Sprawozdanie Lab06

Autor: Kamil Szóstak

Demodulacja ASK

```
import math
import numpy as np
import matplotlib.pyplot as plt
import array
import math
import numpy as np
import matplotlib.pyplot as plt
import array


def Binary(string,switch):
    if (switch == 0):
        bin = ''.join(format(i, 'b') for i in bytearray(string, encoding ='utf-8'))
        bin = list(map(int, bin))
        #print('variant=littleEndian\n','conversion of ', string,' to binary is equal to [ ', bin,' ]')
        return bin
    else:
        rev=string[::-1]
        bin2 = ''.join(format(i, 'b') for i in bytearray(rev, encoding ='utf-8'))
        bin2 = list(map(int, bin2))
        #print('variant=BigEndian\n','conversion of ', string,' to binary is equal to [ ',''.join(bin2),'
]')
        return bin2


def MT(LIMIT):
    ycords=[] ; xcords = []
    for i in range (LIMIT):
```

```
        if(mt[i]==0):
            for x in np.linspace(1/10,2/10):
                ycords.append(0)
        else:
            for x in np.linspace(1/10,2/10):
                ycords.append(1)
    xcords=np.linspace(0,1,len(ycords))
    plt.subplot(231)
    plt.title('Sygnal wejsciowy')
    plt.plot(xcords,ycords)

def ASK(LIMIT):
    ycords=[] ; xcords = []
    for i in range (LIMIT):
        if(mt[i]==0):
            for x in np.linspace(1/10,2/10):
                ycords.append(0)
        else:
            for x in np.linspace(1/10,2/10):
                ycords.append(np.sin(40*np.pi*(x  - 1/10 )*1))
    xcords=np.linspace(0,1,len(ycords))
    plt.subplot(232)
    plt.title('ASK')
    plt.plot(xcords,ycords)
    return ycords

def demodulator1(Modulacja,h):
    Demo=[] ; Final=[] ;  Demov2 = []
    sum=0 ; sumabs = 0
```

```
for i in range (len(Modulacja)):
    if (Modulacja[i] == 0.0000000):
        sum = 0
    sum=sum+Modulacja[i]
    Demo.append(sum)

xcords = np.linspace(0,1,len(Demo))
plt.subplot(235)
plt.title('Demodulacja ASK x(t)')
plt.plot(xcords,Demo)

for i in range (len(Modulacja)):
    if (Modulacja[i] == 0.0000000):
        sumabs = 0
    if (Modulacja[i]>h):
        sumabs=sumabs+(Modulacja[i])
    Demov2.append(sumabs)

xcords = np.linspace(0,1,len(Demov2))
plt.subplot(236)
plt.title('Demodulacja ASK p(t)')
plt.plot(xcords,Demov2)

for i in range (len(Demov2)):
    roundmt=round(Demov2[i],0)
    if(roundmt==0):
        for x in np.linspace(1/10,2/10):
            Final.append(0)
```

```
        else:
            for x in np.linspace(1/10,2/10):
                Final.append(1)

    xcords=np.linspace(0,1,len(Final))
    plt.subplot(234)
    plt.title('Demodulacja ASK m(t)')
    plt.plot(xcords,Final)

plt.figure()
mt=Binary('Kamil',0)
tb=0.1 ;N=10;f=N*(tb**(-1));f0=(N+1)/tb;f1=(N+2)/tb;A=1;A1=0;A2=1
LIMIT=10
MT(LIMIT)
ycords=ASK(LIMIT)
demodulator1(ycords,0.2)
plt.show()
```

Demodulacja FSK:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

def Binary(string):
    bin = ''.join(format(i, 'b') for i in bytearray(string, encoding ='utf-8'))
    bin = list(map(int, bin))
    #print('variant=littleEndian\n','conversion of ', string,' to binary is equal to [ ', bin,' ]')
    return bin
```

```
def FSK():
    FSK=[]
    for i,j in zip(TBs,t):
        if i == 1:
            FSK.append(A1 * np.sin(2 * np.pi *f*j + fi))
        if i==0:
             FSK.append(A1 * np.sin(2 * np.pi *f1*j + fi))
    return FSK

def demo():
    DemoX1=[] ; DemoX2=[]
    for i,j in zip(FSK,t):
        DemoX1.append(i* A1 * np.sin(2 * np.pi *f1*j + fi))

    for i,j in zip(FSK,t):
        DemoX2.append(i* A1 * np.sin(2 * np.pi *f2*j + fi))

    pt1 = []
    for i in range(z1):
        x0 = 0
        for j in range(50):
            x0 = x0 + DemoX1[(i * 50) + j]
        pt1.append(x0)

    pt2 = []
    for i in range(z1):
        x1 = 0
        for j in range(50):
            x1 = x1 + DemoX2[(i * 50) + j]
```

```
        pt2.append(x1)

    pt = []
    for i in range(z1):
        pt.append(pt1[i] - pt2[i])

    interpolatingFSK=interp1d(x, pt, kind='previous')
    FSK_pt=interpolatingFSK(t)

    return DemoX1,DemoX2,FSK_pt

def wartoscProgowa (pt,h):
    wp = []
    for p in pt:
        if p < h:
            wp.append(1)
        else:
            wp.append(0)
    return wp

mt=Binary('Kamil')
plt.figure()
fi0=0 ; fi1=np.pi;fi = np.pi;A1=1;A2=0.3;Tb=1 ;N=1/Tb;f = N * (Tb ** -1)
f1 = (N + 1)/Tb;f2 = (N + 2)/Tb;x=50;z1=len(mt);prb=x*(z1/Tb);prb1=int(prb)
t = np.linspace(0,z1,prb1);x = np.linspace(0,z1,z1);h=10

interpolacja = interp1d(x, mt, kind='previous')
TBs = interpolacja(t)
plt.subplot(511)
plt.title('sygnal wejsciowy')
```

```
plt.plot(t,TBs)

FSK=FSK()
plt.subplot(512)
plt.title('FSK')
plt.plot(t,FSK)

[DemoX1,DemoX2,FSK_pt]=demo()
FSKwp=wartoscProgowa(FSK_pt,h)

plt.subplot(513)
plt.title('Demodulacja FSK x1(t)')
plt.plot(t,DemoX1)

plt.subplot(514)
plt.title('Demodulacja FSK x2(t)')
plt.plot(t,DemoX2)

plt.subplot(515)
plt.title('Demodulacja FSK p(t)')
plt.plot(FSKwp)

plt.show()
```

Demodulacja PSK:

```
import math
import numpy as np
import matplotlib.pyplot as plt
import array
import math
import numpy as np
```

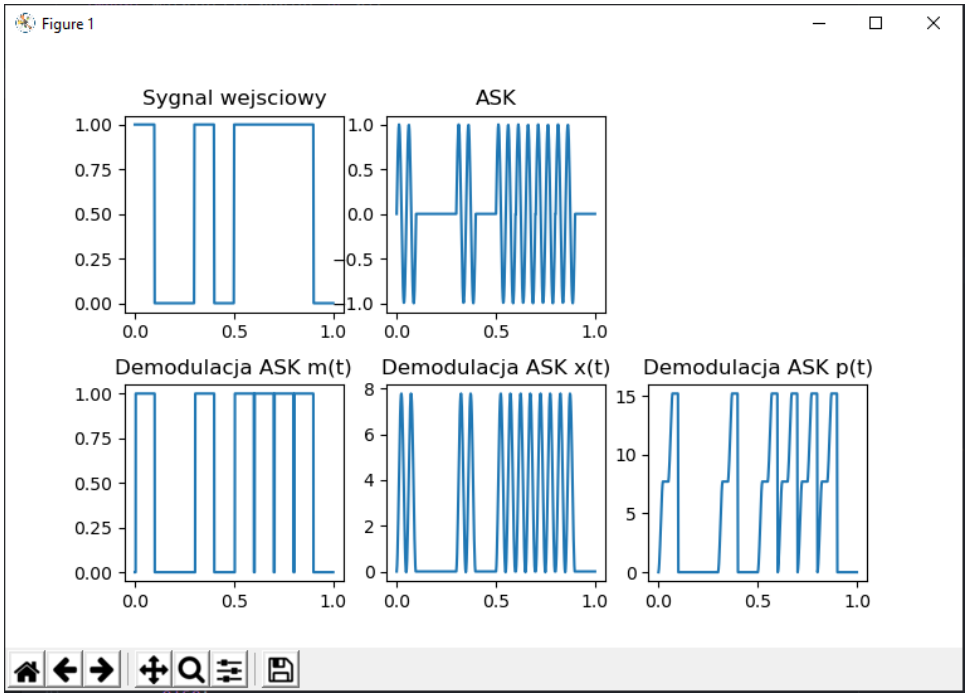
```
import matplotlib.pyplot as plt
import array

def Binary(string,switch):
    if (switch == 0):
        bin = ''.join(format(i, 'b') for i in bytearray(string, encoding ='utf-8'))
        bin = list(map(int, bin))
        #print('variant=littleEndian\n','conversion of ', string,' to binary is equal to [ ', bin,' ]')
        return bin
    else:
        rev=string[::-1]
        bin2 = ''.join(format(i, 'b') for i in bytearray(rev, encoding ='utf-8'))
        bin2 = list(map(int, bin2))
        #print('variant=BigEndian\n','conversion of ', string,' to binary is equal to [ ',''.join(bin2),'
]')
        return bin2

def MT(LIMIT):
    ycords=[] ; xcords = []
    for i in range (LIMIT):
        if(mt[i]==0):
            for x in np.linspace(1/10,2/10):
                ycords.append(0)
        else:
            for x in np.linspace(1/10,2/10):
                ycords.append(1)
    xcords=np.linspace(0,1,len(ycords))
    plt.subplot(231)
    plt.title('sygnal wejsciowy')
    plt.plot(xcords,ycords)
```

```
def ZPT(mt, t, f, tb, koniec):
    ycords=[] ; x = []
    for i in range(len(t)):
        if(mt[i]==0):
          if i + 1 < len(t):
            cords_sin = np.linspace(t[i], t[i+1])
            for m in cords_sin:
              tb=t[i]+5
              ampl = 2*np.pi/ ((t[i+1]-t[i])*0.5 )
              ycords.append(np.sin(ampl*(m  - 1/len(t))+3.14))
          else:
            cords_sin = np.linspace(t[i], koniec)
            for m in cords_sin:
              ycords.append(np.sin(ampl*(m  - 1/len(t))+3.14))
        else:
          if i + 1 < len(t):
            cords_sin = np.linspace(t[i], t[i+1])
            for m in cords_sin:
              tb=t[i]+5
              ampl = 2*np.pi/ ((t[i+1]-t[i])*0.5 )
              ycords.append(np.sin(ampl*(m  - 1/len(t))))
          else:
            cords_sin = np.linspace(t[i], koniec)
            for m in cords_sin:
              ycords.append(np.sin(ampl*(m  - 1/len(t))))
        ycords.append(0)
    plt.subplot(232)
    plt.title('PSK')
```

```
    x=np.linspace(0, 1, len(ycords))
    plt.plot(x, ycords)
    return(ycords)


def demodulator1(Modulacja,h):
    Demo=[] ; Final=[] ;  Demov2 = []
    sum=0 ; sumabs = 0

    for i in range (len(Modulacja)):
        if (Modulacja[i] == 0.0000000):
            sum = 0
        sum=sum+Modulacja[i]
        Demo.append(sum)

    xcords = np.linspace(0,1,len(Demo))
    plt.subplot(235)
    plt.title('Demodulacja PSK x(t)')
    plt.plot(xcords,Demo)

    for j in range(len(mt)):
        if(mt[j]==0):
            for i in range (len(Modulacja)):
                if(Modulacja[i]<0):
                    sumabs=sumabs+Modulacja[i]
                    Demov2.append(sumabs)
        else:
            for i in range (len(Modulacja)):
                if(Modulacja[i]>0):
```

```
                sumabs=sumabs+Modulacja[i]
                Demov2.append(sumabs)
        sumabs=0

    xcords = np.linspace(0,0.1,len(Demov2))
    plt.subplot(236)
    plt.title('Demodulacja PSK p(t)')
    plt.plot(xcords,Demov2)

    for i in range (len(Demov2)):
        roundmt=round(Demov2[i],0)
        if(roundmt<=0):
            for x in np.linspace(1/10,2/10):
                Final.append(0)
        else:
            for x in np.linspace(1/10,2/10):
                Final.append(1)

    xcords=np.linspace(0,1,len(Final))
    plt.subplot(234)
    plt.title('Demodulacja PSK m(t)')
    plt.plot(xcords,Final)

plt.figure()
mt=Binary('AB',0)
mt=[0,1,1,0,1,0]
x=np.linspace(0, 1, len(mt), endpoint=False)
tb=0.1 ;N=10;f=N*(tb**(-1));f0=(N+1)/tb;f1=(N+2)/tb;A=1;A1=0;A2=1
LIMIT=10
```

```
MT(len(mt))
ycords=ZPT(mt, x, f, tb, 1)


for i in range(len(ycords)):
    if ( ycords[i] == 0):
        print('zero')
#ycords=ZAT(len(mt))
demodulator1(ycords,0.2)
plt.show()
```



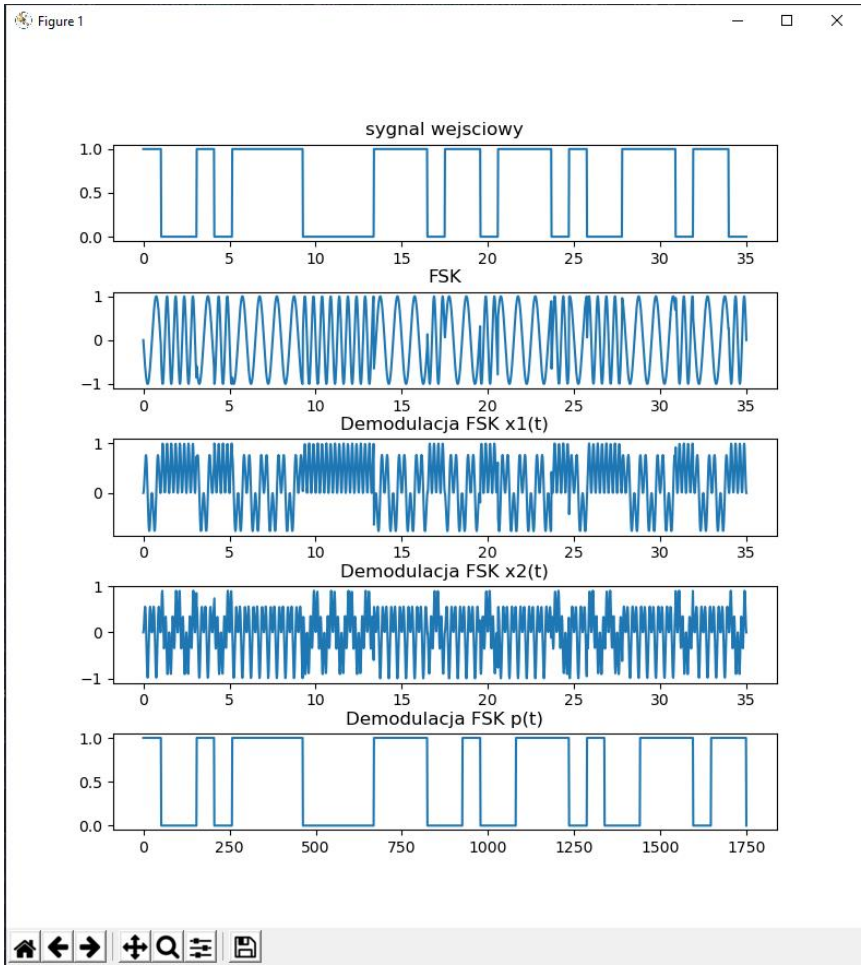


Figure 1
sygnal wejsciowy
FSK
Demodulacja FSK x1(t)
Demodulacja FSK x2(t)
Demodulacja FSK p(t)

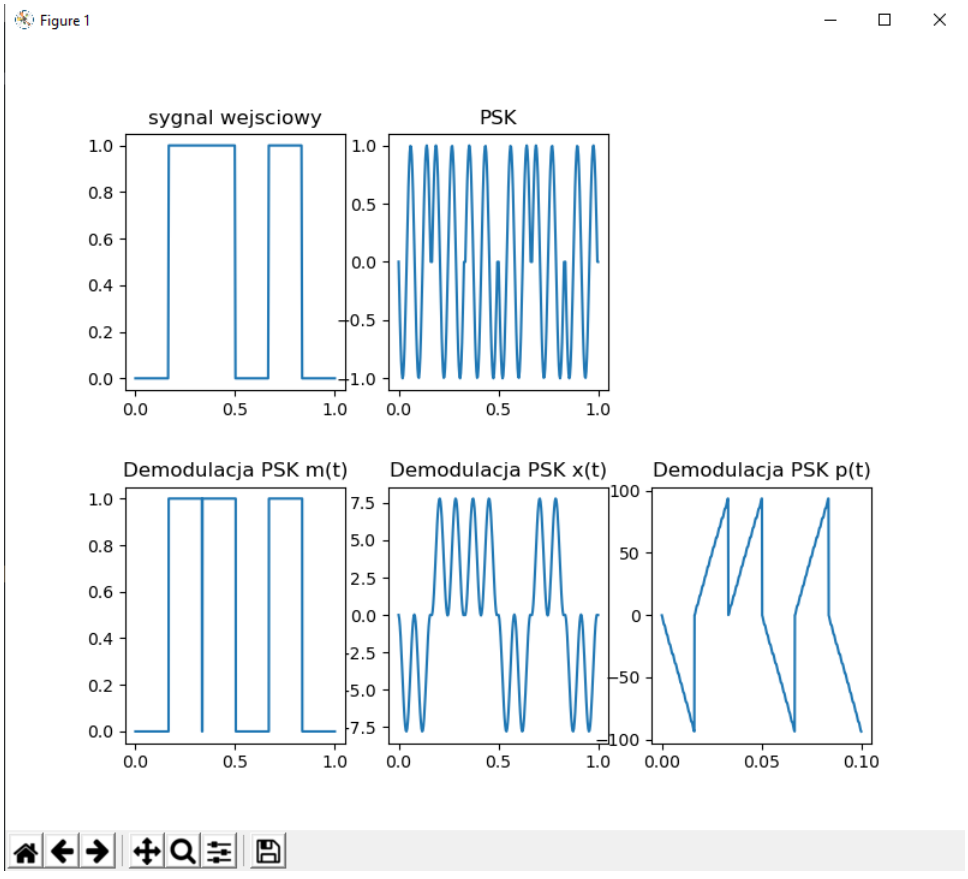


Figure 1
sygnal wejsciowy
PSK
Demodulacja PSK m(t)
Demodulacja PSK x(t)
Demodulacja PSK p(t)