



Dr. Sadık YILDIZ

sadikyildiz06@gmail.com

**Gazi Üniversitesi
Elektrik Elektronik Mühendisliği
(Teknoloji Fakültesi)**

2025– ANKARA

Sayısal Taşıyıcı Modülasyonu

- *Sayısal modülasyon: Dijital veriye göre taşıyıcı sinyalin özelliklerinin değiştirilmesi*
- Taşıyıcı: Genellikle sinüzoidal bir sinyal
- Modülasyon Türleri:
 - ASK (Amplitude Shift Keying)
 - FSK (Frequency Shift Keying)
 - PSK (Phase Shift Keying)
 - QPSK (Quadrature PSK)

Sayısal Taşıyıcı Modülasyonu

3. PSK– Faz Kaydırmalı Anahtarlama (Phase Shift Keying)

- Phase Shift Keying (Faz Kaydırmalı Anahtarlama), taşıyıcı sinyalin fazını değiştirerek sayısal veri iletimini sağlayan bir modülasyon tekniğidir.
- Binary PSK (BPSK), en basit PSK türüdür.
- Sadece iki farklı faz değeri kullanır:
 - Bit 1 için: taşıyıcı sinyal fazı 0
 - Bit 0 için: taşıyıcı sinyal fazı π (yani 180° faz kayması)

➤ PSK Modülasyonu

- Taşıyıcı Sinyali şu şekilde ifade edilebilir.

$$c(t) = A \cos(2\pi f_c t) \text{ veya } c(t) = A \sin(2\pi f_c t)$$

- BPSK modüle edilmiş sinyal

$$s(t) = \begin{cases} A \cos(2\pi f_c t) & , \text{ eğer } b(t) = 1 \\ A \cos(2\pi f_c t + \pi) = -A \cos(2\pi f_c t) & , \text{ eğer } b(t) = 0 \end{cases}$$

- Alternatif olarak genel ifade şu şekilde de ifade edilebilir.

$$s(t) = A \cos(2\pi f_c t + \pi b(t))$$

- A: taşıyıcı genliği
- f_c : taşıyıcı frekansı
- $b(t) \in \{0,1\}$: iletilmek istenen bit
- Faz Farkı: π (180°)

Sayısal Taşıyıcı Modülasyonu

3. PSK– Faz Kaydırmalı Anahtarlama (Phase Shift Keying)

➤ PSK Demodülasyonu

- Demodülasyon işleminde gönderilen sinyal, **eş fazlı taşıyıcı sinyal** ile çarpılarak ve daha sonra **entegral alınarak** yorumlanır.
- Alınan sinyal şu şekilde ifade edilir.

$$r(t) = s(t) + n(t)$$

- $n(t)$: Kanal Gürültüsü (varsa).
- Demodülasyon işlemi ise şu şekilde ifade edilir.

$$z = \int_0^T r(t) \cos(2\pi f_c t) dt$$

- Karar kuralıda şu şekilde ifade edilir.

$$\hat{b}(t) = \begin{cases} 1 & , \text{ eğer } z > 0 \\ 0 & , \text{ eğer } z < 0 \end{cases}$$

- A: taşıyıcı genliği
- f_c : taşıyıcı frekansı
- $b(t) \in \{0,1\}$: iletilmek istenen bit
- *Faz Farkı*: π (180°)

Sayısal Taşıyıcı Modülasyonu

3. PSK– Faz Kaydırmalı Anahtarlama (Phase Shift Keying)

```

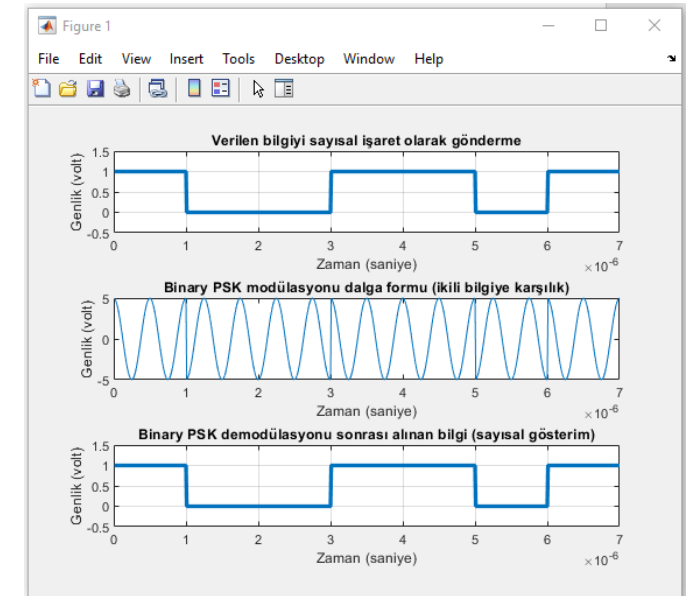
1 % Binary PSK modülasyonu ve demodülasyonu için MATLAB kodu
2 clc;
3 clear all;
4 close all;
5
6 x = [1 0 0 1 1 0 1]; % İkili bilgi dizisi
7 bp = 0.000001; % Bit süresi (bit period)
8 disp('Verici tarafındaki ikili bilgi :');
9 disp(x);
10
11 % İkili bilgiyi sayısal işaret olarak gösterimi
12 bit = [];
13 for n = 1:length(x)
14     if x(n) == 1
15         se = ones(1,100); % 1 için yüksek genlik
16     else
17         se = zeros(1,100); % 0 için sıfır genlik
18     end
19     bit = [bit se];
20 end
21
22 t1 = bp/100 : bp/100 : 100*length(x)*(bp/100);
23 subplot(3,1,1);
24 plot(t1, bit, 'lineWidth', 2.5); grid on;
25 axis([0 bp*length(x) -0.5 1.5]);
26 ylabel('Genlik (volt)');
27 xlabel('Zaman (saniye)');
28 title('Verilen bilgiyi sayısal işaret olarak gönderme');
29
30 %Binary PSK Modülasyonu
31 A = 5; % Taşıyıcı genliği
32 br = 1 / bp; % Bit hızı
33 f = br * 2; % Taşıyıcı frekansı
34 t2 = bp/99 : bp/99 : bp; % Zaman eksenini (bir bit süresi)
35 ss = length(t2);
36 m = [];
37
38 for i = 1:length(x)
39     if x(i) == 1
40         y = A * cos(2 * pi * f * t2); % Bit 1 için sinyal
41     else
42         y = A * cos(2 * pi * f * t2 + pi); % Bit 0 için fazı n kadar kaydır
43     end
44     m = [m y]; % Sinyali birleştir
45 end

```

```

46
47 t3 = bp/99 : bp/99 : bp*length(x);
48 subplot(3,1,2);
49 plot(t3, m);
50 xlabel('Zaman (saniye)');
51 ylabel('Genlik (volt)');
52 title('Binary PSK modülasyonu dalga formu (ikili bilgiye karşılık)');
53
54 %Binary PSK Demodülasyonu
55 mn = [];
56
57 for n = ss:ss:length(m)
58     t = bp/99 : bp/99 : bp;
59     y = cos(2 * pi * f * t); % Yerel taşıyıcı sinyal
60     mm = y .* m((n - (ss - 1)) : n); % Giriş sinyaliyle çarp
61     t4 = bp/99 : bp/99 : bp;
62     z = trapz(t4, mm); % İntegral alma (enerji ölçümü)
63     zz = round((2 * z / bp)); % Normalize et ve yuvarla
64
65     if zz > 0
66         a = 1; % Karar: pozitifse 1 alınmıştır
67     else
68         a = 0; % Karar: negatifse 0 alınmıştır
69     end
70
71     mn = [mn a]; % Alınan bit dizisine ekle
72 end
73
74 disp('Alıcı tarafındaki ikili bilgi :');
75 disp(mn);
76
77 % Demodülasyon sonrası elde edilen ikili bilgiyi sayısal işaret olarak göster
78 bit = [];
79
80 for n = 1:length(mn)
81     if mn(n) == 1
82         se = ones(1,100);
83     else
84         se = zeros(1,100);
85     end
86     bit = [bit se];
87 end
88
89 t4 = bp/100 : bp/100 : 100*length(mn)*(bp/100);
90 subplot(3,1,3);
91 plot(t4, bit, 'LineWidth', 2.5); grid on;
92 axis([0 bp*length(mn) -0.5 1.5]);
93 ylabel('Genlik (volt)');
94 xlabel('Zaman (saniye)');
95 title('Binary PSK demodülasyonu sonrası alınan bilgi (sayısal gösterim)');

```



Sayısal Taşıyıcı Modülasyonu

3. PSK– Faz Kaydırmalı Anahtarlama (Phase Shift Keying)

```

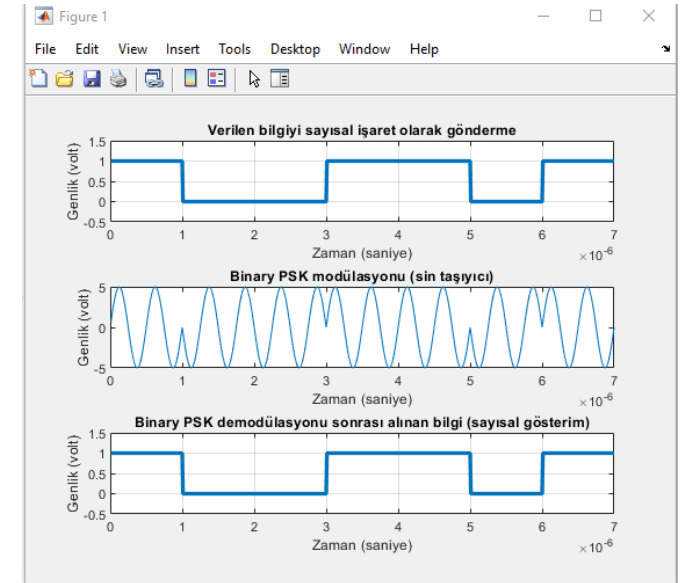
1 % Binary PSK modülasyonu ve demodülasyonu (sin taşıyıcı ile)
2 clc;
3 clear all;
4 close all;
5
6 x = [1 0 0 1 1 0 1]; % İkili bilgi dizisi
7 bp = 0.000001; % Bit süresi
8 disp('Verici tarafındaki ikili bilgi :');
9 disp(x);
10
11 % Sayısal işaret olarak gösterimi
12 bit = [];
13 for n = 1:length(x)
14     if x(n) == 1
15         se = ones(1,100); % 1 için yüksek genlik
16     else
17         se = zeros(1,100); % 0 için sıfır genlik
18     end
19     bit = [bit se];
20 end
21
22 t1 = bp/100 : bp/100 : 100*length(x)*(bp/100);
23 subplot(3,1,1);
24 plot(t1, bit, 'LineWidth', 2.5); grid on;
25 axis([0 bp*length(x) -0.5 1.5]);
26 ylabel('Genlik (volt)');
27 xlabel('Zaman (saniye)');
28 title('Verilen bilgiyi sayısal işaret olarak gönderme');
29
30 % Binary PSK Modülasyonu (sin taşıyıcı ile)
31 A = 5; % Taşıyıcı genliği
32 br = 1 / bp; % Bit hızı
33 f = br * 2; % Taşıyıcı frekansı
34 t2 = bp/99 : bp/99 : bp; % Zaman eksen (bir bit süresi)
35 ss = length(t2);
36 m = [];
37
38 for i = 1:length(x)
39     if x(i) == 1
40         y = A * sin(2 * pi * f * t2); % Bit 1 için sinyal
41     else
42         y = A * sin(2 * pi * f * t2 + pi); % Bit 0 için fazı n kadar kaydır
43     end
44     m = [m y]; % Sinyali birleştir
45 end

```

```

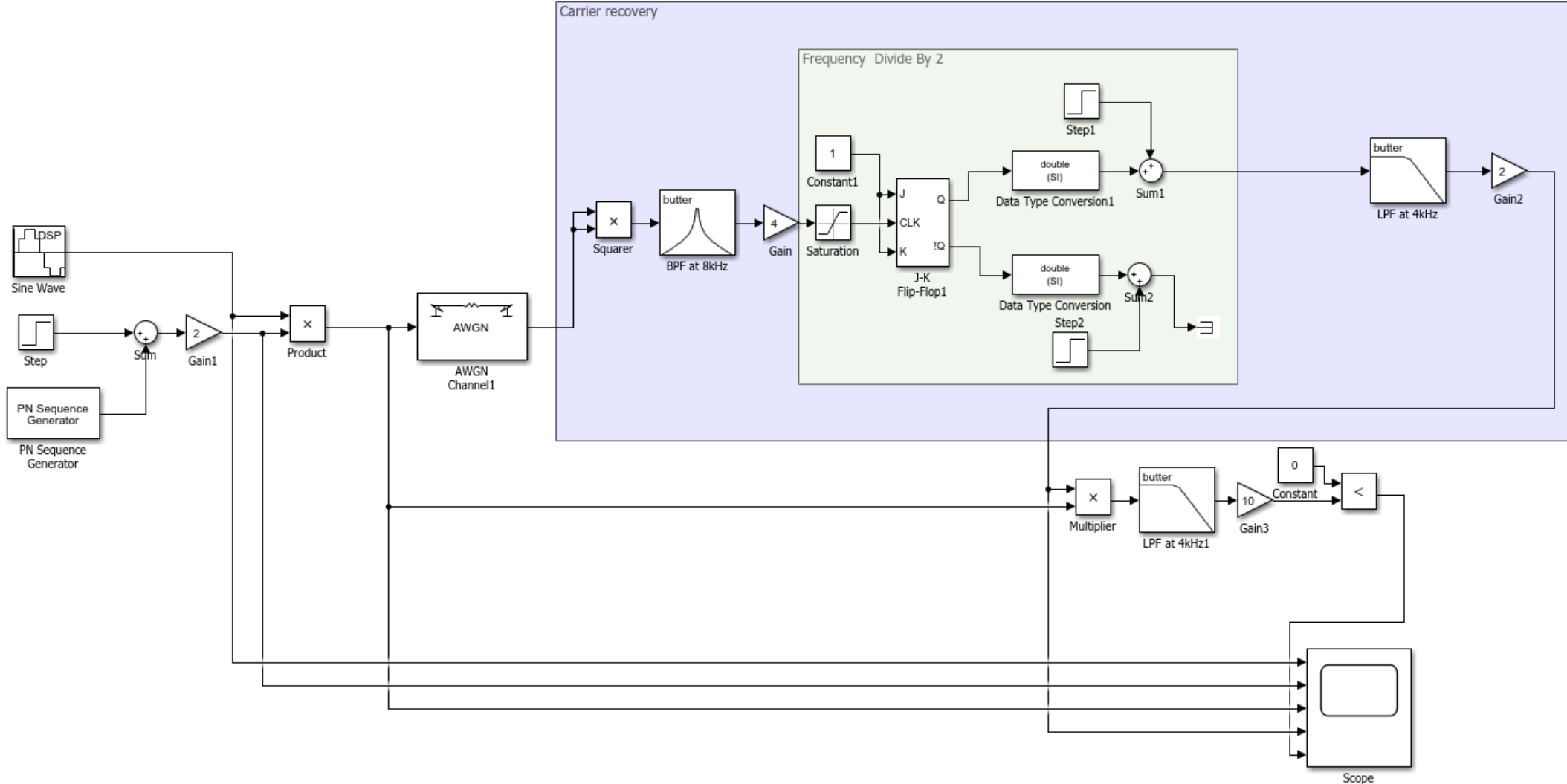
46
47 t3 = bp/99 : bp/99 : bp*length(x);
48 subplot(3,1,2);
49 plot(t3, m);
50 xlabel('Zaman (saniye)');
51 ylabel('Genlik (volt)');
52 title('Binary PSK modülasyonu (sin taşıyıcı)');
53
54 % Binary PSK Demodülasyonu (sin taşıyıcı ile)
55 mn = [];
56
57 for n = ss:ss:length(m)
58     t = bp/99 : bp/99 : bp;
59     y = sin(2 * pi * f * t); % Yerel taşıyıcı sinyal (sin)
60     mm = y .* m((n - (ss - 1)) : n); % Giriş sinyaliyle çarp
61     t4 = bp/99 : bp/99 : bp;
62     z = trapez(t4, mm); % İntegral alma (enerji ölçümü)
63     zz = round((2 * z / bp)); % Normalize et ve yuvarla
64
65     if zz > 0
66         a = 1; % Karar: pozitifse 1 alınmıştır
67     else
68         a = 0; % Karar: negatifse 0 alınmıştır
69     end
70
71     mn = [mn a]; % Alınan bit dizisine ekle
72 end
73
74 disp('Alıcı tarafındaki ikili bilgi :');
75 disp(mn);
76
77 % Demodülasyon sonrası elde edilen ikili bilgiyi sayısal işaret olarak göster
78 bit = [];
79
80 for n = 1:length(mn)
81     if mn(n) == 1
82         se = ones(1,100);
83     else
84         se = zeros(1,100);
85     end
86     bit = [bit se];
87 end
88
89 t4 = bp/100 : bp/100 : 100*length(mn)*(bp/100);
90 subplot(3,1,3);
91 plot(t4, bit, 'LineWidth', 2.5); grid on;
92 axis([0 bp*length(mn) -0.5 1.5]);
93 ylabel('Genlik (volt)');
94 xlabel('Zaman (saniye)');
95 title('Binary PSK demodülasyonu sonrası alınan bilgi (sayısal gösterim)');

```



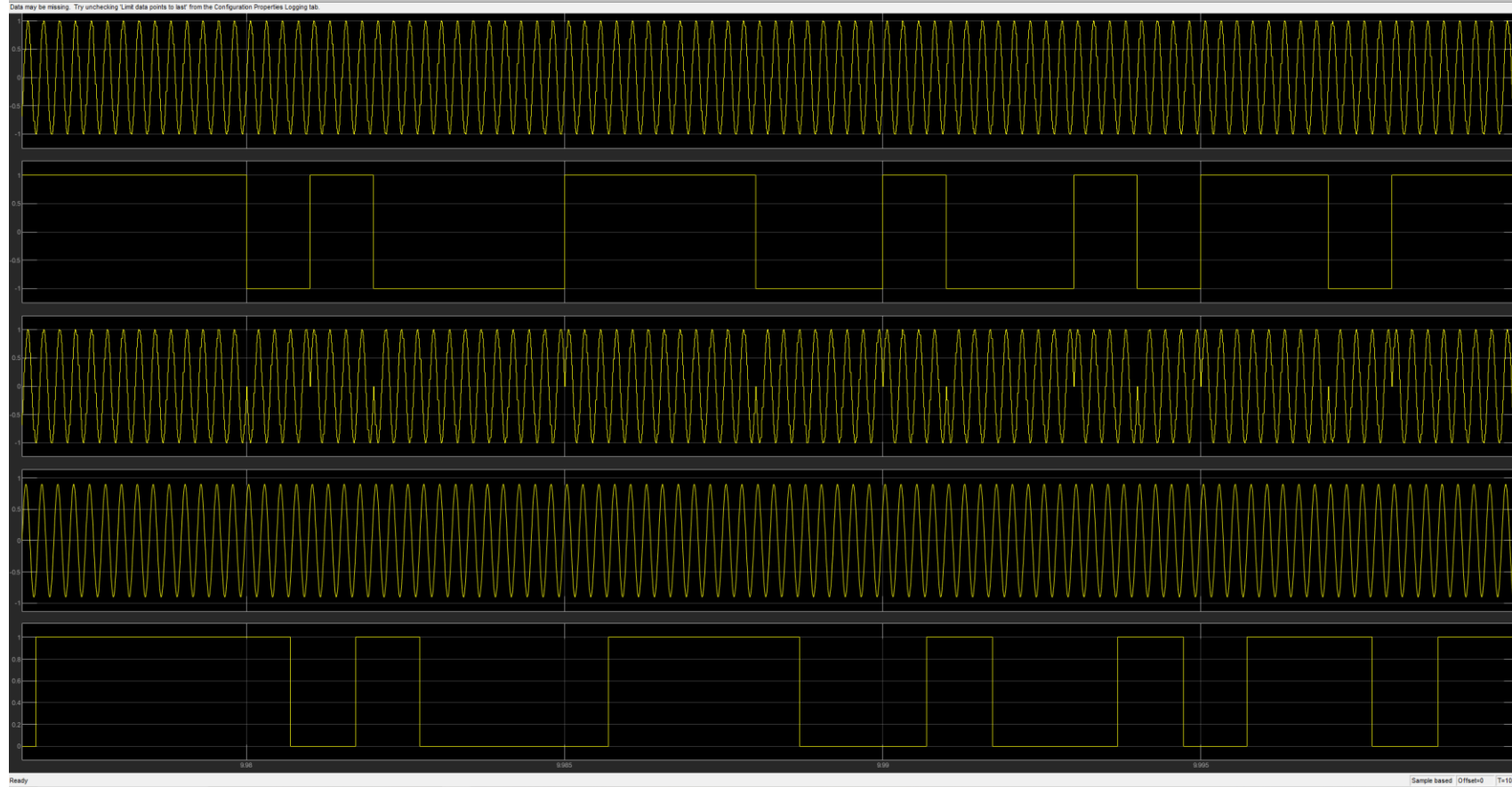
Sayısal Taşıyıcı Modülasyonu

3. PSK– Faz Kaydırmalı Anahtarlama (Phase Shift Keying)



Sayısal Taşıyıcı Modülasyonu

3. PSK– Faz Kaydırmalı Anahtarlama (Phase Shift Keying)



Sayısal Taşıyıcı Modülasyonu

3. PSK– Faz Kaydırmalı Anahtarlama (Phase Shift Keying)

Block Parameters: Sine Wave

Sine Wave (mask) (link)

Output samples of a sinusoid. To generate more than one sinusoid simultaneously, enter a vector of values for the Amplitude, Frequency, and Phase offset parameters.

Main Data Types

Amplitude: 1

Frequency (Hz): 4000

Phase offset (rad): 0

Sample mode: Discrete

Output complexity: Real

Computation method: Trigonometric fcn

Sample time: 1/100000

Samples per frame: 1

Resetting states when re-enabled: Restart at time zero

OK Cancel Help Apply

Block Parameters: Step

Step

Output a step.

Main Signal Attributes

Step time: 1/300

Initial value: -.5

Final value: -.5

Sample time: 0

☒ Interpret vector parameters as 1-D

☒ Enable zero-crossing detection

OK Cancel Help Apply

Block Parameters: PN Sequence Generator

PN Sequence Generator (mask) (link)

Generate a pseudorandom noise (PN) sequence using a linear feedback shift register (LFSR). The LFSR is implemented using a simple shift register generator (SSRG, or Fibonacci) configuration.

The 'Generator polynomial' parameter values specify the shift register connections. Enter these values as either a polynomial string, a binary vector, or a descending-ordered polynomial. For the binary vector representation, the first and last elements of the vector must be 1. For the descending-ordered polynomial representation, the last element of the vector must be 0.

The 'Output mask source' may be from a dialog parameter or an input port. The 'Output mask vector' is a binary vector corresponding to the shift register states that are to be XORed to produce the output sequence values. Alternatively, you may enter an integer 'scalar shift value' to produce an equivalent advance or delay in the output sequence.

For variable-size output signals, the current output size is either specified from the 'oSiz' input or inherited from the 'Ref' input.

Parameters

Generator polynomial: [1 0 0 0 1 1]

Initial states: [0 0 0 0 1]

Output mask source: Dialog parameter

Output mask vector (or scalar shift value): -1

☐ Output variable-size signals

Sample time: 1/1000

Samples per frame: 1

☐ Reset on nonzero input

☐ Enable bit-packed outputs

Output data type: double

OK Cancel Help Apply

Block Parameters: AWGN Channel1

AWGN Channel (mask) (link)

Add white Gaussian noise to the input signal. The input signal can be real or complex. This block supports multichannel processing.

When using either of the variance modes with complex inputs, the variance values are equally divided among the real and imaginary components of the input signal.

Parameters

Initial seed: 67

Mode: Variance from mask

Variance: 0.2

OK Cancel Help Apply

Sayısal Taşıyıcı Modülasyonu

3. PSK– Faz Kaydırmalı Anahtarlama (Phase Shift Keying)

Block Parameters: BPF at 8kHz

Analog Filter Design (mask) (link)

Design one of several standard analog filters, implemented in state-space form.

Parameters

Design method: **Butterworth**

Filter type: **Bandpass**

Filter order: **4**

Lower passband edge frequency (rad/s): **$7900 \cdot 2 \cdot \pi$**

Upper passband edge frequency (rad/s): **$8100 \cdot 2 \cdot \pi$**

OK Cancel Help Apply

Block Parameters: Saturation

Saturation

Limit input signal to the upper and lower saturation values.

Main Signal Attributes

Upper limit: **1**

Lower limit: **0**

☒ Treat as gain when linearizing

☒ Enable zero-crossing detection

OK Cancel Help Apply

Block Parameters: Data Type Conversion1

Data Type Conversion

Convert the input to the data type and scaling of the output.

The conversion has two possible goals. One goal is to have the Real World Values of the input and the output be equal. The other goal is to have the Stored Integer Values of the input and the output be equal. Overflows and quantization errors can prevent the goal from being fully achieved.

Parameters

Output minimum: **0** Output maximum: **0**

Output data type: **double**

☐ Lock output data type setting against changes by the fixed-point tools

Input and output to have equal: **Stored Integer (SI)**

Integer rounding mode: **Nearest**

☐ Saturate on integer overflow

OK Cancel Help Apply

Block Parameters: J-K Flip-Flop1

JKFlipFlop (mask) (link)

J-K Flip-Flop Block (Negative Edge Triggered)

Parameters

Initial condition (state of Q): **0**

OK Cancel Help Apply

Block Parameters: LPF at 4kHz

Analog Filter Design (mask) (link)

Design one of several standard analog filters, implemented in state-space form.

Parameters

Design method: **Butterworth**

Filter type: **Lowpass**

Filter order: **4**

Passband edge frequency (rad/s): **$4000 \cdot 2 \cdot \pi$**

OK Cancel Help Apply