
End Dialog

Dim dlg As UserDialog
rc = Dialog(dlg)

CancelOptions:
End Sub

' ************************************************************************
'       Callback function for the Notebook Options dialog box
' ************************************************************************
Function OptionsCallback(CtlId$, Action, SuppVal)
Select Case Action
        '--- Initialize dialog box
        Case 1
        '--- init format drop down list
        currFmt$ = GetDocumentVar$("MATLABFmt")
        If Len(currFmt$) > 0 Then
                DlgText("MLFormatList", currFmt$)
        Else
                DlgText("MLFormatList", "Short")
        End If
        '--- init format option buttons
        If GetDocumentVar$("MATLABFmtStyle") = "Compact" Then
                DlgValue("FormatOptions", 1)
        Else
                DlgValue("FormatOptions", 0)
        End If
        '--- init figure options
        If GetDocumentVar$("MATLABFigEmbed") = "YES" Then
                DlgValue("FigEmbedFlag", 1)
                DlgEnable("Fig16ColorFlag", 1)
```

```
                DlgEnable("FigUnits", 1)
                DlgEnable("FigWidth", 1)
                DlgEnable("FigHeight", 1)
        Else
                DlgValue("FigEmbedFlag", 0)
                DlgEnable("Fig16ColorFlag", 0)
                DlgEnable("FigUnits", 0)
                DlgEnable("FigWidth", 0)
                DlgEnable("FigHeight", 0)
        End If

        If GetDocumentVar$("MATLABFig16Color") = "YES" Then
                DlgValue("Fig16ColorFlag", 1)
        Else
                DlgValue("Fig16ColorFlag", 0)
        End If

        '--- init fig units drop down list and
        '--- width and height
        currFmt$ = GetDocumentVar$("MATLABFigUnits")
        oldFigUnits$ = currFmt$
        If Len(currFmt$) > 0 Then
                DlgText("FigUnits", currFmt$)
        Else
                DlgText("FigUnits", "Inches")
        End If
        DlgText("FigWidth", GetDocumentVar$("MATLABFigWidth"))
        DlgText("FigHeight", GetDocumentVar$("MATLABFigHeight"))

        '--- Handle button press
        Case 2
        If (ctlId$ = "FigEmbedFlag") Then
                '--- toggle the enabled state of figure settings
                DlgEnable("Fig16ColorFlag")
                DlgEnable("FigUnits")
                DlgEnable("FigWidth")
                DlgEnable("FigHeight")
        End If
        If (ctlId$ = "FigUnits") Then
                AdjustFigWHforUnits
        End If

        If (ctlId$ = "OKButt") Then
                SetDocumentVar("MATLABFmt", DlgText$("MLFormatList"))

                If DlgValue("FormatOptions") = 1 Then
                        SetDocumentVar("MATLABFmtStyle", "Compact")
                Else
                        SetDocumentVar("MATLABFmtStyle", "Loose")
                End If
                If DlgValue("FigEmbedFlag") = 1 Then
                        SetDocumentVar("MATLABFigEmbed", "YES")
                Else
                        SetDocumentVar("MATLABFigEmbed", "NO")
                End If
                If DlgValue("Fig16ColorFlag") = 1 Then
                        SetDocumentVar("MATLABFig16Color", "YES")
                Else
                        SetDocumentVar("MATLABFig16Color", "NO")
                End If
                SetDocumentVar("MATLABFigUnits", DlgText$("FigUnits"))
                If Val(DlgText$("FigWidth")) > 0 Then
                        SetDocumentVar("MATLABFigWidth", DlgText$("FigWidth"))
                End If
                If Val(DlgText$("FigHeight")) > 0 Then
                        SetDocumentVar("MATLABFigHeight", DlgText$("FigHeight"))
                End If

                '--- send new settings to MATLAB
                InitFromSavedSettings
        End If

        Case Else
End Select
```

```
End Function

' ************************************************************************
'       Adjust the Figure Width and Height settings when the user
'       changes the units setting.
' ************************************************************************
Sub AdjustFigWHforUnits
        '--- adjust figure width
        tempInches$ = AnyToInches$(DlgText$("FigWidth"), \
                                        oldFigUnits$)
        DlgText("FigWidth", \
                        InchesToAny$(tempInches$, DlgText$("FigUnits")))
        '--- adjust figure height
        tempInches$ = AnyToInches$(DlgText$("FigHeight"), \
                                        oldFigUnits$)
        DlgText("FigHeight", \
                        InchesToAny$(tempInches$, DlgText$("FigUnits")))
        oldFigUnits$ = DlgText$("FigUnits")
End Sub
' ** Converts a value from other units to inches
' ** Returns new value
Function AnyToInches$(value$, units$)
        Select Case units$
        Case "Centimeters"
                AnyToInches$ = Str$(Val(value$) / 2.54)
        Case "Points"
                AnyToInches$ = Str$(Val(value$) / 72)
        Case Else
                AnyToInches$ = value$
        End Select
End Function
' ** Converts a value in inches to other units
' ** Returns new value
Function InchesToAny$(value$, units$)
        Select Case units$
        Case "Centimeters"
                InchesToAny$ = Str$(Val(value$) * 2.54)
        Case "Points"
                InchesToAny$ = Str$(Val(value$) * 72)
        Case Else
                InchesToAny$ = value$
        End Select
End Function

' ************************************************************************
'       Initialize the Notebook options from the settings saved with
'       the document.
' ************************************************************************
Sub InitFromSavedSettings

        '--- surpress screen updates, set cursor to hourglass
        ScreenUpdating 0
        WaitCursor 1

        '--- set settings for showing cell markers in the doc
        oldDirtyFlag = IsTemplateDirty()
        cellMarks = Val(GetDocumentVar$("ShowCellMarkers"))
        If cellMarks > 0 Then
                cellMarkMenuName$ = "Hide &Cell Markers"
        Else
                cellMarkMenuName$ = "Show &Cell Markers"
        End If
        ToolsOptionsView .BookMarks = cellMarks
        ToolsCustomizeMenus .Category = 2, .Name = "ShowHideCellMarks", .
Position = 9, .Menu = "Notebook", .MenuText = cellMarkMenuName$, .Rename, .
Context = 1
        If oldDirtyFlag = 0 Then
                SetTemplateDirty 0
                SetDocumentDirty 0
        End If

        '--- send format settings to MATLAB
        If Len(GetDocumentVar$("MATLABFmt")) > 0 And \
```

```
                Len(GetDocumentVar$("MATLABFmtStyle")) > 0 Then
                If LCase$(GetDocumentVar$("MATLABFmt")) = "plus" Then
                  cmd$ = "format +;"
                Else
                  cmd$ = "format " + LCase$(GetDocumentVar$("MATLABFmt")) + ";"
                End If
                EvalString(cmd$)
        End If

        '--- send figure settings to MATLAB
        If Len(GetDocumentVar$("MATLABFigUnits")) > 0 And \
                Len(GetDocumentVar$("MATLABFigWidth")) > 0 And \
                Len(GetDocumentVar$("MATLABFigHeight")) > 0 Then
                If GetDocumentVar$("MATLABFigEmbed") = "YES" Then
                        cmd$ = "system_dependent(12,'on');"
                        whitebg$ = "whitebg(0,'white');"
                Else
                        cmd$ = "system_dependent(12,'off');"
                        whitebg$ = "whitebg(0,'black');"
                End If
                If GetDocumentVar$("MATLABFig16Color") = "YES" Then
                        cmd$ = cmd$ + "system_dependent(14,'on');"
                Else
                        cmd$ = cmd$ + "system_dependent(14,'off');"
                End If
                cmd$ = cmd$ + "set(0,'defaultfigureinverthardcopy','off');" + \
                           whitebg$ + "close all;"
                EvalString(cmd$)
        End If

        '--- set focus back to Word
        WordName$ = "Microsoft Word ÀÒÉÒä·Â"
        If DocMaximize() Then
                WordName$ = WordName$ + " - " + WindowName$()
        End If
        AppActivate WordName$, 1

        '--- restore screen updates, set cursor to normal
        ScreenUpdating 1
        WaitCursor 0

End Sub

' ************************************************************************
' Subroutine for sending a string to MATLAB to be
' evaluated.
'       ** This function is a duplicate of the one in MATLABEval macro
'       ** It must also be here because the NotebookOptions macro may be
'       ** called from MATLABEval, and Word doesn't allow circular calls.
'       ** (ie MATLABEval -> MATLABStartup ->
'       ** NotebookOptions.InitFromSavedSettings -> MATLABEval.EvalString)
'       ** This is for the pathelogical case where the user shuts down
'       ** MATLAB while the Notebook is running.
'       ** This the one place where DDE code exists outside of MATLABEval
' ************************************************************************
Sub EvalString(command$)
        On Error Goto handleError
        DDETerminateAll
        chan = DDEInitiate("MATLAB", "Engine")
        Print "MATLAB computing..."
        DDEExecute(chan, command$)
        Print ""
        DDETerminate(chan)
        Goto fin

handleError:
        Print "Error: MATLAB is not running"
        Beep

fin:
End Sub
' ************************************************************************
'                       End of Macro
' ************************************************************************
```

5. มาโคร *ToggleOutput*

```
' ************************************************************************
'       Toggle Output
'       Toggle MATLAB Output Between MATLAB figure and WORD
' ************************************************************************

Sub MAIN
        Begin Dialog UserDialog 400, 112, "โหมดการแสดงผลทางกราฟฟิก", .
OptionsCallback
                PushButton 77, 78, 88, 21, "ตกลง", .OKButt
                PushButton 188, 79, 88, 21, "ยกเลิก", .OKCancel
                Text 80, 11, 270, 13, "เลือกโหมดการแสดงผลทางกราฟฟิก", .Text1
                OptionGroup .FigOptions
                        OptionButton 81, 31, 250, 16, "แสดงผลบนไฟล์เอกสาร WORD", .
FigEmbed
                        OptionButton 81, 48, 250, 16, "แสดงผลทางหน้าต่าง Figure", .
NoFigEmbed
        End Dialog

        Dim dlg As UserDialog
        ButtonPush = Dialog(dlg, - 1)

End Sub

Function OptionsCallback(CtlId$, Action, SuppVal)
Select Case Action
        Case 1  '--- Initial Options
                If GetDocumentVar$("MATLABFigEmbed") = "YES" Then
                        DlgValue("FigOptions", 0)
                Else
                        DlgValue("FigOptions", 1)
                End If
        Case 2  '--- Handle Button
                If (ctlId$ = "OKButt") Then
                        If DlgValue("FigOptions") = 0 Then
                                SetDocumentVar("MATLABFigEmbed", "YES")
                        Else
                                SetDocumentVar("MATLABFigEmbed", "NO")
                        End If
                        NoteBookOptions.InitFromSavedSettings
                End If
        Case Else
        End Select

End Function
' ************************************************************************
'                       End of Macro
' ************************************************************************
```

6. มาโคร ShowFigure

```
'*****************************************************
'  Search and Show Figure window in Figure Mode
'*****************************************************
Sub MAIN
        If AppIsRunning("Figure No. 1") Then
                If AppIsRunning("Figure No. 2") Then
                        AppActivate "Figure No. 2"
                Else
                        AppActivate "Figure No. 1"
                EndIf
        Else
                MsgBox "ไม่พบหน้าต่าง Figure", "เรียกหน้าต่าง Figure" , 48
        End If

End Sub
' ***********************************************************
'               End of Macro
' ***********************************************************
```

7. มาโคร LABBack

```
'*************************************************************
' Close current document without prompting and saving data
'*************************************************************
Sub MAIN
        DocClose 2
End Sub
```

ภาคผนวก ค

รายละเอียดไฟล์ .m ของชุดทดลองด้านระบบสื่อสารแบบดิจิตอล

ความนำ

ภาคผนวกนี้เสนอรายละเอียดของไฟล์ .m ของชุดทดลองด้านระบบสื่อสารแบบดิจิตอล ไฟล์ .m เหล่านี้นอกจากใช้ในชุดทดลองแล้ว ยังสามารถเรียกใช้งานได้ในสภาพแวดล้อมของโปรแกรม MATLAB ด้วย ในชุดทดลองด้านระบบสื่อสารแบบดิจิตอล เมื่อมีการเปิดไฟล์เอกสารสารบัญการทดลอง ก็จะมีการโหลดค่าตัวแปรที่ใช้ในแต่ละการทดลองโดยเรียกไฟล์ `initial?.m` โดย ? หมายถึงเลขการทดลองที่เท่าไหร่ รายละเอียดของไฟล์ได้กล่าวไว้ในส่วนถัดไป และเมื่อมีการเปิดไฟล์เอกสารการทดลองย่อย ก็จะมีการโหลดค่าตัวแปรที่ใช้ในแต่ละการทดลองย่อยเช่นกันโดยเรียกไฟล์ .m ซึ่งมีรูปแบบดังเช่น

`init5_1.m` หมายถึง โหลดตัวแปรที่ใช้ในการทดลองที่ 5.1 เป็นต้น

โดยมีรายละเอียดของไฟล์ `init5_1.m` ดังนี้

```
load lab5_1;
```

โดย `lab5_1` หมายถึงไฟล์ `lab5_1.mat` ที่เก็บข้อมูลต่างๆ ที่ใช้ในการทดลองที่ 5.1 สำหรับการทดลองอื่นๆ ก็จะมีลักษณะทำนองเดียวกัน

รายละเอียดไฟล์ .m ของชุดทดลอง

1. ไฟล์ initial1.m

```
% INITIAL  initialize global parameters used in lab1.
%       Somboon Pr. 21-Sep-95
START_OK = 1;
global START_OK;
SAMPLING_CONSTANT  = 10;
BINARY_DATA_RATE   = 1000;
SAMPLING_FREQ      = BINARY_DATA_RATE * SAMPLING_CONSTANT;

CARRIER_FREQUENCY  = [ 1000000 4000000 ];

NYQUIST_BLOCK   = 8;          % Number of blocks for Nyquist pulse generation
NYQUIST_ALPHA   = 0.5;        % Default value of "Excessive BW factor"
DUOBINARY_BLOCK = 8;          % Number of blocks for Duobinary pulse.

global START_OK;
global SAMPLING_CONSTANT;
global SAMPLING_FREQ;
global BINARY_DATA_RATE;
global CARRIER_FREQUENCY;
global NYQUIST_BLOCK;
global NYQUIST_ALPHA;
global DUOBINARY_BLOCK;

BELL    = 'fprintf(''\007\007\007'')';
WARNING = 'fprintf(''\n\t * NOT SUFFICIENT INPUT ARGUMENTS \t * USAGE:\n'')';

global BELL;
global WARNING;
```

ในกรณีของการทดลองอื่นๆ ไฟล์ initial?.m จะมีรายละเอียดเหมือนกันยกเว้นบรรทัดกำหนดค่า SAMPLING_CONSTANT ซึ่งในแต่ละการทดลองจะมีค่าต่างๆ กันดังนี้

การทดลองที่ 2 ใช้ไฟล์ initial2.m : `SAMPLING_CONSTANT = 100;`

การทดลองที่ 4 ใช้ไฟล์ initial4.m : `SAMPLING_CONSTANT = 8;`

การทดลองที่ 5 ใช้ไฟล์ initial5.m : `SAMPLING_CONSTANT = 10;`

การทดลองที่ 6 ใช้ไฟล์ initial6.m : `SAMPLING_CONSTANT = 10;`

การทดลองที่ 7 ใช้ไฟล์ initial7.m: `SAMPLING_CONSTANT = 100;`

การทดลองที่ 8 ใช้ไฟล์ initial8.m : `SAMPLING_CONSTANT = 40;`

สำหรับการทดลองที่ 3 ไม่มีเรียกไฟล์ initial?.m

2. **ไฟล์** calc_mp.m

```
%=============================================================
% Script file  'calc_mp'
% Calculate for mag and phase
% Output is     mag_z, mag_zr, mag_p, mag_pr
%               phase_z, phase_zr, phase_p, phase_pr
%=============================================================

% Somboon Pr.
% 14-May-96

zEmpty = isempty(Z_POINT);
zrEmpty = isempty(ZR_POINT);
pEmpty = isempty(P_POINT);
prEmpty = isempty(PR_POINT);
dim = fliplr(size(w));

if zEmpty & zrEmpty & pEmpty & prEmpty
        magz = zeros(dim);  phasez = zeros(dim);
        magp = ones(dim);   phasep = zeros(dim);

     elseif zEmpty & zrEmpty & pEmpty & ~prEmpty
        mag_z = ones(dim);   phase_z = zeros(dim);
        mag_zr = ones(dim);  phase_zr = zeros(dim);
        mag_p = ones(dim);   phase_p = zeros(dim);
        [mag_pr,phase_pr] = sfactor(PR_VALUE,w);
        magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
        magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

     elseif zEmpty & zrEmpty & prEmpty & ~pEmpty
        mag_z = ones(dim);   phase_z = zeros(dim);
        mag_zr = ones(dim);  phase_zr = zeros(dim);
        mag_pr = ones(dim);  phase_pr = zeros(dim);
        [mag_p,phase_p] = sfactor(P_VALUE,w);
        magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
        magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

    elseif zEmpty & pEmpty & prEmpty & ~zrEmpty
        mag_z = ones(dim);   phase_z = zeros(dim);
        mag_p = ones(dim);   phase_p = zeros(dim);
        mag_pr = ones(dim);  phase_pr = zeros(dim);
        [mag_zr,phase_zr] = sfactor(ZR_VALUE,w);
        magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
        magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

    elseif zrEmpty & pEmpty & prEmpty & ~zEmpty
        mag_zr = ones(dim);   phase_zr = zeros(dim);
        mag_p = ones(dim);    phase_p = zeros(dim);
        mag_pr = ones(dim);   phase_pr = zeros(dim);
        [mag_z,phase_z] = sfactor(Z_VALUE,w);
        magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
```

```
    magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

elseif zEmpty & zrEmpty & ~pEmpty & ~prEmpty
    mag_z = ones(dim);   phase_z = zeros(dim);
    mag_zr = ones(dim);  phase_zr = zeros(dim);
    [mag_p,phase_p] = sfactor(P_VALUE,w);
    [mag_pr,phase_pr] = sfactor(PR_VALUE,w);
    magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
    magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

elseif pEmpty & prEmpty & ~zEmpty & ~zrEmpty
    mag_p = ones(dim);   phase_p = zeros(dim);
    mag_pr = ones(dim);  phase_pr = zeros(dim);
    [mag_z,phase_z] = sfactor(Z_VALUE,w);
    [mag_zr,phase_zr] = sfactor(ZR_VALUE,w);
    magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
    magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

elseif zEmpty & pEmpty & ~zrEmpty & ~prEmpty
    mag_z = ones(dim);   phase_z = zeros(dim);
    mag_p = ones(dim);   phase_p = zeros(dim);
    [mag_zr,phase_zr] = sfactor(ZR_VALUE,w);
    [mag_pr,phase_pr] = sfactor(PR_VALUE,w);
    magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
    magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

elseif zrEmpty & prEmpty & ~zEmpty & ~pEmpty
    mag_zr = ones(dim);  phase_zr = zeros(dim);
    mag_pr = ones(dim);  phase_pr = zeros(dim);
    [mag_z,phase_z] = sfactor(Z_VALUE,w);
    [mag_p,phase_p] = sfactor(P_VALUE,w);
    magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
    magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

elseif zEmpty & prEmpty & ~zrEmpty & ~pEmpty
    mag_z = ones(dim);   phase_z = zeros(dim);
    mag_pr = ones(dim);  phase_pr = zeros(dim);
    [mag_zr,phase_zr] = sfactor(ZR_VALUE,w);
    [mag_p,phase_p] = sfactor(P_VALUE,w);
    magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
    magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

elseif zrEmpty & pEmpty & ~zEmpty & ~prEmpty
    mag_zr = ones(dim);  phase_zr = zeros(dim);
    mag_p = ones(dim);   phase_p = zeros(dim);
    [mag_z,phase_z] = sfactor(Z_VALUE,w);
    [mag_pr,phase_pr] = sfactor(PR_VALUE,w);
    magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
    magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

elseif zEmpty & ~zrEmpty & ~pEmpty & ~prEmpty
    mag_z = ones(dim);   phase_z = zeros(dim);
    [mag_zr,phase_zr] = sfactor(ZR_VALUE,w);
    [mag_p,phase_p] = sfactor(P_VALUE,w);
    [mag_pr,phase_pr] = sfactor(PR_VALUE,w);
    magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
    magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

elseif zrEmpty & ~zEmpty & ~pEmpty & ~prEmpty
    mag_zr = ones(dim);  phase_zr = zeros(dim);
    [mag_z,phase_z] = sfactor(Z_VALUE,w);
    [mag_p,phase_p] = sfactor(P_VALUE,w);
    [mag_pr,phase_pr] = sfactor(PR_VALUE,w);
    magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
    magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

elseif pEmpty & ~prEmpty & ~zEmpty & ~zrEmpty
    mag_p = ones(dim);   phase_p = zeros(dim);
    [mag_z,phase_z] = sfactor(Z_VALUE,w);
    [mag_zr,phase_zr] = sfactor(ZR_VALUE,w);
    [mag_pr,phase_pr] = sfactor(PR_VALUE,w);
    magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
    magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];
```

```
   elseif prEmpty & ~pEmpty & ~zEmpty & ~zrEmpty
        mag_pr = ones(dim);   phase_pr = zeros(dim);
        [mag_z,phase_z] = sfactor(Z_VALUE,w);
        [mag_zr,phase_zr] = sfactor(ZR_VALUE,w);
        [mag_p,phase_p] = sfactor(P_VALUE,w);
        magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
        magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];


   else
        [mag_z,phase_z] = sfactor(Z_VALUE,w);
        [mag_zr,phase_zr] = sfactor(ZR_VALUE,w);
        [mag_p,phase_p] = sfactor(P_VALUE,w);
        [mag_pr,phase_pr] = sfactor(PR_VALUE,w);
        magz = [ mag_z, mag_zr ];  phasez = [ phase_z, phase_zr ];
        magp = [ mag_p, mag_pr ];  phasep = [ phase_p, phase_pr ];

end
```

3. ไฟล์ exp4_5.m

```
function [out] = exp4_5(power)

% EXP4_5
%
%       EXP4_5( POWER ) performs the procedure outlined in step C6 of
%               experiment at the POWER level and returns the following vector
%               as a result:
%
%               [ signal_power    SQNR(uniform)      SQNR(non-uniform) ];
%
%               All answers are in dBW.

%       AUTHORS : M. Zeytinoglu & N. W. Ma
%              Department of Electrical & Computer Engineering
%              Ryerson Polytechnic University
%              Toronto, Ontario, CANADA
%
%       DATE    : August 1991.
%       VERSION : 1.0

%=======================================================================
% Modifications history:
% ----------------------
%       o   Added "checking"  11.30.1992 MZ
%       o       Tested (and modified) under MATLAB 4.0/4.1 08.16.1993 MZ
%       o   Modified by Somboon Pr.  13-May-96
%=======================================================================

global START_OK;
global BELL;
global WARNING;

check;

%-------------------------------------------------------------------------
%       Input parameter control
%-------------------------------------------------------------------------
if (nargin ~= 1)
   error(eval('eval(BELL),eval(WARNING),help exp6_4'));
   return;
end

no_pow = length(power);
if ( ~isempty(find(power<0)) )
   error('Power values must be a positive.');
end
out = zeros(no_pow,3);

%-------------------------------------------------------------------------
%       Perform required operations
%-------------------------------------------------------------------------
```

```
if( nargout == 0 )
        disp('                                          '),   ...
        disp('  SIGMAs^2    SQNR(un)   SQNR(mu-law)    '),   ...
        disp('------------------------------------------')
end

for ii = 1:no_pow

        s   = limiter(laplace(1000,power(ii)),-1,1); % Original sequence
        sq  = quantize(s,8);                    % Uniform quantized sequence
        msq = mu_inv(quantize(mu_law(s),8)); % Non-uniform quantized sequence

        sigma2_s   = var(s);                            % Variance of "s"
        sigma2_sq  = var(s-sq);                         % Variance of "sq"
        sigma2_msq = var(s-msq);                        % Variance of "msq"

        snr_unif   = 10*log10( sigma2_s / sigma2_sq );       % SQNR(uniform)
        snr_nunif  = 10*log10( sigma2_s / sigma2_msq );      % SQNR(non-uniform)

        out(ii,:)   = [ 10*log10(var(s)), snr_unif, snr_nunif ];

end
```

4. ไฟล์ findconj.m

```
function [out,change]  = findconj(in)
% Find Conjugate
% [ itsconjugate,changeStatus] = findconj(clicked_point)
% chnageStatus =       1 -> Z change
%              2 -> ZR change
%              3 -> P change
%              4 -> PR change

% Somboon Pr.
% 13-May-96

global Z_POINT
global ZR_POINT
global P_POINT
global PR_POINT

Z_Idx = find(Z_POINT == in);          ZR_Idx = find(ZR_POINT == in);
P_Idx = find(P_POINT == in);          PR_Idx = find(PR_POINT == in);
Zchange = ~isempty(Z_Idx);            ZRchange = ~isempty(ZR_Idx);
Pchange = ~isempty(P_Idx);            PRchange = ~isempty(PR_Idx);
change = [ Zchange ZRchange Pchange PRchange ];
if sum(change) == 0
        error('Function findconj has error !!!!')
end
change = find(change);

if  Zchange
        if rem(Z_Idx,2) == 0
                out = Z_POINT(Z_Idx - 1);
        else
                out = Z_POINT(Z_Idx + 1);
        end
elseif  ZRchange
        out = in;
elseif  Pchange
        if rem(P_Idx,2) == 0
                out = P_POINT(P_Idx - 1);
        else
                out = P_POINT(P_Idx + 1);
        end
else
        % PRchange
        out = in;

end
```

5. ไฟล์ getpoint.m

```
function out = getpoint()
% GetPoint
% This function returns a handle of object that is clicked.
% point_clicked = getpoint;

% Somboon Pr.
% 15-May-96

keypress = 1;
set(gcf,'pointer','crosshair');
while keypress == 1,
        keypress = waitforbuttonpress;
end
set(gcf,'pointer','arrow');
out = gco;
```

6. ไฟล์ he.m

```
function out = he(v,n)
%=============================================
% Histogram Equalization
% out = HE(v,n)
%               v = random input vector.
%               n = number of bins.
%=============================================

%=====================
% Generate CDF function.
%=====================
[y,x] = mycdf(v,n);

%=================================================
% Prepair Data for create function piecewise linear CDF
%=================================================
xx = [min(v) x(10:10:40) max(v)];
yy = [0 y(10:10:40) 1];

%=========================================
% Piecewise linear approximation of CDF.
% 5 lines.
%=========================================
for i = 1:5
        A(i) = (yy(i+1)-yy(i))/(xx(i+1)-xx(i));
        B(i) = yy(i)-A(i)*xx(i);
end

%=====================
% Piecewise Mapping
%=====================
index1 = find( (v>=xx(1))&(v<=xx(2)) );
index2 = find( (v>xx(2))&(v<=xx(3)) );
index3 = find( (v>xx(3))&(v<=xx(4)) );
index4 = find( (v>xx(4))&(v<=xx(5)) );
index5 = find( (v>xx(5))&(v<=xx(6)) );

out = v;
out(index1) = A(1)*v(index1)+B(1);
out(index2) = A(2)*v(index2)+B(2);
out(index3) = A(3)*v(index3)+B(3);
out(index4) = A(4)*v(index4)+B(4);
out(index5) = A(5)*v(index5)+B(5);
```

7. ไฟล์ hist_eq.m

```
function out = hist_eq(in);
% newPic = Hist_Eq(Pic)
% Histogram Equalization
%       Pic     =  input image matrix
%       newPic  =  output image matrix

% Require file - he.m
%              - openfile.m

% Somboon Pr.
% 28-Feb-96

in = in(:)';
temp = he(in,50);
temp = 256*temp;        % scale to 256 gray scale

%=========================================
% Translate vector (temp) to matrix200x125 (out)
%=========================================
dataID = fopen('temp.$$$','wb');
fwrite(dataID,temp,'integer*1');
out = openfile('temp.$$$');
```

8. ไฟล์ lpfilter.m

```
function xout = lpfilter(xin,f);
% Low pass filter
% x_reconstruct = lpfilter(x_sampled,fcutoff)

% Somboon Pr.
% 12-Jun-96

global fsim;

if (nargin ~= 1) & (nargin ~= 2)
        error('Invalid numbers of argument')
end

if nargin == 1
        fcut = 2*xin/fsim;
        [b,a] = cheby2(9,60,fcut);
        [h,w] = freqz(b,a,1024);
        semilogy(w/pi*1/2*fsim,abs(h)),grid
else
        fcut = 2*f/fsim;
        [b,a] = cheby2(9,60,fcut);
%       [h,w] = freqz(b,a,1024);
        xout = filter(b,a,xin);
end
```

9. ไฟล์ mycdf.m

```
function [co,xo] = mycdf(v,n)
% MyCDF(n,x)    Compute or Display CDF.
%               v = random variable input vector.
%               n = number of bins.

[n,x] = mypdf(v,n);
deltaX = x(2)-x(1);
c = cumsum(n)*deltaX;

if nargout == 0
        bar(x,c)
        axis([min(x) max(x) 0 1])
else
        xo = x;
        co = c;
end
```

10. ไฟล์ mypdf.m

```
function [no,xo] = mypdf(y,x)
%MyPDF  Plot pdf histogram.
%       Modify from hist.m
%       Make for  DSP book.

if nargin == 1
    x = 10;
end
if min(size(y))==1, y = y(:); end
[m,n] = size(y);
if max(size(x)) == 1
    miny = min(min(y));
    maxy = max(max(y));
    binwidth = (maxy - miny) ./ x;
    xx = miny + binwidth*[0:x];
    xx(length(xx)) = maxy;
    x = xx(1:length(xx)-1) + binwidth/2;
else
    xx = x(:)';
    miny = min(min(y));
    maxy = max(max(y));
    binwidth = [diff(xx) 0];
    xx = [xx(1)-binwidth(1)/2 xx+binwidth/2];
    xx(1) = miny;
    xx(length(xx)) = maxy;
end
nbin = max(size(xx));
nn = zeros(nbin,n);
for i=2:nbin
    nn(i,:) = sum(y <= xx(i));
end
nn = nn(2:nbin,:) - nn(1:nbin-1,:);

%=================================
% Modify for display pdf function
%=================================
factor = nn*binwidth;
nn = nn/sum(factor);
%-------------------------------------------------------

if nargout == 0
    bar(x,nn);
else
  if min(size(y))==1, % Return row vectors if possible.
    no = nn';
    xo = x;
  else
    no = nn;
    xo = x';
  end
end
```

11. ไฟล์ mystem.m

```
function [out1,out2] =  mystem(x,y,linetype)
%       Copyright (c) 1984-94 by The MathWorks, Inc.
%       Modify by Somboon Pr.
%       20-May-96

n = length(x);
if nargin == 1
        y = x(:)';
        x = 1:n;
        linetype = '-';
elseif nargin == 2
        if isstr(y)
                linetype = y;
                y = x(:)';
                x = 1:n;
        else
                x = x(:)';
```

```
                y = y(:)';
                linetype = '-';
        end
elseif nargin == 3
        x = x(:)';
        y = y(:)';
end
xx = [x;x;nan*ones(size(x))];
yy = [zeros(1,n);y;nan*ones(size(y))];
out1 = xx(:);
out2 = yy(:);
```

12. ไฟล์ openfile.m

```
function out = openfile(fname);
% Out = OpenFile('fname')
%       Out     =  Output Matrix
%       fname   =  image file's name

% 28-Feb-96

picID = fopen(fname,'rb');
out = fread(picID,[200,125],'uchar'); % for only 256color.raw
fclose(picID);                  % not complete
```

13. ไฟล์ plotf.m

```
function   plotf( xa, dt )
%PLOTF   Plot  Fourier Transform (Mag) of "ANALOG" signal
%   Usage:   plotf( xa, dt )
%       xa :   "ANALOG" signal
%       dt :    sampling interval for the simulation of xa(t)
%
L = length(xa);
Nfft = round( 2 .^ nextpow2(5*L) );
Xa = fft(xa,Nfft);
range = 0:(Nfft/4);
ff = range/Nfft/dt;
plot( ff/1000, abs( Xa(1+range) ) )
title('CONT-TIME FOURIER TRANSFORM (MAG)')
xlabel('FREQUENCY (kHz)'), grid
```

14. ไฟล์ psd_est.m

```
function [fre_out,psd_out] = psd_est(x,fstart,fstop,bw);
% PSD_EST ...Spectral estimation.
%
%       PSD_EST(X) estimates the power spectral density finction of the
%               input sequence X, without explicitly computing its Fourier
%               transform.  The input sequence X will be successively fed into
%               a center-frequency adjustable, band-pass filter with user
%               specified bandwidth.  The frequency interval and the filter
%               bandwidth are interactively specified.
%
%       [f,Px] = PSD_EST(X) returns the mean-square value of the sequence X
%               at the output of a band-pass filter centered at f.

%       AUTHORS : M. Zeytinoglu & N. W. Ma
%              Department of Electrical & Computer Engineering
%              Ryerson Polytechnic University
%              Toronto, Ontario, CANADA
%
%       DATE    : August 1991.
%       VERSION : 1.0
% Modify by Somboon Pr.
% 4-Jun-96

global START_OK;
global SAMPLING_CONSTANT;
global SAMPLING_FREQ;
global BINARY_DATA_RATE;
global BELL;
```

```
global WARNING;

check;

%---------------------------------------------------------------------------
%       Check input parameters
%---------------------------------------------------------------------------

if (nargin ~= 4)
   error(eval('eval(BELL),eval(WARNING),help psd_est'));
   return;
end

%---------------------------------------------------------------------------
%       Default values
%---------------------------------------------------------------------------

fs2 = SAMPLING_FREQ/2;
min_fractional_bw = 200/5000;
min_fractional_Hz = min_fractional_bw * fs2;

norm_factor = sqrt( length(x) );

f_range = [ fstart fstop ];
f_range = sort(f_range);
f_start = min(f_range);
f_stop  = max(f_range);

if ( f_start < 0 )
   fprintf(' WARNING: Start frequency should be non-negative.\n');
   fprintf('          Resetting to 0 Hz.\n');
   f_start = 0;
end

if ( f_stop > fs2 )
   fprintf(' WARNING: Stop frequency must be less than %10.2f Hz.\n',fs2);
   fprintf('          Resetting to %10.2f Hz.\n',fs2);
   f_stop = fs2;
end

if ( bw < 0 )
   error('BPF Bandwidth must be positive.');
end
if ( bw < min_fractional_Hz )
   fprintf('   WARNING:   BPF   bandwidth   must   be   larger   than   %8.2f
Hz:\n',min_fractional_Hz);
   fprintf('          otherwise the BPF will be unstable.\n');
   fprintf('                              Resetting   BPF   bandwidth   to   %10.2f
Hz.\n',min_fractional_Hz);
   bw = min_fractional_Hz;
end

freq_length = f_stop - f_start;
no_bpf      = fix(freq_length/bw);

if ( no_bpf <= 0 )
   error('FREQ_START, FREQ_STOP, and BANDWIDTH values are not compatible.');
end

range = [f_start, f_start+bw];

for k = 1:no_bpf

    if ( min(range) == 0 )                              % LOW-PASS FILTER'
      range_true = max(range);
      f(k) = sum(range)/2;
      range_true = range_true/fs2;
      [b a] = butter(6,range_true);
      y(k) = meansq(filter(b,a,x))/norm_factor;
    elseif ( max(range) == fs2 )                        % HIGH-PASS FILTER
      range_true = min(range);
      f(k) = sum(range)/2;
      range_true = range_true/fs2;
      [b a] = butter(6,range_true,'high');
```

```
      y(k) = meansq(filter(b,a,x))/norm_factor;
    else                                              % BAND-PASS FILTER
      f(k) = sum(range)/2;
      range_true = range/fs2;
      [b a] = butter(6,range_true);
      y(k) = meansq(filter(b,a,x))/norm_factor;
    end

    range = range + bw;

    fprintf(' o PSD estimate at %10.2f [Hz] is %10.2e [W].\n',f(k),y(k));

end

semilogy(f/1000,y), xlabel('Frequency [kHz]'); ylabel('Power [W]');

if (nargout ~= 0)
   freq_out = f;
   psd_out  = y;
end
```

15. ไฟล์ pulsetra.m

```
function pulsetrain(action,in1,in2);
%PULSETRAIN Interactive signal demo - 1 : Periodic Pulse and its fourier series
%       and som property.

% Somboon Pr.
% 16-Sep-96

%       possible actions:
%       'start'
%       'btnWdec'
%       'btnWinc'
%       'btnFdec'
%       'btnFinc'
%       'redraw'
%       'done'

if nargin<1,
        action='start';
end;

global PULSETRAIN_DAT f_env

if strcmp(action,'start'),

        %=================================================
        % Graphics initialization
        oldFigNumber = watchon;
        figNumber = figure;
        set(gcf, ...
                'NumberTitle','off', ...
                'Name','สัญญาณรายคาบและอนุกรมฟูเรียร์', ...
                'backingstore','off',...
                'Units','normalized');

         %=================================================
         % Information for all buttons
        labelColor=192/255*[1 1 1];
        top=0.95;
        bottom=0.05;
        left=0.775;
        yInitLabelPos=0.90;
        eft = 0.78;
        labelWid=0.18;
        labelHt=0.05;
        btnWid = 0.18;
        btnHt=0.07;
        % Spacing between the label and the button for the same command
        btnOffset=0.003;
        % Spacing between the button and the next command's label
        spacing=0.05;
```

```
        %=========================================
        % The CONSOLE frame
        frmBorder=0.02;
        yPos=0.05-frmBorder;
        frmPos=[left-frmBorder yPos btnWid+2*frmBorder 0.9+2*frmBorder];
        h=uicontrol( ...
                'Style','frame', ...
                'Units','normalized', ...
                'Position',frmPos, ...
                'BackgroundColor',[0.5 0.5 0.5]);

        %=========================================
        % Label and button for pulse width change.
        btnNumber=1;
        yLabelPos=top-(btnNumber-1)*(btnHt+labelHt+spacing);
        labelPos=[left yLabelPos-labelHt labelWid labelHt];
        uicontrol( ...
                'Style','text', ...
                'Units','normalized', ...
                'Position',labelPos, ...
                'BackgroundColor',labelColor, ...
                'HorizontalAlignment','left', ...
                'String',' ความกว้างพัลส์');

        btnPos=[left          yLabelPos-labelHt-btnHt-btnOffset          btnWid/2
btnHt*2/3];
        btnWdec = uicontrol( ...
                'Style','Pushbutton', ...
                'Units','normalized', ...
                'Position',btnPos, ...
                'Callback',''pulsetrain(''btnWdec'')', ...
                'string','<--');
        btnPos=[left+btnWid/2     yLabelPos-labelHt-btnHt-btnOffset     btnWid/2
btnHt*2/3];
        btnWinc = uicontrol( ...
                'Style','Pushbutton', ...
                'Units','normalized', ...
                'Position',btnPos, ...
                'Callback',''pulsetrain(''btnWinc'')', ...
                'string','-->');

        %=========================================
        % Label and button for Fundamental Freq. change.
        btnNumber=2;
        yLabelPos=top-(btnNumber-1)*(btnHt+labelHt+spacing);
        labelPos=[left     yLabelPos-labelHt     labelWid     labelHt];
        freq_text = uicontrol( ...
                'Style','text', ...
                'Position', labelPos, ...
                'Units','normalized', ...
                'BackgroundColor',labelColor, ...
                'String',' ความถี่มูลฐาน');

        btnPos  =  [left       yLabelPos-labelHt-btnHt-btnOffset          btnWid/2
btnHt*2/3];
        btnFdec = uicontrol( ...
                'Style','Pushbutton', ...
                'Units','normalized', ...
                'Position',btnPos, ...
                'Callback',''pulsetrain(''btnFdec'')', ...
                'string','<--');
        btnPos=[left+btnWid/2     yLabelPos-labelHt-btnHt-btnOffset     btnWid/2
btnHt*2/3];
        btnFinc = uicontrol( ...
                'Style','Pushbutton', ...
                'Units','normalized', ...
                'Position',btnPos, ...
                'Callback',''pulsetrain(''btnFinc'')', ...
                'string','-->');

        %=========================================
        % The Envelope button
        btnEnvelope = uicontrol( ...
```

```
        'Style','push', ...
        'Units','normalized', ...
        'Position',[left bottom+(labelHt)+spacing btnWid 4/3*labelHt],
...
        'String','การรบกวนปกติ', ...
        'Callback','pulsetrain(''btnEnvelope'')');

%=============================================
% The CLOSE button
done_button=uicontrol('Style','Pushbutton', ...
        'Position',[left bottom btnWid 4/3*labelHt], ...
        'Units','normalized','Callback',...
        'pulsetrain(''done'')','String','จบการทำงาน');

%=============================================
% Create intial signal
cycle = 5;
To = 32;        fo = 1/To;
pulse_width = 8;

pulse = [ ones(1,pulse_width) zeros(1,To-pulse_width) ];
pulse_amp = abs( fftshift( fft( pulse ) ) );
pulse_amp = [ pulse_amp pulse_amp(1) ];
pulse_amp = pulse_amp/max(pulse_amp)*(fo*pulse_width);

f = (-To/2:To/2)/To;

% Prepair for Display
ppulse = pulse'*ones(1,cycle);
ppulse = ppulse(:)';
[xx,yy] = stair(ppulse);

% Create Time Axes
ax_time=axes('Position',[.12 .58 .6 .3] );
time_line=plot(xx,yy,'EraseMode','xor');
axis([0 cycle*To -0.5 1.5]);
xlabel('Time (Seconds)');
ylabel('Waveform');

% Create Freq. Axes
ax_freq=axes('Position',[.12 .14 .6 .3]);
[xx,yy] = mystem(f,pulse_amp);
freq_line=plot(xx,yy,'EraseMode','xor');
ylabel('Fourier Series');
xlabel('Frequency (Hertz)');
hold on

% Create Envelope
T = 1024;               % use for plot envelope
f_env = (-T/2:T/2)/T;
envelope = abs( fftshift( fft( pulse,T ) ) );
envelope = [ envelope envelope(1) ];
envelope = envelope/max(envelope)*(fo*pulse_width);
env_line = plot(f_env,envelope,'w','EraseMode','xor','visible','off');

PULSETRAIN_DAT = [ cycle; To; pulse_width; time_line; freq_line;
btnEnvelope; env_line ];

watchoff(oldFigNumber);

elseif strcmp(action,'btnWdec'),
        cycle = PULSETRAIN_DAT(1);
        To = PULSETRAIN_DAT(2);
        pulse_width = PULSETRAIN_DAT(3);
        time_line = PULSETRAIN_DAT(4);
        freq_line = PULSETRAIN_DAT(5);
        env_line = PULSETRAIN_DAT(7);

        if pulse_width == 0
                return;
        end

        fo = 1/To;
        pulse_width = pulse_width - 1;
```

```
        pulse = [ ones(1,pulse_width) zeros(1,To-pulse_width) ];
        pulse_amp = abs( fftshift( fft( pulse ) ) );
        pulse_amp = [ pulse_amp pulse_amp(1) ];
        pulse_amp = pulse_amp/max(pulse_amp)*(fo*pulse_width);
        % Prepair for Display
        ppulse = pulse'*ones(1,cycle);
        ppulse = ppulse(:)';
        [x,y] = stair(ppulse);
        set(time_line,'YData',y,'XData',x);

        f = (-To/2:To/2)/To;
        [xx,yy] = mystem(f,pulse_amp);
        set(freq_line,'YData',yy,'XData',xx);

        % Create Envelope
        T = 1024;               % use for plot envelope
        envelope = abs( fftshift( fft( pulse,T ) ) );
        envelope = [ envelope envelope(1) ];
        envelope = envelope/max(envelope)*(fo*pulse_width);
        set(env_line,'YData',envelope);

        PULSETRAIN_DAT(3) = pulse_width;

elseif strcmp(action,'btnWinc'),
        cycle = PULSETRAIN_DAT(1);
        To = PULSETRAIN_DAT(2);
        pulse_width = PULSETRAIN_DAT(3);
        time_line = PULSETRAIN_DAT(4);
        freq_line = PULSETRAIN_DAT(5);
        env_line = PULSETRAIN_DAT(7);

        if pulse_width == To
                return;
        end

        fo = 1/To;
        pulse_width = pulse_width +1;
        pulse = [ ones(1,pulse_width) zeros(1,To-pulse_width) ];
        pulse_amp = abs( fftshift( fft( pulse ) ) );
        pulse_amp = [ pulse_amp pulse_amp(1) ];
        pulse_amp = pulse_amp/max(pulse_amp)*(fo*pulse_width);
        % Prepair for Display
        ppulse = pulse'*ones(1,cycle);
        ppulse = ppulse(:)';
        [x,y] = stair(ppulse);
        set(time_line,'YData',y,'XData',x);

        f = (-To/2:To/2)/To;
        [xx,yy] = mystem(f,pulse_amp);
        set(freq_line,'YData',yy,'XData',xx);

        % Create Envelope
        T = 1024;               % use for plot envelope
        envelope = abs( fftshift( fft( pulse,T ) ) );
        envelope = [ envelope envelope(1) ];
        envelope = envelope/max(envelope)*(fo*pulse_width);
        set(env_line,'YData',envelope);

        PULSETRAIN_DAT(3) = pulse_width;

elseif strcmp(action,'btnFinc'),
        cycle = PULSETRAIN_DAT(1);
        To = PULSETRAIN_DAT(2);
        pulse_width = PULSETRAIN_DAT(3);
        time_line = PULSETRAIN_DAT(4);
        freq_line = PULSETRAIN_DAT(5);
        env_line = PULSETRAIN_DAT(7);

        if To == 0
                return;
        end

        To = To - 2;
        fo = 1/To;
```

```
        pulse = [ ones(1,pulse_width) zeros(1,To-pulse_width) ];
        pulse_amp = abs( fftshift( fft( pulse ) ) );
        pulse_amp = [ pulse_amp pulse_amp(1) ];
        pulse_amp = pulse_amp/max(pulse_amp)*(fo*pulse_width);

        % Prepair for Display
        ppulse = pulse'*ones(1,cycle);
        ppulse = ppulse(:)';

        % àµÀÒÂÁ¡ÃÒ¿ãËéàµçÁ range 0-160
        if To < 32
                addpulse = ceil( (160 - max(size(ppulse)))/To );
                ppulse = pulse'*ones(1,cycle+addpulse);
                ppulse = ppulse(:)';
        else
                addpulse = 0;
        end
        [x,y] = stair(ppulse);
        set(time_line,'YData',y,'XData',x);

        f = (-To/2:To/2)/To;
        [xx,yy] = mystem(f,pulse_amp);
        set(freq_line,'YData',yy,'XData',xx);

        % Create Envelope
        T = 1024;               % use for plot envelope
        envelope = abs( fftshift( fft( pulse,T ) ) );
        envelope = [ envelope envelope(1) ];
        envelope = envelope/max(envelope)*(fo*pulse_width);
        set(env_line,'YData',envelope);

        PULSETRAIN_DAT(1) = cycle+addpulse;
        PULSETRAIN_DAT(2) = To;

elseif strcmp(action,'btnFdec'),
        cycle = PULSETRAIN_DAT(1);
        To = PULSETRAIN_DAT(2);
        pulse_width = PULSETRAIN_DAT(3);
        time_line = PULSETRAIN_DAT(4);
        freq_line = PULSETRAIN_DAT(5);
        env_line = PULSETRAIN_DAT(7);

        if To == 160
                return;
        end

        To = To + 2;
        fo = 1/To;
        pulse = [ ones(1,pulse_width) zeros(1,To-pulse_width) ];
        pulse_amp = abs( fftshift( fft( pulse ) ) );
        pulse_amp = [ pulse_amp pulse_amp(1) ];
        pulse_amp = pulse_amp/max(pulse_amp)*(fo*pulse_width);
        % Prepair for Display
        ppulse = pulse'*ones(1,cycle);
        ppulse = ppulse(:)';

        % เตรียมกราฟให้เต็ม range 0-160
        if To < 32
                addpulse = ceil( (160 - max(size(ppulse)))/To );
                ppulse = pulse'*ones(1,cycle+addpulse);
                ppulse = ppulse(:)';
        else
                addpulse = 0;
        end
        [x,y] = stair(ppulse);
        set(time_line,'YData',y,'XData',x);

        f = (-To/2:To/2)/To;
        [xx,yy] = mystem(f,pulse_amp);
        set(freq_line,'YData',yy,'XData',xx);

        % Create Envelope
        T = 1024;               % use for plot envelope
```

```
        envelope = abs( fftshift( fft( pulse,T ) ) );
        envelope = [ envelope envelope(1) ];
        envelope = envelope/max(envelope)*(fo*pulse_width);
        set(env_line,'YData',envelope);

        PULSETRAIN_DAT(1) = cycle+addpulse;
        PULSETRAIN_DAT(2) = To;

elseif strcmp(action,'btnEnvelope'),
        cycle = PULSETRAIN_DAT(1);
        To = PULSETRAIN_DAT(2);
        pulse_width = PULSETRAIN_DAT(3);
        time_line = PULSETRAIN_DAT(4);
        freq_line = PULSETRAIN_DAT(5);
        btnEnvelope = PULSETRAIN_DAT(6);
        env_line = PULSETRAIN_DAT(7);

        check = get(env_line,'visible');
        if check(1:2) == 'of'
                set(env_line,'visible','on');
                set(btnEnvelope,'String','ลบรอบคลื่นกลับ')
        else
                set(env_line,'visible','off');
                set(btnEnvelope,'String','วาดรอบคลื่นกลับ')
        end

elseif strcmp(action,'done'),
        % close the figure window that is showing the window fnction:
        close(gcf);
        clear global PULSETRAIN_DAT
end
```

16. ไฟล์ rp1.m

```
function v = rp1(M,N);
a = 0.02;
b = 5;
Mc = ones(M,1)*b*sin((1:N)*pi/N);
Ac = a*ones(M,1)*[1:N];
v = (rand(M,N)-0.5).*Mc + Ac;
```

17. ไฟล์ rp2.m

```
function v = rp2(M,N);
Ar = rand(M,1)*ones(1,N);
Mr = rand(M,1)*ones(1,N);
v = (rand(M,N)-0.5).*Mr + Ar;
```

18. ไฟล์ rp3.m

```
function v = rp3(M,N);
a = 0.5;
m = 3;
v = (rand(M,N)-0.5)*m + a;
```

19. ไฟล์ sampling.m

```
function out = sampling(x,fs);
% Sampling signal x with sampling freq. fs Hz.
% x_sampled = sampling(x,fs)
% Somboon Pr.
% 10-June-96

global fsim;

m = length(x);
interval = fsim/fs;
xs = x(1:interval:m);
out = zeros(1,m);
out(1:interval:m) = xs;
```

20. ไฟล์ sfactor.m

```
function [mag,phase] = sfactor(a,w)
% Compute magnitude and phase of factor
%       [ s - a ]
% [mag,phase] = sfactor(s,w)

% Somboon Pr.
% 24-Apr-96

dim = size(w(:));
wm = w(:)*ones(1,max(size(a)));  am = ones(dim)*a;
h = j*wm - am;

mag = abs(h);
phase = unwrap(angle(h));
```

21. ไฟล์ sfunctio.m

```
function sfunction(action);
% Sfunction  Interactive frequency response of a single zero or pole.
% This file require    - sfactor.m
%               - calc_mp.m
%               - findconj.m
%               - getpoint.m

% Somboon Pr.
% 14-May-96

if nargin<1,
    action='start';
end;

global SFUNCTION_DAT
global Z_POINT
global Z_VALUE
global ZR_POINT
global ZR_VALUE
global P_POINT
global P_VALUE
global PR_POINT
global PR_VALUE
global w

if strcmp(action,'start'),

        %========== สร้างหน้าต่าง S plane==========
        figure1     =     figure('Visible','off','units','normalized','position',
[0.0038 0.1300 0.4262 0.7983]);
        set(figure1, ...
                'NumberTitle','off', ...
                'Name','S Plane', ...
                'MenuBar','none', ...
                'backingstore','off');
        axis([-10 10 -10 10]),axis('equal'),grid
        hold on

        %========== สร้างเมนู Add and Delete Zeros Poles ==========
        menuAdd =uimenu(figure1,'Label','เพิ่มจำนวน Zero/Pole');
                addZero = uimenu(menuAdd,'Label','เพิ่ม Zero');
                labelList = str2mat('Zero บนแกนจำนวนจริง', 'Zeros ที่คอนจูเกต');
                cmdList = str2mat('sfunction(''addzr'')','sfunction(''addz'')');
                setmenu(addZero,labelList,cmdList);
                addPole = uimenu(menuAdd,'Label','เพิ่ม Pole');
                labelList = str2mat('Pole บนแกนจำนวนจริง', 'Poles ที่คอนจูเกต');
                cmdList = str2mat('sfunction(''addpr'')','sfunction(''addp'')');
                setmenu(addPole,labelList,cmdList);
        menuDel =uimenu(figure1,'Label','ลดจำนวน Zero/Pole');
                delZero = uimenu(menuDel,'Label','ลด Zero');
                labelList = str2mat('Zero บนแกนจำนวนจริง','Zeros ที่คอนจูเกต');
                cmdList = str2mat('sfunction(''delzr'')','sfunction(''delz'')');
                setmenu(delZero,labelList,cmdList);
```

```
        delPole = uimenu(menuDel,'Label',' ลด Pole');
        labelList = str2mat(' Pole บนแกนจำนวนจริง', ' Poles 1 คู่คอนจูเกต');
        cmdList = str2mat('sfunction(''delpr'')','sfunction(''delp'')');
        setmenu(delPole,labelList,cmdList);

%================ set 2 zeros 2 poles value ================
zr = 5;
p = -1+j*5;
pc = p';

zr_point = plot(real(zr),imag(zr),'oy','EraseMode','xor');
p_point = plot(real(p),imag(p),'xy','EraseMode','xor');
pc_point = plot(real(pc),imag(pc),'xy','EraseMode','xor');

%========= Save Zeros - Poles Data ==============
Z_POINT = [];
Z_VALUE = [];
ZR_POINT = [ zr_point ];
ZR_VALUE = [ zr ];
P_POINT = [ p_point, pc_point ];
P_VALUE = [ p, pc ];
PR_POINT = [];
PR_VALUE = [];

%======= เตรียมค่าสำหรับการคำนวณ frequency , phase response ==========
w = logspace(-1,2,200);
[mag_zr,phase_zr] = sfactor(ZR_VALUE,w);
[mag_p,phase_p] = sfactor(P_VALUE,w);

mag = (mag_zr)./(prod(mag_p')');
mag = 20*log10(mag/max(mag));
phase =  phase_zr - sum(phase_p')';

%================ คำนวณ Impulse Reponse ==============
zz = [ ZR_VALUE Z_VALUE ];
pp = [ PR_VALUE P_VALUE ];
[num,den] = zp2tf(zz,pp,1);
[y,x,t] = impulse(num,den);

%==============สร้างหน้าต่าง response==============
figure2 =  figure('visible','off','units','normalized','position',[0.44
0.13 0.5538 0.7983]);
set(figure2, ...
        'NumberTitle','off', ...
        'Name','Frequency and Phase Response', ...
        'MenuBar','none', ...
        'backingstore','off');

%================ สร้างเมนู พล็อตกราฟ ================
menuResp = uimenu(figure2,'Label','พล็อตกราฟ');
        labelList = str2mat('Frequency - Phase Response', 'Frequency -
Impulse Response');
        cmdList     =     str2mat('sfunction(''plot_fp'')','sfunction(''
plot_ft'')');
        setmenu(menuResp,labelList,cmdList);

%==============สร้างเมนูออกจากโปรแกรม==============
menuExit =  uimenu(figure2,'Label','ออกจากโปรแกรม','Callback','sfunction(''
exit'')');


subplot(211), ...
freq_line = semilogx(w,mag);,grid, ...
        ylabel('dB'),xlabel('Radian frequency')

ax = subplot(212);, ...
time_line = plot(t,y);
hold on
set(time_line,'visible','off');
set(ax,'XScale','log');
phase_line = semilogx(w,phase);, grid, ...
        xlabel('Radian frequency (w)'),ylabel('Radians')
```

```
        set(figure2,'visible','on');
        set(figure1,'Visible','on');

        %=== ========== Save Global Varibles ==========
        point_clicked = 0;     point_conjugate = 0;
        change = 0;     fromDelMenu = 0;
        SFUNCTION_DAT    =    [    figure1,    figure2,    freq_line,    phase_line,
point_clicked, point_conjugate, change, fromDelMenu, time_line, ax ];

        %========== ตรวจสอบว่ามีการคลิกเมาส์ที่ zero และ pole ใน z-plane หรือไม่ ==========
        set(zr_point,'ButtonDownFcn','sfunction(''down'')');
        set(p_point,'ButtonDownFcn','sfunction(''down'')');
        set(pc_point,'ButtonDownFcn','sfunction(''down'')');
        drawnow;

elseif strcmp(action,'down'),
        % สมมติว่ามีการคลิกที่จุด zero ใน S-Plane
        fig1 = SFUNCTION_DAT(1);

        if (fig1~=gcf),
                figure(fig1);
                drawnow discard;
        end

        %=== Check clicked point and assign its conjugate ======
        point_clicked = gco;
        [its_conjugate,change] = findconj(point_clicked);

        set(point_clicked,'color','w');
        set(its_conjugate,'color','w');
        drawnow

        SFUNCTION_DAT(5) = point_clicked;
        SFUNCTION_DAT(6) = its_conjugate;
        SFUNCTION_DAT(7) = change;

        set(gcf,'WindowButtonMotionFcn',sprintf('sfunction(''move'')'));
        set(gcf,'WindowButtonUpFcn',sprintf('sfunction(''up'')'));

elseif strcmp(action,'move'),
        point_clicked = SFUNCTION_DAT(5);
        its_conjugate = SFUNCTION_DAT(6);

        %=== หาตำแหน่งของจุดที่ถูกลากไป ====
        pt = get(gca,'currentpoint');
        x = pt(1,1);
        y = pt(1,2);

        % ========== Update S-Plane ==========
        if (point_clicked == its_conjugate)
                set(point_clicked,'YData',0,'XData',x);
        else
                set(point_clicked,'YData',y,'XData',x);
                set(its_conjugate,'YData',-y,'XData',x);
        end

elseif strcmp(action,'up'),
        fig1 = SFUNCTION_DAT(1);
        point_clicked = SFUNCTION_DAT(5);
        its_conjugate = SFUNCTION_DAT(6);
        change = SFUNCTION_DAT(7);

        figure(fig1)
        pt = get(gca,'currentpoint');
        x = pt(1,1);
        y = pt(1,2);

        set(gcf,'WindowButtonMotionFcn','');  % reset เมื่อมีการเคลื่อนเมาส์
        set(gcf,'WindowButtonUpFcn','');      % reset เมื่อปล่อยปุ่มเมาส์

        %========== Save Pole-Zeros value ==========
        if change == 1
                Idx1 = max(find(Z_POINT == point_clicked));
```

```
            Idx2 = max(find(Z_POINT == its_conjugate));
            Z_VALUE(Idx1) = x + j*y;
            Z_VALUE(Idx2) = x - j*y;
          elseif change == 2
            Idx1 = max(find(ZR_POINT == point_clicked));
            ZR_VALUE(Idx1) = x;
          elseif change == 3
            Idx1 = max(find(P_POINT == point_clicked));
            Idx2 = max(find(P_POINT == its_conjugate));
            P_VALUE(Idx1) = x + j*y;
            P_VALUE(Idx2) = x - j*y;
          else
            Idx1 = max(find(PR_POINT == point_clicked));
            PR_VALUE(Idx1) = x;

        end

        sfunction('redraw');

elseif strcmp(action,'redraw'),
        point_clicked = SFUNCTION_DAT(5);
        its_conjugate = SFUNCTION_DAT(6);
        freq_line = SFUNCTION_DAT(3);
        phase_line = SFUNCTION_DAT(4);
        time_line = SFUNCTION_DAT(9);
        fromDelMenu = SFUNCTION_DAT(8);

        calc_mp;                % Check Zero , Pole is empty ?

        mag = prod(magz')'./prod(magp')';
        mag = 20*log10(mag/max(mag));
        phase =  sum(phasez')' - sum(phasep')';

        %========== ค่านวณ Impulse Reponse ==========
        zz = [ ZR_VALUE Z_VALUE ]';
        pp = [ PR_VALUE P_VALUE ]';
        [num,den] = zp2tf(zz,pp,1);
        [y,x,t] = impulse(num,den);

        set(freq_line,'YData',mag);
        set(phase_line,'YData',phase);
        set(time_line,'YData',y,'XData',t);

        if fromDelMenu
                SFUNCTION_DAT(8) = 0;
            else
                set(point_clicked,'color','y');
                set(its_conjugate,'color','y');
        end

        drawnow;

elseif strcmp(action,'addz'),
        [x,y] = ginput(1);
        newz_value = x + j*y;
        newzc_value = x - j*y;
        newz_point = plot(x,y,'oy','EraseMode','xor');
        newzc_point = plot(x,-y,'oy','EraseMode','xor');
        SFUNCTION_DAT(5) = newz_point;     SFUNCTION_DAT(6) = newzc_point;
        Z_POINT = [ Z_POINT, newz_point, newzc_point ];
        Z_VALUE = [ Z_VALUE, newz_value, newzc_value ];

        sfunction('redraw');

        set(newz_point,'ButtonDownFcn','sfunction(''down'')');
        set(newzc_point,'ButtonDownFcn','sfunction(''down'')');

elseif strcmp(action,'addzr'),
        [x,y] = ginput(1);
        newzr_value = x;
        newzr_point = plot(x,0,'oy','EraseMode','xor');
        SFUNCTION_DAT(5) = newzr_point;     SFUNCTION_DAT(6) = newzr_point;
        ZR_POINT = [ ZR_POINT, newzr_point ];
        ZR_VALUE = [ ZR_VALUE, newzr_value ];
```

```
        sfunction('redraw');

        set(newzr_point,'ButtonDownFcn','sfunction(''down'')');

elseif strcmp(action,'addp'),
        [x,y] = ginput(1);
        newp_value = x + j*y;
        newpc_value = x - j*y;
        newp_point = plot(x,y,'xy','EraseMode','xor');
        newpc_point = plot(x,-y,'xy','EraseMode','xor');
        SFUNCTION_DAT(5) = newp_point;     SFUNCTION_DAT(6) = newpc_point;
        P_POINT = [ P_POINT, newp_point, newpc_point ];
        P_VALUE = [ P_VALUE, newp_value, newpc_value ];

        sfunction('redraw');

        set(newp_point,'ButtonDownFcn','sfunction(''down'')');
        set(newpc_point,'ButtonDownFcn','sfunction(''down'')');

elseif strcmp(action,'addpr'),
        [x,y] = ginput(1);
        newpr_value = x;
        newpr_point = plot(x,0,'xy','EraseMode','xor');
        SFUNCTION_DAT(5) = newpr_point;     SFUNCTION_DAT(6) = newpr_point;
        PR_POINT = [ PR_POINT, newpr_point ];
        PR_VALUE = [ PR_VALUE, newpr_value ];

        sfunction('redraw');

        set(newpr_point,'ButtonDownFcn','sfunction(''down'')');

elseif strcmp(action,'delz'),
        point_del = getpoint;
        Idx1 = find( Z_POINT == point_del );
        if isempty(Idx1),  return;  end
        Idx2 = rem(Idx1,2)*(Idx1+1) + (~rem(Idx1,2))*(Idx1-1);%หา index ของคู่คอนจูเกตของมัน
        if Idx2 < Idx1,
                temp = Idx1;
                Idx1 = Idx2;
                Idx2 = temp;
        end
        delete(Z_POINT(Idx1));
        delete(Z_POINT(Idx2));
        Z_POINT(Idx1:Idx2) = [];      Z_VALUE(Idx1:Idx2) = [];

        SFUNCTION_DAT(8) = 1;
        sfunction('redraw');

elseif strcmp(action,'delzr'),
        point_del = getpoint;
        Idx1 = find( ZR_POINT == point_del );
        if isempty(Idx1),  return;  end
        delete(ZR_POINT(Idx1));  ZR_POINT(Idx1) = [];  ZR_VALUE(Idx1) = [];

        SFUNCTION_DAT(8) = 1;
        sfunction('redraw');

elseif strcmp(action,'delp'),
        point_del = getpoint;
        Idx1 = find( P_POINT == point_del );
        if isempty(Idx1),  return;  end
        Idx2 = rem(Idx1,2)*(Idx1+1) +( ~rem(Idx1,2))*(Idx1-1);% หา index ของคู่คอนจูเกตของมัน
        if Idx2 < Idx1,
                temp = Idx1;
                Idx1 = Idx2;
                Idx2 = temp;
        end
        delete(P_POINT(Idx1));
        delete(P_POINT(Idx2));
        P_POINT(Idx1:Idx2) = [];      P_VALUE(Idx1:Idx2) = [];

        SFUNCTION_DAT(8) = 1;
        sfunction('redraw');
```

```
elseif strcmp(action,'delpr'),
        point_del = getpoint;
        Idx1 = find( PR_POINT == point_del );
        if isempty(Idx1), return; end
        delete(PR_POINT(Idx1)); PR_POINT(Idx1) = []; PR_VALUE(Idx1) = [];

        SFUNCTION_DAT(8) = 1;
        sfunction('redraw');

elseif strcmp(action,'plot_ft'),
        phase_line = SFUNCTION_DAT(4);
        time_line = SFUNCTION_DAT(9);
        ax = SFUNCTION_DAT(10);

        set(phase_line,'visible','off');
        set(ax,'XScale','linear');
        set(time_line,'visible','on');
        xlabel('Time (secs)'),ylabel('Amplitude');

elseif strcmp(action,'plot_fp'),
        phase_line = SFUNCTION_DAT(4);
        time_line = SFUNCTION_DAT(9);
        ax = SFUNCTION_DAT(10);

        set(time_line,'visible','off');
        set(ax,'XScale','log');
        set(phase_line,'visible','on');
        xlabel('Radian frequency (w)'),ylabel('Radians');

elseif strcmp(action,'exit'),
        close all
        clear global SFUNCTION_DAT
        clear global Z_POINT
        clear global Z_VALUE
        clear global ZR_POINT
        clear global ZR_VALUE
        clear global P_POINT
        clear global P_VALUE
        clear global PR_POINT
        clear global PR_VALUE
        clear global w

end
```

22. ไฟล์ spect_est.m

```
function [fre_out,psd_out] = spect_est(x);
% SPECT_EST ...        Spectral estimation.
%
%       SPECT_EST(X) estimates the power spectral density function of the
%               input sequence X, without explicitly computing its Fourier
%               transform.  The input sequence X will be successively fed into
%               a center-frequency adjustable, band-pass filter with user
%               specified bandwidth.  The frequency interval and the filter
%               bandwidth are interactively specified.
%
%       [f,Px] = SPECT_EST(X) returns the mean-square value of the sequence X
%               at the output of a band-pass filter centered at f.

%       AUTHORS : M. Zeytinoglu & N. W. Ma
%             Department of Electrical & Computer Engineering
%             Ryerson Polytechnic University
%             Toronto, Ontario, CANADA
%
%       DATE    : August 1991.
%       VERSION : 1.0

%=======================================================================
% Modifications history:
% ----------------------
%       o   Added "checking"  11.30.1992 MZ
%       o       Tested (and modified) under MATLAB 4.0/4.1 08.16.1993 MZ
```

```
%=========================================================================

global START_OK;
global SAMPLING_CONSTANT;
global SAMPLING_FREQ;
global BINARY_DATA_RATE;
global BELL;
global WARNING;

check;

%-------------------------------------------------------------------------
%       Check input parameters
%-------------------------------------------------------------------------

if (nargin ~= 1)
   error(eval('eval(BELL),eval(WARNING),help spect_est'));
   return;
end

%-------------------------------------------------------------------------------
%       Default values
%-------------------------------------------------------------------------------

fs2 = SAMPLING_FREQ/2;
min_fractional_bw = 200/5000;
min_fractional_Hz = min_fractional_bw * fs2;

norm_factor = sqrt( length(x) );

fprintf('-------------------------------------------------------------------------
--\n');
fprintf('FOR SPECTRAL ESTIMATION FREQUENCY RANGE SHOULD BE [ 0, %10.2f ] Hz.\
n',fs2);
fprintf('-------------------------------------------------------------------------
--\n');

f_range = input('Enter frequency range [FREQ_START, FREQ_STOP] [Hz] ..... = ');
f_range = sort(f_range);
f_start = min(f_range);
f_stop  = max(f_range);

if ( f_start < 0 )
   fprintf(' WARNING: Start frequency should be non-negative.\n');
   fprintf('          Resetting to 0 Hz.\n');
   f_start = 0;
end

if ( f_stop > fs2 )
   fprintf(' WARNING: Stop frequency must be less than %10.2f Hz.\n',fs2);
   fprintf('          Resetting to %10.2f Hz.\n',fs2);
   f_stop = fs2;
end

bw      = input('Enter bandwidth for BPF [Hz] ........................ = ');

if ( bw < 0 )
   error('BPF Bandwidth must be positive.');
end
if ( bw < min_fractional_Hz )
   fprintf('   WARNING:   BPF   bandwidth   must   be   larger   than   %8.2f
Hz;\n',min_fractional_Hz);
   fprintf('          otherwise the BPF will be unstable.\n');
   fprintf('                              Resetting   BPF   bandwidth   to   %10.2f
Hz.\n',min_fractional_Hz);
   bw = min_fractional_Hz;
end
disp('')

freq_length = f_stop - f_start;
no_bpf      = fix(freq_length/bw);

if ( no_bpf <= 0 )
   error('FREQ_START, FREQ_STOP, and BANDWIDTH values are not compatible.');
```

```
end

range = [f_start, f_start+bw];

for k = 1:no_bpf

    if ( min(range) == 0 )                              % LOW-PASS FILTER'
      range_true = max(range);
      f(k) = sum(range)/2;
      range_true = range_true/fs2;
      [b a] = butter(6,range_true);
      y(k) = meansq(filter(b,a,x))/norm_factor;
    elseif ( max(range) == fs2 )                        % HIGH-PASS FILTER
      range_true = min(range);
      f(k) = sum(range)/2;
      range_true = range_true/fs2;
      [b a] = butter(6,range_true,'high');
      y(k) = meansq(filter(b,a,x))/norm_factor;
    else                                                % BAND-PASS FILTER
      f(k) = sum(range)/2;
      range_true = range/fs2;
      [b a] = butter(6,range_true);
      y(k) = meansq(filter(b,a,x))/norm_factor;
    end

    range = range + bw;

    fprintf(' o PSD estimate at %10.2f [Hz] is %10.2e [W].\n',f(k),y(k));

end

disp('')
disp('SPECTRAL ESTIMATION process is complete');
input('Hit any key to display the estimated magnitude spectrum.  ');

semilogy(f/1000,y), xlabel('Frequency [kHz]'); ylabel('Power [W]');

if (nargout ~= 0)
   freq_out = f;
   psd_out  = y;
end
```

23. ไฟล์ transfor.m

```
function [out,f] = transform(in)
% Take FFT to signal.

% Somboon Pr. 17-Nov-95.

global SAMPLING_FREQ;

len = length(in);
%m = fftsize(len);
%IN =  fft(in,m)*2/len;
IN = fft(in)/len;

out = fftshift(IN);
fscale = 0:(len-1);
f = ( fscale - len/2  )*SAMPLING_FREQ/len;
```

24. ไฟล์ uifft.m

```
function uifft(action);
% UIFFT User Interface FFT Progame
%       You can input any waveform.

% Somboon Pr.
% 17-Jun-96

%  ??? nSig = ?  -> Amplitude of freq response

global POINT
global POINT_VALUE
```

```
global GUI_DATA

if nargin<1,
    action='start';
end;

if strcmp(action,'start'),

        %============== สร้างหน้าต่าง โดเมนทางเวลา ==============
        figure1 = figure('Visible','off','units','normalized','position',[0.003
0.6 1-2*0.003 1-0.6]);
        set(figure1, ...
                'NumberTitle','off', ...
                'Name','โดเมนทางเวลา', ...
                'MenuBar','none', ...
                'backingstore','off');
        axis([0 15 -8 8]),grid
        ylabel('Amplitude')
%       xlabel('Time (sec)')
        ax = get(figure1,'Children');
        yscale = [-8 -6 -4 -2 0 2 4 6 8 ];
        xscale = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 ];
        set(ax,'XTick',xscale,'YTick',yscale);

        %================== สร้างเมนู ของ Figure1 ==================
        menuClear = uimenu(figure1,'Label','เคลียร์สัญญาณ','Callback','uifft(''
clearpoint'')');
        menuExit = uimenu(figure1,'Label','ออกจากโปรแกรม','Callback','uifft(''
exit'')');

        hold on

        % Set data and plot
        POINT_VALUE = zeros(1,16);
        for i = 0:15,
                eval( [ 'point' num2str(i) ' = plot(' num2str(i) ',0,''*'',''
EraseMode'',''xor''); ' ] );
                POINT = [ POINT eval( [ 'point' num2str(i) ] ) ];
        end
        POINT_VALUE(1:3) = [4 4 4];
        set(point0,'YDATA',4);
        set(point1,'YDATA',4);
        set(point2,'YDATA',4);

        %================== สร้างหน้าต่าง Response (Figure2) ==================
        figure2 = figure('Visible','off','Units','normalized','Position',[0.003
0.003 1-2*0.003 0.48]);
        set(figure2, ...
                'NumberTitle','off', ...
                'Name','ผลตอบสนองทางความถี่และเฟส', ...
                'MenuBar','none', ...
                'backingstore','off');

        % Calculation for response displays.
        nFFT = 1024;
        nSig = 1;
        out = fft(POINT_VALUE,nFFT)/nSig;
        out = fftshift(out);
        fscale = 0:(nFFT-1);
        f = ( fscale - nFFT/2  )/nFFT;

        subplot(121), ...
                freqline = plot(f,abs(out),'EraseMode','xor');
                axis([-0.5 0.5 0 20]), grid
        subplot(122), ...
                phaseline = plot(f,angle(out),'EraseMode','xor');
                axis([-0.5 0.5 -4 4]), grid

        % Set grid for figure2
        ax2 = get(figure2,'Children');
        xscale = [ -0.4 -0.2 0 0.2 0.4 ];
        yscale_freq = [0 2 4 6 8 10 12 14 16 18 20];
        yscale_phase = [-4 -3 -2 -1 0 1 2 3 4];
```

```
        set(ax2(1),'XTick',xscale,'YTick',yscale_phase);
        set(ax2(2),'XTick',xscale,'YTick',yscale_freq);

        clicked_point = 0;
        clicked_Idx = 0;
        GUI_DATA = [ freqline, phaseline, clicked_point, clicked_Idx, figure1,
figure2 ];

        set(figure2,'Visible','on');
        set(figure1,'Visible','on');

        for i = 1:16,
                set(POINT(i),'ButtonDownFcn','uifft(''down'')');
        end

elseif strcmp(action,'down'),
        clicked_point = gco;
        set(clicked_point,'color','w');
        drawnow

        GUI_DATA(3) = clicked_point;

        set(gcf,'WindowButtonMotionFcn',sprintf('uifft(''move'')'));
        set(gcf,'WindowButtonUpFcn',sprintf('uifft(''up'')'));

elseif strcmp(action,'move'),
        freq_line = GUI_DATA(1);
        phase_line = GUI_DATA(2);
        clicked_point = GUI_DATA(3);

        pt = get(gca,'currentpoint');
        y = pt(1,2);
        if y < -8, y = -8;  end
        if y >  8, y =  8; end

        % Update clicked point position
        set(clicked_point,'YData',y);

        % Update POINT_VALUE
        clicked_Idx = find(POINT == clicked_point);
        POINT_VALUE(clicked_Idx) = y;

        % Update frequency and phase response
        nFFT = 1024;
        nSig = 1;
        out = fft(POINT_VALUE,nFFT)/nSig;
        out = fftshift(out);
        fscale = 0:(nFFT-1);
        f = ( fscale - nFFT/2 )/nFFT;
        set(freq_line,'XData',f,'YData',abs(out));
        set(phase_line,'XData',f,'YData',angle(out));

        GUI_DATA(4) = clicked_Idx;

elseif strcmp(action,'up'),
        clicked_point = GUI_DATA(3);
        clicked_Idx = GUI_DATA(4);

        pt = get(gca,'currentpoint');
        y = pt(1,2);
        if y < -8, y = -8;  end
        if y >  8, y =  8; end

        set(gcf,'WindowButtonMotionFcn','');  % reset เมื่อมีการเคลื่อนเมาส์
        set(gcf,'WindowButtonUpFcn','');      % reset เมื่อปล่อยปุ่มเมาส์

        POINT_VALUE(clicked_Idx) = y;
        set(clicked_point,'color','y');

        uifft('redraw');

elseif strcmp(action,'redraw'),
        freqline = GUI_DATA(1);
        phaseline = GUI_DATA(2);
```

```
        nFFT = 1024;
        nSig = 1;
        out = fft(POINT_VALUE,nFFT)/nSig;
        out = fftshift(out);
        fscale = 0:(nFFT-1);
        f = ( fscale - nFFT/2  )/nFFT;
        set(freqline,'XData',f,'YData',abs(out));
        set(phaseline,'XData',f,'YData',angle(out));

        drawnow;

elseif strcmp(action,'clearpoint'),
        POINT_VALUE = zeros(1,16);
        for i = 1:16,
                set(POINT(i),'YData',0);
        end

        uifft('redraw');

elseif strcmp(action,'exit'),
        figure1 = GUI_DATA(5);
        figure2 = GUI_DATA(6);
        figure(figure1), close
        figure(figure2), close
        clear global POINT
        clear global POINT_VALUE
        clear global GUI_DATA

end
```

25. ไฟล์ uiplotf.m

```
function [ff,Amp] =  uiplotf( xa, dt )
%UIPLOTF   Plot  Fourier Transform (Mag) of "ANALOG" signal
%   Usage:   [f,XA] = uiplotf( xa, dt )
%       xa :   "ANALOG" signal
%       dt :    sampling interval for the simulation of xa(t)
%
L = length(xa);
Nfft = round( 2 .^ nextpow2(5*L) );
Xa = fft(xa,Nfft);
range = 0:(Nfft/4);
ff = range/Nfft/dt;
Amp = abs( Xa(1+range) );
```

26. ไฟล์ uisample.m

```
function uisample(action,in1);
%  Require :   mystem.m
%              sampling.m
%              uiplotf.m

if nargin<1,
    action='start';
end;

global UISAMPLE_DAT
global POINT

if strcmp(action,'start'),

   %=======================
   % Initial Data
   %=======================
    fsim = 180e3;
    fs = 20e3;
    POINT = fsim/fs;
    f = 0.5*fs;
    fmin = 0;
    fmax = 2*fs;
```

```
t = (0:1000)/fsim;
x = cos(2*pi*f*t);
x_sampled = sampling(x,fs);
x_s = x_sampled*fsim/fs;          % Compensate Energy
[ff,H] = uiplotf(x_s,1/(fsim));

%=====================================
% Graphics initialization
oldFigNumber = watchon;
figNumber = figure;
set(gcf, ...
    'NumberTitle','off', ...
    'Name','ทฤษฎีการสุ่มตัวอย่าง', ...
    'backingstore','off',...
    'Units','normalized',...
    'Position',[0.0025 0.2 0.992  0.7] );

%=============================
% Information for all buttons
%=============================
top=0.95;
bottom=0.05;
left=0.82;
yInitLabelPos=0.90;
btnWid = 0.13;
btnHt=0.08;
% Spacing between the label and the button for the same command
btnOffset=0.02;
% Spacing between the button and the next command's label
spacing=0.02;
%bottom=bottom+spacing;

%=====================================
% The CONSOLE frame
frmBorder=0.02;
yPos=0.02;
frmPos=[left-frmBorder bottom-frmBorder btnWid+2*frmBorder ...
        0.9+2*frmBorder];
h=uicontrol( ...
    'Style','frame', ...
    'Units','normalized', ...
    'Position',frmPos, ...
    'BackgroundColor',[0.5 0.5 0.5]);

%=====================================
% The Envelope button
env_button = uicontrol( ...
    'Style','push', ...
    'Units','normalized', ...
    'Position',[left bottom+btnHt+spacing  btnWid btnHt], ...
    'String','เทียบสัญญาณเดิม', ...
    'Callback','uisample(''analog'')');

%=====================================
% The CLOSE button
done_button=uicontrol('Style','Pushbutton', ...
    'Position',[left bottom btnWid btnHt], ...
    'Units','normalized',...
    'Callback','uisample(''done'')','String','ปิดหน้าต่าง');

%=====================================
% Signal freq. slider
%=====================================
f_text=uicontrol('Style','text','Position',[.18 .02 .38 .07],...
    'Units','normalized','BackgroundColor','black',...
    'ForegroundColor','white','String','ความถี่ของสัญญาณที่ถูกสุ่ม (Hz):');

% Text min to max
uicontrol('style','text','pos',[.14 .07 .02 .05],...
    'Units','normalized','BackgroundColor','black',...
    'ForegroundColor','white','String','0');
uicontrol('style','text','pos',[.74 .07 .02 .05 ],...
    'Units','normalized','BackgroundColor','black',...
    'ForegroundColor','white','String','2fs');          % text  '2fs'
```

```
% Edit Text : f
f_field=uicontrol('Style','edit','Position',[.59 .02 .12 .07],...
    'Units','normalized','String',num2str(f),...
    'CallBack','uisample(''set_f'',2); uisample(''redraw'');');

% Slider
fn = f/fs;
fn_max = fmax/fs;
fn_min = fmin/fs;
f_slider=uicontrol('Style','slider','Position',[.12 .12 .6 .04],...
    'Units','normalized','Value',fn,'Max',fn_max,'Min',fn_min,...
    'Callback','uisample(''set_f'',1); uisample(''redraw'');');

%=======================================
% Edit text and label for sampling freq : fs
%=======================================
labelColor=192/255*[1 1 1];
labelWid=0.13;
labelHt=0.05;
btnNumber=1;
yLabelPos=top-(btnNumber-1)*(btnHt+labelHt+spacing);
labelStr='ความถี่ในการสุ่ม (fs)';

% Generic label information
labelPos=[left yLabelPos-labelHt labelWid labelHt];
uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',labelPos, ...
    'BackgroundColor',labelColor, ...
    'HorizontalAlignment','left', ...
    'String',labelStr);

% Edit Text : fs
Pos=[left yLabelPos-labelHt-btnHt-btnOffset btnWid btnHt];
fs_field=uicontrol('Style','edit','Position',Pos,...
           'Units','normalized','String',num2str(fs),...
           'CallBack','uisample(''set_fs''); uisample(''redraw'');');

% ===================================
% Button Fs Decrease and Increase
%===================================
% Button Fs Decrease
bm = yLabelPos-labelHt-btnHt-btnOffset-btnHt;
btnFSdec=uicontrol('Style','Pushbutton', ...
    'Position',[left bm  labelWid/2 labelHt], ...
    'Units','normalized',...
    'Callback','uisample(''btnFSdec'')',...
    'String','<-');

% Button Fs Increase
btnFSinc=uicontrol('Style','Pushbutton', ...
    'Position',[left+labelWid/2 bm labelWid/2 labelHt], ...
    'Units','normalized',...
    'Callback','uisample(''btnFSinc'')',...
    'String','->');

%===================================
% Plot Graphic Time and Freq. Domain
%===================================
% time domain
ax_time=axes('Position',[.12 .66 .6 .26],'XLim',[0 5],'YLim',[-1 1]);
time_line = plot(t*1000,x_sampled);
axis([0 5.6 -1 1]);
grid;
ylabel('Waveform');
xlabel('Time (mSec)');

%  frequency domain
ax_freq=axes('Position',[.12 .28 .6 .26],'XLim',[0 50],'YLim',[0 1000]);
freq_line=plot(ff/1000,H);
axis([0 45 0 600]);
grid;
```

```
    ylabel('Magnitude');
    xlabel('Frequency (kHz)');

    drawnow;

    env_check = 0;   % use for action 'analog'
    UISAMPLE_DAT =  [ fsim; fs; f;  fmin; fmax; ax_time; ax_freq; time_line;
freq_line; fs_field; f_field; f_slider; env_check; btnFSdec; btnFSinc; x' ];
    watchoff(oldFigNumber);

elseif strcmp(action,'redraw'),
    fsim = UISAMPLE_DAT(1);
    fs = UISAMPLE_DAT(2);
    f = UISAMPLE_DAT(3);
    ax_time=UISAMPLE_DAT(6);
    ax_freq=UISAMPLE_DAT(7);

    t = (0:1000)/(fsim);
    x = cos(2*pi*f*t);
    x_sampled = sampling(x,fs);
    x_s = x_sampled*fsim/fs;          % Compensate Energy
    [ff,H] = uiplotf(x_s,1/(fsim));

    % Update f_field and f_slider
    set(UISAMPLE_DAT(11),'string',num2str(f));
    fn = f/fs;
    set(UISAMPLE_DAT(12),'value',fn);

    % Update fs_field
    set(UISAMPLE_DAT(10),'string',num2str(fs));

    % Update Graphic
    axes(ax_time);
    plot(t*1000,x_sampled);
    axis([0 5.6 -1 1]);
    grid
    ylabel('Waveform');
    xlabel('Time (mSec)');
    axes(ax_freq);
    plot(ff/1000,H);
    axis([0 45 0 600]);
    grid;
    ylabel('Magnitude');
    xlabel('Frequency (kHz)');

    UISAMPLE_DAT(16:1016) = x';

    drawnow;

elseif strcmp(action,'set_f'),
    fs = UISAMPLE_DAT(2);

    if (in1==1),    % set from slider
        fn = get(UISAMPLE_DAT(12),'value');
        UISAMPLE_DAT(3) = fn*fs;
    else  % set from edit text
        fmin = UISAMPLE_DAT(4);
        fmax = UISAMPLE_DAT(5);
        f=str2num(get(UISAMPLE_DAT(11),'string'));
        if isempty(f),    % handle non-numeric input into field
            set(UISAMPLE_DAT(11),'string',num2str(UISAMPLE_DAT(3)));
        else
            if (f > fmax),
                f = fmax;
            end;
            if (f < fmin),
               f = fmin;
            end;
            UISAMPLE_DAT(3) = f;
        end
    end

elseif strcmp(action,'set_fs'),
    fsim = UISAMPLE_DAT(1);
```

```
    fs = str2num(get(UISAMPLE_DAT(10),'string'));
    POINT = round(fsim/fs);
    if POINT < 9,
        POINT = 9;
    end
    if POINT > 18,
        POINT = 18;
    end

    fs = fsim/POINT;

    UISAMPLE_DAT(2) = fs;
    uisample('redraw');

elseif strcmp(action,'btnFSdec'),
    fsim = UISAMPLE_DAT(1);

    POINT = POINT+1;
    if POINT > 18,
        POINT = 18;
    end
    fs = fsim/POINT;

    UISAMPLE_DAT(2) = fs;
    uisample('redraw');

elseif strcmp(action,'btnFSinc'),
    fsim = UISAMPLE_DAT(1);

    POINT = POINT-1;
    if POINT < 9,
        POINT = 9;
    end
    fs = fsim/POINT;

    UISAMPLE_DAT(2) = fs;
    uisample('redraw');

elseif strcmp(action,'analog'),
    fsim = UISAMPLE_DAT(1);
    fs = UISAMPLE_DAT(2);
    f =UISAMPLE_DAT(3);
    ax_time=UISAMPLE_DAT(6);
    env_check = UISAMPLE_DAT(13);
    x = UISAMPLE_DAT(16:1016)';

    t = (0:1000)/(fsim);
    axes(ax_time);
    if env_check == 0,
          hold on
          plot(t*1000,x,'w');
          hold off
          env_check = 1;
        else
          x_sampled = sampling(x,fs);
          plot(t*1000,x_sampled)
          axis([0 5.6 -1 1]);
          grid
          ylabel('Waveform');
          xlabel('Time (mSec)');
          env_check = 0;
    end

    UISAMPLE_DAT(13) = env_check;

elseif strcmp(action,'done'),
    close(gcf);
    clear global UISAMPLE_DAT

end
```

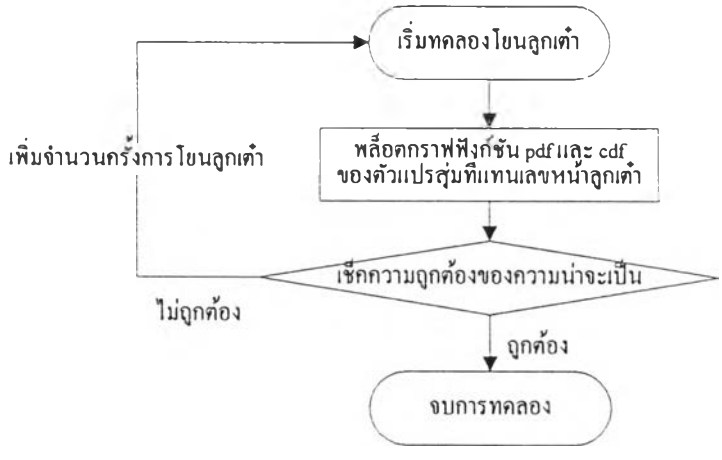
ภาคผนวก ง

แผนผังงานแสดงขั้นตอนการทดลองของชุดทดลองด้านระบบสื่อสารแบบดิจิตอล

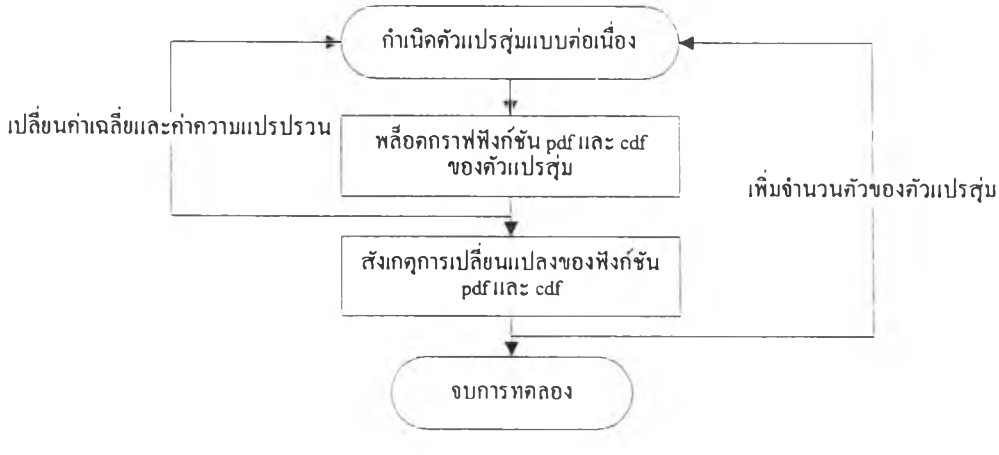
ความนำ

ในภาคผนวก ง นี้ เป็นการแสดงแผนผังงานของแต่ละการทดลองของชุดทดลองด้านระบบสื่อสารแบบดิจิตอล โดยมีรายละเอียดดังต่อไปนี้

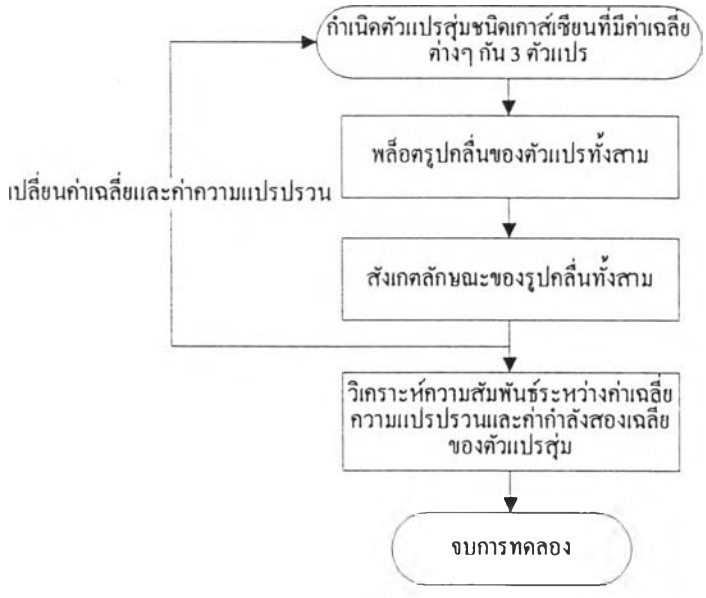
แผนผังงานการทดลองที่ 1 ทฤษฎีความน่าจะเป็น



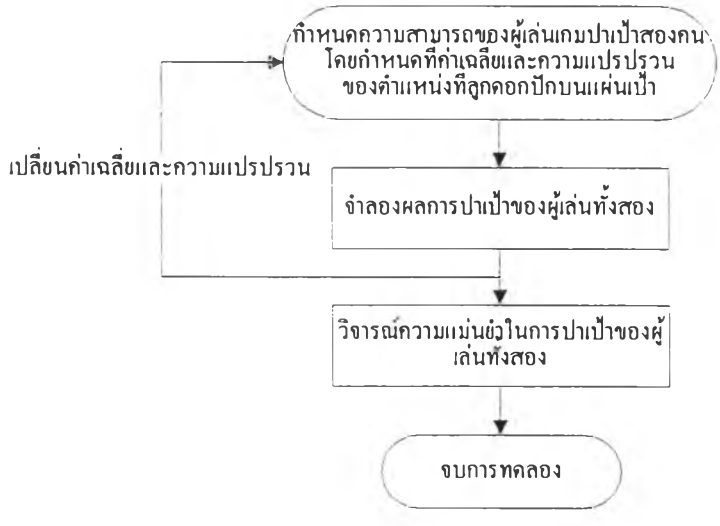
รูปที่ 1 แผนผังงานขั้นตอนการทดลองที่ 1.1 ตัวแปรสุ่มแบบไม่ต่อเนื่อง



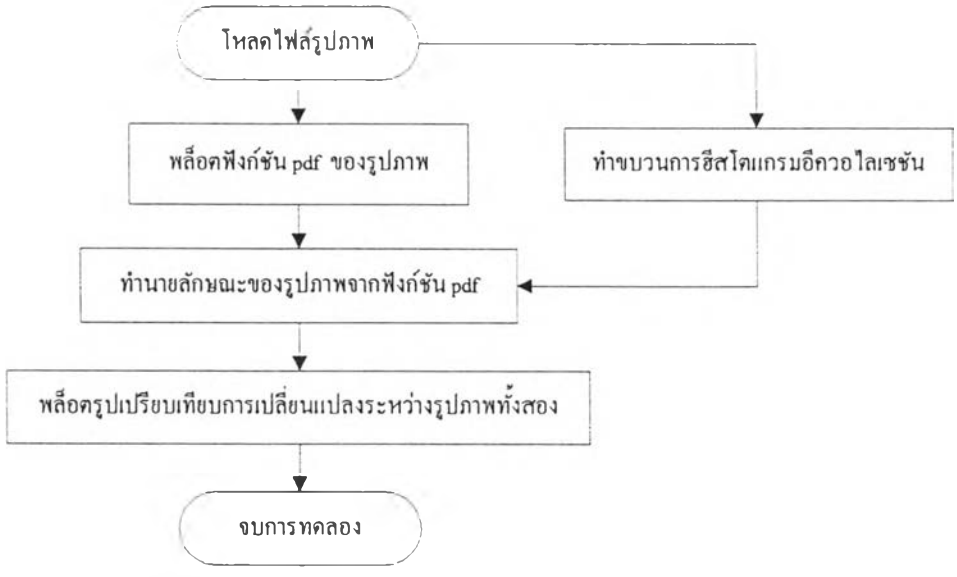
รูปที่ 2 แผนผังงานขั้นตอนการทดลองที่ 1.2 ตัวแปรสุ่มแบบต่อเนื่อง



รูปที่ 3 แผนผังงานขั้นตอนการทดลองที่ 1.3 ค่าเฉลี่ย ความแปรปรวนและกำลังงาน

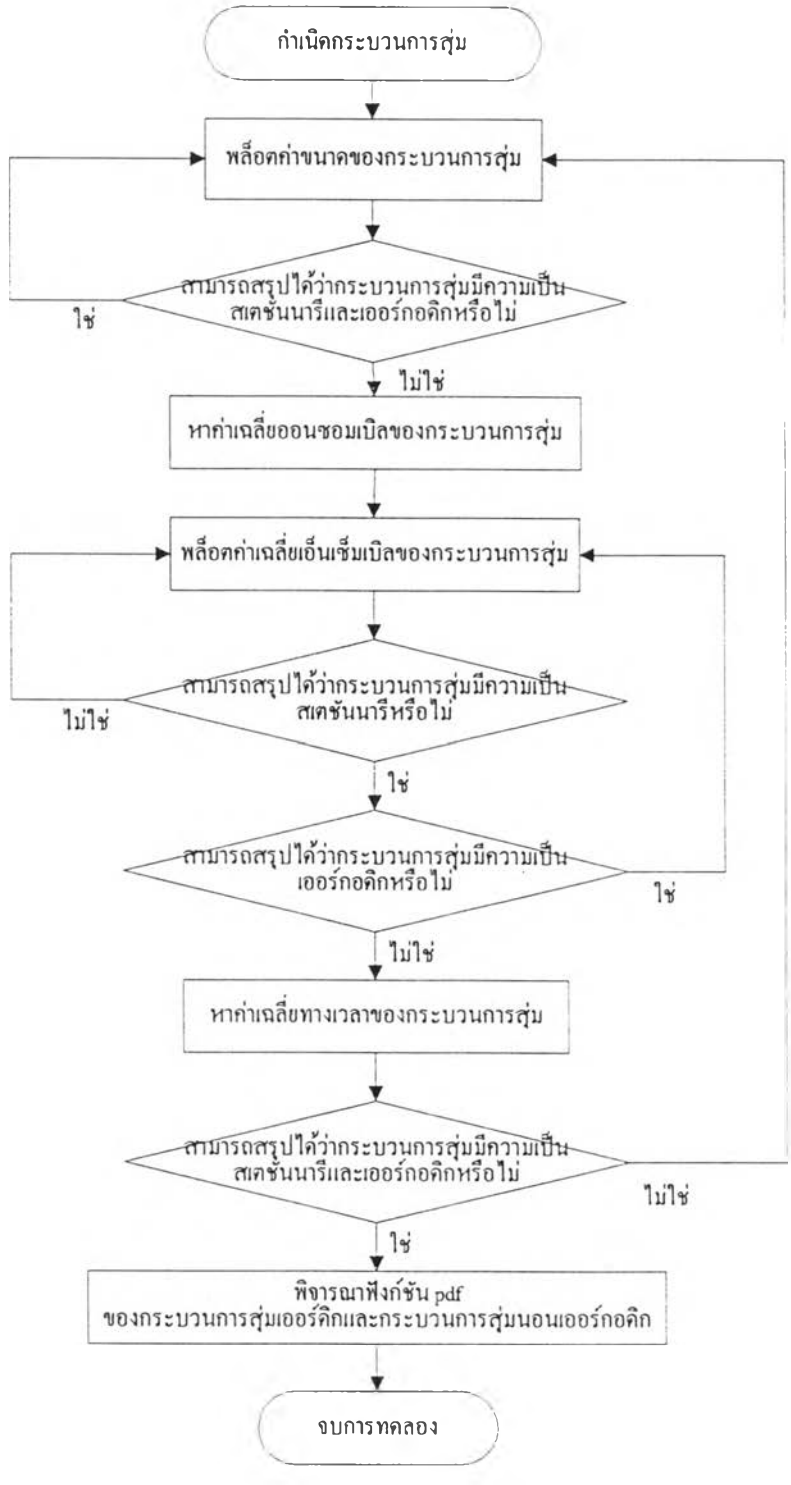


รูปที่ 4 แผนผังงานขั้นตอนการทดลองที่ 1.4 เกมปาเป้า

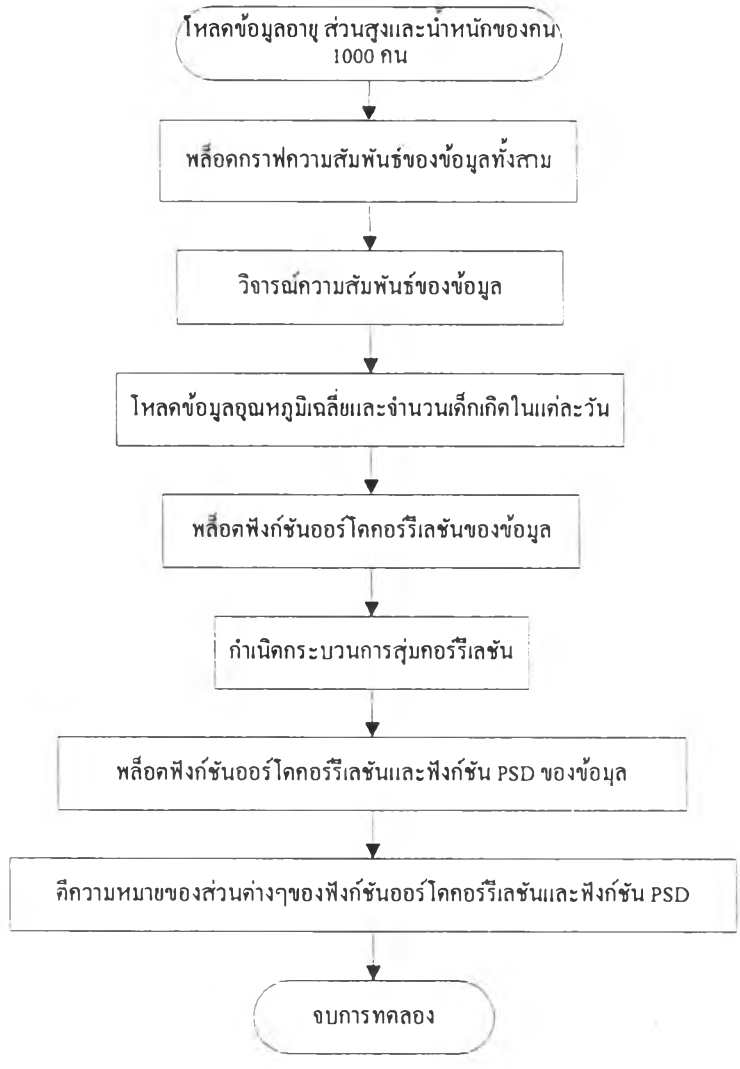


รูปที่ 5 แผนผังงานขั้นตอนการทดลองที่ 1.5 ฟังก์ชัน pdf กับการประมวลผลภาพ

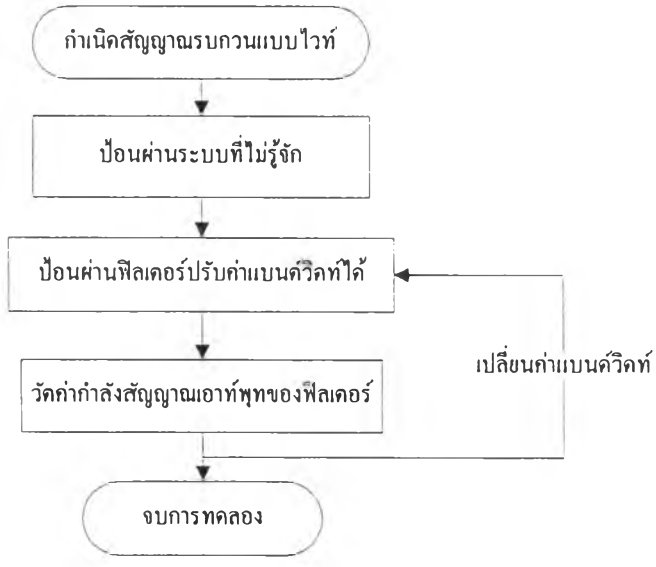
แผนผังงานการทดลองที่ 2 กระบวนการสุ่ม



รูปที่ 6 แผนผังงานขั้นตอนการทดลองที่ 2.1 ความเป็นสเตชันนารีในมุมกว้างและเออร์กอดิก

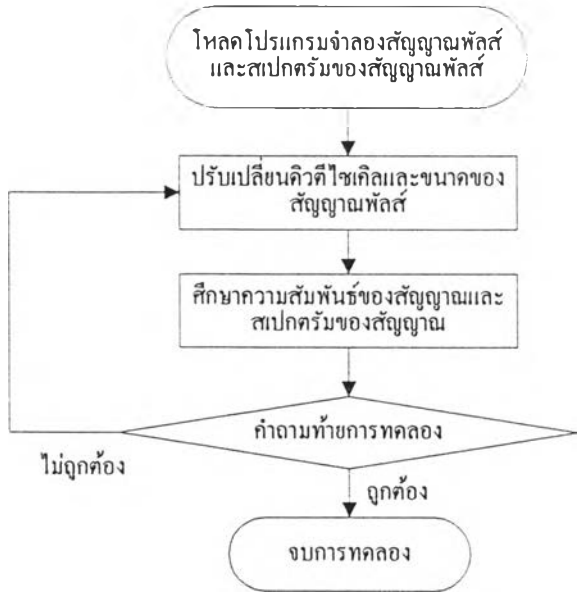


รูปที่ 7 แผนผังงานขั้นตอนการทดลองที่ 2.2 ฟังก์ชันคอร์รีเลชันและความหนาแน่นสเปกตรัมกำลัง

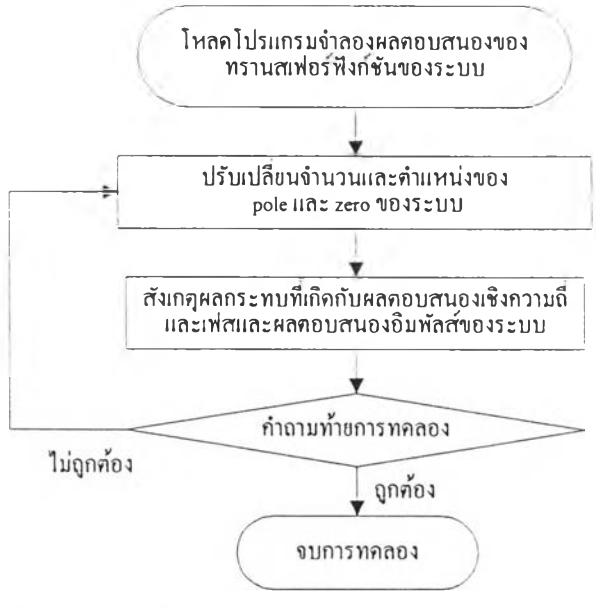


รูปที่ 8 แผนผังงานขั้นตอนการทดลองที่ 2.3 สัญญาณรบกวนแบบไวท์

แผนผังงานการทดลองที่ 3 การวิเคราะห์สัญญาณและระบบเชิงเส้น

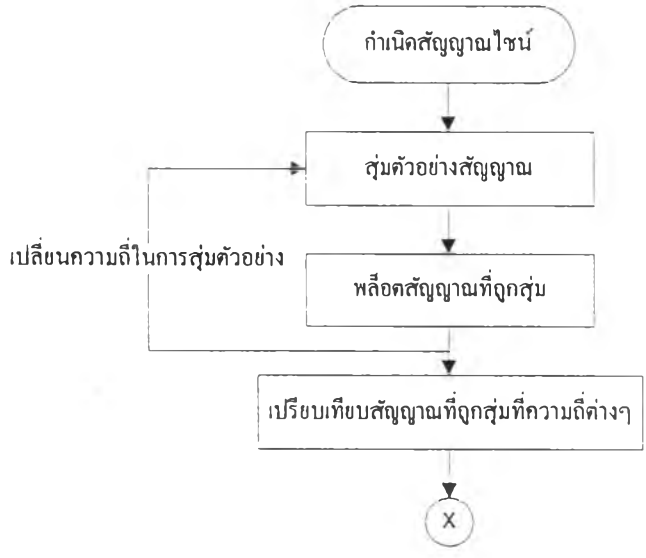


รูปที่ 9 แผนผังงานขั้นตอนการทดลองที่ 3.1 การวิเคราะห์สัญญาณ

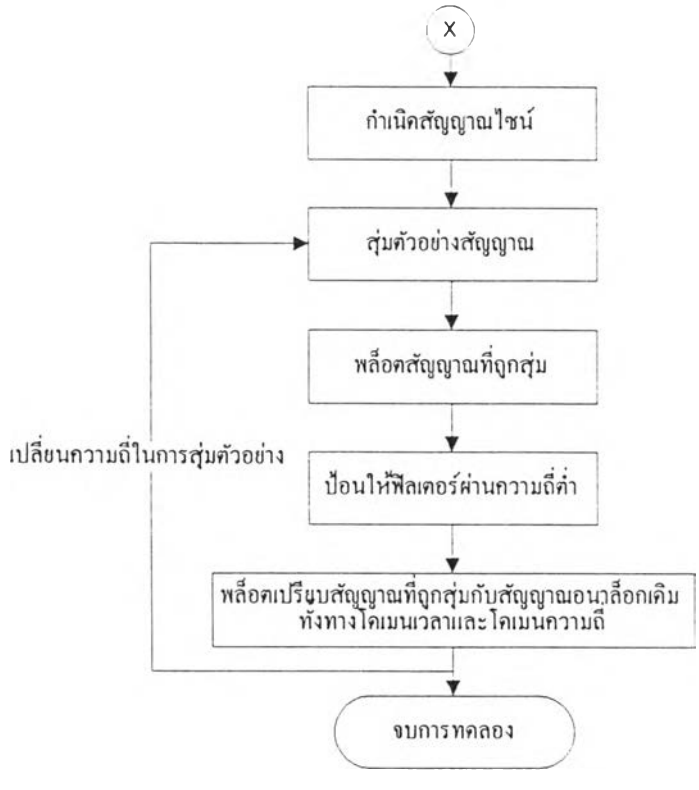


รูปที่ 10 แผนผังงานขั้นตอนการทดลองที่ 3.2 การวิเคราะห์ระบบเชิงเส้น

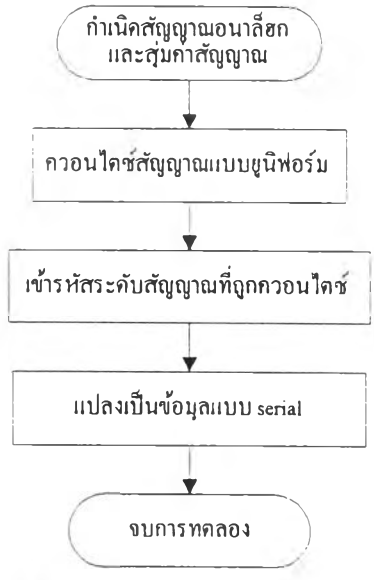
แผนผังงานการทดลองที่ 4 การแปลงเป็นสัญญาณดิจิตอล



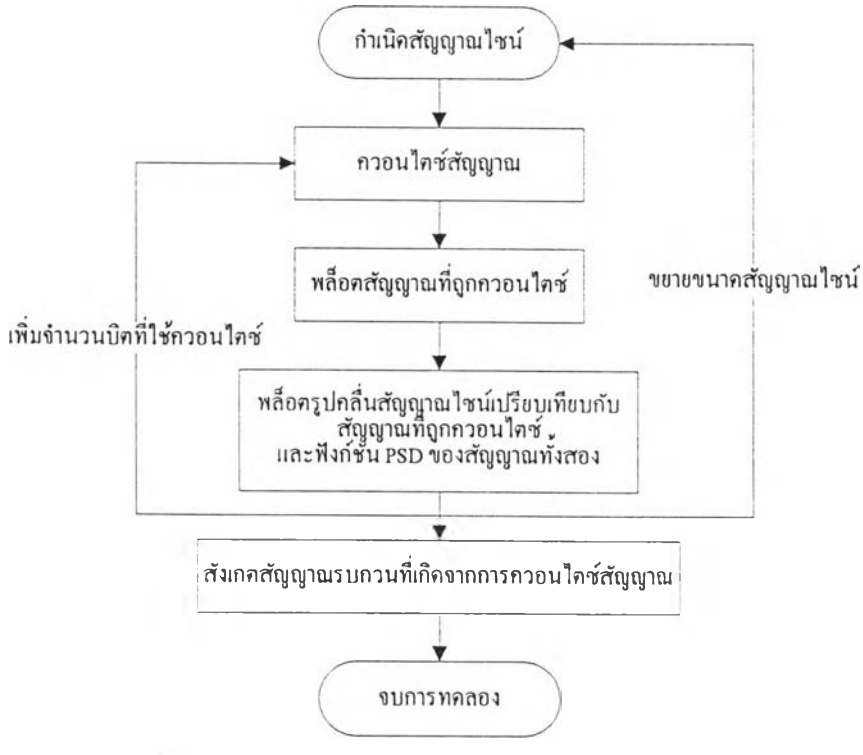
รูปที่ 11 แผนผังงานขั้นตอนการทดลองที่ 4.1 ทฤษฎีการสุ่มตัวอย่าง (วิเคราะห์เชิงเวลา)



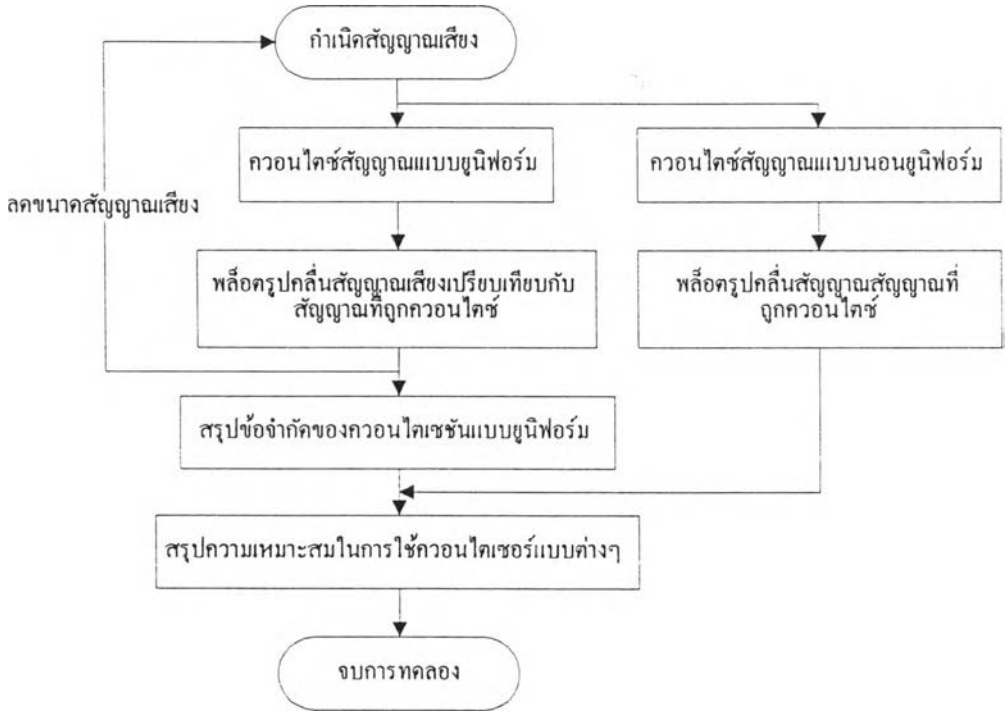
รูปที่ 12 แผนผังงานขั้นตอนการทดลองที่ 4.1 ทฤษฎีการสุ่มตัวอย่าง (วิเคราะห์เชิงความถี่)



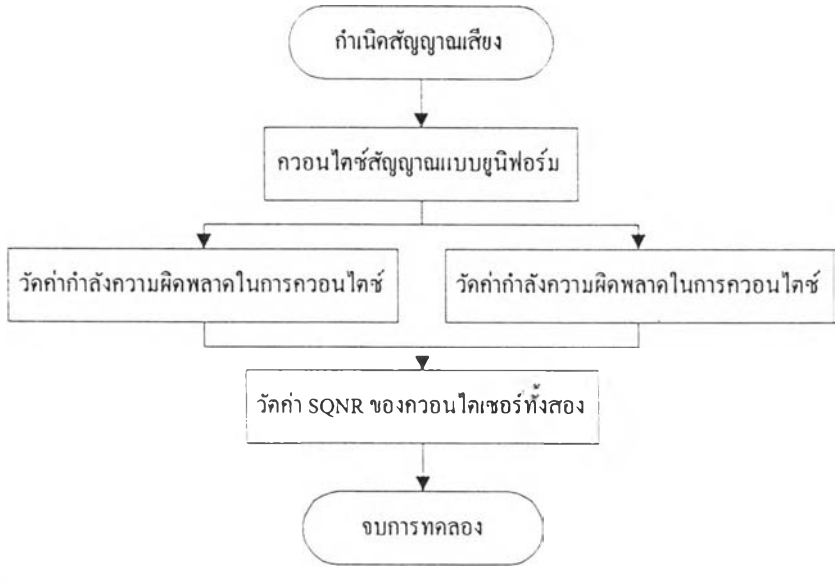
รูปที่ 13 แผนผังงานขั้นตอนการทดลองที่ 4.2 ควอนไตเซชันแบบยูนิฟอร์ม



รูปที่ 14 แผนผังงานขั้นตอนการทดลองที่ 4.3 การผิดเพี้ยนของสัญญาณที่เกิดจากควอนไตเซชัน



รูปที่ 15 แผนผังงานการทดลองที่ 4.4 ควอนไตเซชันแบบนอนยูนิฟอร์ม

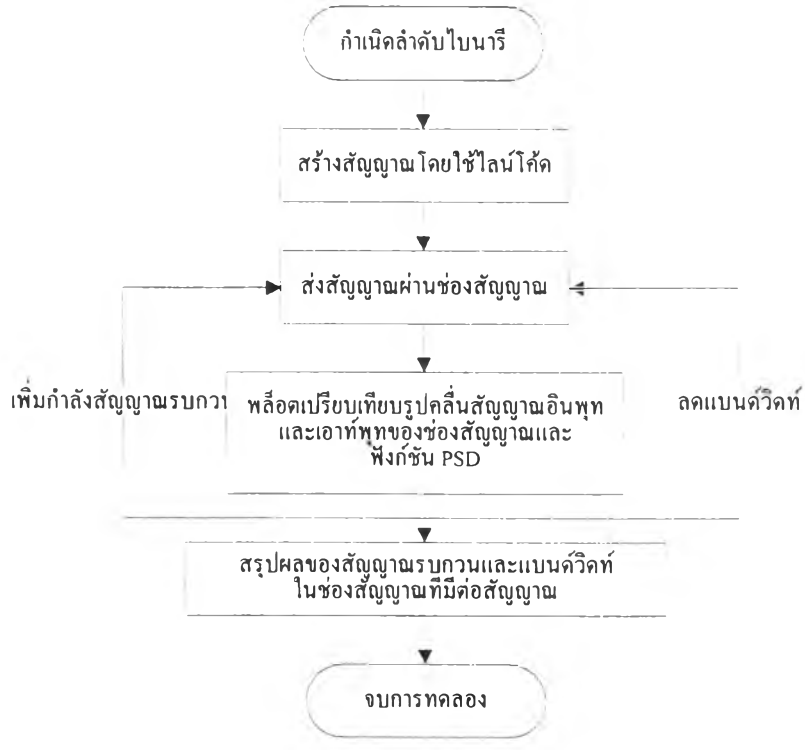


รูปที่ 16 แผนผังงานขั้นตอนการทดลองที่ 4.5 ค่ากำลังของสัญญาณต่อควอนไตเซชันนอยส์

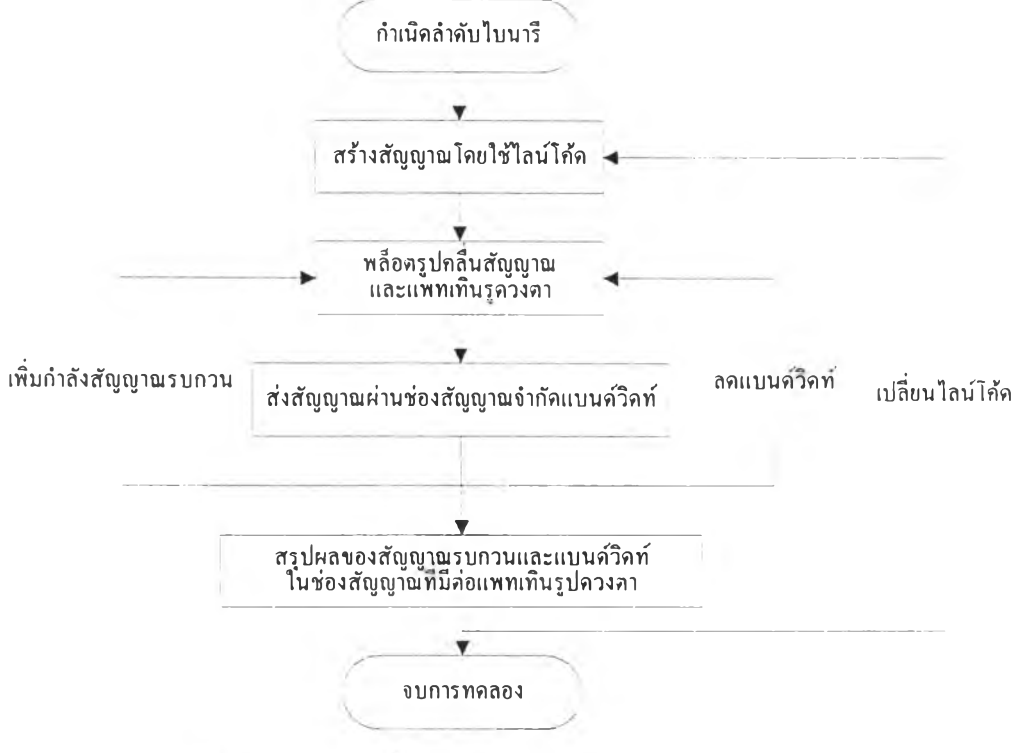
แผนผังงานการทดลองที่ 5 รูปแบบการส่งสัญญาณไบนารี

รูปที่ 17 แผนผังงานขั้นตอนการทดลองที่ 5.1 ไลน์โค้ดแบบต่างๆ ที่ใช้ในระบบสื่อสาร

รูปที่ 18 แผนผังงานขั้นตอนการทดลองที่ 5.2 ความหนาแน่นสเปกตรัมกำลังของไลน์โค้ดแบบต่างๆ

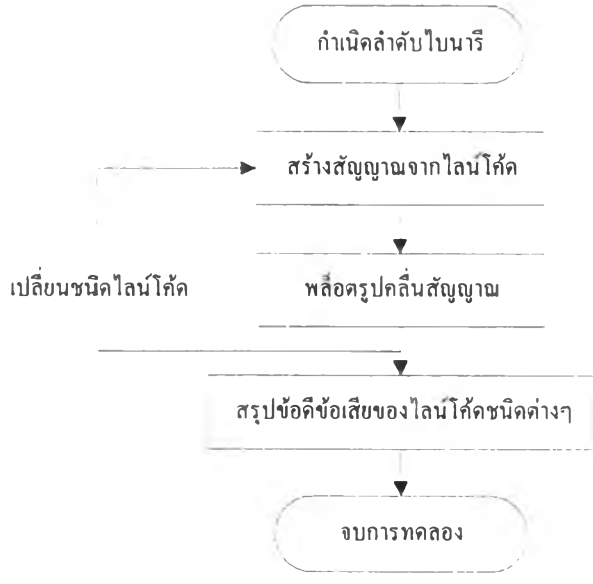


รูปที่ 19 แผนผังงานขั้นตอนการทดลองที่ 5.3 คุณลักษณะของช่องสัญญาณ

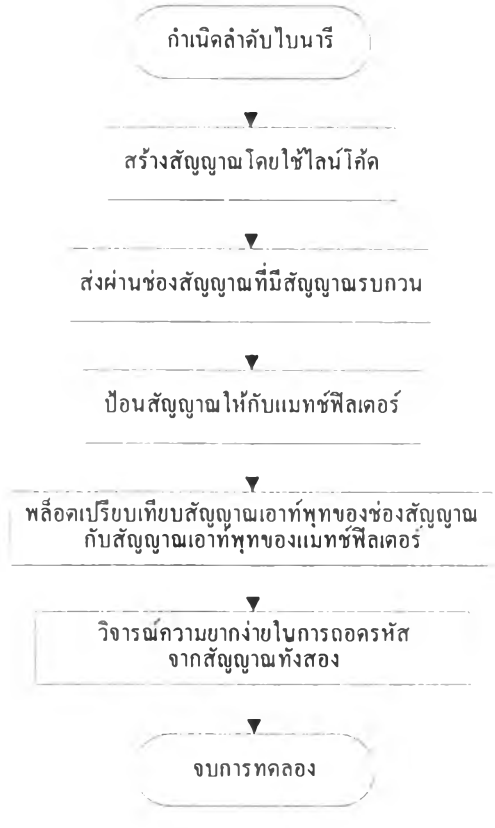


รูปที่ 20 แผนผังงานขั้นตอนการทดลองที่ 5.4 แพทเทินรูปดวงตา

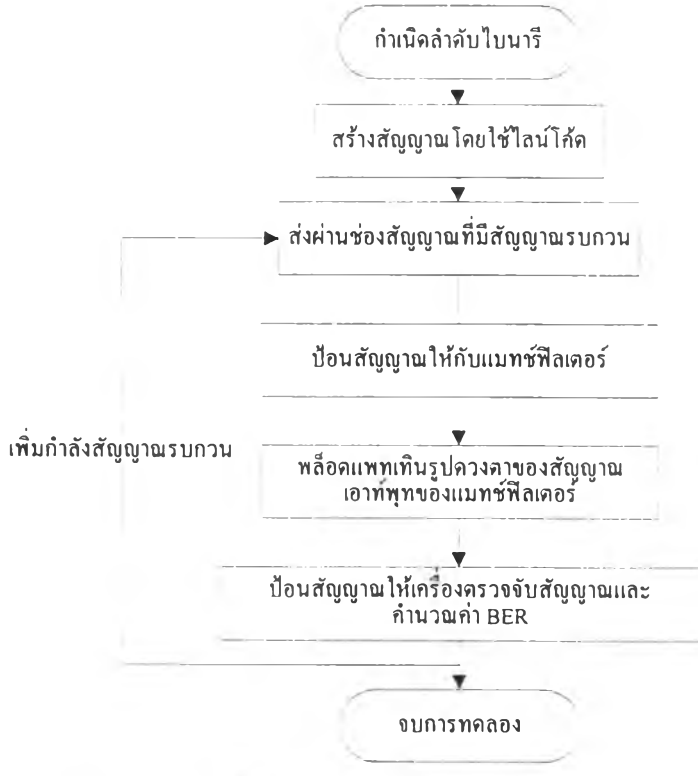
แผนผังงานการทดลองที่ 6 การตรวจจับสัญญาณ



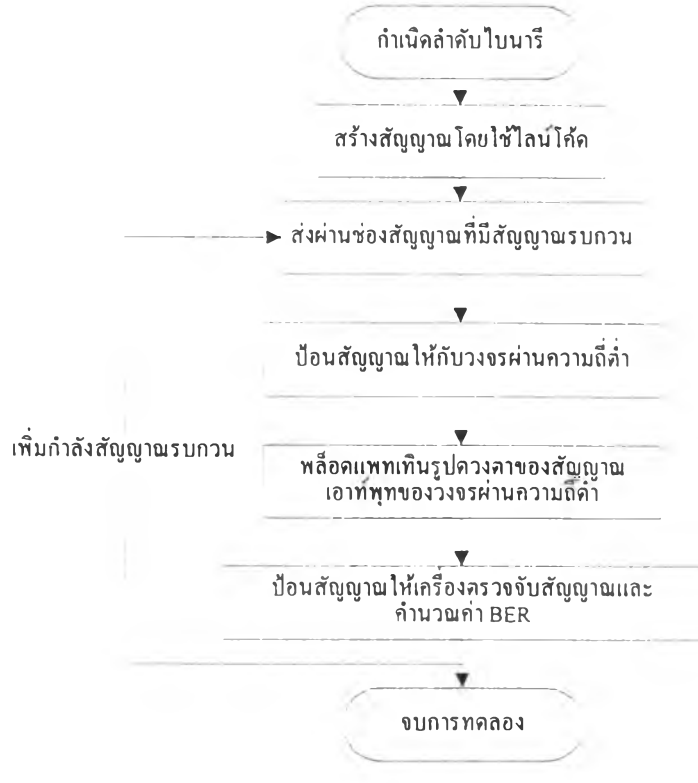
รูปที่ 21 แผนผังงานขั้นตอนการทดลองที่ 6.1 คุณลักษณะต่างๆ ของแมทช์ฟิลเตอร์



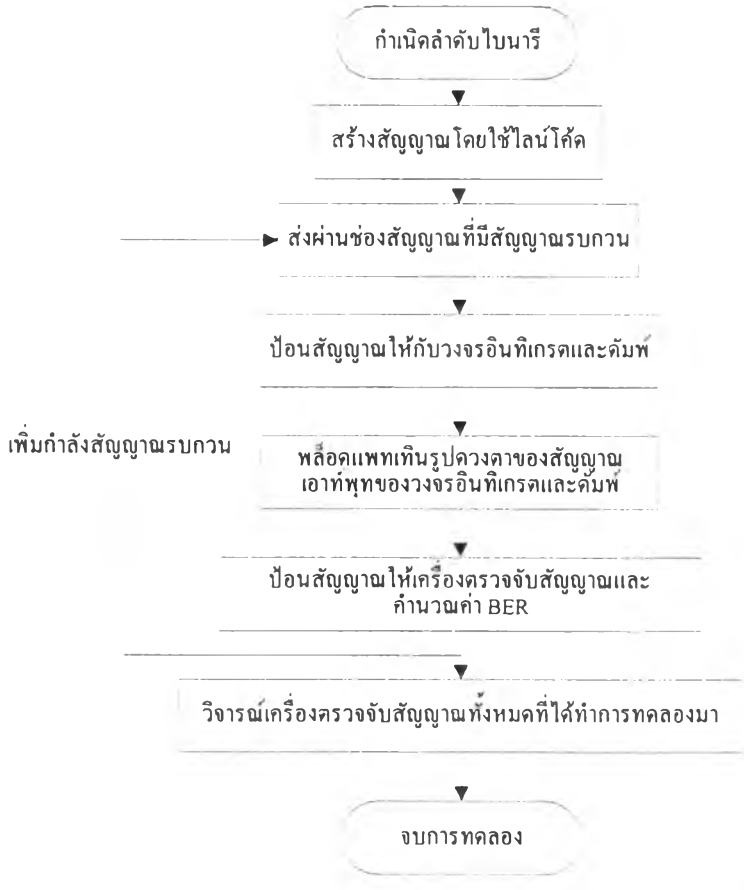
รูปที่ 22 แผนผังงานขั้นตอนการทดลองที่ 6.2 การตรวจจับสัญญาณ



รูปที่ 23 แผนผังงานขั้นตอนการทดลองที่ 6.3 เครื่องรับสัญญาณแมทช์ฟิลเตอร์

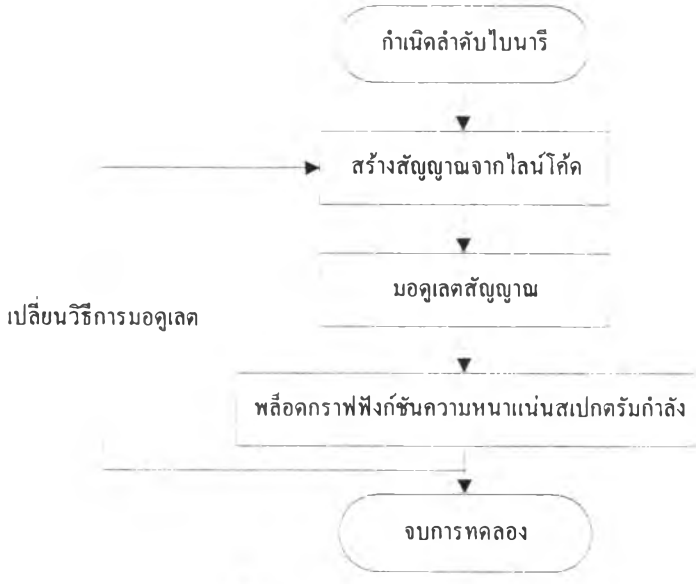


รูปที่ 24 แผนผังงานขั้นตอนการทดลองที่ 6.4 เครื่องรับสัญญาณวงจรผ่านความถี่ต่ำ

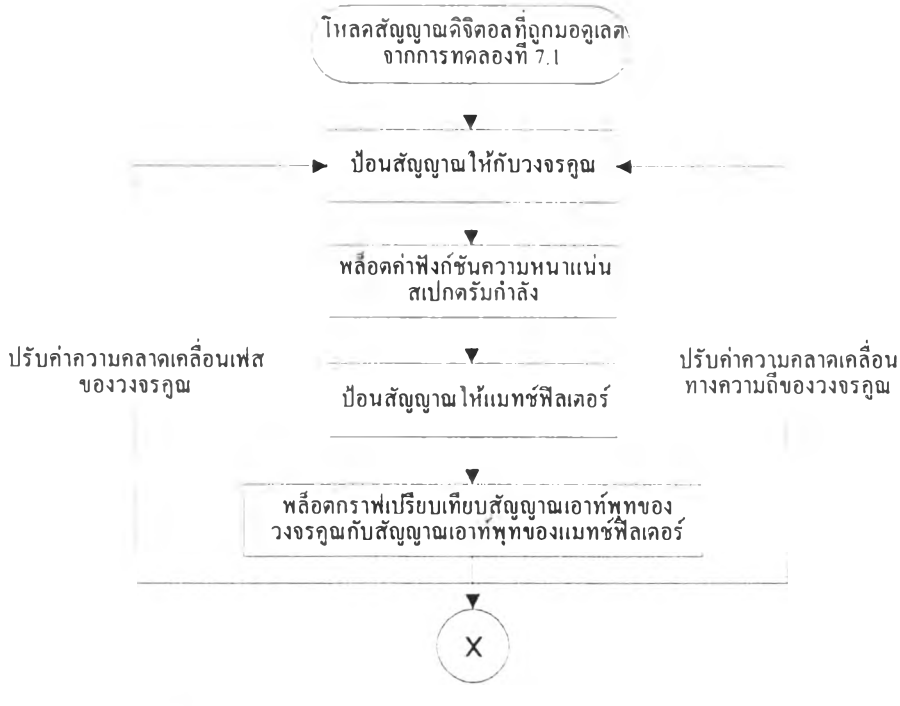


รูปที่ 25 แผนผังงานขั้นตอนการทดลองที่ 6.5 การประยุกต์ใช้งานแมทช์ฟิลเตอร์

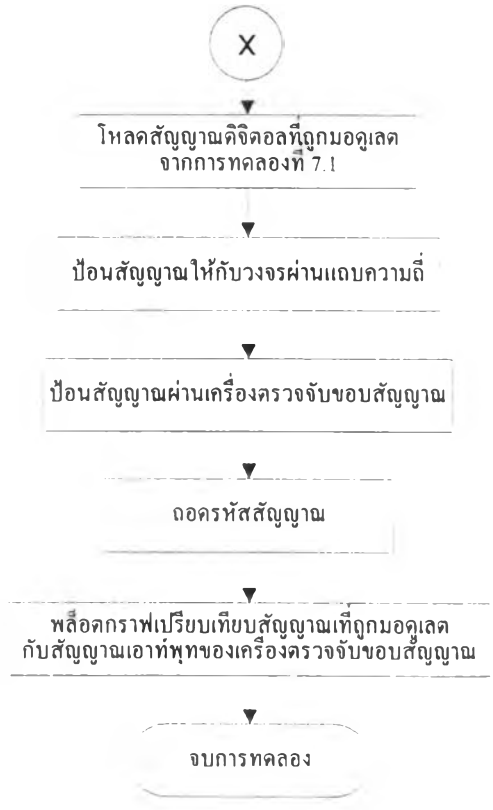
แผนผังงานการทดลองที่ 7 การมอดูเลตสัญญาณดิจิตอล



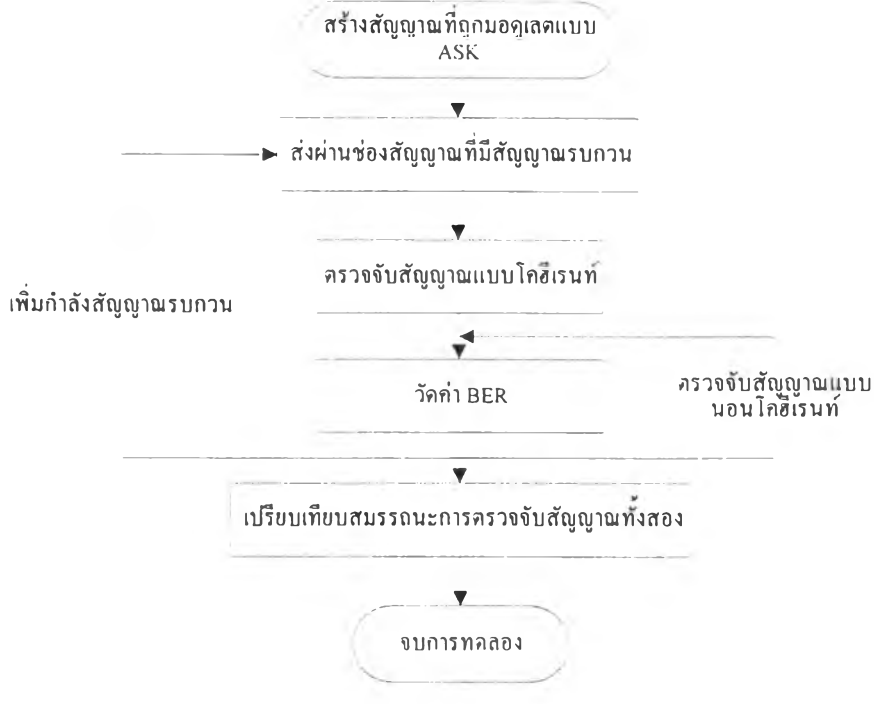
รูปที่ 26 แผนผังงานขั้นตอนการทดลองที่ 7.1 การมอดูเลตสัญญาณดิจิตอลแบบต่างๆ



รูปที่ 27 แผนผังงานขั้นตอนการทดลองที่ 7.2 การตรวจจับสัญญาณดิจิตอลที่ถูกมอดูเลต (โคฮีเรนท์)

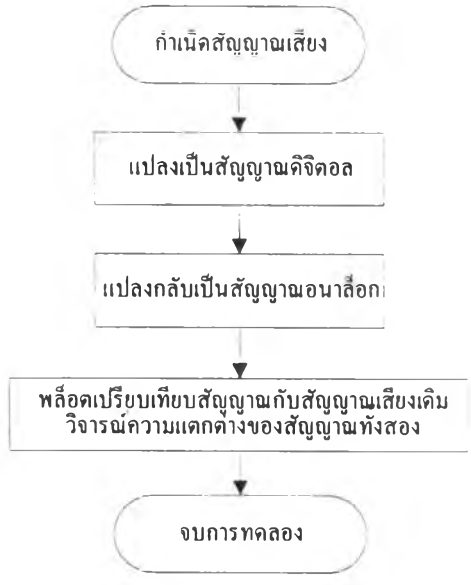


รูปที่ 28 แผนผังงานขั้นตอนการทดลองที่ 7.2 การตรวจจับสัญญาณดิจิตอลที่ถูกมอดูเลต (นอนโคฮีเรนท์)

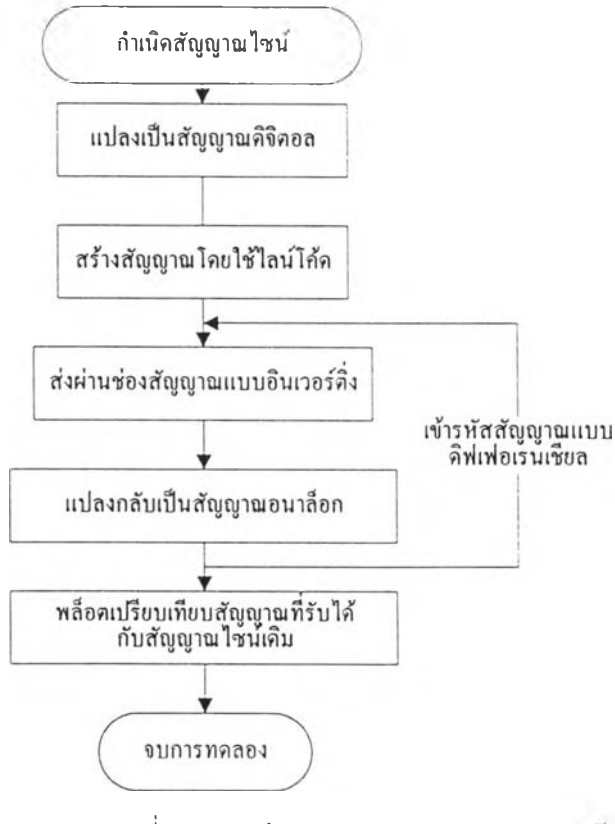


รูปที่ 29 แผนผังงานขั้นตอนการทดลองที่ 7.3 สมรรถนะของระบบในสภาวะมีสัญญาณรบกวน

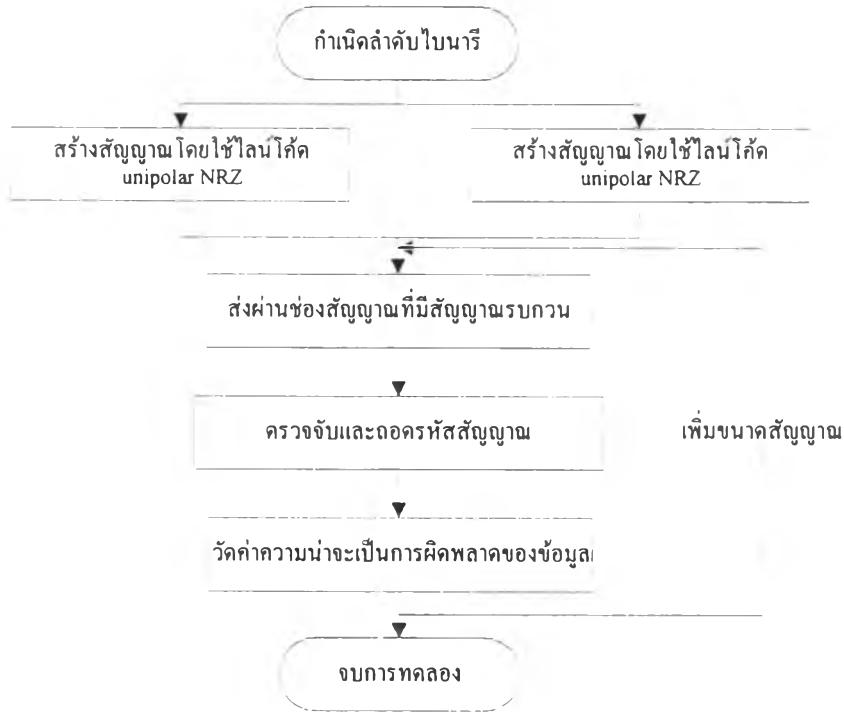
แผนผังงานการทดลองที่ 8 การสื่อสารแบบดิจิตอล



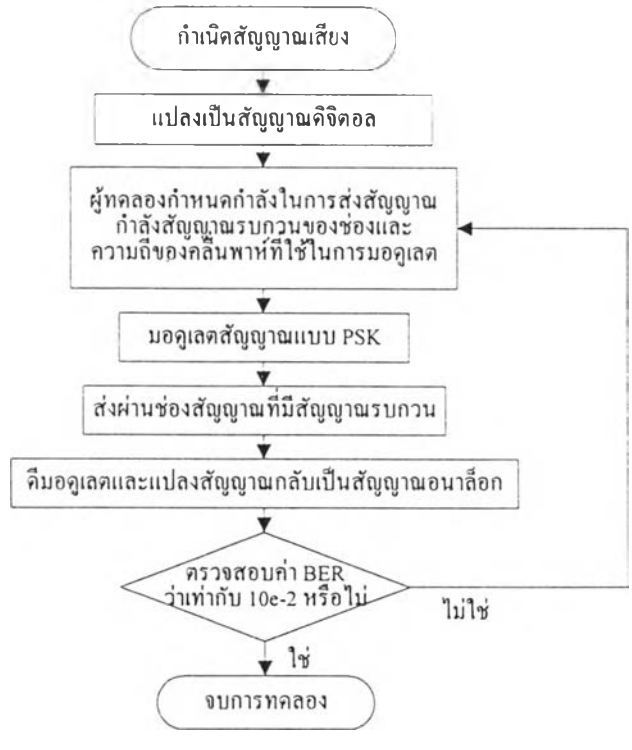
รูปที่ 30 แผนผังงานขั้นตอนการทดลองที่ 8.1 การแปลงสัญญาณระหว่างสัญญาณอนาล็อกและสัญญาณดิจิตอล



รูปที่ 31 แผนผังงานขั้นตอนการทดลองที่ 8.2 การเข้ารหัสสัญญาณแบบดิฟเฟอเรนเชียล



รูปที่ 32 แผนผังงานขั้นตอนการทดลองที่ 8.3 การสื่อสารที่ความถี่เบสแบนด์



รูปที่ 33 แผนผังงานขั้นตอนการทดลองที่ 8.4 การสื่อสารที่ความถี่แบนด์พาส

ภาคผนวก จ

รายละเอียดการทดลองของชุดทดลองด้านระบบสื่อสารแบบดิจิตอล

ความนำ

ในภาคผนวก จ นี้ แสดงรายละเอียดของชุดทดลอง เป็นการนำไฟล์เอกสารของชุดทดลองจริงๆ มาย่อแสดงให้เห็นถึงเนื้อหาและหน้าตาของชุดทดลองทั้งหมด โดยชุดทดลองประกอบด้วยการทดลองทั้งหมด 8 การทดลอง และแต่ละการทดลองประกอบด้วยการทดลองย่อยต่างๆ

ในส่วนท้ายของภาคผนวกนี้ ได้สรุปแสดงไดอะแกรมการเชื่อมโยงกันระหว่างไฟล์เอกสารต่างๆ ของชุดทดลองไว้ด้วย

ชุดทดลองด้านระบบสื่อสารแบบดิจิตอล

สารบัญการทดลอง

ทำความเข้าใจในการใช้ชุดทดลอง
การทดลองที่ 1 ทฤษฎีความน่าจะเป็น
การทดลองที่ 2 กระบวนการสุ่ม
การทดลองที่ 3 การวิเคราะห์สัญญาณและระบบเชิงเส้น
การทดลองที่ 4 การแปลงเป็นสัญญาณดิจิตอล
การทดลองที่ 5 รูปแบบการส่งสัญญาณไบนารี
การทดลองที่ 6 การตรวจจับสัญญาณ
การทดลองที่ 7 การมอดูเลตสัญญาณดิจิตอล
การทดลองที่ 8 การสื่อสารแบบดิจิตอล

รูปที่ 1 ไฟล์ commulab.doc

การใช้งานชุดทดลองด้านระบบสื่อสารแบบดิจิตอล

ชุดทดลองด้านระบบสื่อสารแบบดิจิตอลเป็นชุดทดลองที่อาศัยการทำงานร่วมกันระหว่างโปรแกรมในไมโครซอฟต์เวิร์ดภาษาไทยเวอร์ชัน 6.0a และโปรแกรม MATLAB เวอร์ชัน 4.2c1 โดยผู้ใช้สามารถรันคำสั่งของโปรแกรม MATLAB บนไฟล์เอกสารของโปรแกรมไมโครซอฟต์เวิร์ดได้ โดยที่ข้อความคำสั่งต่างๆ ที่ปรากฏอยู่ในไฟล์เอกสารมีรูปแบบแตกต่างจากข้อความปกติดังนี้

ตารางที่ 1 ลักษณะประเภทข้อความต่างๆ

ประเภทข้อความ	ตัวอย่าง
ข้อความคำสั่ง (input cell)	[b = binary(2000);]
ข้อความอัตโนมัติ (autoinit cell)	[init6_3;]
ข้อความแจ้งความผิดพลาด (error cell)	
ข้อความแสดงผลลัพธ์ (output cell)	[a = 10000]

จากตารางที่ 1 สังเกตว่าข้อความทุกประเภทที่ติดต่อกับโปรแกรม MATLAB จะ ถูกล้อมรอบด้วยวงเล็บก้ามปู [...]

สำหรับส่วนของทูลบาร์ของชุดทดลอง มีรายละเอียดการใช้งานดังต่อไปนี้

ตารางที่ 2 ความหมายของไอคอนต่างๆ ของทูลบาร์

รูป icon	คำอธิบาย
[redacted]	ประมวลคำสั่ง input cell ที่เลือกไว้
[redacted]	เลือกโหมดการแสดงผลทางกราฟฟิก
[redacted]	เรียกหน้าต่าง Figure ของ MATLAB
	กลับไปยังไฟล์เอกสารที่แล้ว
	ตั้งค่า path ที่เก็บไฟล์ *.m
	เรียกโปรแกรมแสดงและแก้ไขตัวแปรของ MATLAB
	เรียกโปรแกรมความช่วยเหลือของทูลบ็อกซ์ด้านระบบสื่อสาร
	เรียกโปรแกรมความช่วยเหลือของ MATLAB
	เรียกโปรแกรมความช่วยเหลือของ Notebook

การรันคำสั่ง

การรันคำสั่ง MATLAB จากไฟล์เอกสารสามารถกระทำได้ดังนี้ คือ

1. เลือกข้อความคำสั่ง input cell ที่ต้องการ โดยเลื่อนเคอร์เซอร์ไปที่คำสั่งนั้น
2. กดปุ่ม "ประมวลผลคำสั่ง" ที่ทูลบาร์ของชุดทดลอง หรือ กดปุ่ม Ctrl-Enter

การสร้าง input cell

การสร้าง input cell บนไฟล์เอกสารสามารถกระทำได้ดังนี้ คือ

1. พิมพ์คำสั่งที่ต้องการ เสร็จแล้วอย่ากดปุ่ม Enter
2. กดปุ่ม Ctrl-Enter

รูปที่ 2 ไฟล์ tutorial.doc

การทดลองที่ 1 ทฤษฎีความน่าจะเป็น

วัตถุประสงค์
หนังสืออ้างอิง
กลับไปยังสารบัญการทดลอง

สารบัญการทดลองย่อย

การทดลองที่ 1.1 ตัวแปรสุ่มแบบไม่ต่อเนื่อง
การทดลองที่ 1.2 ตัวแปรสุ่มแบบต่อเนื่อง
การทดลองที่ 1.3 ค่าเฉลี่ย ความแปรปรวนและกำลังงาน
การทดลองที่ 1.4 เกมน่าเป็ว
การทดลองที่ 1.5 ฟังก์ชัน pdf กับการประมวลผลภาพ

วัตถุประสงค์

ในการทดลองนี้ จะศึกษา

- คุณลักษณะต่างๆ ของตัวแปรสุ่มทั้งแบบต่อเนื่องและไม่ต่อเนื่อง
- ตัวแปรสุ่มชนิดยูนิฟอร์มและเกาส์เสียน โดยพิจารณาจากฟังก์ชันความหนาแน่นความน่าจะเป็น (PDF) และฟังก์ชันการแจกแจงความน่าจะเป็น(CDF)
- การวัดค่าทางสถิติต่างๆ คือ ค่าเฉลี่ย (mean) แวเรียนซ์ (variance) และค่ากำลังสองเฉลี่ยของเพาเวอร์ (mean-square power)

หนังสืออ้างอิง

1. "หลักการระบบสื่อสาร", รศ.ดร. ประสิทธิ์ ประพิณมงคลการ บริษัท ซีเอ็ดยูเคชั่น จำกัด หน้า 38-46
2. "หลักการไฟฟ้าสื่อสาร", รศ.ดร.บัณฑิต ไวจน์อารยานนท์ สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย หน้า 65-92
3. "Communication Systems," Third edition., A.B.Carlson, McGraw-Hill,1986 หน้า 118-147
4. "Digital and Analog Communication Systems," Third edition., L.W. Couch II., MacMillan,1990 หน้า 660-684
5. "An Introduction to Analog & Digital Communications.", S.Haykin, Wiley & Sons, 1989. หน้า 404-430
6. "Digital Image Processing.", Rafael C. Gonzalez, Richard E. Woods, Addison Wesley, 1992 หน้า 173-180

รูปที่ 3 ไฟล์ lab1.doc

1.1 ตัวแปรสุ่มแบบไม่ต่อเนื่อง

ลำดับแรก กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[init1_1; ]
```

เราจะใช้ฟังก์ชัน dice ในการสุ่มตัวอย่างที่เกิดจากการทดลอง โยนลูกเต๋าทั้งหมด 2000 ครั้ง โดยแต่ละหน้าของลูกเต๋าไม่มีการถ่วงน้ำหนัก

```
[ x = dice(2000,6,'fair'); ]
```

คำนวณพร้อมทั้งพล็อตค่าฟังก์ชันความหนาแน่นความน่าจะเป็น (pdf) และฟังก์ชันการแจกแจง (cdf) ของลำดับ x

```
[ subplot(121), pdf(x)
  subplot(122), cdf(x) ]
```

คำถาม ฟังก์ชัน pdf และ cdf มีความหมายอย่างไร และมีความสัมพันธ์กันอย่างไร

คำถาม จากรูปที่ได้ ทำไมความน่าจะเป็นของการทดลองโยนลูกเต๋าที่ได้จึงไม่เท่ากันเป็นไปตามที่คาดไว้หรือไม่ และถ้าไม่ ทำอย่างไรจึงจะทำให้ได้ค่าตามที่คาดหวังไว้

ต่อไป เราจะใช้ฟังก์ชัน dice ในการสุ่มตัวอย่างเหตุการณ์ที่เกิดจากการทดลองโยนลูกเต๋าทั้งหมด 2000 ครั้งอีกที แต่คราวนี้ ลูกเต๋าจะถูกถ่วงน้ำหนักแต่ละหน้าไม่เท่ากัน

```
[ y = dice(2000,6,'biased'); ]
```

ลบรูปที่พล็อตครั้งที่แล้ว และพล็อตฟังก์ชัน pdf ของลำดับ y ด้วยคำสั่ง pdf

```
[ clf, pdf(y) ]
```

วาดรูปฟังก์ชัน cdf จากกราฟฟังก์ชัน pdf ที่ได้ข้างบนและเราสามารถเช็คดูความถูกต้องได้จากการใช้ฟังก์ชัน cdf

คำถาม จากการทดลองโยนลูกเต๋าทั้งหมด 2000 ครั้ง เหตุการณ์ที่ออกแต่ละหน้าของลูกเต๋ามีจำนวนครั้งประมาณกี่ครั้ง

รูปที่ 4 ไฟล์ lab1_1.doc

1.2 ตัวแปรสุ่มแบบต่อเนื่อง

ตัวแปรสุ่มชนิดเกาส์เซียน (นอร์มอล)

ตัวแปรสุ่มชนิดยูนิฟอร์ม

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[init1_2;]
```

ในการทดลองนี้ เราจะใช้ฟังก์ชัน unif_pdf และ unif_cdf พล็อตกราฟฟังก์ชัน pdf และ cdf ของตัวแปรสุ่มชนิดยูนิฟอร์ม U(2,6)

```
[subplot(121), unif_pdf(2,6), axis([0, 8, -0.2, 1.2]);
 subplot(122), unif_cdf(2,6), axis([0, 8, -0.2, 1.2]);]
```

คำถาม ถ้าให้ตัวแปรสุ่ม U ~ U(2,6) จงหาค่าความน่าจะเป็นเพื่อนำมาใส่ลงในกราฟของฟังก์ชัน cdf และ pdf

ตารางที่ 1.1

P(0 < U ≤ 3)	P(3 < U ≤ 5)	P(U = 3)

คำถาม ทำไมค่าความน่าจะเป็น P(U = 3) จึงต่างไปจาก P(X = 3) ในการทดลองที่ 1.1

ต่อไป ทดลองคำนวณกับค่าสถิติที่ได้จากการสุ่มตัวอย่างจริงๆ ดูกำเนิดตัวแปรสุ่ม u ซึ่งมีการกระจายแบบยูนิฟอร์มในช่วง (2,6) จำนวน 500 ตัวอย่าง

```
[u = uniform(2,6,500);]
```

คำนวณหาค่าเฉลี่ยและความแปรปรวนของตัวแปรสุ่ม u

```
[mean_u = mean(u), var_u = var(u)]
```

คำถาม ค่า mean_u และ var_u ที่ได้กับค่าที่คำนวณตามทฤษฎี มีค่าเท่ากันหรือไม่ ทำไมจึงเป็นเช่นนั้น

ตัวแปรสุ่มชนิดเกาส์เซียน (นอร์มอล)

สำหรับตัวแปรสุ่มชนิดเกาส์เซียน เราจะใช้ gauss_pdf และ gauss_cdfในการพล็อตกราฟฟังก์ชันความหนาแน่นความน่าจะเป็นและฟังก์ชันการแจกแจงความน่าจะเป็น ตามลำดับ โดยกรณีนี้ เราจะใช้ตัวแปรสุ่มชนิดเกาส์เซียนที่ค่าเฉลี่ยเท่ากับ mean_u และ ค่าความแปรปรวนเท่ากับ var_u ดังนี้

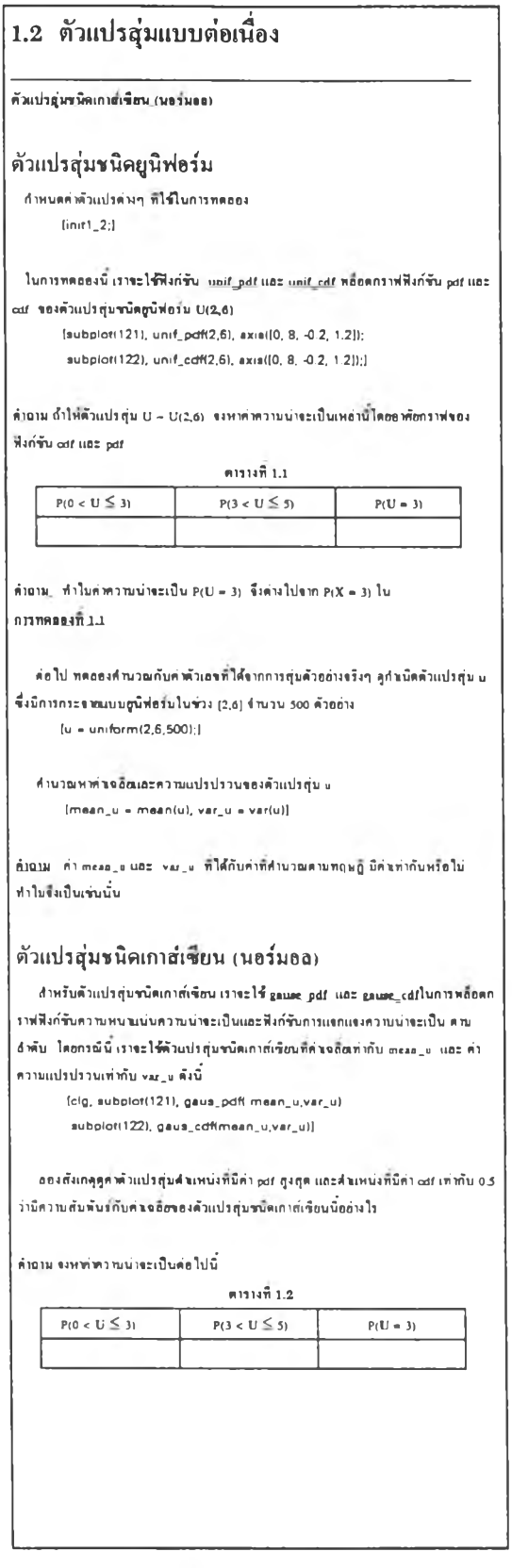
```
[clg, subplot(121), gaus_pdf( mean_u,var_u)
 subplot(122), gaus_cdf(mean_u,var_u)]
```

ลองสังเกตดูค่าตัวแปรสุ่มตำแหน่งที่มีค่า pdf สูงสุด และตำแหน่งที่มีค่า cdf เท่ากับ 0.5 ว่ามีความสัมพันธ์กับค่าเฉลี่ยของตัวแปรสุ่มชนิดเกาส์เซียนนี้อย่างไร

คำถาม จงหาค่าความน่าจะเป็นต่อไปนี้

ตารางที่ 1.2

P(0 < U ≤ 3)	P(3 < U ≤ 5)	P(U = 3)



รูปที่ 5 ไฟล์ lab1_2.doc

จะเห็นว่าตัวแปรสุ่ม G ในตาราง 1.2 และตัวแปรสุ่ม U ในตาราง 1.1 มีค่าเฉลี่ยและความแปรปรวนค่าเดียวกัน จงเปรียบเทียบค่าในตาราง 1.2 กับค่าในตาราง 1.1 พร้อมทั้งอธิบายความแตกต่าง

จั่นต่อไป เราจะสังเกตดูผลกระทบของค่าความแปรปรวน σ^2 ที่มีต่อตัวแปรสุ่มชนิดเกาส์เซียน โดยกำหนดให้ตัวแปรสุ่ม $X \sim N(\mu,\sigma^2)$ สมมติให้ X มีค่า $\mu = 1$ คงที่ และค่าความแปรปรวนเปลี่ยนไปดังนี้ $\sigma^2 \in \{0.5, 1, 2, 5, 10\}$

```
[clf
 m = 1; gaus_pdf(m,0.5)
 axis([-10 10 0 0.6]), hold on]
[gaus_pdf(m,σ2)]
```

คำถาม ถ้าให้เหตุการณ์ $A = \{0 < X < 2\}$ โดยที่ตัวแปรสุ่ม $X \sim N(1,\sigma^2)$ ซึ่ง $\sigma^2 \in \{0.5, 1, 2, 5, 10\}$ จงหาว่า σ^2 ที่ค่าใดจึงจะทำให้ P(A) มีค่ามากที่สุด

ต่อไป ลองเปลี่ยนจากฟังก์ชัน pdf เป็น cdf และเปลี่ยนค่า σ^2 ดูบ้าง

```
[clf
 m = 1; gaus_cdf(m,10)
 axis([-10 10 0 1]), hold on]
[gaus_cdf(m,σ2)]
```

คำถาม ถ้าค่า σ^2 มีค่าน้อยมากๆ กราฟฟังก์ชัน cdf จะมีรูปร่างเป็นอย่างไร (ตัวอย่างเช่น σ^2 ประมาณ 0.00001)

การที่จะศึกษาความหนาแน่นของตัวแปรสุ่มซึ่งมีค่าความแปรปรวนน้อยมากๆ อาจสังเกตได้จากฟังก์ชัน pdf ดังนี้

```
[clf
 unif_pdf(m,0.00001)
 axis([0 2 0 200]) ]
```

คำถาม จากรูปคำสั่งข้างบน ถ้าเปลี่ยนจากตัวแปรสุ่มชนิดเกาส์เซียนเป็นตัวแปรสุ่มชนิดยูนิฟอร์ม กราฟฟังก์ชัน pdf ที่ได้ จะแตกต่างจากเดิมหรือไม่

ต่อไป เราสังเกตดูผลกระทบของค่าเฉลี่ย (μ) ที่มีต่อตัวแปรสุ่มชนิดเกาส์เซียน เมื่อกำหนดความแปรปรวนคงที่เท่ากับ 1 และเปลี่ยนค่า μ ดังนี้ $\mu \in \{-4, -1, 2, 5\}$

```
[clf
 s=1; gaus_pdf(-4,s)
 axis([-8 8 0 0.5]), hold on]
[gaus_pdf(μ,s)]
```

คำถาม ถ้ากำหนดให้ $X(\mu,\sigma^2) \sim N(\mu,\sigma^2)$ แล้ว ความน่าจะเป็น $P(-5 < X(-4,1) < -3)$ มีค่าเท่ากันกับ $P(4 < X(5,1) < 6)$ หรือไม่

จากการสังเกตดูฟังก์ชัน pdf เราลองเปลี่ยนมาสังเกตดูฟังก์ชัน cdf แทนว่ามีการเปลี่ยนแปลงอย่างไร เมื่อค่า μ เปลี่ยนไป

```
[clf
 s=1; gaus_cdf(-4,s)
 axis([-8 8 0 1]), hold on ]
[gaus_cdf(μ,s)]
```

รูปที่ 6 ไฟล์ lab1_2.doc (ต่อ)

1.3 ค่าเฉลี่ย ความแปรปรวนและกำลังงาน

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init1_3; ]
```

กำเนิดลำดับของตัวแปรสุ่มที่มีค่าเฉลี่ยต่างๆ กัน โดยใช้ฟังก์ชัน gauss ดังนี้

```
[ x = gauss(-5,1,100);
  y = gauss(0,1,100);
  z = gauss(5,1,100);
  clf
  plot(x)
  axis([1 100 -10 10]), grid on, hold on ]
[ plot(y) ]
[ plot(z) ]
```

คำถาม ค่าเฉลี่ยมีความสัมพันธ์อย่างไรกับระดับ dc ของรูปคลื่นสัญญาณ

ต่อไป กำเนิดลำดับของตัวแปรสุ่มชนิดเกาส์เซียนซึ่งมีค่าความแปรปรวนต่างๆ กัน

```
[ a = gauss(0, 4, 100);
  b = gauss(0, 1, 100);
  c = gauss(0, 0.5, 100);
  d = gauss(0, 0.01, 100);
  clf
  subplot(221), plot(a), axis([1 100 -10 10]) ]
[ subplot(222), plot(b), axis([1 100 -10 10]) ]
[ subplot(223), plot(c), axis([1 100 -10 10]) ]
[ subplot(224), plot(d), axis([1 100 -10 10]) ]
```

คำสั่ง ใช้ฟังก์ชัน mean และ var หาค่าเฉลี่ยและค่าความแปรปรวนของแต่ละลำดับสุ่มและใส่ค่าลงในตาราง 1.3

ตาราง 1.3

ลำดับสุ่ม	E(x)	variance	$E(x^2)$
a			
b			
c			
d			

*** เราสามารถเช็คค่า $E(x^2)$ ได้จากฟังก์ชัน meansq

คำถาม ถ้าให้รูปคลื่นต่างๆ ที่พล็อตในการทดลองนี้ เสมือนเป็นสัญญาณรบกวนรูปคลื่นรูปไหน ที่น่าจะพบเจอได้ในระบบสื่อสาร และ ถ้าจะให้รูปคลื่นเหล่านี้แทนรูปคลื่นของสัญญาณที่ปราศจากสัญญาณรบกวน รูปคลื่นนั้นน่าจะเป็นรูปไหน

รูปที่ 7 ไฟล์ lab1_3.doc

1.4 เกมปาเป้า

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init1_4; ]
```

วัตถุประสงค์ของเกมนี้คือต้องการแสดงให้เห็นถึงความหมายและความสำคัญของค่าเฉลี่ยและความแปรปรวน โดยจะใช้ฟังก์ชัน dart ทำของเหตุการณ์ในการปาเป้าวงกลมโดยที่เราสามารถกำหนดความสามารถในการปาเป้าของผู้ปาได้โดยกำหนดจากค่าเฉลี่ยและความแปรปรวนของตำแหน่งที่ลูกดอกปักบนแผ่นเป้า

ต่อไปนี้เป็นการกำหนดความสามารถในการปาเป้าของผู้เล่นคนหนึ่ง

```
[ mean_x = 0.2; mean_y = 0.2;
  var_x = 0.1; var_y = 0.1;
  no_dart = 20;
  clf
  dart( [mean_x mean_y], [var_x var_y], no_dart ) ]
```

ลองเปลี่ยนค่าเฉลี่ยและความแปรปรวน ในการปาเป้าเป็นค่าอื่นๆ และสังเกตุดูผลที่ได้โดยเฉพาะกรณีที่ค่าความแปรปรวนในแกนหนึ่งหรือทั้งสองแกนมีค่าเป็นศูนย์

คำถาม พิจารณาความสามารถในการปาเป้าของผู้เล่น 2 คนนี้

ผู้เล่นคนที่ 1 : [mean_x mean_y] = [0 0]
[var_x var_y] = [0.5 0.5]

ผู้เล่นคนที่ 2 : [mean_x mean_y] = [0.5 0.5]
[var_x var_y] = [0.01 0.01]

ผู้เล่นทั้งสองคนใครปาเป้าเก่งกว่ากัน และวิจารณ์ความสามารถในการปาเป้าของผู้เล่นทั้งสองคนนี้

รูปที่ 8 ไฟล์ lab1_4.doc

1.5 ฟังก์ชัน pdf กับการประมวลผลภาพ

ความรู้เบื้องต้นเกี่ยวกับการประมวลผลภาพแบบดิจิตอล
การปรับปรุงคุณภาพของรูปภาพด้วยวิธี Histogram Equalization

ในการทดลองนี้ จะกล่าวถึงขบวนการทางดิจิตอลในการปรับปรุงคุณภาพของรูปภาพ โดยใช้วิธีที่เรียกว่า **Histogram equalization**

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init1_5; ]
```

ลำดับแรก โหลดรูปภาพที่เก็บอยู่ในไฟล์ชื่อ dummy.raw ซึ่งเป็นภาพขาวดำ (gray scale) ขนาด 125x200 pixels มีระดับสีทั้งหมด 256 สี และเก็บไว้ในเมตริกซ์ p

```
[ p = openfile('dummy.raw'); ]
```

พล็อต histogram ของภาพ dummy.raw โดยใช้ฟังก์ชัน mypdf และสังเกตุดูการกระจายของจุด pixel ที่ความเข้มต่างๆ

```
[ subplot(211),
 mypdf(p,50),
 axis([0 255 0 0.06]),
 a = axis;
 grid ]
```

คำถาม จากฟังก์ชัน pdf ที่ได้ จะสามารถทำนายได้หรือไม่ว่ารูปภาพในไฟล์ dummy.raw มีลักษณะเป็นอย่างไร

พล็อตภาพในไฟล์ dummy.raw ดูว่าเป็นไปตามที่ทำนายไว้หรือไม่

```
[ figure(2)
 subplot(121)
 image(p)
 colormap(gray(256))
 axis('image') ]
```

คำถาม ภาพที่พล็อตได้ คุณภาพของภาพดีหรือไม่ สอดคล้องกับฟังก์ชัน pdf ที่ได้หรือไม่ และสามารถบอกได้หรือไม่ว่าเป็นภาพของอะไร

คำถาม จงบอกว่าภาพ dummy และฟังก์ชัน pdf ของมัน องค์ประกอบส่วนไหนที่เป็นสาเหตุที่ทำให้ภาพมีคุณภาพไม่ดี

ต่อไป เราจะทำการปรับปรุงคุณภาพของรูปภาพ dummy ให้ดีขึ้น โดยใช้วิธี Histogram equalization แล้วเก็บภาพไว้ในเมตริกซ์ p_new

```
[ p_new = hist_eq(p); ]
```

พล็อตฟังก์ชัน pdf เทียบกับของภาพในเมตริกซ์เดิม และสังเกตุความแตกต่าง

```
[ figure(1)
  subplot(212)
  mypdf(p_new,50)
  axis(a)
  grid ]
```

คำถาม จากฟังก์ชัน pdf ที่เปลี่ยนไปนี้ จะมีแนวโน้มทำให้ภาพ dummy เปลี่ยนไปอย่างไร เพราะเหตุใด

พล็อตภาพจากเมตริกซ์ p_new เทียบกับภาพ dummy เดิม

```
[ figure(2)
  subplot(122)
  image(p_new)
  colormap(gray(256))
  axis('image') ]
```

รูปที่ 9 ไฟล์ lab1_5.doc

การทดลองที่ 2 กระบวนการสุ่ม

วัตถุประสงค์
หนังสืออ้างอิง
กลับไปยังสารบัญการทดลอง

สารบัญการทดลองย่อย

การทดลองที่ 2.2 ความเป็นสเตชันนารีในมุมกว้างและเออร์กอดิก
การทดลองที่ 2.2 ฟังก์ชันคอร์รีเลชันและความหนาแน่นความน่าจะเป็น
การทดลองที่ 2.3 สัญญาณรบกวนแบบไวท์

วัตถุประสงค์

ศึกษาพฤติกรรมและคุณสมบัติต่างๆ ของสัญญาณสุ่ม ได้แก่

- ความเป็นสเตชันนารีในมุมกว้างและความเป็นเออร์กอดิก (Wide-sense stationarity and Ergodicity)
- ฟังก์ชันคอร์รีเลชัน (Correlation) เครื่องมือวัดความสัมพันธ์ระหว่างกระบวนการสุ่ม 2 กระบวนการ
- ความสัมพันธ์ระหว่างกระบวนการสุ่ม ฟังก์ชันออโตคอร์รีเลชัน และฟังก์ชันความหนาแน่นสเปกตรัม

หนังสืออ้างอิง

1. "หลักการระบบสื่อสาร", รศ.ดร. ประสิทธิ์ ประพิณมงคลการ บริษัท ซีเอ็ดยูเคชั่น จำกัด หน้า 24-32
2. "หลักการไฟฟ้าสื่อสาร", รศ.ดร บัณฑิต ไรฉน์อารยานนท์ สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย หน้า 93-108
3. "Communication Systems," Third edition., A.B.Carlson, McGraw-Hill,1986. หน้า 152-180
4. "Digital and Analog Communication Systems," Third edition., L.W. Couch II., MacMillan,1990 หน้า 447-477
5. "An Introduction to Analog & Digital Communications,", S.Haykin, Wiley & Sons, 1989 หน้า 334-364

รูปที่ 10 ไฟล์ lab2.doc

2.1.1 ความเป็นสเตชันนารีและเออร์กอดิก

2.1.2 การหาค่าเฉลี่ยเชิงสถิติโดยใช้ค่าเฉลี่ยของเอ็นเซมเบิล
2.1.3 การหาค่าเฉลี่ยทางสถิติโดยใช้ค่าเฉลี่ยทางเวลา
2.1.4 การคำนวณทางทฤษฎี
2.1.5 กระบวนการสุ่มแบบเออร์กอดิก ฟังก์ชันความหนาแน่นความน่าจะเป็น

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init2_1; ]
```

ในการทดลองนี้ เราจะสร้างตัวอย่างของกระบวนการสุ่มขึ้นมา 3 กระบวนการ เพื่อใช้ในการศึกษาสภาวะสเตชันนารีและเออร์กอดิก โดยใช้ฟังก์ชัน MATLAB คือ rp1(.), rp2(.) และ rp3(.) กำเนิดกระบวนการสุ่มทั้งสาม

```
[ v1 = rp1(4,100);
  v2 = rp2(4,100);
  v3 = rp3(4,100); ]
```

คำสั่งข้างบน เป็นการสร้างกระบวนการสุ่ม v1, v2 และ v3 โดยแต่ละกระบวนการประกอบด้วยสัญญาณตัวอย่าง 4 สัญญาณ และแต่ละสัญญาณมีจำนวน 100 ค่า พล็อตกราฟของสัญญาณทั้งสี่ในแต่ละกระบวนการสุ่มและสังเกตดูผลที่ได้

```
[ subplot(411),plot(v?(1,:))
  subplot(412),plot(v?(2,:))
  subplot(413),plot(v?(3,:))
  subplot(414),plot(v?(4,:)) ]
```

คำถาม กระบวนการสุ่มทั้งสาม มีความเป็นสเตชันนารีและเออร์กอดิกหรือไม่

กลับไปที่ส่วนบนสุด

2.1.2 การหาค่าเฉลี่ยเชิงสถิติโดยใช้ค่าเฉลี่ยของเอ็นเซ็มเบิล

กลับไปที่ส่วนบนสุด

ในการหาค่าเฉลี่ยออนซอมเบิล จำเป็นต้องใช้จำนวนสัญญาณตัวอย่างจำนวนมาก สำหรับในแต่ละกระบวนการสุ่ม ซึ่งในที่นี้กำหนดไว้ให้ 1000 สัญญาณ และแต่ละสัญญาณมีขนาด 100 ค่า

กำเนิดกระบวนการสุ่ม 3 กระบวนการคือ v1, v2 และ v3 ดังนี้

```
[ u1 = rp1(1000,50);
  u2 = rp2(1000,50);
  u3 = rp3(1000,50); ]
```

จงหาค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน (standard deviation) โดยใช้ฟังก์ชัน mean(.) และ std(.) และพล็อตค่าเหล่านี้เทียบกับเวลา พิจารณาอีกครั้งหนึ่งว่าแต่ละกระบวนการสุ่มมีความเป็นสเตชันนารีและเออร์กอดิกหรือไม่

```
[ mu = mean(u?);
  plot(mu), axis([0 50 0 1]), grid ]

[ su = std(u?);
  plot(su), axis([0 50 0 1.5]), grid ]
```

คำถาม พิจารณาอีกครั้งว่า กระบวนการสุ่มแต่ละกระบวนการ มีความเป็นสเตชันนารีและเออร์กอดิกหรือไม่

กลับไปที่ส่วนบนสุด

รูปที่ 11 ไฟล์ lab2_1 doc

2.1.3 การหาค่าเฉลี่ยทางสถิติโดยใช้ค่าเฉลี่ยทางเวลา

กลับไปที่ส่วนบนสุด

ในทางทฤษฎี การหาค่าเฉลี่ยทางเวลาให้ได้ค่าที่ถูกต้องนั้น ขนาดของสัญญาณจะต้องมีจำนวนเข้าใกล้อนันต์ แต่ในการทดลองนี้ จะใช้ขนาดของสัญญาณ N = 1000 ซึ่งถือว่าเพียงพอต่อการคู่เข้าของค่าเฉลี่ย

กำหนดกระบวนการสุ่ม v1, v2 และ v3

```
[ x1 = rp1(4,1000);
  x2 = rp2(4,1000);
  x3 = rp3(4,1000); ]
```

หาค่าเฉลี่ยทางเวลา $\langle V_\lambda \rangle$ โดย λ = 1, 2, 3, 4 สำหรับสัญญาณต่างๆในแต่ละกระบวนการสุ่ม

```
[ mx? = mean(x?') ]
```

พิจารณาอีกครั้งว่า แต่ละกระบวนการสุ่มมีความเป็นสเตชันนารีและเออร์กอดิกหรือไม่ (ใช้ผลที่ได้จาก การทดลองที่ 2.1.2 ช่วยในการวิเคราะห์ ถ้าจำเป็น)

กลับไปที่ส่วนบนสุด

2.1.4 การคำนวณทางทฤษฎี

จงวิเคราะห์คำสั่งต่างๆ ของ MATLAB ในฟังก์ชัน rp*.m โดยใช้คำสั่ง type และเขียนสูตรทางคณิตศาสตร์ของแต่ละกระบวนการสุ่ม และถ้าเป็นไปได้ หาสูตรของฟังก์ชัน pdf ของแต่ละกระบวนการสุ่มมาด้วย

คำนวณหาค่าเฉลี่ยเชิงสถิติ (Expected value) และ แวเรียนซ์ของแต่ละกระบวนการสุ่ม สุดท้ายพิจารณาว่าแต่ละกระบวนการสุ่มมีความเป็นสเตชันนารีและเออร์กอดิกหรือไม่

```
[ type rp*.m ]
```

ค่าถาม กำหนดให้กระบวนการสุ่ม $Y(\phi,t) = \sin(\Omega t + \phi)$ โดยที่ $\phi \in [\,0,\ \pi/2,\ \pi,\ 3\pi/2\,]$ กระบวนการสุ่มนี้มีความเป็นสเตชันนารีและเออร์กอดิกหรือไม่

กลับไปที่ส่วนบนสุด

2.1.5 กระบวนการสุ่มแบบเออร์กอดิกและฟังก์ชันหนาแน่นความน่าจะเป็น

สำหรับกระบวนการสุ่มแบบเออร์กอดิก เราสามารถหาฟังก์ชัน pdf ของกระบวนการสุ่มได้จากการหาค่าเฉลี่ยทางเวลา จงอธิบายความสัมพันธ์ระหว่างฟังก์ชัน pdf กับค่าเฉลี่ยทางเวลาของกระบวนการสุ่มเออร์กอดิกจากกระบวนการสุ่ม rp*.m ที่เป็นเออร์กอดิก กำเนิดสัญญาณให้มีขนาด N = 100, 1000 และ 8000 ตามลำดับ แล้วพล็อตฟังก์ชัน pdf โดยประมาณของกระบวนการสุ่มนั้นโดยใช้ฟังก์ชัน mypdf.)

```
[ v = rp?(1,N);
  subplot(31?)
  pdf(v) ]
```

พล็อตกราฟทั้งสามกรณีด้วยฟังก์ชัน subplot เปรียบเทียบกราฟทั้งสามและสังเกตุการคู่เข้าของฟังก์ชัน pdf เมื่อค่า N มีค่ามากขึ้น

ค่าถาม เป็นไปได้หรือไม่ที่จะวัดค่าฟังก์ชัน pdf ของกระบวนการที่ไม่มีความเป็นสเตชันนารี (ใช้ฟังก์ชัน pdf)

กลับไปที่ส่วนบนสุด

รูปที่ 12 ไฟล์ lab2_1.doc (ต่อ)

2.2.1 ฟังก์ชันคอร์รีเลชัน

2.2.2 ฟังก์ชันออร์โตคอร์รีเลชัน

2.2.3 ฟังก์ชันออร์โตคอร์รีเลชันกับฟังก์ชันความหนาแน่นสเปกตรัม (PSD)

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init2_2; ]
```

ขั้นแรก กำเนิดข้อมูลของคนที่มีอายุอยู่ระหว่าง 20 ถึง 50 ปี จำนวน 1000 คน โดยบันทึก อายุ ส่วนสูง และน้ำหนักของแต่ละคนไว้

```
[ a = person_data(1000); ]
```

จากข้อมูลที่เก็บไว้ในอาร์เรย์ a พล็อต scatter diagram ซึ่งแสดงค่าน้ำหนักเทียบกับส่วนสูง อายุเทียบกับน้ำหนัก และ อายุเทียบกับส่วนสูง

```
[ subplot(131), stat_plot(a, 'weight', 'height');
  subplot(132), stat_plot(a, 'age', 'height');
  subplot(133), stat_plot(a, 'age', 'weight'); ]
```

ค่าถาม ระหว่างค่า อายุ ส่วนสูง และน้ำหนัก ข้อมูลสองชนิดไหนที่มีความสัมพันธ์กันมากที่สุด

2.2.2 ฟังก์ชันออร์โตคอร์รีเลชัน

กลับไปที่ส่วนบนสุด

ฟังก์ชันออร์โตคอร์รีเลชันเป็นเครื่องมือที่ใช้ในการวัดความสัมพันธ์ของแต่ละค่าตัวอย่างในขบวนการสุ่มหนึ่งๆ ลองพิจารณากระบวนการสุ่ม T(t) ซึ่งแทนอุณหภูมิเฉลี่ยในแต่ละวันในประเทศแคนนาดาในช่วงวันที่ 1 มิถุนายน ถึงวันที่ 31 สิงหาคม

```
[ clf
  subplot(211), temperature ]
```

หลังจากนั้น กำเนิดกระบวนการสุ่มชนิดเกาส์เซียน B(t) ซึ่งเป็นบันทึกจำนวนของเด็กที่เกิดในเมืองโตรอนโตในระหว่างวันที่ 1 มิถุนายน ถึงวันที่ 31 สิงหาคม

```
[ subplot(212), new_born ]
```

ค่าถาม จากกระบวนการสุ่มทั้งสอง กระบวนการสุ่มไหนที่ค่าข้อมูลใกล้ๆ กันมีความสัมพันธ์กันมากกว่า

พล็อตกราฟของอุณหภูมิและจำนวนเด็กที่เกิด และฟังก์ชันออร์โตคอร์รีเลชันของ T(t) - E[T(t)] และ B(t) - E[B(t)]

```
[ clf, subplot(211), temperature(0) ]
[ figure(2), subplot(211), new_born(0) ]
```

ถ้าให้ R(3) แทนคอร์รีเลชันระหว่างค่าตัวอย่าง 2 ค่าของกระบวนการสุ่มที่อยู่ห่างกันเท่ากับ 3 หน่วยทางเวลา จากรูป 2 รูปเมื่อสักครู่ จงวัดค่า R(3) ของกระบวนการสุ่มที่แทนอุณหภูมิและจำนวนเด็กเกิด

$R_T(3)$ =

$R_B(3)$ =

ค่าถาม ถ้าต้องการทราบค่าคอร์รีเลชันของอุณหภูมิเฉลี่ยระหว่างวันที่ 10 มิถุนายนกับวันที่ 28 มิถุนายน จะต้องใช้ค่า R_T ค่าไหน และอธิบายว่า ทำไมฟังก์ชันออร์โตคอร์รีเลชัน จึงเป็นฟังก์ชันลดของค่า k เมื่อ k > 0

กลับไปที่ส่วนบนสุด

รูปที่ 13 ไฟล์ lab2_2.doc

2.2.3 ฟังก์ชันออร์โตคอร์รีเลชันและฟังก์ชันความหนาแน่นสเปตรัม

กลับไปส่วนบนสุด

กำนิดกระบวนการสุ่ม Z(t) ที่แต่ละค่าตัวอย่างมีคอรีเลชันกัน จำนวน 4096 ค่า โดยใช้ฟังก์ชัน corr_seq และพล็อตรูปที่ได้

```
[ close(2), clf
z = corr_seq(0.85, 4096, 3, 0);
waveplot(z) ]
```

คำนวณและพล็อตค่าฟังก์ชันออร์โตคอร์รีเลชัน $R_Z(\tau)$ และฟังก์ชัน PSD $S_Z(f)$ ของกระบวนการสุ่ม Z(t)

```
[ clf, subplot(211), acf(z)
  subplot(212), psd(z) ]
```

อภิปราย ใช้รูป $R_Z(\tau)$ที่ได้ จงหาค่า mean และ mean-square ของกระบวนการสุ่ม Z(t)

E[Z(t)] =

$E[Z^2(t)]$ =

การหาค่า mean-square ของกระบวนการสุ่ม Z(t) อีกสามารถหาได้อีกวิธีจากกราฟ $S_z(f)$ ค่า mean-square มีค่าเท่ากับพื้นที่ใต้กราฟของฟังก์ชัน PSD และเนื่องจากเกิดผลกระทบจากการสุ่มตัวอย่างใน MATLAB ค่า mean-square จึงมีค่าเท่ากับพื้นที่ใต้กราฟคูณกับค่าคงที่ 0.04096 (ขนาดของข้อมูล/ความถี่ในการสุ่มตัวอย่าง)

$E[Z^2(t)]$ = [? * 0.04096] =

กลับไปส่วนบนสุด

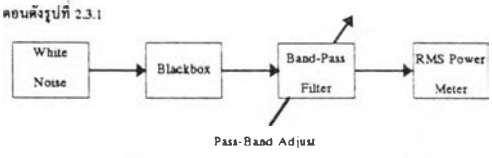
รูปที่ 14 ไฟล์ lab2_2.doc (ต่อ)

2.3 สัญญาณรบกวนแบบไวท์

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init2_3; ]
```

เนื่องจากฟังก์ชัน PSD ของกระบวนการสุ่มชนิดไวท์นอยส์ มีค่าเท่ากันตลอดช่วงแบนด์วิดท์ทั้งหมดของระบบที่พิจารณา ดังนั้น เราสามารถใช้กระบวนไวท์นอยส์เป็นเครื่องมือในการเอกเอกระบบ (system identification) โดยป้อนกระบวนการไวท์นอยส์เป็นอินพุทให้กับระบบที่ไม่รู้จัก ตัวอย่างเช่น ต้องการทราบผลตอบสนองของระบบๆหนึ่ง ก็จะมีขั้นตอนดังรูปที่ 2.3.1



รูปที่ 2.3.1 การวัดระบบโดยใช้กระบวนการสุ่มไวท์นอยส์

ก่อนอื่น กำนิดกระบวนการไวท์เกาส์เซียนนอยส์ที่มีค่าเฉลี่ยเท่ากับ 0 และค่าความแปรปรวนเท่ากับ 1 จำนวน 1024 ค่า ป้อนกระบวนการนี้เป็นอินพุทให้กับ blackbox ซึ่งแทนฟิลเตอร์ที่ไม่ทราบแบนด์วิดท์และออร์เดอร์

```
[ wn = gauss(0,1,1024);
  cn = blackbox(wn); ]
```

หลังจากนั้น ป้อนเอาท์พุทของ blackbox ให้กับฟิลเตอร์แบบ bass-pass ซึ่งเปลี่ยนค่า pass-band ไปได้เรื่อยๆ และทำการวัดกำลัง RMS ของสัญญาณที่เอาท์พุทของฟิลเตอร์ โดยขั้นตอนทั้งสอง คือ บล็อก Band-Pass Filter และ RMS Power Meter ถูกรวมไว้ในฟังก์ชัน psd_est

ในการทดลองนี้ จะใช้ช่วงความถี่ในช่วง [0, 5 kHz] ในการประมาณสเปกตรัม และให้ band-pass filter ปรับค่าแบนด์วิดธ์ไปทีละ 250 Hz. ดังนี้

```
[ clf, psd_est(cn,0,5000,250) ]
```

พล็อตฟังก์ชัน PSD ของเอาท์พุทของ blackbox เปรียบเทียบกับเอาท์พุทที่ได้จากฟังก์ชัน spect_est เพื่อดูความถูกต้องในการประมาณสเปกตรัมของระบบ

```
[ hold on, psd(cn) ]
```

อภิปราย หาค่า pass-band และออร์เดอร์ของฟิลเตอร์ (blackbox) ที่ไม่ทราบค่านี้

รูปที่ 15 ไฟล์ lab2_3.doc

การทดลองที่ 3 การวิเคราะห์สัญญาณและระบบเชิงเส้น

วัตถุประสงค์
หนังสืออ้างอิง
กลับไปสู่สารบัญการทดลอง

สารบัญการทดลองย่อย

การทดลองที่ 3.1 การวิเคราะห์สัญญาณในเชิงเวลาและเชิงความถี่
การทดลองที่ 3.2 การวิเคราะห์ระบบเชิงเส้นในเชิงเวลาและเชิงความถี่

วัตถุประสงค์

เพื่อศึกษารายละเอียดเกี่ยวกับ

- สเปกตรัมของสัญญาณรายคาบและอนุกรมฟูเรียร์ของสัญญาณ
- สเปกตรัมของสัญญาณไม่เป็นรายคาบและฟูเรียร์ทรานสฟอร์ม
- ความหมายและผลกระทบในเชิงเวลาและเชิงความถี่ของ pole-zero ในทรานสเฟอร์ฟังก์ชันของระบบ

หนังสืออ้างอิง

1. "หลักการระบบสื่อสาร", รศ.ดร. ประสิทธิ์ ประพิณมงคลการ บริษัท ซีเอ็ดยูเคชั่น จำกัด หน้า 16-18
2. "หลักการไฟฟ้าสื่อสาร", รศ.ดร.บัณฑิต โรจน์อารยานนท์ สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย หน้า 12-32
3. "Communication Systems," Third edition., A.B.Carlson, McGraw-Hill,1986. หน้า 152-180
4. "Digital and Analog Communication Systems," Third edition., L.W. Couch II., MacMillan,1990 หน้า 49-64, 78-93

รูปที่ 16 ไฟล์ lab3.doc

3.1 การวิเคราะห์สัญญาณ

ในการทดลองนี้ เราจะใช้ฟังก์ชัน pulsetrain.m ในการศึกษาสเปกตรัมของสัญญาณรายคาบ เมื่อเรียกฟังก์ชันนี้แล้ว จะปรากฏหน้าต่างแสดงสัญญาณพัลส์สี่เหลี่ยมรายคาบและสเปกตรัมที่เกิดจากการกระจายอนุกรมฟูเรียร์ของสัญญาณพัลส์นี้ ทดลองเปลี่ยนค่า คาบเวลาของสัญญาณ (T) และ ความกว้างของพัลส์ พร้อมทั้งสังเกตุการเปลี่ยนแปลงของสเปกตรัมของสัญญาณ และ ตอบคำถามต่อไปนี้

[pulsetrain;]

คำถาม ถ้าให้ความกว้างของพัลส์มีค่าเท่ากับคาบเวลาของสัญญาณ สเปกตรัมของสัญญาณจะมีลักษณะเป็นอย่างไร

คำถาม ถ้าให้ความกว้างของพัลส์มีค่าน้อยมากๆ เมื่อเทียบกับคาบเวลาของสัญญาณ สเปกตรัมของสัญญาณจะมีลักษณะเป็นอย่างไร

คำถาม คาบเวลาของสัญญาณพัลส์ มีผลต่อสเปกตรัมของสัญญาณอย่างไร

คำถาม สัญญาณพัลส์สี่เหลี่ยมรายคาบควรมีลักษณะอย่างไร จึงจะทำให้สเปกตรัมของสัญญาณมีความต่อเนื่องมากขึ้น

รูปที่ 17 ไฟล์ lab3_1.doc

3.2 การวิเคราะห์ระบบเชิงเส้น

ระบบเชิงเส้นเป็นระบบพื้นฐานที่พบในวงจรไฟฟ้าและระบบสื่อสารทั่วไป ซึ่งเราสามารถแทนวงจรเหล่านี้ได้ด้วยสมการดิฟเฟอเรนเชียลแบบไม่เป็นไอโมเจเนียส (nonhomogeneous differential equation) สมการประเภทนี้จะมีคุณสมบัติเป็นเชิงเส้น แต่เนื่องจากความยุ่งยากในการแก้ปัญหาสมการดิฟเฟอเรนเชียล จึงได้มีการนำ ลาปาส ทรานสฟอร์ม (Laplace Transform) มาใช้แก้ปัญหาแทน

ในการทดลองนี้ จะศึกษาคุณลักษณะต่างๆ ของทรานเฟอร์ฟังก์ชันของระบบ ศึกษาความสำคัญของ pole และ zero ของทรานสเฟอร์ฟังก์ชัน โดยเรียกฟังก์ชัน sfuction เมื่อเรียกฟังก์ชัน sfuction จะปรากฏหน้าต่าง 2 หน้าต่าง คือ หน้าต่าง s-plane แสดงถึงตำแหน่ง pole และ zero ของทรานสเฟอร์ฟังก์ชันของระบบ และหน้าต่างแสดงผลตอบสนองทางความถี่และเฟสของทรานสเฟอร์ฟังก์ชันนั้นๆ

ต่อไป เรียกฟังก์ชัน sfuction และลองเปลี่ยนตำแหน่ง pole และ zero พร้อมทั้งสังเกตการเปลี่ยนแปลง

[sfunction;]

รูปที่ 18 ไฟล์ lab3_2.doc

การทดลองที่ 4 การแปลงเป็นสัญญาณดิจิตอล

วัตถุประสงค์
หนังสืออ้างอิง
กลับไปยังสารบัญการทดลอง

สารบัญการทดลองย่อย

การทดลองที่ 4.1 ทฤษฎีการสุ่มตัวอย่าง
การทดลองที่ 4.2 ควอนไตเซชันแบบยูนิฟอร์ม
การทดลองที่ 4.3 การผิดเพี้ยนของสัญญาณที่เกิดจากควอนไตเซชัน
การทดลองที่ 4.4 ควอนไตเซชันแบบนอนยูนิฟอร์ม
การทดลองที่ 4.5 ค่ากำลังของสัญญาณกำลังควอนไตเซชัน

วัตถุประสงค์

ในการทดลองนี้ จะศึกษา

- ทฤษฎีการสุ่มตัวอย่างและปรากฏการณ์เอเลียสซิ่ง
- คุณลักษณะต่างๆ ของควอนไตเซชันทั้งแบบยูนิฟอร์มและนอนยูนิฟอร์ม
- สังเกตผลกระทบจากการใช้ควอนไตเซชันแบบต่างๆ
- วัดอัตราส่วนระหว่างกำลังของสัญญาณต่อควอนไตเซชันนอยส์ (Signal to Quantization error ratio) สำหรับควอนไตเซอร์แบบต่างๆ เป็นฟังก์ชันของกำลังสัญญาณอินพุท

หนังสืออ้างอิง

1. หนังสือ "หลักการไฟฟ้าสื่อสาร" ของ รศ.ดร.บัณฑิต โรจน์อารยานนท์ สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย หน้า 246-258
2. หนังสือ "Communication Systems," Third edition ของ A.B.Carlson, McGraw-Hill,1986 หน้า 430-439
3. หนังสือ "Digital and Analog Communication Systems," Third edition, L.W. Couch II, MacMillan,1990 หน้า 130-144
4. หนังสือ "An Introduction to Analog & Digital Communications," S.Haykin, Wiley & Sons, 1989 หน้า 187-197

รูปที่ 19 ไฟล์ lab4.doc

4.1 ทฤษฎีการสุ่มตัวอย่าง

การศึกษาทฤษฎีการสุ่มตัวอย่างโดยพิจารณาที่โดเมนความถี่

เนื่องจากการทดลองนี้เป็นการทดลองบนคอมพิวเตอร์ซึ่งเป็นไปไม่ได้ที่จะสร้างสัญญาณอนาล็อกที่แท้จริงขึ้นมา แต่อย่างไรก็ตาม เราสามารถสร้างสัญญาณที่เสมือนเป็นสัญญาณอนาล็อกได้โดยกำหนดอัตราส่วนระหว่างความถี่ในการจำลองสัญญาณอนาล็อกกับความถี่ในการศึกษาทฤษฎีการสุ่มตัวอย่างที่เหมาะสม (ความถี่ในการจำลองสัญญาณอนาล็อกนี้ค่าประมาณ 10 เท่าของความถี่ที่ใช้สุ่มตัวอย่าง)
ในการขั้นแรก เราจะศึกษาปัญหา aliasing ของสัญญาณ โดยพิจารณาสัญญาณไซน์ความถี่เดียวที่มีสมการดังนี้

$$x(t) = \sin(2\pi f_o t + \varphi)$$

สมมติเราสุ่มตัวอย่างสัญญาณ x(t) นี้ ที่อัตราความถี่ $f_s = 1/T_s$ ก็จะได้สัญญาณชนิดไม่ต่อเนื่องทางเวลา (discrete-time signal) ดังนี้

$$x[n] = x(t)|_{t=nT_s} = x(t)|_{t=\frac{n}{f_s}} = \sin\left(2\pi \frac{f}{f_s} n + \varphi\right)$$

เราจะใช้สมการข้างบนวิเคราะห์ปัญหา aliasing ของสัญญาณ
ก่อนอื่น กำหนดค่าตัวแปรที่ใช้ในการทดลอง

```
[ init4_1; ]
```

กำหนด fs = 8 kHz แล้วสุ่มตัวอย่างสัญญาณไซน์ที่มีความถี่ f = 300 Hz. และพล็อตสัญญาณที่ได้โดยใช้ฟังก์ชัน stem

```
[ fs = 8000; Ts = 1/fs; f = 300;
  n = 0:(fs/f);
  x = sin(2*pi*f/fs*n);
  stem(n*Ts,x) ]
```

คำถาม จากรูปที่พล็อตได้ เมื่อดูด้วยตา เราสามารถวาดสัญญาณไซน์คืนได้หรือไม่

ต่อไป จะเปลี่ยนความถี่ f ของสัญญาณไซน์จากความถี่ 100 ถึง 475 Hz. โดยเปลี่ยนทีละ 125 Hz. โดยยังคงใช้ความถี่ในการสุ่มตัวอย่าง fs = 8 kHz. อยู่เช่นเดิม พล็อตสัญญาณที่สุ่มได้ และสังเกตุดูการเปลี่ยนแปลงที่เกิดขึ้น

```
[ n = 0:(fs/100); ]
[ f = 100;
  x = sin(2*pi*f/fs*n);
  subplot(411), stem(n*Ts,x) ]
```

คำถาม จากกราฟที่พล็อตได้ สัญญาณไซน์ที่ถูกสุ่มนิความถี่ถูกต้องตามที่คาดไว้หรือไม่

คราวนี้ จะเปลี่ยนความถี่ f ของสัญญาณไซน์จากความถี่ 7525 ถึง 7900 Hz. บวก และเปลี่ยนความถี่ทีละ 125 Hz. พล็อตสัญญาณที่สุ่มได้ และสังเกตุดูการเปลี่ยนแปลงที่เกิดขึ้น

```
[ n = 0:(fs/7525)*100; ]
[ f = 7625;
  x = sin(2*pi*f/fs*n);
  subplot(411), stem(n*Ts,x) ]
```

<u>คำถาม</u> จากกราฟที่พล็อตได้ สัญญาณไซน์ที่ถูกสุ่มนิความถี่ถูกต้องตามที่คาดไว้หรือ ไม่ และอธิบายด้วยว่าทำไมจึงเป็นเช่นนี้

คำถาม ถ้าเปลี่ยนความถี่ของสัญญาณไซน์เป็นความถี่ 32,100 ถึง 32,457 โดยเปลี่ยนทีละ 125 Hz. ถามว่าเราสามารถทำนายได้หรือไม่ว่า สัญญาณไซน์ที่ถูกสุ่มด้วยความถี่ 8 kHz นี้ จะมีความถี่ที่ปรากฏเพิ่มขึ้นหรือลดลง

กลับไปที่ส่วนบนสุด

รูปที่ 20 ไฟล์ lab4_1.doc

การศึกษาทฤษฎีการสุ่มตัวอย่างโดยพิจารณาที่โดเมนความถี่

กลับไปที่ส่วนบนสุด

เมื่อสัญญาณต่อเนื่องทางเวลาถูกสุ่มตัวอย่าง สเปกตรัมของสัญญาณจะเป็นตัวบอกถึงผลกระทบของ aliasing เนื่องจากจอบเชคของสเปกตรัมของสัญญาณจะถูกเลื่อนออกไปเป็นจำนวนเท่าของความถี่ที่ใช้ในการสุ่มตัวอย่าง

ในการทดลองนี้ เราจะใช้ความถี่ที่ใช้ในการจำลองสัญญาณอนาล็อก fsim เท่ากับ 200 kHz และจำลองสัญญาณต่อเนื่องทางเวลาซึ่งคือสัญญาณโคไซน์ที่มีความถี่ f0 เท่ากับ 6 kHz.

```
[ fsim = 200e3;
  f0 = 6000;
  n = (0:1000)/fsim;
  x = cos(2*pi*f0*n); ]
```

ขั้นต่อไป สุ่มตัวอย่างสัญญาณโคไซน์นี้ด้วยความถี่ fs เท่ากับ 20 kHz. พร้อมพล็อตสัญญาณที่สุ่มมาได้

```
[ fs = 20e3;
  x_sampled = sampling(x,fs);
  stem(n,x_sampled) ]
```

จะเห็นได้ว่าสัญญาณโคไซน์ที่ถูกสุ่มตัวอย่างแล้วเป็นสัญญาณไม่ต่อเนื่องทางเวลา เราจะนำสัญญาณนี้ป้อนให้กับฟิลเตอร์ผ่านความถี่ต่ำเพื่อสร้างสัญญาณอนาล็อกคืนกลับมาโดยฟิลเตอร์นี้จะมี cut-off frequency (fcut) เท่ากับ fs/2 Hz.

```
[ xr = lpfilter(x_sampled,fs/2); ]
```

พล็อตกราฟของสัญญาณ xr เทียบกับสัญญาณอนาล็อกเดิม x และสังเกตดูความแตกต่าง

```
[ plot(n(1:300),x(1:300)), grid, hold on
  plot(n(1:300),xr(1:300),'w') ]
```

คำถาม ความถี่ของสัญญาณ xr ที่สร้างกลับคืนมา เปลี่ยนไปจากเดิมคือสัญญาณ x หรือไม่

ขั้นสุดท้าย พิจารณาสัญญาณที่โดเมนความถี่ โดยพล็อตสเปกตรัมของสัญญาณอนาล็อกเดิม x และสัญญาณอนาล็อกหลังจากที่สร้างคืนมา xr เทียบกันเพื่อสังเกตุการเปลี่ยนแปลง โดยใช้ฟังก์ชัน fplot

```
[ subplot(211), fplot(x, 1/fsim)
  subplot(212), fplot(xr, 1/fsim) ]
```

จากขั้นตอนข้างบนทั้งหมด ตั้งแต่ การจำลองสัญญาณอนาล็อก การสุ่มตัวอย่างสัญญาณ และ การสร้างสัญญาณอนาล็อกกลับคืนมา จะถูกรวบรวมไว้ในฟังก์ชัน testalias ทดลองเปลี่ยนค่าความถี่ของสัญญาณโคไซน์ f0 ∈ { 6 kHz. , 7 kHz. , 9 kHz. , 10 kHz. ,15 Hz } และพล็อตกราฟสเปกตรัมสัญญาณทั้งสองเปรียบเทียบกัน

```
[ testalias(15e3) ]
```

คำถาม ที่ความถี่ใด ที่สัญญาณเริ่มเกิดปรากฏการณ์เอเลียสซิ่ง ทำไมจึงเป็นเช่นนั้น

รูปที่ 21 ไฟล์ lab4_1.doc (ต่อ)

4.2 ควอนไตเซชันแบบยูนิฟอร์ม

ในการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล หลังจากที่สัญญาณถูกสุ่มตัวอย่าง ขั้นตอนต่อไป คือควอนไตซ์ค่าสัญญาณที่สุ่มได้และเนื่องจากสัญญาณอนาล็อกมีขนาดที่ต่อเนื่อง พิสัยของสัญญาณที่ถูกสุ่มตัวอย่างจะมีจำนวนระดับสัญญาณเป็น อนันต์ ดังนั้นเราจะไม่สามารถแทนค่าระดับสัญญาณนี้ได้ด้วยโค้ดเวิร์ดที่มีความยาวจำกัด ควอนไตเซชัน (quantization) จึงเป็นการประมาณขนาดสัญญาณที่ถูกสุ่มด้วยขนาดสัญญาณแบบไม่ต่อเนื่อง (decrete)

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init4_2; ]
```

เริ่มต้นการทดลองด้วยการพล็อตกราฟการแมป (mapping) สัญญาณอินพุทและเอาท์พุทของควอนไตเซอร์แบบยูนิฟอร์ม 2 บิต (2 bit uniform quantizer) โดยใช้ฟังก์ชัน quant_ch

```
[ quant_ch(2,'uniform') ]
```

กำหนดให้ x แทนขนาดของพัลส์แบบอนาล็อก และ x_q แทนสัญญาณเอาท์พุทของควอนไตเซอร์เมื่ออินพุทเป็น x สมมติให้ขนาดของพัลส์ x ถูกนอร์มอไลซ์ (normalize) ค่าเป็น $|x| \le 1$

คำถาม จากกราฟที่ได้ เติมค่าต่างๆ ลงในคอลัมน์ที่ 1 และ 2 ในตาราง 4.2.1

ตาราง 4.2.1

ช่วงของสัญญาณอินพุท	x_q	โค้ดไบนารี

คำถาม ในควอนไตเซชันแบบยูนิฟอร์ม ความกว้างของช่วงของสัญญาณอินพุทมีความสัมพันธ์กับจำนวนระดับสัญญาณเอาท์พุทของควอนไตเซอร์อย่างไร

ต่อไป กำหนดลำดับ x จำนวน 8 ค่า โดยแต่ละค่าจะแทนขนาดของพัลส์แบบอนาล็อกที่ sampling instant นั้น

```
[ x = [0.8, 0.6, 0.2, -0.4, 0.1, -0.9, -0.3, 0.7]; ]
```

คำถาม จากตาราง 6.1.1 ถ้าป้อนลำดับพัลส์ x เป็นอินพุทของควอนไตเซอร์ และ x_q เป็นสัญญาณเอาท์พุทของควอนไตเซอร์ จงเติมค่าในคอลัมน์ที่ 1 และ 2 ของตาราง 4.2.2

ตาราง 4.2.2

ลำดับที่	x	x_q	โค้ดไบนารี
1			
2			
3			
4			
5			
6			
7			
8			

ลองเปรียบเทียบผลที่ได้กับค่าที่คำนวณจากฟังก์ชัน quantize

```
[ xq = quantize(x,2) ]
```

รูปที่ 22 ไฟล์ lab4_2.doc

คำถาม จงออกแบบโค้ดไบนารีขนาด 2 บิตที่แทนระดับควอนไตเซชันแต่ละระดับในตาราง 4.2.1 โดยใช้ natural code (แทนค่าระดับควอนไตเซชันด้วย 00)

ขั้นตอนในคำถามข้างบนนี้ เรียกว่า source coding และใช้ขบวนการนี้กับลำดับ xq และเติมค่าลงในตาราง 4.2.2

เราสามารถเขียนคำสั่งอย่างโค้ดโดยใช้ฟังก์ชัน bin_enc (natural binary encoder)

```
[ xbin = bin_enc(xq,2) ]
```

คำถาม เป็นไปได้หรือไม่ที่จะเข้ารหัสลำดับ x และ xq ด้วยโค้ดขนาด 2 บิต โดยแต่ละค่ามีโค้ดที่ไม่ซ้ำกันเลย

ในขบวนการการแปลงสัญญาณแบบอนาล็อกเป็นสัญญาณแบบดิจิตอล มีวิธีๆ หนึ่งที่นิยมกัน เรียกว่า Pulse Code Modulation (PCM) ซึ่งขั้นตอนการทำงานดังนี้

- การสุ่มตัวอย่าง (sampling)
- ควอนไตเซชัน (quantization)
- การเข้ารหัส (source coding)
- แปลงข้อมูลเป็น serial data stream

ดังนั้น ถ้าลำดับ xbin ถูกแปลงเป็นข้อมูลแบบ serial เราก็จะได้ข้อมูล PCM ที่แทนลำดับ x ดังนี้

```
[ par2ser(xbin) ]
```

รูปที่ 23 ไฟล์ lab4_2.doc (ต่อ)

4.3 การผิดเพี้ยนของสัญญาณที่เกิดจากควอนไตเซชัน

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init4_3; ]
```

ควอนไตเซชันเป็นการประมาณค่าให้กับลำดับข้อมูลที่แท้จริง เพื่อที่จะลดผลของการควอนไตส์สัญญาณ เราจะกำเนิดสัญญาณไซน์ขนาด 1 หน่วย ดังนี้

```
[ x = sin(2*pi*20*[1:400]/SAMPLING_FREQ); ]
```

ค่าความถี่ในการสุ่มตัวอย่างที่ตั้งไว้คือ 8 kHz (SAMPLING_FREQ) ขั้นต่อไป ใช้ควอนไตเซอร์แบบยูนิฟอร์มขนาด 2 บิต และพล็อตกราฟเปรียบเทียบรูปคลื่น x ก่อนและหลังการควอนไตซ์

```
[ clf, waveplot(x)
  hold on,waveplot(quantize(x,5)) ]
```

รันคำสั่งข้างบนซ้ำ โดยใช้ควอนไตเซอร์ขนาด 3, 4 และ 5 บิต

เนื่องจากลำดับที่ถูกควอนไตส์เป็นลำดับที่ถูกประมาณมาจากลำดับข้อมูลที่แท้จริง ดังนั้นควอนไตเซชันจึงทำให้เกิดการผิดเพี้ยนของสัญญาณ (distortion) ซึ่งแบ่งออกเป็น 2 ประเภท คือ

1. overload distortion (clipping)
2. quantization noise

โดย overload distortion จะเกิดขึ้นเมื่อสัญญาณอินพุทมีขนาดเกินเรนจ์อินพุทของควอนไตเซอร์ (ในการทดลองนี้คือ (-1,1))

ลองเซ็ตค่าขนาดของสัญญาณไซน์ x เท่ากับ 0.9 และพล็อตรูปคลื่น x และรูปคลื่น x ที่ถูกควอนไตซ์ขนาด 3 บิต

```
[ a = 0.9;
  clf, waveplot(a*x)
  hold on, waveplot(quantize(a*x,3)) ]
```

สังเกตดูกราฟที่ได้ เนื่องจาก max (x) = 0.9 ซึ่งน้อยกว่า 1 จึงไม่เกิด overload distortion ต่อไป ลองรันคำสั่งข้างบนใหม่โดยเปลี่ยนค่าขนาดของสัญญาณ x เป็น a = 1.25, 2 และ 5

คำถาม จงหาว่าสัญญาณ x กี่เปอร์เซนต์ที่ถูกควอนไตเซอร์คลิปสัญญาณเมื่อเซ็ตค่าขนาดสัญญาณ x เป็น 0.9, 2 และ 5 และสำหรับควอนไตเซอร์แบบยูนิฟอร์มขนาด 3 บิต จะมีระดับควอนไตเซชันสูงสุดและต่ำสุดเท่ากับเท่าไหร่

ขั้นต่อไป สังเกตดูความผิดพลาดที่เกิดจากควอนไตเซชันของสัญญาณ x โดยที่ N = 2, 3, 4 และ 5 พร้อมทั้งบันทึกค่าความผิดพลาดลงในตาราง 4.3.1

```
[ xq = quantize(x,N); xe = x-xq;
  clf, waveplot(xe) ]
```

ตาราง 4.3.1

N	2	3	4	5
max (xe)				

คำถาม หาสูตรของค่าความผิดพลาดสัมบูรณ์ของควอนไตเซชันแบบยูนิฟอร์ม N บิต ในรูปฟังก์ชันของ N และในกรณีไหนที่ค่าความผิดพลาดนี้มีค่าสูงสุด

กำเนิดลำดับสัญญาณ y และพล็อตฟังก์ชัน PSD ของสัญญาณ

```
[ y = sin(2*pi*512*[1:2048]/SAMPLING_FREQ);
  psd(y) ]
```

ต่อไป เป็นการพล็อตกราฟฟังก์ชัน PSD ของรูปคลื่น xq และหาค่ากำลังของความผิดพลาดในการควอนไตซ์ (dB) และบันทึกค่าลงในตาราง 6.2.2

```
[ yq = quantize(y,2);
  psd(yq)
  sq2 = 10*log10(var(y-yq)) ]
```

รูปที่ 24 ไฟล์ lab4_3.doc

ตาราง 4.3.2

N	2	3	4	5	6
σ_q^2					

ภายใต้สมมติฐานที่ว่า ขนาดของค่าความผิดพลาดที่เกิดจากการควอนไตซ์มีการ กระจายแบบยูนิฟอร์มในช่วง [-q/2 , q/2] โดยที่ q คือขนาดของแต่ละช่วงในการควอนไตซ์ของควอนไตเซอร์แบบยูนิฟอร์มขนาด N บิต และกำลังของสัญญาณรบกวนเท่ากับ

$\sigma_q^2 = -(4.77 + 6.02N)$ dB

เปรียบเทียบค่าที่ได้ในตาราง 4.3.2 กับค่าที่คำนวณได้จากสูตรข้างบน

รูปที่ 25 ไฟล์ lab4_3.doc (ต่อ)

4.4 ควอนไตเซชันแบบนอนยูนิฟอร์ม

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init4_4; ]
```

กำเนิดลำดับสัญญาณ s แทนสัญญาณเสียงขนาด 100 ค่าด้วยฟังก์ชัน speech และพล็อตกราฟของลำดับนี้

```
[ s = speech(100);
  subplot(211), waveplot(s) ]
```

สังเกตดู จะพบว่ากราฟที่พล็อตมีขนาดกระจายอยู่ในช่วง [-1, 1] ซึ่งเป็นเรนจ์ของอินพุทของควอนไตเซอร์

ลำดับต่อไป ควอนไตซ์สัญญาณ s โดยใช้ควอนไตเซอร์แบบยูนิฟอร์มขนาด 6 บิต และพล็อตรูปคลื่นที่ถูกควอนไตซ์แล้ว

```
[ sq = quantize(s,6);
  subplot(212), waveplot(sq) ]
```

จะเห็นว่ารูปคลื่นที่ถูกควอนไตซ์ sq มีรูปร่างใกล้เคียงกับรูปคลื่นที่แท้จริง s มาก ต่อไป เราจะเปลี่ยนขนาดของสัญญาณ s ด้วยค่าคงที่ K โดยที่ max |Ks| $\leq$ 0.01

```
[ x = normalize(s, 0.01); ]
```

เราอาจสมมติได้ว่าสัญญาณ x คือสัญญาณ s ที่ถูกลดทอน(attenuate) ต่อไปพล็อตรูปคลื่นของสัญญาณ x และควอนไตซ์สัญญาณ x โดยใช้ควอนไตเซอร์แบบยูนิฟอร์มขนาด 6 บิตเหมือนในตอนแรก พร้อมทั้งพล็อตรูปคลื่นสัญญาณที่ถูกควอนไตซ์ xq

```
[ xq = quantize(x,6);
  subplot(211), waveplot(x)
  subplot(212), waveplot(xq) ]
```

คำถาม ทำไมรูปคลื่น xq และ sq จึงแตกต่างกันมากขนาดนี้

รูปที่ 26 lab4_4.doc

ควอนไตเซชันแบบนอนยูนิฟอร์ม

พล็อตกราฟแสดงคุณลักษณะต่างๆของควอนไตเซอร์ที่ใช้กฎ μ-law companding ขนาด 3 บิต

```
[ clf, quant_ch(3,'mu_law') ]
```

คำถาม สมมติให้สัญญาณที่สุ่มตัวอย่างมามีขนาดพ้อส์จำกัดอยู่ในช่วง [-0.25, 0.25] ถ้าสัญญาณนี้ถูกควอนไตซ์ด้วยควอนไตเซอร์แบบ μ-law companding ขนาด 3 บิต จะมีระดับควอนไตเซชันที่ถูกใช้จริงๆ กี่ค่า และหาในกรณีที่ใช้ควอนไตเซอร์แบบยูนิฟอร์มด้วย

ในการควอนไตซ์แบบนอนยูนิฟอร์ม เราสามารถทำได้โดยมีขั้นตอนดังนี้คือ

- companding
- uniform quantization

และเนื่องจาก สัญญาณที่ถูกควอนไตซ์ขั้นสุดท้ายต้องถูกแปลงค่าให้เป็นไปตามค่าที่แท้จริงก่อนถูกควอนไตซ์ ดังนั้นจึงต้องผ่านขั้นตอนสุดท้ายคือ

- expanding

ต่อไป ควอนไตซ์แบบนอนยูนิฟอร์มกับสัญญาณพ้อส์ที่สุ่มมา s และ x และแปลงค่าคืนจากสัญญาณที่ถูกควอนไตซ์

```
[ msq = mu_inv(quantize(mu_law(s),6));
  mxq = mu_inv(quantize(mu_law(x),6)); ]
```

พล็อตรูปคลื่น msq และ mxq และเปรียบเทียบค่ากับสัญญาณที่แท้จริง s และ x สำหรับแรกเปรียบสัญญาณที่มีขนาดใหญ่ก่อน คือสัญญาณ s

```
[ clf, subplot(311), waveplot(s)
  subplot(312), waveplot(sq)
  subplot(313), waveplot(msq) ]
```

ลำดับต่อไป ในกรณีสัญญาณที่มีขนาดเล็กหรือสัญญาณ x

```
[ figure(2), subplot(311), waveplot(x)
  subplot(312), waveplot(xq)
  subplot(313), waveplot(mxq) ]
```

ในกรณีที่ระดับควอนไตเซชันจำนวนเท่าๆ กัน ควอนไตเซอร์แบบยูนิฟอร์มจะเป็นวิธีที่เหมาะสมที่สุดถ้าขนาดของสัญญาณอินพุทมีการกระจายแบบยูนิฟอร์มในช่วง [-1, 1] แต่จะไม่เหมาะ ถ้าเป็นสัญญาณเสียงซึ่งขนาดของสัญญาณจะมีการกระจายแบบลาปาส (Laplace)

สุ่มลำดับที่มีการกระจายแบบลาปาสที่มีค่า variance เท่ากับ 0.01 จำนวน 2,000 ค่าโดยใช้ฟังก์ชัน laplace

```
[ close(2), figure(1), clf
  a = laplace(2000,0.01);
  pdf(a,30) ]
```

คอมแพนด์ลำดับ a โดย μ-law compander และพล็อตฟังก์ชัน pdf ของลำดับผลลัพธ์ที่ได้

```
[ b = mu_law(a);
  hold on, pdf(b,30) ]
```

คำถาม ฟังก์ชัน pdf ของลำดับ b ที่ถูกคอมแพนด์ มีลักษณะเป็นอย่างไร เหมาะสมที่จะถูกควอนไตส์แบบยูนิฟอร์มหรือไม่

รูปที่ 27 lab4_4.doc (ต่อ)

การทดลองที่ 5 รูปแบบการส่งสัญญาณไบนารี

วัตถุประสงค์
หนังสืออ้างอิง
กลับไปยังสารบัญการทดลอง

สารบัญการทดลองย่อย

การทดลองที่ 5.1 ไลน์โค้ดแบบต่างๆ ที่ใช้ในระบบสื่อสาร
การทดลองที่ 5.2 ความหนาแน่นสเปกตรัมกำลังของไลน์โค้ดแบบต่างๆ
การทดลองที่ 5.3 คุณลักษณะของช่องสัญญาณ
การทดลองที่ 5.4 แผนภูมิรูปตวงตา

วัตถุประสงค์

ศึกษาการส่งผ่านข่าวสารแบบไบนารีที่ความถี่เบสแบนด์ โดยเฉพาะ

- ไลน์โค้ดแบบต่างๆ ที่ใช้ในระบบสื่อสาร
- ฟังก์ชันความหนาแน่นสเปกตรัม (PSD) ของไลน์โค้ดแบบต่างๆ
- ความผิดเพี้ยนของสัญญาณที่เกิดจากช่องสัญญาณสื่อสาร
- ผลกระทบของการรบกวนระหว่างสัญญาณ (ISI) และสัญญาณรบกวนของช่องสัญญาณเมื่อพิจารณาจากแผนภูมิรูปตวงตา

หนังสืออ้างอิง

1. "หลักการไฟฟ้าสื่อสาร", รส.ดร บัณฑิต โรจน์อารยานนท์ สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย หน้า 273-288
2. "Communication Systems," Third edition, A.B.Carlson, McGraw-Hill,1986, หน้า 381-391.
3. "Digital and Analog Communication Systems," Third edition, L.W. Couch II, MacMillan,1990 หน้า 144-165
4. "An Introduction to Analog & Digital Communications,", S.Haykin, Wiley & Sons, 1989 หน้า 197-200 และ 227-230

รูปที่ 28 lab5.doc

5.1 ไลน์โค้ดแบบต่างๆที่ใช้ในระบบสื่อสาร

สัญญาณไบนารีในระบบสื่อสารต่างๆ เช่น ระบบ PCM รูปแบบในการส่งสัญญาณเป็นบิตอย่างต่อเนื่องอาจมีได้หลายรูปแบบซึ่งเราจะเรียกรูปแบบต่างๆ ว่า ไลน์โค้ด (line codes) ในการทดลองนี้เราจะศึกษาไลน์โค้ดชนิดต่างๆ และคุณสมบัติของมัน

ก่อนอื่นกำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init5_1; ]
```

เราจะใช้ฟังก์ชัน wave_gen กำเนิดรูปคลื่นสัญญาณของลำดับไบนารีต่างๆ

ขั้นแรก สร้างลำดับสัญญาณไบนารี b1

```
[ b1 = [ 1 0 1 0 1 1 ]; ]
```

กำเนิดรูปคลื่น x1 ของลำดับไบนารี b1 โดยใช้ไลน์โค้ดแบบ Unipolar NRZ โดยมี R_b = 1 kbps และพล็อตรูปคลื่น x1

```
[ x1 = wave_gen( b1, 'unipolar_nrz', 1000);
  waveplot(x1) ]
```

เราสามารถสร้างไลน์โค้ดแบบอื่นๆ โดยเปลี่ยนค่า lin_code_name ในคำสั่ง wave_gen เป็นค่าอื่น ดังต่อไปนี้

- unipolar RZ ('unipolar_rz')
- bipolar RZ ('bipolar_rz')*
- polar NRZ ('polar_nrz')
- bipolar NRZ ('bipolar_nrz')*
- polar RZ ('polar_rz')
- manchester ('manchester')

* ไลน์โค้ด bipolar มีชื่อเรียกอีกชื่อหนึ่งคือ Alternate Mark Inversion (AMI)

คำถาม ไลน์โค้ดแบบใดที่ไม่มีส่วนประกอบกระแสตรง เมื่อสมมติให้ความน่าจะเป็นที่เกิดบิต 0 และ 1 ในลำดับไบนารีมีค่าเท่ากัน และไลน์โค้ดแบบใดที่ไม่มีส่วนประกอบกระแสตรงเสมอ

คำถาม ในขณะส่งผ่านข้อมูล ถ้ามีการส่งรูปคลื่นกลับเครื่องหมายจากเดิม ต้องใช้ไลน์โค้ดแบบใด จึงจะสามารถถอดลำดับไบนารีกลับคืนได้อย่างถูกต้อง

รูปที่ 29 lab5_1.doc

5.2 ความหนาแน่นสเปกตรัมกำลังของไลน์โค้ดแบบต่างๆ

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init5_2; ]
```

สุ่มสัญญาณไบนารี b2 มาจำนวน 1,000 ค่า

```
[ b2 = binary(1000); ]
```

พล็อตค่า PSD ของไลน์โค้ดแบบ unipolar NRZ ของลำดับไบนารี b2 โดยใช้ฟังก์ชัน psd

```
[ psd(wave_gen(b2,'unipolar_nrz')) ]
```

กำหนดให้

f_{p1} = first spectral peak f_{n1} = first spectral null

f_{p2} = second spectral peak f_{n2} = second spectral null

จากกราฟที่ได้ บันทึกค่าความถี่เหล่านี้ลงในตารางที่ 5.2.1

ตารางที่ 5.2.1

R_b =	f_{p1}	f_{n1}	f_{p2}	f_{n2}	B_T
unipolar NRZ					
polar NRZ					
bipolar RZ					
manchester					

** ค่าหนึ่งของ first spectral null จะเป็นตัวกำหนดแบนด์วิดท์ที่ใช้ในการส่ง B_T

สังเกตดูค่า PSD ของไลน์โค้ดแบบ manchester ในกรณีที่มีค่าอัตราข้อมูล $R_b \in$ { 5 kbps,10 kbps, 20 kbps }

```
[ psd(wave_gen(b2, 'manchester', 5000)) ]
```

จากคำสั่งข้างบน ทดลองกับเปลี่ยนไลน์โค้ดเป็นแบบอื่นๆ และสังเกตดูความแตกต่าง

คำถาม ค่าหนึ่งของค่าสูงสุดของสเปกตรัม (spectral peaks) และค่าศูนย์ของสเปกตรัม (nulls) มีความสัมพันธ์อย่างไรกับอัตราข้อมูล R_b

รูปที่ 30 ไฟล์ lab5_2.doc

5.3 คุณลักษณะของช่องสัญญาณ

ผลกระทบของสัญญาณรบกวนในช่องสัญญาณที่มีต่อรูปคลื่นสัญญาณ
ผลกระทบของแบนด์วิดท์ของช่องสัญญาณที่มีต่อรูปคลื่นสัญญาณ

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init5_3; ]
```

ในที่นี้ เราจะใช้ฟังก์ชัน channel จำลองลอดอบสนองของช่องสัญญาณ

สุ่มลำดับไบนารี b3 มา 10 ค่า และกำเนิดรูปคลื่นของ b3 เป็นไลน์โค้ดแบบ polar NRZ โดยมีอัตราข้อมูล (R_b) เท่ากับ 1 kbps

```
[ b3 = binary(10);
  x3 = wave_gen( b3, 'polar_nrz', 1000); ]
```

คำถาม แบนด์วิดท์ที่ใช้ในการส่ง สัญญาณ x3 ควรมีค่าเป็นเท่าไร

พิจารณาช่องสัญญาณแบบเบสแบนด์ที่มีอัตราขยาย (gain) เท่ากับ 1 และมีสัญญาณรบกวนแบบเกาส์เซียนโดยมีค่ากำลังสัญญาณรบกวน (noise_power) เท่ากับ 10^{-3} W และแบนด์วิดท์ของช่องสัญญาณ (bandwidth) มีค่าเท่ากับ 4.9 kHz.

ส่งรูปคลื่นสัญญาณ x3 ผ่านช่องสัญญาณนี้ และพล็อตรูปคลื่นอินพุทและเอาท์พุทของช่องสัญญาณ

```
[ y3 = channel( x3, 1, 0.01, 4900 );
  subplot(211), waveplot(x3);
  subplot(212), waveplot(y3) ]
```

คำถาม จากรูปคลื่นเอาท์พุทที่ได้ เราสามารถคาดคะเนได้หรือไม่ว่าลำดับสัญญาณไบนารีที่ส่งมามีค่าเป็นอย่างไร

ผลกระทบของสัญญาณรบกวนในช่องสัญญาณที่มีต่อรูปคลื่น

กลับไปที่ส่วนบนสุด

เพิ่มค่ากำลังสัญญาณรบกวนของช่องสัญญาณ โดยกำหนดค่าแบนด์วิดท์ของช่องสัญญาณคงที่เท่ากับ 4.9 kHz และสังเกตุดูการเปลี่ยนแปลงของสัญญาณเอาท์พุทของช่องสัญญาณโดยใช้คำสั่งต่อไปนี้

```
[ subplot(212),waveplot(channel(x3,1,0.1,4900)) ]
```

ลองเปลี่ยนค่ากำลังสัญญาณรบกวน noise_power ∈ (0.1, 0.5, 1, 2, 5) และสังเกตดูความเปลี่ยนแปลง

คำถาม ที่ระดับกำลังสัญญาณรบกวนขนาดไหนที่เราไม่สามารถแยกแยะรูปคลื่นเอาท์พุทของช่องสัญญาณออกจากสัญญาณรบกวนได้

ต่อไป มาดูผลกระทบที่เกิดจากการเพิ่มกำลังสัญญาณรบกวนของช่องสัญญาณ โดยสังเกตุจาก PSD ของรูปคลื่นที่ด้านเอาท์พุทของช่องสัญญาณ

```
[ b4 = binary(1000); ]
[ x4 = wave_gen(b4, 'polar_nrz', 1000); ]
[ subplot(121), psd(x4), a = axis;
  subplot(122), psd(channel(x4, 1, 0.01, 4900))
  axis(a),hold on ]
```

ถ้าเพิ่มค่ากำลังสัญญาณรบกวนของช่องสัญญาณจาก 10^{-3} W เป็น 1 W และ 5 W ตามลำดับ กราฟ PSD ที่ได้ จะมีลักษณะเปลี่ยนแปลงไปอย่างไร จงพล็อตกราฟโดยใช้คำสั่งต่อไปนี้

```
[ psd(channel(x4, 1, noise_power, 4900)) ]
```

รูปที่ 31 ไฟล์ lab5_3.doc

ผลกระทบของแบนด์วิดท์ของช่องสัญญาณที่มีต่อรูปคลื่น
กลับไปที่ส่วนบนสุด

ความผิดเพี้ยนของรูปคลื่นที่เกิดขึ้น อาจเกิดจากช่องสัญญาณที่ส่งผ่านสัญญาณคู่นั้นเป็นช่องสัญญาณที่มีแบนด์วิดท์จำกัด ในส่วนนี้เราจะมาศึกความผิดเพี้ยนของรูปคลื่นในลักษณะนี้ พิจารณาเฉพาะแบนด์วิดท์ของช่องสัญญาณเท่านั้น โดยตั้งค่าทำลังสัญญาณรบกวนให้มีค่าเป็นศูนย์

```
[ b5 = binary(10); ]
[ x5 = wave_gen(b5, 'polar_nrz', 1000); ]
[ subplot(211), waveplot(x5)
  subplot(212), waveplot(channel(x5, 1, 0, 4900)) ]
```

สังเกตุผลกระทบของแบนด์วิดท์ที่ต่อรูปคลื่นที่จากช่องสัญญาณโดยเปลี่ยนค่าแบนด์วิดท์(bw) ∈ { 3000, 2000, 1000, 500}ในคำสั่งข้างล่างนี้

```
[ subplot(212), waveplot( channel(x5, 1, 0, bw) ) ]
```

สังเกตุดูเวลาหน่วง (delay) ของรูปคลื่นเอาท์พุทที่เกิดจากคุณลักษณะการฟิลเตอร์ของช่องสัญญาณว่ามีความสัมพันธ์อย่างไรกับแบนด์วิดท์ของช่องสัญญาณ

รูปที่ 32 ไฟล์ lab5_3.doc (ต่อ)

5.4 แพทเทินรูปดวงตา

ความหมายของส่วนต่าง ๆ ที่ปรากฏในแพทเทินรูปดวงตา

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init5_4; ]
```

ขั้นตอนต่อไปนี้ เป็นการสร้างแพทเทินรูปดวงตาจากลำดับไบนารี b6

```
[ b6 = [ 1 0 0 1 0 1 1 0 ]; ]
[ x6 = wave_gen( b6, 'polar_nrz', 1000); ]
[ subplot(211), waveplot(x6)
  subplot(212), eye_diag(x6) ]
```

ต่อไปสร้างแพทเทินรูปดวงตาจากสัญญาณที่ผิดเพี้ยนไปเนื่องจากแบนด์วิดท์ของช่องสัญญาณที่จำกัด แต่ปราศจากสัญญาณรบกวน

```
[ y6 = channel( x6, 1, 0, 4000); ]
[ subplot(211), waveplot(y6)
  subplot(212), eye_diag(y6) ]
```

ทดลองกำหนดแพทเทินรูปดวงตาจากรูปคลื่นแบบ polar NRZ ที่ด้านเอาท์พุทของช่องสัญญาณโดยใช้ค่าทำลังสัญญาณรบกวน (noise) และแบนด์วิดท์ของช่องสัญญาณ (bw) ตามตาราง 5.4.1 พร้อมทั้งบันทึกค่า t* A และ B ของแต่ละกรณีด้วย

```
[ b7 = binary(100); ]
[ x7 = wave_gen( b7, 'polar_nrz', 1000); ]
[ clf
  eye_diag( channel(x7, 1, s2, bw)) ]
```

รูปที่ 33 ไฟล์ lab5_4.doc

ตารางที่ 5.4.1

Polar NRZ Line Code				
s2	bw	t*	A	B
0.01	3000			
	2000			
	1000			
0.02	4000			
0.08				
0.10				

เปลี่ยนไอน์ไค้ดเป็นแบบ manchester และบันทึกค่าต่างๆ ลงในตารางที่ 5.4.2

ตารางที่ 5.4.2

Manchester Line Code				
s2	bw	t*	A	B
0.01	3000			
	2000			
	1000			
0.02	4000			
0.08				
0.10				

คำถาม จากชุดคำสั่งข้างบน ถ้าให้ s2 = 0.001 และ bw = 1000 ทดลองสร้างแพทเทินรูปดวงตาจากไอน์ไค้ดแบบ polar NRZ และ manchester แพทเทินรูปดวงตาจากไอน์ไค้ดแบบไหนที่คุณช่วยรูปดวงตามากกว่า ทำไมจึงเป็นเช่นนั้น อธิบายความแตกต่างในด้านคุณสมบัติของไอน์ไค้ดทั้งสอง

คำถาม จากชุดคำสั่งข้างบน ลองเปลี่ยนไอน์ไค้ดเป็นแบบ polar RZ, unipolar RZ และ unipolar NRZ แล้วสังเกตดูว่าไอน์ไค้ดเหล่านี้ทำให้แพทเทินรูปดวงมีรูปร่างและความสมบูรณ์อย่างไร

รูปที่ 34 ไฟล์ lab5_4.doc (ต่อ)

การทดลองที่ 6 การตรวจจับสัญญาณ (ดีเทคชัน)

วัตถุประสงค์
หนังสืออ้างอิง
กลับไปยังสารบัญการทดลอง

สารบัญการทดลองย่อย

การทดลองที่ 6.1 คุณลักษณะต่าง ๆ ของแมทช์ฟิลเตอร์
การทดลองที่ 6.2 การตรวจจับสัญญาณ
การทดลองที่ 6.3 เครื่องรับสัญญาณแมทช์ฟิลเตอร์
การทดลองที่ 6.4 เครื่องรับสัญญาณวงจรผ่านความถี่ต่ำ
การทดลองที่ 6.5 การประยุกต์ใช้งานโครงสร้างแมทช์ฟิลเตอร์

วัตถุประสงค์

ในการทดลองนี้ จะศึกษากระบวนการตรวจจับสัญญาณ (detection) โดยศึกษาส่วนประกอบต่างๆ ของเครื่องรับสัญญาณ และกระบวนการถอดรหัส (decoding process) ดังนี้

- ศึกษาคุณลักษณะของแมทช์ฟิลเตอร์
- ศึกษาสมรรถนะของเครื่องรับสัญญาณที่ใช้วงจรฟิลเตอร์แบบต่างๆ โดยวัดจากความน่าจะเป็นการผิดพลาดบิต (probability of bit error)
- การใช้แพทเทิร์นรูปดวงตา (eye pattern) เป็นเครื่องมือในการตั้งค่าพารามิเตอร์ต่างๆ ในกระบวนการตรวจจับสัญญาณ

หนังสืออ้างอิง

1. "หลักการไฟฟ้าสื่อสาร", รศ.ดร บัณฑิต ไรจน์อารยานนท์ สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย หน้า 293-304
2. "Communication Systems," Third edition, A.B.Carlson, McGraw-Hill,1986. หน้า 391-400
3. "Digital and Analog Communication Systems," Third edition, L.W. Couch II, MacMillan,1990 หน้า 497-504, 521-532
4. "An Introduction to Analog & Digital Communications.", S.Haykin, Wiley & Sons,1989. หน้า 197-200 และ 539-573

รูปที่ 35 ไฟล์ lab6.doc

6.1 คุณลักษณะต่างๆ ของแมทช์ฟิลเตอร์

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init6_1; ]
```

กำเนิดพัลส์สี่เหลี่ยมขนาดเท่ากับ 1 หน่วย และมีความกว้างเท่ากับ 1 มิลลิวินาที

```
[ r = wave_gen(1,'polar_nrz',1000); ]
```

พล็อตรูปคลื่นของสัญญาณ r และผลตอบสนองอิมพัลส์ของแมทช์ฟิลเตอร์ที่อ้างอิงกับสัญญาณ r

```
[ subplot(311), waveplot(r)
  subplot(312), match('polar_nrz') ]
```

ขั้นต่อไป ป้อนสัญญาณ r ให้กับแมทช์ฟิลเตอร์ พร้อมทั้งสังเกตุเอาท์พุทของแมทช์ฟิลเตอร์

```
[ rm = match('polar_nrz', r);
  subplot(313), waveplot(rm) ]
```

คำถาม ที่เวลาไหน ที่เอาท์พุทของแมทช์ฟิลเตอร์มีค่าสูงที่สุด และที่เวลานั้นมีความสัมพันธ์กับรูปคลื่น r อย่างไร

จากคำสั่งข้างบนดั้งเดิม จงทำการโดดเปลี่ยนรหัสของพัลส์เป็นพัลส์สามเหลี่ยม('triangle') ที่มีขนาด 1 หน่วย และมีความกว้างพัลส์เท่ากับ 10 มิลลิวินาที และคาดคะเนคุณผลตอบสนองอิมพัลส์ของแมทช์ฟิลเตอร์และเอาท์พุทของแมทช์ฟิลเตอร์จะมีลักษณะเป็นอย่างไร

คำถาม ถ้าพัลส์สามเหลี่ยมเปลี่ยนขนาดความกว้างเป็น 1 มิลลิวินาที ที่เวลาไหนที่เอาท์พุทของแมทช์ฟิลเตอร์มีค่าสูงสุด

จากนั้น จงเปลี่ยนรหัสของพัลส์เป็นพัลส์แบบ manchester ขนาด 1 หน่วย และมีความกว้างพัลส์เท่ากับ 10 มิลลิวินาที คูปัน และทำนายพฤติกรรมผลตอบสนองอิมพัลส์ของแมทช์ฟิลเตอร์และเอาท์พุทของแมทช์ฟิลเตอร์จะมีลักษณะเป็นอย่างไร

จากที่ทำการทดลองมาทั้งหมด เป็นเฉพาะในกรณีของพัลส์เพียงลูกเดียว การทดลองต่อไปนี้ จะเป็นกรณีสัญญาณที่ถูกส่งมาเป็นลำดับพัลส์ปัน

กำเนิดรูปคลื่นรหัส polar NRZ ที่แทนลำดับไบนารี [1 0 0 1 0] โดยมีอัตราข้อมูล (R_b) เท่ากับ 1 kbps และมีขนาดของพัลส์เท่ากับ 1 หน่วย

```
[ x5 = wave_gen([1 0 0 1 0],'polar_nrz',1000);
  clf, subplot(211), waveplot(x5) ]
```

ป้อนสัญญาณ x5 ให้กับแมทช์ฟิลเตอร์ และพล็อตรูปคลื่นเอาท์พุทของแมทช์ฟิลเตอร์

```
[ subplot(212), waveplot(match('polar_nrz',x5)) ]
```

คำถาม ถ้าอินพุทของแมทช์ฟิลเตอร์เปลี่ยนเป็นรูปคลื่นรหัส unipolar NRZ ที่มีแทน ลำดับไบนารี [1 0 0 1 0] จงสเก็ตรูปคลื่นเอาท์พุทของแมทช์ฟิลเตอร์

รูปที่ 36 ไฟล์ lab6_1.doc

6.2 การตรวจจับสัญญาณ (Signal Detection)

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init6_2; ]
```

สุ่มลำดับไบนารีจำนวน 10 ค่า พร้อมทั้งรูปคลื่นของลำดับไบนารีนี้ โดยกำหนดเป็นรูปคลื่น polar NRZ

```
[ b10 = binary(10);
  x10 = wave_gen(b10,'polar_nrz',1000);
  clf, subplot(211), waveplot(x10) ]
```

ป้อนสัญญาณ x10 ผ่านช่องสัญญาณที่มีแบนด์วิดท์ 4.9 kHz และมีสัญญาณรบกวนแบบ AWGN โดยมีกำลังสัญญาณรบกวนเท่ากับ 2 วัตต์ พล็อตรูปคลื่นเอาท์พุทของช่องสัญญาณ (y10)

```
[ y10 = channel(x10,1,2,4900);
  subplot(212), waveplot(y10) ]
```

คำถาม จงถอดรหัสลำดับไบนารีจากรูปคลื่น y10

ลำดับต่อไป ป้อนสัญญาณ y10 ให้กับแมทช์ฟิลเตอร์ และพล็อตรูปคลื่นเอาท์พุทของแมทช์ฟิลเตอร์ (z10)

```
[ z10 = match('polar_nrz',y10);
  subplot(212), waveplot(z10) ]
```

ถ้ากำหนดให้ T_b คือ คาบเวลาของลำดับไบนารี และทำการสุ่มสัญญาณเอาท์พุทของแมทช์ฟิลเตอร์ที่เวลา kT_b โดยที่ k = 1,2,...,10 ค่าที่ได้จะนำมาตัดสินใจว่าเป็น 0 หรือ 1 จากกฎต่อไปนี้

$$\hat{b}_k = \begin{cases} 0 & if \quad z10(kT_b) \le 0 \\ 1 & if \quad z10(kT_b) > 0 \end{cases}$$

โดยที่ $\hat{b}_k$ คือค่าที่ประมาณได้ของข้อมูลตัวที่ k ของลำดับไบนารี b10

ลองใช้กฎการตัดสินใจข้างบนนี้กับสัญญาณเอาท์พุทของแมทช์ฟิลเตอร์ (z10)

$\hat{b10}$ =

เปรียบเทียบค่าที่ประมาณได้กับลำดับไบนารีที่ต้นทาง (b10)

คำถาม จงวิจารณ์ความยากง่ายในการถอดรหัสลำดับไบนารีที่ถูกส่งมา ทั้งถอดรหัสโดยตรงจากสัญญาณเอาท์พุทของช่องสัญญาณ (y10) หรือถอดรหัสจากสัญญาณเอาท์พุทของแมทช์ฟิลเตอร์ (z10) และถ้าค่า sampling instants มีค่าต่างจากที่ระบุไว้ข้างบน ความน่าจะเป็นของการถอดรหัสผิดพลาดจะมีค่ามากขึ้น ทำไมจึงเป็นเช่นนั้น

6.3 เครื่องรับสัญญาณแมทช์ฟิลเตอร์

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init6_3; ]
```

สุ่มลำดับไบนารี b จำนวน 2,000 ค่า และกำหนดเป็นรูปคลื่นชนิด polar NRZ

```
[ b = binary(2000);
  x = wave_gen(b,'polar_nrz'); ]
```

ส่งสัญญาณ x ผ่านช่องสัญญาณที่มีแบนด์วิดท์ 4.9 kHz และมีสัญญาณรบกวน AWGN โดยมีกำลังสัญญาณ σ^2_n เท่ากับ 0.5 วัตต์ ให้ y เป็นสัญญาณเอาท์พุทของช่องสัญญาณ

```
[ y = channel(x,1,0.5,4900); ]
```

หลังจากนั้น ป้อนสัญญาณ x ให้กับแมทช์ฟิลเตอร์ และพล็อตแพทเทิร์นรูปดวงตาของสัญญาณเอาท์พุทของแมทช์ฟิลเตอร์ z

```
[ z = match('polar_nrz',y); ]
[ clf, eye_diag(z); ]
```

ในการถอดรหัสสัญญาณที่ถูกส่งมา จะต้องทราบค่า sampling instant, threshold value และ binary data period ตัวอย่างเช่น เมื่อกำหนดให้ binary data period เท่ากับ T_b และ sampling instant เท่ากับ t_1 ดังนั้นเครื่องถอดรหัสจะสุ่มตัวอย่างสัญญาณที่เวลา t_1, $t_1 + T_b$, $t_1 + 2T_b$, ...

คำถาม จากแพทเทิร์นรูปดวงตาที่ได้ จงหาค่า sampling instants และ threshold value ที่เหมาะสมในการถอดรหัสสัญญาณที่ถูกส่งมา

v_th = โวลต์

sampling instant = วินาที

นำค่า v_th และ sampling instant ที่ได้มาใช้กับเครื่องถอดรหัส (ฟังก์ชัน detect) ซึ่งจะทำงานอยู่ที่ด้านเอาท์พุทของแมทช์ฟิลเตอร์ บันทึกค่าความน่าจะเป็นการผิดพลาดของข้อมูล P_e (อ้างอิงถึงอัตราการผิดพลาดของข้อมูล (BER)) ในตารางที่ 6.3.1

```
[ detect(z,v_th,sampling_instant,b); ]
```

ตารางที่ 6.3.1

σ^2_n (W)	P_e - empirical	P_e - theoretical
0.5		
1.0		
1.5		
2.0		

**** จับคำสั่ง MATLAB ข้างบนข้างใหม่ในกรณี σ^2_n มีค่าเท่ากับ 1, 1.5 และ2 โดยไม่ต้องพล็อตแพทเทิร์นรูปดวงตาและจากการทดลองที่ 4 เราพบว่า ค่า sampling instant และ threshold ที่เหมาะสม ไม่ขึ้นกับกำลังสัญญาณรบกวนในช่องสัญญาณ ดังนั้นเราสามารถนำค่า sampling instant และ threshold value ที่หาได้ในครั้งแรกข้างบน มาใช้ในกรณีที่ σ^2_n ของช่องสัญญาณมีค่าต่างๆ กันได้*

ถ้ามีการใช้ค่า sampling instant ค่าอื่นที่ไม่เหมาะสม ค่า BER ที่ได้จะมีค่าเพิ่มขึ้น เราสามารถสังเกตุได้จากการถอดรหัสสัญญาณ(ฟังก์ชัน detect)โดยใช้ค่า sampbeg_instant เท่ากับ 0.9 และ 0.5 เท่าของค่าที่เหมาะสมที่หาได้ตอนแรกในคำถามข้างบน

คำถาม จงหาค่าความน่าจะเป็นการผิดพลาดของข้อมูล (probability of bit error) ทางทฤษฎีในทุกกรณีของตารางที่ 6.3.1 โดยมีฟังก์ชัน PSD ของกระบวนการไวท์นอยส์เป็นดังนี้

$$S_n(f) = \frac{N_o}{2} = \frac{\sigma^2_n}{2 \times system\_bandwidth}$$

โดย system_bandwidth ในการทดลองนี้เท่ากับ 5 kHz.

รูปที่ 37 ไฟล์ lab6_2.doc

รูปที่ 38 ไฟล์ lab6_3.doc

6.4 เครื่องรับสัญญาณวงจรผ่านความถี่ต่ำ

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

[init6_4;]

จั้นแรก ป้อนสัญญาณพัลส์สี่เหลี่ยม 1 ลูกให้กับฟิลเตอร์แบบ first-order RC ที่มีแบนด์วิดท์เท่ากับ 1 kHz และพล็อตสัญญาณเอาท์พุทของฟิลเตอร์ และวัดจนาดสูงสุดของสัญญาณ (A_c)

[r = wave_gen(1,'unipolar_nrz');
r_lpf = rc(1000,r);
subplot(211), waveplot(r)
subplot(212), waveplot(r_lpf);]

A_c = โวลต์

กำเนิดสัญญาณรบกวนไวท์นอยส์ที่มีค่า mean เท่ากับ 0 และมีกำลังเท่ากับ 0.5 วัตต์ จำนวน 2,000 ค่า ป้อนสัญญาณรบกวนนี้ให้กับฟิลเตอร์ RC และบันทึกค่า RMS ของกำลังของสัญญาณที่เอาท์พุทของฟิลเตอร์

[n = gauss(0,0.5,2000);
meansq(rc(1000,n))]

σ_n^2 = วัตต์

คำถาม จากค่า A_c และ σ_n^2 ที่ได้ หาค่า A_c/σ_n และถ้าป้อนสัญญาณ y จากการทดลองที่ 6.3 ให้กับเครื่องรับสัญญาณฟิลเตอร์ RC ของการทดลองนี้ จงหาค่า BER.

กำเนิดสัญญาณ y จากการทดลองที่ 5.4 อีกครั้งหนึ่ง แล้วป้อนสัญญาณ y ให้กับฟิลเตอร์ RC และพล็อตแพทเทิร์นรูปตางตาของสัญญาณเอาท์พุท z_lpf

[y = channel(x,1,0.5,4900);
z_lpf = rc(1000,y);
clf, eye_diag(z_lpf);]

จากแพทเทิร์นรูปตางตาที่ได้ หาค่า sampling instant และ threshold value ที่เหมาะสม และถอดรหัสลำดับไบนารีจากรูปคลื่น z_lpf

[detect(z_lpf,v_th,sampling_instant,b);]

เปรียบเทียบค่า BER ที่ได้กับค่า BER ใน การทดลองที่ 6.3

รันคำสั่ง detect ข้างบนใหม่ โดยใช้ค่ากำลังสัญญาณรบกวนเป็น 1, 1.5 และ 2 วัตต์ และบันทึกค่าลงในตารางที่ 6.4.1

ตารางที่ 6.4.1

σ_n^2 [W]	P_e	
	BW = 1.0 kHz.	BW = 0.5 kHz.
0.5		
1.0		
1.5		
2.0		

ลองป้อนสัญญาณ y ให้กับฟิลเตอร์ RC อีกครั้งหนึ่ง โดยแบนด์วิดท์ของฟิลเตอร์เปลี่ยนไปเป็น 500 Hz. และบันทึกค่า BER ที่ได้ลงในตารางที่ 6.4.1

คำถาม จงอธิบายว่า ทำไมค่า BER ที่ได้จากฟิลเตอร์ผ่านความถี่ต่ำที่มีแบนด์วิดท์ 500 Hz. ถึงมีค่าน้อยกว่าค่า BER ที่ได้จากฟิลเตอร์ที่มีแบนด์วิดท์เท่ากับ 1 kHz. และถ้าฟิลเตอร์ผ่านความถี่ต่ำมีแบนด์วิดท์เท่ากับ 100 Hz ค่า BER จะมีค่าลดลงอีกหรือไม่

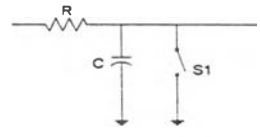
รูปที่ 39 ไฟล์ lab6_4.doc

6.5 การประยุกต์ใช้งานโครงสร้างแมทช์ฟิลเตอร์

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

[init6_5;]

ในทางปฏิบัติ ฟิลเตอร์อินทิเกรตแอนด์ดัมพ์ (Integrate and dump filter) เป็นวงจรๆ หนึ่งที่ใช้ประยุกต์แทนแมทช์ฟิลเตอร์



รูปที่ 6.5.1 วงจรอินทิเกรตแอนด์ดัมพ์

จากรูป 6.5.1 สวิทช์ S1 จะถูกปิดเป็นจังหวะเป็นเวลาสั้นๆ เพื่อ discharge คาร์ปาซิเตอร์ และวงจร RC ที่มี time constant มีค่ามากๆ นี่ จะทำหน้าที่เป็นอินทิกเกรเตอร์ (integrator)

เพื่อให้เกิดความเข้าใจในการทำงานของฟิลเตอร์ชนิดนี้ เราจะป้อนสัญญาณ x5 จาก การทดลองที่ 6.1 ให้กับฟิลเตอร์ และสังเกตุสัญญาณอินพุทและเอาท์พุทของฟิลเตอร์

[y5 = int_dump(x5);
clf, waveplot(x5), hold on, waveplot(5*y5)]

คำถาม จากรูปที่ได้ จงระบุว่าส่วนไหนของสัญญาณที่คาร์ปาซิเตอร์ charge และdischarge ส่วนไหนที่สวิทช์ S1 ปิดวงจร และเราสามารถหาค่า sampling instant ที่เหมาะสมได้หรือไม่ ถ้าไม่พิจารณาแพทเทิรนรูปตางตา

กำหนดสัญญาณ y ใน การทดลองที่ 6.3 และป้อนให้กับวงจรอินทิเกรตแอนด์ดัมพ์ เปรียบเทียบค่า BER ที่ได้กับค่า BER ที่ใช้ฟิลเตอร์ผ่านกานถี่ต่ำและที่ใช้แมทช์ฟิลเตอร์ จาก การทดลองที่ 6.4 และ การทดลอง 6.3

รูปที่ 40 ไฟล์ lab6_5.doc

การทดลองที่ 7 การมอดูเลตสัญญาณดิจิตอล

วัตถุประสงค์
หนังสืออ้างอิง
กลับไปยังสารบัญการทดลอง

สารบัญการทดลองย่อย

การทดลองที่ 7.1 การมอดูเลตสัญญาณดิจิตอลแบบต่าง ๆ
การทดลองที่ 7.2 การตรวจจับสัญญาณดิจิตอลที่ถูกมอดูเลต
การทดลองที่ 7.3 สมรรถนะของระบบในสภาวะมีสัญญาณรบกวน

ในการทดลองนี้ มีการเซ็ตค่าต่างๆ ดังนี้

อัตราข้อมูลไบนารี (binary data rate, R_b) = 1 kbps

ขนาดสูงสุดของสัญญาณที่ถูกมอดูเลต = 1 V

คาบเวลาของแต่ละบิต (bit period, T_b) = $1/R_b$ (แทนด้วย 100 samples)

วัตถุประสงค์

ในการทดลองนี้ เป็นการประยุกต์ใช้ความรู้ในเรื่องการส่งสัญญาณดิจิตอลแบบเบสแบนด์ (baseband) และการมอดูเลตคลื่นแบบต่อเนื่อง เพื่อศึกษาการส่งสัญญาณดิจิตอลแบบแบนด์พาส (band pass) โดยมีเนื้อหาดังนี้

- รูปแบบต่างๆ ของรูปคลื่นของสัญญาณดิจิตอลที่ถูกมอดูเลต
- การตรวจจับสัญญาณที่ถูกมอดูเลตทั้งแบบ coherent (synchronous) และ noncoherent (eavelope)
- สมรรถนะของระบบในสภาวะที่มีสัญญาณรบกวน

หนังสืออ้างอิง

1. หนังสือ "หลักการไฟฟ้าสื่อสาร" ของ รศ.ดร บัณฑิต ไวจ์อารยานนท์ สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย หน้า 313-345
2. หนังสือ "Communication Systems," Third edition, ของ A.B.Carlson, McGraw-Hill,1986, หน้า 512-542
3. หนังสือ "Digital and Analog Communication Systems," Third edition, L.W. Couch II, MacMillan,1990 หน้า 331-344 และ 532-551
4. หนังสือ "An Introduction to Analog & Digital Communications," S.Haykin, Wiley & Sons, 1989 หน้า 368-374 และ 574-583

รูปที่ 41 ไฟล์ lab7.doc

7.1 การมอดูเลตสัญญาณดิจิตอลแบบต่างๆ

PHASE-SHIFT KEYING(PSK)
FREQUENCY-SHIFT KEYING (FSK)

Amplitude-Shift Keying (ASK)

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init7_1; ]
```

สุ่มลำดับไบนารี 50 ค่า โดยที่ 5 บิตแรกมีค่าเป็น [1 0 0 1 0]

```
[ b = [ 1 0 0 1 0 binary(45) ]; ]
```

กำเนิดสัญญาณ ASK (sa) ที่มีความถี่คลื่นพาห์ (carrier frequency) เท่ากับ 8 kHz. โดยมีขั้นตอนดังนี้

- จากลำดับไบนารี b กำเนิดสัญญาณเบสแบนด์ชนิด unipolar NRZ (xu)
- mix สัญญาณ xu กับเอาท์พุทของออสซิเลเตอร์ที่ทำงานที่ความถี่ 8 kHz.

```
[ xu = wave_gen(b,'unipolar_nrz');
  sa = mixer(xu,osc(8000)); ]
```

หลังจากนั้น พล็อตรูปคลื่น xu และ sa 500 ค่าแรก ซึ่งแทน 5 บิตแรกของลำดับไบนารี b เปรียบเทียบรูปคลื่นทั้งสอง

```
[ tt = [1:500];
  subplot(211), waveplot(xu(tt))
  subplot(212), waveplot(sa(tt)) ]
```

ต่อไป พล็อตฟังก์ชัน PSD ของสัญญาณ xu และ sa โดยพล็อตในช่วงความถี่ [0, 20 kHz.] โดยใช้ฟังก์ชัน psd

```
[ fr = [0, 2000];
  subplot(211), psd(xu,fr)
  subplot(212), psd(sa,fr) ]
```

Phase-Shift Keying (PSK)

กำเนิดสัญญาณ PSK (sp) ที่มีความถี่คลื่นพาห์เท่ากับ 8 kHz. โดยมีขั้นตอนดังนี้

- จากลำดับไบนารี b กำเนิดสัญญาณเบสแบนด์ชนิด polar NRZ (xp)
- mix สัญญาณ xp กับเอาท์พุทของออสซิเลเตอร์ที่ทำงานที่ความถี่ 8 kHz

```
[ xp = wave_gen(b,'polar_nrz');
  sp = mixer(xp,osc(8000)); ]
```

พล็อตรูปคลื่น 500 ค่าแรกของสัญญาณ xp และ sp

```
[ subplot(211), waveplot(xp(tt))
  subplot(212), waveplot(sp(tt)) ]
```

คำถาม ที่เวลาระหว่างคาบของบิตที่ 1 และ 2 สัญญาณ sp และคลื่นพาห์ sin(2*pi*f_c*t) มีเฟสต่างเฟสกันอยู่เท่าไร

พล็อตฟังก์ชัน PSD ของสัญญาณ xp และ sp ในช่วงความถี่ [0, 20 kHz.]

```
[ subplot(211), psd(xp,fr)
  subplot(212), psd(sp,fr) ]
```

Frequency-Shift Keying (FSK)

กำเนิดสัญญาณ continuous phase FSK (sf) ที่มีความถี่ mark และ space เท่ากับ 4 และ 8 kHz. ตามลำดับ โดยมีขั้นตอนดังนี้

- จากลำดับไบนารี b กำเนิดสัญญาณเบสแบนด์ชนิด polar NRZ (xp)
- ป้อนสัญญาณ xp ให้กับ voltage controlled oscillator (VCO) โดยในการทดลองนี้ VCO ได้ตั้ง free-running frequency ไว้ที่ 6 kHz และมีค่า frequency sensitivity เท่ากับ -2 kHz/V

รูปที่ 42 ไฟล์ lab7_1.doc

```
[ sf = vco(xp); ]

พล็อตรูปคลื่นของสัญญาณ xp และ sf ในช่วง 0 < t < 5T_b
[ subplot(211), waveplot(xp(fr))
  subplot(212), waveplot(sf(fr)) ]

พล็อตฟังก์ชัน PSD ของสัญญาณ FSK (sf)
[ clf
  psd(sf,fr) ]
```

ข้อสังเกต เราสามารถนำสัญญาณ ASK 2 สัญญาณ(2 ความถี่) มาทำมอดิสสัญญาณ FSK ได้อย่างไร และสำหรับระบบที่ต้องการใช้แบนด์วิดท์ให้มีประสิทธิภาพที่สุด วิธีการมอดูเลตแบบไหนถึงจะเหมาะสม

รูปที่ 43 ไฟล์ lab7_1.doc (ต่อ)

7.2 การตรวจจับสัญญาณดิจิตอลที่ถูกมอดูเลต

Coherent Detection

ตัวอย่างประกอบ

```
กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง
[ init7_2; ]

อันดับแรก นำสัญญาณ sa จากการทดลองที่แล้วมาคูณกับคลื่นพาห์ของเครื่องรับซึ่งมีความถี่และเฟสต่างสัมพัทธ์กับคลื่นพาห์ที่ใช้ในการกำเนิดสัญญาณ sa
ต่อจากนั้น พล็อตรูปคลื่นสัญญาณ ya ที่ได้นเอาท์พุทของวงจรคูณ (multiplier) จำนวน 5 บิตแรก และพล็อตฟังก์ชัน PSD ในความถี่ช่วง (0, 20] kHz
[ ya = mixer(sa, osc(8000));
  clf, subplot(211), waveplot(ya(fr))
  subplot(212), psd(ya,fr) ]

ป้อนสัญญาณ ya ให้กับแมทช์ฟิลเตอร์ (matched filter) และพล็อตค่าเอาท์พุทของแมทช์ฟิลเตอร์ที่เวลา 0 < t < 5T_b
[ za = match('unipolar_nrz',ya);
  subplot(212), waveplot(za(fr)) ]

คำถาม จงหา unpulse response ของแมทช์ฟิลเตอร์ และสังเกตสัญญาณ za จะมีลักษณะคล้ายกับรูปคลื่นเอาท์พุทของแมทช์ฟิลเตอร์ที่ใช้สำหรับสัญญาณชนิด unipolar NRZ หรือไม่ ทำไมจึงเป็นเช่นนั้น

ต่อไป สังเกตดูผลกระทบของความคลาดเคลื่อนทางเฟสของการดีมอดูเลตสัญญาณ sa โดยใช้ออสซิเลเตอร์ที่ภาครับซึ่งมีเอาท์พุทเป็น sin(2*pi*f_c*t+∅) ในที่นี้ ∅ คือค่าเฟสที่คลาดเคลื่อนไปเมื่อเทียบกับคลื่นพาห์ที่ใช้มอดูเลต บันทึกขนาดสูงสุดของสัญญาณที่ค้นเอาท์พุทของแมทช์ฟิลเตอร์ในกรณีเกิดความคลาดคลื่นทางเฟสต่างๆ ดังในตาราง 7.2.1
[ ya1 = mixer(sa,osc(8000,phase_error));
  za1 = match('unipolar_nrz',ya1);
  subplot(212), waveplot(za1(1:500)) ]
```

รูปที่ 44 ไฟล์ lab7_2.doc

ตาราง 7.2.1

ค่าความคลาดเคลื่อนเฟส	ขนาดสูงสุดของสัญญาณ (V)
0°	
20°	
60°	
80°	
120°	

คำถาม เนื่องจากค่า BER จากการตรวจจับสัญญาณในสภาวะที่มีสัญญาณรบกวน มีค่าขึ้นกับขนาดสูงสุดของสัญญาณเอาท์พุทของแมทช์ฟิลเตอร์ จากตาราง 7.2.1 ที่ค่าความคลาดเคลื่อนเฟสเท่าใดที่จะทำให้ค่า BER มีค่าน้อยที่สุด

ดีมอดูเลตสัญญาณ sa ที่มีค่าความคลาดเคลื่อน 60° และ 120° และถอดรหัสสัญญาณเอาท์พุทของแมทช์ฟิลเตอร์เพื่อที่จะหาอำดับ b จำนวน 5 บิตแรก และบันทึกอำดับที่ถูกถอดรหัสแล้วและเปรียบเทียบความแตกต่าง

Phase error = 60° $\hat{b}_{1-5}$ =

Phase error = 120° $\hat{b}_{1-5}$ =

ต่อไป สังเกตดูผลกระทบของความคลาดเคลื่อนทางความถี่ ของการมอดูเลตสัญญาณ ASK โดยมอดูเลตสัญญาณ sa ด้วยออสซิเลเตอร์ที่ภาครับที่ตั้งความถี่ไว้ที่ 7,900 Hz. พล็อตสัญญาณ ya1 และ ya2 และเปรียบเทียบสัญญาณทั้งสอง

```
[ ya2 = match('unipolar_nrz',mixer(sa,osc(7900)));
  subplot(211), waveplot(ya1(fr))
  subplot(212), waveplot(ya2(fr)) ]
```

คำถาม เราสามารถถอดรหัสสัญญาณ ya2 ได้อย่างถูกต้องหรือไม่

พิจารณาอีกกรณีหนึ่งซึ่งความถี่ของออสซิเลเตอร์ที่ภาครับถูกตั้งไว้ที่ 7,985 Hz. ดีมอดูเลตสัญญาณ sa และกำเนิดสัญญาณเอาท์พุทของแมทช์ฟิลเตอร์

```
[ ya3 = match('unipolar_nrz',mixer(sa,osc(7985)));
  subplot(211), waveplot(ya1)
  subplot(212), waveplot(ya3) ]
```

คำถาม หาค่าความถี่ของ envelope ของสัญญาณเอาท์พุทของแมทช์ฟิลเตอร์

คำถาม พิจารณาสัญญาณ ASK $s_1(t)$ ที่ถูกมอดูเลตด้วยคลื่นพาห์ที่ความถี่ f_c ถ้าสัญญาณ $s_1(t)$ ถูกดีมอดูเลตโดยคูณกับสัญญาณเอาท์พุทของออสซิเลเตอร์ภาครับที่ตั้งความถี่ไว้ที่ f_o โดยที่ $f_o \neq f_c$ แล้ว จงหา(envelope) ของเอาท์พุทจากแมทช์ฟิลเตอร์ของตัวดีเทคเตอร์จะถูกมอดูเลตด้วยสัญญาณไซน์อีกที จงหาความถี่ของสัญญาณเอาท์พุทนี้ในรูปของฟังก์ชัน f_c และ f_o

Noncoherent Detection

ตัวอย่างประกอบ

```
ตั้งค่าเบนด์วิดท์ของ LPF ที่อยู่ในฟังก์ชัน envelope ให้มีค่าเท่ากับ 4,000 Hz. ป้อนสัญญาณ ASK (sa) ให้กับฟังก์ชัน envelope และพล็อตสัญญาณ sa และสัญญาณเอาท์พุทของฟังก์ชัน envelope
[ ya4 = envelope(sa,4000);
  clf, subplot(211), waveplot(sa(fr))
  subplot(212), waveplot(ya4(fr)) ]
```

คำถาม จงถอดรหัสสัญญาณที่ถูกส่งมา 5 บิตแรก

คำถาม เราสามารถนำ noncoherent detector มาใช้กับสัญญาณ PSK ได้หรือไม่

รูปที่ 45 ไฟล์ lab7_2.doc (ต่อ)

7.3 สมรรถนะของระบบในสภาวะมีสัญญาณรบกวน

กำหนดค่าตัวแปรที่ใช้ในการทดลอง

```
[ init7_3; ]
```

กำเนิดสัญญาณ ASK ที่แทนอำดับไบนารี 500 ค่า

```
[ b = [1 0  0 1 0 binary(495)];
  sa = mixer(wave_gen(b,'unipolar_nrz'), osc(8000)); ]
```

ส่งสัญญาณ sa ผ่านช่องสัญญาณที่มีอัตราขยาย (gain) เท่ากับ 1 มีกำลังสัญญาณรบกวน (σ_n^2) เท่ากับ 1 วัตต์ และมีแบนด์วิดท์ที่เพียงพอจนไม่เกิดการผิดเพี้ยนทางสัญญาณ (distortion) เกิดขึ้น พล็อตสัญญาณ ASK (sa) และสัญญาณเอาท์พุทของช่องสัญญาณ y ในช่วง $0 < t < 5T_b$

```
[ y = channel(sa,1,1.5,49000);
  subplot(211), waveplot(sa(rt))
  subplot(212), waveplot(y(rt)) ]
```

ใช้ coherent detector คืบอดูเอดสัญญาณ y และพล็อตแพทเทิร์นรูปตาของสัญญาณเอาท์พุทของแมทช์ฟิลเตอร์

```
[ zm = match('unipolar_nrz',mixer(y,osc(8000)));
  clf, eye_diag(zm); ]
```

จากแพทเทิร์นรูปตาวงตา หาค่า sampling instant ที่เหมาะสมและค่า threshold ป้อนสัญญาณ zm ไปให้กับวงจรตัดสินใจ (decision circuit) โดยใช้ฟังก์ชัน detect และบันทึกค่าความน่าจะเป็นการผิดพลาดข้อมูล (probability of bit error : BER)

```
detect(zm,vth,sampling_instant,b);
```

Coherent detection : P_e =

คำถาม คำนวณค่าความน่าจะเป็นการผิดพลาดข้อมูลในการทฤษฎีในกรณีของการทดลองนี้ โดยที่ฟังก์ชัน PSD ของสัญญาณรบกวนของช่องสัญญาณเป็นดังนี้

$$S_n(f) = \frac{N_o}{2} = \frac{\sigma_n^2}{2 \times system\_bandwidth}$$

system_bandwidth ในการทดลองนี้คือ 50 kHz.

ใช้ noncoherent detection ถอดรหัสสัญญาณเอาท์พุทของช่องสัญญาณ (y) และเปรียบเทียบค่า BER ที่ได้กับในกรณีที่แล้ว (coherent)

```
[ ze = envelope(y,4000);
  detect(ze,vth,sampling_instant,b); ]
```

รูปที่ 46 ไฟล์ lab7_3.doc

การทดลองที่ 8 การสื่อสารแบบดิจิตอล

วัตถุประสงค์
หนังสืออ้างอิง
กลับไปยังสารบัญการทดลอง

สารบัญการทดลองย่อย

บทนำก่อนทำการทดลอง
การทดลองที่ 8.1 A/D-D/A Conversion
การทดลองที่ 8.2 Differential Encoding
การทดลองที่ 8.3 การสื่อสารที่ความถี่เบสแบนด์
การทดลองที่ 8.4 การสื่อสารที่ความถี่แบนด์พาส

วัตถุประสงค์

ในการทดลองนี้ จะนำฟังก์ชันต่างๆ ในการทดลองที่ผ่านๆ มา ประกอบกันจนเป็นระบบการสื่อสารแบบดิจิตอล โดยเฉพาะ

- ระบบสื่อสารแบบดิจิตอลแบบ baseband และ band-pass
- ความสัมพันธ์ระหว่าง A/D - D/A Conversion และเครื่องส่งสัญญาณกับเครื่องรับสัญญาณ
- Phase reversal เมื่อใช้ differential encoding
- การเลือกค่าพารามิเตอร์ต่างๆ ของเครื่องส่งสัญญาณเมื่อพิจารณาจากคุณลักษณะของช่องสัญญาณสื่อสาร

หนังสืออ้างอิง

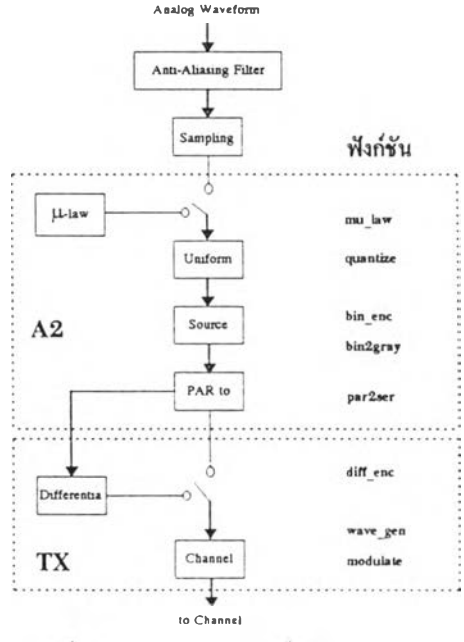
1. หนังสือ "หลักการไฟฟ้าสื่อสาร" ของ รศ.ดร บัณฑิต ไรจน์อารยานนท์ สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย หน้า
2. หนังสือ "Communication Systems," Third edition ของ A.B Carlson, McGraw-Hill,1986. หน้า 430-439 และ 512-532
3. หนังสือ "Digital and Analog Communication Systems," Third edition., MacMillan Publishing, 1990. หน้า 130-144 และ 547
4. หนังสือ "An Introduction to Analog & Digital Communications," J. Wiley & Sons Inc. 1989 หน้า 177-202 และ 539-580

รูปที่ 47 ไฟล์ lab8.doc

บทนำก่อนการทดลองที่ 8

Analog Waveform to Channel Code Transformation

บล็อกไดอะแกรมที่แสดงในรูปที่ 1 แสดงถึงการแปลงสัญญาณอนาล็อกไปเป็นสัญญาณดิจิตอลและเปลี่ยนไฟล์ในฟอร์มที่เหมาะสมกับคุณลักษณะของช่องสัญญาณ ในรูป จะมีบล็อกหลักๆ ที่ใช้คือ A/D converter (a2d) และ transmitter (tx)



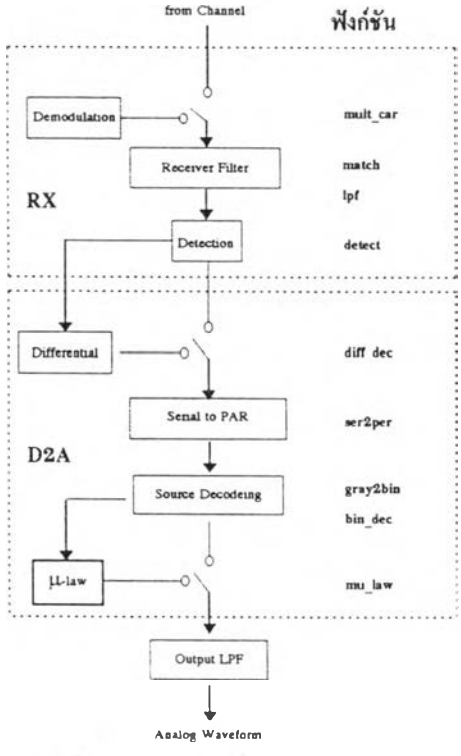
รูปที่ 1 บล็อกไดอะแกรมของ A/D และเครื่องส่งสัญญาณ

รูปที่ 48 ไฟล์ lab8_0.doc

Channel Output to Analog Waveform

Transformation

บล็อกไดอะแกรมในรูปที่ 2 แสดงถึงขบวนการถอดรหัสสำหรับในบาริทีส่งไดเมเครื่องรับสัญญาณจะรับสัญญาณที่มาจากเอาท์พุทของช่องสัญญาณ และจั่นสุดท้าย ค่าสำหรับในบาริทีถอดรหัสได้จะถูกแปลงเป็นสัญญาณอนาล็อก ในรูปจะมีบล็อกหลักๆ อยู่ 2 บล็อกคือ บล็อกเครื่องรับสัญญาณ(ฟังก์ชัน rx) และบล็อก D/A converter (ฟังก์ชัน d2a)



รูปที่ 2 บล็อกไดอะแกรมของเครื่องรับสัญญาณและ D/A

8.1 A/D - D/A Conversion

คำอธิบายประกอบ

กำหนดค่าตัวแปรที่ใช้ในการทดลอง

```
[ init8_1; ]
```

ในการทดลองนี้ จะทดสอบว่าฟังก์ชัน a2d และ d2a เป็นฟังก์ชันอินเวอร์ส (inverse) ของกันและกัน โดยกำเนิดสัญญาณเสียงชนิดหนึ่งจำนวน 100 ค่า

```
[ s = speech(100);
  s_binary = a2d(s,6);
  s_analog = d2a(s_binary,6); ]
```

ตรวจสอบดูว่า s_binary คือค่าสำหรับในบาริจริงหรือไม่ โดยแสดงค่าของตัวแปร s_binary มาหนึ่งจำนวนหนึ่ง

```
[ s_binary(1:10) ]
```

เปรียบเทียบสัญญาณข่าวสาร s กับสัญญาณเอาท์พุทที่ได้เมื่อผ่าน A/D และ D/A Converter : s_analog

```
[ subplot(211), waveplot(s)
  subplot(212), waveplot(s_analog) ]
```

คำถาม จากขบวนการแปลงสัญญาณจากสัญญาณอนาล็อกไปเป็นสัญญาณดิจิตอลและแปลงกลับมาเป็นสัญญาณอนาล็อกอีกทีหนึ่ง จะก่อให้เกิดการผิดเพี้ยนทางสัญญาณหรือไม่ ถ้าเกิดการผิดเพี้ยนขึ้น จงแยกแยะบาริเกิดการผิดเพี้ยนของสัญญาณที่ขั้นตอนไหนบ้างในระบบ analog-digital-analog conversion นี้ และนิพารามิเตอร์ใดบ้างที่มีผลโดยตรงต่อการเกิดการผิดเพี้ยนของสัญญาณ

รูปที่ 49 ไฟล์ lab8_0.doc (ต่อ)

รูปที่ 50 ไฟล์ lab8_1.doc

8.2 Differential Encoding

ในการส่งข้อมูลแบบ serial ผ่านช่องสัญญาณสื่อสารที่ประกอบด้วยวงจรต่างๆ จำนวนมาก อาจเกิดการกลับเฟสสัญญาณ 180 องศาขึ้นโดยไม่ได้ตั้งใจ เช่นในกรณีการสลับขั้วของสาย (twisted-pair) เป็นต้น ในการทดลองนี้ เป็นการศึกษาการแก้ไขปัญหาดังกล่าวโดยใช้การเข้ารหัสแบบดิฟเฟอเรนเชียล

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init8_2; ]
```

กำเนิดสัญญาณไบนารีจำนวน 100 ค่า แปลงสัญญาณเป็นสัญญาณแบบดิจิตอล และเตรียมสัญญาณไบนารีที่จะใช้ส่งผ่านช่องสัญญาณแบบเบสแบนด์โดยใช้โอนไค้ดแบบ manchester โดยเรียกใช้ฟังก์ชัน tx

```
[ x = sin(2*pi*400*[1:100]/SAMPLING_FREQ); ]
[ x_pcm = a2d(x,6);
  xw = tx(x_pcm,'manchester','no_diff',1000); ]
```

ส่งสัญญาณ xw ผ่านช่องสัญญาณ ideal ที่มีแบนด์วิดท์ 19,900 Hz. และมีกำลังสัญญาณรบกวนเท่ากับ 0.01 W

```
[ y = channel(xw,1,0.01,19000); ]
```

ถอดรหัสสัญญาณเอาท์พุทของช่องสัญญาณโดยใช้เบท์ซฟิลเตอร์ตามด้วยการตรวจจับสัญญาณ (detection) และต่อด้วย D/A conversion โดยใช้ฟังก์ชัน rx เปรียบเทียบรูปคลื่น x และ x_analog

```
[ x_digital = rx(y,'manchester');
  x_analog = d2a(x_digital,6);
  subplot(211), waveplot(x)
  subplot(212), waveplot(x_analog) ]
```

คำถาม ในระหว่างการแปลงสัญญาณจากอนาล็อกเป็นสัญญาณดิจิตอล ลองเปลี่ยนชนิดการเข้ารหัสจากเดิมแบบ Natural เป็นรหัสแบบ Gray และส่งสัญญาณผ่านช่องสัญญาณอินเวอร์ติ้ง แล้วสังเกตดูรูปคลื่นที่ภาครับสัญญาณ เปรียบเทียบกับรูปคลื่นของสัญญาณเสียงก่อนส่ง พร้อมทั้งอธิบายความแตกต่างที่เกิดขึ้น

ต่อไป จะเปลี่ยนขบวนการต่างๆ ข้างบนมาใช้ differential encoding โดยเปลี่ยนค่าพารามิเตอร์อินพุทของฟังก์ชัน tx และ rx ดังนี้

```
[ u = tx(x_pcm,'manchester','diff',1000);
  z = -channel(u,1,0.01,19000);
  u_digital = rx(z,'manchester','diff');
  u_analog = d2a(u_digital,6);
  subplot(211), waveplot(x)
  subplot(212), waveplot(u_analog) ]
```

คำถาม อธิบายความสำคัญของการป้องกันการกลับเฟส 180 องศา ของสัญญาณอนาล็อกและสัญญาณดิจิตอล

คำถาม โอนไค้ดชนิดไหนที่ไม่รับผลกระทบจากการส่งสัญญาณผ่านช่องสัญญาณในกรณีเป็นช่องสัญญาณแบบอินเวอร์ติ้ง

8.3 การสื่อสารที่ความถี่เบสแบนด์

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init8_3; ]
```

ในการทดลองนี้ เราจะหาอัตราการผิดพลาดของข้อมูล (BER) ที่เกิดจากการส่งข้อมูลผ่านช่องสัญญาณสื่อสารที่ความถี่เบสแบนด์

ลำดับแรก สุ่มลำดับไบนารีจำนวน 1,000 ค่า กำเนิดสัญญาณจากลำดับไบนารีนี้โดยใช้โอนไค้ดชนิด unipolar NRZ และ manchester ที่อัตราข้อมูล R_b = 1 kbps สำหรับสัญญาณที่จะส่งผ่านช่องสัญญาณที่ความถี่เบสแบนด์

```
[ b = binary(1000);
  Rb = 1000;
  u = tx(b,'unipolar_nrz',Rb);
  m = tx(b,'manchester',Rb); ]
```

พิจารณาช่องสัญญาณที่มีลักษณะผ่านความถี่ต่ำ (low-pass communication channel) โดยที่

- channel gain = 0 dB
- channel noise power σ_n^2 = 1
- channel bandwidth = 19 kHz

กำเนิดสัญญาณเอาท์พุทของช่องสัญญาณนี้และถอดรหัสลำดับไบนารีที่ถูกส่งมาโดยใช้ฟังก์ชัน rx โดยค่า A และ ch_input เป็นไปตามตาราง 8.3.1 และบันทึกค่า

```
[ ch_output = channel(A*ch_input,1,1,19000);
  rx(ch_output,'linecode',b); ]
```

ตารางที่ 8.3.1

A (โวลต์)	UNIPOLAR NRZ		MANCHESTER	
	E_b / N_o	P_e	E_b / N_o	P_e
0.2				
0.3				
0.4				
0.5				
0.6				
0.7				

*** $N_o = \sigma_n^2 / 20,000$ (จากการทดลองที่ 5)

เปรียบเทียบค่า P_e กับค่าที่คำนวณได้ตามทฤษฎี และพล็อตกราฟเปรียบเทียบกราฟทั้งสองบนแกนเดียวกัน

พิจารณาช่องสัญญาณสื่อสารที่ความถี่เบสแบนด์ในคำสั่งต่อไปนี้

```
[ ch_output = channel(ch_input,1,2,19000); ]
```

คำถาม จงหาค่า N_o เมื่อกำหนดให้ σ_n^2 = 2 และ R_b = 1 kbps และถ้าอินพุทของช่องสัญญาณเป็นสัญญาณ u จงหาค่ากำลังในการส่งสัญญาณที่ต้องการ(ในเทอมของ E_b) ที่จะทำให้ $P_e \leq 10^{-3}$ และหาในกรณีที่อินพุทคือสัญญาณ m ด้วย

รูปที่ 51 ไฟล์ lab8_2.doc

รูปที่ 52 ไฟล์ lab8_3.doc

8.4 การสื่อสารที่ความถี่แบนด์พาส

กำหนดค่าตัวแปรต่างๆ ที่ใช้ในการทดลอง

```
[ init8_4; ]
```

กำเนิดสัญญาณเสียงขนาด 100 ค่า

```
[ s = speech(100); ]
```

ในการทดลองนี้ เราจะเตรียมสัญญาณในบาริของสัญญาณ s สำหรับส่งผ่านช่องสัญญาณแบบแบนด์พาส โดยมีพารามิเตอร์ต่างๆ ที่ระบุในเครื่องส่งสัญญาณและช่องสัญญาณดังนี้

- A/D conversion - a2d
  8 bit, μ-law quantization
- เครื่องส่งสัญญาณ - tx
  ระบบการมอดูเลตแบบดิจิตอล : PSK
  อัตราข้อมูล R_b = 100 kbps
- ช่องสัญญาณ - channel
  อัตราขยายของช่องสัญญาณ : 0 dB
  กำลังของสัญญาณรบกวน : 1 W
  แบนด์วิดท์ : 600 kHz ถึง 1,400 kHz.
- Bit Error Rate
  $P_e = 10^{-2}$

ความสัมพันธ์ระหว่างสัญญาณอินพุทและเอาท์พุทของช่องสัญญาณเป็นดังนี้

```
[ out = channel(A*in, 1, 1, [600e3, 1400e3] ); ]
```

<u>คำแนะ</u> จาก $No = \sigma_n^2/(20R_b)$ โดยที่ $\sigma_n^2 = 1$ และ A = ขนาดของรูปคลื่น จงคำนวณหาค่า SNR ของช่องสัญญาณ, E_b/N_o เพื่อให้ได้ค่า BER ที่ต้องการ กำหนดค่ากำลังของเครื่องส่งสัญญาณสัญญาณรบกวนของช่องสัญญาณ และความถี่ของคลื่นพาห์ และใช้ฟังก์ชัน MATLAB คำนวณหาค่า BER และเปรียบเทียบกับค่า BER ต้องการ

E_b	N_o	f_c	P_e

ขั้นสุดท้าย แปลงสัญญาณเอาท์พุทของเครื่องรับสัญญาณ (ฟังก์ชัน rx) ไปเป็นสัญญาณอนาล็อกโดยใช้ฟังก์ชัน d2a และเปรียบเทียบสัญญาณเสียงเดิม s กับสัญญาณที่สร้างกลับคืนมา

<u>คำถาม</u> ค่า BER ที่คำนวณได้จากฟังก์ชัน MATLAB กับค่า BER ทางทฤษฎีมีค่าต่างกันหรือไม่ ถ้าต่างกันมาก จงอธิบายเหตุผล

<u>รูปที่ 53</u> ไฟล์ lab8_4.doc

ประวัติผู้เขียน

นายสมบูรณ์ พฤกษ์ไพรผดุง เกิดวันที่ 2 สิงหาคม พ.ศ 2514 จังหวัดกรุงเทพมหานคร จบการศึกษาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ เมื่อปี พ.ศ. 2537

พ.ศ. 2537 ได้เข้าศึกษาต่อระดับปริญญามหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย