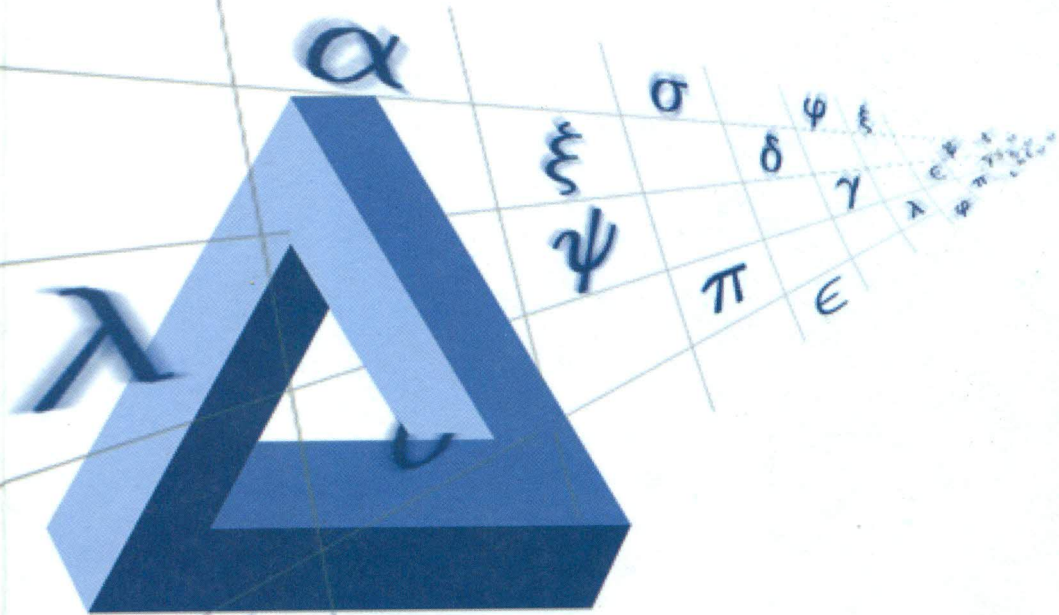




ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ. Ә. Өжікенов

ЖҮЙЕЛЕРДІ МОДЕЛЬДЕУДІҢ БАҒДАРЛАМАЛЫҚ ҚҰРАЛДАРЫ (MATLAB/SIMULINK)



Алматы, 2012

Қ. Ә. ӨЖІКЕНОВ

**ЖҮЙЕЛЕРДІ МОДЕЛЬДЕУДІҢ
БАҒДАРЛАМАЛЫҚ ҚҰРАЛДАРЫ
(MATLAB/SIMULINK)**

Оқулық

ӘОЖ 004.4(075.8)
КБЖ 32.973я73
Ө-23

*Қазақстан Республикасы Білім және ғылым министрлігінің
«Оқулық» республикалық ғылыми-практикалық орталығы ұсынған*

Пікір берушілер:

техника ғылымдарының докторы, профессор **Н. Т. Исембергенов**;
техника ғылымдарының докторы, профессор **Е. Т. Үтепбергенов**;
техника ғылымдарының докторы **Ш. Ә. Жомартова**.

Қ. Ә. Өжікенов

Ө-23 Жүйелерді модельдеудің бағдарламалық құралдары (MATLAB/Simulink): Оқулық. – Алматы: 2012. – 304 бет.

ISBN 978-601-217-359-8

«Жүйелерді модельдеудің бағдарламалық құралдары (MATLAB/Simulink)» оқулығы жоғары оқу орындарындағы техникалық бағытта білім алып жатқан студенттерге арналған. Мұнда MATLAB/Simulink бағдарламалық қолданбалы пакетін қолдана отырып, осы ортада жүйелерге модель құрудың және оларға есептеулер жүргізудің әдістерінен теориялық мағлұматтар берілген.

Автор ұсынылып отырған бұл оқулықтың MATLAB/Simulink бағдарламалық қолданбалы пакетін өз бетінше игергісі келетін оқырмандарға да пайдасы тиер деген сенімде.

ӘОЖ 004.4(075.8)
КБЖ 32.973я73

© Өжікенов Қ.Ә., 2012
© ҚР Жоғары оқу орындарының
қауымдастығы, 2012

ISBN

КІРІСПЕ

Компьютерлік технологияның қарқынды дамуына байланысты техникалық жүйелерді жобалаудағы күрделі мәселелерді шешу жолдарына да үлкен өзгерістер енуде.

MATLAB, DesignLab, WorkBench, OrCad және т.б. қолданбалы пакеттері әр түрлі техникалық жүйелерді зерттеудегі және жобалаудағы инженерлік қолданыстарды кеңейтіп, сапалы өзгерістерге қол жеткізуге мүмкіндік береді. Қолданыс аумағының шексіз болуынан қолданбалы пакеттердің ішіндегі бірден-бір болашағы зор болып MATLAB пакеті саналатыны сөзсіз. Сондықтан да, соңғы жылдары жоғары және кәсіптік орта оқу орындарының оқу жоспарларында жүйелерді модельдеудің мәселесіне көбірек уақыт бөлетін болды. Компьютерлік технологияның соңғы жетістіктерін қолдана отырып, студенттердің техникалық пәндерді оқуы кезінде де, инженердің техникалық жүйелерді жобалауы кезінде де жүйелердегі өтетін үдерістер физикасын және осы үдерістерге зерттеулер жүргізу үшін қолданылатын математикалық тәсілдерді тереңдетіп білулері талап етіледі. Үдерістер физикасын жеткілікті деңгейде білу - нақты жүйеге сай модель құруға және модельдеу нәтижелерінің нақты болуына кепіл бола алады.

MATLAB пакеті жайында төл тілімізде алғаш рет жарық көріп отырған бұл оқулық заман талабына сай жазылып, жаратылыстану мен техникалық салаларда мемлекеттік тілімізде білім алу барысында жүйелерді модельдеуге бағытталған пәндерді оқытуда қолдану үшін шығарылып отыр. Қазақ тілінде жазылған бұл оқулық MATLAB пакеті жайындағы бүкіл мағлұматтарды бірден толық қамтып береді деу, әрине, артықтау болар, дегенмен де автор осы пакетті қолданудың әдіс-тәсілдерін үйренудегі ең басты да, негізгі деген мағлұматтарды қамтуға тырысты.

Оқулықта теориялық мағлұматтарды берумен қатар, оларға көптеген нақты мысалдарды қолдану қарастырылған.

Сонымен қатар, мемлекеттік тілімізге әлі де техникалық терминдердің толық аударылып және мемлекеттік стандартпен бекітіліп еңбегендіктен, кітаптың соңында қолданылған терминдердің аудармалары келтірілген.

Автор қазақ тілінде жарық көріп отырған бұл оқулықты, төл тілімізде дәріс алып жатқан студенттеріміз үшін, MATLAB ортасында бағдарламалар құру және осы ортада жүйелерді модельдеудің қыры мен сырын меңгеру жолында көмекші құрал бола алады деген сенімде.

1 MATLAB ОРТАСЫ. КОМАНДАЛЫҚ РЕЖИМДЕГІ ЖҰМЫС

1.1 MATLAB ортасы

Жұмыс істеу кезінде инженердің де, ғалымның да, тіпті қарапайым студенттің де математикалық есептеулерге жүгінуіне тура келетін кездері болады. Ал мұндай есептеулер бір жағдайларда қарапайымдау болса, енді бір жағдайларда күрделі де болып келеді. Ал күрделі есептерді шығаруда MATLAB (*MATrix LABoratory*) сияқты математикалық пакеттерді қолдануға болады.

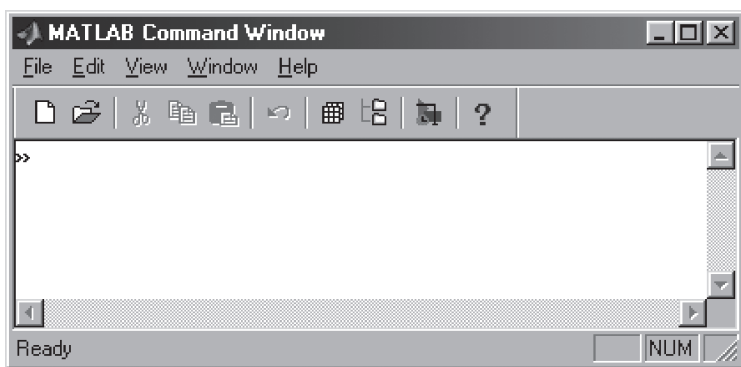
MATLAB программалау тілі ретінде Нью-Мексико университетінің компьютер ғылымдары факультетінің деканы болған Клив Моулермен (ағылш. *Cleve Moler*) 1970-ші жылдардың соңында жасалды. Оны жасаудағы мақсат – факультет студенттеріне ФОРТРАН-ды оқытпай-ақ, *Linpack* және *EISPACK* бағдарламалық кітапханаларын қолданудың мүмкіндігін жасау болды. Көп ұзамай-ақ бұл жаңа тіл басқа университеттерге де тарап кетеді де, қолданбалы математика саласындағы ғалымдардың да аса қызығушылығына тап болады. 1983 жылы инженер Джон Литтл (ағылш. *John N. (Jack) Little*) Клив Моулермен Стэнфорд университетіндегі кездесуінде осы тілмен таныс болып, бұл тілдің болашағының зор боларын білген ол Клив Моулермен және Стив Бангертпен (ағылш. *Steve Bangert*) бірігеді. Біріккен күшпен олар MATLAB-ты C тілінде қайта жазып, оны ары қарай дамыту мақсатында 1984 жылы *The MathWorks* компаниясын құрады. Бастапқы уақытта MATLAB тек басқару жүйелерін жобалауда ғана қолданылады да, артынан басқа да ғылыми және инженерлік қолданыстарда болады.

MATLAB негізінде матрицалық есептеулерді жүргізу үшін арналған сұхбаттық орта болып жасалған болатын (*MATrix LABoratory*). Уақыт өте келе оның графикалық жүйесі дамытылып, арнайы кластағы есептеулерді жүргізу үшін командалар кітапханаларымен (*Toolboxes* және т.б.) және Maple-дің компьютерлік алгебра ортасымен толықтырылады. Қазіргі таңда

MATLAB ғылыми зерттеулер жүргізудегі ең мықты құралдың бірі болып отыр.

MATLAB-тағы барлық жұмыс бағдарламаны *matlab.exe* арқылы қосқанда пайда болатын командалық терезе (*Command Window*) арқылы іске асырылады.

1.1-суретте MATLAB жүйесінің командалық терезесі көрсетілген. Онда мәзір жолы, құрал-саймандар жақтауы, жұмыс аймағы мен күй жолы бар. Жұмыс аймағында командалар енгізіледі және есептеу нәтижелері шығарылады. Командаларды енгізу «>>» түріндегі командалық жолға шыққан шақыру белгісі арқылы іске асырылады. Терезенің төменгі жағында орналасқан күй жолына жүйенің деректері шығарылып тұрады.



1.1-сурет. MATLAB жүйесінің командалық терезесі

1.2 Командалық режимдегі жұмыс

MATLAB-та есептеуді бағдарламалық режимде немесе командалық режимде жұмыс істеуге болады. Бағдарламалық режимде MATLAB тілінде жазылған арнайы бағдарлама шақырылады. Бұл бағдарлама арқылы алғашқы деректер енгізіліп, есептеулер жүргізіліп, соңынан олардың нәтижелерін шығару іске асырылады.

Командалық режимде командалар мен есептелінетін өрнектер командалық терезедегі енгізу жолдары арқылы беріледі. Есептеу нәтижелері командалық терезеге шығарылады.

Мысалы, $4+5$ өрнегін командалық терезеге енгізіп, пернетақтадағы *Enter*-ді басқанда компьютер экранында келесідей нәтижені көруге болады:

```
» 4+5;  
ans =  
  9  
»
```

Өрнектің есептелген мәні *ans* (*answer*) арнайы айнымалысына меншіктеледі де, жеке жолға шығарылады. *Ans* айнымалысында басқа айнымалыға меншіктелмеген соңғы есептелген өрнектің мәні сақталады. Ал, бұл мәнді әрі қарай жүргізілетін есептеулерге де қолдануға болады:

```
» 4+5;  
ans =  
  9  
» ans/3;  
» ans  
ans =  
  3  
»
```

Егер командалық жолға енгізілген өрнектің соңына «;» (нүктелі үтір) белгісін қойса, онда есептеу нәтижесі шығарылмайды. Мысалы (бағанның оң жағында сол командалар «;» белгісізсіз енгізілген):

<pre>» 4+5; » ans/3; » ans ans = 3 »</pre>	<pre>» 4+5 ans = 9 » ans/3 ans = 3 » ans ans = 3 »</pre>
--	--

Сонымен қатар «;» белгісін командалық жолда берілетін өрнектерді ажырату үшін де қолдануға болады:


```

» g=8; h=5; k=g>h
» g
g =
    8
» h
h =
    5
» k
k =
    1
»

```

Командалық жолға енгізілген командалар мен өрнектердің бәрі де сақталады. Оларды қайтара енгізу командаларды тік және кері бағытта көруге мүмкіндік беретін \uparrow және \downarrow батырмалары арқылы іске асырылады.

Командалық терезені тазарту үшін *clc* командасы қолданылады.

Егер енгізілетін өрнек бір жолдың бойына симайтын болса, онда оларды келесі жолға көшіру үшін «...» көпнүктелік белгісі қойылады (үш немесе одан да көп):

```

» 3+...
5-7+9/...
3
ans =
    4
»

```

MATLAB-та жұмыс сеансы *сессия* деп аталады. Сессия енгізу-шығару жолдарынан және ескертулер мен қателіктер жайындағы деректерді шығаратын жолдардан тұрады. Қателіктер жайындағы деректер алдын-ала үш сұрақ белгісімен беріледі де, есептеу тоқтатылады:

```

» 1h=5
??? 1
|
Missing operator, comma, or semi-colon.
»

```

Ескертулер алдын-ала *Warning* сөзімен беріледі. Ескерту есептеуді тоқтатпайды, ол тек қана қолданушыға табылған қате-нің кейінгі есептеу нәтижелеріне кері әсер етуінің мүмкіндігін ғана ескертеді:

```
» 1/0
Warning: Divide by zero.
ans =
    Inf
»
```

1.3 Математикалық өрнектер

Математикалық өрнектерге сандық тұрақтылар, айнымалы атаулары, функциялар, арифметикалық операторлар белгілері, дөңгелек жақшалар жатады.

1.3.1 Нақты сандар

Сандарды жазуда олардың бүтін бөлігін бөлшек бөлігінен ажырату үшін «.» (нүкте) белгісі қолданылады. Мантиссаны ретінен ажырату үшін «e» белгісі қолданылады. Мысалы:

```
7 -2 5.8 2.456e-5 9.914e12
```

1.3.2 Комплекстік сандар

MATLAB-та нақты сандармен қатар комплекстік сандармен де есептеулер жүргізуге болады. ($\sqrt{-1}$) жорамал бірлігін белгілеу үшін *i* немесе *j* атты айнымалылар қолданылады.

Алгебралық түрдегі комплекстік сандарды енгізгенде жорамал бөлігіндегі сан тұрақтысы мен жорамал бірлігінің арасына көбейтінді белгісін қоймаса да болады:

```
» 7+5i
ans =
7.0000 + 5.0000i
```

Ал егер жорамал бөлік айнымалы болса, онда комплекстік сандарды енгізу кезінде оның жорамал бөлігі мен жорамал бірлігінің арасына көбейту белгісін қою керек:

```
» 7
ans =
7
» 5+ans*i
ans =
5.0000 + 7.0000i
»
```

Комплекстік сандармен жұмыс істеу функциялары 1.3.8 тарауда берілген.

1.3.3 Айнымалылар

Айнымалылар сандық, белгілік, матрицалық немесе векторлық түрде болуы мүмкін.

Айнымалы атауларының қалыптасу тәртіптері:

1. Айнымалы атауы кез келген белгі санынан тұрады, бірақ оның алғашқы 31-і ғана сақталады және анықталады.

2. Атау басқа айнымалылар мен функциялардың атауларымен ұқсас болмауы керек.

3. Атау әріптен, саннан және астыңғы сызық белгісінен тұруы мүмін.

4. Атау әріптен басталу керек.

5. Айнымалы атауларында бас әріп пен кіші әріптің айырмашылығы бар.

Айнымалы атауын қолданылуының дұрыстығын тексеру үшін

isvarname айнымалы атауы

түрде шақырылатын арнайы функция қолданылады. Бұл функция егер айнымалы атауы дұрыс болса 1 мәнін қабылдайды, ал кері жағдайда 0 мәнін қабылдайды. Мысалы:

```
» isvarname alfa
ans =
1
» isvarname Ia
ans =
0
»
```

Айнымалыға мәнді меншіктеу үшін меншіктеу операциясы қолданылады.

айнымалы атауы = өрнек

Мысалы:

» $a=5$
$a =$
5
»

Айнымалы типі айнымалыға берілген өрнек арқылы анықталады. Айнымалының мәнін экранға шығару үшін оның атын командалық жолға жазып, *Enter*-ді басу керек.

1.3.4 Жүйелік айнымалылар

MATLAB-та келесідей жүйелік айнымалылар қолданылады:

<i>i</i> не <i>j</i>	жорамал бірлік
<i>Realmax</i>	модулі бойынша ең жоғарғы нақты сан ($1.7977e+308$)
<i>Realmin</i>	модулі бойынша ең төменгі нақты сан ($2.2251e-308$)
<i>Inf</i>	шексіздік (нөлге бөлудің нәтижесі)
<i>Ans</i>	басқадай айнымалыға берілмеген соңғы есептелген өрнектің мәні
<i>NaN</i>	сан емес. Анықталмаған нәтижені көрсетеді ($0/0$ немесе <i>Inf/Inf</i> амалдарының нәтижесінде шығады)
<i>Pi</i>	π саны
<i>Eps</i>	жылжымалы нүктесі бар санмен жүргізілген амалдың дәлсіздігі (1.0 саны мен келесі жақын тұрған жылжымалы нүктесі бар санның аралығы 2^{-52} -ге тең)

Жүйелік айнымалылар жүйемен беріледі, бірақ қайтара анықталуы мүмкін.

1.3.5 Арифметикалық амалдар

Нақты және комплекстік сандарға қосу, алу, көбейту, бөлу және дәрежелік амалдар қолдануға болады (олар, тиісінше, «+», «-», «*», «/», «^») белгілерімен көрсетіледі).

Басымдық тәртібі бойынша арифметикалық операциялардың орындалу реті келесідей түрде болады:

1. Дәрежелену (^).
2. Көбейту мен бөлу (*, /).
3. Қосу мен алу (+, -).

Басымдықтары бірдей амалдар солдан оңға қарай орындала береді. Амалдардың орындалу реттілігін дөңгелек жақшаларды қолдану арқылы да өзгертуге болады.

Өрнектің мәнін есептеу үшін оны алдымен командалық жолға енгізіп, пернетақтадан *Enter*-ді басу керек:

$$\gg 4^2-5/(2+3^2)$$

ans =

$$15.5455$$

»

Комплекстік сандарды есептеуде операндтар дөңгелек жақшаға алынады:

$$\gg 4+3i-(5-7i)/(3+6i)$$

ans =

$$4.6000 + 4.1333i$$

»

Комплекстік-түйіндес сандарды есептеуде дәйекше (‘) қолданылады:

$$\gg d=7-4i$$

d =

$$7.0000 - 4.0000i$$

» *d'*

ans =

$$7.0000 + 4.0000i$$

»

1.3.6 Қатынас амалдары

Операндтарды салыстыру үшін келесідей қатынас амалдары қолданылады:

==	тең
~=	тең емес

<	кіші
>	үлкен
<=	кіші немесе тең
>=	үлкен немесе тең

Қатынас амалдарының нәтижесі «ақиқат» немесе «жалған» болады. MATLAB-та «ақиқат» 1 саны арқылы, ал «жалған» 0 саны арқылы белгіленеді.

Қатынас амалдарының басымдығы арифметикалық амалдарға қарағанда төмен болады.

1.3.7 Логикалық амалдар

Логикалық амалдар және оларға сәйкес функциялары 1.1-кестеде келтірілген.

1.1-кесте. Логикалық амалдар және функциялар

Амал	Атауы	Функция
&	логикалық ЖӘНЕ	And
	логикалық НЕМЕСЕ	Or
~	логикалық ЕМЕС	Not
	НЕМЕСЕ-ні шығару	Xor

Логикалық амалдар орындалғанда нөлге тең емес операндтар «ақиқат» болып, ал олардың нөлге тең болғандары «жалған» болып саналады. Логикалық операторлар мен функциялардың жұмысы 1.2-кестеде келтірілген.

1.2-кесте. Логикалық операторлар мен функцияларының жұмысы

Операндтар		Логикалық операторлар мен функциялар			
X	y	$x \& y$ <i>and(x,y)</i>	$x y$ <i>or(x,y)</i>	$\sim x$ <i>not(x)</i>	$x \text{ xor } y$
1	1	1	1	0	0
1	0	0	1	0	1
0	1	0	1	1	1
0	0	0	0	1	0

Логикалық амалдардың басымдықтары қатынас амалдарына қарағанда аса төмен саналады. Ал ЖӘНЕ логикалық амалының басымдығы НЕМЕСЕ амалына қарағанда жоғары саналады.

1.3.8 Элементар функциялар

MATLAB функциялары кірістірілген және сыртқы болып жіктеледі. Кірістірілген элементар функциялар тізімі 1.3-кестеде келтірілген. Сыртқы функциялар *.m* кеңейтілуімен жеке файлдарда сақталады. Сыртқы функциялар 3 тарауда қарастырылады.

1.3-кесте. MATLAB-тың кірістірілген элементар функциялары

Функция атауы	Қызметі	Функция атауы	Қызметі
<i>Тригонометриялық</i>			
$\sin(x)$	Синус	$\text{asin}(x)$	арксинус
$\sinh(x)$	гиперболалық синус	$\text{asinh}(x)$	гиперболалық арксинус
$\cos(x)$	Косинус	$\text{acos}(x)$	Арккосинус
$\cosh(x)$	гиперболалық косинус	$\text{acosh}(x)$	гиперболалық арккосинус
$\tan(x)$	Тангенс	$\text{atan}(x)$	Арктангенс
$\tanh(x)$	гиперболалық тангенс	$\text{atanh}(x)$	гиперболалық арктангенс
$\sec(x)$	Секанс	$\text{asec}(x)$	Арксеканс
$\text{sech}(x)$	гиперболалық секанс	$\text{asech}(x)$	гиперболалық арксеканс
$\csc(x)$	Косеканс	$\text{acsc}(x)$	Арккосеканс
$\text{csch}(x)$	гиперболалық косеканс	$\text{acsch}(x)$	гиперболалық арккосеканс
$\cot(x)$	Котангенс	$\text{acot}(x)$	Арккотангенс
$\text{coth}(x)$	гиперболалық котангенс	$\text{acoth}(x)$	гиперболалық арккотангенс
$\text{atan2}(y,x)$	$(-\pi, \pi)$ аралығындағы y/x -тің арктангенсі		
<i>Экспоненциалдық, көрсеткіштік және логарифмдік</i>			
$\exp(x)$	экспонента	$\log(x)$	натуралдық логарифм
$\log_{10}(x)$	ондық логарифм	$\text{nextpow2}(x)$	x -ке тең немесе үлкен болатын жақын санды алу үшін 2 санын дәрежелееу
$\text{sqrt}(x)$	квадрат түбірі		
$\text{pow2}(x)$	2 санының x дәрежесі	$\log_2(x)$	негізі 2 болатын логарифм

Функция атауы	Қызметі	Функция атауы	Қызметі
<i>Комплекстік сандармен жұмыс істеу функциялары</i>			
<i>real(x)</i>	комплекстік санның нақты бөлігі	<i>abs(x)</i>	комплекстік санның модулі
<i>imag(x)</i>	комплекстік санның жорамал бөлігі	<i>angle(x)</i>	комплекстік санның аргументі
<i>complex(a,b)</i>	$a+jb$ комплекстік санын құрайды	<i>conj(x)</i>	комплекстік түйіндес санды x -ке кері қайтарады
<i>Дөңгелектеу және бөлуден қалдықты ажырату</i>			
<i>fix(x)</i>	нөлге қарай бағыттағы жақын тұрған бүтінге қарай дөңгелектеу	<i>floor(x)</i>	теріс шексіздіктегі жақын тұрған бүтінге қарай дөңгелектеу
<i>ceil(x)</i>	оң шексіздіктегі жақын тұрған бүтінге қарай дөңгелектеу	<i>round(x)</i>	жақын тұрған бүтінге қарай дөңгелектеу
<i>mod(x,y)</i>	таңбаны есепке алуымен бүтін санды бөлуден қалдық	<i>rem(x,y)</i>	модуль бойынша бүтін санды бөлудегі қалдық
<i>sign(x)</i>	санның таңбасын анықтау		

Функцияны шақыру үшін оның атынан кейінгі дөңгелек жақшада жазып көрсету керек:

```

» sin(pi/2)
ans =
    1
»

```

Тригонометриялық функциялардың аргументі радианмен берілуі керек, ал кері тригонометриялық функциялар да нәтижелерін радианмен шығарып береді.

Элементар функциялар нақты және комплекстік аргументтермен жұмыс істеуге арналған.

Help elfun командасы қысқа деректерімен элементар функциялардың барлық тізімін шығарып береді. Функция туралы толық дерек алу үшін


help функция атауы

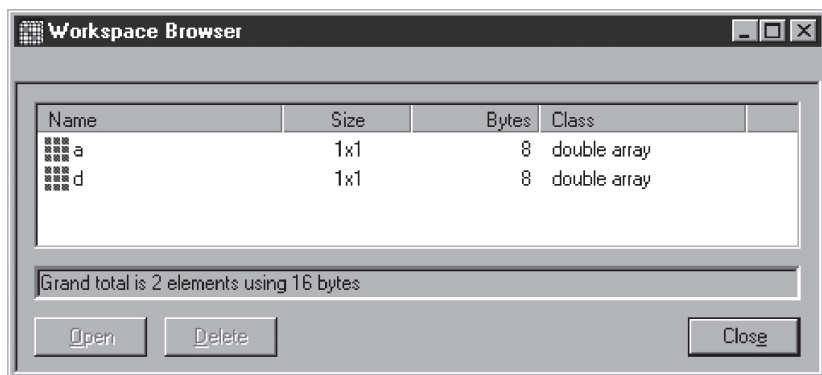
командасы қолданылады. Анықтама жүйесін қолдану 1.7 таарауда қарастырылады.

1.4 Жұмыс аймағы

Жұмыс істеу кезінде құрылған айнымалылар жадының жұмыс аймағына (*workspace*) орналасады. Жұмыс аймағындағыларды көру үшін *who* и *whos* командалары қолданылады. *Who* командасы қолданыстағы айнымалылардың аттарын экранға шығару үшін, ал *whos* командасы экранға жиымдардың өлшемдері және айнымалылардың жадыдағы шамасы мен типтері жайындағы деректерді экранға шығару үшін қолданылады:

```
» a=5
a =
    5
» d=2
d =
    2
» who
Your variables are:
a      d
» whos
Name    Size    Bytes Class
a       1x1      8 double array
d       1x1      8 double array
Grand total is 2 elements using 16 bytes
»
```

Сонымен қатар, енгізу жолына *workspace* командасын теріп шақыру арқылы, немесе  (*Workspace Browser*) батырмасын басу арқылы, не болмаса *File/Show Workspace* командасын таңдау арқылы *Workspace Browser* терезесінде (1.2-сурет) жұмыс аймағын көруге болады. *Workspace Browser* терезесінде айнымалыны бөліп алып, сонан соң *Delete* батырмасын басу арқылы өшіруге де болады, ал егер *Open* батырмасын таңдаса, онда *Editor/Debugger* (редактор/дұрыстаушы) терезесінде бөліп алған айнымалының мәнін көруге болады.



1.2-сурет. *Workspace Browser* терезесі

Жұмыс аймағын толық тазарту үшін *clear* командасы қолданылады. Бірнеше таңдамалы айнымалыларды жою үшін *clear* айнымалы атауларының тізімі («босорын» белгісі арқылы жазылады)

командасы қолданылады. Мысалы:

```

» who
Your variables are:
j    q    r    u    y
» clear q
» who
Your variables are:
j    r    u    y
» clear u j
» who
Your variables are:
r    y
» clear
» who
»

```

Айнымалылардың енгізілуі мен жойылуы шараларынан жұмыс аймағы бөлшектенген күйге түседі. Ал бұл, өз кезегінде, есептеу жылдамдығының төмендеуіне алып келеді. Жұмыс аймағын жинақтау *pack* командасын орындау арқылы іске асырылады.

Жұмыс аймағын сақтау *save* командасы арқылы орындалады:

<i>save</i> файл атауы	жұмыс аймағындағы атауы көрсетілген және <i>.mat</i> кеңейтілуімен берілген барлық айнымалыларды файлға жазу
<i>save</i> файл атауы айнымалы атауларының тізімі («босорын» белгісі арқылы жазылады)	жұмыс аймағындағы барлық айнымалыларды файлға жазу

Ертеректе сақталған жұмыс аймағын жүктеу *load* командасы арқылы іске асырылады:

load файл атауы немесе *load* файл атауы айнымалы атауларының тізімі («босорын» белгісі арқылы жазылады)

Айнымалыларды әр түрлі файлдардан таңдап алып жүктеуге болады. *Load* командасының екінші бір нұсқасы айнымалыларды таңдамалы түрде жүктеуде қолданылады.

Жұмыс аймағын сақтау және оны жүктеу үшін, тиісінше, *File/Save Workspace As* және *File/Load Workspace* командалары қолданылуы мүмкін.

Жұмыс аймағын жазу және жүктеу командаларын қолдану мысалдары төменде келтірілген:

```
» save fg
» what
MAT-files in the current directory C:\MATLABR11\work
fg
» clear
» who
» load fg
» who
Your variables are:
ans    h    u
»
```

Келтірілген мысалдағы *what* командасы ағымдағы каталогындағы файлдардың атауларын шығару үшін қолданылды.

1.5 Күнделік жүргізу

Сессия күнделігі мәтіндік файлда барлық командалар мен есептеулердің нәтижелерін көрсетеді. Мұндай файлды кейіннен іске қосуға болмайды, оны тек қана қарауға болады. Күнделік файлы үнсіз келісім бойынша *.m* кеңейтілуін алады, бірақ-та кез келген басқадай кеңейтілу түрінде де берілуі мүмкін.

Күнделік жазуды бастау үшін

diary файл атауы

командасы қолданылады.

Diary off командасы файлға жазуды тоқтатады, ал *diary on* командасы күнделіктің жазылуын қайта бастайды. Күнделік файлының мәтінін экранға шығару

type файл атауы

командасы арқылы орындалады.

Келесі мысал күнделікті қолдануды сипаттап көрсетеді:

```
» diary dl
```

```
» k=5
```

```
k =
```

```
5
```

```
» diary off
```

```
» h=k/2
```

```
h =
```

```
2.5000
```

```
» diary on
```

```
» whos
```

```
Name      Size      Bytes Class
```

```
1x1          8 double array
```

```
k      1x1          8 double array
```

```
Grand total is 2 elements using 16 bytes
```

```
» diary off
```

```
» type dl
```

```
k=5
```

```
k =
```

```
5
```

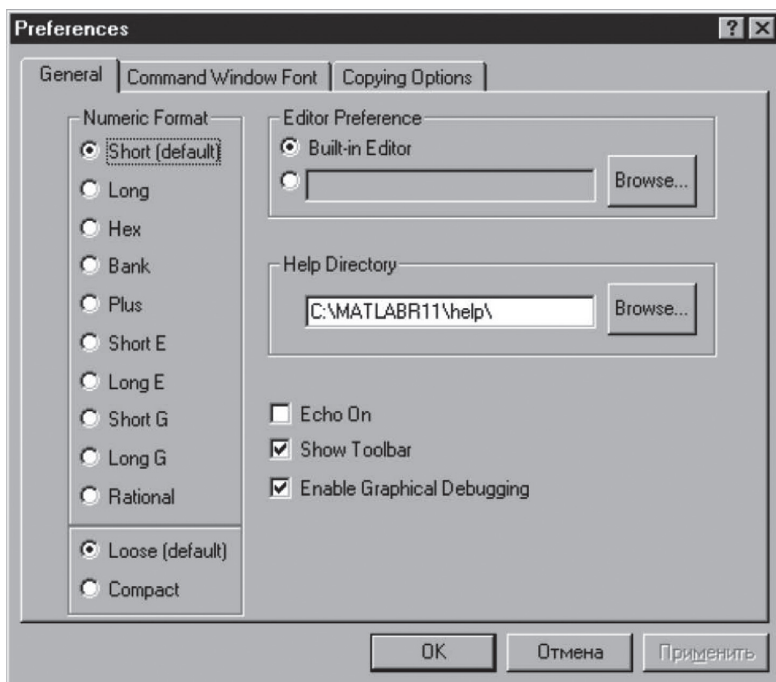
```
diary off
```

<i>Name</i>	<i>Size</i>	<i>Bytes</i>	<i>Class</i>
<i>h</i>	<i>1x1</i>	<i>8</i>	<i>double array</i>
<i>k</i>	<i>1x1</i>	<i>8</i>	<i>double array</i>
<i>Grand total is 2 elements using 16 bytes</i>			
<i>diary off</i>			
»			

Мысалда күнделіктегі файлдың ішіндегілері ерекшеленіп, бөлініп қойылған. Бұл файлды мәтіндік файлдар үшін қолданылатын кез келген әдіспен көруге болады.

1.6 Жүйе параметрлерін күйге келтіру

File/Preferences командасы параметрлерді күйге келтіретін сұхбат терезесін шақыруға мүмкіндік береді. Терезеде үш қыстырма бар (1.3-сурет). *General* қыстырмасының өрістерінің қызметтері 1.4-кестеде келтірілген.



1.3-сурет. *Preferences* терезесінің *General* қыстырмасы

1.4-кесте. General қыстырмасының өрістерінің қызметтері

<i>Өріс</i>	<i>Қызметі</i>
<i>Numeric Format</i>	сандарды көрсетудің қалыбын таңдау (1.5-кесте) және нәтижелер қорытындысының түрлері: <i>loose</i> – еркін (нәтиже қорытындысы бос жолдармен кезектеседі) и <i>compact</i> (шағын, бос жолсыз шығару).
<i>Editor Preference</i>	қолданылатын мәтіндік редакторды беру (үнсiздік бойынша салынып қойылған)
<i>Help Directory</i>	анықтама файлдары орналасатын каталог
<i>Echo On</i>	<i>Script</i> -файл сценарийдің орындау командаларын экранға шығару керек-керек еместігін анықтау
<i>Show Toolbar</i>	құрал-саймандар жақтауын көрсету
<i>Enable Graphical Debugging</i>	Графиктің дұрыстауын қолдану

1.5-кесте. Сандарды көрсетудің қалыптары

<i>Қалыбы</i>	<i>Қалыптың сипаттамасы</i>	<i>Мысал</i>
<i>short</i>	бекітілген нүктесі (5 таңбалы) бар қалыптағы қысқаша сандарды көрсету	1.2345
<i>long</i>	бекітілген нүктесі (15 таңбалы) бар қалыптағы көп сандарды көрсету	0.00045634487671
<i>hex</i>	он алтылық есептеу жүйесіндегі сандарды көрсету	3f3de8325237f974
<i>bank</i>	ондық нүктеден кейін екі таңбамен көрсету (ақшалар үшін қолданылады)	1.23
<i>plus</i>	санның тек таңбасы ғана шығарылады (+ немесе -)	+
<i>short e</i>	жылжымалы нүктесі бар қалыптағы сандарды қысқаша көрсету (мантиссаның 5 таңбасы және реттің 3 таңбасы)	1.2345e+000 4.5634e-004
<i>long e</i>	жылжитын нүктесі бар қалыптағы көп сандарды толық көрсету (мантиссаның 15 таңбасы және реттің 3 таңбасы)	4.56344876713453 e-004
<i>short g</i>	MATLAB бекітілген немесе жылжитын нүктемен берілген сандарды көрсетуде қысқаша қалыпты таңдайды	1.2345
<i>long g</i>	MATLAB бекітілген немесе жылжитын нүктемен берілген сандарды көрсетуде ұзын қалыпты таңдайды	1.234567890123456
<i>rational</i>	рационалды баған түрінде көрсету	10/3

Сонымен бірге, сандарды көрсету қалыбын *format* қалып атауы командасымен де беруге болады. Мысалы, *format long*. Экран бетіне барлық түсінікті қалыптағы деректерді шығару үшін *help format* командасын пайдалану қажет.

Command Window Font қыстырмасында командалық терезенің қаріптерінің параметрлері беріліп, ал *Copying Options* қыстырмасында көшірме опциялары таңдап алынуы мүмкін.

1.7 Анықтама жүйесін қолдану

MATLAB бойынша мәліметтерді келесідей әдістер арқылы алуға болады:

–*help* командасы;

–*lookfor* командасы;

–*HELP* мәзірі;

–*PDF* қалыбы;

–*The Math Works* фирмасының *Web*-желісі.

Функцияның қолданылуы туралы анықтамалық мәліметті

help функция атауы

командасын енгізу арқылы алуға болады.

Дерек мәтінінде функция атаулары кіші әріптермен жазылғанмен, нақты функцияда бас әріптермен жазылу керек екенін естен шығармау қажет.

MATLAB функциялары қолданылуы бойынша топтас-тырылған. *Help* командасы экран бетіне функция топтарының тізімін экранға шығарады. Нақты топ бойынша мәлімет алу үшін

help топтың атауы

командасы қолданылады.

lookfor түйінді сөз немесе *lookfor* ‘түйінді сөз тіркесі’

командасы түсініктеменің бірінші қатарындағы түйінді сөз бойынша *M*-функциясын іздеу үшін қолданылады. Түйінді сөз бойынша табылған жол экран бетіне шығарылады. *All* – бұл опцияның қосылуы. Ал

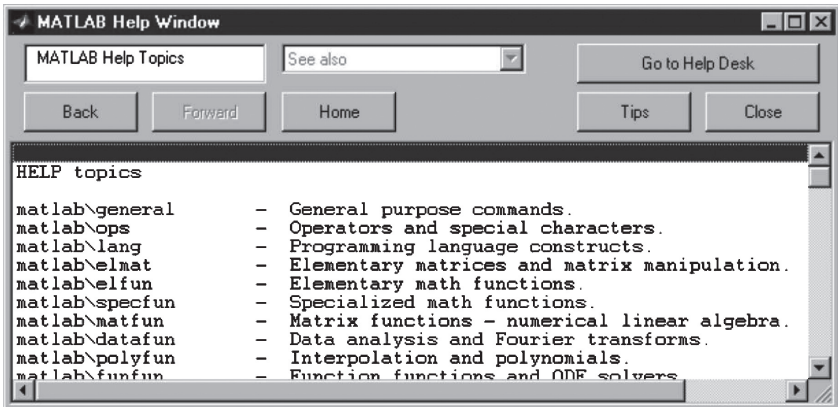
lookfor түйінді сөз – *all* немесе *lookfor* ‘түйінді сөз тіркесі’ – *all* түсініктеменің толық мәтінінің шығуын қамтамасыз етеді.

Help мәзірінің командаларының қызметі 1.6-кестеде келтірілген.

1.6-кесте. Help мәзірінің командаларының қызметі

Команда	Қызметі
Help Window	дерек терезесі
Help Tips	анықтама алуға арналған мәлімет терезесі
Help Desk (HTML)	қатты дискіге орналастырылған анықтама жүйесіне рұқсат алу
Examples and Demos	мүмкіндіктерді көрсету
About MATLAB	орнатылған нұсқа туралы мәлімет

Help Window терезесін (1.4-сурет) Help/Help Window, Help/Help Tips командалары арқылы немесе құрал-саймандар жақтауындағы **?** батырмасымен, не болмаса helpwin командасы арқылы шақыруға болады. Қажетті бөлімге көшу үшін тышқанмен оның тақырып атауына екі рет басу қажет немесе helpwin **бөлім аты** командасын қолдану керек. Back, Forward, Home батырмалары белсенді алдыңғы, келесі және бастапқы анықтамалық терезелеріне көшуге мүмкіндік береді.



1.4-сурет. Help Window терезесі

Tips батырмасы еске салу терезесін экранға шығарады. Бұл режимде lookfor, which, demo және general командалары арқылы See also тізімі белсенді болады.

Help/Help Desk (HTML) командасы HTML қалыбындағы анықтама дерегіне қол жеткізуді қамтамасыз етеді. Сол терезеде PDF қалыбындағы ақпаратқа да қол жеткізуге арналған батырма бар.

2 ЖИЫМДАРМЕН ЖҰМЫС

MATLAB-та барлық деректер жиым түрінде қолданылады. Негізінен, әрбір айнымалы жиым түріндегі шама ретінде алынады, мысалы, $R=45$ жазуы бір элементтен тұратын сан, мәні 45-ке тең R -векторы екендігін білдіреді. Мысалы, 1.2-суреттен *Workspace Browser* терезесінде берілген a және d айнымалыларының 1×1 өлшемімен көрсетілгендігін байқауға болады.

MATLAB-та бір өлшемді (вектор-баған немесе вектор-жол), екі өлшемді (матрица) және көпөлшемді жиымдармен де жұмыс істеуге болады. Жиымда элементтерді нөмірлеу 1-ден басталады. Жиым өлшемі – бұл әр өлшем бойындағы элементтер саны.

2.1 Векторлар мен матрицалардың қалыптасуы

Вектор-жолды енгізу

[вектордың атауы]=[үтір не «босорын» арқылы берілген вектор элементтерінің мәні]

түрінде жүзеге асырылады.

Вектор-бағанның элементтерін енгізу үшін нүктелі үтірмен бөлу немесе әрбір элементтің мағынасының енгізілуінен кейін *Enter* батырмасын басу қажет, яғни

```
» a=[2 7 3 9 4];
» s=[3;5;7];
» h=[3
4]
h =
    3
    4
» a
a =
    2    7    3    9    4
» s
s =
```

```
3
5
7
»
```

Матрица элементтерінің жолын енгізу кезінде «босорын» немесе үтір арқылы бөлуге болады, ал жолдары бір-бірінен нүктелі үтір немесе *Enter* батырмасын басу арқылы бөлінеді.

```
» s=[1,2,3; 4 5 6
7 8,9]
s =
     1     2     3
     4     5     6
     7     8     9
»
```

Комплекстік сандар мен формулалар жиым элементтері болады.

```
» h=[8*sin(0.54) 3-9 sqrt(-1) 3+4i]
h =
     4.1131     -6.0000     0 + 1.0000i     3.0000 + 4.0000i
»
```

A , B , C , D түрінде берілген бірнеше жиымдар немесе векторлардың көлденеңінен (бірігетін жиымдардың жол саны өзара тең болуы керек) бірігуі келесі түрде жүзеге асырылады:

$[A,B,C,D]$ немесе $[A B C D]$

A , B , C , D жиымдарының тігінен (бірігетін жиымдардың баған саны өзара тең болуы керек) біріктіру үшін нүктелі үтірмен немесе *Enter* батырмасын басу арқылы оларды бөлу қажет.

Біріккен жиымдардың мысалы:

```
» a=[1 2;3 4];
» b=[5 6 7;8 9 10];
» d=[11 12 13];
» c=[8 5 6 4];
» f=[a b]
```

```

f =
  1  2  5  6  7
  3  4  8  9 10
» h=[d c]
h =
 11 12 13  8  5  6  4
» k=[d;b
d]
k =
 11 12 13
  5  6  7
  8  9 10
 11 12 13
»

```

Тізім мәнінен басқа жиым элементтері келесі түрде берілген:

$\boxed{\text{бастапқы мән}}$: $\boxed{\text{адым}}$: $\boxed{\text{соңғы мән}}$

Егер адым шамасы бірге тең болса, онда оны бермеуге де болады.

Мысалы, элементтері бірлік адыммен 3-тен 7 аралығында жататын k векторы қалыптассын. P матрицасының әрбір жолы 3-тік адыммен және әртүрлі бастапқы және ақырғы мәндермен берілген. U матрицасын бергенде аралық бірінші жолдың алғашқы төрт элементін және екінші жолдың соңғы бес элементін береді.

```

» k=[3:1:7]
k =
  3  4  5  6  7
» P=[0:3:9; 1:3:10]
P =
  0  3  6  9
  1  4  7 10
» U=[0:3:9 100 30; 200 1:2:9]
U =
  0  3  6  9 100 30
 200  1  3  5  7  9
»

```

Егер вектордың барлық элементтері аралықпен берілсе, квадрат жақшаларды қолданбауға болады:

» $w=8:4:32$

$w =$

8 12 16 20 24 28 32

»

Бірлік адымды көрсетпеуге болады:

» $q=3:10$

$q =$

3 4 5 6 7 8 9 10

»

2.2 Жиым элементіне сілтеме

Вектор элементтеріне сілтемеде бір индекс қолданылады (вектор-жолға және вектор-бағанға да бірдей) және ол вектордың атауынан кейін дөңгелек жақшада жазылады, мысалы, $g(7)$ - g векторының жетінші элементі.

Матрица элементтеріне сілтемеде дөңгелек жақшада үтір арқылы жазылатын екі индекс қолданылады – баған нөмірі және жол нөмірі, мысалы, $k(3,7)$ - үшінші жолдың және жетінші бағананың қиылысуында орналасқан k матрицасының элементі.

Есептелген жиым элементтеріне қарапайым мәндерді енгізуге болады. Осы кезде талап ететін өлшемдегі матрица автоматты түрде құрылады. Сандық мәнге ие болмаған басқа элементтер нөлге тең болады:

» $r(3,2)=7$

$r =$

0 0

0 0

0 7

» $r(5,1)=5$

$r =$

0 0

0 0

```
0 7
0 0
5 0
```

»

Жоғарғы мысалда өлшемі 3×2 болатын матрица автоматты түрде құрылды, ал кезекті мәндерді берумен оның өлшемі үлкейтілді де, ол 5×2 болды.

Жиым элементтерінің мәндерін беруде жиымның төменгі оң жақ бұрышында орналасқан элементтен бастаған дұрысырақ болады. Бұлай элементтерді қосу жиымның өлшемін өзгертпейді (жиымға арналған жад бөлу бір рет орындалады).

Матрица элементтеріне матрицаның бірінен кейін бірі тұрған баған элементтері бойынша алынған бірөлшемді жиымдағы орындарын анықтайтын бір индексті қолдану арқылы сілтеме жасауға болады. Мысалы, $f(2,2)$ және $f(6)$ өрнектері f жиымының бір элементіне ғана жасалған сілтеме екенін көрсетеді:

```
» f=[1 2 ;3 4; 5 6; 7 8]
```

```
f =
```

```
1 2
3 4
5 6
7 8
```

```
» f(2,2)
```

```
ans =
```

```
4
```

```
» f(6)
```

```
ans =
```

```
4
```

```
»
```

2.3 «Қос нүкте» операторын қолдану

Жиым элементтерінің индекс аралықтарын беру үшін «:» («қос нүкте») операторы қолданылады.

Бұл оператордың көмегімен «қос нүкте» жиымның бөлігін ерекшелеуге болады, мысалы,

```
m(in:ik,jn:jk)
```

өрнегі m матрицасынан in -нен ik -ға дейінгі жолды, jn -нен jk -ға дейінгі бағанды бөліп алады. Мысалы:

```
» h=[1 2 3 ;4 5 6;7 8 9]
h =
     1     2     3
     4     5     6
     7     8     9
» h(2:3,1:2)
ans =
     4     5
     7     8»
```

M матрицасында бағаны мен жолының көрсетілуі қос нүкте операторы арқылы келесідей жүргізіледі:

$M(i, j)$ және $M(i, :)$ (баған нөмірі)

Мысалы:

```
» d=[1 2 3 ;4 5 6;7 8 9]
d =
     1     2     3
     4     5     6
     7     8     9
» d(:,2)
ans =
     2
     5
     8
» d(1,:)
ans =
     1     2     3
»
```

Қос нүкте операторын соңғы жол нөмірі орнына (бағанның) қолдануда *end* сөзін қолдануға болады. Төменгі мысалда экранға жиымның екінші жолынан бастап екінші бағанның соңына дейінгі матрица үзіндісі шығарылады.

```
» d=[1 2 3 ;4 5 6;7 8 9]
d =
     1     2     3
```

```

    4  5  6
    7  8  9
» d(2:end, 2:end)
ans =
    5  6
    8  9
»

```

2.4 Берілген жиымдағы жолды және бағанды өшіру

Берілген жиымда жолды және бағанды өшіруде бос жиым қолданылады (қос квадрат жақша []). Бос жиым өлшемі 0×0 . Берілген жиымдағы жолды және бағанды өшіруде бос жиым енгізу қажет.

```

» d=[1 2 3; 4 5 6; 7 8 9]
d =
    1  2  3
    4  5  6
    7  8  9
» d(1,:)=[]
d =
    4  5  6
    7  8  9
»

```

Бірден бірнеше жолды немесе бағанды өшіруге болады. Төменгі мысалда бірінші және екінші бағандар өшірілген.

```

» d=[1 2 3; 4 5 6; 7 8 9]
d =
    1  2  3
    4  5  6
    7  8  9
» d(:,1:2)=[]
d =
    3

```

```
6
9
»
```

2.5 Векторлар мен матрицаларға элементар функцияларды қолдану

MATLAB-та элементар функцияларға жиымдар жатады. Мұндай жағдайда функция аргумент өлшеміне сай өлшемдегі жиымды кері қайтарады. Қайтарылатын жиымның әр элементі жиымның элементі – аргументіне сәйкес функция мәніне тең болады. Мысалы:

```
» h=[1 2 3 ;4 5 6;7 8 9]
h =
1 2 3
4 5 6
7 8 9
» sqrt(h)
ans =
1.0000 1.4142 1.7321
2.0000 2.2361 2.4495
2.6458 2.8284 3.0000
»
```

2.6 Матрицалық және векторлық амалдар

Матрица және векторлық амалдар (2.1-кесте) векторлық және матрицалық есептеу ережелерімен сәйкес.

2.1-кесте. Матрицалық және векторлық амалдар

Амал	Сипаты	Мысал
+	бірдей өлшемдегі матрицалардың (векторлардың) қосындыларын есептеу. Амал нәтижесі жиым өлшеміне сәйкес, әрбір элемент келесі жиым элементтеріне лайықты	» b=[1 2 3;4 5 6]; » c=[7 2 4; 4 8 3]; » b+c ans = 8 4 7 8 13 9 »

-	бір өлшемдегі жиымдардың әр түрлілігі	$\gg c = [7 \ 2 \ 4; 4 \ 8 \ 3];$ $\gg b = [1 \ 2 \ 3; 4 \ 5 \ 6];$ $\gg b - c$ $ans =$ $\begin{matrix} -6 & 0 & -1 \\ 0 & -3 & 3 \end{matrix}$ \gg
*	матрицаларды көбейту, векторлардың скалярлық көбейтіндісі (егер сол операнд бағандарының саны оң операнд жолдың санымен бірдей болса, онда жиымдарды көбейтуге болады)	$\gg a = [1 \ 2; 3 \ 4];$ $\gg b = [1 \ 2 \ 3; 4 \ 5 \ 6];$ $\gg a * b$ $ans =$ $\begin{matrix} 9 & 12 & 15 \\ 19 & 26 & 33 \end{matrix}$
'	(дәйекше) - нақты элементтері бар жиымдардың тасымалдануы, жиымдардың комплекстік элементтермен комплекстік түйіндесуі және тасымалдануы	$\gg c = [7 \ 2 \ 4; 4 \ 8 \ 3]$ $c =$ $\begin{matrix} 7 & 2 & 4 \\ 4 & 8 & 3 \end{matrix}$ $\gg c'$ $ans =$ $\begin{matrix} 7 & 4 \\ 2 & 8 \\ 4 & 3 \end{matrix}$ \gg $\gg d$ $d =$ $3.0000 + 1.0000i \quad 3.0000 - 7.0000i$ $\gg d'$ $ans =$ $3.0000 - 1.0000i$ $3.0000 + 7.0000i$
.'	жиымдарды тасымалдау	$\gg d = [3.0000 + 1.0000i \quad 3.0000 - 7.0000i]$ $d =$ $3.0000 + 1.0000i \quad 3.0000 - 7.0000i$ $\gg d'$ $ans =$ $3.0000 + 1.0000i$ $3.0000 - 7.0000i$ $\gg c = [5 \ 6 \ 7]; c'$ $ans =$ $\begin{matrix} 5 \\ 6 \\ 7 \end{matrix}$ \gg

^	<p>матрицаны дәрежеләу (матрицалардың өз-өзіне көбейтүі жүргізіледі).</p> <p>матрицаны (-1)-ге дәрежеләу кері матрицаны береді. Сонымен қатар, <i>Inv</i> (<i>A</i>) функциясы кері (<i>A</i>) матрицасын есептеу үшін арналған.</p> <p>матрицаны (-<i>n</i>)-ге кері дәрежеләу, ол кері матрицаны өз-өзіне <i>n</i> рет көбейтуіне тең.</p>	<pre> » a=[2 3; 6 7] a = 2 3 6 7 » a^3 ans = 206 255 510 631 » a*a*a ans = 206 255 510 631 » a^-1 ans = -1.7500 0.7500 1.5000 -0.5000 » ans*a ans = 1 0 0 1 » a^-2 ans = 4.1875 -1.6875 -3.3750 1.3750 » inv(a) ans = -1.7500 0.7500 1.5000 -0.5000 » </pre>
---	--	---

Егер қосу, азайту, көбейту немесе бөлу амалдарының бірі немесе бөлу амалының операнды сан болса, жиымның әрбір элементіне және санға лайықты амал орындалады. Мысалы, *A* матрицасының және 2 санының қосындысын тауып, алынған матрицаны 5-ке бөлу керек:

<pre> » A=[1 2; 3 4] A = 1 2 3 4 » A+2 ans = 3 4 </pre>
--

```

    5  6
  » ans/5
  ans =
    0.6000  0.8000
    1.0000  1.2000
  »

```

d және q векторларының векторлық көбейтіндісін есептеу үшін (үш элементтен тұрады) $cross(d,q)$ функциясы қолданылады:

```

  » d=[1 2 3];
  » q=[4 5 6];
  » cross(d,q)
  ans =
    -3  6  -3
  »

```

Бірөлшемді d және q векторларының скалярлық көбейтіндісін табу $dot(d,q)$ функциясы арқылы іске асырылады. Оның қолданылуы кезінде векторлардың элементтерінің саны мен бағандар саны және жолдар санының комбинациясы бірдей болулары керек:

```

  » a=[1 2 3 4];
  » b=[4 3 2 1];
  » dot(a,b)
  ans =
    20
  » a*b
  ??? Error using ==> *
  Inner matrix dimensions must agree.

  » a*b'
  ans =
    20
  »

```

Бұл мысалдағы $a*b$ амалы қателікке алып келді. Себебі, екі операнд та жолдық. Ал енді b векторын тасымалдаудан кейін қателіктің шығуына ешқандай да себеп болмайды.

2.7 Жиымдарға элементтері бойынша орындалатын амалдар

Матрицалар мен векторларға элементтер бойынша амал (2.2-кесте) тек бір өлшемді жиымдарда орындалады (егер олардың екі операнды да жиым болса). Нәтижесінде бір өлшемдегі және бір типтегі жиым болады.

2.2-кесте. Матрица және вектормен элементтер бойынша амалдар

Амал	Сипаты	Мысал
.*	Бір жиымды келесі жиымға элементтері бойынша көбейту. Әрбір элементке қатысты жиым элементінің көбейтіндісі қорытынды жиым болып табылады	<p>» $a=[1\ 4; 2\ 5]$ $a =$ $\begin{matrix} 1 & 4 \\ 2 & 5 \end{matrix}$ » $d=[1\ 2 ; 3\ 4]$ $d =$ $\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$ » $a.*d$ $ans =$ $\begin{matrix} 1 & 8 \\ 6 & 20 \end{matrix}$ »</p>
./	Сол жақтағы операндты оң жақтағы операндқа элементтер бойынша бөлу	<p>» $a./d$ $ans =$ $\begin{matrix} 1.0000 & 2.0000 \\ 0.6667 & 1.2500 \end{matrix}$</p>
.\	Оң жақтағы операндты сол жақ операндқа элементтер бойынша бөлу	<p>» $a.\d$ $ans =$ $\begin{matrix} 1.0000 & 0.5000 \\ 1.5000 & 0.8000 \end{matrix}$</p>

.^	Элементтер бойынша дәрежелуе	<pre> » a.^d ans = 1 16 8 625 » a.^2 ans = 1 16 4 25 » a^2 ans = 9 24 12 33 » </pre>
----	------------------------------	--

Ешқандай цикл қолданбай, жиымдардың элементтері бойынша амалы, бір ғана жалғыз формуламен жиым элементтерінің мәндерін есептеуге мүмкіндік береді. Мысалы, функция мәндерінің кестесін есептеу үшін қос нүкте операторы арқылы вектор аргументтерін құруға болады. Содан кейін, матрицалық амалдар мен элементар функцияларды қолдана отырып, элементтер бойынша векторлық функцияларды есептеуге болады. Мысалы,

$$f(x) = \frac{\cos(2x)}{\sqrt{1+x \cdot \sin^2(x)}}$$

функциясының мәндерінің кестесін 5-тен 7-ге дейінгі аралықта 0.5 адымымен есептеу керек:

```

» x=5:0.5:7
x =
    5.0000    5.5000    6.0000    6.5000    7.0000
» f=cos(2*x)./sqrt(1+x.*sin(x).^2)
f =
   -0.3546    0.0023    0.6964    0.7956    0.0682
»

```

2.8 Жиыммен жұмыс істеу функциялары

2.3-кестеде жиымдарды түрлендіру мен функцияларды құру көрсетілген. Егер функцияны шақыру кезінде жиымның шамасы көрсетілсе, шақыру функциясы бір аргументпен квадратты матрицаны құрады.

2.3-кесте. Жиыммен жұмыс істеу функциялары

Функция	Орындалатын әрекет	Мысал
<i>zeros(n,m)</i> <i>zeros(n)</i>	Нөлдік элементтермен $n \times m$ өлшемдік матрица құру. Бұны орындау үшін, оң жақтағы төменгі бұрышты 0-ге теңестіреміз. Егер функцияны шақыру кезінде бір аргумент берілсе, онда квадраттық матрица қалыптасады	» $s = \text{zeros}(2,2)$ $s =$ $\begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix}$ » $h(3,2)=0$ $h =$ $\begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix}$ »
<i>ones(n,m)</i> <i>ones(n)</i>	Әрбір элементі 1-ге тең $n \times m$ өлшемді матрица құру	» $\text{ones}(2,3)$ $\text{ans} =$ $\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$ »
<i>eye(n,m)</i> <i>eye(n)</i>	Бас диагональдарының элементі 1-ге, қалғаны 0-ге тең $n \times m$ өлшемді матрица құру	» $\text{eye}(3)$ $\text{ans} =$ $\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$ » $\text{eye}(2,3)$ $\text{ans} =$ $\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{matrix}$
<i>rand(n,m)</i> <i>rand(n)</i>	0 мен 1-дің арасында бірдей орналасқан кездейсоқ сандардың $n \times m$ өлшемді матрицасын құру. Егер бір ғана аргумент берілсе, n -реттік квадраттық матрица құрылады	» $\text{rand}(2,3)$ $\text{ans} =$ $\begin{matrix} 0.8913 & 0.4565 \\ 0.8214 & 0.0185 \\ 0.7621 & 0.0185 \\ 0.4447 & \end{matrix}$ »

$randn(n,m)$ $randn(n)$	Нормальдық заңдылықпен орналасқан (математикалық күтумен - 0 және квадрат ауытқумен - 1) элементтері кездейсоқ сандар болатын матрица құру. Аргументері $rand$ функциясына сәйкес беріледі	» $randn(2)$ $ans =$ -0.4326 0.1253 -1.6656 0.2877 »
$magic(n)$	Әрбір баған, жол және диагональ элементтерінің қосындылары өзара тең болатын n -реттік квадрат матрицасын - сиқырлы матрица құру	» $magic(3)$ $ans =$ 8 1 6 3 5 7 4 9 2
$diag(A)$ $diag(A,k)$	Егер A функциясының аргументі ретінде матрица болса, функция бас диагональ элементтерінің векторын қайтарады. Егер A функциясының аргументі ретінде вектор болса, функция негізгі диагональ элементтері нақты вектор элементтеріне тең болатын диагональдық матрицаны кері қайтарады. Егер функцияны шақыруы кезінде екі параметр берілсе, екінші параметр диагональ нөмірін анықтайды (санау нүктесі – негізгі диагональ, ол 0 тең. Диагоналдан жоғары орналасқан элементтер оң нөмірге ие, диагоналдан төмені - теріс)	» $f=[1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9]$ $f =$ 1 2 3 4 5 6 7 8 9 » $diag(f)$ $ans =$ 1 5 9 » $g=[11\ 22\ 33];$ » $diag(g,-2)$ $ans =$ 0 0 0 0 0 0 0 0 0 0 11 0 0 0 0 0 22 0 0 0 0 0 33 0 0 »
$cat(q, A, B, C)$	A, B, C матрицаларының аралық вертикаль ($q=1$) немесе горизонталь ($q=2$) бойынша қосылуы	» $f=[1\ 2\ 3];$ » $k=[4\ 4\ 5];$ » $cat(1,f,k)$ $ans =$ 1 2 3 4 4 5 » $cat(2,f,k)$ $ans =$ 1 2 3 4 4 5 »

$flipr(A)$	A матрицаларының бағандарын кері ауыстырады (матрица бағандарын салыстырмалы вертикаль ось бойынша ауыстыру)	<pre> » d=[1 2 3; 4 5 6; 7 8 9] d = 1 2 3 4 5 6 7 8 9 » flipr(d) ans = 3 2 1 6 5 4 9 8 7 </pre>
$flipud(A)$	A матрицасының жолдарын кері алмастыру	<pre> » d=[1 2 3; 4 5 6; 7 8 9] d = 1 2 3 4 5 6 7 8 9 » flipud(d) ans = 7 8 9 4 5 6 1 2 3 » </pre>
$rot90(A)$	A матрицасын сағат тіліне қарсы 90° -қа бұру	<pre> » d d = 1 2 3 4 5 6 7 8 9 » rot90(d) ans = 3 6 9 2 5 8 1 4 7 » </pre>
$inv(A)$	Кері матрицаны есептеу	<pre> » a=[2 3; 6 7] a = 2 3 6 7 » inv(a) ans = -1.7500 0.7500 1.5000 -0.5000 » </pre>

$det(A)$	Квадраттық матрицаның анықтаушысын есептеу	<pre> » a=[2 3; 6 7] a = 2 3 6 7 » det(a) ans = -4 » </pre>
$ndims(A)$	Көпөлшемді жиымның сандық өлшемін есептеу	<pre> » s=[4 5; 7 8] s = 4 5 7 8 » ndims(s) ans = 2 </pre>
$size(A)$	Бірінші элементі A матрицасының жолдарының саны, екіншісі - бағандарының саны болатын векторды қайтарады	<pre> » h=[2 3 4; 4 5 6] h = 2 3 4 4 5 6 » s=size(h) s = 2 3 » s(1) ans = 2 » s(2) ans = 3 » </pre>
$length(V)$	V вектор элементерінің санын қайтарады	<pre> » s=[1 8 3 7] s = 1 8 3 7 » length(s) ans = 4 » </pre>

$sum(A)$ $sum(A,k)$	<p>A векторы элементтерінің қосындысын қайтарады. Егер A матрица болса, функция әрбір баған элементтерінің қосындыларының векторын қайтарады.</p> <p>Функция екі аргументпен шақырылуы мүмкін: бірінші - жиым, екінші аргумент жол ($k=2$) немесе баған элементтерінің қосындыларын есептейді</p>	<pre> » s=[1 4 3 7];sum(s) ans = 15 » h=[1 2 3; 4 5 6];sum(h) ans = 5 7 9 » h=[1 2 3; 4 5 6];sum(h,2) ans = 6 15 » </pre>
$prod(A)$ $prod(A,k)$	<p>A векторы элементтерінің көбейтіндісін қайтарады. Егер A матрица болса, функция әрбір баған элементтерінің көбейтіндісінің векторын қайтарады.</p> <p>Екінші параметр sum функциясы секілді, жиым жолының және бағанының көбейтіндісін есептейді.</p>	<pre> » s=[1 4 3 7];prod(s) ans = 84 » h=[1 2 3; 4 5 6];prod(h) ans = 4 10 18 » </pre>
$sort(A)$ $sort(A, k)$	<p>Вектор элементтерін (матрица бағандарының элементтерін ($k=1$ немесе екінші аргумент жоқ болса) немесе матрица жолдарының элементтерін ($k=2$) өсуі бойынша жинақтау.</p> <p>Егер екі шығыс параметрі бар функцияны шақырса, біріншіге жинақталған жиым орналасады, екіншіге жинақталған жиым элементтердің индекстерінің ескі мәндері орналасады.</p>	<pre> » f=[3 6 1 3];sort(f) ans = 1 3 3 6 » h=[7 3 5; 1 5 2] h = 7 3 5 1 5 2 » sort(h) ans = 1 3 2 7 5 5 » sort(h,2) ans = 3 5 7 1 2 5 » [a,b]=sort(h,2) a = 3 5 7 1 2 5 b = 2 3 1 1 3 2 » </pre>

<p><i>sortrows(A)</i> <i>sortrows(A,k)</i></p>	<p><i>Sortrows(A)</i> - матрица жолдарын бірінші баған элементтерінің өсуі бойынша жинақтау <i>Sortrows(A, k)</i> - <i>k</i> баған элементтерін жол ауыстыру арқылы өсуі бойынша жинақтау. Егер <i>k</i> вектор болса, нөмірі бірінші, содан соң (элементтердің теңдігі бойынша) екіншісіне және т.б. вектор элементіне тең болатын баған бойынша жинақтау жүреді</p>	<p>» $a=[1\ 7\ -5; -4\ 7\ 9; 4\ -1\ 0]$ $a =$ $\begin{matrix} 1 & 7 & -5 \\ -4 & 7 & 9 \\ 4 & -1 & 0 \end{matrix}$ » <i>sortrows(a)</i> $ans =$ $\begin{matrix} -4 & 7 & 9 \\ 1 & 7 & -5 \\ 4 & -1 & 0 \end{matrix}$ » <i>sortrows(a,[2,1])</i> $ans =$ $\begin{matrix} 4 & -1 & 0 \\ -4 & 7 & 9 \\ 1 & 7 & -5 \end{matrix}$ » <i>sortrows(a,2)</i> $ans =$ $\begin{matrix} 4 & -1 & 0 \\ 1 & 7 & -5 \\ -4 & 7 & 9 \end{matrix}$ »</p>
<p><i>min(A)</i> <i>max(A)</i> <i>min(A, [], k)</i> <i>max(A, [], k)</i></p>	<p>Ең аз (көп) вектор элементін немесе матрицаның әрбір бағанының ең аз (көп) элементтерінің векторын қайтару Матрица жолдарының ең аз (ең көп) элементтерін (<i>k=2</i>) немесе оның бағандарының (<i>k=1</i>) элементтерін іздейтін үш параметрі бар функциясны шақыру түрі Функция қайтарылатын екі параметрмен шақырылуы мүмкін: біріншісі - ең аз (ең көп) элемент немесе олардың векторы, екіншісі - ең аз (ең көп) элементтер Егер ең аз (ең көп) вектордағы элементтер (жолға, бағанға) бірнеше функция бірінші элемент көрсеткіштерін қайтарады</p>	<p>» <i>f</i> $f =$ $\begin{matrix} 3 & 6 & 1 & 3 \end{matrix}$ » <i>min(f)</i> $ans =$ 1 » $[a,b]=min(f)$ $a =$ 1 $b =$ 3 » <i>h</i> $h =$ $\begin{matrix} 7 & 3 & 5 \\ 1 & 5 & 2 \end{matrix}$ » $[a,b]=min(h)$ $a =$ $1\ 3\ 2$ $b =$ $\begin{matrix} 2 & 1 & 2 \end{matrix}$ » <i>max(h,[],2)</i> $ans =$ 7 5 »</p>

<p><i>any(A)</i> <i>any(A, k)</i></p>	<p>егер берілген векторда бір нөлдік емес элемент болса, онда 1-ді қайтарады. Егер аргумент матрица болса, функция векторды қайтарады, элементтері матрица бағандарымен сәйкес. Функция екі параметрмен шақырылуы мүмкін: біріншісі - жиым, екіншісі - бағандарды анықтайды ($k=1$) немесе жолдарды ($k=2$) анықтайды, ол үшін нөлдік емес элементтердің бар-жоғын анықтау керек</p>	<p>» $c=[0\ 0; 4\ 6; 2\ 0]$ $c =$ $0\ 0$ $4\ 6$ $2\ 0$ » <i>any(c)</i> $ans =$ $1\ 1$ » <i>any(c,2)</i> $ans =$ 0 1 1 »</p>
<p><i>all(A)</i> <i>all(A, k)</i></p>	<p>егер берілген векторда нөлдік элемент жоқ болса, онда 1-ді қайтарады. Егер аргументі матрица болса, функция элементтері матрица бағандарымен сәйкес векторды қайтарады. Функция екі параметрмен шақырылуы мүмкін: біріншісі - жиым, екіншісі - бағандарды анықтайды ($k=1$) немесе жолдарды ($k=2$) анықтайды, ол үшін нөлдік емес элементтердің бар-жоғын анықтау қажет</p>	<p>» c $c =$ $0\ 0$ $4\ 6$ $2\ 0$ » <i>all(c)</i> $ans =$ $0\ 0$ » <i>all(c,2)</i> $ans =$ 0 1 0 »</p>
<p><i>reshape</i> <i>(A,n,m)</i></p>	<p>A жиымының баған элементтерінен тізбектелу жолымен құрылған $n \times m$ өлшемді жиымын қайтарады</p>	<p>» $f=[2\ 4; 5\ 6; 4\ 5]$ $f =$ $2\ 4$ $5\ 6$ $4\ 5$ » $s=reshape(f,1,6)$ $s =$ $2\ 5\ 4\ 4\ 6\ 5$</p>
<p><i>repmat</i> <i>(A,n,m)</i></p>	<p>A блоктың n вертикаль және m горизонталь бойынша екі өлшемді жиымды қайтарады</p>	<p>$a =$ $1\ 2$ $3\ 4$ » $A=repmat(a,2,2)$ $A =$ $1\ 2\ 1\ 2$ $3\ 4\ 3\ 4$ $1\ 2\ 1\ 2$ $3\ 4\ 3\ 4$</p>

$k=find(x)$ $k=find$ ($xuapm$)	$k=find(x)$ және $[i,j]=find(X)$ нөлдік емес x бірөлшемді және екі өлшемді X жиым элементтерінің индекстерін қайтарады	» a $a =$ 1 2 -7 8 4 -23 56 0 3
$[i,j]=find$ (X)	$k=find(xuapm)$ және $[i,j]=find(Xuapm)$ шартты қанағаттандыратын	» $[i,j]=find(a>5)$
$[i,j]=find$ ($Xuapm$)	x бірөлшемді және екі өлшемді X жиым элементтерінің индекстерін қайтарады	$i =$ 2 3 $j =$ 1 1 »

Функцияларды өңдеудің толық тізімі *help datafun* командасымен шақырылуы мүмкін.

2.9 Сызықтық теңдеулер жүйелерін шешу

$A \cdot x = B$ (мұндағы A - коэффициенттер матрицасы, B - бос мүшелердің векторы, x - шешімдер векторы) түріндегі сызықтық теңдеулер жүйелерін шешуде $x = A \setminus B$ амалын қолданады.

Сонымен қатар, сызықтық теңдеулер жүйесін коэффициенттер матрицасын айналдыру жолымен шешуге де болады: $x = inv(A) * B$. Айта кетер бір жайт - ол шешімнің рационалдық болмайтындығы.

Мысалы, теңдеулердің жүйесін шешейік.

$$\begin{cases} 2 \cdot x_1 + 4 \cdot x_2 + 7 \cdot x_3 = 3 \\ 4 \cdot x_1 + 5 \cdot x_2 + 8 \cdot x_3 = 5 \\ 3 \cdot x_1 + 5 \cdot x_2 + x_3 = 1 \end{cases}$$

» $A=[2 \ 4 \ 7; 4 \ 5 \ 8; 3 \ 5 \ 1]$
$A =$
2 4 7
4 5 8
3 5 1
» $B=[3; 5; 1]$
$B =$
3

```

5
1
» x1=A\B
x1 =
1.0444
-0.5111
0.4222
» x2=inv(A)*B
x2 =
1.0444
-0.5111
0.4222
» A*x1
ans =
3.0000
5.0000
1.0000
»

```

B/A амалы $B*inv(A)$ өрнегіне эквивалентті болады.

2.10 Жиымдарға қатынас және логикалық амалдарды қолданудың ерекшеліктері

Қатынас амалының операнды ретінде сандар да, жиымдар да бола алады. Салыстырылатын жиымдар бір өлшемді болулары қажет. Салыстыру амалының нәтижесінде бірлер мен нөлдерден тұратын өлшемі операндыларға сәйкес жиым алынады (салыстыру нәтижесіндегі әрбір элемент салыстырылатын жиым элементіне сәйкес болады):

```

» a=[1 5; 2 6]
a =
    1    5
    2    6
» b=[4 6; 1 4]
b =
    4    6

```

```

1 4
» a>b
ans =
0 0
1 1
»

```

Егер операндтардың біреуі скаляр, ал екіншісі жиым болса, онда скаляр әрбір жиым элементімен салыстырылады. Алынған нәтиже шамасы жиым-операндтың мөлшерімен сәйкес болатын жиым болады:

```

» a
a =
1 5
2 6
» a<=2
ans =
1 0
1 0
»

```

Комплекстік элементтері бар жиымдарға $>$, $>=$, $<$, $<=$ амалдарын орындаған кезде, жиымдардың комплекстік элементтерінің тек нақты бөліктері ғана салыстырылады. Ал $==$ және \sim амалдарын орындаған кезде, нақты және жорамал бөліктері ескеріледі:

```

» c=[3+1i 3-7i]
c =
3.0000 + 1.0000i 3.0000 - 7.0000i
» d=[1-3i 3-7i]
d =
1.0000 - 3.0000i 3.0000 - 7.0000i
» c>d
ans =
1 0
» c==d
ans =
0 1
»

```

Бірөлшемді матрицаларда логикалық амалдар матрицаның әр элементіне орындалады (1.2-кесте):

```
» a=[1 3 0 0];b=[0 1 3 2]; a&b
ans =
    0    1    0    0
» a|b
ans =
    1    1    1    1
»
```

2.11 Амалдарды орындаудағы басымдықтар

Төменде амалдарды орындаудағы басымдықтардың реті көрсетілген:

1. Дөңгелек жақша.
2. Тасымалдау (.), комплекстік түрде тасымалдау (‘), дәрежелену (^), элементтік дәрежелену (.^).
3. Қосылғыш (+), азайтқыш (-). Логикалық терістеу (~).
4. Көбейту және бөлу (*, ./, .\, *, /, \).
5. Қосу (+) және азайту (-).
6. Амалдар арасындағы байланыс (>, >=, <, <=, ==, ~=).
7. Логикалық ЖӘНЕ (&).
8. Логикалық НЕМЕСЕ (|).

2.12 Матрицалармен жұмыс істеудің мысалдары

1-мысал. Матрицаның қосымша диагоналының элементтерін кемуі бойынша орналастыру керек.

```
»A=[1 2 7; 8 4 23; 56 3 9]
A =
    1    2    7
    8    4   23
   56    3    9
```



```

»A1=rot90(A);D=diag(A1);D1=flipud(sort(D));B1=diag(D);C
l=diag(D1);A1=A1-B1+C1;
»A=rot90(rot90(rot90(A1)))
A =
    1    2   56
    8    7   23
    4    3    9
»

```

2-мысал. Бірөлшемді A жиымы берілген. Барлық оң элементтерді P жиымына, ал теріс және нөлдік элементтерді N жиымына орнату керек. P және N жиымдарын қалыптастыру кезінде элементтерінің алғашқы орналасу ретін сақтау керек. Алынған жиымдардың элементтер санын экранға шығару керек.

```

» A=[1 2 -7 8 4 -23 56 0 3 9]
A =
    1    2   -7    8    4  -23   56    0    3    9
» at=A.';at(:,2)=at(:,1)>0;as=sortrows(at,2);np=sum(at(:,2));n
=length(A);
» N=as(1:n-np,1).'
N =
   -7  -23    0
» P=as(n-np+1:n,1).'
P =
    1    2    8    4   56    3    9
» length(P)
ans =
    7
» length(N)
ans =
    3
» n

```

```
n =  
    10  
» np  
np =  
    7  
»
```

3-мысал. A жиымы берілген. Жол мен бағанның қиылысқан жерінде максималдық мән орналасқан элементін өшіру керек.

```
» A=[1 2 -7; 8 4 -23; 56 0 3]
```

```
A =  
    1    2   -7  
    8    4  -23  
   56    0    3
```

```
» maxa=max(max(A))
```

```
maxa =  
    56
```

```
» [i,j]=find(A==maxa)
```

```
i =  
    3
```

```
j =  
    1
```

```
»
```

```
A(i,:)=[]; A(:,j)=[];
```

```
» A
```

```
A =  
    2   -7  
    4  -23
```

```
»
```


4-мысал. A жиымы берілген. Матрицаның бірінші жолына сәйкес келетін жолдар санын анықтау керек.

```
» A=[1 2 -7; 8 4 -23;1 2 -7; 56 0 3;1 2 -7; ]
A =
    1     2    -7
    8     4   -23
    1     2    -7
   56     0     3
    1     2    -7»
v=size(A)
v =
     5     3
» n=v(1)
n =
     5
» af= repmat(A(1,:),n-1,1)
af =
    1     2    -7
    1     2    -7
    1     2    -7
    1     2    -7
» y=A(2:end,')==af
y =
     0     0     0
     1     1     1
     0     0     0
     1     1     1
» ns=sum(all(y,2))
ns =
     2
»
```

3 М-ФАЙЛДАР: СЦЕНАРИЙЛЕР ЖӘНЕ ФУНКЦИЯЛАР

MATLAB-та мәтіндік файлда жазылған командалардың ретпен орындалу режимі сияқты командалық режимде де жұмыс істеуге болады (командалар мен операторлар командалық терезеге енгізіледі, дереу орындалады және нәтижесі алынады). MATLAB-тағы команда мен операторлары бар файл *.m* кеңейтілуімен жазылады және ол *M-файл* деп аталады.

M-файлдың екі түрі бар: *сценарий* және *функция*. Сценарийлер мен функциялар MATLAB-тың кез келген командалары мен операторларынан тұруы мүмкін.

M-файлды құру үшін *File/New/M-file* (құрал-саймандар жақтауындағы  *New M-file* батырмасы) командасын қолдану қажет. Бұл команда бағдарламаны енгізу мен шығаруға арналған M-файлдың редакторлық терезесін ашады. Енгізу аяқталғаннан кейін M-файл сақталған болуы тиіс (ағымдағы немесе кез келген басқа каталогта да). M-файлды қолдану үшін осы файлға алып баратын іздеу жолы орнатылуы тиіс.

Егер құрылған M-файлға өзгерістер енгізу талап етілсе, онда оны ашып, өзгеріс енгізіп және модификацияланған файлды сақтау қажет (*File/Save* командасы (*File/Save as* – басқа атпен сақтау) немесе редактордың құрал-саймандар жақтауындағы ұқсас батырма).

M-файлда бір жолда таңбалармен бөлінген бірнеше команда жазылуы мүмкін (нүктелі үтір), бірақ көрнектілік үшін оларды жеке жолдарда орналастырған ыңғайлы. M-файлдармен жұмыс істеген кезде нүктелі үтір жеке команда нәтижелерінің командалық терезесіне шығарылуы үшін де қолданылуы мүмкін. Ұзын өрнек бірнеше жолды алу мүмкін. Жолдарды тасымалдау кезінде «көп нүкте» таңбасы қолданылады.

M-файлда айнымалылардың түрлері жарияланбайды, ал олардың мәндері меншіктелгеннен кейін автоматты түрде анықталады.

Бағдарламаны қолдану кезінде *түсініктемелерді* қолдануға кеңес беріледі, бағдарламаның орындалуы кезінде бұл

түсініктемелер еленбейді. Түсініктеме – бұл пайыз (%) таңбасынан кейін орналасқан жол немесе оның бөлігі. Түсініктеме жеке жолдан да тұруы немесе операторлардан кейін де орналасуы мүмкін.

Бірінші операторға немесе бірінші бос жолға тиісті түсініктемелер анықтамалық жүйе ретінде қолданылады. Түсініктеменің бірінші жолы келесідей командамен шығарылады.

lookfor түйінді сөз және *help* каталог атауы

Ал

help функция атауы

командасы бос жолда немесе бірінші операторда орналасқан түсініктеменің барлық жолдарын экранға шығарады.

3.1 Сценарий

Сценарий көп ретті орындауды қажет ететін жұмыстарды автоматтандыру үшін арналған, яғни оларды қолдану командалық жолға команданы қайтара енгізу қажеттілігінен құтқарады.

Сценарийдің кіріс және шығыс аргументтері болмайды. Сценарийдің орындалуы кезінде оның командасына жұмыс аймағындағы деректер қолжетімдік дәрежеде болады. Егер сценарийдің орындалуы кезінде айнымалылар құрылса, олар жұмыс аймағына орналастырылады және сценарийді орындау аяқталғаннан кейін де сонда қалады. Осылайша, сценарий тек жұмыс кеңістігінің айнымалыларымен ғана жұмыс істейді және өзінің жеке айнымалылары болмайды. Сондықтан да, бірнеше сценарийлердің орындалуы кезінде олардың ортақ айнымалыларының сәйкестігіне назар аудару керек.

Сценарийді орындау үшін командалық жолда сақталған файлдың атын енгізіп және

Enter-ді басу немесе *run* файлдың атауы

командасын қолдану керек.

Мысалы, *ex1.m* файлында сақталған бағдарламаны қосу үшін командалық жолға *ex1* немесе *run ex1* командасын енгізу қажет. Сондай-ақ бағдарлама редактор/түзеткіш терезесінде де іске қосылуы мүмкін.

М-файлды командалық жолдан шақырған кезде келесідей іс-қимылдар жүзеге асырылады:

1. Жұмыс кеңістігінде аттас айнымалының бар-жоғы тексеріледі.

2. Егер жұмыс кеңістігінде айнымалы болмаса, берілген атпен аттас функцияны іздеу орындалады.

3. Егер алдыңғы адымдар оң нәтиже бермесе, онда енгізілген атпен М-файлды іздеу орындалады: алдымен іздеу ағымдағы жұмыс каталогында, содан соң іздеу жолында орнатылған каталогтарда орындалады.

4. Егер файл табылса, ол орындалады. Ал егер ол орындалмаса, онда қате туралы хабарлама шығады.

1-мысал. Бастапқы матрицаның бірінші бағанының максималдық және минималдық элементтерінің өзара қайта құрылуы үшін сценарий құру керек.

Сценарий мәтіні:


```
a=[3 4 5;6 4 2;9 7 5] % алғашқы матрица  
[imax,imax]=max(a(:,1)); % бірінші бағанның максималдық  
элементін анықтаймыз  
[imin,imin]=min(a(:,1)); % бірінші бағанның минималдық  
элементін анықтаймыз  
  
% өзара қайта құруды орындаймыз  
% бірінші бағанның максималдық және минималдық элементі  
v= a(imax,1);  
a(imax,1)=a(imin,1);  
a(imin,1)=v;  
  
a % қайта құрудан кейінгі матрица
```

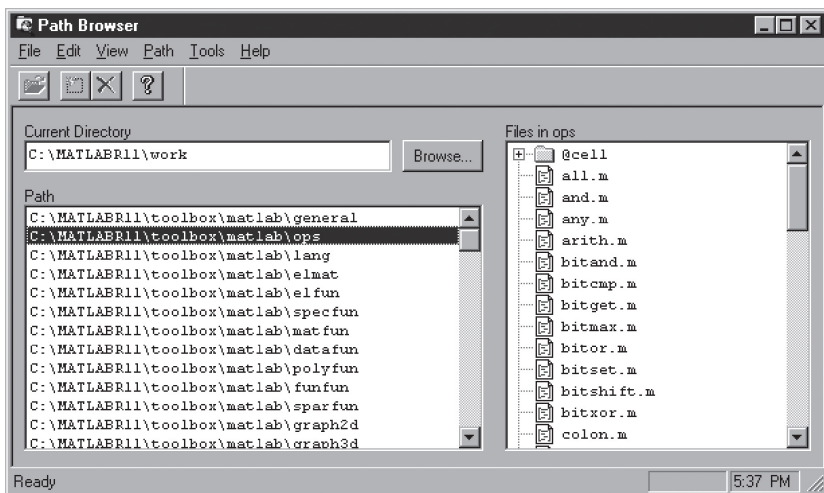
3.2 Іздеу жолының қондырғысы

Орындауға тиісті болатын кез келген файл ағымдағы немесе іздеу (ену) жолының тізімінде көрсетілген каталогта орналасуы керек. Әйтпесе, талап етілетін файл табылмайды. Сондықтан да, егер файл іздеу жолы тізімінде көрсетілмеген каталогта сақталған болса, онда осы каталогқа жолды іздеу жолының тізіміне енгізу қажет.

Келесі командалар іздеу жолының тізімімен жұмыс істеуге арналған:

<i>Path</i>	Іздеу жолының тізімін экранға шығару
<i>addpath</i> іздеу жолы	Каталогты іздеу жолы тізіміне енгізу, мысалы: <i>addpath 'c:\matlab_w'</i>
<i>rmpath</i> іздеу жолы	Каталогты іздеу жолы тізімінен өшіру, мысалы: <i>rmpath 'c:\matlab_w'</i>

Айтылған командалардан басқа да іздеу жолын қараудың ыңғайлырақ құралы – *Path Browser* бар, бұл терезе *File/ Set Path...* командасымен немесе 3.1–суретте келтірілгендей командалық терезенің  *Path Browser* батырмасымен шақырылады.




3.1–сурет. *Path Browser* терезесі

Path Browser іздеу жолын қарауға, өшіруге және тағайындауға, олардың тізімдегі реттілігін өзгертуге, MATLAB-тың барлық файлы қарауға мүмкіндік береді.

Current Directory жолы ағымдағы каталогты бейнелеу және өзгертуге арналған (жолдың оң жағындағы *Browse* батырмасы). *Path* жолы ену жолы тізімінен тұрады. *Files in* Path жолында белгіленген каталогтың аты жолы файлдар тізімі мен белгіленген каталогтың ішкі каталогтарынан тұрады. *Path Browser* мәзірінің командасы мен оларға сәйкес келетін құрал-саймандар жақтауындағы батырмалар 3.1–кестеде келтірілген.

3.1–кесте. Path Browser мәзірінің командалары

Мәзір	Команда	Құрал-саймандар жақтауының батырмалары	Қызметі
<i>File</i>	<i>Open</i>		Файлды ашу
	<i>Save Path</i>		Келесі жұмыс сеансы үшін ену жолын сақтау.
	<i>Exit Path Brouser</i>		<i>Path Brouser</i> -ден шығу
<i>Edit</i>	<i>Undo</i>		Соңғы амалды болдырмау
	<i>Redo</i>		Соңғы амалды қайтару
<i>View</i>	<i>Toolbar</i>		Құрал-саймандар жақтауын шығару
	<i>Status Bar</i>		Күй тақтасын шығару
	<i>Editor Debugger</i>		М-файл редакторының терезесін ашу
	<i>Path Browser</i>		<i>Path Browser</i> -ды көрсету
	<i>Workspace Browser</i>		<i>Workspace Browser</i> терезесін ашу
<i>Path</i>	<i>Add to Path</i>		Каталогты іздеу жолы тізіміне қосу
	<i>Remove from Path</i>		Каталогты іздеу жолы тізімінен алып тастау
	<i>Refresh</i>		Іздеу жолын жаңарту (командалық терезедегі командамен өзгерту енгізгеннен кейін)
	<i>Restore Defaults</i>		Үнсіздікпен қабылданып қойылғандарды қалпына келтіру
<i>Tools</i>	<i>Run</i>		М-файлды орындау
	<i>Customize</i>		Күйге келтірулерді таңдау
	<i>Options</i>		Қызметтерді орнату
	<i>Font</i>		Қаріпті орнату
<i>Help</i>	<i>About Matlab Path Browser</i>		<i>Path Browser</i> туралы ақпараттан қорытынды

Path аймағында каталогты басқа іздеу жолы тізімі позициясына орын ауыстыру үшін оны тышқанның сол жақ батырмасымен ұстап тұрып жылжыту қажет.

3.3 Функциялар

Функциялар белгілі бір тапсырманы шешетін бағдарламаның оқшауланған фрагментін іске асыру үшін арналған. Олар кіріс және шығыс аргументтерінен тұруы мүмкін (параметрлері бар және бір немесе бірнеше мәндерді қайтара алады). Функция параметрлері мен қайтарылу мәндері әр типті, әр түрлі жиымдар болуы мүмкін. Сценарий сияқты функциялар да MATLAB командалары мен операторларынан тұрады және *.m* кеңейтілуі бар тексттік файлдарда сақталады. Функция сақталатын М-файлдың атауы функция атауымен сәйкес келу керек. Функция MATLAB-қа ену тізіміне кіретін каталогтардың бірінде орналасуы тиіс.

Функция командалық жолдан, басқа функциялардан немесе сценарийлерден шақырылуы мүмкін. Егер функция қайтаратын шаманың типі бір өрнекке сәйкес болатындай болса, оның шақырылуы сол өрнекке енуі мүмкін.

Функция мәтіні (оның анықтамасы) функцияның тақырыбынан басталады, одан кейін функцияның денесі болады. Тақырып функцияның интерфейсін анықтайды және келесідей болады:

function [*y1, y2, y3, ...*] = функция атауы (*x1, x2, x3, ...*)

мұндағы *y1, y2, y3, ...* - қайтарылатын мәндер; *x1, x2, x3, ...* - функция параметрлері.

Мысалы, бір *a* параметрі мен үш *x, y, z* қайтарылатын мәндері бар *F1* атты функцияның мәні келесідей түрге ие:

function [*x,y,z*] = *F1(a)*

a, b екі параметрлі және *x* бір қайтарылатын шамасы бар *F2* функциясының тақырыбы мына түрде көрініс табады:

function *x* = *F2(a,b)*

a, b, c үш параметрі бар және ешқандай қайтарылатын шамасы жоқ *F3* функциясы:

function *F3(a,b,c)*

F4 функциясының параметрлері де, қайтарылатын мәндері де жоқ болса, онда

function *F4*

Функция MATLAB операторлары мен командалардан тұрады. Функцияда қайтау айнымалыларының меншікті мәндері болуы тиіс (кері жағдайда, функцияны шақыртқан кезде сәйкес хабарлама шығады).

Бірінші бос жолға дейінгі немесе бірінші операторға дейінгі функцияның тақырыбынан кейінгі орналасқан түсініктеме

`help` [функция атауы]

командасымен шығады. Осы түсініктеменің бірінші жолы сценарий түсініктемесінің бірінші жолында қолданылады.

Бірнеше мәнді ($Y1, Y2, Y3, \dots$ айнымалыларына меншіктелетін) қайтаратын функцияны шақырту келесідей түрде жазылады:

[$Y1, Y2, Y3, \dots$] = [функция атауы] ([параметрлер тізімі үтір арқылы])

Мысалы, тақырып алдында қарастырылған $F1, F2, F3, F4$ функцияларының шақырылуын жазу керек болсын. Осы кезде қайтатын мәнге ие болатын айнымалылар атауы мен параметрлерінің мәнін көрнекілік үшін бас әріптермен жазылады:

```
[X,Y,Z]=F1(A)
X=F2(A,B) немесе F2(A,B) – өрнекте
F3(A,B,C)
F4
```

2-мысал. Үлкен сандармен берілген матрица элементтерінің санын анықтау үшін функция құру. Осы элементтердің индекстерін анықтау.

Функцияны анықтау мына түрде болады (`I3_p2.m` файлында сақталған):

```
function [k, ind]=fk(A,Y)
%Y үлкен сандармен берілген A матрицасының элементтерінің
санын анықтау
k=sum(sum(A>Y));%санды анықтау
[i,j]=find(A>Y);%, Y үлкен сандармен берілген A матрица
элементтерінің индексі
ind=[i,j];%индексті матрицаға жазамыз (бірінші баған - жол,
ал екінші- баған нөмірі)
```

A матрицасын құрамыз да командалық жолдан функцияны шақырамыз:

```
» A=[6 8 2;5 9 12;4 2 78]
A =
     6     8     2
     5     9    12
     4     2    78
```

```

» [g,h]=l3_p2(a,7)
g =
    4
h =
    1    2
    2    2
    2    3
    3    3
»

```

М-файлда бірнеше функцияның анықтамасы болуы мүмкін. Осы функциялардың біреуі негізгі файлдағыдай атқа ие болуы керек. Осы кезде (анықтамасы басқа файлда орналасқан сценарий немесе функцияның командалық жолынан) тек негізгі функция шақырылуы мүмкін. Қалған функциялар (олар көмекші немесе ішкі функциялар деп аталады) негізгісімен немесе бір-бірімен шақырылуы мүмкін.

3-мысал. Кесінді ұзындығы дюйм ($1\text{дюйм}=2,54\text{см}$) арқылы берілген. Ұзындық мәнін метрлік жүйеге, яғни оны метр, сантиметр, миллиметрге ауыстыру керек. Мысалы, $21\text{дюйм}=0\text{м } 53\text{см } 3,4\text{мм}$.

Есепті шешу үшін үш функция құрамыз: *pr* атты негізгі және екі көмекші *D* (дюймді миллиметрге ауыстыруға) және *M* (миллиметрді метр, сантиметрге ауыстыру үшін). М-файлдың *pr.m* тексті *pr*, *D* және *M* функцияларымен:

```

function [m, sm, mm]=pr(d)
%pr- дюймдағы кесінді ұзындығын метр, сантиметр, милли-
метрге ауыстыру
mm=D(d);
[m,sm,mm]=M(mm);

function mm=D(d)
%дюймді миллиметрге ауыстыру
mm=d*25.4;

function [m,sm,mm]=M(mm)
%миллиметрлерді метр, сантиметр, миллиметрге ауыстыру
m=fix(mm/1000);%метрлер

```

```
mm=rem(mm,1000);%метрдің бүтін санын алып қаламыз  
sm=fix(mm/10);%сантиметрлер  
mm=rem(mm,10);%миллиметрлер
```

Суреттеу үшін ұзындығы 121 дюйм кесіндіні метр, сантиметр, миллиметрге түрлендіреміз (*pr* функциясының шақырылуына назар аударыңыз):

```
» [m1, sm1, mm1]=pr(121)
```

```
m1 =
```

```
3
```

```
sm1 =
```

```
7
```

```
mm1 =
```

```
3.4000
```

```
» whos
```

```
Name Size Bytes Class
```

```
m1 1x1 8 double array
```

```
mm1 1x1 8 double array
```

```
sm1 1x1 8 double array
```

```
Grand total is 3 elements using 24 bytes
```

```
» help pr
```

Pr - ұзындығы дюйм кесіндіні метр, сантиметр, миллиметрге айналдыру

```
» D(12)
```

```
??? Undefined function or variable 'D'.
```

```
»
```

D көмекші функциясын шақырған кезде командалық жолдан қате туралы хабарлама шығады.

Функцияда қолданылған барлық айнымалылар жергілікті болып табылады (жұмыс кеңістігінде және басқа да функциялар үшін олар қол жетімсіз). Бұл алдыңғы мысалдағы *whos*

командасының орындалу нәтижесін суреттейді. Бұл мысалдағы бір атқа ие әртүрлі функция айнымалылары негізінен әртүрлі айнымалылар болып табылады.

Егер функцияны айтылған каталог функциясымен ғана шақыру мүмкіндігін шектеу қажет болса, онда *private* деп аталатын ішкі каталог құрылуы тиіс және файлды функциямен бірге сонда қою керек. *Private* каталогындағы барлық функциялар тек алдыңғы каталогтағы функциялармен ғана көрінеді. *Private* каталогын *File/Set Path...* командасының іздеу жолы тізіміне енгізу мүмкін емес.

Функцияны шақырған кезде айнымалыны іздеу сол атпен немесе келесі реттегі функциямен (сондай атпен алғашқы кездескен функциямен):

1. Құрылған функция.
2. Сол М-файлдағы ішкі функция.
3. *Private* ішкі каталогтағы М-функциясы.
4. Ағымдық жұмыс каталогындағы М-функция.
5. Іздеу жолында орнатылған каталогтағы М-функция (тізімдегі жоғарыдан төмен қарайғы ретпен).

Функцияны тапқан жағдайда ол орындалады, ал кері жағдайда қате туралы хабарлама шығады.

3.4 Аумақты айнымалылар

Функцияда жергілікті айнымалылар, ал сценарийде – жұмыс кеңістігінің айнымалылары қолданылады. Қажеттілік жағдайда жұмыс кеңістігінің аттас айнымалылары және бірнеше функциялар *global* айнымалылардың аттары үтір арқылы беріледі командасының көмегімен аумақтық түрде хабарландырылуы мүмкін.

Бұндай хабарлама аумақтық айнымалылар қолданылатын кез келген функцияға қойылуы тиіс, егер аумақтық айнымалы жұмыс аймағында да қолжетімдік дәрежеде болуы үшін командалық терезде жасалуы тиіс.

Мысалы, *n* аумақтық айнымалысы *r1* және *r2* функцияларын шақырудың жалпы санын санағышы ретінде қолданылсын. Онда функцияны анықтау:

```
function r1
```

```
global n
```

```
n=n+1;
```

```
function r2
```

```
global n
```

```
n=n+1;
```

сессия фрагменті:

```
global n
```

```
» n=0;
```

```
» r1
```

```
» n
```

```
n =
```

```
  1
```

```
» r2
```

```
» n
```

```
n =
```

```
  2
```

```
» r2
```

```
» n
```

```
n =
```

```
  3
```

```
» whos
```


```
Name  Size      Bytes  Class
```

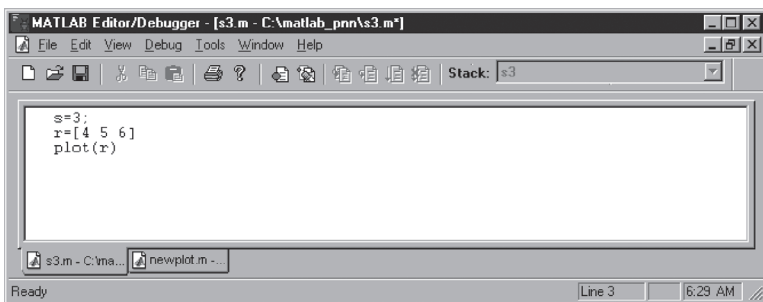
```
n      1x1      8 double array (global)
```

```
Grand total is 1 elements using 8 bytes
```

```
»
```

3.5 М-файлдың редактор/дұрыстауы. Бағдарламаны дұрыстау

М-файлдың *Editor/Debugger* редактор/дұрыстауы (3.2-сурет) М-файлды құру, түзету және дұрыстау үшін арналған. Редактор/дұрыстау командалық жолдан *edit* командасы немесе *File/New/M-file* ( *New M-file* батырмасымен) командасымен шақырылуы мүмкін.



3.2–сурет. М-файлдың *Editor/Debugger* терезесі

Егер редакторда бірнеше файл ашық тұрса, онда ол көптерезелі болады. Әрбір файл жеке қосымшаларда орналасады, олардың арасын айырып-қосу қосымшаның түбін тышқанмен түрту арқылы жүзеге асырылады. М-файл редакторының тақырып жолында белсенді файлдың атауы мен оған дисктегі жолы көрінеді. Егер аттан кейін «*» таңбасы бейнеленсе, онда бұл файл өзгеріс енгізілгеннен кейін сақталмағанын білдіреді.

File мәзірі М-файлмен жұмыс істейтін командалардан тұрады (құру, ашу, жабу, сақтау, басып шығару).


Edit мәзірінде М-файл текстi фрагменттерiн орындалған iс-қимылдан бас тарту, оны қайталау, белгілеу, көшіру, кесу және қою командалары бар. *Edit/Go To Line* командасы берілген нөмір бойынша жедел түрде жолға өтуге мүмкіндік береді (жол нөмірі күй жолының оң жақ төменгі бұрышында бейнеленеді).

View мәзірі командасы құрал-саймандар жақтауының (*Toolbar*) және күй жолының (*Status Bar*) бейнеленуін басқаруға мүмкіндік береді; *Path Browser* және *Workspace Browser* шақыру командасынан тұрады. *Evaluate Selection* командасы М-файлдың белгіленген

фрагментін орындау үшін арналған. *Auto Indent Selection* командасы автоматты орындалулар үшін қызмет атқарады.

Debug мәзірінің командалары және оларға сай келетін құрал-саймандар жақтауындағы батырмалар бағдарламаны дұрыстау үшін арналған. Олар 3.2-кестеде келтірілген.

3.2–кесте. *Debug* мәзірі командалары

Команда	Батырма	Сипаттама
<i>Continue</i>		М-файлды орындауды ол аяқталғанша немесе бірінші тоқтау нүктесіне дейін жалғастыру
<i>Single Step</i>		Ағымдағы жолды орындау
<i>Step in</i>		Ағымдағы жолды орындау, бірақ егер онда М-функциясының шақыруы болса, шақырылатын функцияның бірінші орындаушы жолына өтуін жүзеге асырады
<i>Quit Debugging</i>		Дұрыстау режимінен шығу
<i>Set/Clear Breakpoint</i>		Курсор орналасқан жердегі жолға бақылау нүктесін орнату/өшіру
<i>Clear All Breakpoints</i>		Барлық бақылау нүктелерін өшіру
<i>Stop if Error</i>		Қате бар болған жағдайда тоқтату
<i>Stop if Warning</i>		Ескерту болған жағдайда тоқтату
<i>Stop if NaN or Inf</i>		Егер нәтиже <i>NaN</i> немесе <i>Inf</i> болған кезде тоқтату

Tools мәзірі редактор, қаріп параметрлерін ретке келтіру командаларынан, *Run* – М-файлды орындау командасынан тұрады.

М-файлды дұрыстау үшін тоқтау нүктесі мен бағдарламаның адымдық орындалуы қолданылады (3.2-кесте). Тоқтау нүктесі көрсетілген жолдар сол жағынан қызыл домалақпен белгіленеді. Тоқтау нүктесі берілгеннен кейін функция (сценарий) *Tools/Run* командасының орындалуына жіберіледі.

Бағдарламаны орындау кезінде тағайындалған бірінші тоқтау нүктесінде оның тоқтауы жүзеге асырылады. Бағдарламаның орындалуы тоқтау нүктесінде тоқтатылса, онда жолдың сол жағында сары бағдар пайда болады. Оңға бағытталған бағдар бұл

жолдың келесі ретте орындалатынын білдіреді, ал төменге қарай бағытталған бағдар функцияның соңына жеткендігін білдіреді. Тоқтау нүктесінде тоқтаған кезде командалық жолда командаға енуге шақыру таңбасы >>-тан K>>-ға өзгереді.

Тоқтау нүктесінде айнымалылар мәнін, функция, өрнек параметрлерін талдауға болады. Айнымалының мәнін көру үшін бағдарлама мәтініндегі оның атының жанына курсорды алып бару жеткілікті, осыдан кейін экранда ішінде айнымалының мәні бар сары төртбұрыш шығады. Дұрыстау үдерісінде айнымалы мәні сондай-ақ командалық терезеге шығуы мүмкін. Бұл үшін айнымалыны белгілеп және *View/ Evaluate Selection* командасын қолдану керек. Бұдан басқа, бағдарламаны орындар алдында уақытша таңбаны өшіруге болады.

Дұрыстау үдерісінде айнымалылардың мәні өзгеруі мүмкін. Бұл үшін өзгерту енгізу керек жерде тоқтау нүктесін белгілеп және үзіліс кезінде командалық терезеде талап етілген мәндегі айнымалыны меншіктеуі тиіс.

Бағдарламаны орындауды *Continue*, *Single Step* немесе *Step in* командаларының бірімен жалғастыруға болады.

Мәзір және құрал-саймандар жақтауының көмегімен енгізілетін редактор/дұрыстаушы дұрыстау командаларынан басқа командалық терезенің енгізу жолында енгізілетін ұқсас командалар да бар.

4 ГРАФИК ТҰРҒЫЗУ

MATLAB-та екі немесе үш өлшемді графиктерді тұрғызуға және қалыптауға болатын құралдар бар.

4.1 Екі өлшемдік графиктерді тұрғызу

Екі өлшемді графиктер декарттық және полярлық координаттар жүйесінде сызықтық және логарифмдік масштабтарда құрылуы мүмкін.

Функция графигін тұрғызу үшін аргумент мәнінің векторын беру қажет, функция мәнінің векторын есептеп және график функциясын шақыру керек.

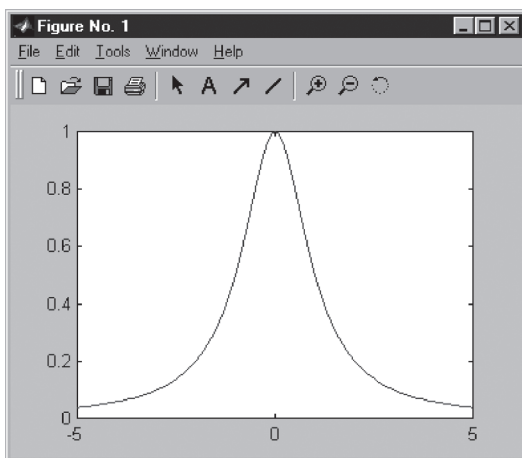
4.1.1 Декарттық координат жүйесінде график тұрғызу

Сызықтық масштабта функция графигін тұрғызу үшін $plot(X,Y)$ функциясы қолданылады. X векторында функцияның аргумент мәндерінің жиымы, ал Y векторында функция мәндерінің жиымы жазылуы керек. $Plot$ функциясы график нүктелерін түзу сызықтармен біріктіреді.

Мысалы, $[-5,5]$ аралығында $f(x) = \frac{1}{x^2 + 1}$ функциясының графигін құру керек. Ол үшін команда жолында

```
»x=-5:0.1:5;y=1./(x.^2+1);plot(x,y)
```

теріледі. *Enter*-ді басқанда экранда графигі бар терезе пайда болады. Графикті шығарған терезеге график аймағы, құрал-саймандар жақтауы және мәзір кіреді. Графиктерді тұрғызу командасын қолданғанда график терезесі автоматты түрде ашылады (4.1-сурет).



4.1-сурет. $f(x) = \frac{1}{x^2 + 1}$ функциясының графигі

Егер $plot(Y)$ функциясының аргументі ретінде бір бірөлшемді жиым болса, онда Y векторының элементтерінің индекстерінен тәуелділік құрылады. Егер Y векторының элементтері комплекстік болса, онда $plot(real(Y), imag(Y))$ графиктері құрылады. Егер Y векторы екі өлшемді жиым болса, онда баған элементтерінің жол нөмірінен тәуелділігі құрылады.

Егер $plot(X, Y)$ командасының бір немесе екі аргументі екі өлшемді жиым болса, онда келесідей құрылымдар беріледі:

- егер X жиымы бір өлшемді, ал Y - екі өлшемді болса, онда Y жиымының бағандарының X векторының элементтерінен тәуелді графигі құрылады;

- егер Y жиымы бір өлшемді, ал X - екі өлшемді болса, онда Y векторының элементтерінің X жиымының бағандарынан тәуелді графигі құрылады;

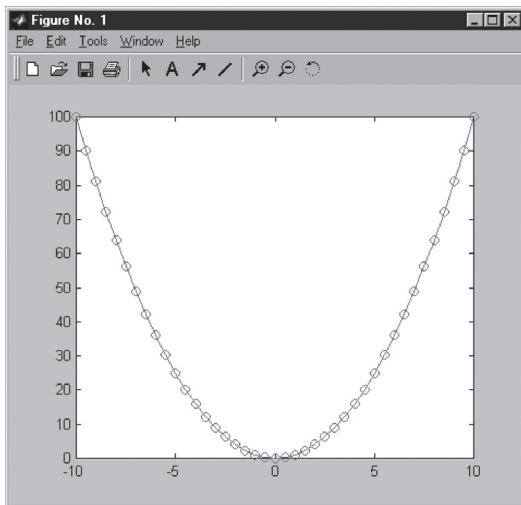
- егер X және Y жиымдары екі өлшемді болса, онда Y жиымының бағаны X жиымының бағанынан тәуелді графигі құрылады.

$Plot(X, Y, LS)$ командасы LS жолдық айнымалының көмегімен (дәйекшеге алынады) сызықтарды белгілеуге, түсін анықтауға және нүкте маркерін беруге мүмкіндік береді.

Мысалы,

» $a=-10:0.5:10; plot(a, a.^2, 'ro-')$

командасы $[-10,10]$ аралығында дөңгелек маркермен және тұтас қызыл сызықпен параболаны салуды береді (4.2-сурет).



4.2-сурет. Сызық қалыбын беру

4.1-кесте. Сызықтардың параметрлері

Сызықтың типі	символ	Сызықтың түсі	символ	Маркер типі	символ
Біртекті	-	жасыл	y	нүкте	.
үзік	--	күлгін	m	қосу	+
пунктирлі	:	көгілдір	c	жұлдызша	*
үзік-пунктирлі	-.	қызыл	r	дөңгелек	o
маркер бар болғандағы символ жоқ	сызық жоқ	жасыл	g	крест	x
		көк	b	квадрат	s
		ақ	w	ромб	d
		қара	k	бесбұрыш	p
				алтыбұрыш	h
				солға бағыттауыш	<
				оңға бағыттауыш	>

СЫЗЫҚТЫҢ ТИПІ	СИМВОЛ	СЫЗЫҚТЫҢ ТҮСІ	СИМВОЛ	Маркер типі	СИМВОЛ
				<i>төменге бағыттауыш</i>	□
				<i>жоғарыға бағыттауыш</i>	^

Plot(X1,Y1,LS1, X2,Y2,LS2, ...) командасы әртүрлі параметрлі сызықтармен графиктер тобын салуды орындайды.

Plot(X,Y,LS, 'PN',PV ...) командасы график тұрғызуда қолданылатын сызықтың қасиеттерін береді (*'PN'* – қасиеттің атауы, *PV* – қасиеттің берілген мәні). Мысалы, алдыңғы мысалдағы сызықтың жуандығына 3 мәнін беру үшін берілетін команда

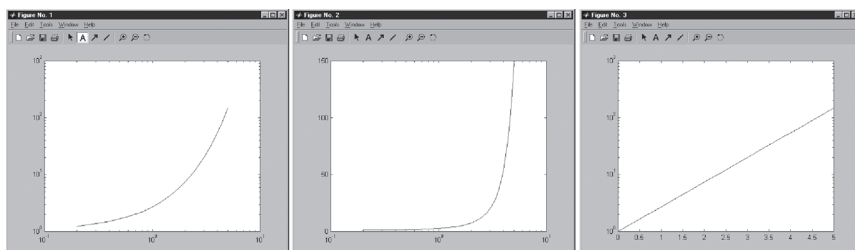
```
»a=-10:0.5:10;plot(a,a.^2,'r-h','LineWidth',3)
```

түрде болады.

Екі координаттық осьте логарифмдік масштабта график салу үшін *plot* функциясының орнына бұрынғы аргументтер жиынтығымен *loglog* функциясы қолданылады. *X* осі бойынша логарифмдік масштабта және *Y* осі бойынша сызықтық масштабта функция графикін тұрғызу *semilogx* функциясы арқылы орындалады. Ал *semilogy* функциясы *Y* осі бойынша логарифмдік масштабты және *X* осі бойынша сызықтық масштабта функция графикін тұрғызады. Мысалы:

```
»x=0:0.2:5;loglog(x,exp(x))
»figure;
»semilogx(x,exp(x))
»figure;
»semilogy(x,exp(x))
»
```

Нәтижелері 4.3-суретте көрсетілген.

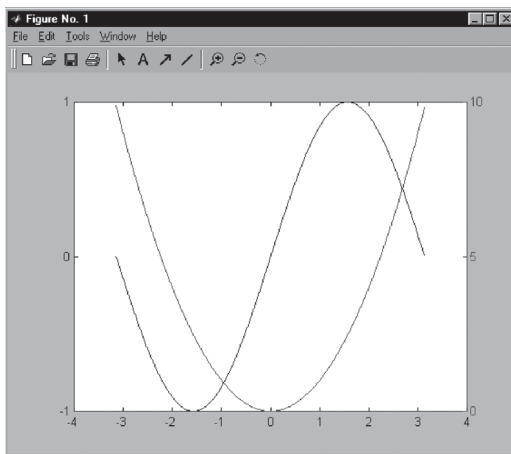


4.3-сурет. Логарифмдік масштабта тұрғызылған функция графигі



Plotуу ($x1,y1,x2,y2$) функциясы $x1,y1$ және $x2,y2$ жиымдарымен берілген екі координат осі бар екі функция графикін тұрғызады. (сол жағындағы ось бірінші графикке, ал оң жағында екінші графикке сәйкес келеді). Егер $y1, y2$ – екі өлшемді жиым болса, графиктер саны ондағы бағандардың санына тең болады. Мысалы, $-\pi$ -ден π -ге дейінгі аралықта синусоида және парабола тұрғызайық.

```
»x=-pi:0.01:pi;plotуу(x,sin(x),x,x.^2)
```

Нәтижелері 4.4-суретте көрсетілген.



4.4-сурет. Екі координаттық осьтері бар графиктер

Графиктік терезенің құрал-саймандар жақтауындағы **A**,  және  пернелері батырмалары графиктік аймаққа мәтін, бағыттауыш және сызықтар қосу үшін арналған. График тұрғызылғаннан кейін графиктерді және оның құрама бөліктерін жөндеу, осьтер мен сызықтардың параметрлерін беру *Edit* және *Tools* командалары арқылы беріледі. Графиктік терезені толық сақтау үшін *File/Save* командалары қолданылады, ал қолданыстағы файл *.fig* кеңейтілуі арқылы жазылады. Графиктік терезені ашу *File/Open* командасымен іске асырылады.

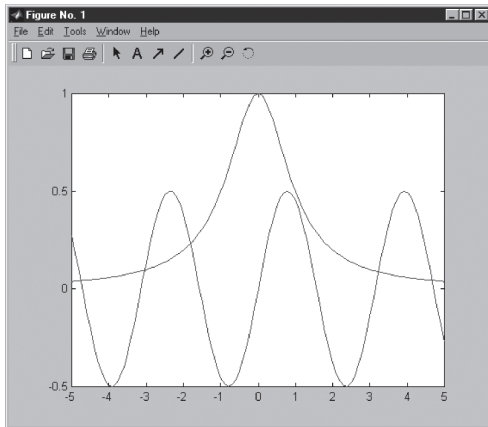
Графиктік терезені белгілі бір графикалық қалыпта сақтау үшін оның кеңейтілімін көрсету керек. Графиктерді экспорттау үшін *File/Save* және *File/Save As* командаларынан басқа *File/Export* командасын қолдануға болады, алайда «Тип файла» тізімінде сақтау кезінде керекті қалыпты таңдау керек.

4.1.2 Бір терезеде бірнеше график тұрғызу

Графикті тұрғызудың кезекті командасын бергенде терезеде бұрынғысы жойылып, жаңа график пайда болады. Бір терезеде бірнеше график тұрғызу үшін *hold on* (алдыңғы графикті сақтау режимін іске қосады) командасы қолданылады. Бұрынғы графикті сақтау режимін алып тастау үшін *hold off* командасы қолданылады (алдыңғы графикті сақтау режимін тоқтатады). *Hold* командасы бір режимнен екінші режимге өтуді жүзеге асырады. Мысалы:

```
»x=-5:0.1:5;y=1./(x.^2+1); plot(x,y); hold on; plot(x,0.5*sin(2*x)); hold off
```

График тұрғызудың нәтижелері 4.5-суретте көрсетілген.

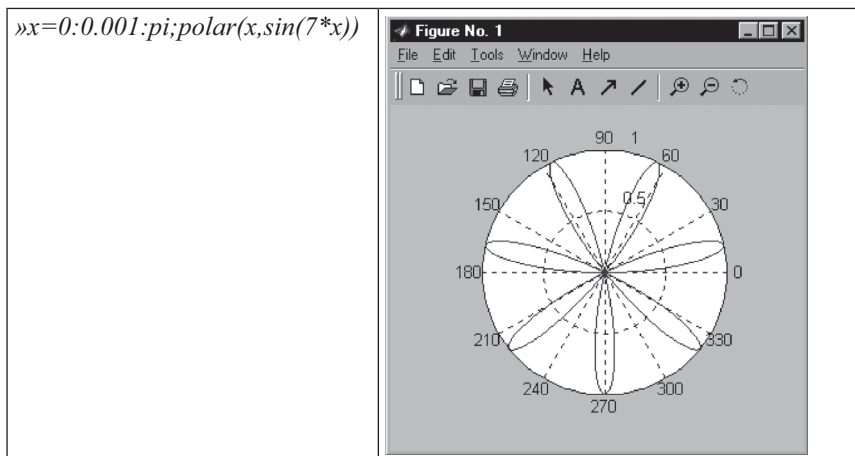


4.5-сурет. *Hold on* командасы арқылы бір терезеге бірнеше график тұрғызу

4.1.3 Полярлық координаттар жүйесінде график тұрғызу

Мәндері F және R жиымдарына жазылатын полярлық бұрыш пен полярлық радиус арқылы берілген функцияның графигін полярлық координаттар жүйесінде тұрғызу $polar(F,R,LS)$ функциясын қолдану арқылы іске асырылады. Міндетті емес жолдық айнымалы LS сызықтың қасиетін береді ($plot$ функциясындағыдай). Егер F және R жиымдары бірдей мөлшердегі екі өлшемді жиымдар болса, онда графиктер олардың әрқайсысының бағандары бойынша салынады.

Мысалы, $(0, \pi)$ аралығында $\rho(\varphi) = \sin(7\varphi)$ функциясының графигін тұрғызу керек.



4.6.1-сурет. $\rho(\varphi) = \sin(7\varphi)$ функциясының графигі

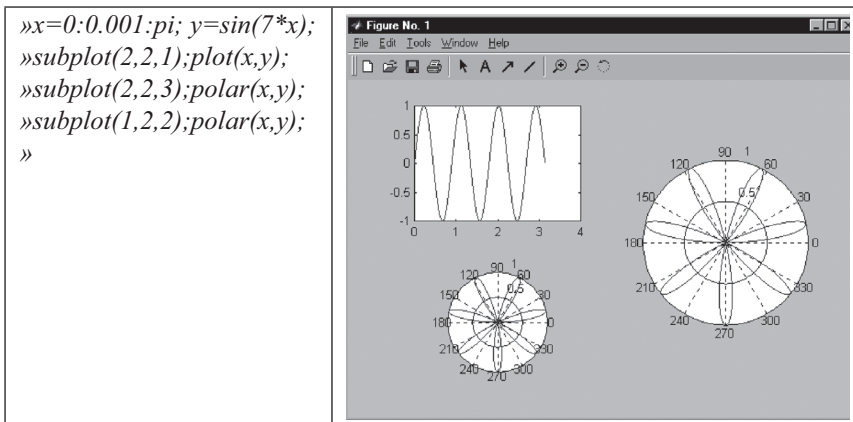
4.2 Графиктік терезелермен жұмыс

Графиктік терезеге өту үшін немесе жаңа графиктік терезе ашу үшін *figure(n)* командасы қолданылады. Мұндағы *n* графиктік терезенің нөмірі. Егер берілген нөмірдегі графиктік терезе жоқ болса, онда жаңа терезе пайда болады. Егер нөмірмен берілген терезе бар болса, қалған графиктік командалар сол терезеге қатысты болады. Параметрсіз *figure* командасы ең кіші нөмірдегі жаңа терезені ашады.

Close(n) немесе *close n* командасы *n* нөмірлі графиктік терезені жабады. *Close all* командасы барлық ашық тұрған графиктік терезелерді жабады.

4.3 Графиктік терезелердің шағын терезелерге бөлінуі

Графиктік терезені бірнеше терезелерге бөлуге болады және олардың әрқайсысына әр түрлі функциялардың графиктерін тұрғызуға болады. Терезені жалпы саны тігінен *n*, ал көлденеңінен *m* болатын шағын терезелерге бөлу үшін және нәтижені реті бойынша *k*-ші терезесіне шығару үшін (егер солдан оңға және жоғарыдан төмен қарай санасақ) *subplot(n, m, k)* функциясы қолданылады. Мысалы:



4.6.2-сурет. Графиктік терезені бірнеше терезелерге бөлу

4.4 Үш өлшемді графиктің тұрғызылуы

Кеңестіктегі сызықтарды құру үшін $plot3(X,Y,Z)$ функциясы қолданылады.

Бұл функция $plot$ функциясының үшөлшемді аналогы болып табылады және X , Y , Z жиымдарында жазылған абсцисса, ордината және аппликата нүктелерін тік сызықтармен қосу үшін арналған.

Ал келесі көрініс кеңестіктегі қисық сызықтың пайда болуына алып келеді (4.7-сурет).

```

»z=0:0.01:70;x=(1-z/70).*cos(z);y=(1-z/70).*sin(z);plot3(x,y,z)

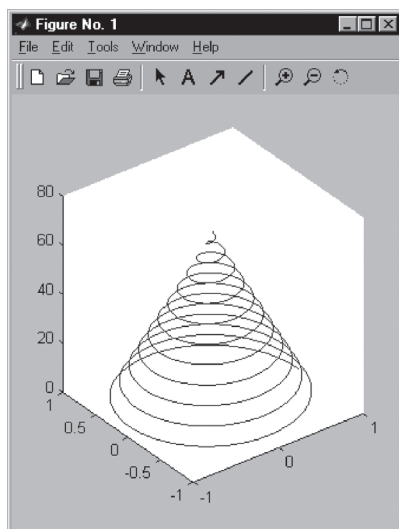
```

Екі айнымалысы бар функция графигін тұрғызу үшін:

1. $[X,Y]=meshgrid(x,y)$ функциясының көмегімен абсциссасы x және ординатасы y болатын графиктің әрбір түйіндік нүктелеріндегі мәндер арқылы X , Y матрицалары құрылады. Сонымен бірге матрицаның әр X жолы x векторының көшірмесі, ал матрицаның әр Y бағаны y векторының көшірмесі болады.

2. Әрбір түйіндік нүктеде екі айнымалысы бар функция мәндерінің Z матрицасы есептелінеді.

3. Үш өлшемді графиктер тұрғызу функцияларының біреуі қолданылады (4.2-кесте).



4.7-сурет. *Plot3* функциясын қолдану арқылы құрылған кысық

4.2-кесте. Үш өлшемді графиктерді тұрғызу функциялары

Функция	Функцияның сипаты
$mesh(x, y, z)$	түйіндік нүктелердің координаттары x, y, z матрицаларының құрамында болатын беткі қабаттың қаңқасын тұрғызу
$meshc(x, y, z)$	XU жазықтығына проекцияланатын функцияның деңгей сызығымен шектелген беттің қаңқасын тұрғызу
$meshz(x, y, z)$	XU жазықтығына перпендикуляр проекцияланатын жазықтықтың шеткі нүктелерімен шектелген қаңқалы бетті тұрғызу
$surf(x, y, z)$	түйіндік нүктелердің координаттары x, y, z матрицаларының құрамында болатын боялған бетті тұрғызу
$surfz(x, y, z)$	XU жазықтығына проекцияланатын боялған бет пен функцияның деңгей сызықтарын тұрғызу
$contour(x, y, z)$	түйіндік нүктелердің координаттары x, y, z матрицаларының құрамында болатын беттік деңгей сызықтарын тұрғызу. Егер $contour$ функциясы қайтаратын параметрді $clabel$ функциясына берілсе, яғни $clabel(contour(x, y, z))$, онда графиктегі деңгей сызықтарына функция мәндері түседі.
$contourf(x, y, z)$	аралықтарын әр түрлі түстермен бояп беттің деңгей сызықтарын тұрғызу
$contour3(x, y, z)$	кеңістіктегі беттің деңгей сызықтарын тұрғызу

Тұрғызылған графиктің сырт пішінін басқару үшін 4.3-кестеде келтірілгендей әр түрлі функциялар қолданылады.

4.3-кесте. Графиктің сырт пішінін басқару үшін арналған функциялар

Функция	Функцияның сипаты
<i>hidden on (hidden off)</i>	кестедегі көрінбейтін сызықтарды алып тастауды қосу (ажырату)
<i>shading</i>	беттерді көлеңкелеудің келесі тәсілдерінің бірін орнату: <i>shading faceted</i> - қара қырлы ұяшықтарының бір қалыпты боялуы; <i>shading flat</i> - ұяшықтар немесе қырлардың тор түйінінің түсімен тәуелділікте боялуы; <i>shading interp</i> - тор түйіндеріндегі түс интерполяциясымен анықталатын түспен боялуы
<i>view</i>	шолу нүктесін басқару: <i>view(A,E)</i> - E дөңестік бұрышын және A азимутын орнатады (<i>азимут</i> - Z осін айналдыра Y осінің теріс бағытында сағат тіліне қарсы айналдыру бұрышы, <i>дөңестік бұрыш</i> - координата басынан шыққан шолу нүктесіне дейінгі кесіндімен XY жазықтығы аралығындағы бұрыш); <i>view([x,y,z])</i> шолу нүктесін (x, y, z) координаталарымен береді; <i>view(2)</i> , <i>view(3)</i> - шолу нүктесі үшін екі өлшемді және үш өлшемді графиктерінің кестелер бойымен қабылданған нүктелерінің орнын көрсетіп отырады
<i>colorbar</i>	Z осінің шкаласы мен ағымдағы түс түрінің тік бағанын шығару
<i>colormap(MN)</i>	түс түрін MN атты түс түріне ауыстыру. Стандартты және қосымша түс түрлерінің атаулары: <i>cool</i> (көгілдір мен қарақошқыл рең), <i>gray</i> (сұр рең), <i>hot</i> (ақ-сары-сарғылт-қызыл-қара), <i>hsv</i> (кемпіркосақ түсі), <i>jet</i> (қызыл-сары-жасыл-көгілдір-көк), <i>copper</i> (мыс реңі), <i>autumn</i> (сары-сарғылт-қызыл), <i>winter</i> (көк-жасыл), <i>summer</i> (сары-жасыл), <i>spring</i> (сары-қарақошқыл), <i>pink</i> (қоңыр рең). Үнсіздікпен қолданылатын түс түріне қайтып келу <i>colormap('default')</i> командасы арқылы іске асырылады
<i>Camlight</i>	жарық көзін жасау: <i>camlight(A,E)</i> – берілген E дөңестік бұрыштық және A азимуттық координаталарында; <i>camlight headlight</i> – шолу нүктесінде; <i>camlight right</i> – шолу нүктесінен жоғары және оңға; <i>camlight left</i> – шолу нүктесінен жоғары және солға

Функция	Функцияның сипаты
<i>Material</i>	Беттік материалдың қасиетін беру: <i>material shiny</i> – жылтыр; <i>material dull</i> – бұлыңғыр; <i>material metal</i> – металл түстес; <i>material default</i> – үнсіздікпен қабылдану
<i>brighten(B)</i>	Түс түрлерінің айқындылығын өзгерту: $0 < B < 1$ болғанда айқындылық артады, $1 < B < 0$ болғанда айқындылық кемиді

Үшөлшемдік график тұрғызудың мүмкіндіктерін екі айнымалысы бар $z(x, y) = \sin(x) * \sin(y)$ функциясының графигін салу арқылы көрсетуге болады. Ол үшін

```

»x=-pi:0.1:pi; y=x; [x1, y1]=meshgrid(x,y); z=sin(x1).*sin(y1);
»subplot(2,2,1); mesh(x1, y1, z);
»subplot(2,2,2); meshz(x1, y1, z);
»subplot(2,2,3); surf(x1, y1, z); view ([0,-pi,0]); colorbar; view
([0,-pi,0])
»subplot(2,2,4); surfc(x1, y1, z)
;»figure(2)
»subplot(2,1,1); clabel(contour(x1, y1, z));
»subplot(2,1,2); contourf(x1, y1, z);
»

```

командалары қолданылып, 4.8 және 4.9-суреттерде көрсетілген графиктер алынды.

4.5 Графиктік терезеге жазулар, тақырыптар және координаталық кесте шығару

График терезесінде координаталық кестені салу үшін *grid on* командасы қолданылады. Ал координаталық кестені алып тастау үшін *grid off* командасы қолданылады. *Grid* командасы бір күйден екінші күйге ауыстыру үшін қолданылады.

График аймағына жазу шығару үшін келесідей функциялар қолданылады (*SV* мәтіндік айнымалының дәйекшеге алынған түріндегі функцияны шақыруы берілсе):

- xlabel(SV)* – x осінің атауы;
- ylabel(SV)* – y осінің атауы;
- zlabel(SV)* – z осінің атауы;

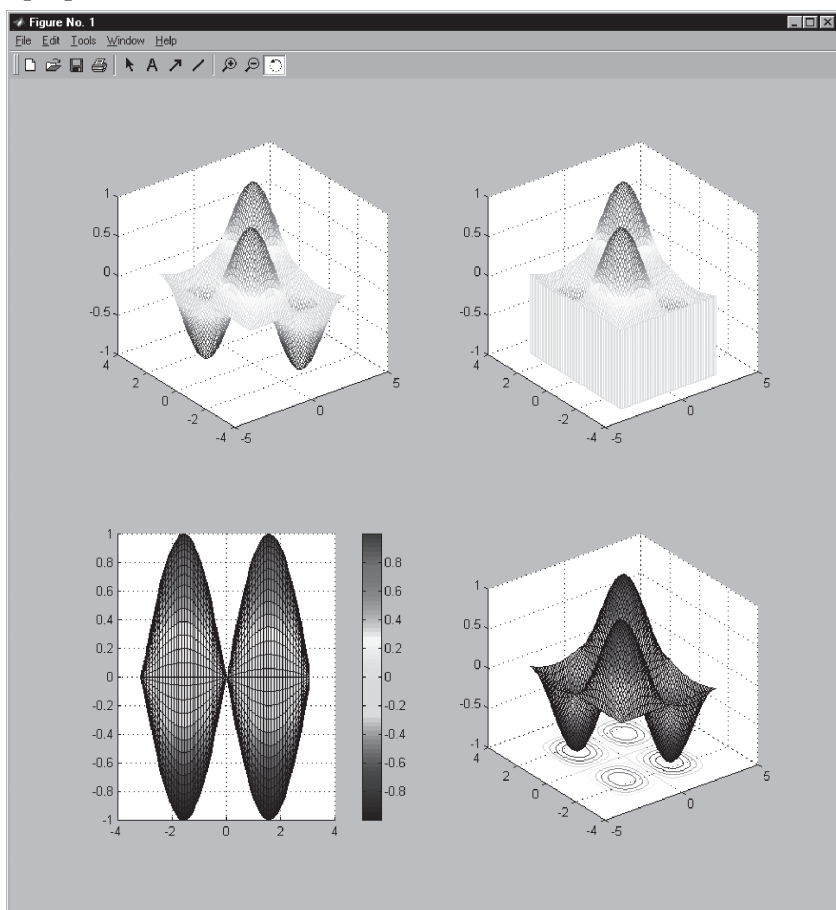
$title(SV)$ – графиктің үстіңгі жағына орналасқан тақырыбы;

$text(X,Y,Z,SV)$ - (X,Y,Z) координаттық нүктеден басталатын мәтін.

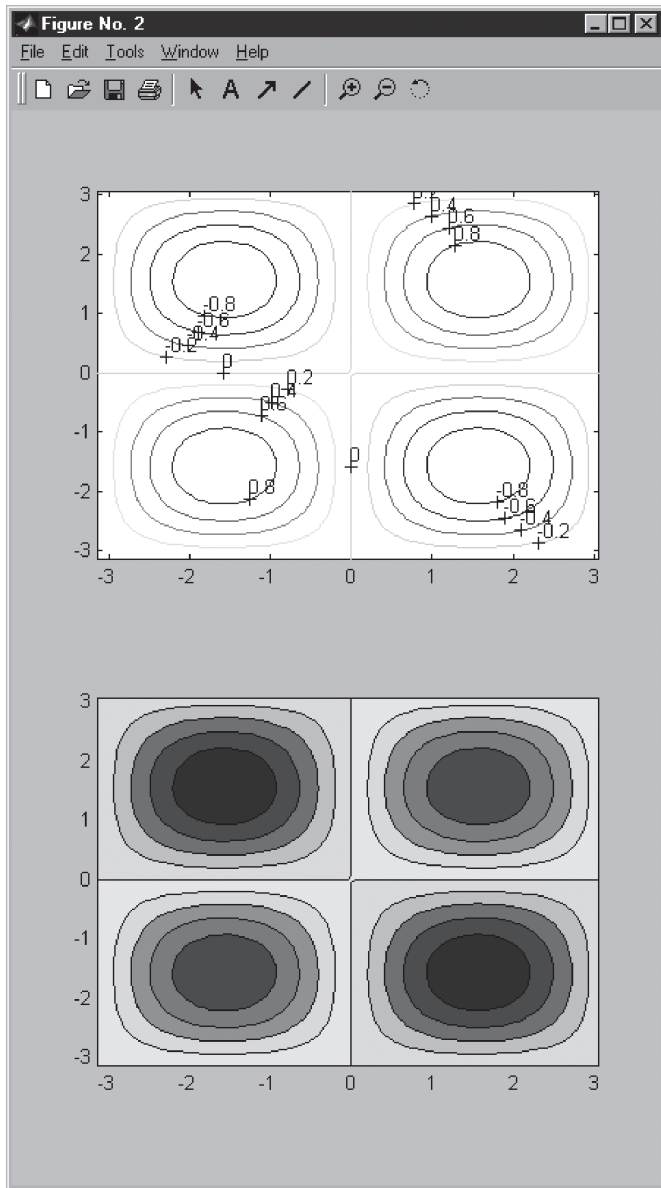
Нүкте координатасы сол график тұрғызылған координаттар жүйесінде беріледі. Жазықтықтағы график үшін Z координатасы болмайды;

$gtext(SV)$ – SV жолының шығар орнын көрсететін белгіні шығару командасы. Енгізуді тоқтату үшін кез келген пернені басып немесе тышқанмен шерту керек.

$legend(SV1,SV2,SV3 \dots pos)$ – график аймағына әр графикке, сызықтың немесе маркердің түрі және түсініктеме беретін мәтіндік $SV1, SV2, SV3$ айнымалылары түрінде жазылған түсініктемелік мәтіні бар әфсана легенданы қосады.



4.8-сурет. $z(x,y) = \sin(x) * \sin(y)$ функциясының графиктері



4.9-сурет. $z(x,y) = \sin(x) \cdot \sin(y)$ функциясының графиктері

Pos параметрі арқылы әфсана легенданың орны беріледі:

<i>Pos</i> мәні	әфсана легенданың орны
-1	графиктің оң жақ сырты
0	графиктің нүктелерін әфсана легендамен неғұрлым азырақ жабатындай
1	графиктің оң жақ жоғарғы бұрышы
2	графиктің сол жақ жоғарғы бұрышы
3	графиктің сол жақ төменгі бұрышы
4	графиктің оң жақ төменгі бұрышы

Box +on (box off) командасы үшөлшемді объект орналасқан параллелепипедтің пішінін көрсету (алып тастау) үшін қолданылады. *Box* командасы бір күйден екінші күйге ауыстыру рөлін атқарады.

4.6 График масштабын өзгерту

Axis([xmin, xmax, ymin, ymax]) командасы екіөлшемді графиктің x , y координаттары бойынша, ал *axis([xmin, xmax, ymin, ymax, zmin, zmax])* – үшөлшемді графиктің x , y , z осьтері бойынша масштабты орнатады. $Xmin$, $ymin$, $zmin$ параметрлері x , y , z осі бойынша ең кіші мәндерді береді де, ал $xmax$, $ymax$, $zmax$, тиісінше, ең үлкен мәндерді береді. *Axis* командасы барлық осьтер бойынша $[xmin, xmax, ymin, ymax, zmin, zmax]$ шектік мәндері бар векторды параметрсіз кері қайтарады.

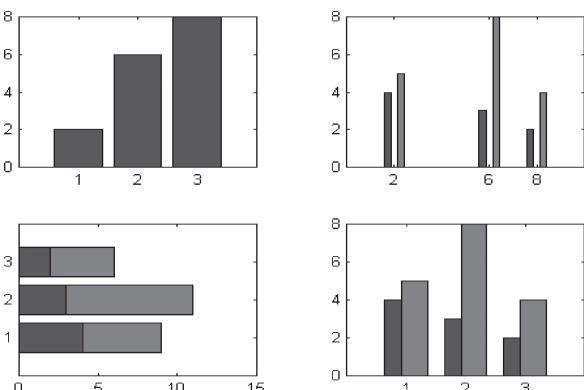
Axis('auto') командасы барлық осьтер бойынша, *axis('auto x')* командасы x осі бойынша, ал *axis('auto xy')* x және y осьтері бойынша шектік мәндердің автоматты түрде орнатылуын береді. *Axis equal* командасы барлық осьтер бойынша бірдей масштабтық коэффициент орнатады.

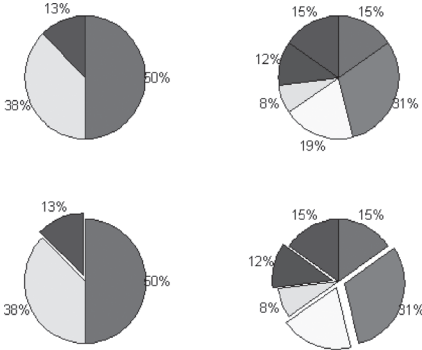
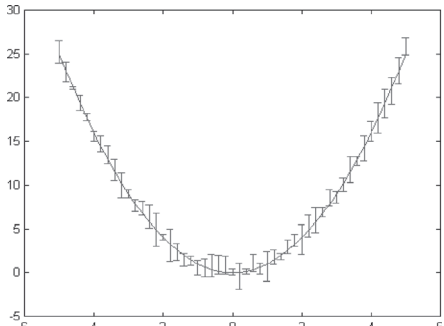
4.7 Арнайы график

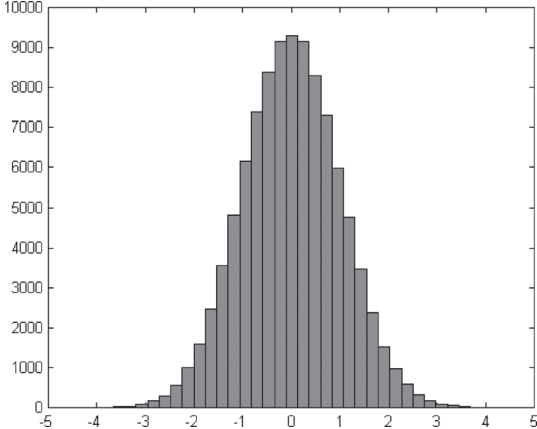
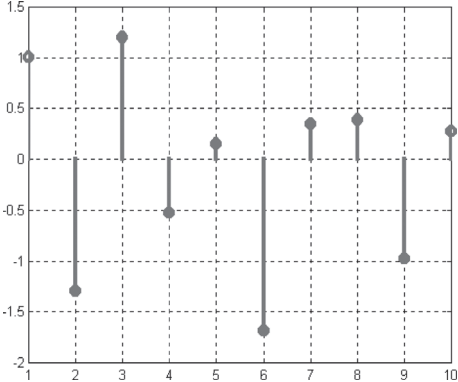
Арнайы екіөлшемді график функциялары 4.4-кестеде берілген.

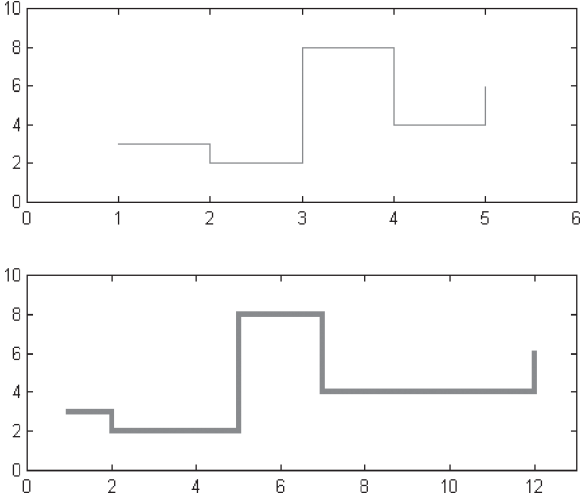
4.4-кесте. Арнайы екіөлшемді график

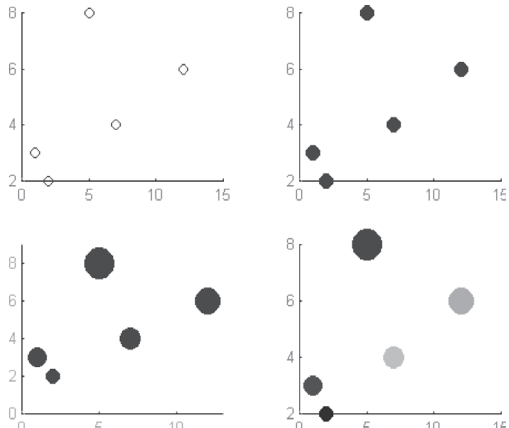
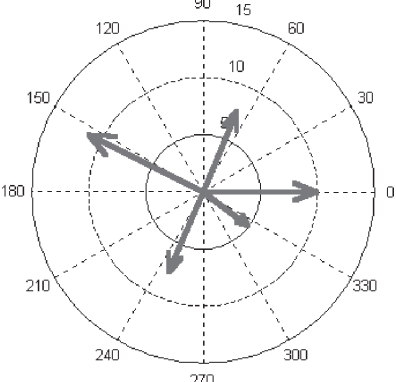
Функция	Сипаттамасы
$bar(Y,h,s)$	<p>h баған биіктігінің салыстырмалы ені (үнсіздікпен $h=0.8$) мен s стиліндегі (баған ені мен стилі функцияны шақыру кезінде берілмеуі мүмкін) тік (bar) және көлденең ($barh$) бағандық диаграммалар тұрғызу:</p> <p>$bar(Y,h,s)$ - егер Y бірөлшемді жиым болса, онда вектордың әр элементіне диаграмманың бағаны сәйкес келеді, егер Y – матрица болса, онда жол элементтерінің мәндеріне сәйкес болатын бағандар тобы шығады (ал x осі жол нөміріне сәйкес болатын белгілермен белгіленеді);</p> <p>$bar(X,Y,h,s)$ – егер Y бірөлшемді жиым болса, онда X векторының элементтерінің мәндерімен анықталатын орында Y векторының элементтері шығарылады, егер Y матрица болса, онда X векторының элементтерінің мәндерімен анықталатын орындарда Y матрицасының элементтеріне сәйкес бағандар тобы орналасады;</p> <p>мүмкін болатын стильдер: ‘group’ – диаграмма бағандар топтары түрінде (бұл стиль үнсіздікпен қолданылады), ‘stack’ – Y матрицасының әрбір жолына әр бөлігінің ұзындығы Y жиымының элементіне сәйкес келетін мәнмен анықталатын бір құрама баған сәйкес келеді.</p> <p>Мысалы:</p> <pre> »x=[2 6 8];y=[4 5; 3 8; 2 4];subplot(2,2,1);bar(x);subplot ot(2,2,2); bar(x,y,0.5) »subplot(2,2,3);barh(y,'stack');subplot(2,2,4);bar(y,1.5) </pre>
$bar(X,Y,h,s)$	

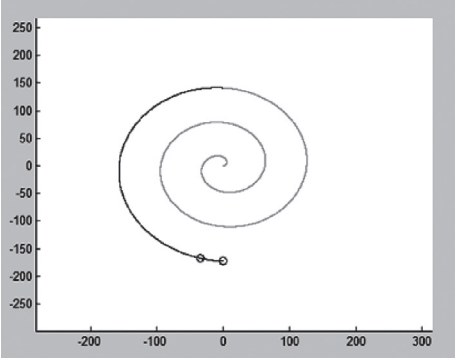


Функция	Сипаттамасы
<p><i>pie</i></p>	<p>дөңгелек диаграмма құру: $pie(x)$ – x жиымының әрбір элементін дөңгелек диаграмманың секторы түрінде бейнелейді. $pie(x, v)$ – қосымша дөңгелек диаграммада v жиымының нөлге тең емес элементтері сәйкес болатын қосымша секторлар бөлінеді (v жиымы x жиымы сияқты бір өлшемде болуы тиіс) Мысалы: $\gg x=[2\ 6\ 8]; y=[4\ 5; 3\ 8; 2]; subplot(2,2,1); pie(x); subplot(2,2,2); pie(y)$ $\gg c=[1\ 0\ 0]; d=[0\ 1; 1\ 1; 0\ 0]; subplot(2,2,3); pie(x,c); subplot(2,2,4); pie(y,d)$</p> 
<p><i>errorbar</i></p>	<p>графикті аралық қателігін көрсете тұрғызу. $errorbar(y, e)$ және $errorbar(x, y, e)$ функциялары графиктің әр нүктесіне қатысты e қателігімен тұрғызуға арналған $errorbar(x, y, L, U)$ функциясы y-ің x-тен тәуелділігінің графикін әр нүктесіне қатысты $-L$ және $+U$ қателігімен тұрғызуға арналған Мысалы: $\gg x=-5:0.2:5; y=x.^2; l=rand(1, length(y)) * 2; u=rand(1, length(y)) * 2;$ $\gg errorbar(x, y, l, u)$</p> 

Функция	Сипаттамасы
<p><i>hist</i></p>	<p>гистограмма тұрғызу. $hist(y)$ – 10 аралық үшін гистограмманы салады, $hist(y,n)$ - n аралық үшін гистограмма салады. Мысалы: $\gg x=randn(100000,1);$ $\gg hist(x,40)$</p> 
<p><i>stem</i></p>	<p>$stem(y,LS)$ - y бірөлшемді жиымының элементтерінің маркермен бітетін тік сызықтар түріндегі индекстен тәуелді графигін тұрғызу. $stem(x,y,LS)$ - y бірөлшемді жиымының элементтерінің x векторының элементтерінен тәуелділігінің графигін тұрғызу. Мысалы: $\gg x=randn(10,1);k=stem(x,'r');grid on;set(k,'LineWidth',3)$</p> 

Функция	Сипаттамасы
<i>stairs</i>	<p><i>stairs(Y,LS)</i> – ординатасы бірөлшемді Y жиымы, ал абсциссасы X болатын сатылы график тұрғызады. Строковая переменная LS жолдық айнымалы сызық параметрлерін береді;</p> <p><i>stairs(X,Y,LS)</i> - X жиымының элементтерінің мәндері сатылы графиктің абсциссасы, ал Y жиымының элемент мәндері ордината болады.</p> <p>Мысалы: <code>»x=[1 2 5 7 12];y=[3 2 8 4 6];</code> <code>»figure(2);subplot(2,1,1);stairs(y);axis([0,6,0,10]);</code> <code>»subplot(2,1,2);h=stairs(x,y);set(h,'LineWidth',3);axis([0,13,0,10]);</code></p> 
<i>scatter</i>	<p>абсцисса және ордината мәндері X және Y векторларында болатын жазықтықта нүктелер тобын құру:</p> <p><i>scatter(X,Y)</i> – көк түсті шеңбер түрінде;</p> <p><i>scatter(X,Y,S)</i> – шеңбер өлшемдері S векторы арқылы беріледі;</p> <p><i>scatter(X,Y,S,C)</i> – шеңбер түсі C векторы арқылы беріледі;</p> <p><i>scatter(..., 'filled')</i> – шеңбер боялады.</p> <p>Мысалы: <code>»x=[1 2 5 7 12];y=[3 2 8 4 6];</code></p>

Функция	Сипаттамасы
<p><i>scatter</i></p>	<pre> »figure; »subplot(2,2,1);scatter(x,y) »subplot(2,2,2);scatter(x,y,100,'filled') »subplot(2,2,4);scatter(x,y,50*y,y,'filled') »subplot(2,2,4);scatter(x,y,50*y,y,'filled') »subplot(2,2,3);scatter(x,y,50*y,'filled') »axis([0,13,0,9]) </pre> 
<p><i>compass</i></p>	<p><i>compass(Z, LS)</i> - Z жиымына жазылған комплекстік сандарға геометриялық түсінік беретін векторлар салады; <i>compass(X,Y, LS)</i> – <i>compass(X+j*Y)</i>-ке сәйкес; Міндетті емес <i>LS</i> параметрі жолдық айнымалы болады және сызықтың параметрлерін береді. Мысалы: »<i>z=[3+7j 4- 3j -10+5j -3-7j 10]</i>; »<i>h=compass(z);set(h,'LineWidth',3)</i></p> 

Функция	Сипаттамасы
<p><i>comet</i></p>	<p>Нүкте қозғалысын кометаның басы мен құйрығы түріндегі траектория бойынша бейнелейді: <i>comet(Y)</i> - траектория <i>Y</i> бірөлшемді жиымымен берілген (<i>Y</i> жиымының элементтерінің индекстері нүктенің абсцисса мәндеріне, ал элемент мәндері ордината мәндеріне тең); <i>comet(X,Y)</i> - траектория <i>X</i> және <i>Y</i> бірөлшемді жиымдармен берілген; <i>comet(X,Y,P)</i> – <i>P</i> параметрімен комета құйрығының салыстырмалы ұзындығы беріледі (комета құйрығының ұзындығы $P * length(Y)$-ге тең). Мысалы: Комета шиыршық бойынша қозғалады. »$f=0:0.001:10*pi;r=f*10; x=r.*cos(f);y=r.*sin(f);$ »$figure;comet(x,y)$</p>  <p><i>comet3(X,Y,Z,P)</i> функциясы <i>X, Y, Z</i> бірөлшемді жиымдармен берілген нүктенің кеңестіктегі траекториясы бойынша қозғалысын салады. <i>P</i> параметрі арқылы комета құйрығының ұзындығы беріледі</p>

4.8 Қосымша мысалдар

1-мысал. Берілген $[xp; xk]$ аралықта берілген

$$y(x) = \begin{cases} 50 \frac{\sin(x)^2}{x} & \text{егер } x < -4 \\ \frac{10 \cdot e^x}{x^2 + 1} & \text{егер } -4 < x < 4 \\ x^2 / 5 & \text{егер } 4 < x < 10 \\ x & \text{егер } 10 < x \end{cases}$$

функциясының графигін салу үшін функция құру керек (мәндері dx адыммен есептелінеді).

Функция мәтіні:

```
function v=Vg(xn,dx,xk)
```

```
x=xn:dx:xk;
```

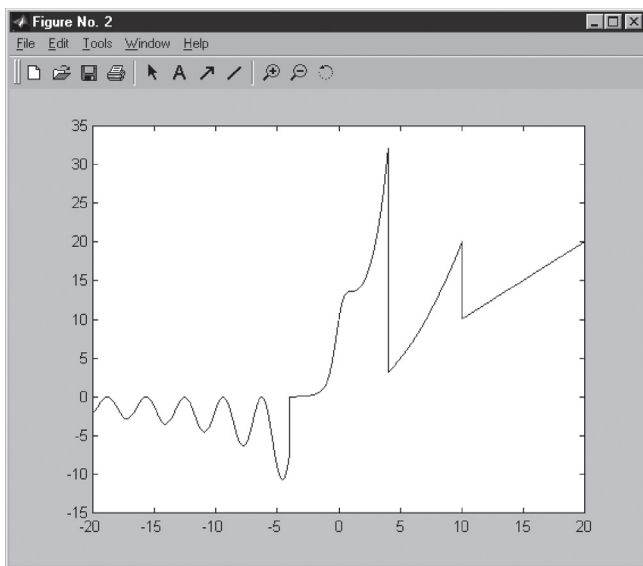
```
y=(x<=-4).*(50*sin(x).^2./x)+(x>-4&x<=4).*(10*exp(x)./(x.^2+1))+(x>4&x<=10).*(x.^2/5)+(x>10).*x;
```

```
plot(x,y)
```

Функцияны шақыру:

```
»Vg(-20,0.0001,20)
```

Нәтижесі:



2-мысал. Алдыңғы мысалдағы $y(x)$ функциясы мен $f(x) = x^3 - 7x + 6$ функциясының аралық шектерінің қиылысу нүктелерінің санын анықтау керек. Табылған аралықтардың орталықтарынан шеңбер салу керек. Кометаны пайдаланып, берілген функция және табылған қиылысу нүктелері графиктерімен шектелген жазықтық бөлігін жүргізу. Комета қозғалысын екінші қиылысу нүктесінде бастап-аяқтау керек: бірінші аймақты сағат тілімен бағыттас жүргізу керек, ал екіншісін - сағат тіліне қарсы бағытта.

Функция:

```

function l4_p2_3(xn,xk,dx)
% 2-мысал
% есептен, график құру
hold off
x=xn:dx:xk;
y1=f1(x);
y2=f2(x);
plot(x,y1,'g','LineWidth',3)
hold on
plot(x,y2,'m','LineWidth',2)
axis([-20 20 -20 40]) % масштабты өзгерту% графиктегі
қиылысу нүктелердің координаттарын шамамен анықтау
y1_y2=y1-y2;
r=[x(1:end-1);
y1_y2(1:end-1).*y1_y2(2:end)<0];
rt=r';
rs=flipud(sortrows(rt,2));
nt=sum(rs(:,2))
xn=rs(1:nt,1)
xk=xn+dx
xr=xn+dx/2;
xr=sort(xr);
yr=f1(xr);% графиктердегі шамамен табылған қиылысу
нүктелерінің орталығынан шеңбер сызуscatter(xr,yr,dx*10000)
dx=dx/2; % адымды кеміту% комета траекториясының коор-
динаттарын беру
xc1= xr(2):-dx:xr(1);
xc2=xr(1):dx:xr(2);
xc3=xr(2):dx:xr(3);
xc4=xr(3):-dx:xr(2);
xc=[xc1 xc2 xc3 xc4];
yc=[min([f1(xc1); f2(xc1)]) max([f1(xc2); f2(xc2)])
min([f1(xc3);f2(xc3)]) max([f1(xc4); f2(xc4)]) ];
comet(xc,yc) % кометаның траектория бойынша қозғалысы
function y=f1(x)

```

```

y=(x<=-4).*(50*sin(x).^2./x)+...
(x>-4&x<=4).*(10*exp(x)./(x.^2+1))+...
(x>4&x<=10).*(x.^2/5)+...
(x>10).*x;
function y=f2(x)
y=x.^3-7*x+6;

```

Нәтижесі (нөлге бөлу туралы ескертусіз): Алынған график:

```

» l4_p2_3(-20,20,0.05)
nt =
    3
xp =
    3.5000
   -0.2500
   -3.0000
xk =
    3.5500
   -0.2000
   -2.9500

```

3-мысал. Абсциссасы x_1 -ден x_2 -ге дейін және ординатасы y_1 -ден y_2 -ге дейін болатын координат жазықтығында кездейсоқ түрде n нүктелер беріледі. Сол жақ жоғарғы бұрышының координаттары (xp_1, yp_1) , ал оң жақ төменгі бұрышының координаттары (xp_2, yp_2) болатын тік төртбұрышқа (оның шекараларын қоса) жататын нүктелер санын анықтау керек. Тік төртбұрышты және оған жататын нүктелерді бір түспен боялған дөңгелекпен, ал жатпайтындарын басқа түспен боялған кіші радиустағы дөңгелекпен салу керек.

Функция:

```

function nt=l4_p3(x1,x2,y1,y2,n,xp1,yp1, xp2,yp2)
hold off
xp=[xp1,xp2,xp2,xp1 xp1];
yp=[yp1, yp1,yp2,yp2, yp1];
plot(xp,yp)% тік төртбұрыш салу
hold on
axis([x1,x2,y1,y2])
grid on

```



```

x=fv(x1,x2,n);
y=fv(y1,y2,n);
r=x>=xp1&x<=xp2&y>=yp2&y<=yp1;%   тік   төртбұрышқа
нүктенің жатуы
xy=[x y r]
nt=sum(xy(:,3)); % тік төртбұрышта жататын нүктелер саны
scatter(xy(:,1),xy(:,2),50+50*xy(:,3),4+xy(:,3)*9, 'filled') % нүктелер-
ді салу
function k=fv(x1,x2,n)
k=x1+(x2-x1)*rand(n,1);

```

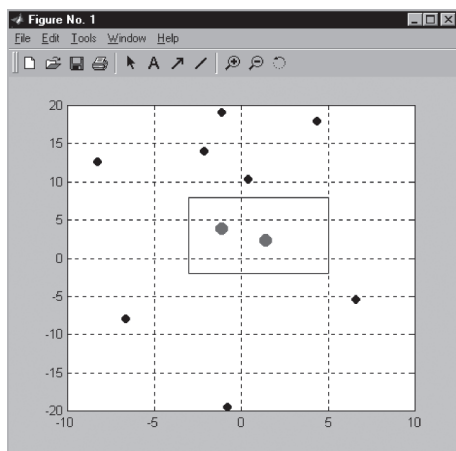
Нәтижесі:

```

» l4_p3(-10,10,-20,20,10,-3,8, 5,-2)
xy =
 6.5773  -5.3480    0
-6.6746  -7.8986    0
-2.1219  14.0738    0
 0.4151  10.3792    0
 4.3625  17.9904    0
 1.3838   2.3175   1.0000
-0.7839 -19.4307    0
-1.0939   3.8471   1.0000
-8.2451  12.6482    0
-1.1303  19.0837    0
ans =
 2
»

```

Графигі:



5 БАҒДАРЛАМАЛАУ

5.1 Функцияны енгізу және шығару

Disp(айнымалы атауы) функцияның атауын көрсетпей-ақ айнымалы мәнін экран бетінен енгізуге арналған. *Disp* функциясының аргументі дәйекшелерге алынған мәтіндік жолда болуы мүмкін:

$disp('символдық жол')$

Әрбір *disp* функциясын шақыру жаңа жолға ауысуды іске асырады.

Input функциясы экранға сұраныс жасау үшін және бір айнымалының мәнін енгізуді қорытындылау үшін арналған немесе MATLAB жүйесіндегі (енгізілетін функция өрнектері шақырылған функция мәні болуы мүмкін) өрнектерді кері қайтарады. Функцияны шақыру келесідей түрде жазылады.

$айнымалы\ атауы = input('шақыру')$

Қайтарылатын функцияның айнымалы мәнін тағайындайды, оның атауы шақыру арқылы берілген (жиымды енгізу кезінде оның элементтері тік жақшада және матрицалар жолдардың соңында енгізу керек;). Мысалы, *inp.m*: файлда сақталған мәтін сценарийі:

```
x=input('x айнымалы мәнін енгізіңіз')
s=input('s айнымалы мәнін енгізіңіз')
v=input('v векторын енгізіңіз ')
h=input('h матрицаны енгізіңіз ')
```

Нәтижесі:

```
» inp
айнымалы мәнін енгізіңіз x 4
x =
    4
айнымалы мәнін енгізіңіз s sin(0.3)/2+x^2-7
s =
    9.1478
```

```
v [3 5 1] векторды енгізіңіз
v =
    3    5    1
h [2 4; 7 8] матрицаны енгізіңіз
h =
    2    4
    7    8
»
```

Input функциясын шақыру келесідей түрде жазылады.

```
айнымалы атауы =input(' шақыру ', 's')
```

Мысалы:

```
» h=input('Өзіңіздің атыңызды енгізіңіз','s')
Өзіңіздің атыңызды енгізіңіз ГҮЛНАР
h =
ГҮЛНАР
»
```

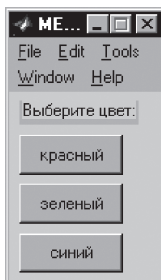
Menu функциясын шақыру келесі түрде жазылады.

```
айнымалы атауы =menu(' мәзір тақырыбы ', ' 1 таңдау ',  
' 2 таңдау ', ... , ' n таңдау ')
```

Экран бетіне таңдау батырмалары, мәзір және тақырып шығарылады. Функция таңдалған батырма нөмірін қайтарады (функцияны шақыру кезіндегі тізімде). Мысалы:

```
» k=menu('Түсті таңдаңыз:', 'қызыл', 'жасыл', 'көк')
k =
    3
»
```

Функцияларды орындаған кезде экранға келесідей мәзір пайда болады:



«Көк» батырмасын басқанда функция 3 мәнін қайтарды.

5.2 Циклдік операторлар

MATLAB-та екі циклдік оператор бар: *for* операторы берілген санның циклдік орындалуын қамтамасыз етеді, *while* циклдік операторлар тобын логикалық өрнектің мәніне сәйкес орындайды. Циклдік операторлар есептеулердің командалық терезесінде, сценарийлер мен функцияларда қолданылады.

5.2.1 For циклдік операторы

Циклдік оператордың жазылуы:

```
for [цикл параметрі] = [бастапқы мән] : [адым] : [соңғы мән]
    [циклдік дене] операторлар
end
```

Цикл денесі цикл параметрінің бастапқы мәннен соңғы мәніне дейінгі мәндерді қабылдау аралығында жұмыс істейді. Егер адым бірге тең болса, оны көрсетпеуге де болады. Циклдік дене операторлары әртүрлі жолдарда немесе бір жолда орналаса алады (бұл жағдайда оларды үтір немесе нүктелі үтір арқылы ажыратуға болады).

Мысалы, берілген адым мен аралықтағы сандардың туындысы мен қосындысын анықтау үшін функция құрастырамыз:

```
function [s,p]=sp(in,ik,di)
s=0;
p=1;
for i=in:di:ik
    s=s+i;
    p=p*i;
end
```

Нәтижесі:

```
» [S,P]=sp(3,15,2)
S =
    63
P =
    2027025
»
```

Цикл параметріне мән беру үшін вектор-жолды пайдалануға болады:

```
for=цикл параметрі= А жиымы  
(циклдік дене) операторлар  
end
```

Егер A вектор-жол болса, онда цикл параметрі вектор элементтерінің мәнін дәйекті қабылдайды. Мысал ретінде вектор элементтерінің мәндерінің қосындысын анықтайтын функция құраймыз.

```
function s=sv(v)  
s=0;  
for i=v  
    s=s+i;  
end
```

Нәтижесі:

```
» sv([4 6 3])  
ans =  
    13  
»
```

Егер A матрицасының әрбір қатары цикл параметрі болса, онда матрицада қанша қатар болса, сонша цикл орындалады.

Алдында құралған вектор элементтерінің қосындыларының мәнін есептеу функциясын қолданып, матрица қосындысының мәнін анықтайтын функция құрастырайық:

```
function s=sm(a)  
s=0;  
for i=a  
    s=s+sv(i');%бағананы тасымалдаймыз  
end
```

Нәтижесі:

```
» f=[3 4; 5 7; 2 1]  
f =  
     3     4  
     5     7  
     2     1  
» sm(f)
```

```
ans=  
22
```

5.2.2 While циклдік операторы

While циклдік операторының жазылуы келесі түрдегідей болады.

```
While логикалық өрнек операторы  
(циклдік дене) операторлар ai  
end
```

Цикл денесі логикалық өрнек шынайы болғанда орындала береді. Логикалық өрнектің мәні жиым болса және оның барлық элементтері шынайы болса (нөлге тең емес), сонымен қатар жиым бос болмаса, ([] - бос жиым), онда оның мәні шынайы болады.

Функция мәтіні (функцияның қосындысынан басқа итерациялардың санын қайтарады):

```
function [s,i]=sr1(eps)  
s=0;  
i=0;  
y=1;  
while abs(y)>eps  
    i=i+1;  
    y=1/i/(i+1);  
    s=s+y;  
end
```

Нәтижесі:

```
» [s1,n1]=sr1(0.01)  
s1 =  
    0.9091  
n1 =  
    10  
» [s2,n2]=sr1(0.0000000005)  
s2 =  
    1.0000  
n2 =  
    44721
```

Егер қайтарылатын мәндерде вектор қалыптаспаған болса, онда функция тек қана біріншісін қайтарады:

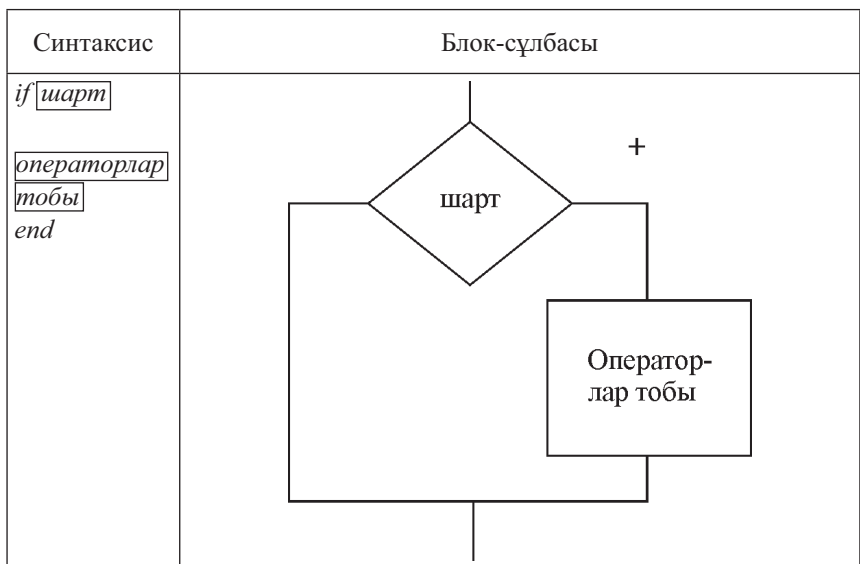
```
» sr1(0.0000000005)
ans =
    1.0000
```

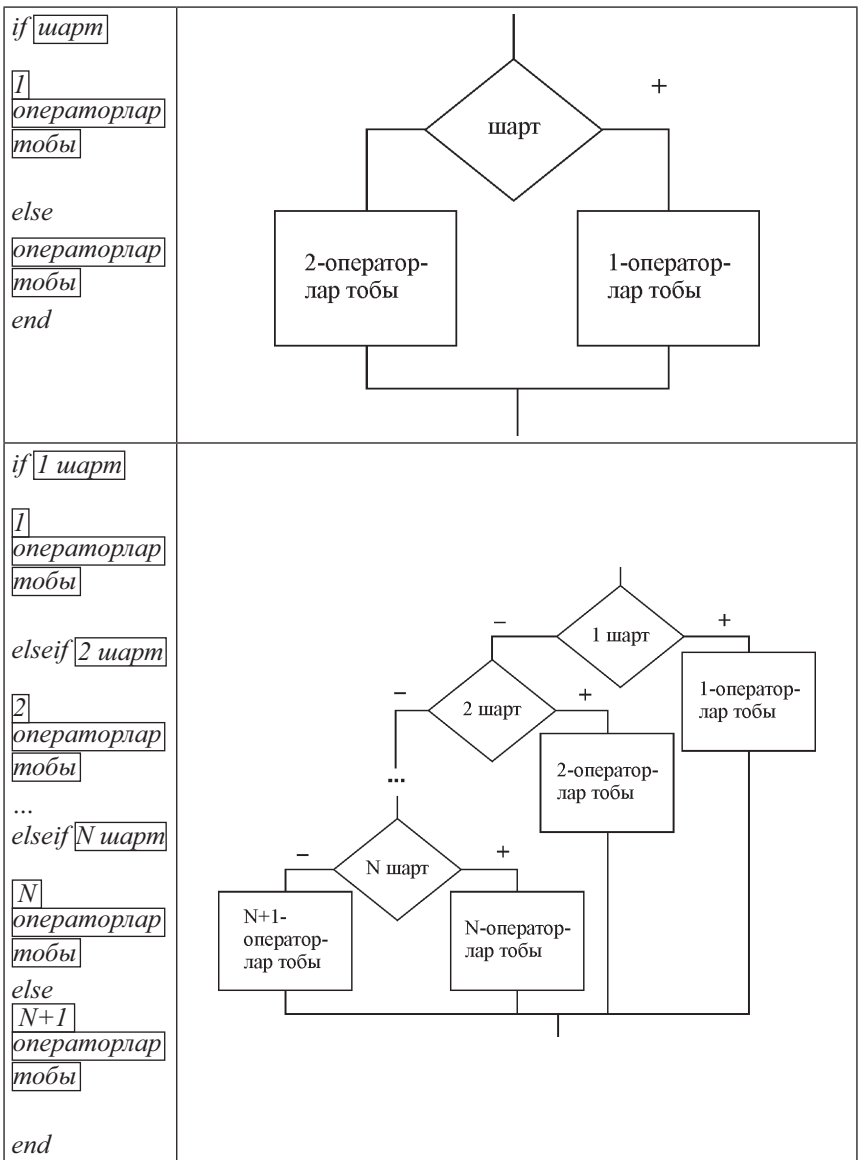
5.3 Тармақталған оператор

Тармақталған операторға ауыстырып-қосу операторы және шартты оператор жатады.

5.3.1 Шартты оператор

Шартты оператор келесі формалардың біреуіндегі түрде жазылуы мүмкін:





Шарт ретінде кез келген өрнек қолданылуы мүмкін (әдетте, логикалық өрнектер қолданылады). Егер шарт ретінде жиым қолданылса, ол жиымның барлық элементтері нөлге тең және бос болмаса, онда ол «ақиқат» болады.

Мысалы, үш саннан көп санды анықтау үшін *max* функциясын құраймыз:

```
function max=l5_p1(a,b,c)
  if a>b
    if a>c
      max=a;
    else
      max=c;
    end
  elseif b>c
    max=b;
  else
    max=c;
  end
```

Нәтижесі:

```
» l5_p1(13, 4, 5)
ans =
    13
```

5.3.2 Ауыстырып-қосқыш оператор

Ауыстырып-қосқыш операторы келесі түрде жазылады.

```
switch [BI айнымалы немесе өрнек]
  case [1-ші мағынасы]
    [1 операторлар тобы]
  case { [2-ші мағынасы], [3-ші мағынасы], [4-ші мағынасы], ... }
    [2 операторлар тобы]
  ...
  otherwise
    [n операторлар тобы]
end
[Z оператор]
```

Switch түйінді сөзі жазылғаннан кейін, өрнек пен айнымалы мәніне тәуелді болуын тармақталған оператор ұйымдастырады.

Case тобының саны әркалай бола береді. *Case* түйінді сөзінен кейін өрнектің мәні немесе фигуралы жақшаға алынған бірнеше мәндер жазылады.

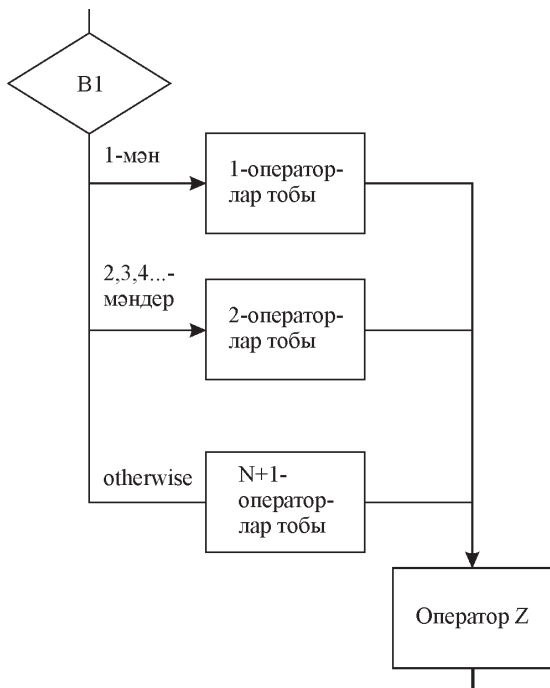
Оператор келесідей жағдайларда жұмыс істейді:

- *switch* түйінді сөзінен жазылғаннан кейінгі өрнектің мәнін есептейді;

- егер өрнек мәні *case* түйінді сөздерінің бірінен кейінгі жазылған мәнге сәйкес келсе, онда сол *case* түйінді сөзінен кейінгі операторлар тобы кезекпен орындалады. Әрі қарай *end* түйінді сөзінен кейінгі операторларға өту орындалады. Егер *case* түйінді сөзінен кейін бірнеше мән фигуралы жақша ішінде жазылса, онда операторлар тобы орындалады;

- егер өрнек мәндері ешбір *case* түйінді сөзінен кейінгі жазылған мәнімен сәйкес келмесе, онда *otherwise* түйінді сөзінен кейінгі операторлар тобы орындалады (егер ол операторлар бар болса).

Блок-сұлбасы:



Мысалы, $\sin(x)$, x^2 , e^x функцияларының біреуінен график құру функциясын құраймыз:

```
function graph3
v=menu('Функцияны енгізіңіз','sin(x)','x^2','exp(x)');
xn=input('аргументтің бастапқы мәнін енгізіңіз');
xk=input('аргументтің соңғы мәнін енгізіңіз');
dx=(xk-xn)/500;
x=xn:dx:xk;
switch v
case 1
    y=sin(x);plot(x,y)
case 2
    y=x.^2;plot(x,y)
case 3
    y=exp(x);plot(x,y)
otherwise %егер мәзірді шақырғаннан кейін v айнымалының
мәні өзгереді.
    disp('дұрыс емес таңдау')
end
```

Дәл сол мысал жолдық айнымалы үшін:

```
function graph4
v=input('функцияны енгізіңіз','s');
xn=input('аргументтің бастапқы мәнін енгізіңіз');
xk=input('аргументтің соңғы мәнін енгізіңіз');
dx=(xk-xn)/500;
x=xn:dx:xk;
switch v
case 'sin(x)'
    y=sin(x);;
    plot(x,y)
case 'x^2'
    y=x.^2;
    plot(x,y)
case 'exp(x)'
    y=exp(x);
    plot(x,y)
```

```
otherwise
disp('дұрыс емес таңдау')
end
```

Мәзірдің таңдаулы тармағы және мәзірге қайтаруына байланысты одан шығатын тармақтың орындалуын ұйымдастыру мысалы:

```
function l5_p2
v=1
while v~=3
v=menu('таңдау', '1', '2', 'шығу')
switch v
case 1
disp(1)
case 2
disp(2)
case 3
disp('ШЫҒУ')
end
end
disp('СОҢЫ')
```

5.4 Басқа операторлар

Break операторы *for* және *while* циклдерін тоқтады. Мысалы:

```
for i=1:1:5
for j=1:5
if i==2&j==3
break
end
end
if i==2
break
end
end
i
j
```

Return операторы шақырылған функцияны кері қайтару немесе пернемен жұмыс істеу режимін орындайды.

```
function [i,j]=return1
for i=1:1:5
    for j=1:5
        if i==2&j==3
            i
            j
        return
    end
end
end
```

Нәтижесі:

```
» [z,x]=return1
i =
    2
j =
    3
z =
    2
x =
    3
»
```

Бағдарламаны орындау кезінде кез келген пернені басқанға дейін үзіліс жасау үшін *pause* командасы қолданылады. *Pause(n)* командасы *n* секундта жұмыс жүйесін тоқтатады. *Pause on* командасы үзілісті іске қосса, *pause off* командасы оның жұмысын тоқтатады.

Мысалы:

```
function pause1
xn=input('аргументтің бастапқы мәнін енгізіңіз');
xk=input('аргументтің соңғы мәнін енгізіңіз');
dx=(xk-xn)/500;
x=xn:dx:xk;
for v=1:3
```

```

subplot(2,2,v)
switch v
case 1
    y=sin(x);
    plot(x,y)
case 2
    y=x.^2;
    plot(x,y)
case 3
    y=exp(x);
    plot(x,y)
end
pause
end

```

5.5 Мысалдар

1-ші мысал. Жазықтықтарда n нүкте берілген. Ең кіші массасы бар нүктенің кейбіреуі өзіне ең жақын тұрған нүктеге өз массасын алып беріп, өзі жойылады. Ол үдеріс жалғыз нүкте қалғанша жүреді. Сол нүктенің координатасын анықтау керек.

Функциялардың мәтіндері:

```

function l5_p3_s_1(n,x1,x2,y1,y2,m1,m2)
x=fv(x1,x2,n);
y=fv(y1,y2,n);
m=fv(m1,m2,n);
[X,Y]=l5_p3_1(x,y,m,x1,x2,y1,y2)

function k=fv(x1,x2,n)
k=x1+(x2-x1)*rand(n,1);

```

```

function [X,Y]=l5_p3_1(x,y,m,xn,xk,yn,yk)
a=[x y m];
s=size(a);
scatter(x,y,sqrt(m/pi)*100,'filled')
axis([xn,xk,yn,yk])
for i=1:s(1)-1
    [min,imin]=min(a(:,3));
    b=a(imin,:);

    pause

    a(imin,:)=[];
    a(:,4)=sqrt((a(:,1)-b(1)).^2+(a(:,2)-b(2)).^2);
    [rmin,ib]=min(a(:,4));
    a(ib,3)=a(ib,3)+b(3)
    scatter(a(:,1),a(:,2),sqrt(a(:,3)/pi)*100,'filled')
    axis([xn,xk,yn,yk])
end
X=a(1);
Y=a(2);

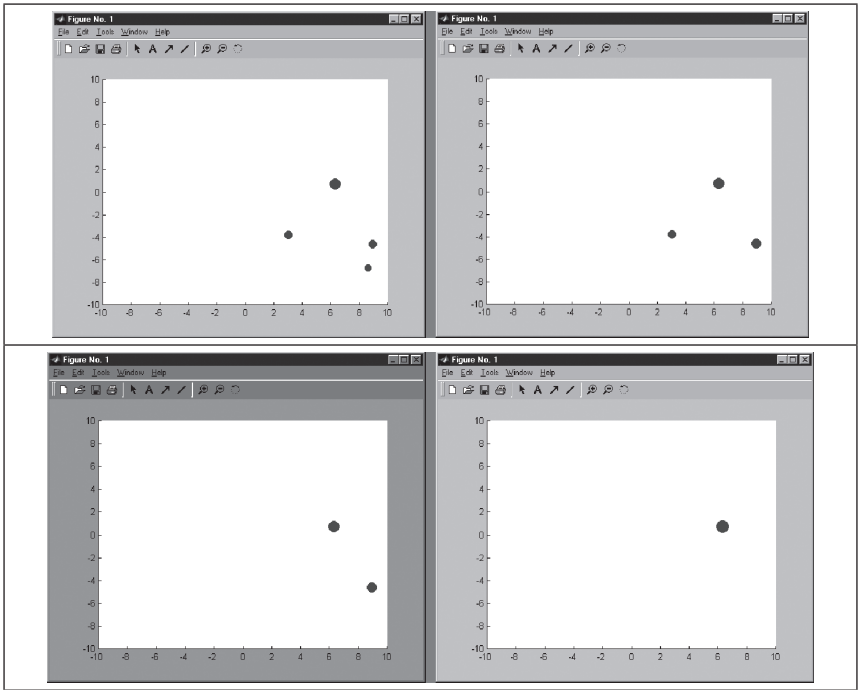
```

Нәтижесі:

```

» l5_p3_s_1(4,-10,10,-10,10,1,10)
a =
    3.0310   -3.8012    2.8989    6.2986
    8.9217   -4.6237    4.4263    2.1344
    6.3177    0.7290    6.8661    7.8061
a =
    8.9217   -4.6237    4.4263    5.9479
    6.3177    0.7290    9.7650    5.5969
a =
    6.3177    0.7290   14.1913    5.9525
X =
    6.3177
Y =
    0.7290
»

```



2-ші мысал. Бастапқы матрицаның нөлдік элементтерінің индекстерін және санын анықтау керек.

Функция (нөлдік элементтердің саны - k , a жиымында ол нөлдік элементтердің индексі, бірінші бағанда - жолдың нөмірі, екінші бағанда - бағанның нөмірі):

```
function [a,k]=l5_p4(b)
s=size(b);
k=0;
for i=1:s(1)
    for j=1:s(2)
        if b(i,j)==0
            k=k+1;
            a(k,1)=i;
            a(k,2)=j;
        end
    end
end
```


Нәтижесі:

» $h = [4 \ 0 \ -3; \ 6 \ 8 \ 0; \ 0 \ 0 \ 0], [R, K] = l5_p4(h)$

$h =$

4 0 -3

6 8 0

0 0 0

$R =$

1 2

2 3

3 1

3 2

3 3

$K =$

5

»

6 MATLAB-ТАҒЫ ТИПТІК ЕСЕПТЕР

6.1 Сызықтық емес теңдеулерді және теңдеулер жүйесін шешу

$f(x) = 0$ теңдеуінің түбірін анықтау үшін $fzero(fun, x0, tol)$ түрдегі функциясы қолданылады. Мұндағы, fun – түбірін табу керек функция, $x0$ - түбіріннің бастапқы жуықтауы. $fzero$ функциясы түбірдің жуық мәнін қайтарады; tol – анықталатын түбірдің қателігі (егер қателік берілмесе, ол eps -ге тең болып саналады). fun функциясы MATLAB ережесі бойынша формула ретінде беріледі, дәйекшеде немесе M-функция атымен теңдеудің сол жағын анықтауыш ретінде жазылады. Бастапқы жуықтау скалярлық түрде немесе аралық түрінде берілуі мүмкін. Функция соңында әртүрлі белгі арқылы беріледі (векторлық координат басы және аралық соңы).

Мысалы, $2^x + \sin(x) + x = 0$ теңдеуінің түбірін анықтайық. Теңдеудің сол жағын есептеу үшін $f7$ аламыз:

```
unction y=f7(x)
y=2.^x+sin(x)+x;
```

Келесі $fzero$ функциясын әртүрлі мәнде аламыз:

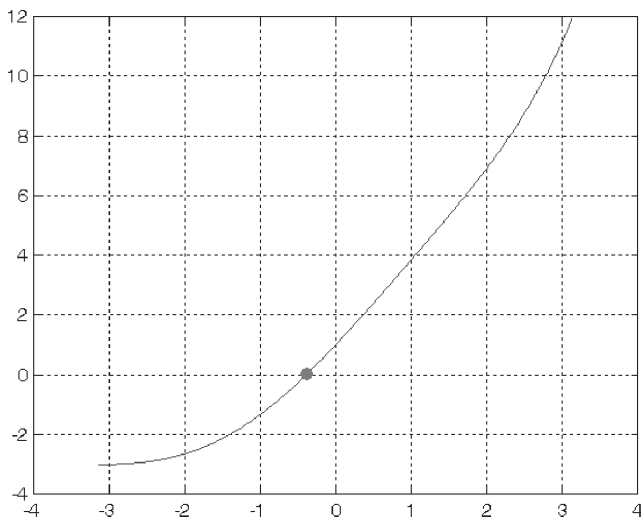
```
x=-pi:0.01:pi;
plot(x,f7(x));
grid on
fzero('2.^x+sin(x)+x',-0.5)
fzero('2.^x+sin(x)+x',-0.5,1e-3)
fzero('f7',-0.5,1e-4)
xr=fzero('f7',[-1,0],1e-4);
hold on;
scatter(xr,f7(xr),40,'r','filled');
```

Нәтижесі:

```
» l7_p0
Zero found in the interval: [-0.38686, -0.58].
ans =
-0.3871
```

```
ans =
-0.3869
ans =
-0.3871
»
```

Графигі:

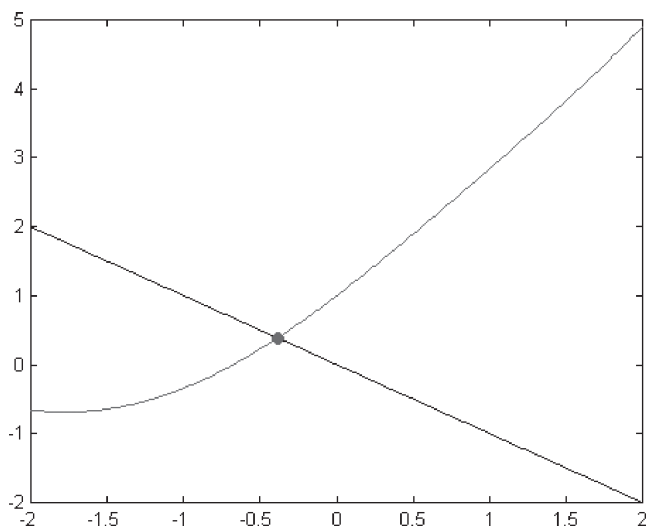


fzero функциясы берілген функцияның абсцисса осіндегі қиылысқан нүктелерін анықтайды. Функцияның абсцисса осімен қиылысқан нүктелері 0 болып есептелмейді.

Теңдеулер мен теңдеулер жүйесін символдық түрде шешу үшін *solve* функциясы қолданылады. Ол функцияны қолданғанда, теңдеу доғал жақшаға алынып, дәйекшемен үтір арқылы жазылады. Мысалы, $2^x + \sin(x) = -x$ теңдеуінің графикалық шешімін мына түрде орындауға болады:

```
»xr=solve('2^x+sin(x)=-x'); x=-2:0.01:2;y1=2.^x+sin(x); y2=-x;...
plot(x,y1,x,y2);hold on; scatter(double(xr),double(-xr),40,'filled')
» xr
xr =
-.38712465392757487234848391052801
»
```

Алдағы суретте $y = 2^x + \sin(x)$ және $y = -x$ функцияларының графигі және олардың қиылысқан нүктелері көрсетілген:



Келтірілген мысалда xr символдық айнымалы қолданылады, сол себепті нүктенің құрылуында *double* типі қолданылады (*double (xr)*).

Келесі мысал теңдеулер жүйесінің шешімін көрсетеді:

$$\begin{cases} \sin(x + y) - 1.2x = 0.2 \\ x^2 + y^2 = 1 \end{cases}$$

```
» [xr,yr]=solve('sin(x+y)-1.2*x=0.2','x^2+y^2=1')
xr
=-.95682502589007371450373183654668
yr
=-.29066453142834570132077207025201
»
```

Тексеру:

```
» x=double(xr)
x =
    -0.95683
» y=double(yr)
y =
    -0.29066
» abs(sin(x+y)-1.2*x-0.2)<eps
```

```
» abs(sin(x+y)-1.2*x-0.2)<eps
```

```
ans =
```

```
1
```

```
» abs(x^2+y^2-1)<eps
```

```
ans =
```

```
1
```

6.2 Бір немесе бірнеше айнымалысы бар функциялардың минимумын анықтау

Функцияның минимумын табу бір айнымалысы бар *fmin* функциясының көмегімен іске асырылады және келесі түрде жазылады:

```
xmin=fmin('функция атауы', x1,x2)
```

```
xmin=fmin('функция атауы', x1,x2, options)
```

```
xmin=fmin('функция атауы', x1,x2, options, p1, p2, ..., p10),
```

мұндағы, *функция атауы* – табу керек минимумның

M-функциясының атауы; *x1*, *x2*- аралықтың басы және соңы; *options* – басқаратын параметр векторы, ол оптимизациялау алгоритмінің құрылғысына арналған, егер бәрі қалыпты күйде қалса, онда бұл аргумент орнына бос жиым жазылады. *p1*, *p2*, ..., *p10* – функция параметрлері.

Мысалы, функцияның минимумдарын аралық арқылы анықтау:

Функция	Аралық
$\cos(x)$	$[1, 4]$
$\cos(p1*x+p2)$ егер $p1=2, p2=1$	$[1, 4]$
$\cos(p1*x+p2)$ егер $p1=1, p2=2$	$[5, 8]$

M-функциясы:

```
function y=l7_p9f1(x,p1,p2)
```

```
y=cos(x*p1+p2);
```

Сценарийдің мәтіні:

```
xmin1=fmin('cos',1,4)
```

```
xmin1_2=fmin('l7_p9f1',1,4,[],1,0)
```

```
xmin2=fmin('l7_p9f1',1,4,[],2,1)
```

```

xmin3=fmin('l7_p9f1',5,8,[],1,2)
x=0:0.01:pi*4;
plot(x,[cos(x);l7_p9f1(x,2,1);l7_p9f1(x,1,2)])
hold on
grid on
scatter([xmin1;xmin2;xmin3],[cos(xmin1);l7_p9f1(xmin2,2,1);l7_p9f1(xmin3,1,2)],70,'filled')

```

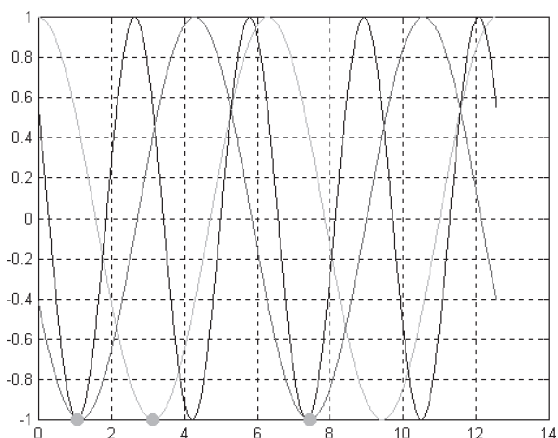
Нәтижесі:

```

» l7_p9
xmin1 =
    3.14158640896816
xmin1_2 =
    3.14158640896816
xmin2 =
    1.07078246937276
xmin3 =
    7.42479085012525

```

Графигі:



Бірнеше айнымалысы бар функцияларды минимизациялау келесі функциямен анықталады:

$$xmin=fmins(' \boxed{\text{функция атауы}} ',x0)$$

Айнымалыдағы *xmin* векторын осы функция қайтарады, функция мұнда бастапқы жуықтаудың минимальді мәндерін *x0* векторында қабылдайды.

$f(x, y) = \cos(x) * \sin(y)$ функциясының минимумының координаттарын (3,-5) нүктелер арқылы арқылы табамыз. График жоғарғы түрде салынып, сызықтары минимум нүктесін көрсетеді.

Сценарий:

```
close all;
x=-2*pi:0.3:2*pi;
y=x;
x0=[3, -5];
xmin=fmins('l7_p10f2',x0)
[xx,yy]=meshgrid(x,y);
subplot(1,2,1)
mesh(xx,yy,l7_p10f1(xx,yy))
axis([-2*pi,2*pi,-2*pi,2*pi,-1,1])
hold on
scatter3(xmin(1),xmin(2),l7_p10f1(xmin(1),xmin(2)),70,2,'filled')
subplot(1,2,2)
contour(xx,yy,l7_p10f1(xx,yy))
grid on
hold on
scatter3(xmin(1),xmin(2),l7_p10f1(xmin(1),xmin(2)),70,2,'filled')
```

l7_p10f1 және *l7_p10f2* функциясы:

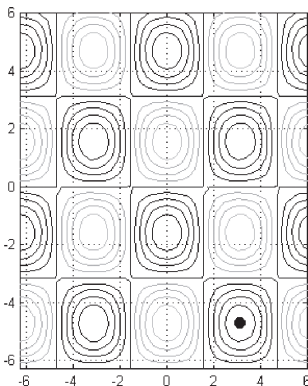
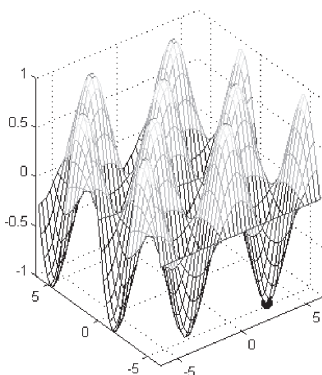
```
function f=l7_p10f1(x,y)
f=cos(x).*sin(y);
```

```
function f=l7_p10f2(x)
f=cos(x(1)).*sin(x(2));
```

Нәтиже:

```
» l7_p10
xmin =
    3.14155102269033    -4.71242727650315
```

Алынған график:



6.3 Сандық дифференциалдау мен шектік айырымды есептеу

$diff(x,n)$ функциясы шектік айырымды есептейді. Егер n элементтен тұратын x аргументі бірөлшемді жиым болса, функция бірөлшемді жиымды $[x(2)-x(1) \ x(3)-x(2) \dots x(n)-x(n-1)]$ қайтарады. Қайтарылатын жиымдағы элементтер саны $n-1$.

$diff(x,n)$ функциясы $diff(x,n)=diff(diff(x,n-1))$ қатынасындағы n ретті шектік айырымды есептейді.

```
» y=[4 3 7 1]; dy=diff(y)
```

```
dy =
```

```
 -1  4 -6
```

```
» ddy=diff(dy)
```

```
ddy =
```

```
  5 -10
```

```
» ddyy=diff(ddy,2)
```

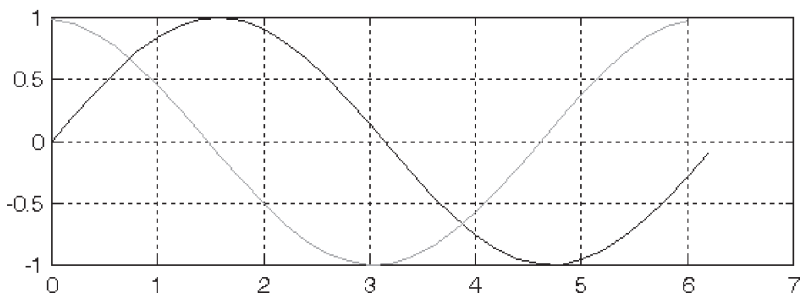
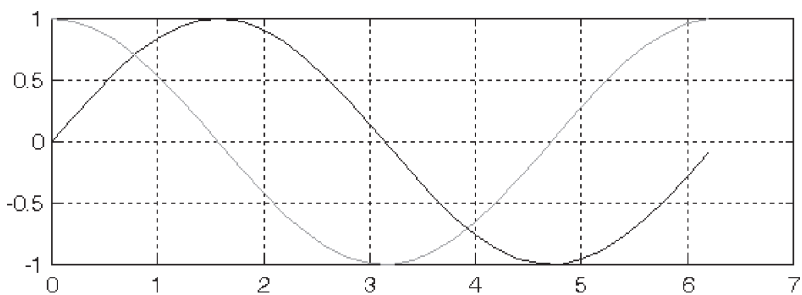
```
ddyy =
```

```
  5 -10
```

```
»
```

Кесте бойынша берілген жуық мән алғашқы функцияның туындысы $y(x)$ бойынша $diff(y)/diff(x)$ қатынасына байланысты.

Мысалдағы бірінші терезеде $sin(x)$ функциясының 1-ші реттік туындысының функциясының графигі берілген.



Сценарий:

```

close all
x=0:0.2:2*pi; y= sin(x);
r=size(x);
n=r(2);
dy=cos(x);d2y=-sin(x);
subplot(2,1,1);
plot(x,y);
hold on;
grid on;
plot(x,dy,'r');
plot(x,d2y,'g');
d1=diff(y)./diff(x);d2=diff(d1)./diff(x(1:n-1)));
subplot(2,1,2);
plot(x,y);
hold on;
grid on;
plot(x(1:n-1),d1,'r');
plot(x(1:n-2),d2,'g');

```

6.4 Сандық интегралдау

y функциясынан x айнымалысы бойынша алынған интегралдың мәнін трапеция әдісімен есептейтін $trapz(x,y)$ функциясы кері қайтарады. Мысалы, $y = x$ функциясының интегралын 1-ден 2-ге дейінгі аралықта есептейміз:

```
x=1:0.1:2;  
y=x;  
i=trapz(x,y)%сандық интегралдау  
k=2^2/2-1/2%тексеру
```

Нәтижесі:

```
» I7_p2  
i =  
    1.5  
ik =  
    1.5
```

Егер функция бір ғана $trapz(y)$ аргументімен шақырылса, онда 1-ге тең интегралдың тұрақты адымы есептеледі (адымның туынды мәніне ие болу үшін интегралдың алынған мәнін интегралдаудың адым санына көбейту керек), яғни

```
x=1:0.1:2;  
y=x;  
i=trapz(y)%сандық интегралдау  
i=i*0.1  
ik=2^2/2-1/2%тексеру
```

Нәтижесі:

```
» I7_p3  
i =  
    15  
i =  
    1.5  
ik =  
    1.5
```

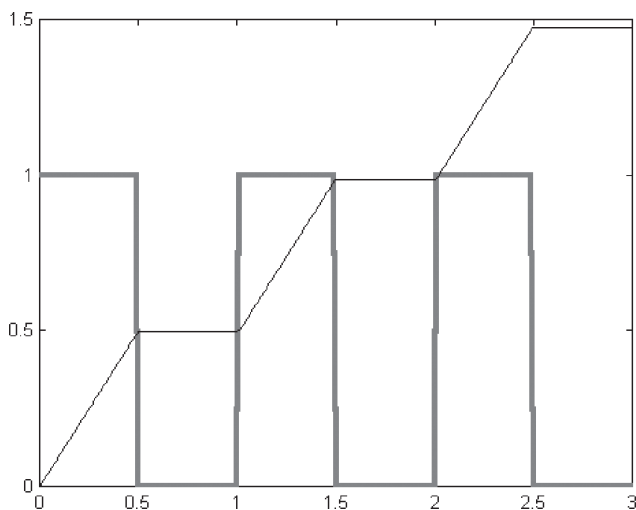
Ал $cumtraps(x,y)$ ($cumtraps(y)$) функциялары қосымша өтпелі нәтижелердің мәнін есептейді. Тікбұрышты импульс түріндегі

(графикте жуан қызыл түспен бейнеленген) функция үшін *cumtraps* функциясы ізделінген интегралдың жинақталуын көрсететін функцияны көрсетуге көмектеседі. *t1* туынды уақыт моментіндегі мәні 0-ден *t1*-ге дейінгі аралықтағы интегралдың мәніне тең.

Сценарий мәтіні:

```
x=0:0.01:3;
y=(x<0.5|x>1&x<1.5|x>2&x<2.5);
i=cumtrapz(x,y)%сандық интегралдау
plot(x,y,'r','LineWidth',3)
hold on
plot(x,i)
```

Графигі:



quad және *quad8* функциясы интегралды квадраттау әдісі бойынша есептеуге көмектеседі (*quad* функциясы Ньютон-Котес формуласы бойынша екінші реттік болып саналады).

Бір типтік функциялар

<i>quad</i> функциясы	<i>quad8</i> функциясы
<i>quad</i> ('функция атауы', <i>a</i> , <i>b</i>)	<i>quad8</i> ('функция атауы', <i>a</i> , <i>b</i>)
<i>quad</i> ('функция атауы', <i>a</i> , <i>b</i> , <i>tol</i>)	
<i>quad</i> ('функция атауы', <i>a</i> , <i>b</i> , <i>tol</i> , <i>trace</i>)	
<i>quad</i> ('функция атауы', <i>a</i> , <i>b</i> , <i>tol</i> , <i>trace</i> , <i>p1</i> , <i>p2</i> , ...)	<i>quad8</i> (...)

мұндағы a , b – төменгі және жоғарғы интегралдың шектері; tol – интегралдаудың салыстырмалы қателігі, тұрақты $1e-3$; егер $trace \neq 0$ – қосымша нақты интеграласты функциясын орындауға қажет; $p1$, $p2$, ... - интеграласты функциясының параметрлері.

Бұл функцияларды $\int_{-\pi/2}^{\pi/2} \cos^2(x)dx$ интегралын есептеу үшін қолданаық.

Сценарий мәтіні:

```
close all;
i1=quad('l7_p5f',-pi,pi);
i1_8=quad8('l7_p5f',-pi,pi);
i2=quad('l7_p5f',-pi,pi,1e-1);
i2_8=quad8('l7_p5f',-pi,pi,1e-10,1);
disp(['i1=',num2str(i1,20),'          i1_8=',num2str(i1_8,20),'
i2=',num2str(i2,20),' i2_8=',num2str(i2_8,20)])
```

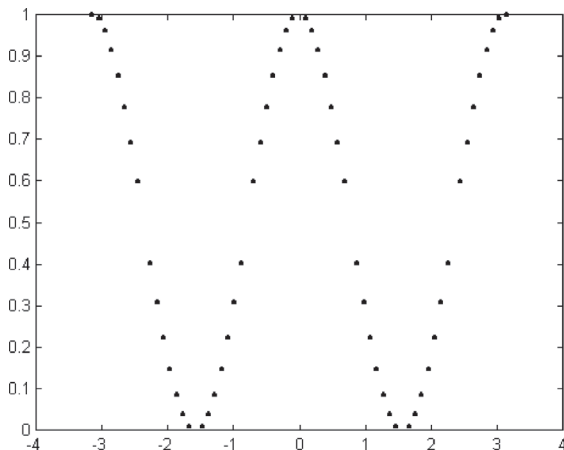
M-функциясы интеграласты өрнегінің мәнін есептеу үшін

```
function y=l7_p5f(x)
y=cos(x).^2;
```

Нәтижесі:

```
» l7_p5
i1=3.1415926535897931 i1_8=3.1415926535897931
i2=3.1415926535897931 i2_8=3.1415926535897931
```

Интеграласты функциясының нүктелік графигі:



$\int_{y_1}^{y_2} \int_{x_1}^{x_2} f(x, y) dx dy$ түрдегі екі еселі интегралды шешуді *dblquad*

функциясы атқарады.

dblquad:

dblquad('функция атауы', x_1, x_2, y_1, y_2)

dblquad('функция атауы', x_1, x_2, y_1, y_2, tol)

dblquad('функция атауы', $x_1, x_2, y_1, y_2, tol, metod$)

мұндағы 'функция атауы' - М-функциясының атауы, интеграласты функциясының мәнін береді. Оның екі параметрі болады: біріншісі ішкі интегралдағы айнымалының векторлық мәні, екіншісі сыртқы интегралдағы айнымалының скалярлық мәні. Интегралданатын функция векторды қайтару керек; x_1, x_2 – ішкі интегралдың төменгі және жоғарғы шектері; y_1, y_2 – сыртқы интегралдың төменгі және жоғарғы шектері;

Мысалы, $\int_0^{2\pi} \int_0^{\pi} \sin(x)^2 \cdot \cos(x)^2 \cdot y dx dy$ интегралының мәнін таба-

мыз. М-функциясының мәтіні:

```
function f=l7_p6f(x,y)
f=sin(x).^2.*cos(x).^2*y;
```

Функцияны шақыру және есептеудің нәтижелері:

```
» i=dblquad('l7_p6f',0,pi,0,2*pi)
i =
    7.75156917007495»
i=dblquad('l7_p6f',0,pi,0,2*pi,'quad8')
i =
    7.75156917007495
```

6.5 Көпмүшелерге қатысты есептерді шығару

Полином немесе көпмүше

$$P(x) = a_1 x^n + a_2 x^{n-1} + a_3 x^{n-2} + \dots + a_n x + a_{n+1}$$

түрінде беріледі.

Көпмүше коэффициенттердің векторы арқылы беріледі (айнымалы дәрежесінің кему ретімен) $A = [a_1, a_2, \dots, a_{n+1}]$ (вектор

элементінің коэффициент саны ретінде көпмүше дәрежесіне бір санға артық болатын θ -дік коэффициенттер есепке алынады).

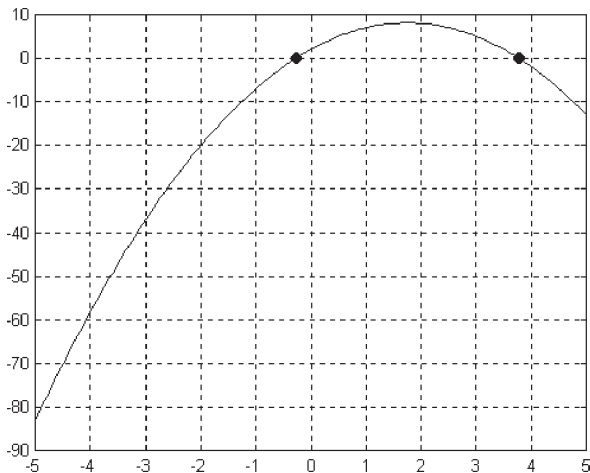
$Polyval(A,x)$ функциясы x аргументінде коэффициенті A векторымен бірге дәрежелі көпмүшенің мәнін қайтарады:

```
» h=[-2 7 2];polyval(h,3)
ans =
    5
» -2*3^2+7*3+2
ans =
    5
```

Егер x - жиым болса, $polyval$ функциясы дәрежелік көпмүше мәндерінің шамасын x жиым элементтерінің барлық мәндерінде кері қайтарады. A векторының коэффициенттерінің бар s көпмүшесінің түбірлерін есептеу $roots(A)$ функциясы арқылы жүзеге асырылады:

```
»r=roots(h)
r =
    3.7656
   -0.26556
»X=-5:0.05:5;Y=polyval(h,X);plot(X,Y);hold on; scatter(r,polyval(h,r),40,'filled'); grid on
```

Алынған график:



Conv функциясы w коэффициентінің векторының көпмүшесін қайтарады және ол берілген коэффициенттердің векторы a және b болатын екі көпмүшенің туындысы болып табылады, яғни $w=conv(a,b)$

Deconv функциясы u және v коэффициенттерінің векторының көпмүшесін қайтарады. Ол көпмүшеден қалған қалдық болып саналады. Коэффициенттің a және b векторлары арқылы d көпмүшесіне берілген. $[u,v]=deconv(a,b)$.

Мысалы:

```

» a=[4 5 8]; b=[2 3 4 7]; d=conv(a,b)
d =
    8    22    47    72    67    56
» deconv(d,b)
ans =
     4     5     8
» d(1)=d(1)+3
d =
    11    22    47    72    67    56
» deconv(d,b)
ans =
     5.5     2.75     8.375
» [x,y]=deconv(d,b)
x =
     5.5     2.75     8.375
y =
     0         0         0    -2.625    14.25    -2.625

```

Көпмүшелерді дифференциалдауды *polyder* функциясы орындайды. Бұл функцияның дәлелі дифференциалданып отырған көпмүше коэффициенттерінің векторы болып табылады. Функция, тиісінше, туынды көпмүшенің коэффициенттерінің векторын кері қайтарады.

```

» a=[4 5 7]; d= polyder(a)
d =
     8     5
»

```

MATLAB-та көпмүшені интегралдаудың да функциясы бар.

6.6 Аппроксимациялау

$B=polyfit(x,y,n)$ функциясы n дәрежелі көпмүшенің $y(x)$ кестелі функциясының аппроксимациясына пайдаланылады, ол дәрежелік көпмүшенің B векторын кері қайтарады. Функция параметрлері: x , y – аргумент пен функцияның векторлық мәні, n – көпмүшенің дәрежесі. Функцияны аппроксимациялаудың нәтижелерін 2, 3, 4 және 5 дәрежелік көпмүшелер арқылы қарастырамыз. Сценарийі:

```
x=1:1:10;  
y=5+(0.1*sin(x));  
scatter(x,y,50,'filled')  
  
b2=polyfit(x,y,2)  
b3=polyfit(x,y,3)  
b4=polyfit(x,y,4)  
b5=polyfit(x,y,5)  
  
xp=0:0.1:11;  
hold on  
plot(xp,polyval(b2,xp),'r')  
plot(xp,polyval(b3,xp),'g')  
plot(xp,polyval(b4,xp),'m')  
plot(xp,polyval(b5,xp),'b','LineWidth',2)
```

Нәтижесі:

```
» I7_p11  
b2 =  
    0.00323858317961304          -0.0397381205521077  
5.10798609433367  
b3 =  
Columns 1 through 3  
    -0.00258158641063002          0.0458347589550084  
-0.236196846401052  
Column 4  
    5.32948620836573  
b4 = Columns 1 through 3  
    -0.000954614601775185          0.0184199348284241  
-0.107858191930798
```

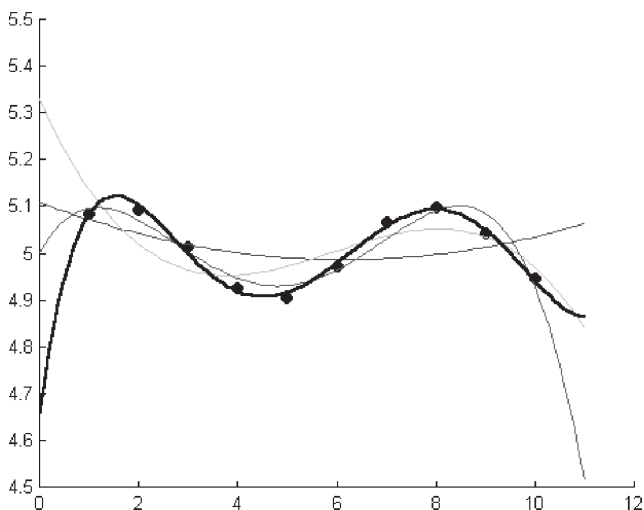


```

Columns 4 through 5
    0.183833578380035    5.00186247703647
b5 = Columns 1 through 3
    0.000245186199226074    -0.00769723508049221
0.0862547832809714
Columns 4 through 6    -0.411276113473067    0.761819178
6889854.65124621214316
»

```

Графигі:



6.7 Интерполяциялау

Бір айнымалылы текше сплайндық кестелік $(y)x$ функциясын интерполяциялау үшін $y_i = \text{spline}(x, y, x_i)$ функциясы қолданылады. Мұндағы, x, y - аргумент пен функция мәні, y_i функциясы x_i аргументі үшін функция мәнін қайтарады.

Мысалы:

```

%кестелі функцияны береміз:
x=1:1:10;
y=5+(0.1*sin(x).^0.1);
scatter(x,y,50,'filled')

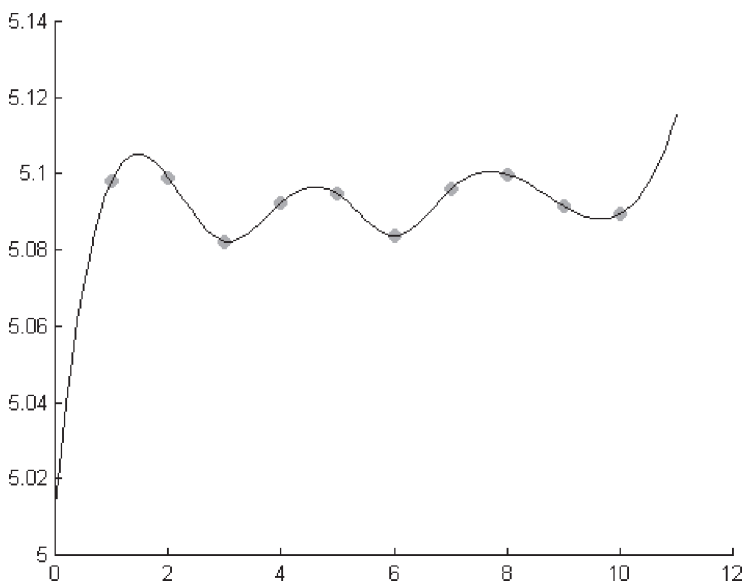
```

```

hold on
xp=0:0.1:11;
%функция графигін құру үшін интерполяция сплайынын
пайдаланамыз:
plot(xp,spline(x,y,xp),'r')

```

Графигі:



$y_i = \text{interp1}(x, y, x_i, 'әдіс')$ функциясы $y(x)$ кестелік функциясын интерполяциялау үшін қолданылады. Мұндағы, interp1 функциясы x_i аргументіндегі y_i кестелік функциясының мәнін кері қайтарады. Ол функцияны қолданғанда бірнеше әдіс қолданылуы мүмкін: *'nearest'* – сатылық интерполяция; *'linear'* - сызықтық интерполяция (егер әдіс көрсетілмесе, онда ол тұрақты орындалады); *'spline'* - текше сплайндарымен интерполяция (сол сияқты spline функциясы); *'cubic'* - текше интерполяциясы.

Мысалы:

```

close all;
%кестелік функция береміз
x=1:1:10;
y=5+(0.1*sin(x).^0.1);

```

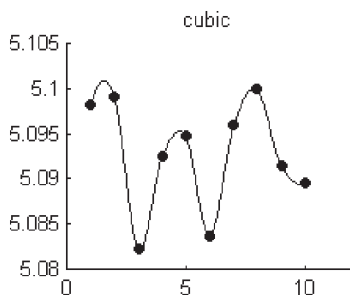
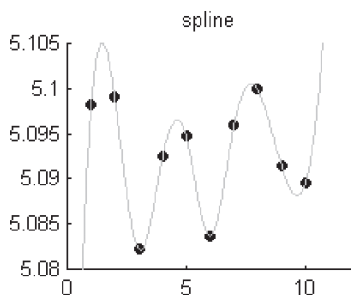
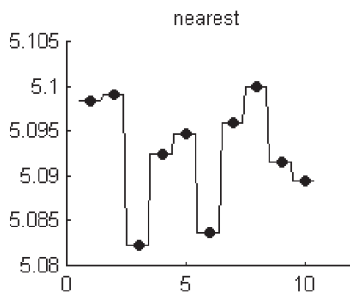
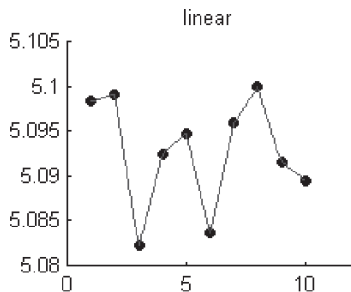
```
xp=0.5:0.1:11;
subplot(2,2,1)
scatter(x,y,30,'filled')
hold on
plot(xp,interp1(x,y,xp),'r')
title('linear')
axis([0,12,5.08,5.105])

subplot(2,2,2)
scatter(x,y,30,'filled')
hold on
plot(xp,interp1(x,y,xp,'nearest'),'k')
title('nearest')
axis([0,12,5.08,5.105])

subplot(2,2,3)
scatter(x,y,30,'filled')
hold on
plot(xp,interp1(x,y,xp,'spline'),'g')
title('spline')
axis([0,12,5.08,5.105])

subplot(2,2,4)
scatter(x,y,30,'filled')
hold on
plot(xp,interp1(x,y,xp,'cubic'),'b')
title('cubic')
axis([0,12,5.08,5.105])
```

Әртүрлі мысалдардан алынған графиктер:



7 МАТЛАВ-ТАҒЫ ДИФФЕРЕНЦИАЛДЫҚ ТЕҢДЕУЛЕР ЖҮЙЕСІ

Дифференциалдық теңдеулер – бұл бір немесе бірнеше айнымалылардан тұратын белгісіз функциясы бар және тәуелсіз айнымалылары бар белгісіз функциясының туындысы бар теңдеу. Бір айнымалысы бар дифференциалдық теңдеудің түрі келесідей болады.

$$\frac{dy}{dt} = f(t, y).$$

Ал n -ші ретті дифференциалдық теңдеулер жүйесі

$$\begin{cases} y_1' = f_1(t, y_1, y_2 \dots y_n) \\ y_2' = f_2(t, y_1, y_2 \dots y_n), \\ \dots \\ y_n' = f_n(t, y_1, y_2 \dots y_n) \end{cases}$$

мұндағы t - тәуелсіз айнымалы, тәртіп бойынша – бұл уақыт.

Дифференциалдық теңдеулер жүйесінің дербес шешімін (Коши есебі) табу үшін бастапқы шарттар: $y_1(t_0), y_2(t_0), y_3(t_0), \dots y_n(t_0)$ берілуі тиіс. Дифференциалдық теңдеулер жүйесінің алғашқы шарттары бойынша шешімі $y_1(t), y_2(t), y_3(t), \dots y_n(t)$ тәуелділіктеріне алып келеді.

Әрбір n -ші ретті айқын түрде берілген дифференциалдық теңдеулерді $y^{(n)}(t) = f(t, y, y', y'', \dots, y^{(n-1)})$ жаңа белгісіз $y_1 = y, y_2 = y', y_3 = y'', \dots y_n = y^{(n-1)}$ функцияларын енгізу арқылы n дифференциалдық теңдеулер жүйесіне түрлендіруге болады:

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ \dots \\ y_{n-1}' = y_n \\ y_n = f(t, y_1, y_2, \dots, y_n) \end{cases}$$

MATLAB-та дифференциалдық теңдеулер жүйесін сандық жолмен шешу үшін *ode45*, *ode23*, *ode113*, *ode15s*, *ode23s*, *ode23t* және *ode23tb* функциялары қолданылады. Бұл функциялар түрлі сандық әдістерді қолданады, мысалы, *ode45* - Рунге-Куттаның 4-ші және 5-ші реттік әдістері, *ode23* - Рунге-Куттаның 2-ші және 3-ші реттік әдістері. Ал *ode15s*, *ode23s*, *ode23t*, *ode23tb* функциялары дифференциалдық теңдеулердің қатты жүйелерін шешуде қолданылады. Бұл дифференциалдық теңдеулердің шешімдері тәуелсіз айнымалылардың түрлі аралықтарында әр түрлі болады, яғни біріншісінде өте жоғары ауытқулар байқалса, келесісінде өте төмен болады.

Дифференциалдық теңдеулердің жүйелерін шешуде *ode45* функциясының қолданылуын келесідей мысалдан көруге болады. $Y = [y_1, y_2, \dots, y_n]$ және $F = [f_1, f_2, \dots, f_n]$ векторлары енгізілсін. Онда дифференциалдық теңдеулер жүйесі $Y' = F(t, Y)$ түрге ие болады.

Дифференциалдық теңдеулер жүйесін шешетін функцияны қолдану үшін $(F(t, Y))$ теңдеулер жүйесінің оң жақ бөлігін шешетін арнайы m -функциясын құру керек.

Функцияны шақыру үшін $[t, Y] = \text{ode45}('функция атауы', ts, Y0)$ қолданамыз. *Ode45* функциясы дифференциалдық теңдеулер жүйесін $ts = [tn, tk]$ уақыт аралығында $Y0$ бастапқы шартымен интегралдайды. Мұндағы бірінші параметр – ол дифференциалдық теңдеулер жүйесінің оң жағын есептейтін m -функцияның атауы. Кері қайтарылатын Y жиымының әр жолы t векторы арқылы анықталатын уақыттағы шешімі болады (бірінші бағанда – y_1 , ал екіншісінде - y_2 және т.т.).

Мысалы. *Ode45* u *ode15s* функцияларын қолданып,

$$\begin{cases} \frac{dy_1}{dt} = 5y_2 * \sin(t) \\ \frac{dy_2}{dt} = t + 1 - y_1 - t \cdot y_2 \end{cases}$$

түріндегі дифференциалдық теңдеулер жүйесін шешу керек. Алынған тәуелділік графигін әрбір терезеге жекелеп тұрғызайық.

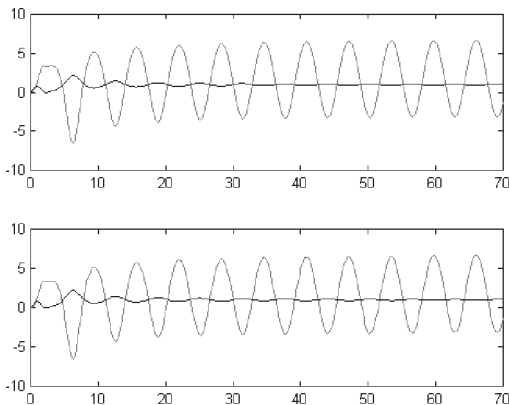
Дифференциалдык тендеулер жүйесінің оң жақ бөлігін шешуге арналған m -функциясы:

```
function Y=l7_p7f(t,y)
Y=[ 5*y(2)*sin(t);t+1-y(1)-t*y(2) ];
```

Сценарий мәтіні:

```
y0=[0,0];
[T,Y]=ode45('l7_p7f',[0,70],y0);
subplot(2,1,1)
plot(T,Y);
[T1,Y1]=ode15s('l7_p7f',[0,70],y0);
subplot(2,1,2)
plot(T1,Y1);
```

Алынған графиктер:



8. SIMULINK

8.1 Simulink туралы жалпы деректер

Simulink бағдарламасы *MATLAB* пакетіне қосымша болып табылады. *Simulink* бағдарламасын пайдаланып, модельдеу кезінде визуалды бағдарламалау ұстанымы жүзеге асады да, қолданушы экранда стандартты блок кітапханасынан құрылғының моделін және есептеулерін жасайды. Сонымен қатар, модельдеудің классикалық әдістерден өзгеше, тұтынушыға бағдарламалау тілін толық игерудің де, математиканың сандық әдістерін игерудің де қажеттілігі болмайды, тек қана өзінің кәсіби саласы бойынша компьютерде жұмыс істей білсе жеткілікті болады.

Simulink *MATLAB*-тың жетерліктей дербес аспабы болып табылады және онымен жұмыс істегенде *MATLAB*-тың өзін де, оның басқадай қосымшаларын да білуді аса қажет етпейді. Бір жағынан *MATLAB*-тың функцияларына және оның басқа құралдарының атқаратын қызметтеріне рұқсаттама ашық қалады және оларды *Simulink*-те пайдалануға болады. Пакеттердің құрамына кіретіндердің біразында *Simulink*-ке орнатылатын да құралдар бар (мысалы, *Control System Toolbox*-тың *LTI-Viewer* қосымшасы – пакеттің басқару жүйесін жасау үшін). Сонымен қатар, басқадай салалар үшін де қолданылатын блоктардың қосымша кітапханалары бар (мысалы, *Power System Blockset* – электротехникалық құрылғыларды модельдеу, *Digital Signal Processing Blockset* – сандық құрылғыларды жасау үшін арналған блоктар жиынтығы және т.б.).

Simulink-пен жұмыс істеу барысында қолданушының кітапханалық блоктарды жаңартуға мүмкіндігі бар және өзінің жеке блогын құрай алады, сонымен қатар блоктардың жаңа да кітапханаларын құрай алады.


Модельдеу кезінде қолданушы дифференциалдық теңдеулерді шешудің әдісін, сонымен қатар модельдеу уақытын өзгертудің әдісін өзі таңдай алады (тұрақты немесе айнымалы адыммен). Модельдеу кезіндегі үдерісті бақылау мүмкіндігі бар. Ол үшін *Simulink* кітапханасының құрамына кіретін арнайы құрылғы пайдаланылады. Модельдеу нәтижелері кесте немесе график түрінде болуы мүмкін.

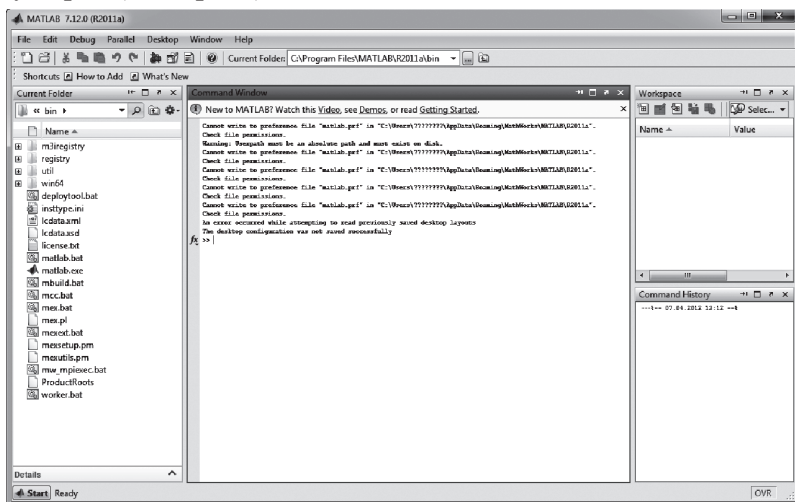
Simulink-тың және бір артықшылығы – ол *MATLAB* тілінде, тағы сол сияқты *C++*, *Fortran* және *Ada* тілдерінде жазылған шағын бағдарламалармен блоктар кітапханасын да толықтыруға көмектеседі.

8.2 Simulink-ті іске қосу

Бағдарламаны іске қосу үшін *MATLAB* пакетін ашу керек. *MATLAB* пакетінің негізгі терезесі 8.2.1-суретте көрсетілген.

MATLAB-тың бағдарламасының негізгі терезесін ашқаннан кейін *Simulink* бағдарламасын іске қосу керек. Оны келесідей үш әдістің біреуімен іске асыруға болады:

- *MATLAB* терезесінің құрал-саймандар жақтауынан  *-Simulink* батырмасын басу керек.
- *MATLAB* командалық терезесінің басты жолында *Simulink* деп жазып, пернетақтадағы *Enter* түймесін басу керек.
- *File* мәзіріндегі *Open* командасын орындап, модель файлын ашу керек (.mdl-файл).



8.2.1-сурет. *MATLAB*-тың негізгі терезесі

Соңғы нұсқа модельге жаңа блоктарды қосуды қажет етпейтін, тек есептеулер ғана жүргізуге дайындап қойған модельдерді іске қосуда өте ыңғайлы. Ал бірінші және екінші әдістерді қолдану *Simulink* кітапханасының тарауларының терезесінің ашылуына әкеліп соғады. (8.2.2-сурет).

8.3 Simulink кітапханаларының тарауларын қарастыру

Кітапхана блок терезесі келесі элементтерден тұрады (8.2.2-сурет):

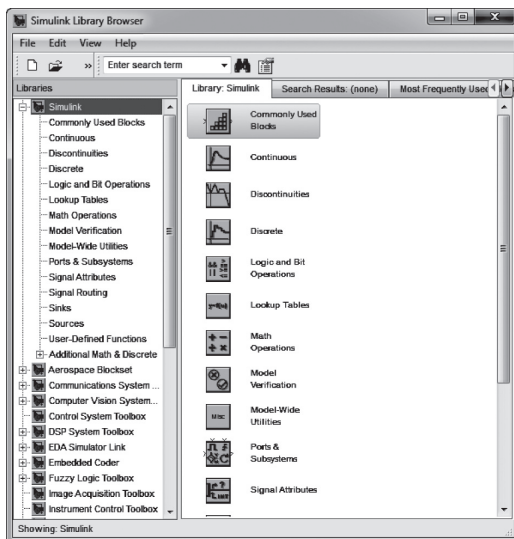
1. Тақырып, терезе атауы – *Simulink Library Browser*.
2. *File, Edit, View, Help* командалары бар мәзір.
3. Құрал-саймандар жақтауы ең жиі пайдаланылатын командалар.
4. Кітапхана тарауларының тізімі тармақтар түрінде орналастырылған.
5. Кітапхана тарауларының (кітапхана тарауының немесе блоктарының тізіміне енгізілген) мазмұны.

6. Атқарылатын қызметті көрсететін көмекші күй жолы.

8.2.2-суретте *Simulink*-тің негізгі кітапхана қоры (терезенің сол жақ бөлігінде) және оның тараулары (терезенің оң жақ бөлігінде) көрсетілген.

Simulink кітапханасы келесі тараулардан тұрады:

- *Commonly Used* – әдеттегі қолданыстағы блоктар;
- *Continuous* - сызықтық блоктар;



8.2.2-сурет. *Simulink* кітапханасының тарауларының терезесі

- *Discontinuities* – сызықсыз блоктар;
- *Discrete* – дискреттік блоктар;

– *Logic and Bit Operations* – логикалық және биттік іс-әрекеттер;

– *Lookup Tables* – сәйкестік кестелер;

– *Math Operations* – математикалық іс-әрекеттер;

– *Model Verification* – модельді тексеру;

– *Model-Wide Utilities* – кестелер;

– *Ports & Subsystems* – порттар мен ішкі жүйелер;

– *Signal Attributes* – сигнал параметрлері;

– *Signal Routing* – сигнал бағыты;

– *Sinks* – тіркеу құрылғылары;

– *Sources* – әсер мен сигналдар көзі;

– *User-Defined Functions* – қолданушы функциялары;

– *Additional Math & Discrete* – қосымша математикалық және дискреттік блоктар.

Simulink кітапханасында тараулар тізімі тармақ түрінде көрсетілген, онымен жұмыс істеу ережесі ондай тізім үшін жалпы болып келеді:

- Тармақтың жиналған түйіндісінің пиктограммасы «+» символынан тұрады, ал ашық түрдегі пиктограмма «-» символынан тұрады.

- Тармақ түйіндісін ашып немесе бүктеу үшін тышқанның сол жақ түймесін пиктограммада шертсе болғаны (ТСЖТ).

Лайықты кітапхана тарауын таңдағанда терезенің оң жақ бөлігінде оның мазмұны көрінеді. (8.3.1-сурет).



8.3.1-сурет. Кітапхана тарауларының блоктар жиынтығының шолушы терезесі

Тереземен жұмыс істеу үшін мәзірде жиналған командалар қолданылады. Кітапхана шолушысының мәзірі келесі пункттерден тұрады:

- *File (Файл)* — кітапхана файлдарымен жұмыс істеу.
- *Edit (Редакторлау)* — блоктарды қосу және оларды (қызметіне қарай) іздестіру.
- *View (Көрініс)* — интерфейс элементін көрсетуді басқару.
- *Help (Анықтама)* — анықтама терезесін кітапхана шолушысы бойынша шығару.

Шолушымен жұмыс істеу кезінде құрал-сайман жақтауындағы пернелерді қолдануға болады (8.3.2-сурет).




8.3.2-сурет. Кітапхана тарауының шолушысының құрал-сайман жақтауы

Құрал-сайман жақтауының пернелері келесідей қызметтер атқарады:

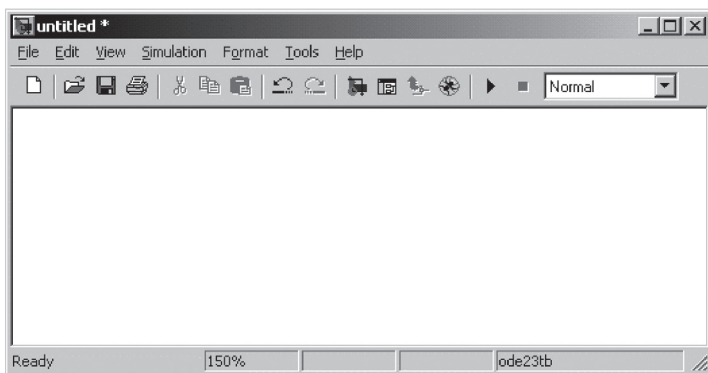
1. Жаңадан *S*-модель жасау (жаңа модель терезесін ашу).
2. Қолданыстағы *S*-модельдердің біреуін ашу.
3. Шолушының сипаттама терезесін өзгерту. Айтылған түйме құрайшының терезесінің кескінін «барлық терезе үстінен» режимін орналастырады. Түймені қайта басса бұл режимді кері қарай қайтарады.
4. Аталуына байланысты блоктарды іздеу (атаудың бірінші әріптерінен). Блок ізделініп табылғаннан кейін, шолушының терезесінде кітапхананың тақырыпқа лайықты тарауы ашылады, ал блок ерекшеленеді. Егер келесідей атаумен блок табылмаса, түсініктеме терезесінде *Not found <блок атауы> (блок табылмады)* хабарламасы шығады.

8.4 Модель құрастыру

Simulink ортасында модель құрастыру үшін келесі әрекеттерді жасау керек:

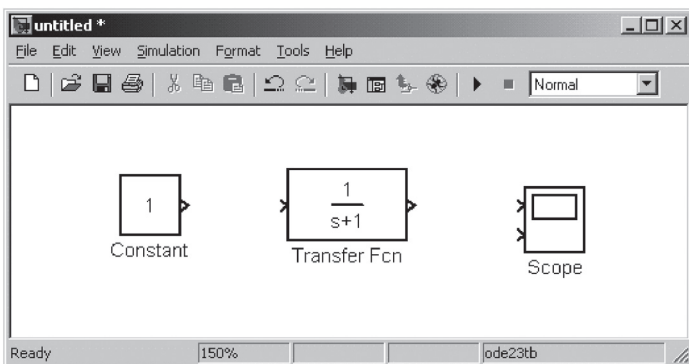
1. Жаңа модель файлын *File/New/Model* командалары арқылы жасау, немесе  батырмасын басу арқылы жасауға болады. 8.4.1-суретте модель терезесін жасау көрсетілген.

2. Модель терезесінде блоктарды орналастыру. Ол үшін лайықты кітапхана тарауын ашу керек (мысалы, *Sources*). Одан әрі мегзегішпен керекті блокты көрсетіп, тышқанның сол жақ батырмасын басып, ұстап тұрған күйі дайын терезеге блокты ауыстырып алып келу керек. 8.4.2-суретте модель құрамындағы блок терезесі көрсетілген.



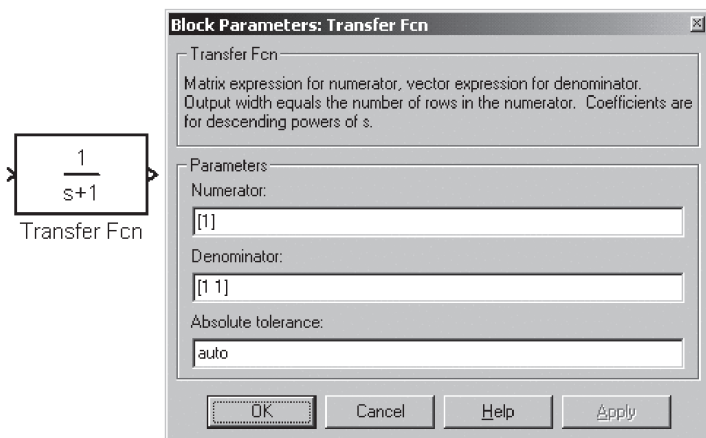
8.4.1-сурет. Модельдің бос терезесі

3. Блокты жою (немесе өшіру) үшін блокты таңдап алу керек (мегзегішті оның суретіне апарып, тышқанның сол жақ батырмасын басу керек), ал одан кейін пернетақтада *Delete* батырмасын басу керек. Блоктың өлшемдерін өзгерту үшін бір блокты таңдап алып, мегзегішті блоктың бұрышына орнатып, сол жақ батырманы басып, блоктың өлшемін өзгерту керек (мегзегіш ол кезде екі жақты болып кетеді).



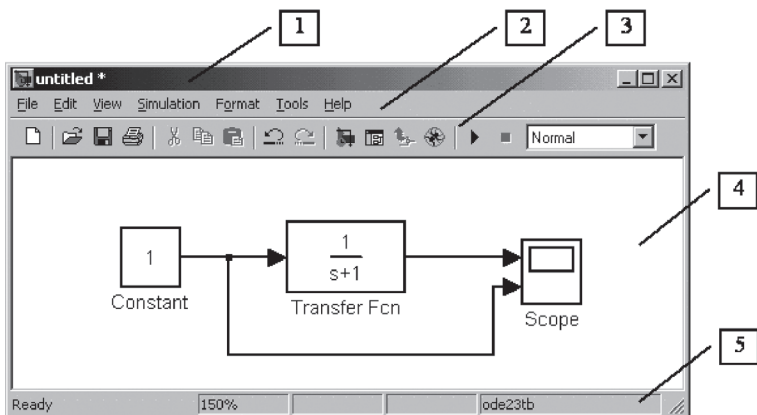
8.4.2-сурет. Модель құрамындағы блок терезесі

Бұдан әрі, егер қажет болса, “үнсiздік” бағдарламасымен қойылған блок параметрлерін өзгерту керек. Ол үшін мегзегішпен блок суретіне сілтеп жіберіп, тышқанның сол батырмасымен екі қайтара шерту керек. Айтылған блоктың параметрін өзгерту терезесі ашылады. Сандық параметр берілгенде, есте болатын жайт, ондық бөлгіш орнына үтір емес, нүкте қолдануы тиіс. Енгізгеннен кейін *OK* батырмасын басып, терезені жабу керек. 8.4.3-суретте мысал ретінде беріліс функциясын модельдеуші блок және айтылған блоктың параметрін күйге келтіретін терезе көрсетілген.



8.4.3-сурет. Беріліс функциясын модельдеуші блок және блоктың параметрін күйге келтіретін терезе

4. Сұлбада барлық блоктарды орнатқаннан кейін, керекті кітапханалардан сұлбадағы элементтердің қосылыстарын орындау керек. Блоктарды қосу үшін мегзегішпен блоктың шығысына апарып, кейін тышқанның сол жақ батырмасын жібермей басып тұрып, басқа блоктың кірісіне дейін сызық жүргіземіз. Одан кейін батырманы жіберу керек. Егер де блоктар дұрыс қосылса, бағдардың бейнесі блокқа кіру кезінде түсін өзгертеді. Нүктелерді айыру үшін нүкте қосушы сызыққа мегзегішті апарып, тышқанның оң жақ батырмасын басып сызықты созу керек. Сызықты өшіру немесе жою үшін керекті сызықты таңдап алып, содан соң *Delete* батырмасын басу керек. 8.4.4-суретте модель сұлбасында блоктар арасындағы байланыстар көрсетілген.



8.4.4-сурет. Модель сұлбасы

5. Есептеу сұлбасын құрғаннан кейін оның атауы мен папкасын көрсетіп және сұлба терезесінде көрсетілген *File/Save As...* пунктін таңдау арқылы файл күйінде сақтауымыз қажет. Есте сақтарлық жайт - ол файлдың атауы 32 символдан аспауы керек және де атаудың басы әріптен басталуы керек, ол кириллицаларды және арнайы таңбаларды қабылдамайды. Бұл талаптар файл жолына және сақтауға арналған папкаларға қатысты. Сұлбаны өзгерту кезінде *File/Save...* пунктін пайдалануға болады. *SIMULINK* бағдарламасын қайталап қосу кезінде сұлбаны *File/Open...* мәзірі, шолушы кітапханасы немесе *MATLAB*-тың негізгі терезесі арқылы іске қосуға болады.

8.5 Модель терезесі

Модель терезесі келесідей элементтерден тұрады (8.4.4-сурет):

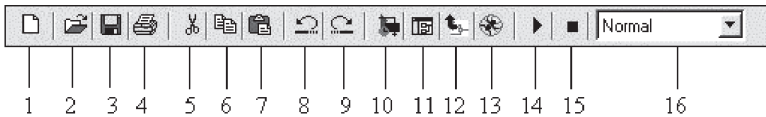
1. Терезенің атауы бар тақырып. Жаңа ашылған терезеге лайықты нөмірі бар *Untitled* аты қойылады.
2. *File, Edit, View* және де т.б. командалары бар мәзір.
3. Құрал-саймандар жақтауы.
4. Модель сұлбасын құрауға арналған терезе.
5. Модель күйін қарастыратын күй жолы.

Терезе мәзірінде модельді өзгерту, есеп үдерісін басқару және күйге келтіру, файлдармен жұмыс істеу және т.б. командалар бар:

- *File (Файл)* — Модель файлдарымен жұмыс
- *Edit (Редакторлау)* — Модельді өзгерту және блоктарды іздеу.
- *View (Көрініс)* — Интерфейс элементтерінің көрсетілімін басқару.
- *Simulation (Модельдеу)* — Модельдеу мен есептеу үдерісін басқару үшін күйге келтіруді беру.
- *Format (Қалыптау)* — Модель мен блоктың біртұтас сыртқы бейнесін өзгерту.
- *Tools (Құрал-саймандар)* — Модельмен жұмыс істеу үшін арнайы құрылғыларды қолдану.
- *Help (Анықтама)* — Анықтама терезесін шығару жүйесі.

8.5.1 Модельмен жұмыс істеу

Модельмен жұмыс істеу үшін құрал-саймандар жақтауындағы пернелерді қолдануға болады (8.5.1-сурет).



8.5.1-сурет. Құрал-сайман жақтауының модель терезесі

Құрал-саймандар жақтауындағы пернелер келесідей атауларға ие:

- *New Model* — жаңа модель терезесін ашу.
- *Open Model* — .mdl-файлы бар файлды ашу.
- *Save Model* — .mdl-файлды дискіге сақтау.
- *Print Model* — модельдің блок-диаграммасын басып шығару.
- *Cut* — уақытаралық сақтау буферіне модельдің белгіленген бөлігін кесіп алу.
- *Copy* — уақытаралық сақтау буферіне модельдің белгіленген бөлігін көшіру.
- *Paste* — уақытаралық сақтау буферіндегі бар деректі модель терезесіне орналастыру.
- *Undo* — алдыңғы өзгерту амалын болдыртпау.
- *Redo* — өзгертілмей қалған амалдың шешімін қалпына келтіру.

– *Library Browser* — кітапхана шолушысының терезесін анықтау.

– *Toggle Model Browser* — модель шолушысының терезесін ашу.

– *Go to parent system* — ішкі жүйеден деңгейі жоғары иерархиялық жүйеге өту. Бұл команда тек ішкі жүйе ашық болса ғана орындалады.

– *Debug* — модельді дұрыстаушыны іске қосу.

– *Start/Pause/Continue Simulation* — модельді іске қосу (*Start* командасы); модельді іске қосқаннан кейін пернеде **||** символы пайда болады, оған *Pause* командасы лайықты, ал модельдеуді жалғастыру үшін сол пернеден тағы бір қайтара шерту керек, оған *Continue* командасы лайықты.

– *Stop* — модельдеуді аяқтау. Бұл перне модельдеуді бастағаннан кейін және *Pause* командасы орындалғаннан кейін де қолжетімдік дәрежеде болады.

– *Normal/Accelerator* — жай/күшейтілген есептеу режимі. Бұл құрал *Simulink Performance Tool* қосымшасы орнатылған жағдайда қолжетімдік дәрежеде болып саналады.

Модель терезесінің төменгі бөлігінде құрал-саймандар жақтауындағы пернелерге және де интерфейстегі мегзегіштің лайықты тұрған элементіне сай мәзір пункттеріне қысқаша түсініктеме беретін күй жолы орналасқан. Бұл мәтіндік аймақ *Simulink Ready* (дайын) немесе *Running* (орындалуы) күйлерінің индикациясы үшін де пайдаланылады. Сонымен қатар күй жолында

– блок-диаграммасының кескінінің шамасы (*пайыздарда, бастапқы мәні 100% тең*),

– модельдеу сеансының аяқталу дәрежесін көрсететін индикатор (*модель іске қосылғаннан кейін пайда болады*),

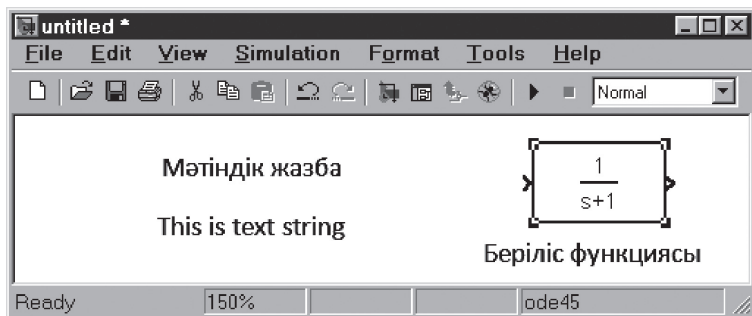
– модельдеу уақытының ағымдық мәні (*модель іске қосылғаннан кейін пайда болады*),

– модель күйін есептеуде қолданылатын алгоритм (*шығару әдісі*) көрсетіледі.

8.6 Модель құрастыру мен оған өзгерістер енгізудің негізгі амалдары

8.6.1 Мәтіндік жазбаларды енгізу

Модельдің көрнекілігін арттыру үшін мәтіндік жазбаларды қолдану тиімді. Жазба құру үшін мегзегішті тышқанмен жазбаға апарып, екі рет сол жақ батырмамен шерту керек. Осыдан кейін енгізу мегзегіші мен тік бұрышты рамка пайда болады. Блок моделінің жазбаларын да жоғарыда аталғандай жолмен өзгертуге болады. 8.6.1-суретте мәтіндік жазба мен функция блок жазбасы көрсетілген.





8.6.1-сурет. Мәтіндік жазба мен *Transfer Function*-дегі мәтіндердің өзгертілуі

8.6.2 Объектілерді белгілеу

Қандай да бір элементпен белгілі бір әрекет жасау үшін (блок-тар, қосушы сызықтар, жазбалар) ол элементті ең бірінші белгілеп алу керек. Ол үшін тышқан мегзегішін керекті объектіге апарып, сосын тышқанның сол батырмасын басу керек. Объект белгіленеді. Оны объект бұрыштарындағы маркерлер анық көрсетеді (8.6.1-сурет). Сонымен қатар, бір емес, бірнеше объект белгілеуге де болады. Ол үшін тышқан мегзегішін объектілер тобының жанына орнатып, тышқанның сол жақ батырмасын басып және оны жіберместен тышқанды жылжыту керек. Пунктирленген рамка пайда болады, тышқанды қозғалтқан сайын олардың өлшемдері өзгеріп отырады. Рамкаға кірген объектілер белгіленген болып


табылады. Барлық объектілерді белгілеу үшін *Edit/Select All* командасы пайдаланылады. Объектілерді белгілегеннен кейін, оларға *Windows* бағдарламасының стандартты жұмыс амалдарын қолдана отырып, уақытаралық сақтау буферіне көшіруге немесе орын ауыстыруға, буферден шығаруға, сонымен қатар өшіруге немесе жоюға және басқадай да әрекеттер жасауға болады.

8.6.3 Уақытаралық сақтау буферінен объектілерді көшіру және орын ауыстыру

Объектіні буферден көшіру үшін оны бірінші белгілеу қажет, одан кейін *Edit/Copy* командасын орындап немесе құрал-сайман жақтауынан  құралын пайдаланамыз. Буферге объектіні кесіп алу үшін оны біріншіден белгілеп алу керек, ондан соң *Edit/Cut* командасын орындап немесе құрал-сайман жақтауынан  құралы пайдаланылады. Берілген операцияларды орындау барысында ескеретін жайт, объектілер MATLAB-тың жеке буферіне орналастырылады. *Edit/Copy model to Clipboard* командаларын пайдалану модельдің графикалық суреттерін *Windows*-тың буферіне көшірілуін жеңілдетеді, сонымен қатар оны басқа бағдарламаларға да тиімді пайдалануға мүмкіндік береді.

Көшірмені басқа да жолмен орындауға болады. Ол үшін тышқанның оң жақ батырмасын басып, оны жібермей объектінің орынын ауыстыруға болады. Осы кезде керекті жерге орналастыруға болатын объектінің көшірмесі пайда болады.

8.6.4 Уақытаралық сақтау буферінен объектілерді кірістіру

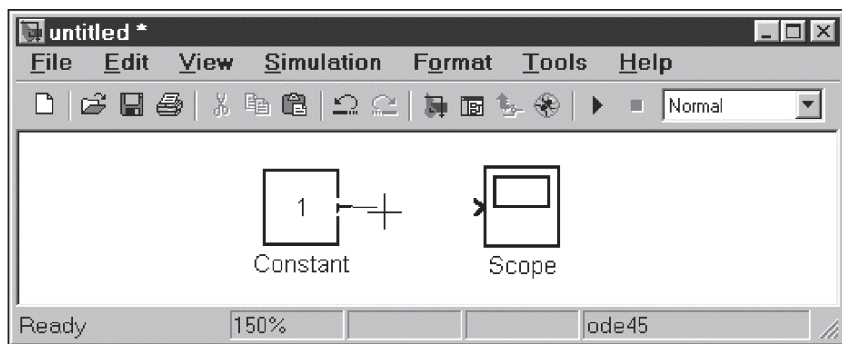
Объектіні буферден алып, орналастыру үшін мегзегішті орнататын жерге апарып, тышқанның сол жақ батырмасын басып, орнататын жерді белгілеуі тиіс, ал одан кейін *Edit/Paste* командасын орындау немесе құрал-сайман жақтауындағы  құралын пайдалану керек.

8.6.5 Объектілерді жою

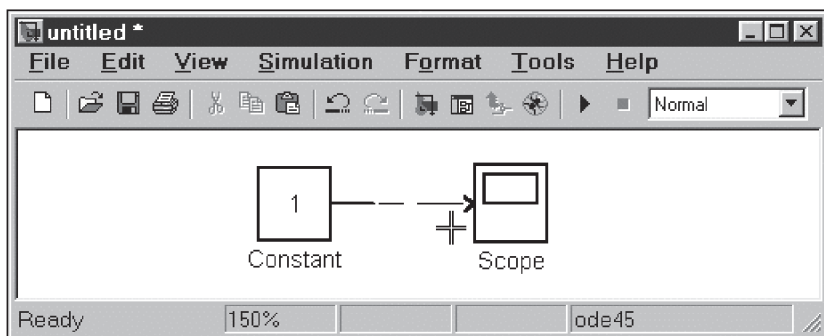
Объектіні жою үшін сол объектіні белгілеу керек, ал одан кейін *Edit/Clear* командасын орындап немесе *Delete* пернесі пайдаланылады. Ескертетін жайт, *Clear* командасы блокты айырбастау буферіне орналастырмастан жойып жібереді. Бірақ бұл амалды *File/Undo* командасымен болдырмауға да болады.

8.6.6 Блоктардың жалғануы

Блоктарды жалғау үшін тышқан мезгегішін блоктардың шығу порттарының біріне орналастыру керек. Тышқанның сол жақ батырмасын басып тұрып, мезгегішті керекті блоктың кіретін портына ауыстыру керек. Мезгегіш бұл кезде жіңішке сызықтардан тұратын крестке айналып кетеді (8.6.3-сурет). Сызықты құрап болғаннан кейін, тышқанның сол жақ батырмасын жіберуге болады. Сонымен блоктың кіру портындағы бағдар жалғануды растайтын жуан сызықпен белгіленеді. Сызықты белгілеу блокты белгілегендей, тышқанның сол жақ пернесін бір рет шерту арқылы іске асырылады. Сызық бойындағы жалғаудағы түйіншектердегі қара маркер сызықтың белгіленгендігін көрсетеді.

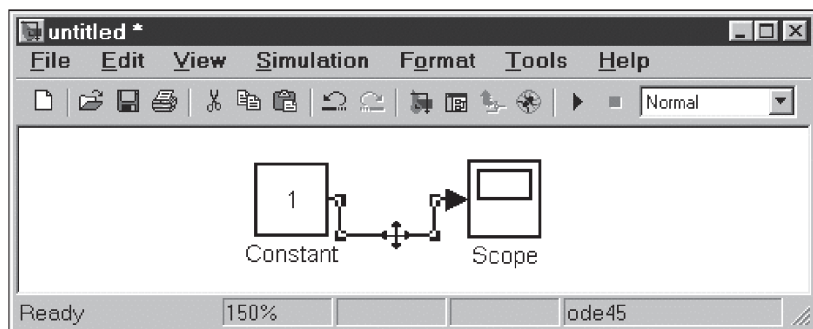


8.6.2-сурет. Жалғау құрылуының бастамасы



8.6.3-сурет. Жалғау құрылымының аяғы

Жалғау сызығының ілгішін құру блоктың орнын ауыстырылуы сияқты орындалады. Жалғау сызығы белгіленеді де, содан соң керекті бөлігі орын ауыстырылады (8.6.4-сурет).

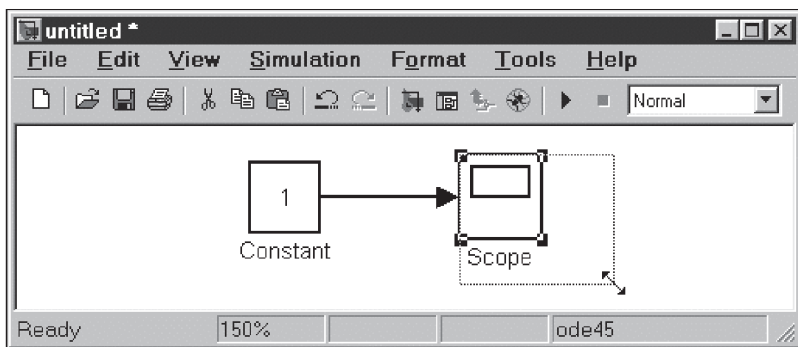


8.6.4-сурет. Жалғау сызығында ілгішті құру

Жалғануды жою басқа да объектілердегідей орындалады (8.6.5-сурет).

8.6.7 Блоктардың өлшемдерін өзгерту

Блоктың өлшемдерін өзгерту үшін оны белгілеп, одан соң тышқан мезгегішін блок бұрыштарындағы маркерлердің біріне орнату керек. Мезгегіш екі жағы бар бағдарға айналғаннан кейін, тышқанның сол жақ батырмасын басып және блок кескінін созып (немесе сығып) жіберу керек (8.6.5-сурет). Блок жазбаларының өлшемдері бұл жағдайда өзгеріссіз қалады.





8.6.5-сурет. Блок өлшемдерін өзгерту

8.6.8 Блоктардың орын ауыстырылуы

Модельдің кез келген түрін орын ауыстыруға болады, оны тек белгілесе болғаны, содан кейін тышқанның сол жақ батырмасын жіберместен орын ауыстыруға болады. Егер де кіру және шығу порттарына жалғаулық сызық жүргізілген болса, олар бір-бірінен ажырамайды, олар тек ұзындығы бойынша қысқартылады немесе ұзартылады. Жалғаулықтарда бір-бір кіріс және шығыс порттары бар блок қоюға болады. Ол үшін оларды талап етілген жерінде жалғаулық сызықтармен орнату керек.

8.6.9 Undo және Redo командаларының қолданылуы

Бағдарламаны игеру үдерісінде тұтынушы қайталанбайтындай көрінетін жағдайларда өзіне керекті әрекеттерді пайдалана алады (мысалы, модель бөлігін байқаусыз жойып алу, көшіру және т.б.). Осындай кезде *Undo* командасын пайдаланған жөн – ол соңғы амалды болдыртпайды. Бұл команданы құралдар жақтауынан  батырмасы арқылы немесе мәзірден *Edit* командасы арқылы шақыруға болады. Болдырмаған амалды кері қайтару үшін *Redo* командасы қызмет етеді ( құралы).

8.6.10 Объектілерді қалыптау

Format (контекстік мәзірдегі сияқты, объектіде тышқанның оң жақ батырмасын басу арқылы) мәзірінде блоктарды қалыптаудың командалар жиынтығы орналасқан. Қалыптау командасы бірнеше топтардан тұрады:

1. Жазба кескінінің өзгеруі:

– *Font* — жазбалық және мәтіндік блоктардың қаріптерін қалыптау.

– *Text alignment* — мәтіндік жазбадағы мәтінді түзету.

– *Flip name* — блок астындағы жазбалардағы орын ауыстыру.

– *Show/Hide name* — блок астындағы жазбаларды жасыру немесе кескіндеу.

2. Блок кескінінің түсін өзгерту:

– *Foreground color* — блоктарды белгілеу сызықтарының түсін өзгерту.

– *Background color* — белгіленген блоктардың реңдік түсін өзгерту.

– *Screen color* — модель терезесінің бүкіл реңдік түсін өзгерту.

3. Блоктың және оның көрініс жағдайының өзгерісі:

– *Flip block* – симметриялық вертикаль осі бойынша кері кескіні.

– *Rotate block* – блокты сағат тілі бағытымен 90⁰-қа бұру.

– *Show drop shadow* — блоктың көлеңкесін көрсету.

– *Show port labels* — порттардың белгісінің көрсетілімі.

4. Өзге қондырғылар:

– *Library link display* — кітапханамен байланысты көрсету.

– *Sample time colors* — уақыттық индикация блогінің түсін өзгерту.

– *Wide nonscalar lines* — скалярлық емес сызықтардың үлкеюі немесе кішіреюі.

– *Signal dimensions* — сигналдардың өлшемін көрсету.

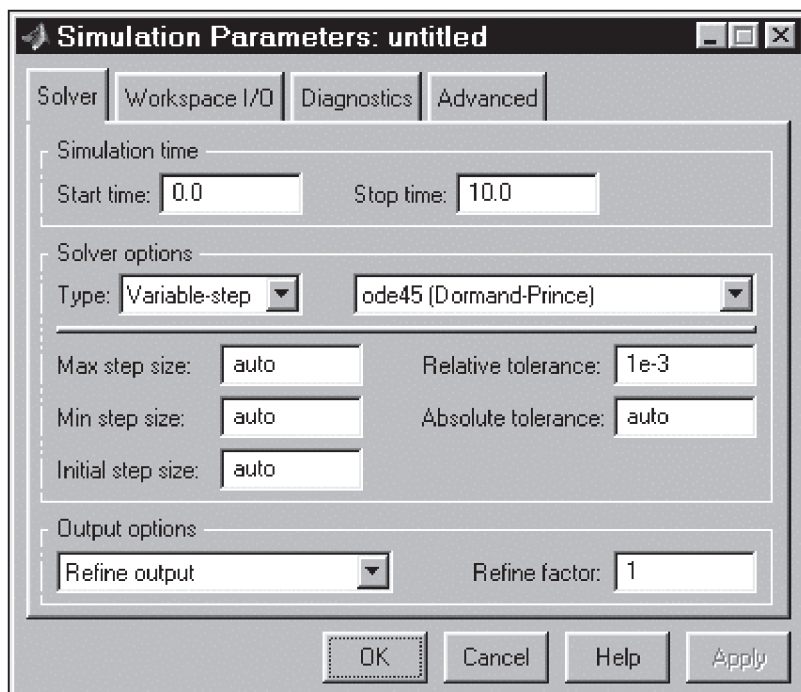
– *Port data types* — порттар түрлері туралы деректер көрсету.

– *Storage class* — жад класы. Бұл *Real-Time Workshop* жұмыс істеп тұрғанда орнатылатын параметр.

– *Execution order* — блоктың реттік нөмірін көрсету.

8.7 Есептеу параметрлерін және оның орындалуын орнату

Есептеуді бастамас бұрын ең бірінші есептің параметрлерін енгізу керек. Есептің параметрлерін енгізу мәзірді басқару жақтауында *Simulation/Parameters* орналасқан. 8.7.1-суретте басқару жақтауының көрсетілімі берілген.



8.7.1-сурет. Басқару жақтауы

Есептің параметрлерін күйге келтіру терезесі 4 қосымша беттен тұрады:

- *Solver (Есептеу)* — модельді есептеудегі параметрлерді орнату.

- *Workspace I/O (Жұмыс аймағына деректерді енгізу/шығару)* — *MATLAB*-тың жұмыс аймағында дерек алмасудағы параметрлерді орнату.

– *Diagnostics (Диагностика)* — диагностикалық режимнің параметрлерін таңдау.

– *Advanced (Қосымша)* — қосымша параметрлерді орнату.

Модельді есептеу параметрлерін орнату *Solver* қосымша бетінде орналасқан басқару элементтері арқылы орындалады. Бұл элементтер үш топқа бөлінеді (8.7.1-сурет): *Simulation time* (моделдеу аралығы немесе басқаша айтқанда есептеу уақыты), *Solver options* (есептеу параметрі), *Output options* (шығыс параметрі).

8.7.1 Модельдің есептеу параметрлерін орнату

8.7.1.1 Simulation time

(Модельдену аралығы немесе есептеу уақыты)

Есептеу уақыты бастапқы (*Start time*) және соңғы (*Stop time*) есептеу уақытының аралығындағы мәнмен анықталады. Бастапқы уақыт, ережеге сай, нөлге тең деп беріледі. Соңғы уақыт аймағы шешіліп жатқан есептің шартына байланысты тұтынушымен белгіленеді.

8.7.1.2 Solver options

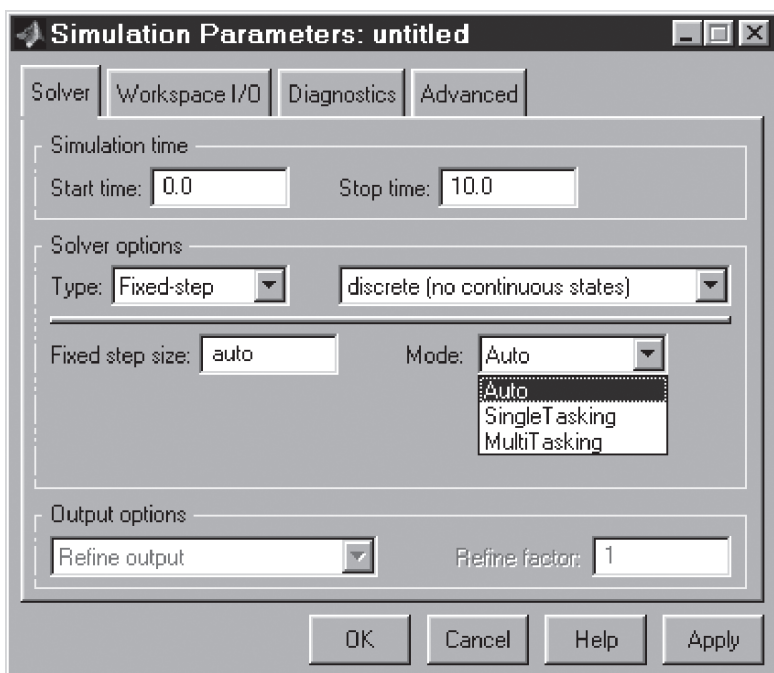
(есептеу параметрлері)

Параметрді таңдау кезінде модельдеу әдісін (*Type*) және жүйені есептеудің жаңа күй әдісін көрсету керек. *Type* параметрі үшін екі нұсқа тиімді – тұрақты (*Fixed-step*) немесе айнымалы (*Variable-step*) адыммен. Тәртіп бойынша, *Variable-step* үзіліссіз жүйені модельдеу үшін, ал *Fixed-step* дискреттік жүйе үшін қолданылады.

Есептеу әдістерінің жаңа күйінің тізімі бірнеше нұсқадан тұрады. Бірінші нұсқа (*discrete*) дискреттік жүйелерді есептеу үшін қолданылады. Қалған нұсқалары үзіліссіз жүйелерді есептеу үшін қолданылады. Бұл нұсқалар айнымалы (*Variable-step*)

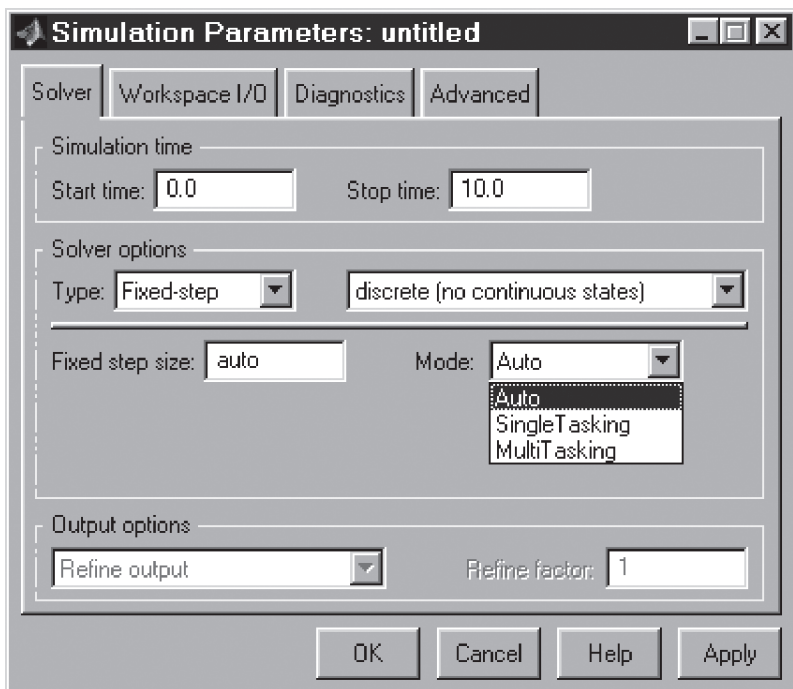
адым үшін де, бекітілген (*Fixed-step*) адым үшін де әрқалай болады және дифференциалдық теңдеулер жүйелерін шешу әдістерінің бірі болып келеді.

Type-ты ашатын екі тізімінің астында ішіндегісі модельдеу уақытының таңдалған әдісіне байланысты өзгертін аймақ бар. *Fixed-step* таңдалған жағдайда жоғарыда айтылған аймақта модельдеу адымының шамасын көрсетуге мүмкіндік беретін мәтіндік өріс *Fixed-step size* пайда болады (8.7.2-сурет).



8.7.2-сурет. Есептеудегі тұрақталған адымды таңдап алуға *Solver* қосымша беті

Модельдеу адымының шамасы өз бетімен жүйе арқылы автоматты (*auto*) түрде орнатылады. Талап етілетін адымның шамасы *auto* мәнінің орнына сан түрінде енгізілуі мүмкін немесе есептелетін өрнек түрінде (жүйенің автоматты түрде орнататын басқа да параметрлеріне қатысты) берілуі мүмкін.



8.7.2-сурет. Есептеу адымын бекітуін таңдау кезіндегі *Solver*-дің қосымша беті

Fixed-step-ті таңдағанда есептеу режимін (*Mode*) беру керек. *Mode* параметрі үшін үш нұсқа қолжетімдік дәрежеде болады. Олар:

- *MultiTasking* (Көп есептік) – модельде бір уақытта жұмыс істеп тұрған ішкі жүйе болған жағдайда пайдалану қажет. Модельдің жұмыс нәтижесі осы ішкі жүйенің уақытша параметрлерінен тәуелді болады. Бұл режим жылдамдықтың үйлесімділігі мен блоктардың бір-біріне беретін сигналдарының дискреттілігін анықтауға мүмкіндік береді.

- *SingleTasking* (Бір есептік) - Модельдеудің жұмыс нәтижесіне тәуелсіз жеке бөліктеріне қатаң синхрондалу жетпейтін модельдер үшін қолданылады.

- *Auto* (Автоматты режимді таңдау) – *Simulink*-ке сигналдарды беруде жылдамдықтары әр түрлі болатын блоктары бар модельдер үшін қолданылады.

дерге *MultiTasking* режимін және бірдей жылдамдықпен әрекет ететін блоктары бар модельдерге *SingleTasking* режимін автоматты түрде орнатуға мүмкіндік береді.

Variable-step-ті таңдауда аймақта үш параметрді орнататын өріс пайда болады:

– *Max step size* – есептеудің ең үлкен адымы. Ол өз бетімен автоматты түрде орнатылады және бұл кезде оның мәні (*StopTime* — *StartTime*)/50-ге тең. Бұл мән көбіне аса үлкен болып келеді. Сондай-ақ көрсетілетін графиктері сынық (түзу емес) болып көрінеді. Бұндай жағдайда есептеудің ең үлкен адымын анық кейіпте беру қажет.

– *Min step size* – есептеудің ең кіші адымы.

– *Initial step size* – модельдеу адымының бастапқы мәні.

Үздіксіз жүйені айнымалы адым арқылы модельдеу кезінде есептеудің дәлдігін не салыстырмалы (*Relative tolerance*), не абсолюттік (*Absolute tolerance*) түрлерінің бірінде көрсету маңызды. Үнсіздікпен олар 10^{-3} -ға және *auto*-ға тең.

8.7.1.3 Output options (шығыс параметрлері)

Solver қосымшасының төменгі жағында модельдейтін жүйенің сыртқа берілетін сигналдарын шығарудың күйге келтірулері беріледі (*Output options*). Бұл параметр үшін төменде берілген үш нұсқаның біреуін таңдап алуға болады:

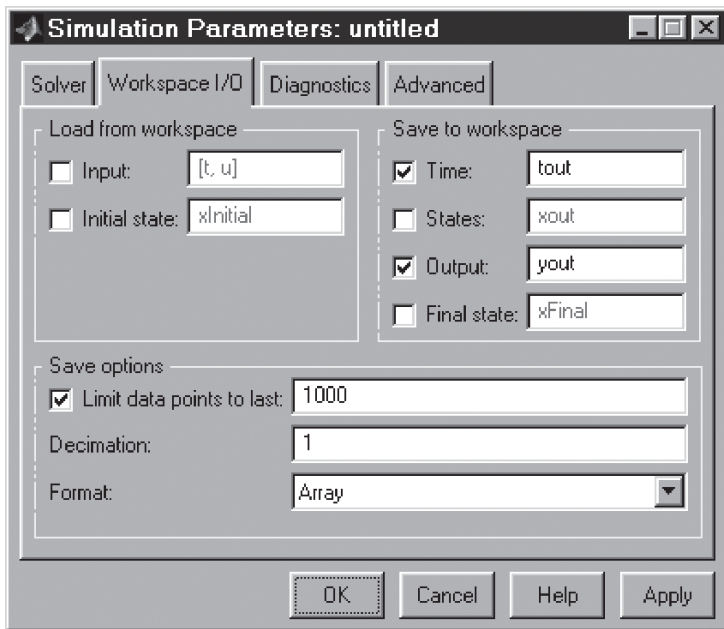
– *Refine output* (түзетілген шығыс) – модельдеу уақытының және *To Workspace* блогы арқылы MATLAB жұмыс аймағында сақталған сигналдардың тіркелу дискреттілігін өзгертуге мүмкіндік береді. Дискреттіліктің мәні оң жақта орналасқан *Refine factor* редакторының тармағында қойылады. Үнсіздік жағдайында *Refine factor* мәні 1-ге тең болады да, бұл тіркеу адымы $D \ t=1$ екендігін көрсетеді. Егер *Refine factor* мәнін екі деп берген жағдайда әр екі сигнал тіркеледі, үшті берген жағдайда, үшінші сигнал тіркеледі және т.т. *Refine factor* параметрлерінің мәні тек оң және бүтін сандар болады.

– *Produce additional output* (қосымша шығыс) — берілген уақыт аралығында модель параметрлерін қосымша тіркеуді қамтамасыз етеді; олардың мәндері редакциялау тармағына квадрат жақша ішінде орналасқан тізім ретінде енгізіледі. Бұл нұсқаны пайдаланған жағдайда негізгі тіркеу адымы $D t=1$ болады. *Output times* (уақытты шығару кезеңдері) тізімінде уақыт мәндері бөлшек мәндері және кез келген дәлдіктегі мәнге ие болуы мүмкін.

– *Produce specified output only* (тек берілген шығысты түзеді) — *Output times* өрісінде көрсетілетін тек белгіленген уақыт кезеңдерінде модель параметрлерінің шығысын көрсетеді.

8.7.2 Жұмыс аймағымен алмасу параметрін қою

MATLAB жұмыс аймағына модельдеудің аралық мәндері мен нәтижесін енгізуді және шығаруды басқаруға мүмкіндік беретін элементтер *Workspace I/O* қосымшасында орналасқан. (8.7.3-сурет).



8.7.3-сурет. Модельдеу параметрін қоятын *Workspace I/O* қосымшасының сұхбат терезесі

Қосымша элементтері 3 өріске бөлінген:

– *Load from workspace* (жұмыс аймағынан жүктеу). Егер *Input* (енгізу мағлұматтары) жалаушасы тұрса, онда оң жақта орналасқан мәтін аймағына белгілі мәндер қалыбын енгізуге болады да, бұл жағдайда MATLAB жұмыс аймағында есептеулер басталады. *Initial State* (алғашқы жағдай) жалаушасын қою онымен байланысқан мәтін алаңында айнымалының атын модельдеудің алғашқы күйінің параметрлері бар енгізудің мүмкіндігін береді.

– *Save to workspace* (жұмыс аймағына жазу) – MATLAB жұмыс аймағына сигнал мәндерін шығару режимін орнатуға және олардың аттарын беруге мүмкіндік тудырады.

– *Save options* (жазу параметрлері) – айнымалыларды жұмыс аймағына берген жағдайда тармақтар санын береді. Егер *Limit rows to last* жалаушасы қойылған жағдайда енгізу алаңына берілетін тармақтар санын көрсетуге болады (тармақшаны есептеу). Жалауша қойылмаған жағдайда барлық мағлұматтар беріледі. *Decimation* (алып тастау) параметрі жұмыс аймағына берілетін айнымалылардың жазылу адымын береді (*Solver* қосымшасындағы *Refine factor* параметрі сияқты). *Format* (берілген мәндер қалыбы) параметрі жұмыс аймағына берілетін мәндердің қалыбын белгілейді. Қолжетімдік дәрежедегілері – *Array* (жиым), *Structure* (құрылым), *Structure With Time* (қосымша «уақыт» өрісі бар құрылым).

8.7.3 Модельді диагноздаудың параметрін орнату

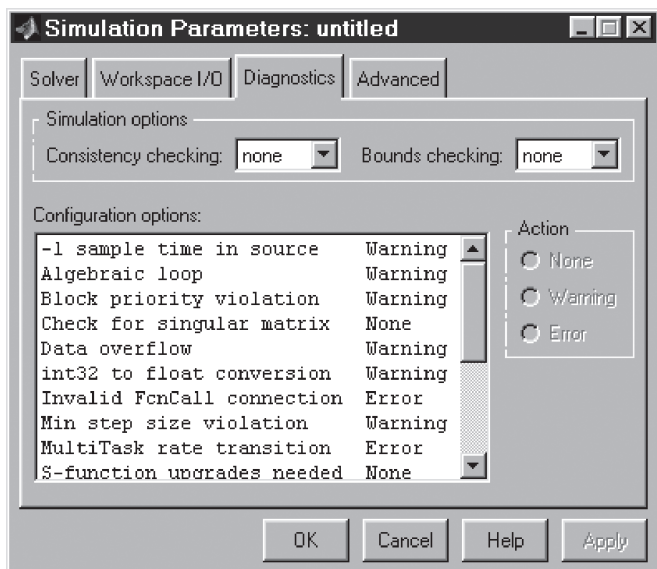
Diagnostics қосымшасы (8.7.4-сурет) MATLAB бұйрық терезесіндегі Simulink шығаратын диагностикалық хабарламалардың тізімін өзгертуге және модель диагностикасына қосымша параметрлер орнатуға мүмкіндік береді.

Модельдеу кезінде Simulink байқаған жобалаушының кірісуін талап ететін қателер мен ахуалды мәселелер туралы хабар MATLAB командалық терезесінде көрінеді. Осындай жағдайлардың тізімі мен оларға жауап беру реакциялары *Diagnostics* қосымшасының тізімінде берілген. Жасаушы *Action* өрісінде (егер тізімде берілген іс-әрекеттің бірі таңдап алынған жағдайда ғана оларға ену мүмкіндігі болады) ауыстырып–қосқыштың біреуін пайда-

лану арқылы әрқайсысына қажетті реакциялардың түрін көрсете алады:

- *None* — ескермеу,
- *Warning* – ескерту беру және модельдеуді жалғастыру,
- *Error* — қате туралы хабар беріп, модельдеу сеансын тоқтату.

Таңдап алынған реакция түрі іс-әрекет атауының қасындағы тізімде пайда болады.



8.7.4-сурет. Модельдеу параметрін қою терезесіндегі *Diagnostics* қосымшасы

8.7.4 Есептеуді жүргізу мен аяқтау

Есептеуді бастау мәзірдегі *Simulation/Start* пунктін немесе құрал-саймандар жақтауындағы ▶ құралы арқылы басталады. Есептеу үдерісін мәзір пунктіндегі *Simulation/Stop* немесе □ құралын таңдап алу арқылы ерте аяқтауға болады. Есептеуді (*Simulation/Pause*) арқылы тоқтатып, (*Simulation/Continue*) арқылы жалғастыруға болады.

Жұмысты аяқтау үшін файлда модельді сақтау, модель терезесін жабу, кітапханаларды шолушы терезе, сонымен қатар MATLAB пакетінің негізгі терезесін жабу қажет.

9. SIMULINK БЛОГЫНЫҢ КІТАПХАНАСЫ

9.1 Sources – сигнал көзі.

9.1.1 Constant тұрақты сигнал көзі

Қызметі: Деңгейі бойынша тұрақты сигнал түзеді.

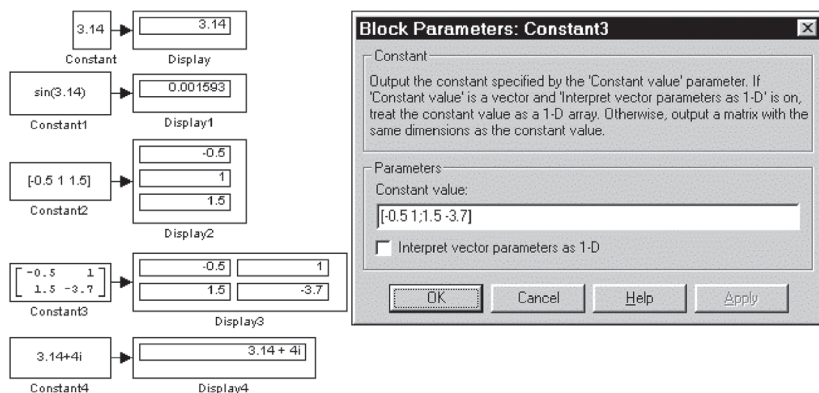
Параметрлері:

1. *Constant value* – Тұрақты шама.

2. *Interpret vector parameters as 1-D* – Параметр векторын бірөлшемдік ретінде түсіндіру (жалауша қойылған жағдайда) Бұл параметр Simulink кітапханасының көптеген блогында кездеседі.

Константа мәні нақты немесе комплекстік сан, вектор немесе матрица түрінде келуі мүмкін.

9.1.1-сурет *Display* сандық индикаторының көмегімен осы сигнал көзінің қолданылуы мен шығыс сигналының өлшенуін көрсетеді



9.1.1-сурет. *Constant* тұрақты сигнал көзі

9.1.2 Sine Wave синусоидалық сигнал көзі

Қызметі: Жиілігі, амплитудасы, фазасы және жылжуы бар синусоидалық сигналды түзеді.

Шығатын сигналдың блок арқылы қалыптасуы үшін екі

алгоритм қолданылуы мүмкін. Алгоритм түрі *Sine Type* параметрі арқылы анықталады (сигналдың түзілу тәсілі):

Time-based – ағымдағы уақыт бойынша.

Sample-based – модельдеу уақытының адымының шамасы бойынша.

9.1.2.1 Үздіксіз жүйе үшін ағымдағы уақыт мәні бойынша шығыс сигналының түзілуі

Шығыс сигналы көзінің шамасы бұл режимде

$$y = \text{Amplitude} * \sin(\text{frequency} * \text{time} + \text{phase}) + \text{bias}$$

өрнегіне сәйкес болады.

Параметрлері:

Amplitude – амплитуда.

Bias – тұрақты сигнал құраушысы.

Frequency (rads/sec) – жиілік (рад/с).

Phase (rads) – бастапқы фаза (рад).

Sample time – модельдеу уақытының адымы. Модельде уақыт көзі мен басқа компоненттердің жұмысын үйлестіру үшін қолданылады. Параметр келесідей мәндерді қабылдай алады:

0 (үзсіздік бойынша) – үздіксіз жүйелерді модельдеу кезінде қолданылады.

> 0 (оң мән) – дискреттік жүйелерді модельдеу үшін беріледі. Бұл жағдайдағы модельдеу уақытын шығыс сигналын уақыт бойынша кванттауымен түсіндіруге болады.

-1 – модельдеу уақытының адымын орнату алдыңғы блоктағыдай, берілген блокқа сигнал келетін жақтан болады. Бұл параметр Simulink-тің көптеген блоктарына беріледі.

Үлкен мәндер үшін есептеулер жүргізілгенде сигналдың шығыс мәндерін есептеу дәлдігі оларды дөңгелектеу кезінде кететін қателіктер әсерінен төмендеп кетеді.

9.1.2.2 Дискреттік жүйе үшін ағымдағы уақыт мәні бойынша шығыс сигналының түзілуі

Әрбір есептеудің келесі адым үшін сигнал көзінің шығыс мәнін анықтау алгоритмі

$$\begin{bmatrix} \sin(t + \Delta t) \\ \cos(t + \Delta t) \end{bmatrix} = \begin{bmatrix} \cos(\Delta t) & \sin(\Delta t) \\ -\sin(\Delta t) & \cos(\Delta t) \end{bmatrix} \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix},$$

өрнегі бойынша анықталады, мұндағы Δt – *Sample time* мәніне тең тұрақты шама.

Берілген режимдегі қателікті жуықтап, үлкен мағынадағы есептің дәлдігін кемітеді.

9.1.2.3 Модельдеу уақытының шамасы мен бір периодтағы есептеу адымының саны бойынша шығыс сигналының түзілуі

Бұл режимдегі шығыс сигналы

$$y = \text{Amplitude} * \sin[(k + \text{Number of offset samples}) / \text{Samples per period}] + \text{bias},$$

өрнегімен анықталады, мұндағы k – есептеудің ағымдағы адымының нөмірі.

Параметрлері:

Amplitude – амплитуда.

Bias – сигналдың тұрақты құраушысы.

Samples per period – синусоидалық сигналдың бір периодындағы есептеу адымының саны:

$$\text{Samples per period} = 2\pi / (\text{frequency} * \text{Sample time})$$

Number of offset samples – сигналдың бастапқы фазасы.

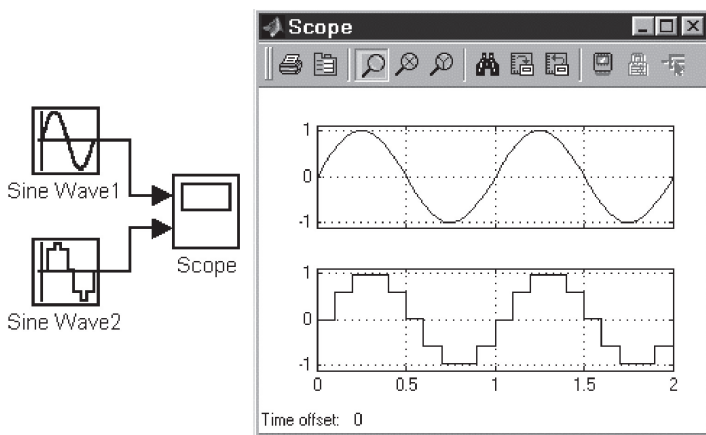
Модельдеу уақытының адым санымен беріледі:

$$\text{Number of offset samples} = \text{Phase} * \text{Samples per period} / (2\pi).$$

Sample time – Модельдеу уақытының адымы.

Берілген режимдегі қателік жинақталмайды, өйткені *Simulink* әр период үшін алдыңғы адымның санақ нөмірін нөлден бастайды.

9.1.2-суретте блоктың модельдеу уақытының әр түрлі мәндегі адымының қолданылуы көрсетілген (*Sine Wave 1* блогы үшін *Sample time*=0 және *Sine Wave 2* блогы үшін *Sample time*=0.1). Модельдеудегі шығыс сигналдарының графиктерін көрсету үшін виртуалдық осциллограф (*Scope*) қолданылған.



9.1.2-сурет. *Sine Wave* блогы

9.1.3 Ramp сызықтық өзгерістегі әсер көзі

Қызметі: Блок

$$y = \text{Slope} * \text{time} + \text{Initial value}$$

түріндегі сызықтық сигналды түзеді.

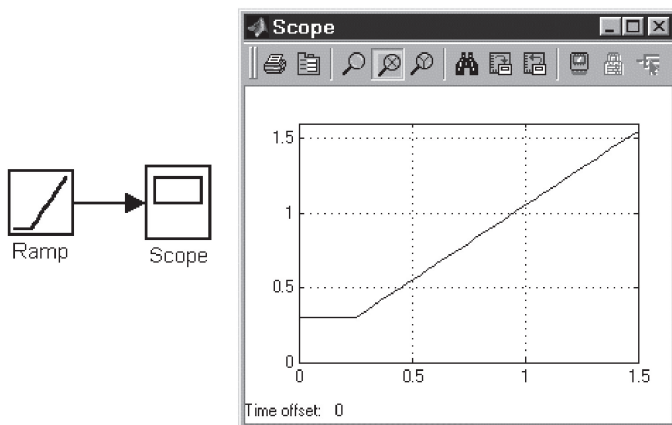
Параметрлері:

Slope — шығыс сигналының өзгеру жылдамдығы.

Start time — бастапқы сигналдың қалыптасу уақыты.

Initial value — блоктан шыққан сигналдың бастапқы шамасы.

9.1.3-суретте берілген блоктың қолданылуы көрсетілген.



9.1.3-сурет. *Ramp* блогы

9.1.4 Step сатылық сигнал генераторы

Қызметі: Сатылық сигналды түзеді.

Параметрлері:

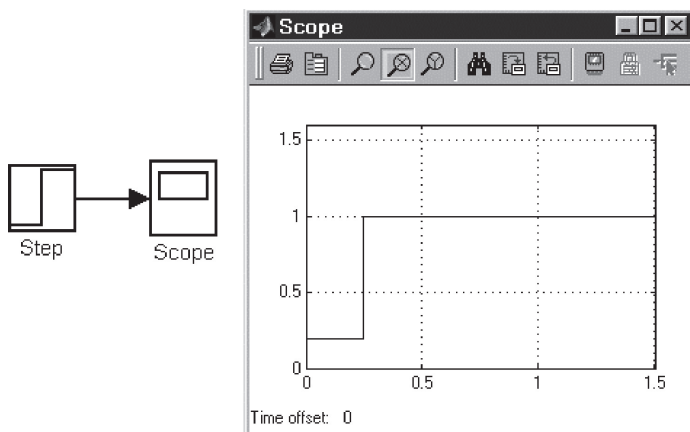
Step time – сигнал секіруінің басталу уақыты (с).

Initial value – сигналдың бастапқы мәні.

Final value – сигналдың соңғы мәні.

Сигналдың секірісі жоғары қарай да (соңғы мән бастапқы мәндерден үлкен), төмен қарай да (соңғы мән бастапқы мәндерден кіші) болуы мүмкін. Бастапқы және соңғы деңгейлердің мәндері оң да теріс те болуы мүмкін (мысалы, сигналдың өзгеру аралығы $[-5, -3]$).

9.1.4-сурет. Сатылық сигналдың қолданылуы көрсетілген.



9.1.4-сурет. Сатылық сигналды қолдану

9.1.5 Signal Generator сигналдар генераторы

Қызметі: Төмендегі периодтық сигналдардың біреуін түзеді:

sine — синусоидалық сигнал.

square — тік бұрыштық сигнал.

sawtooth — ара тәріздес сигнал.

random — кездейсоқ сигнал.

Параметрлері:

Wave form – сигнал түрі.

Amplitude – сигнал амплитудасы.

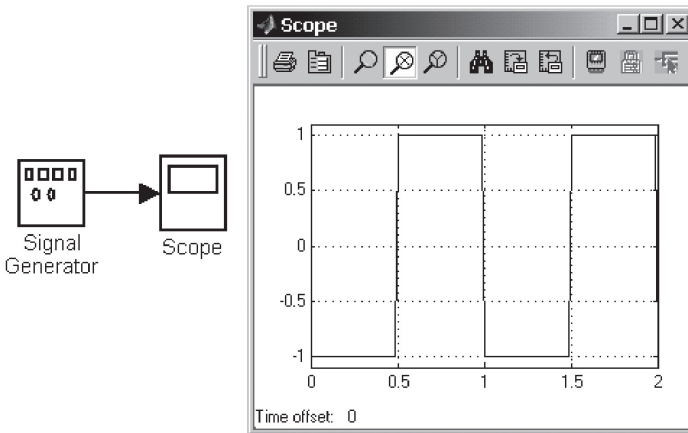
Frequency - жиілік (рад/с).

Units – жиілікті өлшеу шамасы. Ол екі мәнге ие бола алады:

- *Hertz* – Гц,

- *rad/sec* – рад/с.

9.1.5-суретте тік бұрышты сигнал көзін моделдеуде осы аталған сигнал көзінің қолданылуы көрсетілген.



9.1.5-сурет. Сигналдар генераторының блогы

9.1.6 Uniform Random Number біркелкі таратылатын кездейсоқ сигнал көзі

Қызметі: Біркелкі таратылатын кездейсоқ сигналды түзеді.

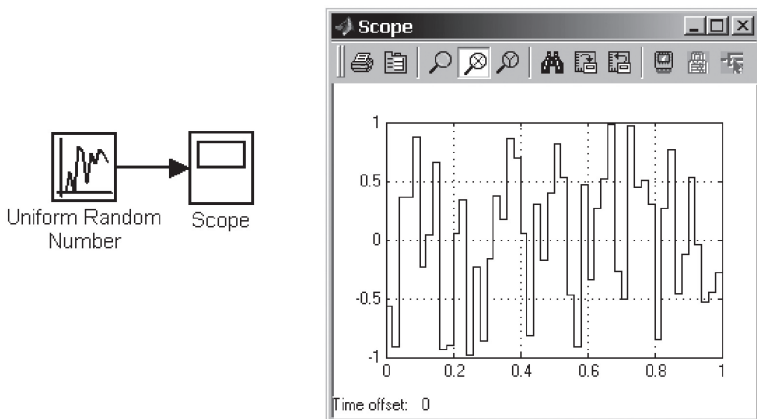
Параметрлері:

Minimum – сигналдың ең төменгі деңгейі.

Maximum – сигналдың ең жоғарғы деңгейі.

Initial seed – бастапқы мән.

9.1.6-суретте блоктың қолданылу мысалы мен шығыс сигналының графигі көрсетілген.



9.1.6-сурет. Біркелкі таралатын кездейсоқ сигналдың көзі

9.1.8.7 Random Number қалыпты таратылатын кездейсоқ сигнал көзі

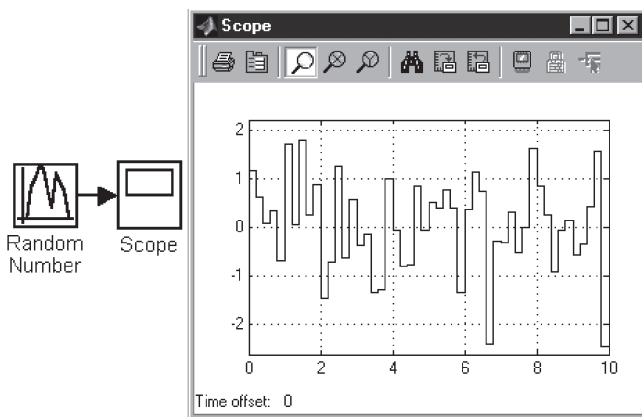
Қызметі: Қалыпты деңгейде таратылатын кездейсоқ сигналды түзеді.

Параметрлері:

Mean - сигналдың орташа мәні.

Variance – дисперсия (орташа квадраттық ауытқу).

Initial seed – бастапқы мән.



9.1.7-сурет. Қалыпты деңгейде таратылатын кездейсоқ сигнал көзі

9.1.8 Pulse Generator импульстік сигнал көзі

Қызметі: Тік бұрышты импульстік сигналдарды түзеді.

Параметрлері:

Pulse Type – сигналдардың қалыптасу әдістері. Ол келесідей екі мәнге ие болуы мүмкін:

- *Time-based* – ағымдағы өтіп жатқан уақыт.

- *Sample-based* – модельдеу уақытының шамасы мен есептеу адымының саны.

Amplitude — амплитуда.

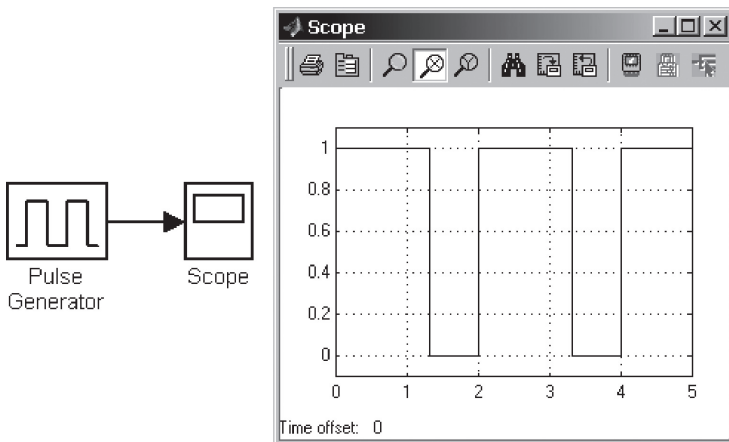
Period — период. *Time-based Pulse Type* үшін секунд немесе *Sample-based Pulse Type* үшін модельдеу адымы бойынша беріледі.

Pulse width — импульстің ені. *Time-based Pulse* үшін периодқа қатынасы % немесе *Sample-based Pulse Type* үшін модельдеу адымы бойынша беріледі.

Phase delay — фазалық бөгеліс. *Time-based Pulse Type* үшін секунд бойынша немесе *Sample-based Pulse Type* үшін модельдеу адымы бойынша беріледі.

Sample time — модельдеу уақытының адымы. *Sample-based Pulse Type* үшін беріледі.

9.1.8-суретте *Pulse Generator*-дың қолданысы көрсетілген.



9.1.8-сурет. Тік бұрышты импульстік сигнал көзі

9.1.9 Chirp Generator жиілігі сызықтық өзгертін генератор

Қызметі: Жиілігі сызықтық өзгертін синусоидалық тербелістерді түзеді.

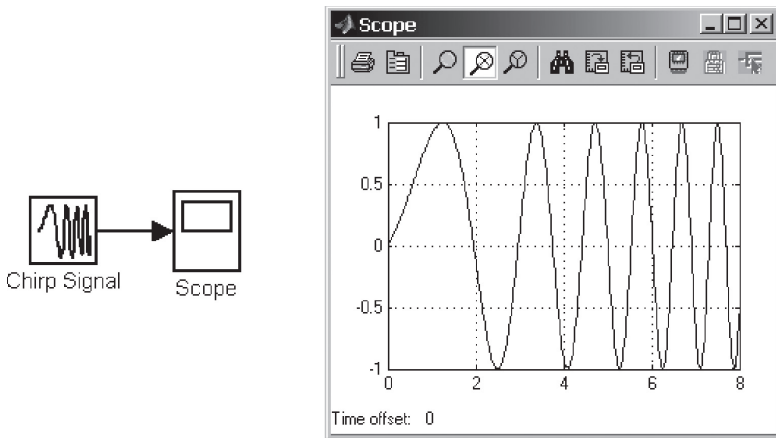
Параметрлері:

Initial frequency — бастапқы жиілік (Гц);

Target time — жиіліктің өзгеру уақыты (с);

Frequency at target time — жиіліктің соңғы мәні (Гц).

9.1.9-суретте блоктың қолданылу мысалы көрсетілген.



9.1.9-сурет. Жиілігі сызықтық өзгертін генератор

9.1.10 Band-Limited White Noise ақ шуыл генераторы

Қызметі: Жиілігі бойынша біркелкі таратылатын берілген қуаттылықтағы сигналды түзеді.

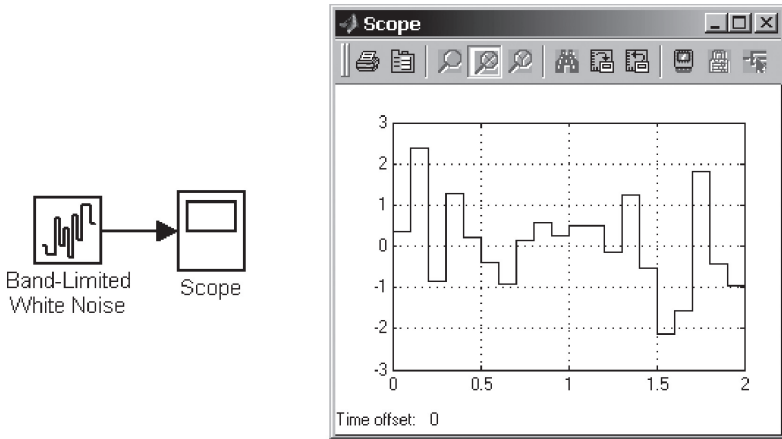
Параметрлері:

Noise Power – дауыс қуаттылығы.

Sample Time – модельдеу уақыты.

Seed – кездейсоқ сандар генераторын атау үшін керекті сан.

9.1.10-суретте ақ шуыл генераторының жұмысы көрсетілген.



9.1.10-сурет. Ақ шуыл генераторы

9.1.11 Clock уақыттық сигнал көзі

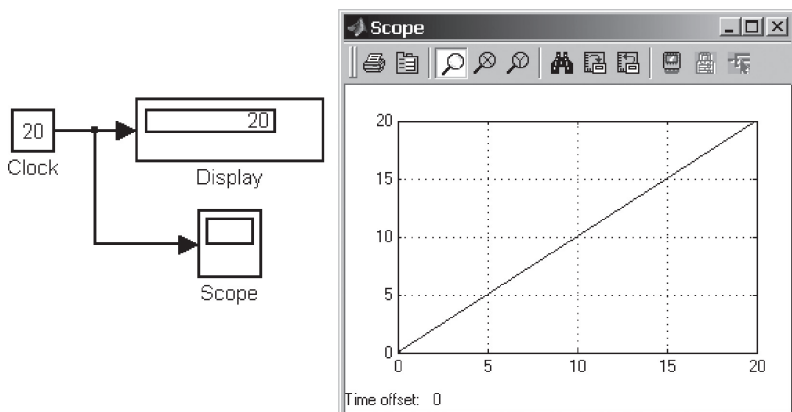
Қызметі: Модельдеудің ағымдағы уақытына тең, есептеу адымының әр шамасына тең сигналды түзеді.

Параметрлері:

Decimation – сигнал көзінің көрінісіндегі уақыт көрсеткішінің жаңару адымы (*Display time* параметрінің жалаушасы тұрған жағдайда). Параметрлері уақыт есептеу адымының саны ретінде беріледі. Мысалы, егер модельдің есептеу адымы *Simulation parameters* сұхбат терезесінде $0.01c$. тең етіп орнатылса, ал *Clock* блогының *Decimation* параметрі 1000 -ға тең деп берілсе, онда уақыт көрсеткішінің жаңаруы модельдеу уақыты бойынша $10c$. сайын болады.

Display time – сигнал көзінің блогындағы уақыт өлшемінің көрінісі.

9.1.11-суретте берілген сигнал көзінің мысалы көрсетілген.



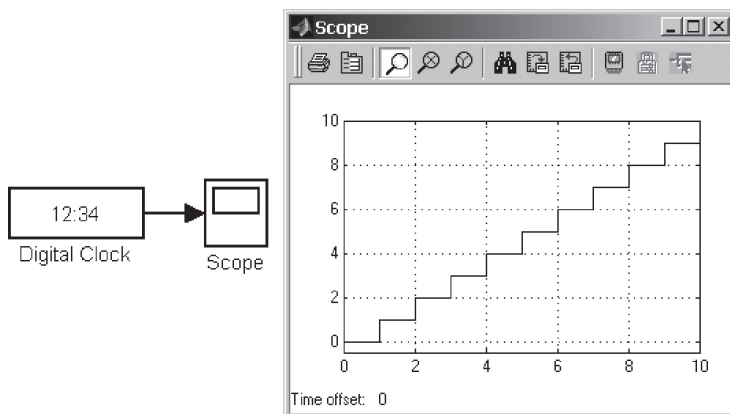
9.1.11-сурет. Уақыттық сигнал көзі

9.1.12 Digital Clock сандық уақыт көзі

Қызметі: Дискреттік уақыт сигналын түзеді.

Параметрлері: *Sample time* – модельдеу уақытының адымы
(с).

9.1.12-суретте *Digital Clock* көзінің жұмысы көрсетілген.



9.1.12-сурет. Сандық уақыт сигналының көзі

9.1.13 From File файлынан деректерді оқу блогы

Қызметі: Деректерді сыртқы файлдан алу.

Параметрлері:

File Name – берілген деректері бар файлдың атауы.

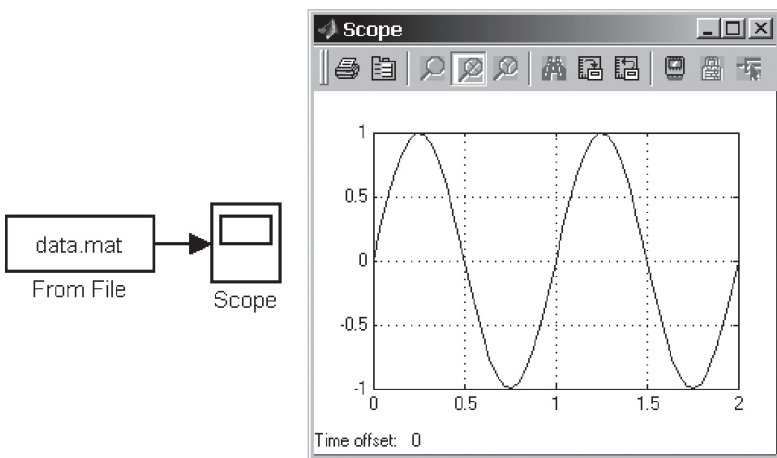
Sample time – блоктың шығыс сигналының өзгеріс адымы.

Файлдағы берілгендер матрица түрінде болуы қажет:

$$\begin{bmatrix} t_1 & t_2 & \dots & t_{\text{final}} \\ u1_1 & u1_2 & \dots & u1_{\text{final}} \\ \dots & \dots & \dots & \dots \\ un_1 & un_2 & \dots & un_{\text{final}} \end{bmatrix}$$

Матрица кем дегенде екі жолдан тұруы тиіс. Уақыттың мәні матрицаның бірінші жолында көрсетілген, ал қалған жолдарда уақыт моментіне сәйкес келетін сигналдардың мәні орналасқан болады. Уақыт мәні өсу ретімен жазылуы тиіс. Егер ағымдағы модельдің есептеу адымы берілгендер файлындағы уақыт есептеулермен сәйкес келмесе, онда Simulink берілгендерге сызықтық интерполяциялау жүргізеді.

Мәндері оқылатын деректер файлы (.mat-файл) мәтіндік болмайды. Файлдың құрылымы MATLAB-тың анықтамалық жүйесінде толық берілген. Simulink-ті қолданатындарға .mat-файлды *To File (Sinks* кітапханасындағы) блогының көмегімен орындау әлдеқайда жеңіл. 9.1.13-суретте берілген блоктың қолданылу мысалы көрсетілген. *data.mat* файлынан синусоидалық сигналдардың мәндері оқылып алынады.



9.1.13-сурет. *From File* блогы

9.1.14 From Workspace жұмыс кеңістігінен деректерді оқу блогы

Қызметі: MATLAB-тың жұмыс кеңістігінен деректерді алу.

Параметрлері:

Data – берілген деректері бар айнымалының (матрицаның немесе құрылымның) аты.

Sample time - блоктың шығыс сигналының өзгеру адымы.

Interpolate data — *Data* айнымалысындағы мәндер мен модельдеу уақытының мәндерімен сәйкес болмайтын деректерді интерполяциялау.

Form output after final data value by - *Data* айнымалысындағы уақыт мәнінің аяқталуынан шығыс сигналының түрі:

- *Extrapolate* – сигналдардың сызықтық экстраполяциясы

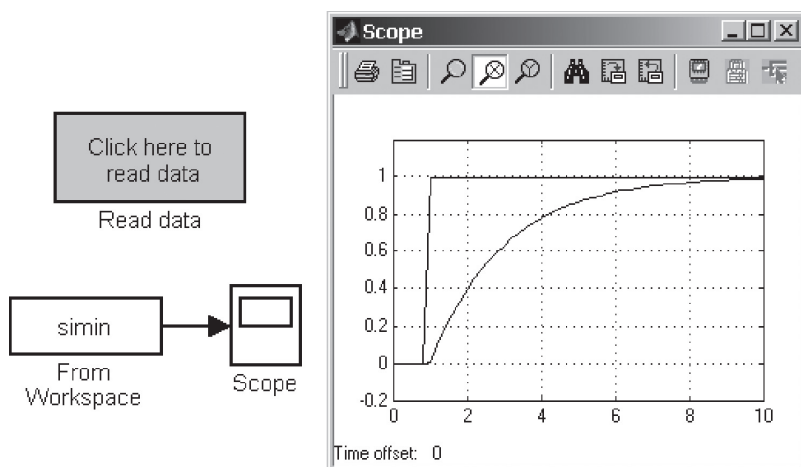
- *SettingToZero* – сигналдардың нөлдік мәні

- *HoldingFinalValue* – шығыс сигналының мәндері соңғы алынған мәндерге тең.

- *CyclicRepetition* – сигналдар мәнінің циклдік қайталануы.

Егер *Data* айнымалысы *Structure without time* қалыбына ие болса, онда берілген нұсқа қолданылады.

9.1.14-суретте блоктың қолданылу мысалы көрсетілген. MATLAB-тың жұмыс аймағындағы *Simin* айнымалысына берілгендер *Read data* блогының көмегімен жазылады.



9.1.14-сурет. *From Workspace* блогы

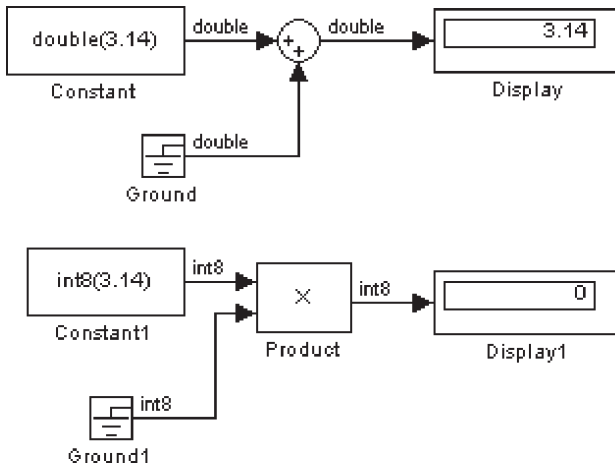
9.1.15 Ground нөлдік деңгейдегі сигналдар блогы

Қызметі: Нөлдік деңгейдегі сигналдарды түзеді.

Параметрлері: Жок.

Егер модельде қандай да бір блок жалғанбаған болса, онда модельдеу барысында MATLAB терезесінде ескерту хаттамасы шығады. Оны түзету үшін блоктың қосылмаған кірісіне *Ground* блогынан сигнал беруге болады.

9.1.15-суретте блокты қолданудың мысалдары келтірілген. Бірінші жағдайда, *Ground* блогынан сигнал жиынтықтаушының кірісінің біреуіне түсіп, екінші жағдайда, көбейту блогының кірісінің біреуіне барып түседі. *Ground* блогынан шыққан сигнал нөлдік мән екенін *Display* блогының көрсеткіштері дәлелдейді. Сонымен қатар, суретте көрсетілгендей, шығыс сигналының типі басқа блоктардың кірістеріне (*Sum* және *Product* блоктарының кірістері) берілген сигналдардың түріне байланысты автоматты түрде орнатылады.



9.1.15-сурет. *Ground* блогының қолданылуы

9.1.16 Repeating Sequence периодтық сигналдар блогы

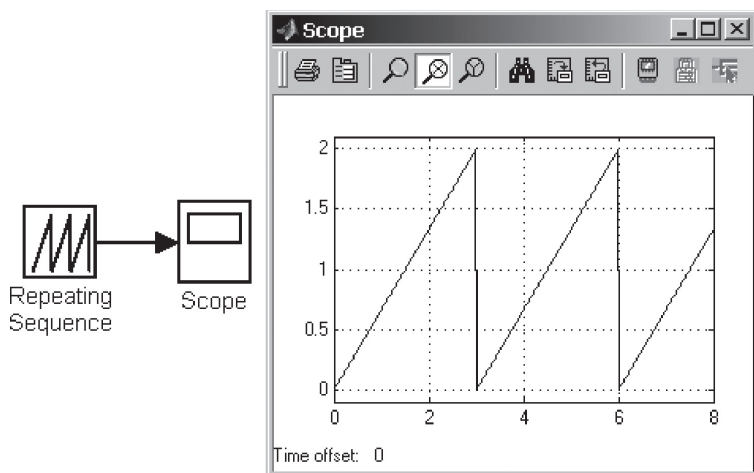
Қызметі: Периодтық сигналдарды түзеді.

Параметрлері:

Time values – модельдеу уақытының мәнінің векторы.

Output values – *Time values* векторы арқылы берілген уақыт моменті үшін сигнал мәнінің векторы.

Бұл блок *Time values* векторымен берілген мәндермен сәйкес келмейтін уақыт моменті үшін шығыс сигналының сызықтық интерполяциясын орындайды. 9.1.16-суретте ара тәріздес сигналды түзу үшін блокты қолданудың мысалы көрсетілген. Модельдеу уақытының мәні $[0\ 3]$ векторымен берілсе, шығыс сигналының мәні $[0\ 2]$ векторы арқылы берілген.



9.1.16-сурет. *Repeating Sequence* блогының қолданылуы

9.1.17 Inport кіріс портының блогы

Қызметі: Ішкі жүйелер немесе иерархияның жоғарғы деңгейінің моделі үшін кіріс портын жасайды.

Параметрлері:

Port number – портың нөмірі.

Port dimensions – кіріс сигналының мөлшері. Егер ол шама – 1-ге тең болса, онда кіріс сигналының мөлшері автоматты түрде анықталады.

Sample time – модельдеу уақытының адымы.

Data type – кіріс сигналының берілгендерінің типтері: *auto*, *double*, *single*, *int8*, *uint8*, *int16*, *uint16*, *int32*, *uint32* немесе *boolean*.

Signal type – кіріс сигналының типі:

auto – типті автоматты түрде анықтау.

real – нақты сигнал.

complex – комплекстік сигнал.

Interpolate data (жалауша) – кіріс сигналын интерполяциялау.

Егер MATLAB-тың жұмыс аймағынан алынған кіріс сигналының уақыттық есептеулері модельдеу уақытымен дәл келмесе, онда блок кіріс сигналын интерполяциялауды орындайды. Ішкі жүйеде *Inport* блогын қолданған жағдайда, бұл параметр қол жетімсіз болып қалады.

9.1.17.1 Ішкі жүйелерде *Inport* блогын қолдану

Ішкі жүйенің *Inport* блоктары оның кірістері болып табылады. *Inport* блогы арқылы ішкі жүйеге ішкі порт арқылы берілген сигнал ішкі жүйенің ішіне беріледі. Кіретін порттың аты ішкі жүйенің суретінде порттың таңбасы ретінде көрсетіледі.

Ішкі жүйелерді құрастырғанда және ішкі жүйеге *Inport* блогын қосқанда *Simulink* келесі ережелерді қолданады:

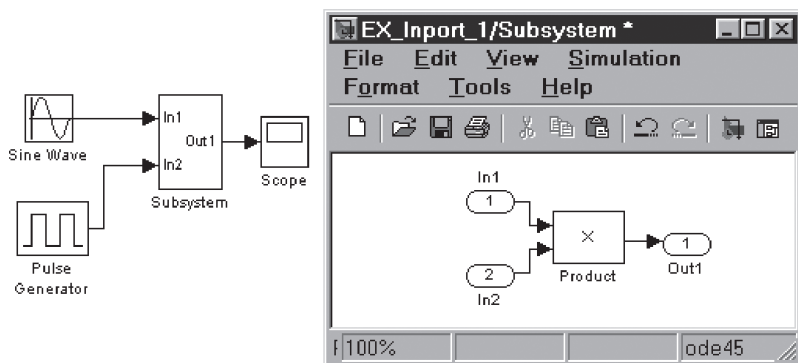
Edit/Create subsystem командасы арқылы ішкі жүйені құрастырғанда кіретін порттар құрылып, оларға *I*-ден бастап, автоматты түрде нөмірлеу қойылады.

Егер ішкі жүйеге жаңадан тағы бір *Inport* блогы қосылса, оған келесі реттік нөмір беріледі.

Егер қандай да бір *Inport* блогы жойылса, онда қалған порттардың реті үздіксіз болатындай болып аталады.

Егер порт нөмірлерінің ретінде бос орындар болса, онда *Simulink* модельдеу кезінде қате деген хаттама жібереді де, есептеуді тоқтатады. Мұндай жағдайда міндетті түрде порт нөмірлерінің реті бұзылмайтындай етіп, порттардың атын қолмен өзгерту керек.

9.1.17-суретте ішкі жүйені пайдаланатын модель мен сол ішкі жүйесінің сұлбасы көрсетілген.



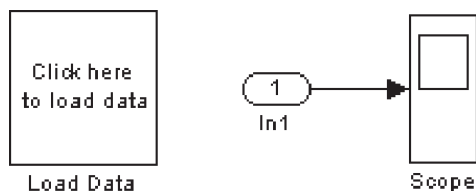
9.1.17-сурет. *Inport* блогының ішкі жүйеде қолданылуы

9.1.17.2 Жоғары дәрежедегі модельде *Inport* блогын қолдану

Жоғары дәрежелі жүйедегі кіріс порты сигналды MATLAB-тың жұмыс аймағын модельге жіберу үшін қолданады.

MATLAB-тың жұмыс кеңістігінен сигналды жіберу үшін модельде тек кіріс портын құру ғана емес, ол үшін *Simulation parameters...* сұхбаттық терезесіндегі *Workspace I/O* қосымшасында енгізу параметрлерін (*Input* параметрі үшін жалаушасы қойылып, кіріс берілгендерін қамтитын айнымалы аты берілуі керек) орнату керек. Енгізуге арналған деректердің *Array* (жиым), *Structure* (құрылым) немесе *Structure with time* («уақыт» өрісімен құрылым) атты типтері осы қосымшада беріледі.

9.1.18-суретте MATLAB-тың жұмыс кеңістігінен кіріс сигналын оқитын модель көрсетілген. *Load Data* ішкі жүйесі файлдан MATLAB-тың жұмыс кеңістігіне деректерді енгізуді қамтамасыз етеді.



9.1.18-сурет. *Input* блогының көмегімен MATLAB жұмыс кеңістігінен кіріс сигналын оқитын модель

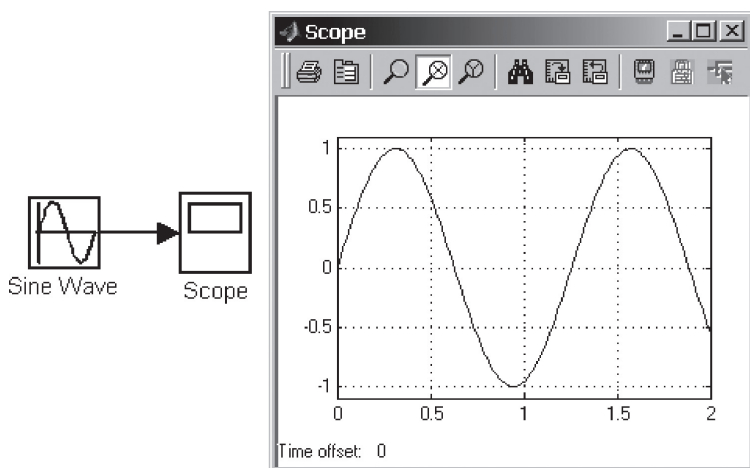
9.2 Sinks – сигнал қабылдағыштар

9.2.1 Scope осциллографы

Қызметі:

Уақыт функциясында зерттелетін сигналдардың графиктерін тұрғызу. Ол модельдеу үдерісіндегі сигналдардың өзгеруін бақылауға мүмкіндік береді.

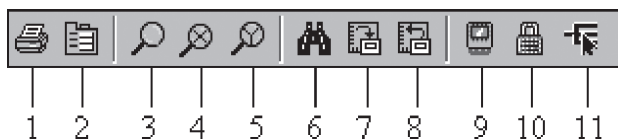
9.2.1-суретінде блоктың суреті мен графиктерді бақылау терезесі көрсетілген.



9.2.1-сурет. Scope осциллографы

Сигналды көру терезесін ашу үшін блоктың суретінде тышқанның сол жақ пернесін екі рет басу арқылы аламыз. Оны есептеудің кез келген жерінде орындауға болады (есептеудің басында, одан кейін және есептеу кезінде). Егер блоктың кірісіне векторлық сигнал түссе, онда әрбір вектор элементі үшін салынған қысық әр түрлі түспен сызылады.

Осциллограф терезесін күйге келтіру құрал-саймандар жақтауы арқылы іске асырылады. (9.2.2-сурет).



9.2.2-сурет. Scope блогының құрал-саймандар жақтауы

Құрал-саймандар жақтауы 11 батырмадан тұрады:

Print – осциллограф терезесінің ішіндегісін баспаға шығару.

Parameters – параметрлерді күйге келтіру терезесіне енгізуге рұқсат.

Zoom – екі ось бойынша масштабтың өсуі.

Zoom X-axis – горизонталь ось бойынша масштабтың өсуі.

Zoom Y-axis – вертикаль ось бойынша масштабтың өсуі.

Autoscale – екі жақ осьтері бойынша автоматты түрде масштаб құру.

Save current axes settings – терезенің ағымдағы күйге келтірулерін сақтау.

Restore saved axes settings – алдында сақталған терезенің күйге келтірулерін қою.

Floating scope – осциллографтың «еркіндік» режиміне аударылуы.

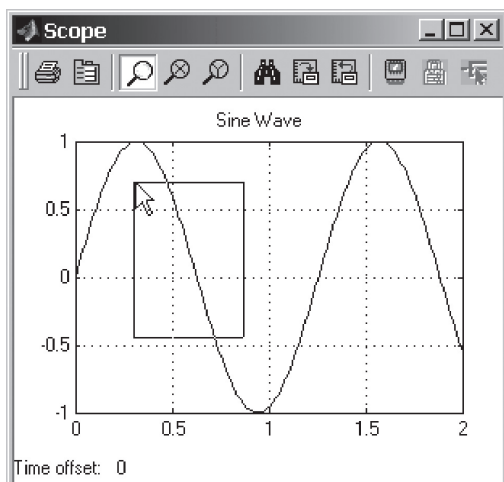
Lock/Unlock axes selection – ағымдағы координаталық жүйе терезесі мен көрсетілетін сигнал арасындағы қатынасты біріктіру/бөлу. Құрал *Floating scope* режимі қосылғанда ғана іске жарамды болады.

Signal selection – көрсету үшін сигналдарды таңдау. Құрал *Floating scope* режимі қосылғанда ғана іске жарамды.

Көрсетілетін графиктердің масштабтарының өзгеруін бірнеше әдістермен орындауға болады:

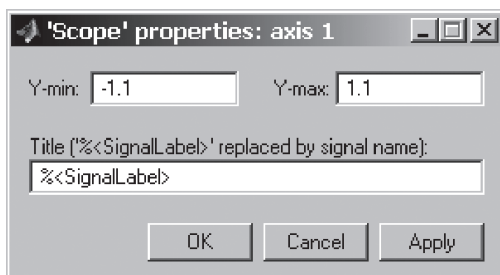
1. Керекті батырманы (🔍, 🔍 немесе 🔍) басу және тышқанның сол жақ пернесі арқылы графиктің керекті жеріне бір рет шерту керек. Масштаб 2,5 есе өседі.

2. Керекті батырманы (🔍, 🔍 немесе 🔍) басу және тышқанның сол жақ пернесін басып, графиктегі суретті динамикалық рамканың немесе кесілген бөлігінің көмегімен үлкейтіп көру үшін керекті аймақты көрсету керек. 9.2.3-суреттен осы үдерісті көруге болады.




9.2.3-сурет. Масштабтың өсу графигі

3. Тышқанның оң жақ пернесімен график терезесіне шерту және контекстік мәзірден *Axis properties...* командасын таңдау керек. *Y-min* және *Y-max* параметрлерінің көмегімен вертикаль осьтерінің шектік мәндерін көрсете алатын графиктің қасиеттер терезесі ашылады. Енгізу жолында *%<SignalLabel>* өрнегін ауыстыру арқылы осы терезеде графиктің (*Title*) тақырыбын көрсетуге болады. Қасиеттер терезесі 9.2.4-суретте көрсетілген.



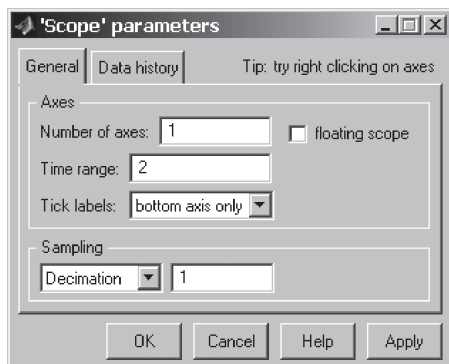
9.2.4-сурет. График қасиеттерінің терезесі

Қызметі: Блок параметрлері құрал-саймандар жақтауындағы  (*Parameters*) құралының көмегімен ашылатын *'Scope'* *parameters* терезесінде қойылады. Параметр терезесінің екі қосымшасы бар:

General – жалпы параметрлер.

Data history – MATLAB-тың жұмыс аймағындағы сигналдарды сақтау параметрлері.

Жалпы параметрлердің қосымшасы 9.2.5-суретте көрсетілген.



9.2.5-сурет. *General* жалпы параметрлердің қосымшасы

General қосымшасында келесі параметрлер беріледі:

1. *Number of axes* — осциллограф кірісінің (координата жүйелері) сандары. Осы параметрдің өзгеруінен блоктың суретінде қосымша кіріс порттары пайда болады.

2. *Time range* — графиктер үшін көрінетін уақыт аралығының өлшемі. Егер модельді есептеу уақыты *Time range* арқылы берілген параметрінен асып кетсе, онда графиктің нәтижесі бөліктермен шығатын болады және графиктің әрбір бөлігінің көрсетілу аралығы берілген *Time range* мәніне тең болады.

3. *Tick labels* — осьтерді және осьтердің белгілеулерін көрсету/жасыру. Ол келесідей үш мәнді қабылдай алады (тізімнен таңдалады):

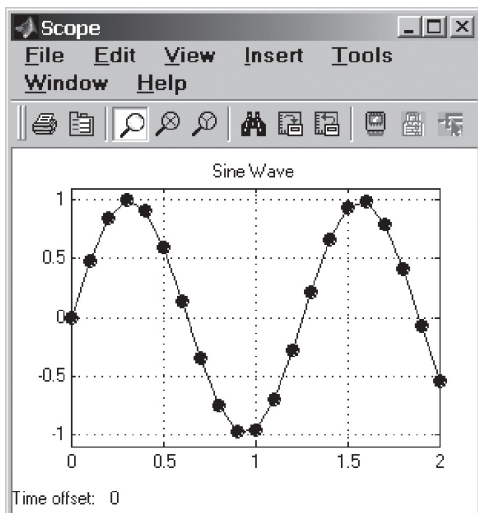
all – барлық осьтер үшін белгілеулері бар,

none – осьтердің және олардың белгілеулерінің жоқ болуы,

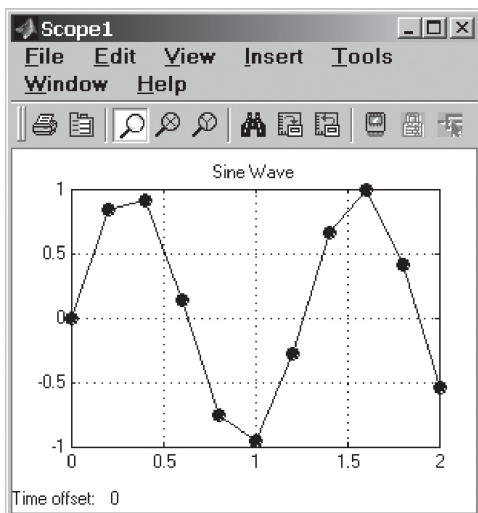
bottom axis only – тек төменгі график үшін горизонталь осьтердің белгілеулері бар.

4. *Sampling* — графиктерді шығару терезесіндегі параметрлерді қою. Есептеліну нүктелерін экранға шығару режимін береді. *Decimation* таңдау кезінде қысқарып шығарылу есептеліну нүктелерін көрсететін санмен құрылады. 9.2.6 және 9.2.7-суреттерде тұрақтандырылған *0.1c*. адыммен есептелген

синуоидалық сигналдың графиктері көрсетілген. 9.2.6-суретте *Scope* блогының терезесінде әрбір есептеліну нүктесі шығарылады (*Decimation* параметрі 1-ге тең). 9.2.7-суретте әрбір екінші мәннің нәтижесі көрсетілген (*Decimation* параметрі 2-ге тең). Графикте маркерлермен есептеліну нүктелері белгіленген.

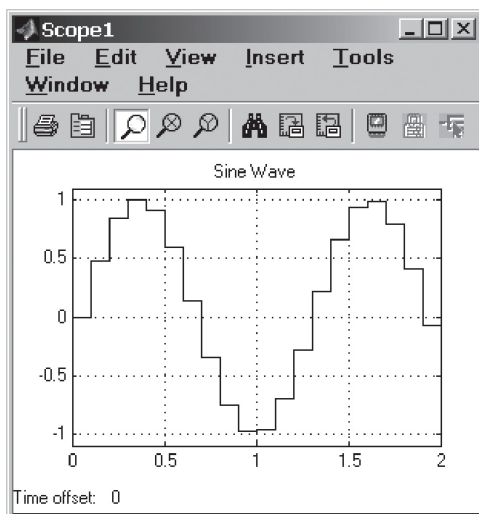


9.2.6-сурет. Синуоидалық сигналдың көрсетілуі (*Decimation=1*)



9.2.7-сурет. Синуоидалық сигналдың көрсетілуі (*Decimation=2*)

Егер есептеліну нүктелерінің көрсету режимі *Sample time* сияқты берілсе, онда оның сандық мәні сигналдың көрсетілуінде аралықтың квантталуымен анықталады. 9.2.8-суретте *Sample time* параметрінің мәні 0.1-ге тең болғандағы синусоидалық сигналдың графигі көрсетілген.

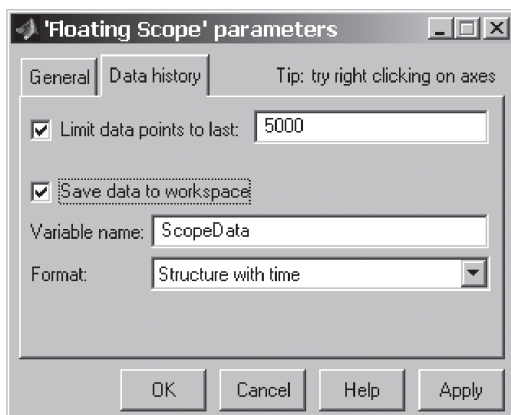


9.2.8-сурет. Синусоидалық сигналдың көрсетілуі ($Sample\ time=0.1$)

5. *floating scope* – осциллографты «еркіндік» режиміне ауысуы (жалауша белгісінің қойылған кезінде).

Data history (9.2.9-сурет) қосымшасында келесі параметрлер беріледі:

1. *Limit data points to last* – графиктің есептеліну нүктелерінің ең көп көрсетілетін саны. Осы саннан асқан кезде графиктің бастапқы бөлігі кесіледі. Егер *Limit data points to last* белгісінің параметрі қойылмаса, онда Simulink барлық есептеліну нүктелері үшін осы параметрдің мәнін автоматты түрде үлкейтеді.



9.2.9-сурет. *Data history* қосымшасы

2. *Save data to workspace* – MATLAB-тың жұмыс аймағында сақталатын сигналдардың мәндері.

3. *Variable name* – MATLAB-тың жұмыс аймағында сигналдарды сақтау үшін арналған айнымалының атауы.

4. *Format* – MATLAB-тың жұмыс аймағында сақтау үшін берілетіндердің қалыбы.

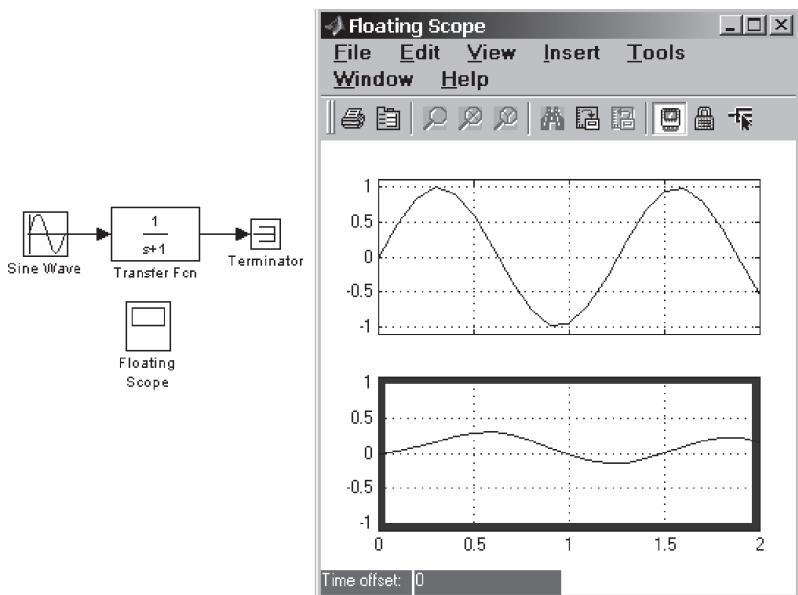
Array – жиым,

Structure – құрылым,


Structure with time – «уақыт» қосымша өрісі бар құрылым.

9.2.2 Floating Scope осциллографы

Floating Scope осциллографы, негізінде «еркіндік» режиміне ауыстырылған кәдімгі *Scope* осциллографы. 9.2.10-суретте *Floating Scope* осциллографымен құрылған модельдің мысалы көрсетілген.



9.2.10-сурет. *Floating Scope* осциллографымен құрылған модель

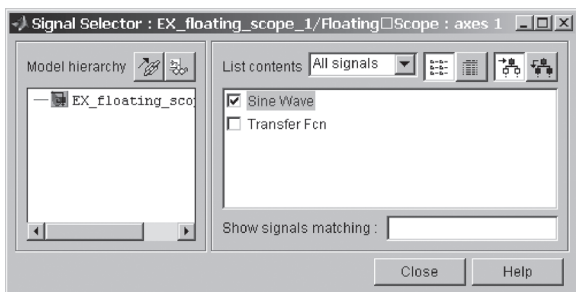
Бұл режимде осциллограф блогының кірісі болмайды, ал көрсету керек болатын сигналды таңдау құрал-саймандар жақтауындағы  (*Signal selection*) құралының көмегімен орындалады. Сигналды таңдап алу үшін келесі әрекеттерді орындалады:

1. График шығарылатын координат жүйесін белгілеп алу. Бұл әрекет керекті жүйеде тышқан пернесінің сол жақ батырмасын басу арқылы орындалады. Таңдап алынған координат жүйесі периметр бойынша көк түспен белгіленетін болады.

2.  құралының көмегімен *Signal Selector* сұхбат терезесін ашу (9.2.11-сурет).

3. Шығыс сигналдары зерттелетін блоктардың аттарын жалауша белгісімен белгілеу.

Есептеу аяқталған соң *Floating Scope* блогының терезесінде таңдалған сигналдар нәтижелері көрсетілетін болады.



9.2.11-сурет. *Signal Selector* сұхбат терезесі

9.2.3 XY Graph сызба құрастырушысы

Қызметі: Бір сигналдың графигін басқа сигналдың функциясы ретінде құрады (графиктің түрі $Y(X)$).

Параметрлері:

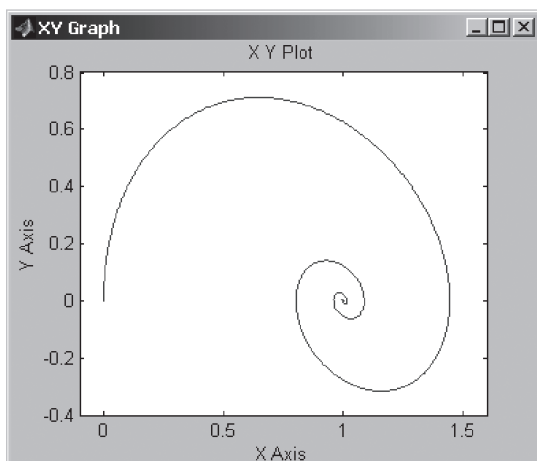
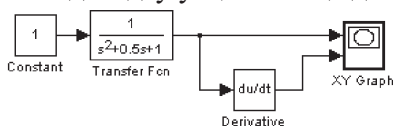
$x-min$ – X осі бойынша сигналдың минималдық мәні.

$x-max$ – X осі бойынша сигналдың максималдық мәні.

$y-min$ – Y осі бойынша сигналдың минималдық мәні.

$y-max$ – Y осі бойынша сигналдың максималдық мәні.

Sample time – модельдеу уақытының адымы.

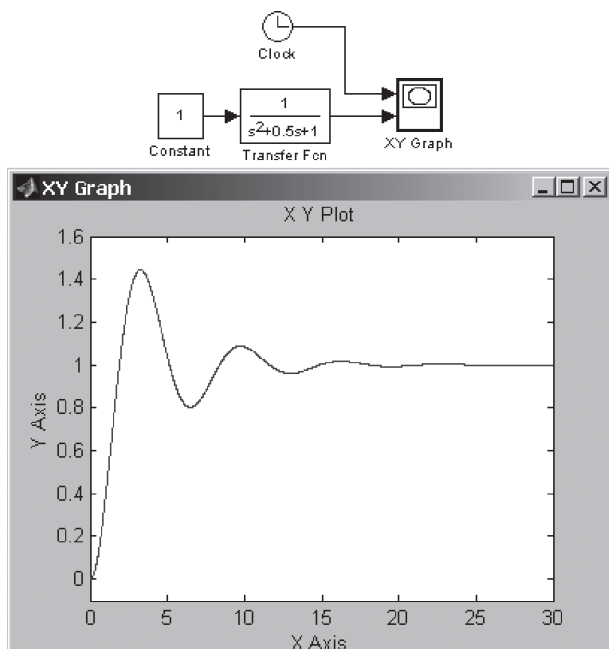


9.2.12-сурет. *XY Graph* сызба құрастырушысының қолданылуы

Блоктың екі кірісі бар. Жоғарғысы - аргументі (X) болатын сигналды беру үшін, ол, ал төменгісі - (Y) функциясының мәнін беру үшін арналған.

9.2.12-суретте тербелмелі звеноның фазалық траекториясын құруда сызба құрастырушыны қолдану мысалы көрсетілген.

Уақытқа тәуелділіктерді құруда да сызба құрастырушыны қолдануға болады. Ол үшін оның бірінші кірісіне *Clock* блогының шығысынан уақыттық сигнал беріледі. Осындай сызба құрастырушыны қолдану 9.2.13-суретте көрсетілген.



9.2.13-сурет. Уақыт тәуелділіктерін көрсету үшін *XY Graph* блогын қолдану

9.2.4 Display сандық дисплейі

Қызметі: Сигнал мәнін сан түрінде көрсетеді.

Параметрлері:

Format – деректердің көріну қалыбы. *Format* параметрі келесідей мәндерді қабылдай алады:

short – 5 мәндік ондық сан.

long – 15 мәндік ондық сан.

short_e –5 мәндік ондық сан және 3 ондық дәрежелік символ.

long_e – 15 мәндік ондық сан және 3 ондық дәрежелік символ.

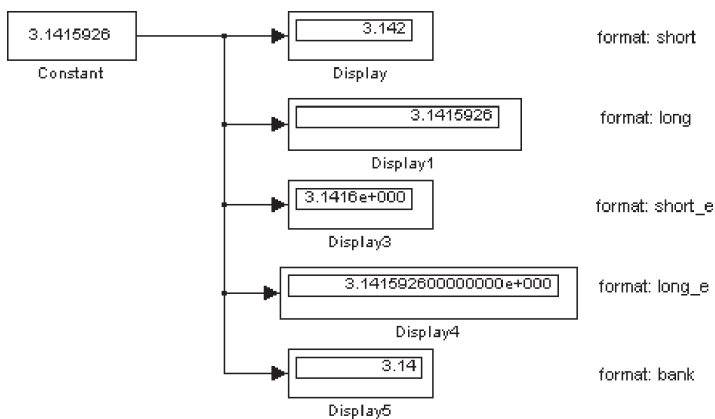
bank – «ақшалы» қалып. Бекітілген нүктенің және екі ондық санның қалыбы.

Decimation – кіріс сигналының еселеніп көрсетілуі. *Decimation=1* болғанда, кіріс сигналының әрбір мәні көрсетіледі, *Decimation=2* болғанда, әрбір екінші мән көрсетіледі, ал *Decimation=3* болғанда – әрбір үшінші мән және т.т.

Sample time – модельдеу уақытының адымы. Деректердің көрсетілуінің дискреттілігін анықтайды.

Floating display (жалауша) – блоктың «еркіндік» режиміне ауысуы. Берілген режимде блоктың кіріс порты жоқ, ал көрсетілу сигналын таңдау тышқан пернесінің сол жақ батырмасын лайықты сигнал сызығында басу арқылы атқарылады. Осы режимде *Signal storage reuse* есептеу параметрі үшін *off* мәні қойылу керек (*Simulation parameters...* сұхбаттық терезесіндегі *Advanced* қосымшасы).

2.9.14-суретте *Format* параметрлерінің әртүрлі нұсқалары үшін *Display* блогының қолданылуы көрсетілген.



2.9.14-сурет. *Format* параметрлерінің әртүрлі нұсқалары үшін *Display* блогының қолданылуы

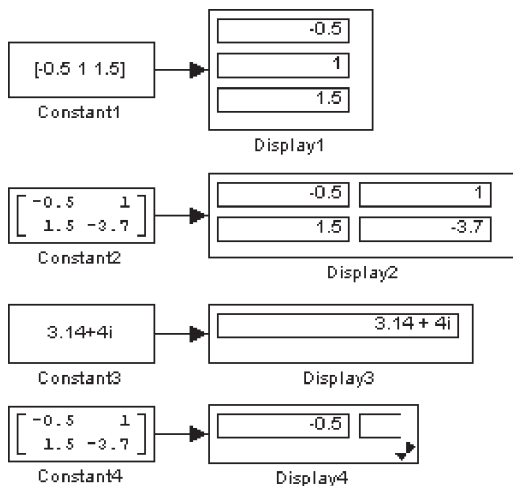
Display блогы тек қана скалярлық қана емес, сонымен қатар векторлық, матрицалық және комплекстік сигналдардың да көрінуі үшін қолданыла береді. (2.9.15-сурет). Егер барлық көрінетін мәндер блок терезесіне сыймаса, онда блогтың оң жақ төменгі бұрышында блогтың керекті өлшемін үлкейту керектігін көрсететін символ шығады. (*Display4* блогы).

9.2.5 Stop Simulation модельдеуді тоқтату блогы

Қызметі: Егер блогтың кіріс сигналы нөлге тең болмаса, есептеудің аяқталуын қамтамасыз етеді.

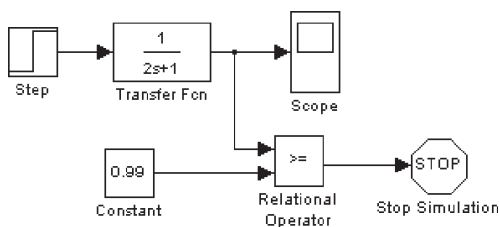
Параметрлері: Жок.

Блок кірісіне нөлдік емес сигнал бергенде, Simulink ағымдағы соңғы есептеу жұмысын атқарып, содан кейін барып модельдеуді тоқтады. Егер блок кірісіне векторлық сигнал берілсе, онда есептеуді тоқтату үшін вектордың бір элементінің нөлдік емес болуы жеткілікті.



9.2.15-сурет. Векторлық, матрицалық және комплекстік сигналдардың көрінуі үшін *Display* блогының қолданылуы

2.9.16-суретте *Stop Simulation* блогының қолданылуы мысалы көрсетілген. Егер *Transfer Function* блогының шығыс сигналы 0.99-дан үлкен немесе тең болса, онда осы мысалдағы есептеу тоқтатылады.



9.2.16-сурет. *Stop Simulation* блогының қолданылуы

9.2.6 Деректерді To File файлында сақтау блогы

Қызметі: Блок өзінің кірісінен берілген деректерді файлға жазады.

Параметрлері:

Filename – жазу үшін арналған файлдың аты. Үнсідікпен файл *untitled.mat* атын қабылдайды. Егер файлдың жолы толық көрсетілмесе, онда файл соңғы жұмыс папкасында сақталады.

Variable name – жазылатын деректерді қамтитын айнымалының аты.

Decimation – кіріс сигналының файлға еселеніп жазылуы. *Decimation=1* болғанда, кіріс сигналының әрбірі жазылады, *Decimation=2* болғанда, әрбір екінші мән жазылады, ал *Decimation=3* болғанда – әрбір үшінші мән және т.т.

Sample time – модельдеу уақытының адымы. Деректердің жазылуының дискреттілігін анықтайды.

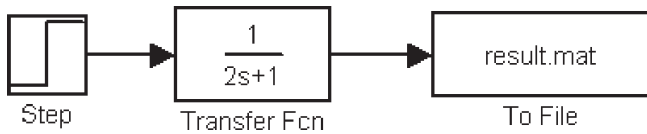
Берілген деректер файлда матрица түрінде сақталады:

$$\begin{bmatrix} t_1 & t_2 & \dots & t_{\text{final}} \\ u1_1 & u1_2 & \dots & u1_{\text{final}} \\ \dots & \dots & \dots & \dots \\ un_1 & un_2 & \dots & un_{\text{final}} \end{bmatrix}$$

Уақыт мәні матрицаның бірінші жолында жазылады, ал қалған жолдарда сол уақыт моменттеріне сәйкес келетін сигналдардың мәндері жазылады.

Берілген деректер жазылатын деректік файл (*mat*-файл) мәтіндік болмайды. Файлдың құрылымы MATLAB-тың әдістемелік жүйесінде толығымен жазылған. Simulink-тің қолданушыларына *mat*-файлдағы деректерді *From File* блогының көмегімен оқу тиімдірек болады (*Sources* кітапханасы).

9.2.17-суреттегі мысалда берілген блоктың қолданылуы көрсетілген. Есептеу нәтижесі *result.mat* файлында сақталады.



9.2.17-сурет. *To File* блогының қолданылуы

9.2.7 Деректерді *To Workspace* жұмыс аймағында сақтау блогы

Қызметі: Блок өзінің кірісінен берілген деректерді MATLAB-тың жұмыс аймағына жазады.

Параметрлері:

Variable name – жазылатын деректерді қамтитын айнымалының аты

Limit data points to last – уақыт бойынша есептеліну нүктелерінің ең көп сақталу мөлшері. Егер *Limit data points to last* параметрінің мәні *inf* арқылы берілсе, онда жұмыс аймағында барлық деректер сақталатын болады.

Decimation – кіріс сигналының файлға еселеніп жазылуы.

Sample time – модельдеу уақытының адымы. Берілген деректердің жазылу дискреттілігін анықтайды.

Save format – деректерді сақтау қалыбы. Ол келесідей мәндерді қабылдайды:

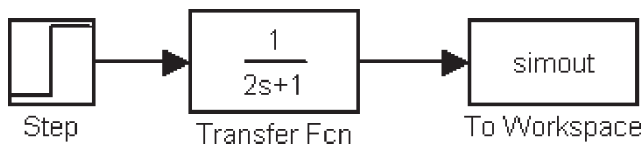
Matrix – матрица. Берілгендер жиым сияқты сақталады. Мұндағы жол саны уақыт бойынша есептелетін нүктелердің санымен, ал баған сандары блок кірісіне берілген вектордың өлшемімен анықталады. Егер кіріске скалярлық сигнал берілсе, онда матрицада тек бір баған ғана болады.

Structure – құрылым. Берілген деректер үш өрісі бар құрылым түрінде сақталады, яғни, *time* – уақыт, *signals* – сигналдардың сақталу мәні, *blockName* – модельдің және *To Workspace* блогының атауы. *Time* өрісі берілген қалып үшін толтырылмаған болып қалады.

Structure with Time – қосымша жолмен берілген құрылым (уақыт). Бұл қалып үшін, алдыңғыға қарағанда, *time* өрісі уақыт мәндерімен толтырылады.

9.2.18-суретте берілген блоктың қолданылуы көрсетілген. Есептеу нәтижесі *simout* айнымалысында сақталады.

MATLAB-тың жұмыс аймағында сақталған деректердің оқылуы үшін *From Workspace* блогын қолдануға болады. (*Sources* кітапханасы).



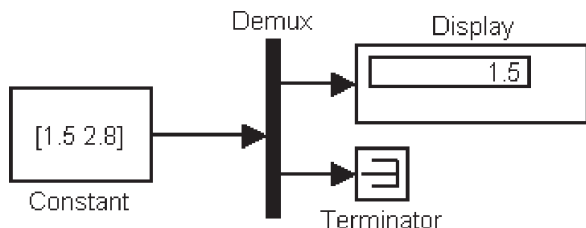
9.2.18-сурет. *To Workspace* блогын қолдану

9.2.8 Terminator соңғы қабылдағышы

Қызметі: Блок басқа блоктың қолданылмайтын шығыс сигналын беруде қолданылады.

Параметрлері: Жоқ.

Егер блоктың шығысы басқа блоктың кірісіне қосылмаған болса, онда Simulink MATLAB-тың командалық терезесіне ескерту хаттамасын жібереді. Оны болдыртпау үшін *Terminator* блогын қолдану қажет. 9.2.19-суретте соңғы қабылдағыштың қолданылуы көрсетілген. *Demux* блогының көмегімен алынған матрицаның екінші элементі мүлдем қолданылмайды, сондықтан ол *Terminator* блогының кірісіне беріледі.



9.2.19-сурет. *Terminator* блогының қолданылуы

9.2.9 Outport шығыс портының блогы

Қызметі: Иерархияның жоғары сатысындағы модель немесе ішкі жүйе үшін шығыс портын құрайды.

Параметрлері:

Port number – порттың нөмірі.

Output when disabled – ішкі жүйе өшірілген кездегі ішкі жүйедегі шығыс сигналының түрі. Ол ішкі жүйені басқару үшін қолданылады. Келесідей мәндерді қабылдауы мүмкін (тізімнен таңдалады):

held – ішкі жүйенің шығыс сигналы соңғы есептелген мәнге тең.

reset – ішкі жүйенің шығыс сигналы *Initial output* параметрімен берілген мәнге тең.

Initial output – ішкі жүйе өшірілген және ішкі жүйенің жұмысы басталғанға дейінгі кездегі шығыс сигналының мәні. Ол ішкі жүйені басқару үшін қолданылады.

9.2.9.1 Ішкі жүйелерде *Outport* блогын қолдану

Ішкі жүйеде *Outport* блогы оның шығысы болады. Ішкі жүйе ішінде *Outport* блогына берілетін сигнал жоғары дәрежелі модельге (ішкі жүйеге) жіберіледі. Шығыс портының атауы ішкі жүйе кескінінде порт белгісі ретінде көрсетіледі.

Ішкі жүйені құрғанда және *Outport* блогын қосқанда, Simulink ішкі жүйеде келесідей ережелерді қолданады:

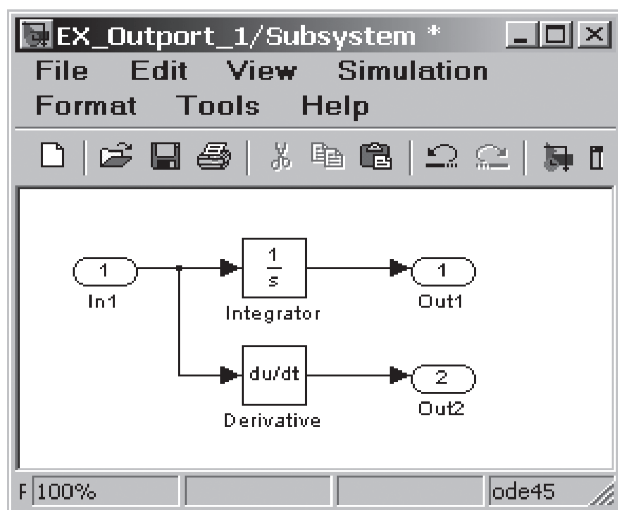
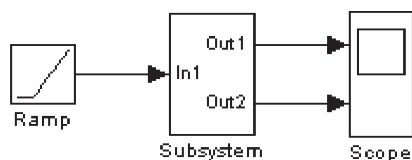
Edit/Create subsystem командасының көмегімен ішкі жүйе құрғанда шығыс порттары құрылады және автоматты түрде 1-ден бастап нөмірленеді.

Егер ішкі жүйеге *Outport* жаңа блогы қосылса, онда оған келесі реттік нөмір қойылады.

Егер қандай да бір *Outport* блогы жойылса, онда қалған порттардың реттілігі өзгермейтіндей болып қойылады.

Егер порт нөмірлерінің ретінде бос орындар болса, онда Simulink модельдеуді атқару кезінде қате деген хаттама жібереді де, есептеуді тоқтатады. Осындай жағдайда міндетті түрде порт нөмірлерінің реті бұзылмайтындай етіп, порттардың атын қолмен өзгерту керек.

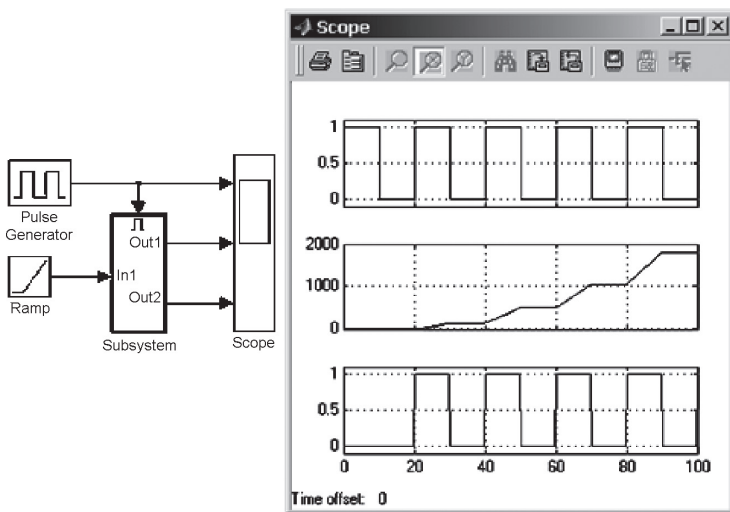
9.2.19-суретте пайдаланылатын ішкі жүйесі бар модель және осы ішкі жүйенің сұлбасы көрсетілген.



9.2.19-сурет. *Outport* блогының ішкі жүйеде қолданылуы

Егер ішкі жүйе басқарылатын болса, онда оның шығыс порттарына ішкі жүйе құлыптанған кездегі уақыт аралығының сигналдары арқылы беруге болады.

9.2.20-суретте басқарылатын ішкі жүйені пайдаланатын модель көрсетілген. Ішкі жүйенің бірінші шығыс портының *Output when disabled* параметрі *held* түрінде берілген, ал екіншісі - *reset* түрінде, ал бас мәнінің шамасы нөлге тең болып берілген. Сигналдар графигінен көруге болатыны, ол ішкі жүйенің құлыптанған кезінде бірінші шығыс портындағы сигнал өзгеріссіз қалады да, екіншісінің сигналы берілген бастапқы мәнге (нөлге) тең болады.



9.2.20-сурет. Шығыс порттарының әртүрлі күйге келтірулері арқылы ішкі жүйені басқару

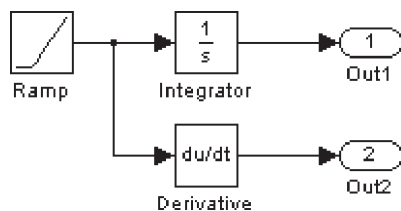
9.2.9.2 Outport блогын жоғары деңгейдегі модельде қолдану

Жоғары дәрежелі жүйедегі шығыс порты екі жағдайда қолданылады:

- MATLAB-тың жұмыс аймағынан сигналды жіберу үшін,
- модель шығысы мен анализ функциясының байланысын қамтамасыз ету үшін.

MATLAB-тың жұмыс кеңістігіне сигналды жіберу үшін модельде тек шығыс портын ғана емес, *Simulation parameters...* сұхбат терезесінде *Workspace I/O* қосымшасының параметрлерін (*Output* параметрі үшін жалаушасы қойылып, деректерді сақтау үшін айнымалы атауы берілуі керек) де қою керек. Сақталатын деректердің типтері - *Array* (жиым), *Structure* (құрылым) немесе *Structure with time* («уақыт» өрісі бар құрылым) осы қосымшада беріледі.

9.2.21-суретте *Outport* блогы көмегімен MATLAB-тың жұмыс кеңістігіне сигнал жіберетін модель көрсетілген.



9.2.21-суретте *Outport* блогы көмегімен MATLAB-тың жұмыс кеңістігіне сигнал жіберетін модель

Outport блогы анализ функциясының моделімен байланыс үшін қолданылуы мүмкін, мысалы: *linmod* және *trim*.

9.3 Continuous – аналогтық блоктар

9.3.1 Derivative туынды есептеу блогы

Қызметі: Кіріс сигналының сандық дифференциалануын атқарады.

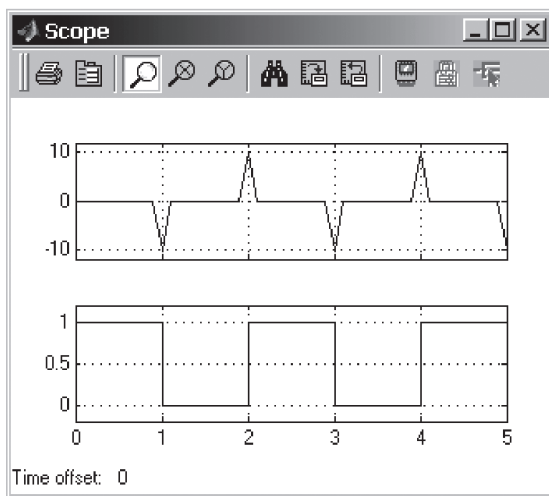
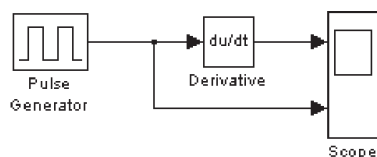
Параметрлері: Жоқ.

Туындыны есептеу үшін Эйлердің жуықтау формуласы қолданылады:

$$\frac{du}{dt} = \frac{\Delta u}{\Delta t},$$

мұндағы Δu - Δt уақытындағы кіріс сигналының өзгеру шамасы, Δt – ағымдағы модельдеу уақытының адымының мәні.

Блоктағы кіріс сигналының мәні есептелу басталғанға дейін нөлге тең болады. Бастапқы шығыс сигналының мәні де нөлге тең болады.



9.3.1-сурет. *Derivative* блогының сигналды дифференциалдау үшін қолданылуы

Туындының есептеу дәлдігі оның есептеуге құрылған адымының шамасына байланысты болады. Яғни, адым шамасы неғұрлым кіші болса, туындының есептелу нәтижесі де соғұрлым нақтырақ болады.

9.3.1-суретте тікбұрышты сигналдың туындысын есептеу үшін дифференциалдаушы блогты қолдану мысалы көрсетілген. Бұл мысалда, көрнекілік үшін, есептеу адымы үлкен етіп алынған.

Берілген блок аналогтық сигналдарды дифференциалдау үшін қолданылады. Ал дискреттік сигналды *Derivative* блогы арқылы дифференциалдау кезінде оның шығыс сигналы дискреттік сигналдың лайықты уақыт моментіндегі секірмелі өзгерісіне сай болатындай импульстер тізбегі түрінде болып келеді.

9.3.2 Integrator интегралдаушы блогы

Қызметі: Кіріс сигналдарын интегралдайды.

Параметрлері:

External reset – сыртқы бастапқы күйге түсіру. Интегратордың бастапқы күйіне түсіруін қамтамасыз ететін сыртқы басқару сигналдарының типі. Ол келесі тізімнен таңдап алынады:

none – жоқ (бастапқы күйге түсіру орындалмайды),

rising – өсетін сигнал (сигналдың алдыңғы сабы),

falling – кемитін сигнал (сигналдың артқы сабы),

either – өсетін немесе кемитін сигнал,

level – нөлдік емес сигнал (егер басқарылушы кірісте сигнал нөлге тең болмаса, онда бастапқы күйге түсіру орындалады);

Егер қандай да бір басқару сигналының типі таңдалса (бірақ *none* емес), онда блоктың суретінде қосымша басқарылушы кіріс пайда болады. Қосымша кірістің жанында басқару сигналының шартты мәні көрсетіледі.

Initial condition source — шығыс сигналының бастапқы мәнінің көзі. Ол келесідей тізімнен таңдалынады:

internal – ішкі.

external – сыртқы. Бұл жағдайда блок кескінінде интегратордың шығыс сигналының бастапқы мәніне тең болатын сигналды беретін x_0 арқылы белгіленген қосымша кіріс пайда болады.

Initial condition — бастапқы шарт. Интегратордың шығыс сигналының бастапқы мәнін қою. Егер шығыс сигналының бастапқы мәнінің ішкі көзі таңдап алынған болса ғана, бұл параметр қолжетімдік дәрежеде болады.

Limit output (жалауша) — шығыс сигналы шегінің пайдаланылуы.

Upper saturation limit — шығыс сигналының шегінің жоғарғы деңгейі. Ол сандық немесе *inf* таңбалық, яғни «+» тізбегі ретінде болып берілуі мүмкін.

Lower saturation limit — шығыс сигналының шегінің төменгі деңгейі. Ол сандық немесе *inf* таңбалық, яғни «-» тізбегі ретінде болып берілуі мүмкін.

Show saturation port — шектеуге шығатын интеграторды растайтын, сигнал шығарушы порттың көрінісін басқарады. Бұл порттың шығыс сигналы келесідей мәндерді қабылдай алады:

Нөл, егер интегратор шектеуде болмаса,

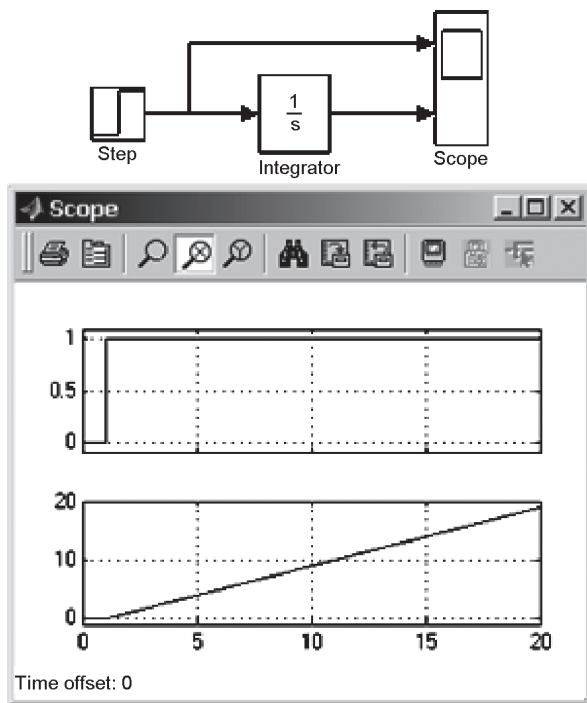
$+1$, егер интегратордың шығыс сигналы жоғарғы шекке жетсе,

-1, егер интегратордың шығыс сигналы төменгі шекке жетсе.

Show state port (жалауша) — блок портының жағдайын көрсету/жасыру. Егер интегратордың шығыс сигналын сол интегратордың кері байланысы сигналы ретінде берсе, онда берілген порт қолданылады.

Absolute tolerance — абсолюттік қателік.

9.3.2-суретте кірісіне сатылық сигнал бергендегі интегратордың жұмысы көрсетілген. Мұнда бастапқы шарт нөлге тең деп берілген.

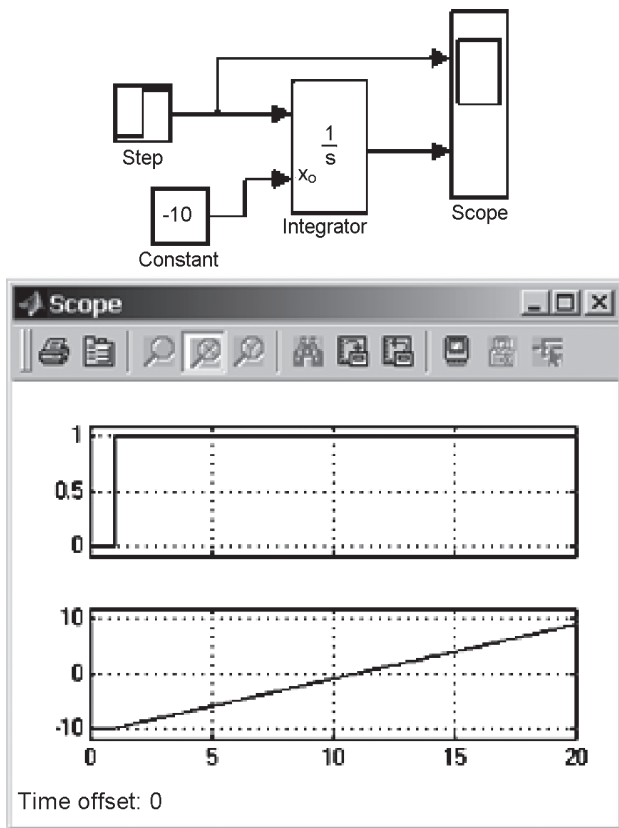


9.3.2-сурет. Сатылық сигналды интегралдау

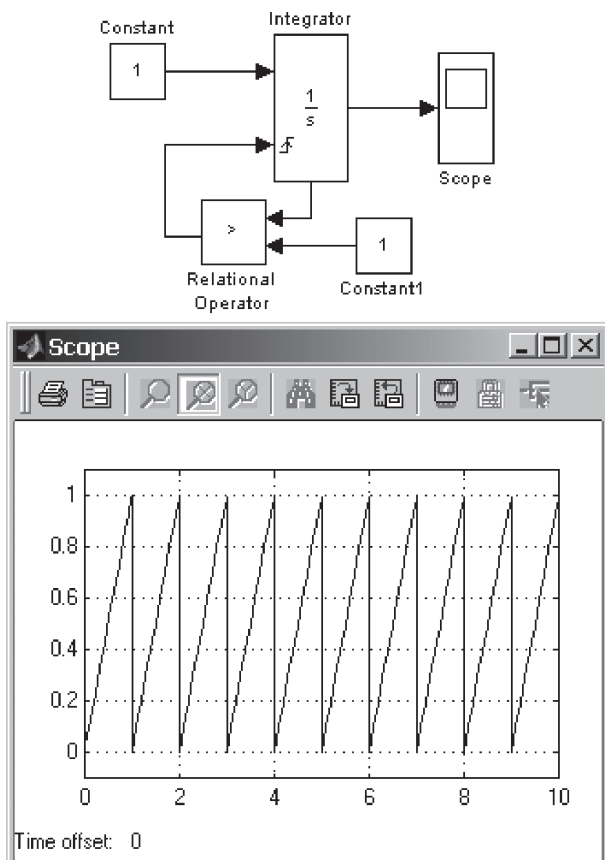
9.3.3-суреттегі алдыңғы мысалдан айырмашылығы бастапқы мәнді сыртқы порт арқылы бергендігінде. Берілген мысалдағы шығыс сигналының бастапқы мәні 10-ға тең.

9.3.4-суреттегі мысал шығыс сигналының бастапқы күйге түсірілуі үшін кіріс портының қолданылуын және кері байланыс орнату қызметінде интегратор портының күйін көрсетеді.

Сұлба келесі түрде жұмыс істейді: тұрақты кіріс сигналы интегратор арқылы сызықтық өзгеретінге түрленеді, шығыс сигналының мәні 1 -ге тең болған кезде *Relational Operator* блок логикалық сигнал шығарады, сол кезде алдыңғы шепте интегратордың бастапқы мәні нөлге тең болғанша шығыс сигналының бастапқы күйге түсірілуі жүреді. Нәтижесінде интегратор шығысында 0 -ден $+1$ -ге дейін өзгеретін ара тәріздес сигнал құралады.

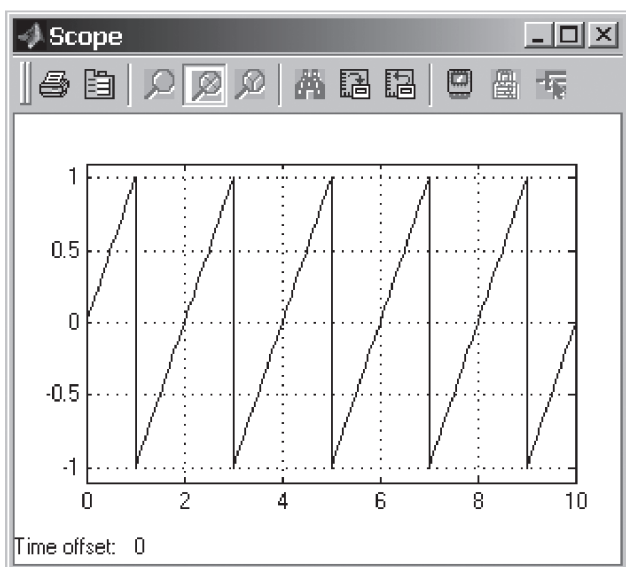
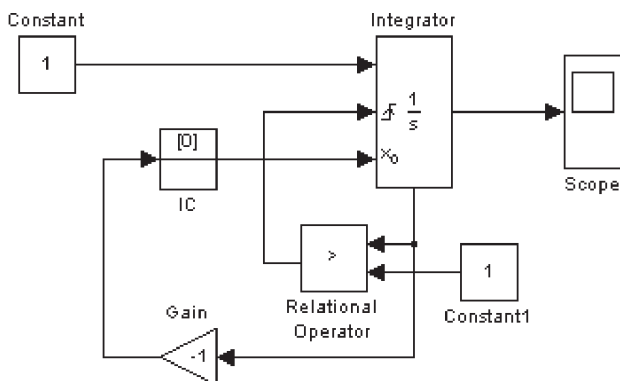


9.3.3-сурет. Шығыс сигналының бастапқы мәнін қою арқылы сатылық сигналды интегралдау



9.3.4-сурет. Интегратор негізіндегі ара тәріздес сигнал генераторы

Келесі сұлба (9.3.5-сурет) интегратордың бастапқы мәнін қою үшін оның шығыс сигналын қолданады. Алғашқы уақытта интегратордың шығыс сигналының бастапқы мәні *IC (Initial Condition)* блогының көмегімен бастапқы уақытта нөлге тең етіп орнатылады. Шығыс сигналымен мән *1*-ге теңескенде *Relational Operator* блогы шығыс сигналын бастапқы деңгейге түсіру сигналын береді, ал мұндағы сигналдың бастапқы деңгейін беретін болып интегратордың кері таңбаланған шығыс сигналы болып саналады (яғни, *-1*). Әрі қарай сұлбаның жұмыс циклі қайталанады. Мұның алдыңғы сұлбадан айырмашылығы генератордың шығыс сигналы болып косполярлы сигнал шығады.



9.3.5-сурет. Интегратор негізіндегі екі полярлық ара тәріздес сигнал генераторы

9.3.3 Memory блогы

Қызметі: Бір уақыттық тактідегі кіріс сигналының бөгелуін орындайды.

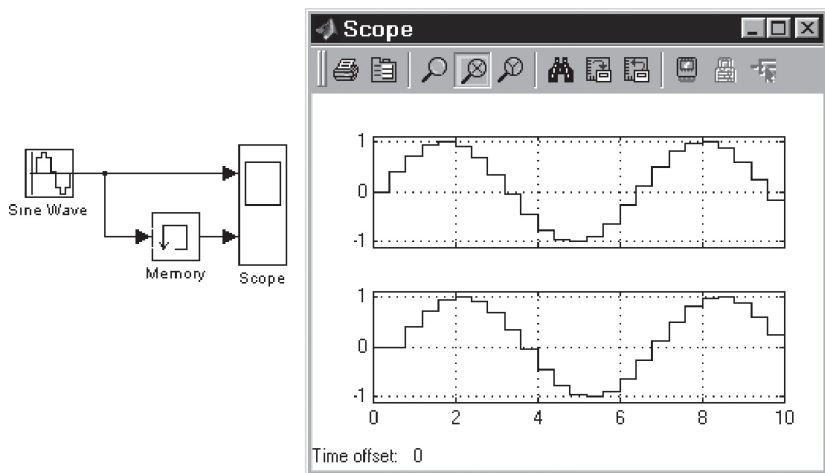
Параметрлері:

Initial condition – шығыс сигналының бастапқы мәні.

Inherit sample time (жалауша) – модельдеу уақытының адымын мұра етіп алу. Егер бұл жалауша орнатылса, онда *Memory* блогы

модельдеу уақытының адымын (*sample time*) дәл алдыңғы блоктағыдай етіп қолданады.

9.3.6.-суретте дискреттік сигналдың бір уақыттық тактіге бөгелуі үшін *Memory* блогының қолданылуы мысалы көрсетілген.



9.3.6-сурет. Сигналды бір уақыттық тактіге бөгелу үшін *Memory* блогының қолданылуы

9.3.4 Transport Delay сигналдың бекітілген бөгеу блогы

Қызметі: Шығыс сигналының берілген уақытқа бөгеуін қамтамасыз етеді.

Параметрлері:

Time Delay — сигналды бөгеу уақыты (теріс емес мән).

Initial input — шығыс сигналының бастапқы мәні.

Buffer size — бөгелген сигналды сақтайтын жадтың көлемі. Ол 8 еселік санмен байт арқылы беріледі (үнсіздікпен 1024).

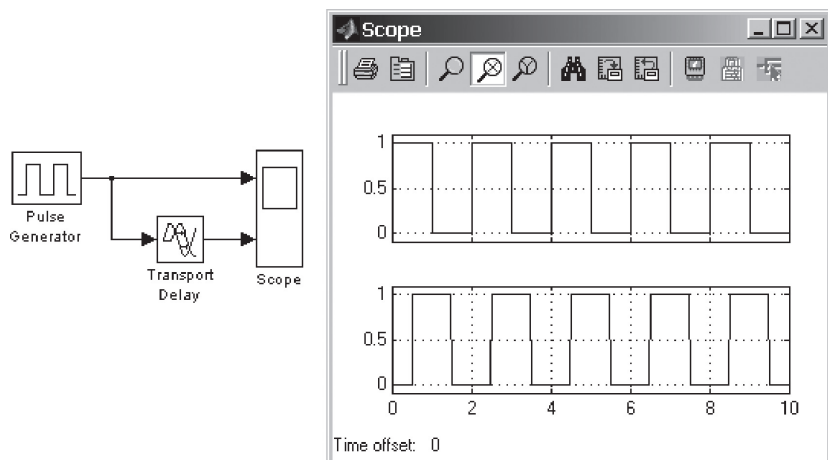
Pad order (for linearization) — Паде қатарының тәртібі. Ол шығыс сигналын жуықтатылуы кезінде қолданылады және бүтін оң санмен беріледі.

Модельдеу кезінде *Transport Delay* блогының ішкі буферінде сигналдың мәні мен оған сәйкес оның модельдеу уақыты сақталады. Сигналдың бөгелу уақыты біткен кезде, ол буферден шығарылып, содан соң блоктың шығысына беріледі. Егер

модельдеу уақытының адымы буферде жазылған сигналдың моменттік уақытының мәніне сәйкес келмесе, онда *Transport Delay* блогы шығыс сигналының жуықтатылуын орындайды.

Егер буфердегі жады көлемінің бастапқы шамасы бөгелген сигналды сақтауға жетпесе, онда Simulink бірден қосымша жад бөледі. Модельдеу аяқталған соң MATLAB-тың командалық терезесінде буфердің керекті мөлшерін көрсететін хабарлама пайда болады.

9.3.7-суретінде тік бұрыштық сигналдың $0.5c$ -қа бөгелуі үшін *Transport Delay* блогын қолданудың мысалы көрсетілген.



9.3.7-сурет. Сигналдың бөгелуі үшін *Transport Delay* блогын қолдану

9.3.5 Variable Transport Delay бөгеуі басқарылатын сигналдың блогы

Қызметі: Басқарылатын сигнал шамасымен берілетін кіріс сигналының бөгелуін орындайды

Параметрлері:

Maximum delay — сигналды бөгеу уақытының ең үлкен мәні (теріс емес мән)

Initial input — шығыс сигналының бастапқы мәні.

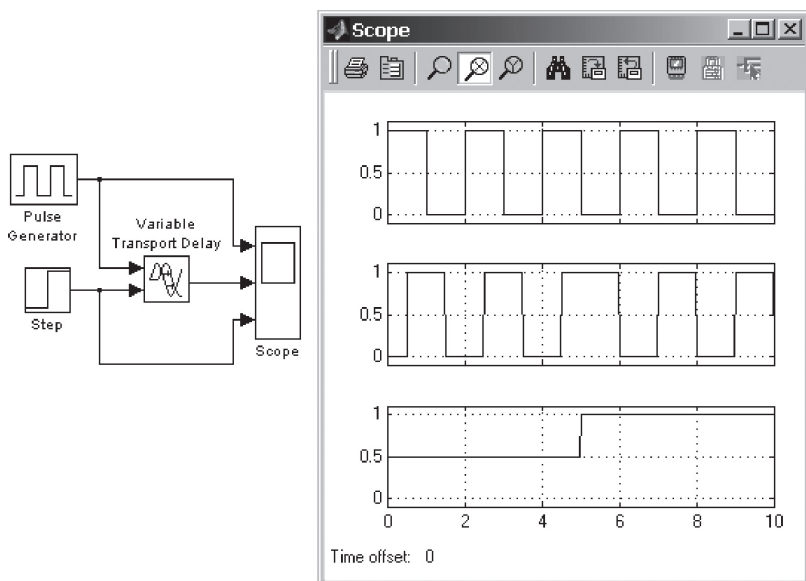
Buffer size — бөгелген сигналды сақтау үшін бөлінген жадтың көлемі. Ол 8 еселі санмен байт арқылы беріледі. (үнсіздікпен 1024).

Pade order (for linearization) — шығыс сигналының жуықта-тылуы кезінде қолданылатын Паде қатарының тәртібі. Ол бүтін, әрі оң санмен беріледі.

Бөгеуі басқарылатын *Variable Transport Delay* блогы тұрақты бөгеуші *Transport Delay* блогына сәйкес жұмыс істейді.

Егер бөгеу мөлшерін беретін басқарушы сигналдың мәні *Maximum delay* параметрімен берілген мәннен артық болса, онда бөгеу *Maximum delay*-мен мөлшерлес болып орындалады.

9.3.8-суретте *Variable Transport Delay* блогының қолданылу мысалы көрсетілген. Уақыт моменті *5c.*-қа тең болғандағы сигналдың бөгеу уақытының мөлшерінің өзгерісі *0.5* пен *1c.* аралығында болады.



9.3.8-сурет. *Variable Transport Delay* блогының қолданылуы

9.3.6 Transfer Fcn блогының функциясы

Қызметі:

Беріліс сипаттамасының блогы *TransferFcn* беріліс функ-циясын

$$H(s) = \frac{y(s)}{u(s)} = \frac{num(s)}{den(s)} = \frac{num(1)s^{m-1} + num(2)s^{m-2} + \dots + num(nn)}{den(1)s^{nd-1} + den(2)s^{nd-2} + \dots + den(nd)}$$

түріндегі полиномдар қатынасы ретінде беріледі. Мұндағы m және nd – беріліс функциясының алымы мен бөлімінің реттік сандары, num – алымдық коэффициенттердің векторы немесе матрицасы, den – бөлімдік коэффициенттердің векторы.

Параметрлері:

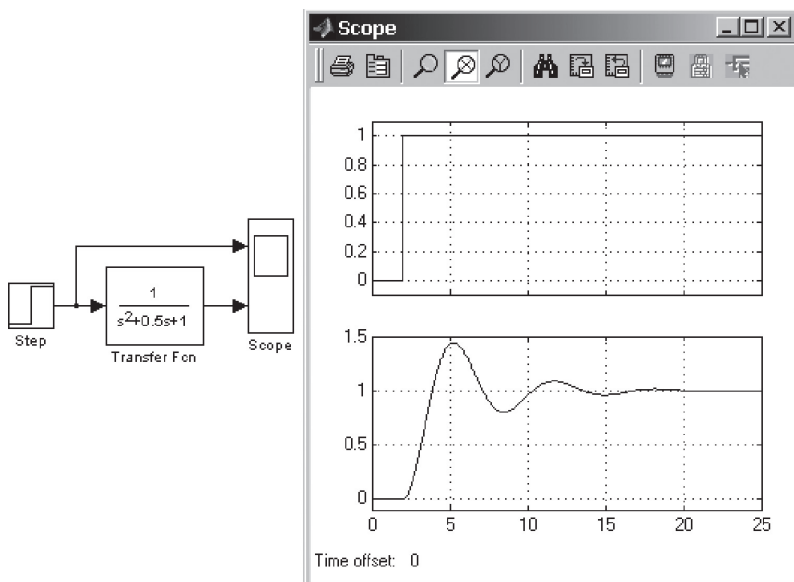
Numerator — алымдық полиномының коэффициенттерінің векторы немесе матрицасы.

Denominator – бөлімдік полиномының коэффициенттерінің векторы.

Absolute tolerance — абсолюттік қателік.

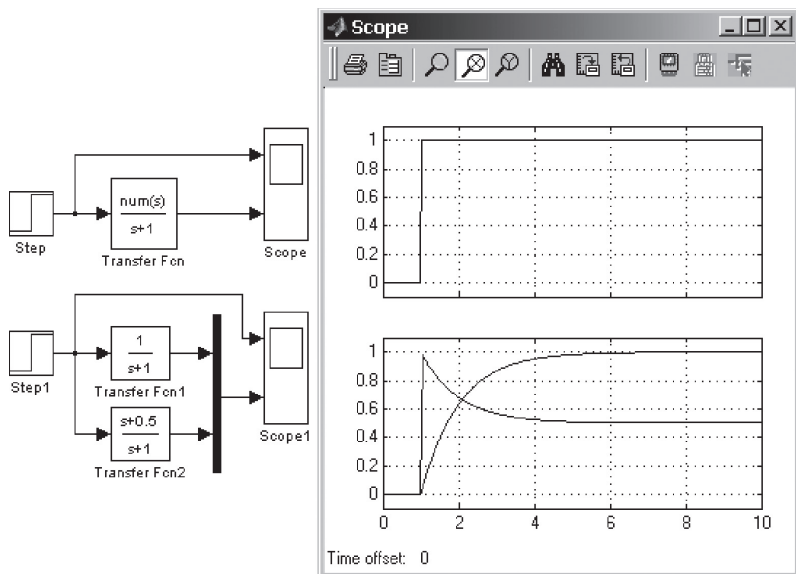
Алымдық полиномның реттік саны бөлімінікінен аспауы керек.

Блоктың кіріс сигналы скалярлы болуы тиіс. Егер алым коэффициентінің реті вектормен берілсе, онда блоктың шығыс сигналы да скалярлық болады (кіріс сигналы сияқты). 9.3.8-суретте *TransferFcn* блогы арқылы тербелмелі буынды модельдеу мысалы көрсетілген.



9.3.8-сурет. Тербелмелі буынды модельдеу

Егер алым коэффициенті матрицамен берілсе, онда полиномы бірдей бөлімдік, бірақ алымдық полиномдары әртүрлі болатын бірнеше беріліс функциясы ретінде түсіндірілетін *Transfer Fcn* блогы векторлық беріліс функциясын модельдейді.



9.3.9-сурет. Векторлық беріліс функциясын және оның аналогын модельдеу

Сондай-ақ блоктың шығыс сигналы векторлық болып табылады және алымның матрицасының жол саны шығыс сигналының мөлшерімен беріледі. 9.3.9-суретте *Transfer Fcn* блогының векторлық беріліс функциясы көрсетілген. Сонымен қатар мұнда *Transfer Fcn*-ның жеке блоктарынан тұратын, қарастырып отырған модельге қасиеті жағынан толық ұқсайтын, модель көрсетілген.

Transfer Fcn блогын қолданғанда функцияның бастапқы шарты нөлдік болуы мүмкін. Егер бастапқы шарт нөлдік болмаса, онда *tf2ss* (*ControlSystemToolbox* құралы) функциясының көмегімен беріліс функциясынан кеңістіктегі күй моделіне және *State-Space* блогының көмегімен динамикалық объектіні модельдеуге өту керек.

9.3.7 Zero-Pole беріліс функциясының блогы

Қызметі: Zero-Pole блогы полюстері мен нөлдері берілген

$$H(s) = K \frac{Z(s)}{P(s)} = K \frac{(s - Z(1))(s - Z(2)) \dots (s - Z(m))}{(s - P(1))(s - P(2)) \dots (s - P(n))}$$

түріндегі беріліс функциясын анықтайды. Мұндағы Z – беріліс функциясының нөлдерінің векторы немесе матрицасы (полином алымының түбірлері), P – беріліс функциясының полюс векторы, K – егер беріліс функциясының нөлдері матрицамен берілгендегі беріліс функциясының коэффициенті немесе коэффициенттер векторы беріледі. Сондай-ақ вектордың өлшемі K нөлдік матрицаның жол санымен анықталады.

Параметрлері:

Zeros – нөлдердің векторы немесе матрицасы.

Poles – полюстердің векторы.

Gain – беріліс функциясының скалярлық немесе векторлық коэффициенттері.

Absolute tolerance — абсолюттік қателік.

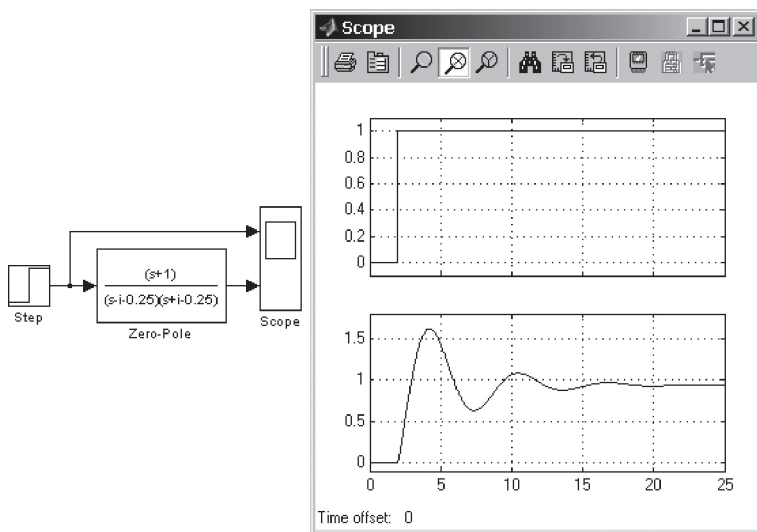
Полюстер саны беріліс функциясының нөлдерінің санынан артпауы қажет.

Егер беріліс функциясы нөлдер матрицасымен берілсе, онда Zero-Pole блок вектордың беріліс функциясын модельдейді.

Нөлдер немесе полюстер комплекстік сандар болып берілуі мүмкін. Мұндай жағдайда нөлдер немесе полюстер олардың комплекстік-түйіндес жұптары ретінде беріледі.

Zero-Pole блогын қолданғанда бастапқы шарт нөлдік болып қабылданады.

9.3.10-суретте Zero-Pole блогының қолдану мысалы көрсетілген. Бұл мысалда беріліс функциясы бір нақты нөлдік және екі комплекстік-түйіндес полюске ие.



9.3.10-сурет. *Zero-Pole* блогының қолданылуы

9.3.8 State-Space динамикалық объекті моделінің блогы

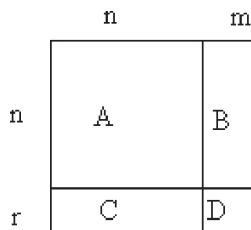
Қызметі: Блок кеңістіктегі

$$\dot{x} = Ax + Bu,$$

$$y = Cx + Du,$$

түріндегі күй теңдеуімен сипатталатын динамикалық объектіні құрайды, мұндағы x – күй векторы, u – кіріс әсерлерінің векторы, y – шығыс сигналының векторы, A , B , C , D – тиісінше, жүйенің, кірістің, шығыстың және айналып өту матрицалары.

Матрица өлшемі 9.3.11-суретте көрсетілген (n – күй айнымалыларының саны, m – кіріс сигналдарының саны, r – шығыс сигналдарының саны).



9.3.11-сурет. *State-Space* блогының матрицаларының өлшемдері

Параметрлері:

A – жүйенің матрицасы.

B – кірістің матрицасы.

C – шығыстың матрицасы.

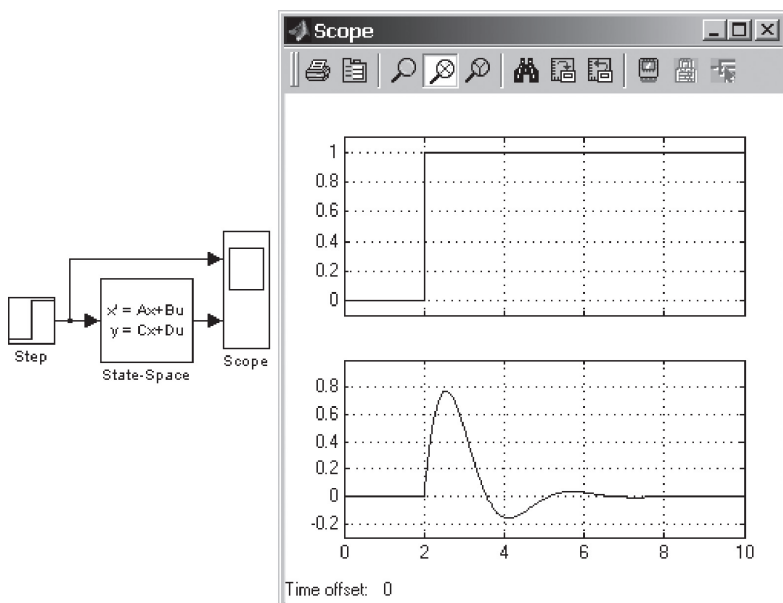
D – айналып өтудің матрицасы.

Initial condition – бастапқы шарт векторы.

Absolute tolerance — абсолюттік қателік.

9.3.11-суретте динамикалық объектіні *State-Space* блогының көмегімен модельдеудің мысалы көрсетілген. Блок матрицалары төмендегідей мәндерге ие:

$$A = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}, C = [0 \quad 1], D = [0].$$



9.3.12-сурет. *State-Space* блогын қолдану

9.4 Discrete – дискреттік блоктар

9.4.1 Unit Delay жеке дискреттік бөгеу блогы

Қызметі: Кіріс сигналын моделдеу уақытының бір адымына бөгеуді орындайды.

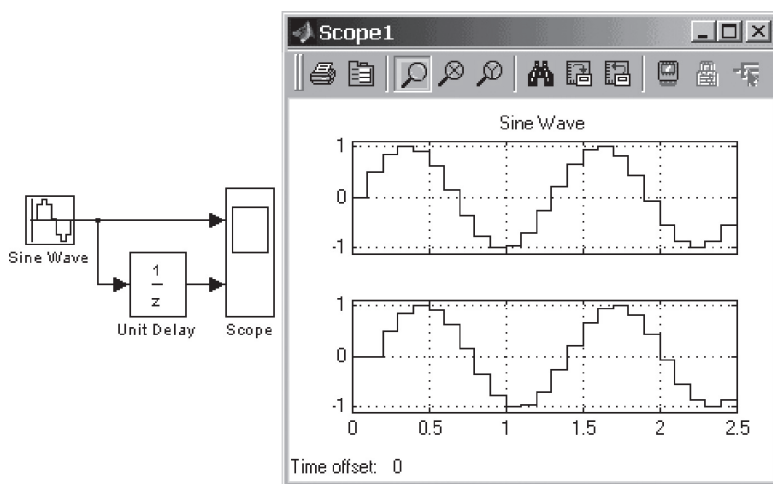
Параметрлері:

Initial condition – шығыс сигналының бастапқы мәні.

Sample time – модельдеу уақытының адымы.

Блоктың кіріс сигналы скалярлық та, векторлық та болуы мүмкін. Егер кірісте векторлық шама болса, онда бөгелу вектордың әрбір элементі үшін орындалады. Блок комплекстік те, нақты да сигналдармен жұмыс істей береді.

9.4.1-суретте дискреттік сигналды $0.1c$ тең болатын бір уақыт адымына бөгеу үшін блокты қолданудың мысалы көрсетілген.



9.4.1-сурет. *Unit Delay* блогын қолдану

9.4.2 Zero-Order Hold нөлінші реттік экстраполятордың блогы

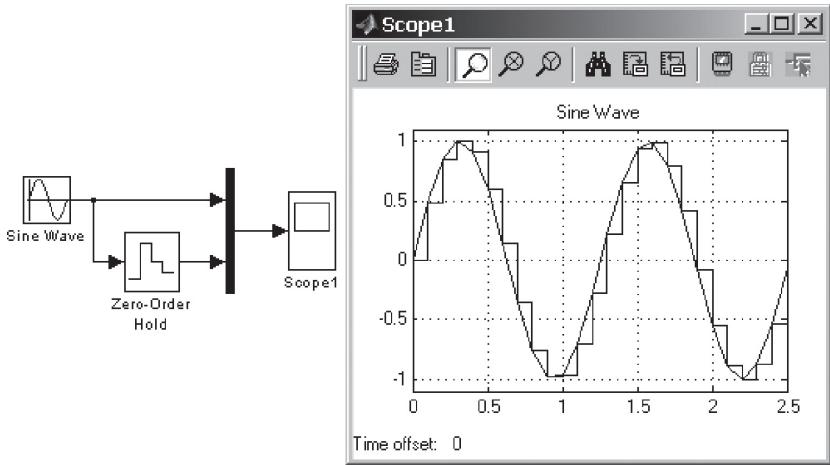
Қызметі: Блок кіріс сигналын уақыт бойынша дискреттейді.

Параметрлері:

Sample time – дискреттеудің уақыт бойынша адымның мөлшері.

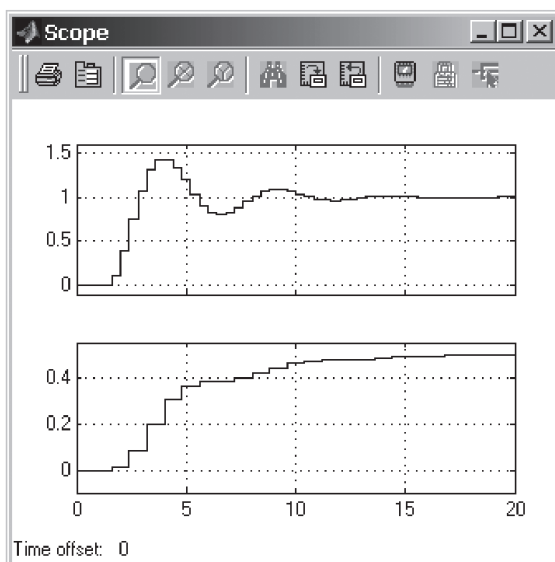
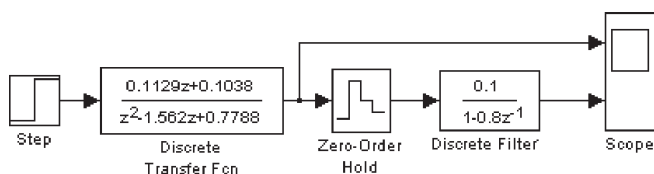
Блок кіріс сигналының мәнін кванталудың аралығының бастапқы кезінде жазып алады да, кванталу аралығы аяқталғанша шығысында ұстап тұрады. Содан соң кванталудың келесі адымында шығыс сигналы секіре өзгеріп, кірістік сигналдың мөлшеріне жетеді.

9.4.2-суретте дискреттік сигналды қалыптастыру үшін *Zero-Order Hold* блогын қолданудың мысалы көрсетілген.



9.4.2-сурет. *Zero-Order Hold* блогының көмегімен дискреттік сигналдың қалыптасуы

Сонымен қатар, нөлінші реттік экстраполятордың блогы әртүрлі кванталу аралықтары бар дискреттік блоктардың бір-бірлерімен қиыса жұмыс істеулері үшін қолданылуы мүмкін. Оны 9.4.3-суреттегі *Zero-Order Hold* блогын қолданудың мысалынан көруге болады. Мұнда *Discrete Transfer Fcn* блогы келесідей параметрге ие: *Sample time* = 0.4, ал *Discrete Filter* блогы үшін бұл параметр 0.8-ге тең етіп орнатылған.



9.4.3-сурет. Дискреттік блоктардың қысқа жұмыс істеуі үшін *Zero-Order Hold* блогының қолданылуы

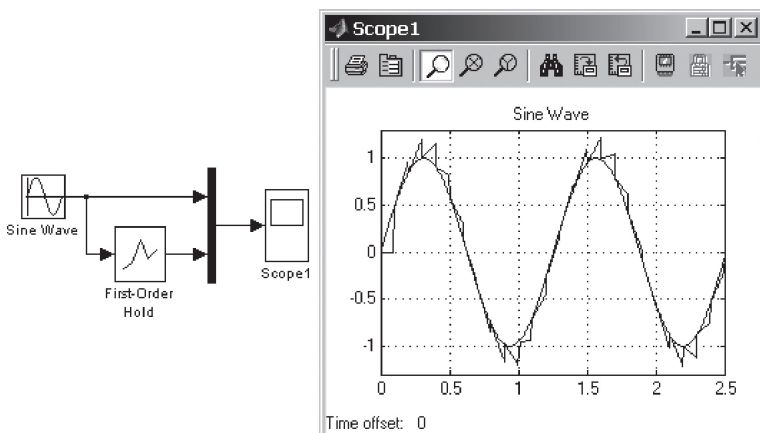
9.4.3 First-Order Hold бірінші реттік экстраполятордың блогы

Қызметі: Блок дискреттеудің алдыңғы аралығындағы кіріс сигналының құлдылығына сәйкес, дискреттеудің әрбір тактісінде шығыс сигналының сызықтық өзгерісін береді.

Параметрлері:

Sample time – уақыт бойынша дискреттеудің адым мөлшері.

9.4.4-суретте синусоидалық сигналдың осы блокпен экстраполяциясының мысалы көрсетілген.



9.4.4-сурет. *First-Order Hold* блогының қолданылуы

9.4.4 Discrete-Time Integrator дискретті интегратордың блогы

Қызметі: Блок дискреттік жүйелерде интегралдау операциясын орындау үшін қолданылады.

Параметрлері:

Integration method – сандық интегралдау әдісі.

Forward Euler – Эйлердің тура әдісі.

Бұл әдіс $1/s$ беріліс функциясының $T/(z-1)$ жуықтауын қолданады. Блоктың шығыс сигналы келесі өрнекпен есептеледі:

$$y(k) = y(k-1) + T * u(k-1),$$

мұндағы y – интегратордың шығыс сигналы, u – интегратордың кіріс сигналы, T – дискреттеу адымы, k – модельдеу адымының нөмірі.

Backward Euler – Эйлердің кері әдісі.

Бұл әдіс $1/s$ беріліс функциясының $T * z / (z-1)$ жуықтауын қолданады. Блоктың шығыс сигналы келесі өрнекпен есептеледі:

$$y(k) = y(k-1) + T * u(k).$$

Trapeziodal – трапеция әдісі.

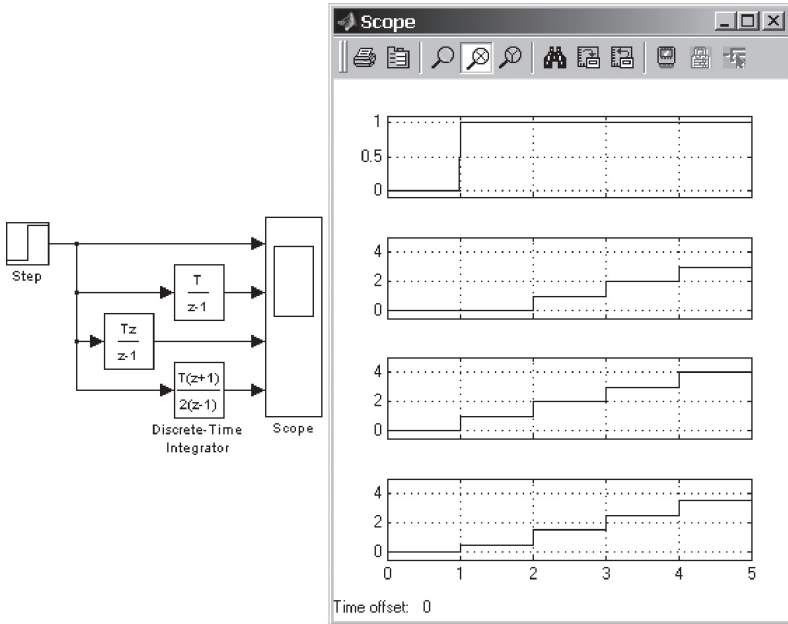
Бұл әдіс $1/s$ беріліс функциясының $T/2 * (z+1)/(z-1)$ жуықтауын қолданады. Блоктың шығыс сигналы келесі өрнекпен есептеледі:

$$x(k) = y(k-1) + T/2 * u(k-1).$$

Sample time — дискреттеудің уақыт бойынша адымы.

Дискреттік интегратордың қалған параметрлері блоктың *Integrator (Continuous* кітапханасы) – аналогтық интеграторымен бірдей.

9.4.5-суретте *Discrete-Time Integrator* блогының сандық интегралдаудағы жоғарыдағы көрсетілген үш әдісі де келтірілген. Суретте көрсетілгендей блоктың көрінісі таңдап алынған интегралдау әдісіне байланысты өзгереді.



9.4.5-сурет. Әртүрлі сандық әдістерді қолданатын *Discrete-Time Integrator* блогы арқылы интегралдауды орындау

9.4.5 Discrete Transfer Fcn дискреттік беріліс функциясы

Қызмемі: *Discrete Transfer Fcn* блогы дискреттік беріліс функциясын келесідей полиномдар қатынасы ретінде береді, яғни

$$H(z) = \frac{num(z)}{den(z)} = \frac{num_0 z^n + num_1 z^{n-1} + \dots + num_m z^{n-m}}{den_0 z^n + den_1 z^{n-1} + \dots + den_n}$$

мұндағы m және n – тиісінше, алымы мен бөлімінің коэф-

фициенттерінің саны, num – алымының коэффициенттерінің матрицасы немесе векторы, den – бөлімінің коэффициенттерінің векторы.

Параметрлері:

Numerator — алым коэффициенттерінің матрицасы немесе векторы.

Denominator – бөлім коэффициенттерінің векторы.

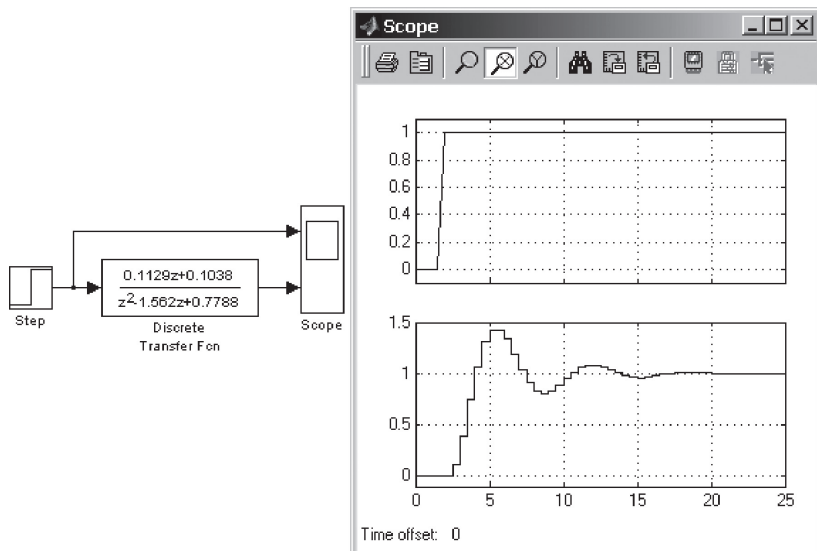
Sample time — дискреттеудің уақыт бойынша адымы.

Алымның реттігі бөлімінің реттігінен аспауы керек.

Блоктың кіріс сигналы скалярлық болуы керек. Егер алымының коэффициенті вектор ретінде берілсе, онда шығыс сигналы скалярлық болады (кіріс сигнал сияқты). 9.4.6-суретте *Discrete Transfer Fcn* блогының қолданылу мысалы көрсетілген. Мұнда тербелмелі буынның дискреттік аналогының жеке сатылық әсерге реакциясы есептелінген:

$$\frac{1}{s^2 + 0.5s + 1}$$

Дискреттеу адымы $0.5c$.-қа тең деп алынған.



9.4.6-сурет. *Discrete Transfer Fcn* блогының қолданылуы

9.4.6 Discrete Zero-Pole дискреттік беріліс функциясының блогы

Қызметі: *Discrete Zero-Pole* блогы полюстермен және нөлдермен берілген дискреттік беріліс функциясын анықтайды, яғни

$$H(z) = K \frac{Z(z)}{P(z)} = K \frac{(z - Z_1)(z - Z_2) \dots (z - Z_m)}{(z - P_1)(z - P_2) \dots (z - P_n)},$$

мұндағы Z – беріліс функцияның нөлдерінің векторы немесе матрицасы, P – беріліс функцияның полюстерінің векторы, K – беріліс функциясының нөлдері матрица ретінде берілген болғандағы беріліс функциясының коэффициенті немесе коэффициенттер векторы. Ал K векторының өлшемі нөлдер матрицасының жол сандарына тең.

Параметрлері:

Zeros – нөлдер векторы немесе матрицасы.

Poles – полюстердің векторы.

Gain – беріліс функцияның скалярлық немесе векторлық коэффициенті.

Sample time — уақыт бойынша дискреттеудің адымы.

Нөлдер саны беріліс функциясының полюс санынан аспауы керек. Егер беріліс функцияның нөлдері матрица ретінде берілген болса, онда *Discrete Zero-Pole* блогы векторлық беріліс функциясын модельдейді.

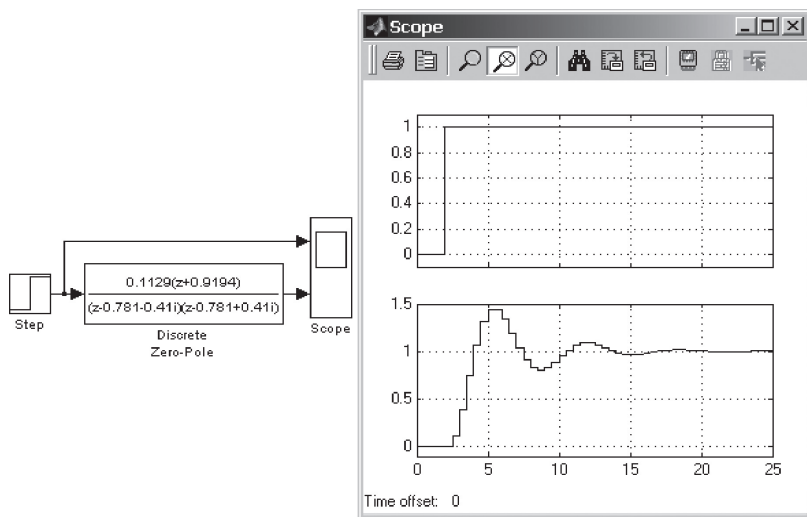
Нөлдер немесе полюстер комплекстік сандар ретінде де берілуі мүмкін. Мұндай жағдайда нөлдер немесе полюстер, тиісінше, нөлдер мен полюстердің комплекстік-түйіндес жұптары арқылы беріледі.

Discrete Zero-Pole блогын қолдануда бастапқы шарттар нөлдік болып келеді.

9.4.7-суретте *Discrete Zero-Pole* блогын қолданудың мысалы көрсетілген. Көрсетілген блоктың арқасында беріліс функцияның дискреттік аналогы модельденеді.

$$\frac{1}{(s + 0.25 \cdot 0.968i)(s + 0.25 + 0.968i)}$$

Дискреттеу адымы 0.5 с.-ге тең болып алынған.



9.4.7-сурет. *Discrete Zero-Pole* блогын қолдану

9.4.7 Discrete Filter дискреттік сүзгі блогы

Қызмеі: *Discrete Filter* дискреттік сүзгі блогы ($1/z$) кері аргументінен дискреттік беріліс функциясын береді, яғни

$$H(1/z) = \frac{num(1/z)}{den(1/z)} = \frac{num_0 z^0 + num_1 z^{-1} + num_2 z^{-2} + \dots + num_m z^{-m}}{den_0 z^0 + den_1 z^{-1} + den_2 z^{-2} + \dots + den_n z^{-n}}$$

m и n – тиісінше, алымы мен бөлімінің коэффициенттерінің саны, num – алымдық коэффициенттердің векторы немесе матрицасы, den – бөлімдік коэффициенттерінің векторы.

Параметрлері:

Numerator — алымдық коэффициенттердің векторы немесе матрицасы.

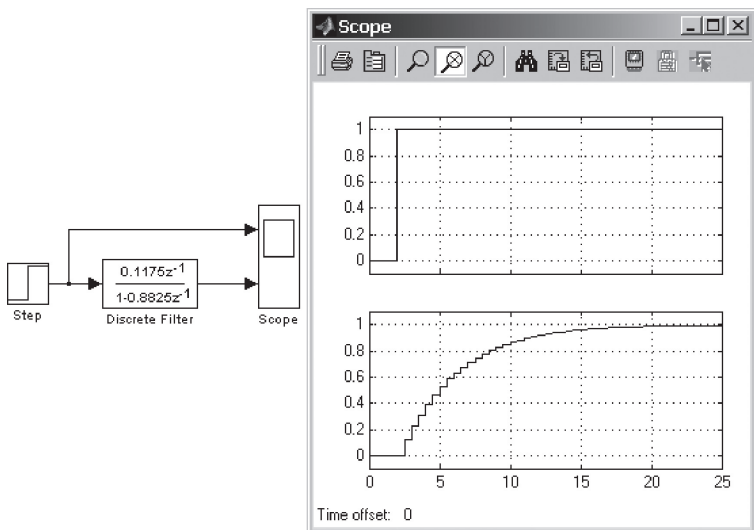
Denominator – бөлімдік коэффициенттерінің векторы.

Sample time — уақыт бойынша дискреттеудің адымы.

9.4.8-суретте *Discrete Filter* блогын қолданудың мысалы көрсетілген. Қарастырылып отырған блоктың көмегімен беріліс функциясының дискреттік аналогы модельденеді:

$$\frac{1}{4s + 1}$$

Дискреттеу адымы $0.5c$ -ге тең деп алынған.



9.4.8-сурет. Discrete Filter блогын колдану

9.4.8 Discrete State-Space динамикалық объекті моделінің блогы

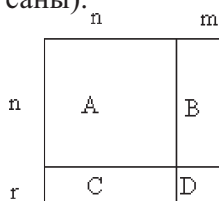
Қызметі: Блок кеңістіктегі

$$x(n+1) = Ax(n) + Bu(n),$$

$$y(n) = Cx(n) + Du(n),$$

түріндегі күй теңдеуімен сипатталатын динамикалық объектіні құрайды, мұндағы x – күй векторы, u – кіріс әсерлерінің векторы, y – шығыс сигналының векторы, A, B, C, D – тиісінше, жүйенің, кірістің, шығыстың және айналып өту матрицалары, n – модельдеу адымының нөмірі.

Матрицаның өлшемі 9.4.9-суретте көрсетілген (n - күй айнымалыларының саны, m - кіріс сигналдарының саны, r - шығыс сигналдарының саны).



9.4.9-сурет. Discrete State-Space блогының матрицаларының өлшемдері

Параметрлері:

A – жүйенің матрицасы.

B – кірістің матрицасы.

C – шығыстың матрицасы.

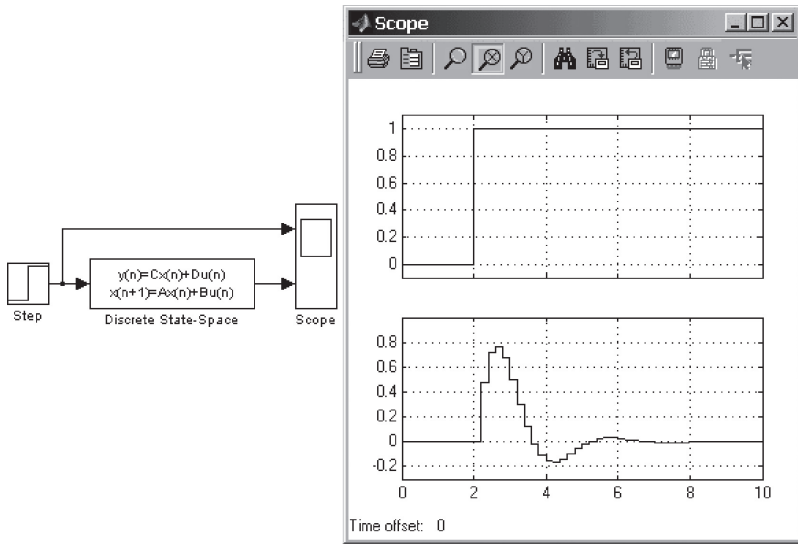
D – айналып өтудің матрицасы.

Initial condition – бастапқы шарт векторы.

Sample time — уақыт бойынша дискреттеудің адымы.

9.4.10-суретте динамикалық объектіні *Discrete State-Space* блогының көмегімен модельдеудің мысалы көрсетілген. Блок матрицалары төмендегідей мәндерге ие:

$$A = \begin{bmatrix} 0.9135 & 0.1594 \\ -0.7971 & 0.5947 \end{bmatrix}, B = \begin{bmatrix} 0.05189 \\ 0.4782 \end{bmatrix}, C = [0 \quad 1], D = [0].$$



9.4.10-сурет. *Discrete State-Space* блогын қолдану

9.5 Nonlinear – сызықсыз блоктар

9.5.1 Saturation шектеу блогы

Қызметі:

Сигнал шамасының шектеу аралығын анықтайды.

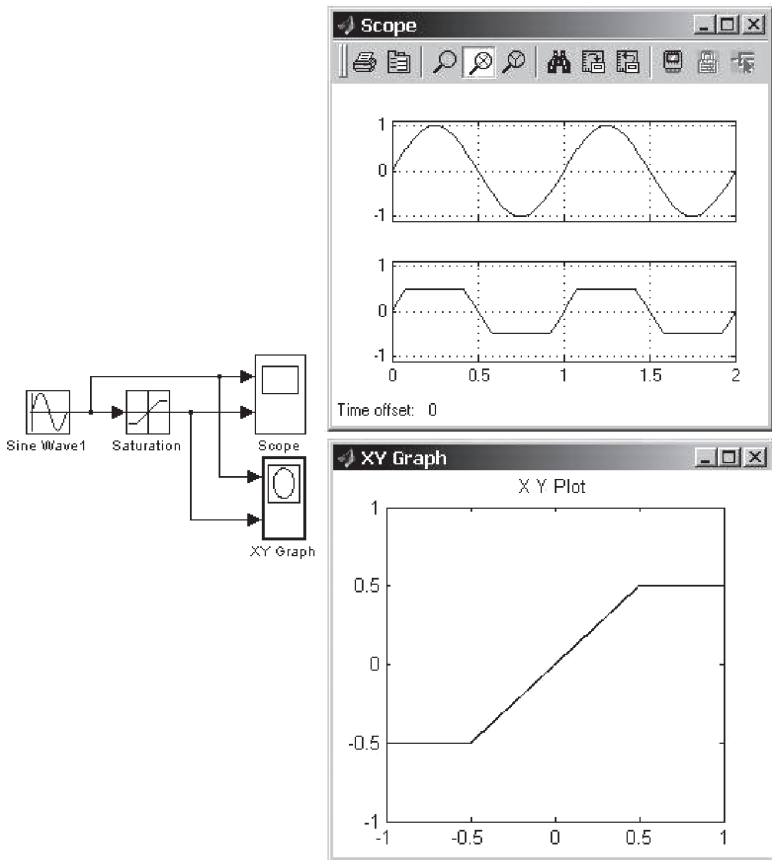
Параметрлері:

Upper limit – шектеудің жоғарғы шегі.

Lower limit – шектеудің төменгі шегі.

Treat as gain when linearizing (жалалуша) – сызықтандыруды орындау кезіндегі беріліс коэффициенті 1-ге тең күшейткіш деп түсіндіру.

Егер блоктың шамасы шектен аспаса, шығыс сигналы кіріс сигналына тең болады. 9.5.1-суретте синусоидалық сигналды шектеу мысалы көрсетілген. Бұл суретте сигналдардың уақыттық диаграммалары және шығыс сигналының кіріс сигналынан тәуелділігі келтірілген.



9.5.1-сурет. *Saturation* блогын қолдану

9.5.2 Dead Zone сезімсіздік аймақ блогы

Қызметі: «Сезімсіздік аймағы (өлі аймақ)» түріндегі сызықсыз тәуелділікті орындайды.

Параметрлері:

Start of dead zone – «Өлі аймақтың» басы (төменгі шек).

End of dead zone – «Өлі аймақтың» соңы (жоғарғы шек).

Saturate on integer overflow (жалауша) – бүтіннің артып кетуін басу. Жалаушаның орнатылған кезінде бүтін типті сигналдарды шектеу дұрыс жүреді.

Treat as gain when linearizing (жалауша) - сызықтандыруды орындау кезіндегі беріліс коэффициенті 1-ге тең күшейткіш деп түсіндіру.

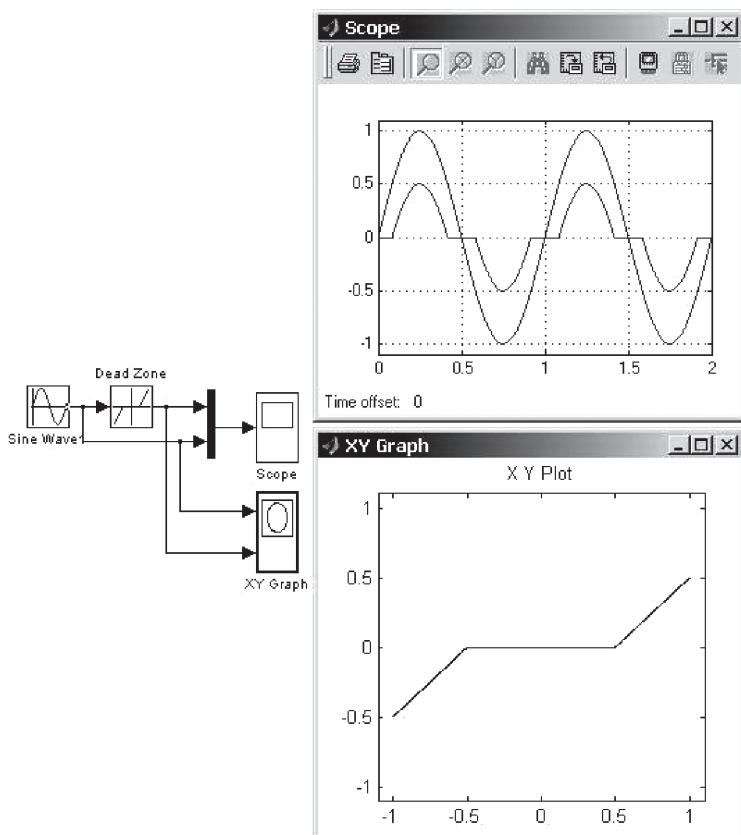
Шығыс сигналы келесі алгоритм бойынша есептеледі:

- Егер кіріс сигналының шамасы өлі аймақтың шегінде болса, онда блоктың шығыс сигналы 0-ге тең.

- Егер кіріс сигналы өлі аймақтың жоғарғы кіріс шегінен үлкен немесе тең болса, онда шығыс сигналы кіріс шегінің азайту шамасына тең.

- Егер кіріс сигналы өлі аймақтың төменгі кіріс шегінен кіші немесе тең болса, онда шығыс сигналы кіріс шегінің азайту шамасына тең.

9.5.2-суретте *Dead Zone* блогын қолданудың мысалы көрсетілген.



9.5.2-сурет. *Dead Zone* блогын қолдану

9.5.3 Relay релелік блогы

Қызметі: Релелік сызықсыздықты іске асырады.

Параметрлері:

Switch on point – қосылу шегі. Бұл мәнде реле қосылады.

Switch off point – өшіру шегі. Бұл мәнде реле өшіріледі.

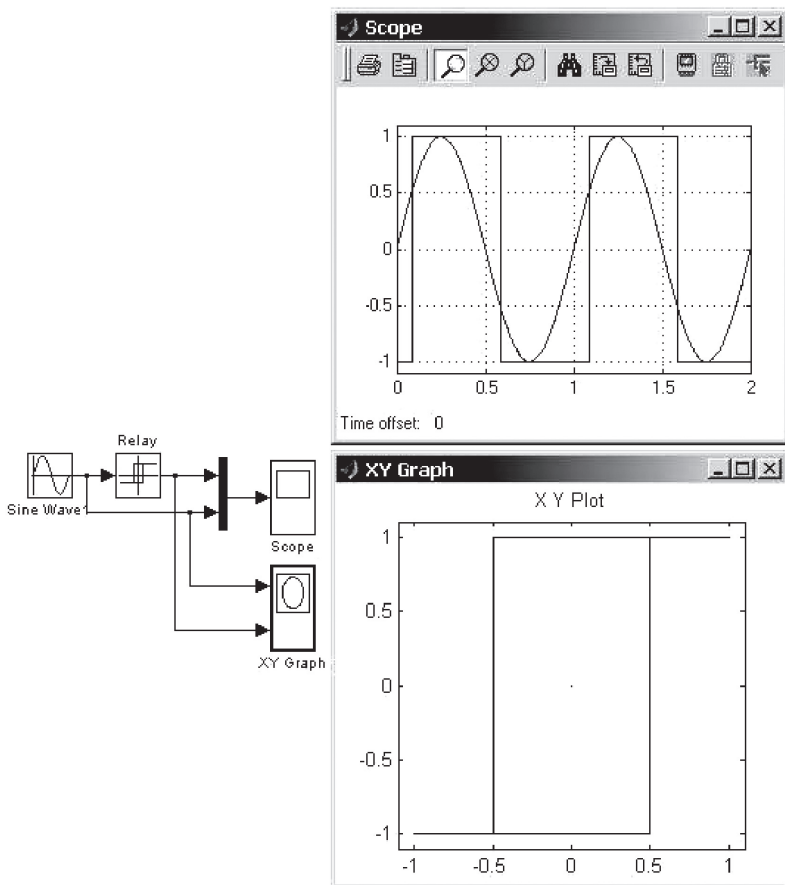
Output when on – қосылыс күйіндегі шығыс сигналының шамасы.

Output when off – өшірулі күйдегі шығыс сигналының шамасы.

Блоктың шығыс сигналы екі мәнге ие болады. Біреуі реленің қосылу күйіне, ал екіншісі өшірулі күйіне сай болады. Бірінші күйден екіншіге күйге өтуі кіріс сигналының реленің қосылулы

не өшірулі шегіне жеткенде секіріспен жүреді. Егер реленің қосылуы және өшірулі шектері әртүрлі мәнге ие болса, онда блок гистерезисі бар релелік сипаттаманы іске асырады. Сол кезде қосылу шегінің мәні өшірілу шегінің мәнінен үлкен болуы керек.

9.5.3-суретте *Relay* блогын қолданудың мысалы көрсетілген. Реленің қосылуы кіріс сигналының шамасы $0,5$ -ке жеткенде, ал өшірілуі $0,5$ -те екені уақыттық диаграммадан көрінеді.



9.5.3-сурет. *Relay* блогын қолдану

9.5.4 Rate Limiter сигналдың өзгеру жылдамдығын шектеу блогы

Қызметі: Блок сигнал өзгерісінің жылдамдығының шектелуін қамтамасыз етеді (бірінші туынды).

Параметрлері:

Rising slew rate – сигналдың ұлғайтылуы кезіндегі жылдамдық шегінің деңгейі.

Falling slew rate – сигналдың азайтылуы кезіндегі жылдамдық шегінің деңгейі.

Сигналдың туындысын есептеу

$$rate = \frac{u(i) - y(i-1)}{t(i) - t(i-1)}$$

өрнегімен орындалады. Мұндағы $u(i)$ – осы ағымдық адымдағы кіріс сигналының мәні, $t(i)$ – осы адымдағы уақыт мәні, $y(i-1)$ – алдыңғы адымдағы шығыс сигналының мәні, $t(i-1)$ – алдыңғы адымдағы модельдеу уақытының мәні.

Есептелген туындының мәні жылдамдықтың шегінің деңгейінің мәндерімен (*Rising slew rate* және *Falling slew rate*) салыстырылады. Егер туындының мәні *Rising slew rate* параметрінің мәнінен көп болса, онда блоктың шығыс сигналы төмендегідей есептеледі:

$$y(i) = \Delta t \cdot R + y(i-1),$$

мұндағы R – сигнал артқандағы жылдамдық шегінің деңгейі.

Егер туынды мәні *Falling slew rate* параметрінің мәнінен кем болса, онда блоктың шығыс сигналы төмендегідей есептеледі:

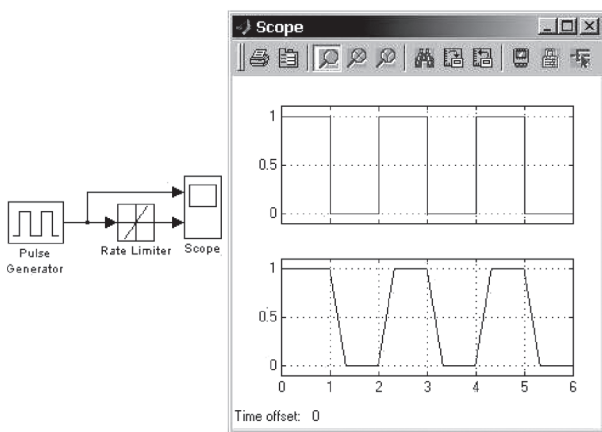
$$y(i) = \Delta t \cdot F + y(i-1),$$

мұндағы F – сигнал кемігендегі жылдамдық шегінің деңгейі.

Егер туынды мәні шектеу дәрежесінің төменгі және жоғарғы шегінің арасында жатса, онда блоктың шығыс сигналы кіріс сигналына тең:

$$y(i) = u(i).$$

9.5.4-суретте *Rate Limiter* блогының кірісіне тікбұрышты периодтық сигнал беру кезіндегі қолданысының мысалы көрсетілген.



9.5.4-сурет. *Rate Limiter* блогын колдану

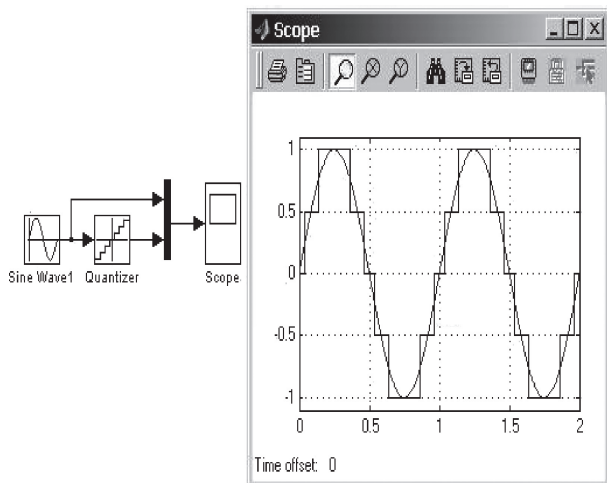
9.5.5 Quantizer деңгей бойынша кванттау блогы

Қызметі: Блок деңгейі бойынша бірдей адымдағы кіріс сигналының квантталануын қамтамасыз етеді.

Параметрлері:

Quantization interval – Деңгей бойынша кванттау адымы.

9.5.5-суретте Синусоидалық сигнал деңгейі бойынша кванттауды орындайтын *Quantizer* блогын қолданудың мысалы көрсетілген. Кванттау адымы 0.5 -ке тең деп берілген.



9.5.5-сурет. *Quantizer* блогын қолдану

9.5.6 Coulomb and Viscous Friction құрғақ және тұтқыр үйкеліс блогы

Қызметі: Құрғақ және тұтқыр үйкелістің эффектісін модельдейді.

Параметрлері:

Coulomb friction value (Offset) – құрғақ үйкелістің шамасы.

Coefficient of viscous friction (Gain) – тұтқыр үйкелістің коэффициенті төмендегідей сызықсыздық сипаттамасын көрсетеді:

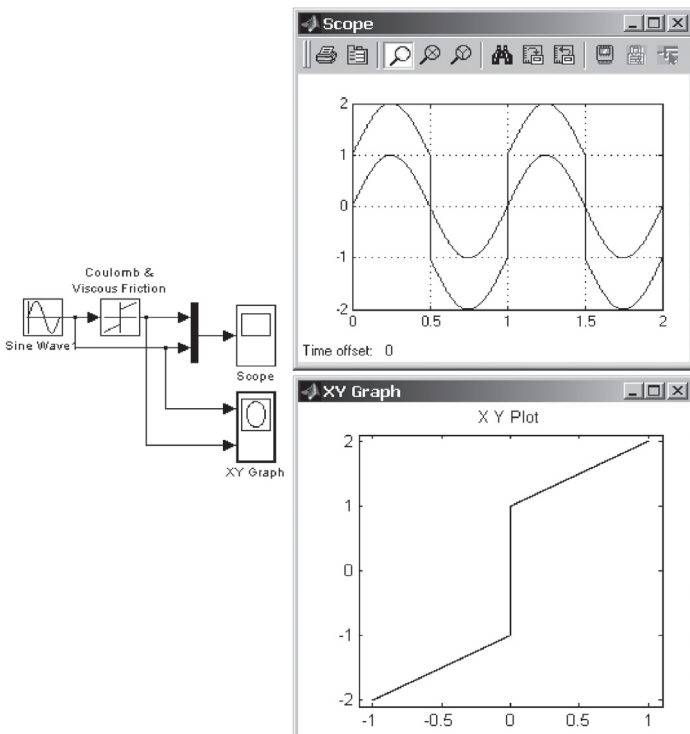
$$y = \text{sign}(u) * (\text{Gain} * \text{abs}(u) + \text{Offset}),$$

мұндағы u – кіріс сигналы, y – шығыс сигналы,

Gain – тұтқыр үйкелістің коэффициенті,

Offset – құрғақ үйкелістің шамасы.

9.5.6-суретте *Coulomb and Viscous Friction* блогын қолдану мысалы көрсетілген. Блоктың екі параметрі де 1-ге тең деп берілген.



9.5.6-сурет. *Coulomb and Viscous Friction* блогын қолдану

9.5.7 Backlash люфт блогы

Қызметі: “Люфт” типтегі сызықсыздықты модельдейді.

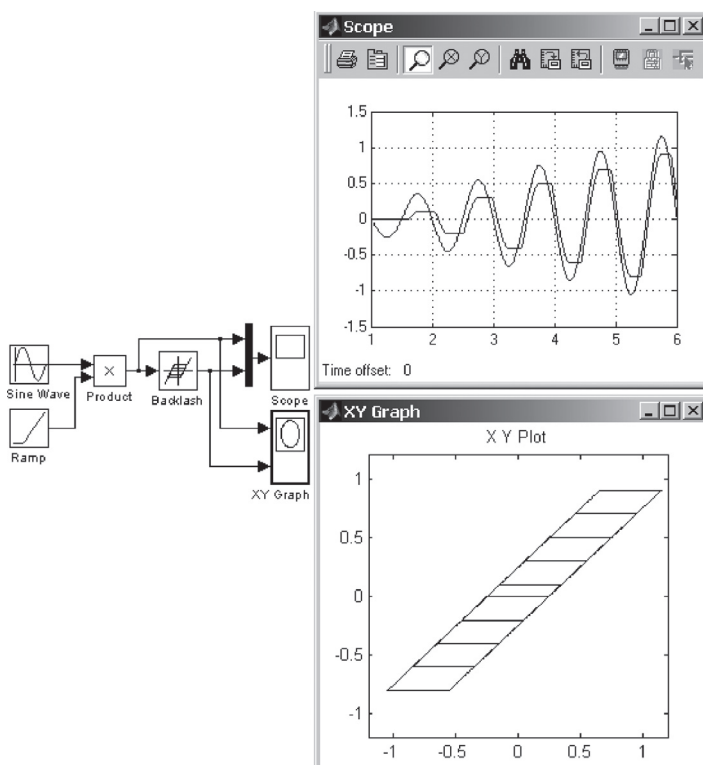
Параметрлері:

Deaband width – люфт ені.

Initial output – шығыс сигналының бастапқы мәні.

Шығыс сигналы кіріс сигналының өсуі (*Deaband width*)/2 мәнге жеткенше (мұндағы U – кіріс сигналы) *Initial output* арқылы берілген мәнге тең болады, бұдан кейін шығыс сигналы $U-(Deaband\ width)/2$ -ге тең болады. Кіріс сигналының өзгеру бағыты ауысқанда, ол өзгеріссіз болады, кіріс сигналының шамасы (*Deaband width*)/2 өзгергенше, сонан соң шығыс сигналы $U+(Deaband\ width)/2$ тең болады.

9.5.7-суретте *Backlash* блогының жұмыс істеу мысалы көрсетілген. Суреттен көрінгендей, блоктың кіріс сигналы сызықтық өсетін амплитудамен гармониялық түрде болады.



9.5.7-сурет. *Backlash* блогын қолдану

9.5.8 Switch өшіріп-қосқыш блогы

Қызметі: Басқару сигналы бойынша кіріс сигналының өшіріліп-қосылуын орындайды.

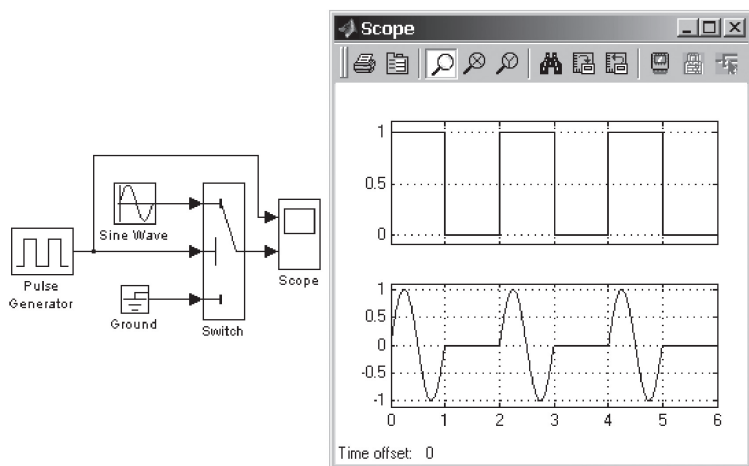
Параметрлері:

Threshold – басқару сигналының шегі.

Блок келесі түрде жұмыс істейді:

Егер ортаңғы кіріске берілетін басқару сигналы *Threshold* шектік шамасынан үлкен болса, онда блоктың бірінші (жоғарғы) кірісінен шығысына қарай сигнал өтеді. Егер басқару сигналы шектік мәннен аз болса, онда блоктың екінші (төменгі) кірісінен шығысына сигнал өтеді.

9.5.8-суретте *Switch* блогының жұмыс істеу мысалы көрсетілген. Егер кілттің басқарылатын кірісіндегі сигнал 1-ге тең болса, блок шығысына гармониялық сигнал өтеді, ал егер де басқару сигналы 0-ге тең болса, онда *Ground* блогынан шығысқа нөлдік деңгейдегі сигнал өтеді. Басқару сигналының шектік мәні 0.5-ке тең деп берілген.



9.5.8-сурет. *Switch* өшіріп-қосқышын қолдану

9.5.9 Multiport Switch көпкірісті өшіріп-қосқыш блогы

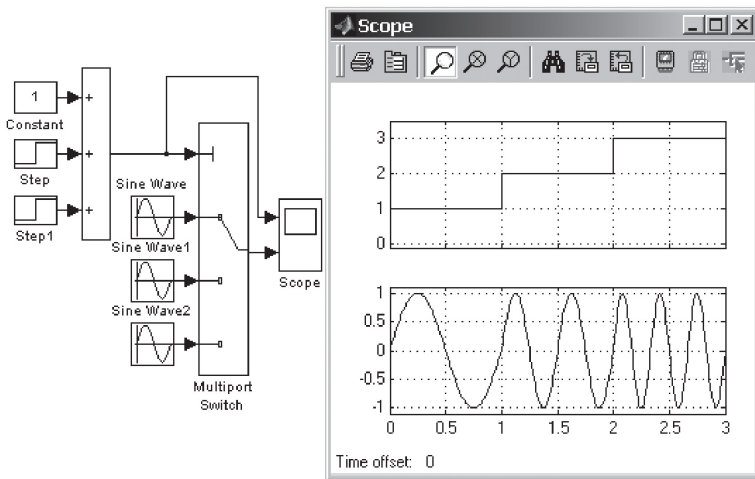
Қызметі: Кіріс портының белсенді нөмірін беретін басқару сигналы бойынша кіріс сигналының өшіріп-қосылуын орындайды.

Параметрлері:

Number of inputs – кіріс саны.

Multiport Switch көпкірісті өшіріп-қосу блогы басқару сигналының ағымдағы мәнінің нөміріне тең болатын кіріс портынан шығысқа сигнал жібереді. Егер басқару сигналы бүтін сигнал түрінде болмаса, онда *Multiport Switch* блогы санның бөлшегін алып тастайды да, MATLAB-тың командалық терезесінде ескертпе жазу пайда болады.

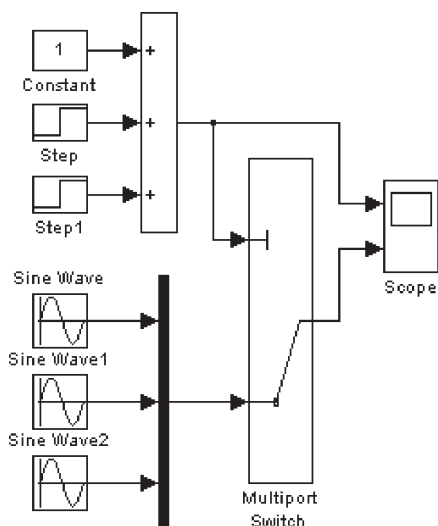
9.5.9-суретте *Multiport Switch* блогының жұмыс істеу мысалы көрсетілген. Басқару сигналын өшіріп-қосудың *Constant*, *Step*, *Step1* және *Sum* блоктары арқылы қалыптасатын үш деңгейі бар. *Multiport Switch* блогының шығысына кіріс сигналының деңгейінен тәуелді болатын әртүрлі жиіліктегі гармониялық сигналдар өтеді.



9.5.9-сурет. *Multiport Switch* өшіріп-қосқышын қолдану

Multiport Switch блогының кіріс санын 1-ге тең деп беруге болады. Бұл жағдайда блоктың кірісіне векторлық сигнал беру қажет, блоктың өзі шығысына сол вектордың нөмірі басқару сигналына сай келетін элементін өткізеді.

9.5.10-суретте векторлық сигналда *Multiport Switch* блогын қолдану мысалы көрсетілген. Жұмыстың уақыттық диаграммалары алдыңғы келтірілген мысалға сай келеді.



9.5.10-сурет. Векторлық кіріс сигналында *Multiport Switch* өшіріп-қосқышын қолдану

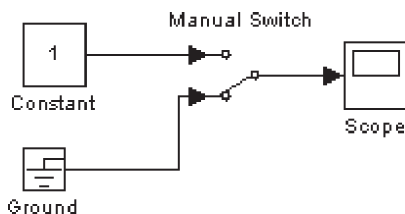
9.5.10 Manual Switch қолмен өшіріп-қосқыш блогы

Қызметі: Қолданушының бұйрығымен кіріс сигналының өшіріп-қосылуын орындайды.

Параметрлері:

Блок суретінде тышқанның сол жақ батырмасын екі рет басу өшіріп-қосу командасы болып табылады. Бұл кезде блок суреті өзгереді, дәл осы уақытта қандай кіріс сигналы блоктың шығысына өтуін көрсетеді. Блоктың өшіріп-қосуын модельдеуді бастамас алдында жасауға болады және есептеу үдерісінде де болады.

9.5.11-суретте *Manual Switch* блогын қолдану мысалы көрсетілген.



9.5.11-сурет. *Manual Switch* блогын қолдану

9.6 Math – математикалық амалдар блогы

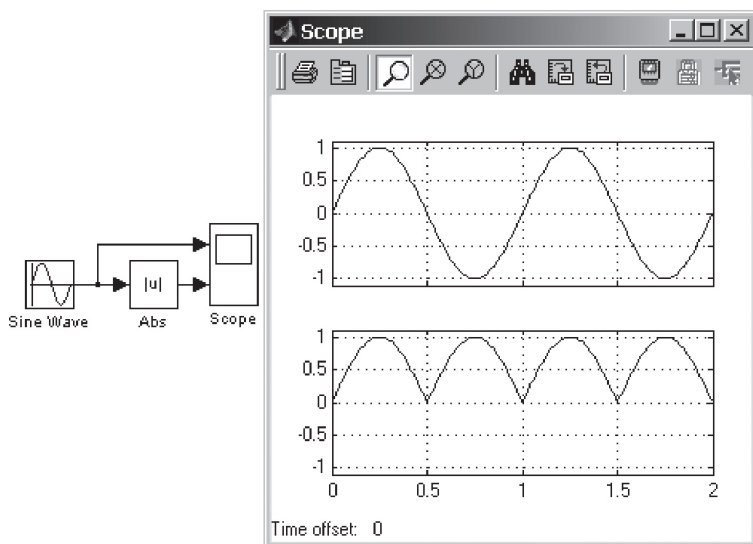
9.6.1 Abs модуль есептеу блогы

Қызметі: Сигнал шамасының абсолюттік мәнін есептеуді орындайды.

Параметрлері:

Saturate on integer overflow (жалауша) – Бүтіннің артып кетуін басу. Жалауша тұрған жағдайда бүтін түрдегі сигналдарды шектеу дұрыс орындалады.

9.6.1-суретте синусоидалық сигналдың ағымдағы мәнінің модулін есептеу үшін *Abs* блогын қолдану мысалы көрсетілген.

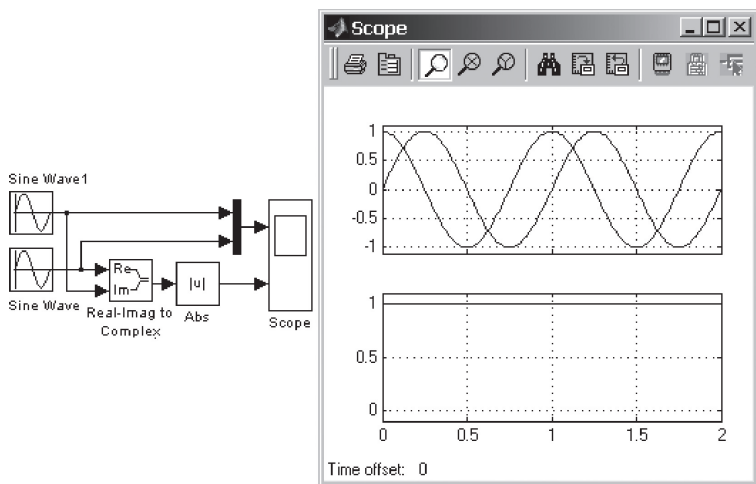


9.6.1-сурет. *Abs* блогын қолдану

Abs блогы комплекстік типтегі сигналдың модулін есептеу үшін қолданылады. 9.6.2-суретте комплекстік типтегі сигнал модулін есептеу мысалы көрсетілген:

$$u = \cos(\omega t) + i \sin(\omega t).$$

Кез келген уақыт үшін бұл модульдің сигналы 1-ге тең.



9.6.2-сурет. Комплекстік сигналдың модулін есептеу үшін *Abs* блогын қолдану

9.6.2 Sum қосындыны есептеу блогы

Қызметі: Сигналдардың ағымдағы мәндерінің қосындысын есептейді.

Параметрлері:

Icon shape – блоктың формасы. Ол келесідей тізімнен таңдап алынады:

- *round* – шеңбер,
- *rectangular* – тікбұрыш

List of sign – белгілер тізімі. Бұл тізімде келесі белгілерді қолдануға болады: + (қосу), – (азайту) және | (ажырату белгісі).

Saturate on integer overflow (жалауша) – бүтіннің артып кетуін басу. Жалауша тұрған жағдайда бүтін түрдегі сигналдарды шектеу дұрыс орындалады.

Кіріс саны мен амал (қосу немесе азайту) *List of sign* белгілер параметрінің тізімі арқылы анықталады, сонымен қатар, кіріс таңбасы тиісті белгілермен белгіленеді. *List of sign* параметрінде кіріс блогының санын көрсетуге болады. Бұл жағдайда барлық кіріс жиынтықтаушы болады.

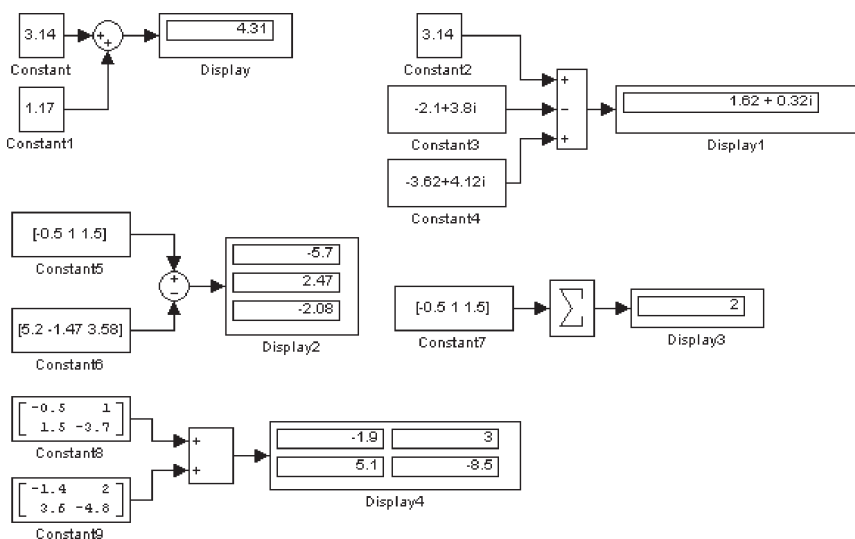
Егер кіріс блогының саны 3-тен асса, онда тікбұрыш пішінді *Sum* блогын қолданған ыңғайлы.

Блок скалярлық, векорлық немесе матрицалық сигналдардың қосындыларын орындауда қолданылады. Қосылатын сигналдардың типтері ұқсас болу керек. Мысалы, бір қосушы блокқа бүтін және нақты типтегі сигналдарды беруге болмайды.

Егер блоктың кіріс саны l -ден көп болса, онда блок векторлық және матрицалық сигналдардың элемент бойынша амалдарын орындайды. Осыған сәйкес, матрицадағы немесе вектордағы элементтер саны бірдей болу керек.

Егер белгілер тізімі ретінде l (бір кіріс) санын көрсетсе, онда блокты вектор элементтерінің қосындыларын анықтауда қолдануға болады.

9.6.3-суретте *Sum* блогын қолданудың мысалдары көрсетілген.



9.6.3-сурет. *Sum* блогын қолдану

9.6.3 Product көбейту блогы

Қызметі: Сигналдардың ағымдағы мәндерінің көбейтінділерін есептейді.

Параметрлері:

Number of inputs – кіріс саны. Сан немесе белгілер тізімі ретінде берілуі мүмкін. Белгілер тізімінде «*» (көбейту) және «/» (бөлу) белгілерін қолдануға болады.

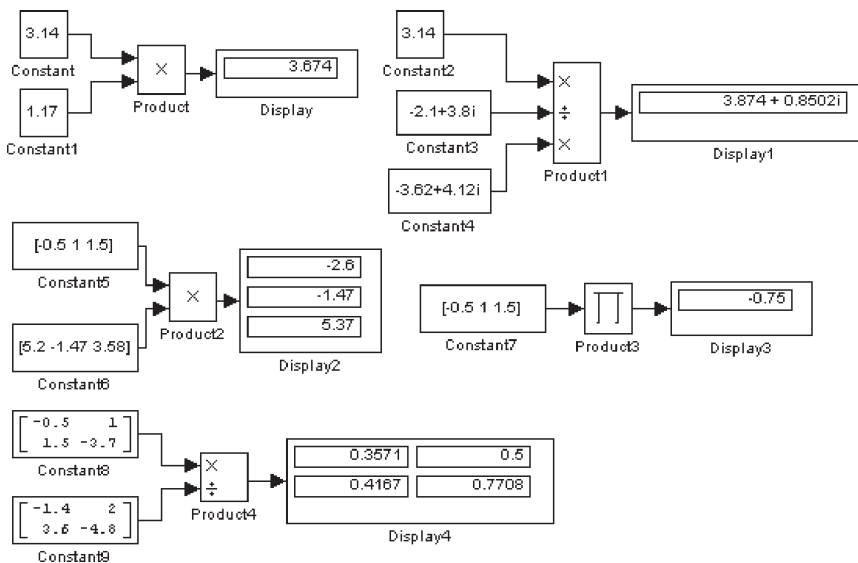
Multiplication – амалдарды орындау әдісі. Ол келесі тізімнен таңдап алынады:

- *Element-wise* – элемент бойынша.
- *Matrix* – матрицалық.
- *Saturate on integer overflow (жалауша)* – бүтіннің артып кетуін басу. Жалауша тұрған жағдайда бүтін түрдегі сигналдарды шектеу дұрыс орындалады.

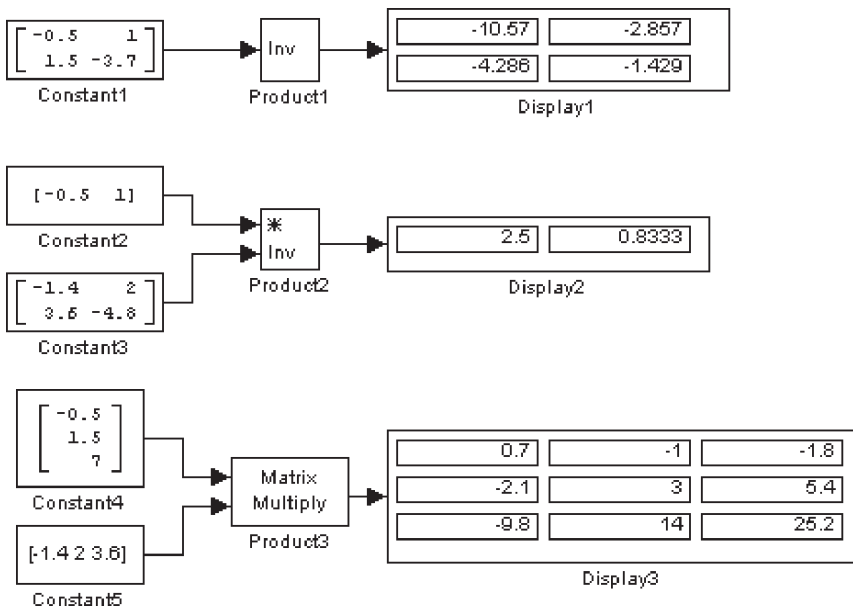
Егер *Number of inputs* параметрі тізім бойынша көбейту және бөлу белгілерінен басқа белгілер арқылы берілсе, онда кіріс таңбасы сәйкес амал таңбаларымен белгіленеді.

Блок скалярлық, векторлық немесе матрицалық сигналдардың көбейту және бөлу амалдарына қолданылады. Блоктың кіріс сигналдарының типтері бір-біріне сәйкес келуі керек. Егер кіріс санын *1* деп (бір кіріс) көрсетсек, онда блокты вектор элементтерінің көбейтінділерін анықтау үшін қолдануға болады.

9.6.4-суретте скалярлық және элемент бойынша амалдарды орындау үшін *Product* блогының қолданылу мысалы көрсетілген.



9.6.4-сурет. Скалярлық және элемент бойынша амалдарды орындау үшін *Product* блогының қолданылуы



9.6.5-сурет. Матрицалық амалдарды орындауда *Product* блогының қолданылуы

Матрицалық амалдарды орындау барысында сақтау қажет болатын кейбір ережелер бар. Мысалы, екі матрицаны көбейту барысында бірінші матрицаның жол санымен екінші матрицаның баған саны бір-біріне тең болулары тиіс. 9.6.5-суретте матрицалық амалдарды орындауда *Product* блогын қолданудың мысалы көрсетілген. Бұл мысалда кері матрицаны қалыптастыру, матрицаларды бөлу және матрицаларды көбейту амалдары көрсетілген.

9.6.4 Sign сигнал таңбасын анықтау блогы

Қызметі: Кіріс сигналының таңбасын анықтайды.

Параметрлері: Жоқ.

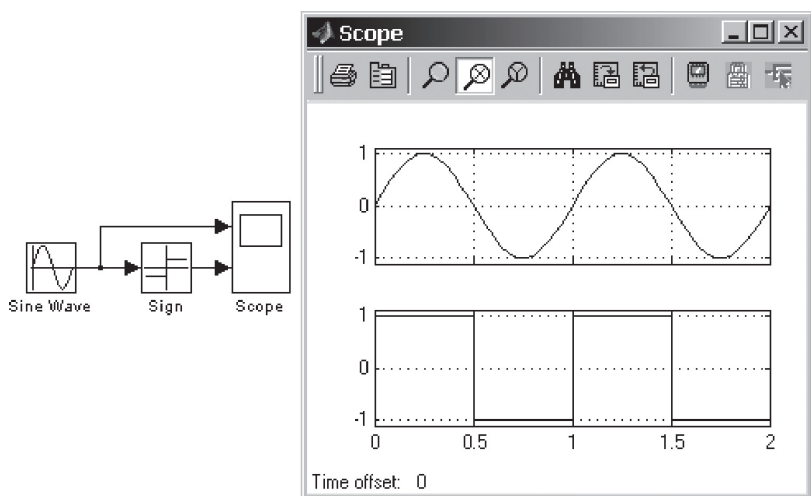
Блок келесі алгоритмге сәйкес жұмыс істейді:

- Блоктың кіріс сигналы оң болса, онда сол шығыс сигнал 1 тең.

- Блоктың кіріс сигналы теріс болса, онда шығыс сигнал -1 тең.

- Егер блоктың кіріс сигналы 0 тең болса, онда шығыс сигналы да 0 тең.

9.6.6-сурет *Sign* блогының жұмысын көрсетеді.



9.6.6-сурет. *Sign* блогын қолдану

9.6.5 Gain және Matrix Gain күшейткіштері

Қызметі: Кіріс сигналын тұрақты коэффициентке көбейтеді.

Параметрлері:

Gain – күшейткіш коэффициенті.

Multiplication – амал орындау әдісі. Ол келесідей мәндерді қабылдайды (тізімнен):

- *Element-wise $K*u$* – элемент бойынша.

- *Matrix $K*u$* – матрицалық. Күшейту коэффициенті сол жақты операнд болып табылады.

- *Matrix $u*K$* – матрицалық. Күшейту коэффициенті оң жақты операнд болып табылады.

- *Saturate on integer overflow (жалауша)* – бүтіннің артып кетуін басу. Жалауша тұрған жағдайда бүтін түрдегі сигналдарды шектеу дұрыс орындалады.

Gain және *Matrix Gain* күшейткіш блоктары негізінен бірдей, олардың айырмашылықтары тек *Multiplication* параметрлерінің алғышарттарының әр түрлі қойылуларында ғана болады.

Gain блогының параметрлері 1-ден үлкен де, кіші де болатын оң және теріс сандар болуы мүмкін. Күшейткіш коэффициентін скаляр, матрица немесе вектор түрінде беруге болады.

Егер де *Multiplication* параметрі *Element-wise $K*u$* түрінде берілген болса, онда блок скалярлық сигналдың берілген коэффициентіне немесе векторлық сигналдың әр элементіне

көбейту амалын орындайды. Ал кері жағдайда, блок берілген сигнал мен матрица арқылы берілген коэффициент арасындағы матрицалық көбейту амалын орындайды.

Күшейткіш коэффициенті, жалпы алғанда, үнсіздікпен *double* типті нақты сан болып табылады.

Элементтер бойынша жүргізілетін күшейту амалдары үшін кіріс сигналы логикалық (*boolean*) типтен басқа кез келген типтегі скалярлық, векторлық немесе матрицалық бола береді. Вектордың элементтері бір типті сигнал болуы тиіс. Блоктың кірісіндегі сигналдың типі қандай болса, оның шығысында да сол типтегі сигнал болады. *Gain* блогының параметрі де логикалық (*boolean*) типтен басқа кез келген типтегі скалярлық, векторлық немесе матрицалық болады.

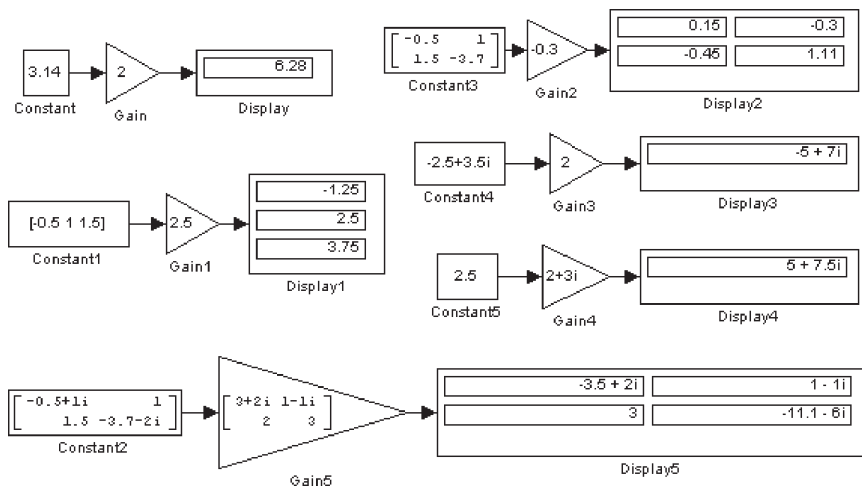
Gain блогының шығыс сигналын есептеуде келесідей ұстанатын шарттары бар:

- Егер кіріс сигнал нақты типте, ал күшейткіш коэффициенті комплекстік болса, онда шығыс сигналы комплекстік болады.

- Егер кіріс сигналының типі күшейткіш коэффициентінің типінен басқа болса, онда *Simulink* күшейткіш коэффициентін типін кіріс сигналының типіне келтіруге тырысады.

Мысалы, кіріс сигналының таңбасы сигналсыз (*uint8*), ал *Gain*-нің параметрі ретінде теріс сан берілетіндей тосын жағдайлар да туындауы мүмкін.

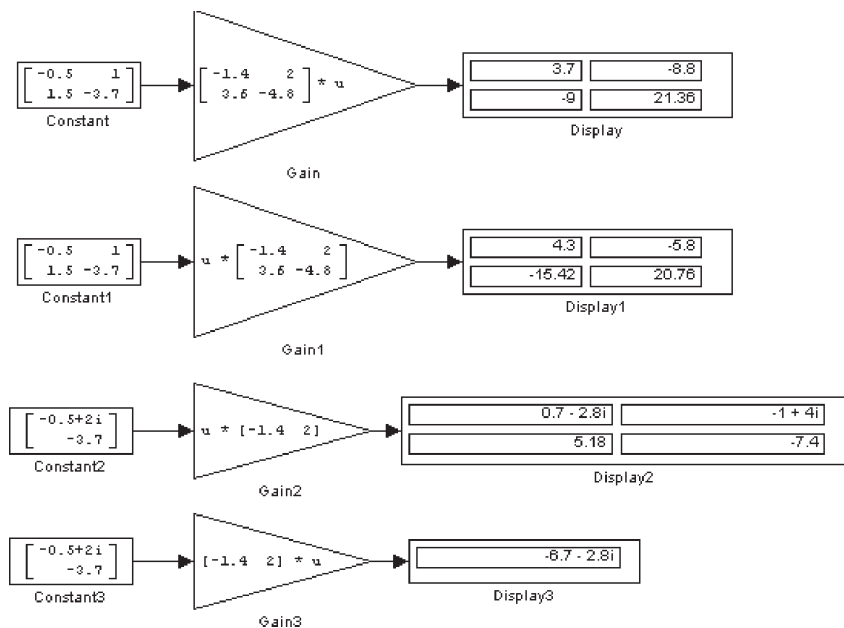
9.6.7-суретте скалярлық және элементтік амалдарды орындауда *Gain* блогын қолданудың мысалдары көрсетілген.



9.6.7-сурет. *Gain* блогын қолдану

Матрицалық күшейту амалы (берілген коэффициентті кіріс сигналына матрицалық көбейту) үшін кіріс сигналы мен күшейткіш коэффициенті *single* немесе *double* типтегі комплекстік немесе нақты скалярлық, векторлық немесе матрицалық мәндер болу керек.

9.6.8-суретте матрицалық амалдарды орындауда *Matrix Gain* блогының қолданылу мысалдары көрсетілген.



9.6.8-сурет. *Matrix Gain* блогын қолдану

9.6.6 Slider Gain жылжымалы реттегіш

Қызметі: Есептеу кезіндегі күшейткіш коэффициентінің өзгеруін қамтамасыз етеді.

Параметрлері:

Low – күшейткіш коэффициентінің төменгі шегі.

High – күшейткіш коэффициентінің жоғарғы шегі.

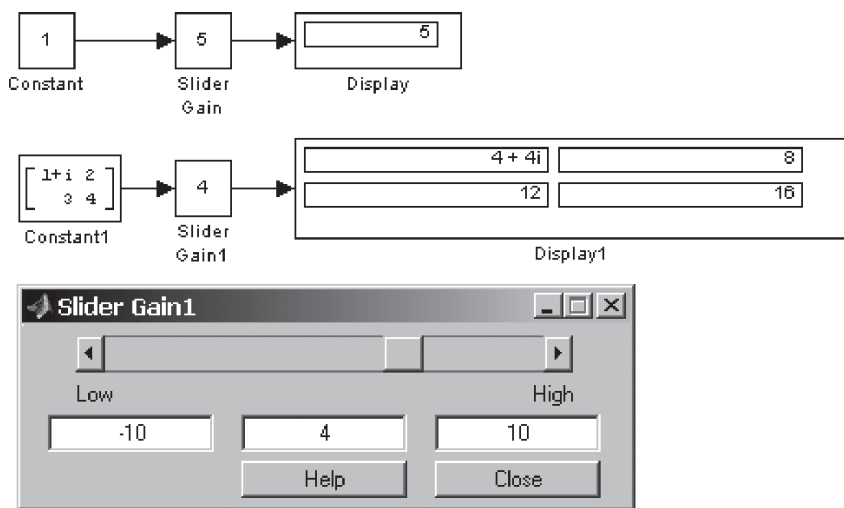
Slider Gain+ блогының күшейткіш коэффициентін өзгерту үшін реттегіштің жылжытпасын орнынан қозғау керек. Жылжытпаның оң жаққа қозғалысы күшейткіш коэффициентінің артуына, ал сол жаққа қозғалысы – керісінше, кемуіне алып келеді. Күшейткіш

коэффициентінің өзгеруі берілген *Low* және *High* параметрлерінің аралығында орындалады.

Егер реттегіштің шкаласының оң немесе сол жақ бағыттауышында тышқан батырмасының көмегімен шертсе күшейткіш коэффициенті орнатылған мөлшерден 1%-ға өзгереді. Ал егер реттегіш шкаласының өзінде жылжытпаның оң не сол жағынан шертсе, онда күшейткіш коэффициенті орнатылған мөлшерден 10%-ға өзгереді. Сондай-ақ блоктың ортаңғы терезесінде коэффициенттің керекті мәнін беруге болады.

Блок, сонымен қатар, векторлық немесе матрицалық сигналдарға элементтік күшейту амалын да жүргізе алады. Кіріс сигналы комплекстік те болуы мүмкін.

9.6.9-суретте *Slider Gain* блогының қолданылу мысалдары көрсетілген.



9.6.9-сурет. *Slider Gain* блогын қолдану

9.6.7 Dot Product скалярлық көбейту блогы

Қызметі: Екі вектордың скалярлық көбейтіндісін орындайды.

Параметрлері: Жоқ.

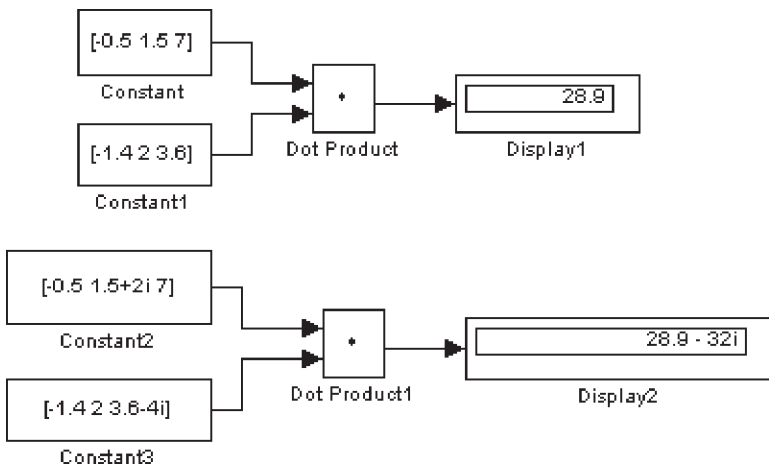
Блок

$$y = \text{sum}(\text{conj}(u1)) .* u2,$$

өрегіне сәйкес шығыс сигналының есептеуін орындайды, мұндағы $u1$ және $u2$ – кіріс векторлары, $conj$ – комплекстік біріккен санын есептеу амалы, sum – қосындыны есептеу амалы.

Егер кіріс векторларының екеуі де нақты болса, онда шығыс сигналы нақты болады. Тиісінше, егер кіріс сигналдарының біреуі комплекстік сигнал болса, онда шығыс сигналы да комплекстік болады.

9.6.10-суретте *Dot Product* блогының жұмысы көрсетілген.



9.6.10-сурет. *Dot Product* блогының жұмысы

9.6.8 Math Function математикалық функцияларды есептеу блогы

Қызметі: Математикалық функциялардың есептеулерін жүргізеді.

Параметрлері:

Function – есептелетін функцияның түрі. Олар келесі тізімнен таңдап алынады:

- exp – экспоненциалдық функция,
- log – натурал логарифмдік функциясы,
- 10^u – 10 -ның дәрежесін есептеу,
- $log10$ – логарифмдік функция,
- $magnitude^2$ – кіріс сигналының модулінің квадратын есептеу,

теу,

- *square* – кіріс сигналының квадратын есептеу,
- *sqrt* – квадрат түбірі,
- *pow* – дәрежелуе,
- *conj* – комплекстік кездесетін санды есептеу,
- *reciprocal* – кіріс сигналын 1-ге бөлуден бөлінді есептеу,
- *hypot* – кіріс сигналдарының қосындыларының квадраттарынан квадрат түбір алу (тік бұрышты үшбұрыштың гипотенузасы мен катеттерінің мәндері),
- *rem* – бірінші кіріс сигналын екіншіге бөлгеннен қалатын қалдықтарды есептеу функциясы,
- *mod* – бөлгеннен қалған қалдықтарды таңбалары бойынша есептеу функциясы,
- *transpose* – матрицалардың тасымалдауы,
- *hermitian* – эрмиттік матрицаны есептеу.

Output signal type – шығыс сигналының типі (тізімнен таңдап алынады):

Auto – типті автоматты түрде анықтау.

Real – нақты сигнал.

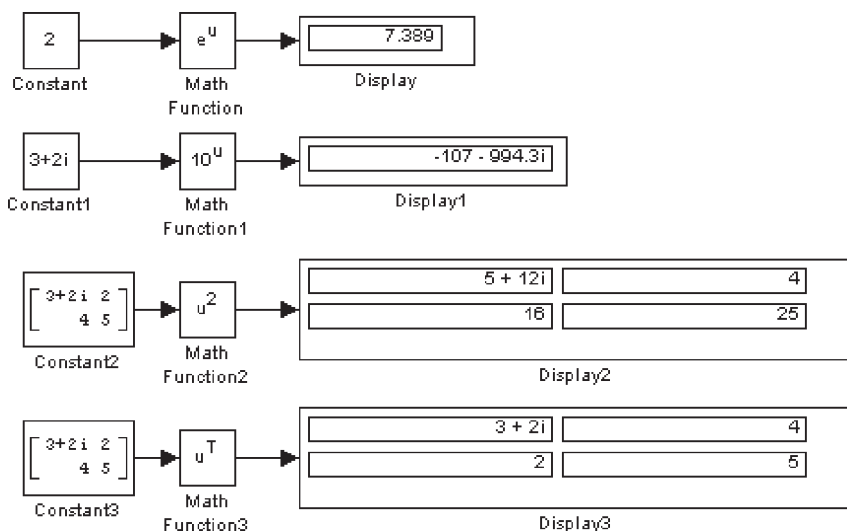
Complex - комплекстік сигнал.

Шығыс сигналының типі кіріс сигналының, *Output signal type* блогының параметрінің және есептелетін функцияның типтеріне байланыстылығы 9.6.1-кестеде келтірілген.

9.6.1-кесте.

Функция	Кіріс сигналы	Шығыс сигналы		
		Auto	Real	Complex
<i>exp, log, 10ⁿ, log10, square, sqrt, pow, reciprocal, conjugate, transpose, hermitian</i>	<i>real</i> <i>complex</i>	<i>real</i> <i>complex</i>	<i>real</i> <i>error</i>	<i>complex</i> <i>complex</i>
<i>magnitude squared</i>	<i>real</i> <i>complex</i>	<i>real</i> <i>real</i>	<i>real</i> <i>real</i>	<i>complex</i> <i>complex</i>
<i>hypot, rem, mod</i>	<i>real</i> <i>complex</i>	<i>real</i> <i>error</i>	<i>real</i> <i>error</i>	<i>complex</i> <i>error</i>

9.6.11-суретте *Math Function* блогын қолданудың мысалдары көрсетілген.



9.6.11-сурет. Math Function блогын қолдану

9.6.9 Trigonometric Function тригонометриялық функцияларды есептеу блогы

Қызметі: Тригонометриялық функциялардың есептеулерін жүргізеді.

Параметрлері:

Function – есептелетін функция түрі (тізімнен тандап алынады): \sin , \cos , \tan , asin , acos , atan , atan2 , \sinh , \cosh және \tanh .

Output signal type – шығыс сигналының типі (тізімнен тандап алынады):

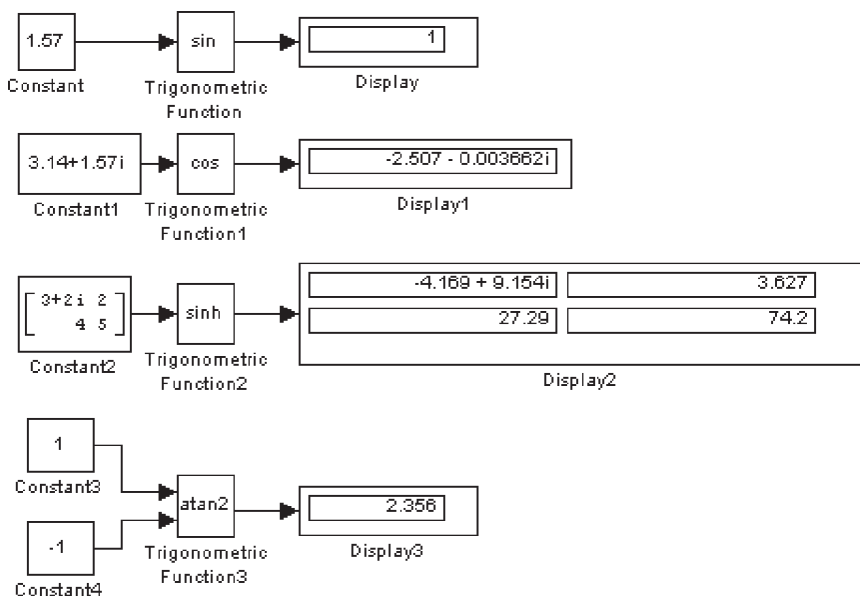
Auto – типті автоматты түрде анықтау.

Real – нақты сигнал.

Complex - комплекстік сигнал.

Векторлық немесе матрицалық кіріс сигналында блок берілген функцияның есептеулерін элементтері бойынша жүргізеді.

9.6.12-суретте *Trigonometric Function* блогының қолданылу мысалы көрсетілген.



9.6.12-сурет. *Trigonometric Function* блогының қолданылуы

9.6.10 Complex to Real-Imag комплекстік санның нақты және жорамал бөлігін есептеу блогы

Қызметі: Комплекстік санның нақты және жорамал бөлігін есептейді.

Параметрлері:

Output – Шығыс сигналы (тізімнен таңдап алынады):

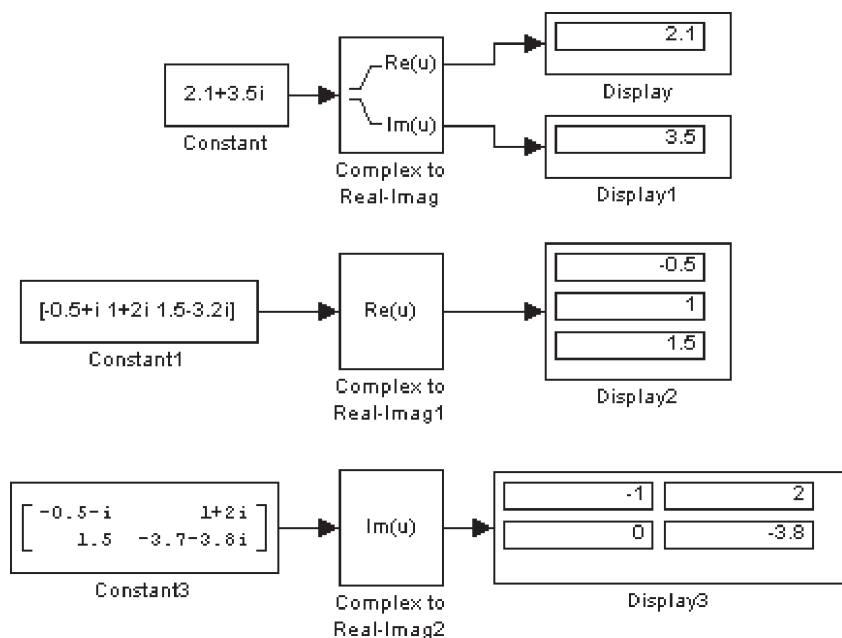
Real – нақты бөлімі.

Image - жорамал бөлімі.

RealAndImage – нақты және жорамал бөлімдері.

Блоктың кіріс сигналы скалярлық, векторлық немесе матрицалық болуы мүмкін.

9.6.13-суретте *Complex to Real-Imag* блогының қолданылу мысалдары көрсетілген.



9.6.13-сурет. *Complex to Real-Imag* блогының қолданылуы

9.6.11 Complex to Magnitude-Angle комплекстік санның аргументін және (немесе) модулін есептейтін блок

Қызметі: Комплекстік санның аргументін және (немесе) модулін есептейді.

Параметрлері:

Output – шығыс сигналы (тізімнен тандап алынады):

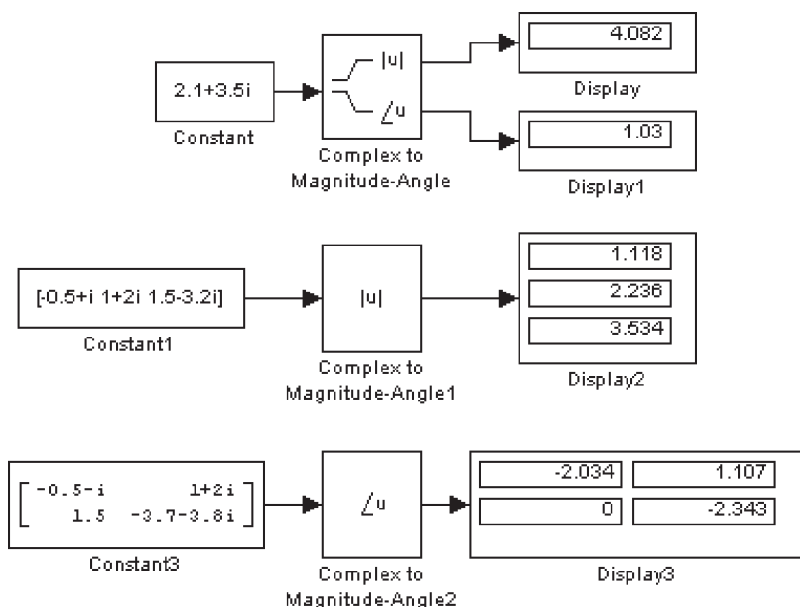
Magnitude – модуль.

Angle – аргумент.

MagnitudeAndAngle – модуль және аргумент.

Блоктың сигналы скалярлық, векторлық немесе матрицалық болуы мүмкін.

9.6.14-суретте *Complex to Magnitude-Angle* блогының қолданылу мысалдары көрсетілген.



9.6.14-сурет. *Complex to Magnitude-Angle* блогының қолданылуы

9.6.12 Real-Imag to Complex комплекстік санды оның нақты және жорамал бөлігі бойынша есептеу блогы

Қызметі: Комплекстік санды оның нақты және жорамал бөлігі бойынша есептеулер жүргізеді.

Параметрлері:

Input – кіріс сигналы (тізімнен тандап алынады):

Real – нақты бөлімі.

Image – жорамал бөлімі.

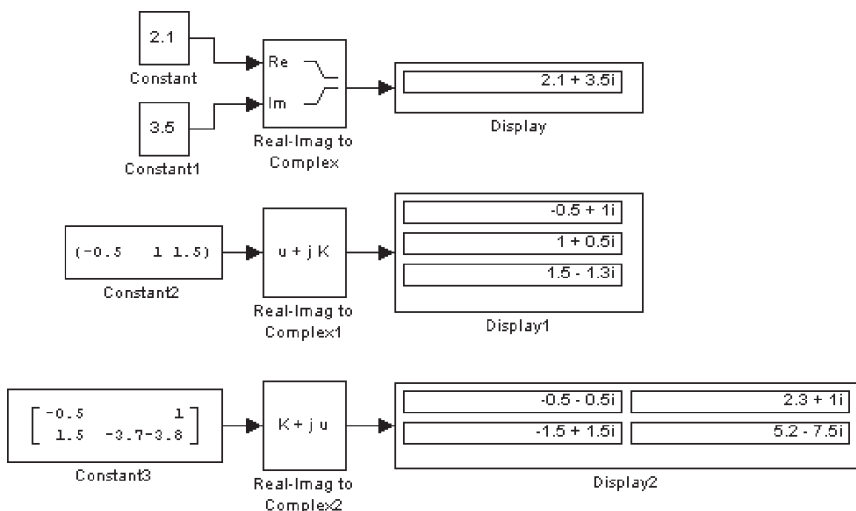
RealAndImage – нақты және жорамал бөлімдері.

Image part – жорамал бөлімі. *Input* параметрі *Real* деп жарияланса, бұл параметр қолданысқа ие болады.

Real part – нақты бөлімі. *Input* параметрі *Image* деп жарияланса, бұл параметр қолданысқа ие болады.

Блоктың кіріс сигналдары скалярлық, векторлық немесе матрицалық бола алады. Егер кіріс сигналы векторлық немесе матрицалық болса, онда *Image part* немесе *Real part* параметрлері векторлық немесе матрицалық болып берілуі тиіс.

9.6.15-суретте *Real-Imag to Complex* блогының қолданылу мысалдары көрсетілген.



9.6.15-сурет. *Real-Imag to Complex* блогының қолданылуы

9.6.13 Magnitude-Angle to Complex комплекстік санды оның модулі мен аргументі бойынша есептеу блогы

Қызметі: Комплекстік санды оның модулі мен аргументі бойынша есептеулер жүргізеді.

Параметрлері:

Input – кіріс сигналы (тізімнен тандап алынады):

Magnitude – модуль.

Angle – аргумент.

MagnitudeAndAngle – модуль және аргумент.

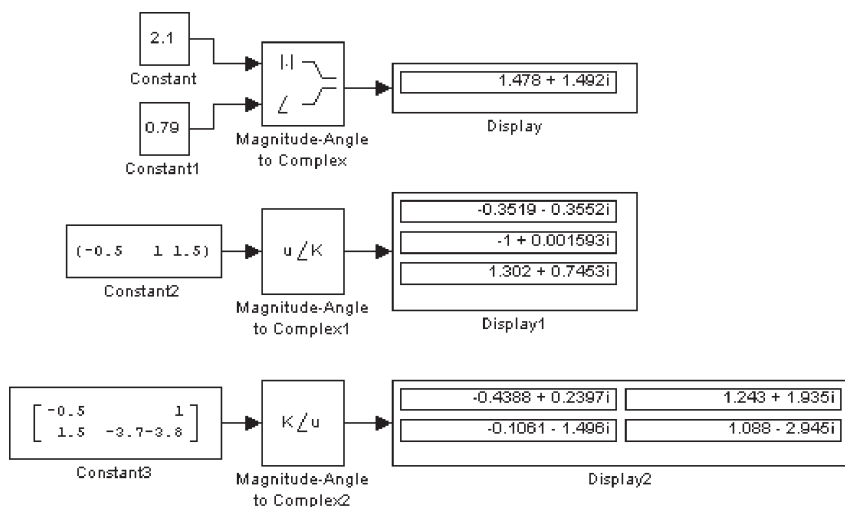
Angle – аргумент. *Input* параметрі *Magnitude* деп жарияланса, бұл параметр қолданысқа ие болады.

Magnitude – модуль. *Input* параметрі *Angle* деп жарияланса, бұл параметр қолданысқа ие болады.

Блоктың кіріс сигналдары скалярлық, векторлық немесе матрицалық болады.

Кіріс сигналы векторлық немесе матрицалық болса, *Angle* немесе *Magnitude* параметрлері векторлық немесе матрицалық болып берілуі тиіс.

9.6.16-суретте *Magnitude-Angle to Complex* блогын қолданудың мысалдары көрсетілген.



9.6.16-сурет. *Magnitude-Angle to Complex* блогының қолданылуы

9.6.14 MinMax минималдық немесе максималдық мәнді анықтайтын блок

Қызметі: Кіріске келіп түсетін сигналдардан максималдық немесе минималдық мәнін анықтайды.

Параметрлері:

Function – шығыс параметрі. Ол тізімнен таңдап алынады:

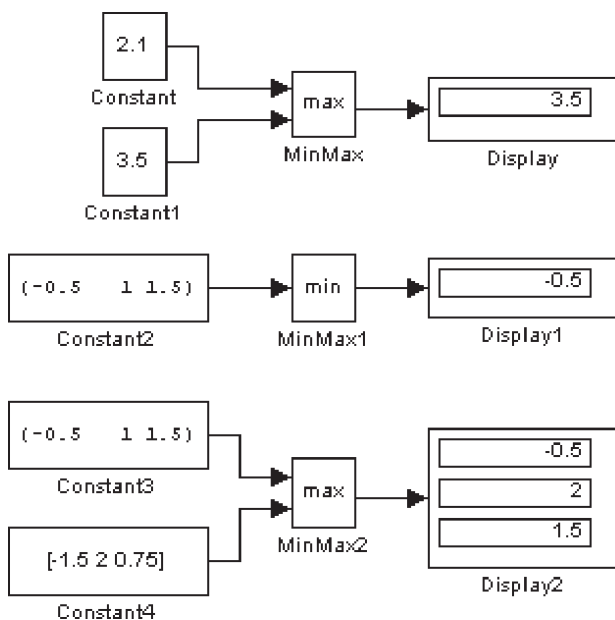
min – минималдық мән.

max – максималдық мән.

Number of input ports – кіріс порттарының саны.

Блоктың кіріс сигналдары скалярлық немесе векторлық болуы мүмкін. Блок кірісіне келіп түсетін барлық скалярлық сигналдардың ішінен максималдық немесе минималдық мәнін анықтайды. Егер кіріс сигналдары векторлық болса, блок минималдық немесе максималдық мәнді іздеу үшін элементтік амалды қолданады. Бұл жағдайда векторлардың мөлшерлері сәйкес келуі тиіс. Егер блоктың кіріс порттарының саны *l*-ге тең деп берілетін болса, онда блок минималдық немесе максималдық мәнін кіріс векторы арқылы анықтайды.

9.6.17-суретте *MinMax* блогының қолданылу мысалдары көрсетілген.



9.6.17-сурет. *MinMax* блогының қолданылуы

9.6.15 Rounding Function сандық мәнді дөңгелектеу блогы

Қызметі: Сандық мәнді дөңгелектеудің амалын орындайды.

Параметрлері:

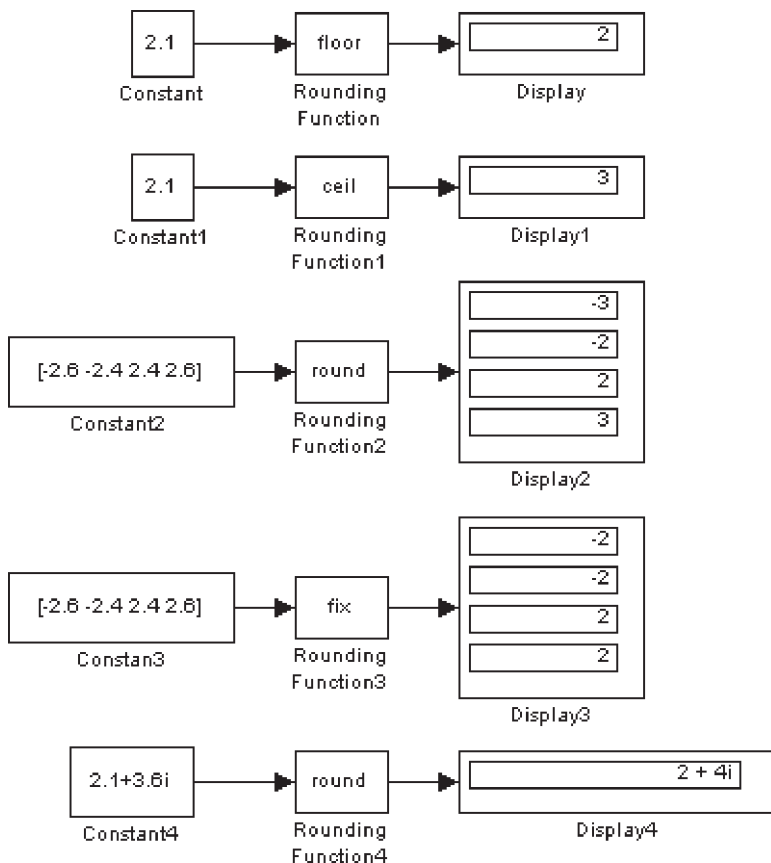
Function – дөңгелектеу әдісі (тізімнен тандап алынады):

floor – жақын тұрған төменгі бүтінге дейін дөңгелектеу,

ceil – жақын тұрған жоғарғы бүтінге дейін дөңгелектеу,

round – жақын тұрған бүтінге дейін дөңгелектеу,

fix – бөлшек бөлімін алып тастау арқылы дөңгелектеу.



9.6.18-сурет. *Rounding Function* блогының қолданылуы

Блоктың кіріс сигналдары нақты немесе комплекстік типтегі скалярлық, векторлық немесе матрицалық сигналдар болуы мүмкін. Векторлық немесе матрицалық кіріс сигналдарында блок элементтік амалдарды орындайды.

Блоктың шығыс сигналдары *double* немесе *single* типтерін қолдана алады.

9.6.18-суретте *Rounding Function* блогын қолданудың мысалдары көрсетілген.

9.6.16 Relational Operator қатынас амалдарын есептеу блогы

Қызметі: Блок кіріс сигналдарының ағымдағы мәндерін салыстырады.

Параметрлері:

Relational Operator – қатынас амалдарының типі (тізімнен таңдап алынады):

$=$ – тең.

\sim – тең емес.

$<$ – азырақ.

$<=$ – азырақ немесе тең.

$>=$ – көбірек немесе тең.

$>$ – көбірек.

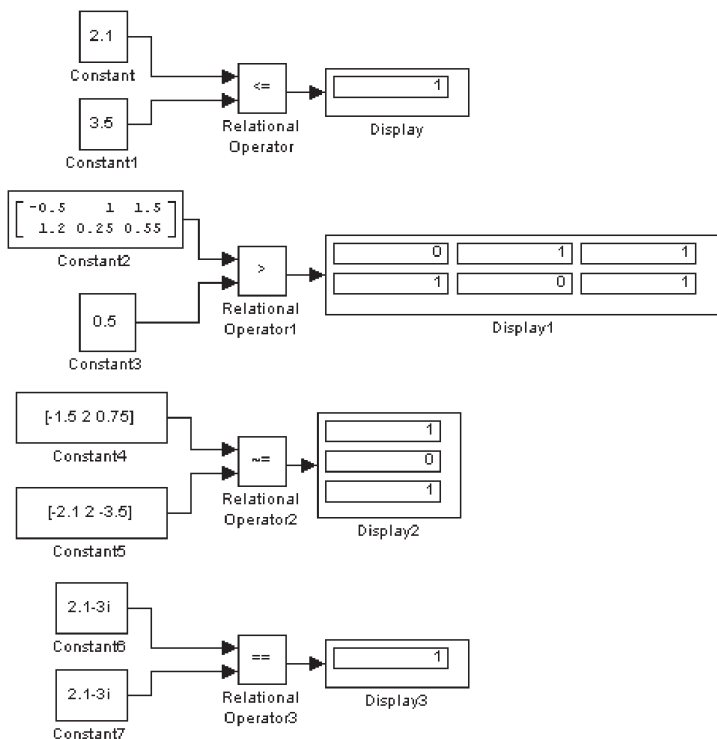
Амалдық қатынастарда бірінші орындалатын амал ретінде блоктың бірінші (жоғарғы) кірісіне берілетін сигнал саналады да, ал орындалатын екінші амал ретінде блоктың екінші (төменгі) кірісіне берілетін сигнал саналады. Егер қатынас амалдарының нәтижесі «ақиқат» болса, онда блоктың шығыс сигналы 1 болады да, егер «жалған» болса, блоктың шығыс сигналы 0 болады.

Блоктың кіріс сигналдары скалярлық, векторлық немесе матрицалық бола алады. Егер екі кіріс сигналы да векторлық немесе матрицалық болса, онда блок элементтер бойынша салыстыру амалын орындайды. Бұл жағдайда кіріс сигналдарының өлшемі бір-біріне сәйкес келуі тиіс. Егер кіріс сигналдарының біреуі вектор немесе матрица болса, ал екіншісі - скаляр болса, онда блок скалярлық кіріс сигналын жиымның әрбір элементімен салыстыру амалын орындайды. Шығыс сигналының өлшемі кіріске берілетін векторлық немесе матрицалық сигналдардың өлшемі бойынша анықталады.

Ал $=$ (*тең*) және \sim (*тең емес*) амалдары үшін комплекстік кіріс сигналдарын да қолдануға болады.

Сонымен қатар, кіріс сигналдары логикалық типте де болуы мүмкін (*boolean*).

9.6.19-суретте *Relational Operator* блогының қолданылу мысалдары көрсетілген.



9.6.19-сурет. Relational Operator блогының қолданылуы

9.6.17 Logical Operation логикалық амалдар блогы

Қызметі: Базалық логикалық амалдардың бірін орындайды.

Параметрлері:

Operator – логикалық амалдардың көрінісі (тізімнен таңдап алынады):

AND – логикалық көбейту (*ЖӘНЕ* амалы).

OR – логикалық қойылу (*НЕМЕСЕ* амалы).

NAND – *ЖӘНЕ-ЕМЕС* амалы.

NOR – *НЕМЕСЕ-ЕМЕС* амалы.

XOR – *НЕМЕСЕ*-ні шығару (2 модулі бойынша қосу амалы).

NOT – логикалық терістеу (*ЕМЕС*).

Number of input ports – кіріс порттарының саны.

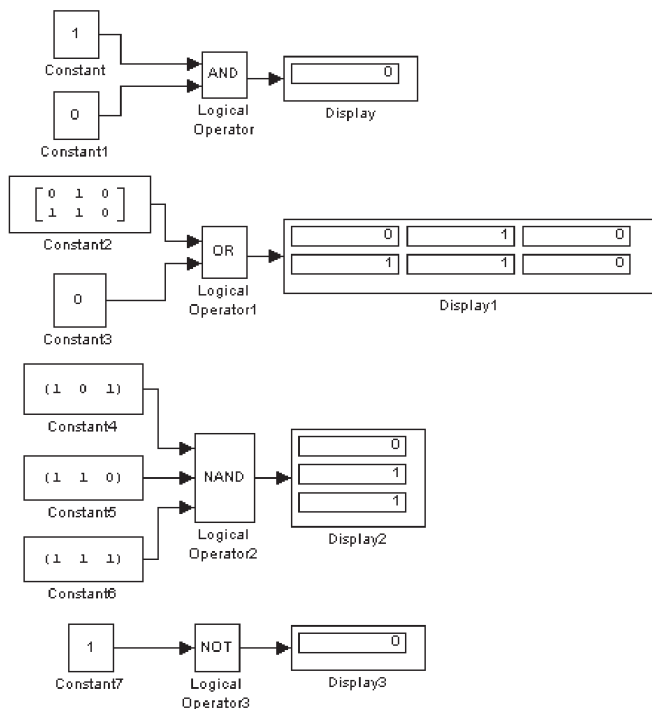
Егер логикалық амалдардың нәтижесі «ақиқат» болса, онда блоктың шығыс сигналы 1 болады да, егер «жалған» болса, блоктың шығыс сигналы 0 болады.

Блоктың кіріс сигналдары скалярлық, векторлық немесе матрицалық бола алады. Егер екі кіріс сигналы да векторлық немесе матрицалық болса, онда блок элементтер бойынша логикалық амалын орындайды. Бұл жағдайда кіріс сигналдарының өлшемі бір-біріне сәйкес келуі тиіс. Егер кіріс сигналдарының біреуі вектор немесе матрица болса, ал екіншісі - скаляр болса, онда блок скалярлық кіріс сигналын жиымның әрбір элементімен логикалық амалын орындайды. Шығыс сигналының өлшемі кіріске берілетін векторлық немесе матрицалық сигналдардың өлшемі бойынша анықталады.

Логикалық терістеу амалын орындау барысында блок тек бір кіріс портын ғана қолданады.

Кіріс сигналдары нақты да, логикалық типте де болуы мүмкін (*boolean*).

9.6.20-суретте *Logical Operation* блогын қолданудың мысалдары берілген.



9.6.20-сурет. *Logical Operation* блогын қолдану

9.6.18 Bitwise Logical Operator биттік логикалық амалдар блогы

Қызметі: Екілік түрдегі бүтін санға қарағандағы базалық логикалық амалдардың бірін жүзеге асырады.

Параметрлері:

Bitwise operator – қолданылатын логикалық амалдар көрінісі (тізімінен таңдалады):

AND – логикалық көбейту (*ЖӘНЕ* амалы).

OR – логикалық қойылу (*НЕМЕСЕ* амалы).

XOR – *НЕМЕСЕ*-ні шығару (2 модулі бойынша қосу амалы).

NOT – логикалық терістеу (*ЕМЕС*).

SHIFT_LEFT – разряд бойынша солға жылжу.

SHIFT_RIGHT – разряд бойынша оңға жылжу.

Second operand – екінші операнд. Символдық түрде оналтылық сан бойынша беріледі.

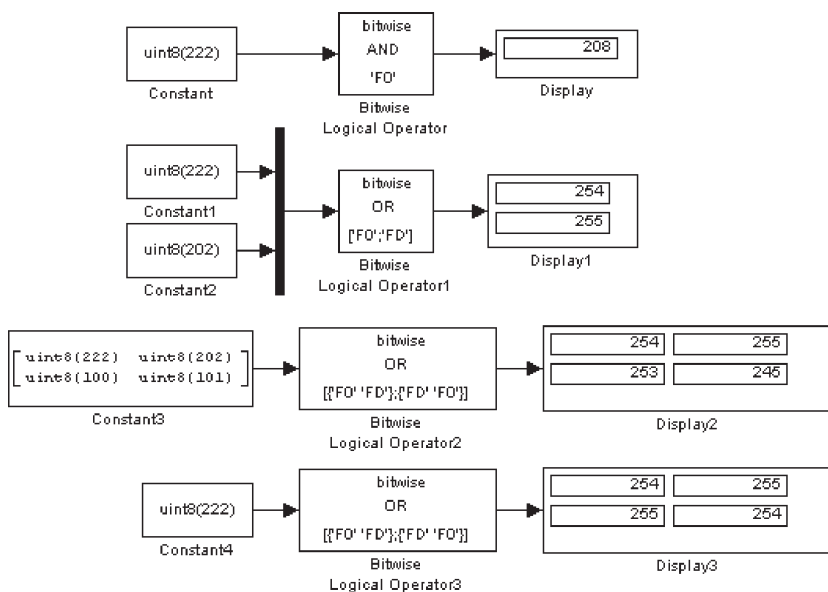
Bitwise Logical Operator блогының кірісіне берілетін сигнал бірінші операнды болып, ал *Second operand* блогының параметрі екінші операнды болып саналады.

Блоктың кіріс сигналдары ретінде *uint8*, *uint16* немесе *uint32* типтегі таңбасыз айнымалы болуы тиіс.

Блоктың кіріс сигналдары скалярлық, векторлық немесе матрицалық бола алады. Егер екі кіріс сигналы да векторлық немесе матрицалық болса, онда блок элементтер бойынша логикалық амалын орындайды. Бұл жағдайда кіріс сигналдарының өлшемі бір-біріне сәйкес келуі тиіс. Егер кіріс сигналдарының біреуі вектор немесе матрица болса, ал екіншісі – скаляр болса, онда блок скалярлық кіріс сигналын жиымның әрбір элементімен логикалық амалын орындайды. Шығыс сигналының өлшемі кіріске берілетін векторлық немесе матрицалық сигналдардың өлшемі бойынша анықталады.

Логикалық терістеу амалын орындау барысында блок тек бір операндты (кіріс сигналын) ғана қолданады.

9.6.21-суретте *Bitwise Logical Operator* блогының қолданылу мысалдары көрсетілген.



9.6.21-сурет. Bitwise Logical Operator блогының қолданылуы

9.6.19 Gombinatorical Logic комбинаторикалық логика блогы

Қызметі: Кіріс сигналдарын ақиқаттық кестесі бойынша өңдейді.

Параметрлері:

Truth table – ақиқат кестесі.

Combinatorial Logic блогы ақиқаттық кестесімен анықталатын ережелеріне сәйкес кіріс сигналдарының өңделуін қамтамасыз етеді. Ақиқаттық кестесі блоктың мүмкін болатын шығыс сигналының мәндерінің кестесін анықтайды. Мұндай құрылымдар жұмысының сипаттамасы шекті автоматтардың теориясында қабылданған. Осындай суреттеу жұмысы автомат теориясы бойынша жүзеге асады. Ақиқат кестесіндегі жолдар саны төмендегі қатынас арқылы анықталады:

$$number\ of\ rows = 2^{\square}(number\ of\ inputs),$$

мұндағы *number of inputs* – кіріс сигналының саны, *number of rows* – ақиқат кестесіндегі жол саны.

Кесте құруда кіріс сигналы берілген болып саналады. Олар блоктың шығыс мәндері жазылатын жолдың индексін (нөмірін) анықтайды. Әр жолдың индексі келесі өрнекпен анықталады:

$$row\ index = 1 + u(m) * 2^0 + u(m-1) * 2^1 + \dots + u(1) * 2^{m-1},$$

мұндағы *row index* – жол индексі, *m* – кіріс сигналының саны (кіріс векторындағы элементтер), *u(1)* – бірінші кіріс сигналы (кіріс вектордың бірінші элементі), *u(m)* – соңғы кіріс сигналы (кіріс вектордың соңғы элементі).

Мысалы, логикалық ЖӘНЕ (AND) амалы жағдайындағы екі операнд үшін жол индексін анықтайтын өрнек келесідей болады:

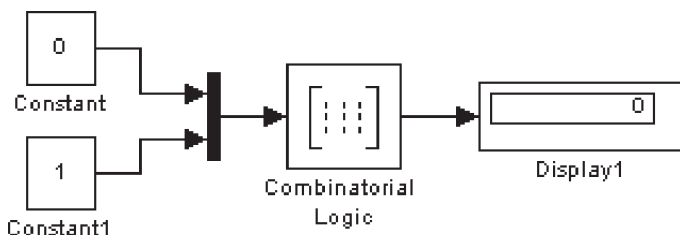
$$row\ index = 1 + u(2) * 2^0 + u(1) * 2^1.$$

Төменде логикалық ЖӘНЕ (AND) амалының екі операнд үшін ақиқат кестесінің қалыптасу мысалы көрсетілген:

9.6.2-кесте

2 кіріс	1 кіріс	Жол индексі үшін өрнек	Жол индексінің мәні	Ақиқат кестесі (шығыс)
0	0	$1 + 0 * 2^0 + 0 * 2^1$	1	0
1	0	$1 + 1 * 2^0 + 0 * 2^1$	2	0
0	1	$1 + 0 * 2^0 + 1 * 2^1$	3	0
1	1	$1 + 1 * 2^0 + 1 * 2^1$	4	1

9.6.22-суретте ЖӘНЕ логикалық амалының *Combinatorial Logic* блок көмегімен іске асуының мысалы көрсетілген. *Truth table* блогының параметрі [0;0;0;1] өрнегімен берілген.



9.6.22-сурет. *Combinatorial Logic* блогының қолданылуы

9.6.20 Algebraic Constraint алгебралық контурының блогы

Қызметі: Алгебралық теңдеулердің түбірін іздеуді орындайды.

Параметрлері:

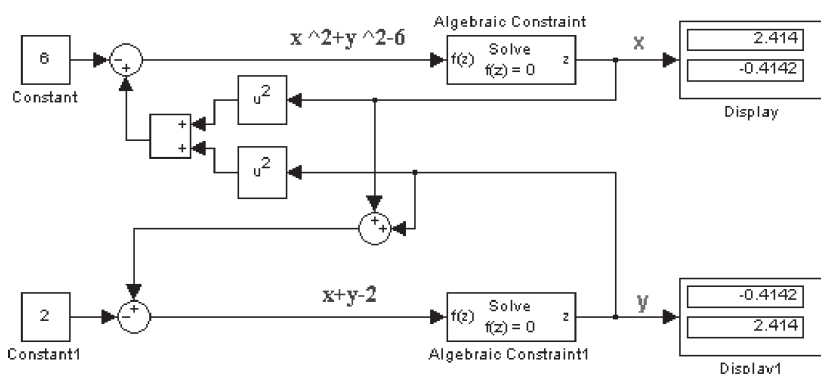
Initial guess – шығыс сигналының бастапқы мәні.

Блок кіріс сигналының мәні нөлге тең болған кездегі шығыс сигналының мәнін табады. Бұл кезде шығыс сигналы кіріс сигналымен төте немесе жанама түрде байланысқан болу керек.

9.6.23-суретте сызықтық емес теңдеулер жүйесінің шешу мысалдары көрсетілген:

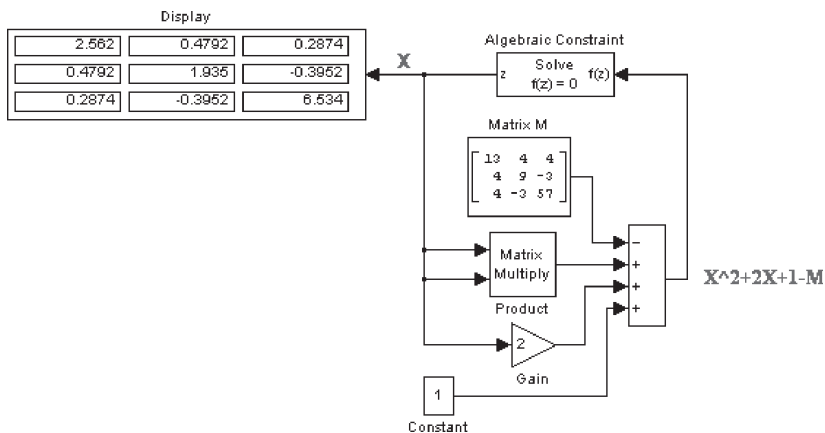
$$\begin{cases} x^2 + y^2 = 6, \\ x + y = 2. \end{cases}$$

Берілген теңдеулер жүйесінің екі шешімі болғандықтан, *Algebraic Constraint* блогының бастапқы мәндері векторлар түрінде берілген. Бірінші (жоғарғы) блок үшін бастапқы мән $[1 \ -1]$ векторымен берілген, ал екінші (төменгі) блок үшін $[-1 \ 1]$ векторымен берілген.



9.6.23-сурет. *Algebraic Constraint* блогын қолдану

Algebraic Constraint блогы сонымен қатар сызықтық емес матрицалық теңдеулерді шешу үшін де қолданылады.



9.6.23-сурет. Algebraic Constraint блогының сызықтық емес матрицалық теңдеулерді шешу үшін қолданылуы

9.6.24-суретте сызықтық емес матрицалық теңдеуінің шығарылу мысалдары көрсетілген:

$$X^2 + 2 \cdot X + 1 = \begin{vmatrix} 13 & 4 & 4 \\ 4 & 9 & -3 \\ 4 & -3 & 57 \end{vmatrix}$$

9.7 Signal&Systems – сигналдарды өңдеу блоктары мен көмекші блоктар

9.7.1 Mux мультиплексоры (араластырғыш)

Қызметі: Кіріс сигналдарын векторларға біріктіру.

Параметрлер:

Number of Inputs – кіріс саны.

Display option – бейнелеу әдісі. Ол келесі тізімнен таңдап алынады:

bar – тар қара түсті вертикальді тікбұрыш.

signals – ақ фонды және кіріс сигналдарының таңбалары бар тікбұрыш.

none – ақ фонды және кіріс сигналдарының таңбалары жоқ тікбұрыш.

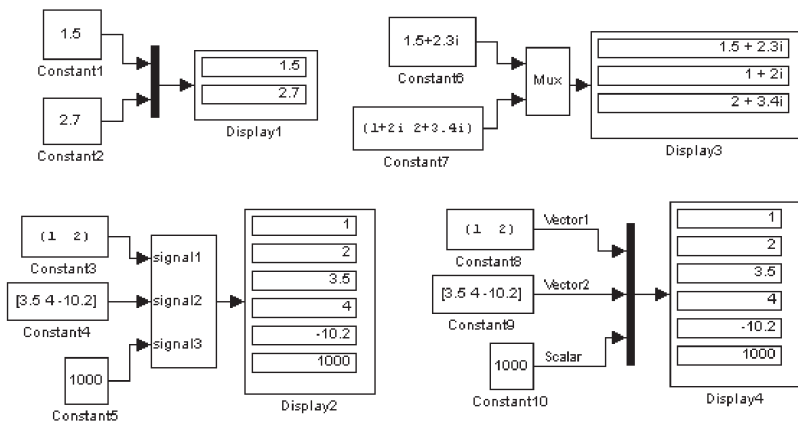
Блоктың кіріс сигналы скалярлық және (немесе) векторлық болуы мүмкін.

Егер кіріс сигналдарының ішінде векторлар бар болса, онда кіріс санын әрбір вектор элементінің санын көрсету арқылы вектор ретінде беруге болады. Мысалы, $[2 \ 3 \ 1]$ өрнегі үш кіріс сигналын анықтайды, бірінші сигнал – екі элементтен құралған вектор, екінші сигнал – үш элементтен құралған вектор және соңғы сигнал – скаляр. Егер кіріс векторының өлшемі *Number of Inputs* параметрінде көрсетілген өлшеммен сәйкес келмесе, онда Simulink есептеу басталғаннан кейін қателік туралы хабар береді. Кіріс векторының өлшемін *-1 (азайту бір)* ретінде беруге болады. Бұл жағдайда кіріс векторының өлшемі кез келген бола алады.

Number of Inputs параметрін сигнал таңбалары түріндегі тізімнен де белгілеуге болады, мысалы; *Vector1*, *Vector2*, *Scalar*. Бұл жағдайда сигналдардың таңбалары сәйкес келетін байланыстырушы сызықтармен бірге көрсетіле алады.

Блокқа берілетін кіріс сигналдары бірдей типте болуы керек (нақты немесе комплекстік).

Mux блогының қолданылу мысалы 9.7.1-суретте көрсетілген.



9.7.1-сурет. *Mux* блогының қолданылуы

9.7.2 Demux демультимплексоры (бөлгіш)

Қызметі: Векторлық кіріс сигналын жеке құраушыларға бөледі.

Параметрлері:

Number of Outputs – Шығыс саны.

Bus Selection Mode (жсалауша) – Векторлық сигналды бөлу режимі.

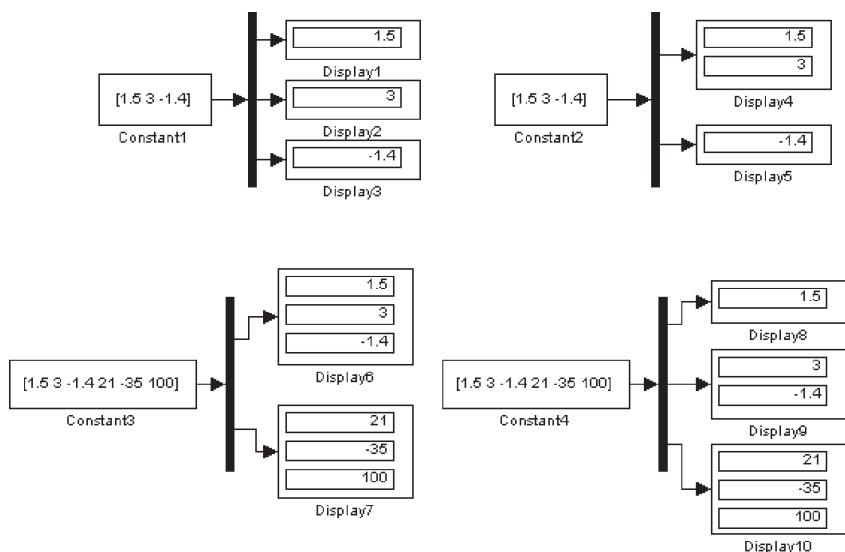
Қарапайым режимдегі кіріс сигналдары ретінде кез келген әдіспен құралған вектор бола алады. Шығыс сигналдары болып саны *Number of Outputs* параметрімен және өлшемі кіріс сигналының параметрмен және өлшемімен анықталатын скаляр немесе векторлар болады.

Егер P шығыс саны (*Number of Outputs* параметрінің мәні) N кіріс сигналының өлшеміне тең болса, онда блок кіріс векторын бөлек элементтерге бөлуді орындайды.

Егер P шығыс саны N кіріс сигналынан аз болса, онда бірінші $P-1$ шығыс сигналының өлшемі N/P дөңгеленген жуық үлкен мән қатынасына тең, ал соңғы шығыс сигналының өлшемі кіріс сигналының өлшемі мен бірінші $P-1$ шығыс өлшемдерінің қосындысы арасындағы түрлілікке тең. Мысалы, егер кіріс сигналының өлшемі 8-ге тең болса, ал шығыс саны 3-ке тең болса, онда бірінші екі шығыс векторы $ceil(8/3)=3$ өлшеміне тең, ал соңғы шығыс векторы $8-(3+3)=2$ өлшеміне тең.

Number of Outputs параметрі сонымен қатар әрбір шығыс сигналының өлшемін анықтайтын вектор көмегімен де берілуі мүмкін. Мысалы, $[2\ 3\ 1]$ өрнегі үш шығыс сигналының өрнегін анықтайды, бірінші сигнал – екі элементтен құралған вектор, екінші сигнал – үш элементтен құралған вектор, және соңғы сигнал – скаляр. Сонымен қатар өлшемді -1 (азайту бір) деп беруге болады. Бұл жағдайда шығыс сигналының сәйкестігінің өлшемі кіріс вектор өлшемі мен берілген шығыс сигналының қосындысы арасындағы айырмашылық ретінде анықталады. Мысалы, егер кіріс векторының өлшемі 6-ға тең, ал *Number of Outputs* параметрі $[1\ -1\ 3]$ өрнегімен берілсе, онда екінші шығыс сигналы, $6-(3+1)=2$ өлшеміне тең болады.

Demux блогын қолдану мысалдары 9.7.2-суретте көрсетілген.



9.7.2-сурет. Demux блогын қолдану

Bus Selection Mode режимінде *Demux* блогы жеке вектор элементтерімен емес, тұтас сигнал векторларымен жұмыс істейді. Бұл режимдегі кіріс сигналы *Mux* немесе басқа бір *Demux* блогымен құралуы тиіс. Бұл жағдайда *Number of Outputs* параметрі шығыс сигналдарының санын анықтайтын скаляр ретінде немесе әрбір элементі берілген шығыс сигналдарындағы векторлық сигналдардың санын анықтайтын түрінде беріледі. Мысалы, $[2 \ 1]$ векторымен берілген *Number of Outputs* параметрі үш вектордан тұратын кіріс сигналы болғанда, біріншісі екі, ал екіншісі бір векторлық сигналдан тұратын, екі шығыс сигналын бөледі.

9.7.3-суретте *Bus Selection Mode* режимінде *Demux* блогын қолданудың мысалдары көрсетілген.

9.7.3 Bus Creator шиналық құраушының блогы

Қызметі: Әр типтегі сигналдардан шинаны құрайды.

Параметрлері:

Signal naming options – сигналды атау тәсілі. Ол келесі тізімнен таңдап алынады:

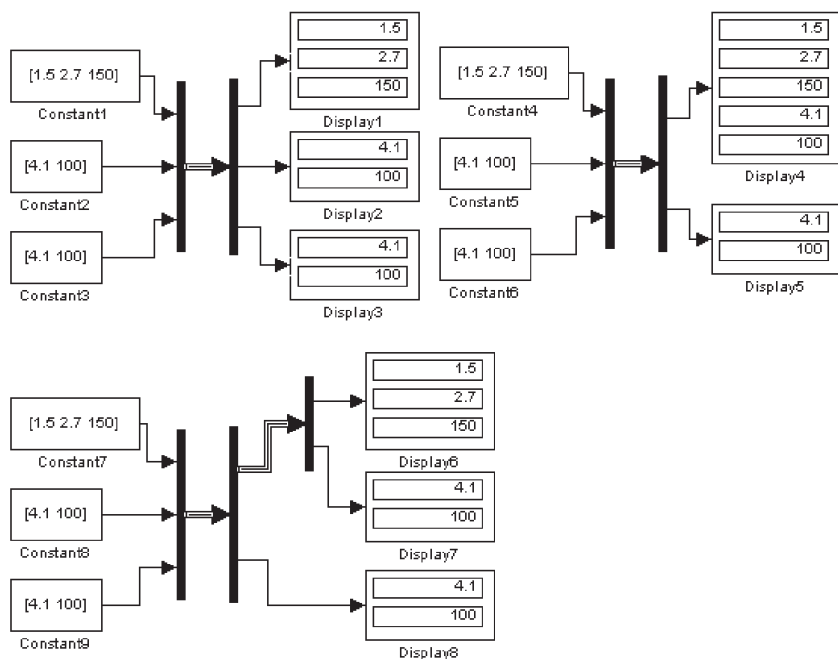
Inherit bus signal names from input ports – кіріс сигналдарының атауын алу.

Require input signal names to match signals below – кіріс сигналдарының атауын енгізу қажет.

Number of inputs ports – кіріс порттардың саны.

Signals in bus – шинаға бірігетін сигналдардың тізімі.

Rename selected signal – ерекшеленіп бөлінген сигналдың жаңа атауы. Бұл параметр *Require input signal names to match signals below* амалын таңдап алғанда ғана қолжетімдік дәрежеде болады.



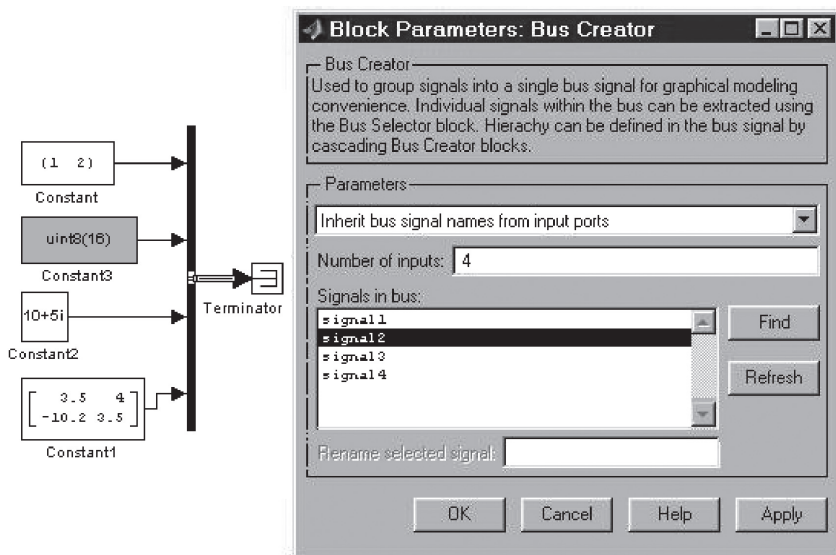
9.7.3-сурет. *Bus Selection Mode* режимінде *Demux* блогын қолдану

Блок кез келген әр түрлі векторлық, матрицалық, комплекстік, нақты және бүтін типтегі болатын сигналдарды бір шинаға біріктіруге мүмкіндік береді. Мұндай шина модельдегі біріккен сызықтардың санын қысқартуға мүмкіндік береді. Шиналарды жеке құраушыларға бөлу үшін *Bus Selector* блогын қолдану қажет.

Параметрлер блогының терезесі сигнал көзі болатын блокты іздеуге мүмкіндік береді. Мұндай іздеуге *Signals in bus* тізімінен сигнал атауын белгілеу керек те, тышқанның көмегімен *Find*

нүктесіне басу қажет. Таңдап алынған сигнал көзі болатын блок белгілі бір түспен белгіленеді.

9.7.4-суретте *Bus Creator* блогының көмегімен шинаны құрау мысалы және осы блоктың параметрлер терезесі көрсетілген. Бұл жерде, жоғарыда келтірілген процедураның *signal 2* сигнал көзі болатын *Constant3* блогы боялған түспен ерекшеленіп көрсетілген.



9.7.4-сурет. *Bus Creator* блогын қолдану

9.7.4 Bus Selector шиналық селектор блогы

Қызметі: Шинадан қажетті сигналдарды бөліп алады.

Параметрлер:

Signals in the bus – шинадағы бар сигналдар (кіріс сигналдары).

Selected signals – бөлінген сигналдар (шығыс сигналдары).

Mixed output (жалпауша) – шығыс сигналдарын біреуге біріктіру.

Шина *Mux* немесе *Bus Creator* блогымен құралуы мүмкін.

Шинадан сигналды шығару үшін блоктың параметрлер терезесін ашу қажет, *Signals in the bus* терезесінде сигналды белгілеп және *Select* тетігінің көмегімен сигналдың атауын

Selected signals терезесіне көшіру қажет. *Selected signals* тізімінен сигналды жою үшін, оң жақ тізімдегі параметрлер терезесінен оның атауын белгілеп, *Remove* батырмасын басу керек.

Up және *Down* батырмаларының көмегімен шинадағы сигналдардың орналасу ретін оларды *Selected signals* терезесінде, тиісінше, жоғары немесе төмен орын ауыстыру арқылы өзгертуге болады.

Mixed output параметрін қою сигналдарды шинаға біріктіруге мүмкіндік береді.

9.7.5-суретте *Bus Selector* блогын және оның параметрлер терезесін қолданудың мысалдары көрсетілген.

9.7.5 Selector селектор блогы

Қызметі: Вектордан немесе матрицадан қажетті элементтерді таңдайды.

Параметрлері:

Input Type – кіріс сигналының типі. Ол келесі тізімнен таңдап алынады:

vector – вектор.

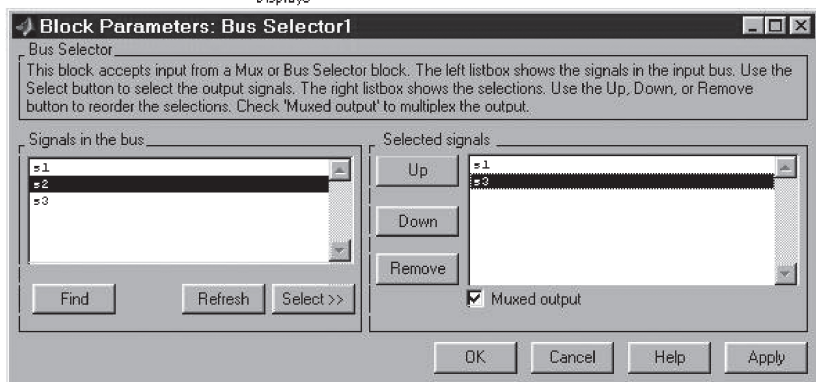
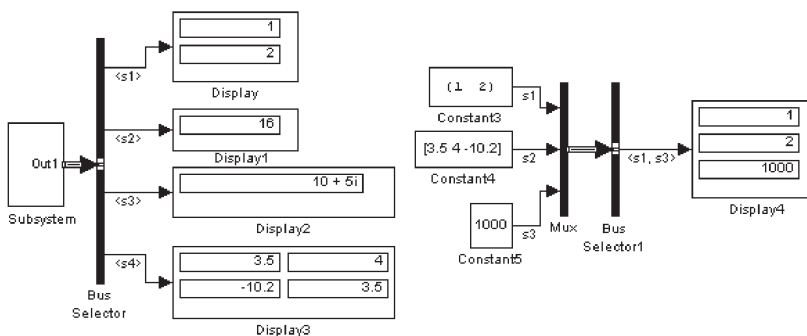
matrix – матрица.

Блок параметрлерінің тізімі кіріс сигналдарының типіне байланысты өзгереді.

Source of element indices – вектор элементтерінің индексінің көзі. Ол келесі тізімнен таңдап алынады:

internal – ішкі. *Elements* параметрімен таңдап алынған вектор элементтерінің индексі беріледі.

external – сыртқы. Вектор элементтерінің индексі, ол сыртқы кіріс сигналының көмегімен беріледі.



9.7.5-сурет. Bus Selector блогын колдану

Elements – блоктың шығысына берілетін кіріс векторының элементтерінің индекстерінің тізімі. Ол вектор түрінде беріледі. Параметрдің l (азайту бір) мәні вектордың барлық элементтерінің таңдауын белгілейді.

Input port width – кіріс векторының өлшемі.

Source of row indices – матрица элементтерінің жол индекстерінің көзі.

Rows – матрица жол индекстерінің тізімі.

Source of column indices – матрица элементтерінің баған индекстерінің көзі.

Columns – матрица баған индекстерінің тізімі.

Блоктың сыртқы түрі белгіленген блок параметріне байланысты өзгереді. Индекс элементтерінің сыртқы көзін таңдағанда блок кескінінде келесі таңбалармен белгіленген қосымша кірістер пайда болады:

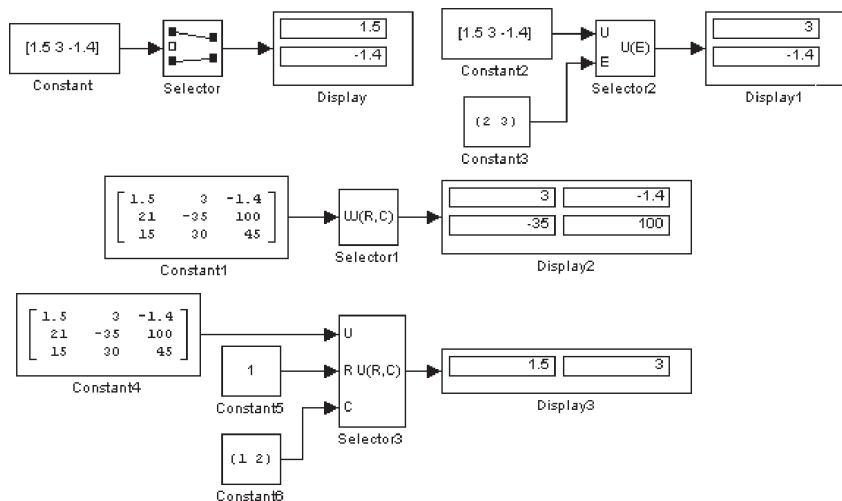
E – таңдалған вектор элементтерінің индекстерін беретін сигнал кірісі.

R – матрица жолының индекстерін беретін сигнал кірісі.

C – матрица бағанының индекстерін беретін сигнал кірісі.

Блок өз параметрлерімен немесе берілген сыртқы кіріс сигналымен анықталатын сигналдарды кіріс векторынан немесе матрицадан таңдап алады да, шығысқа береді.

9.7.6-суретте блоктың түрлі нұсқадағы күйге келтірулері үшін *Selector* блогын қолданудың мысалдары көрсетілген.



9.7.6-сурет. *Selector* блогын қолдану

9.7.6 Assignment жиым элементтеріне жаңа мәндер беру блогы

Қызметі: Вектор немесе матрица элементтерін ауыстырады.

Параметрлері:

Input Type – кіріс сигналының типі. Ол келесі тізімнен таңдап алынады:

vector – вектор.

matrix – матрица.

Блок параметрлерінің тізімі кіріс сигналдарының типіне байланысты өзгереді.

Source of element indices – вектор элементтерінің индексінің көзі. Ол келесі тізімнен таңдап алынады:

internal – ішкі. *Elements* параметрімен таңдап алынған вектор элементтерінің индексі беріледі.

external – сыртқы. Вектор элементтерінің индексі, ол сыртқы кіріс сигналының көмегімен беріледі.

Elements – блоктың шығысына берілетін кіріс векторының элементтерінің индекстерінің тізімі. Ол вектор түрінде беріледі. Параметрдің -1 (азайту бір) мәні вектордың барлық элементтерінің таңдауын белгілейді.

Source of row indices – матрица элементтерінің жол индекстерінің көзі.

Rows – матрица жол индекстерінің тізімі.

Source of column indices – матрица элементтерінің баған индекстерінің көзі.

Columns – матрица баған индекстерінің тізімі.

Блок индекстерді тізіміне сәйкес екінші кіретін жиымның элементтеріне бірінші кіретін жиымның жеке элементтерінің алмастыруын орындайды. Индекс тізімі блок параметрі ретінде немесе сыртқы басқарушы сигналдан есептелу арқылы беріледі.

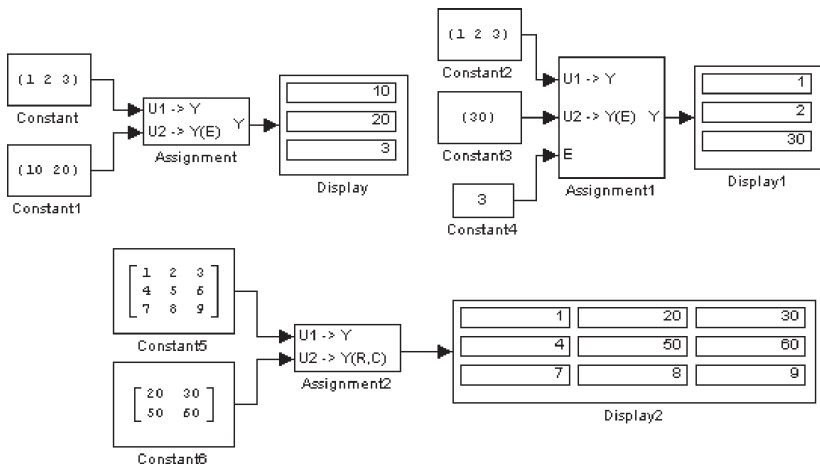
Блоктың сыртқы түрі белгіленген блок параметріне байланысты өзгереді. Индекс элементтерінің сыртқы көзін таңдағанда блок кескінінде келесі таңбалармен белгіленген қосымша кірістер пайда болады:

E – таңдалған вектор элементтерінің индекстерін беретін сигнал кірісі.

R – матрица жолының индекстерін беретін сигнал кірісі.

C – матрица бағанының индекстерін беретін сигнал кірісі.

9.7.7-суретте блоктың түрлі нұсқадағы күйге келтірулері үшін *Assignment* блогын қолданудың мысалдары көрсетілген.



9.7.7-сурет. Assignment блогын қолдану

9.7.7 Merge сигналдарды біріктіру блогы

Қызметі: Блок кіріс сигналдарын бір векторлық сигналға біріктіруін орындайды.

Параметрлері:

Number of inputs – кірістегі сан.

Initial output – шығыс сигналының бастапқы мәні. Егер бұл параметр берілмесе, онда шығыс блогының мәні соңғы есептелген сигнал өтеді.

Allow unequal port widths (жалауша) – кіріс порттарының өлшемдерінің бірдей болмауына рұқсат беру.

Input port offsets – кіріс сигналының жылжуы. Әрбір мәні шығыс векторындағы сәйкес сигналдың орналастырылуын анықтайтын вектор түрінде беріледі.

Блок соңғы болып есептелген шығыс сигналының мәнін береді.

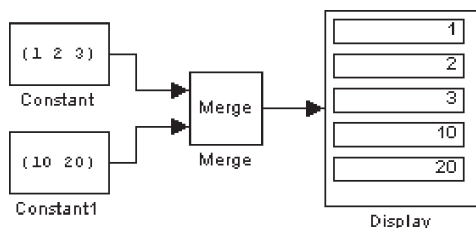
Input port offsets параметрінің көмегімен нәтижелік вектордағы кіріс сигналдарының орналасуын реттеуге болады.

Шығыс сигналының өлшемі келесі өрнекпен анықталады:

$$\max(w_1 + o_1, w_2 + o_2, \dots, w_n + o_n)$$

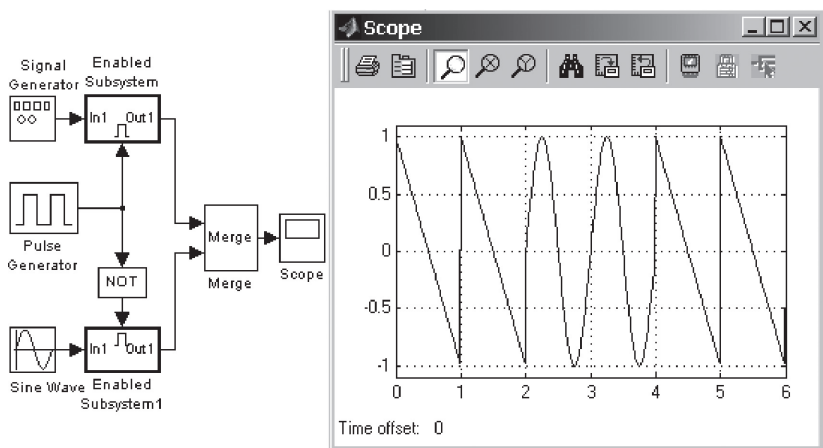
мұндағы w_n – n -ші кіріс сигналының өлшемі, o_n – n -ші кіріс сигналының жылжуы.

9.7.8-суретте *Merge* блогының екі векторды біріктіру үшін қолдану мысалы көрсетілген. *Input port offsets* параметрі мысалда $[0\ 3]$ векторымен берілген.



9.7.8-сурет. *Merge* блогын екі кіріс сигналын біріктіру үшін қолдану

Келесі мысал (9.7.9-сурет) соңғы деп есептелген блоктың қасиетін шығыс сигналына өткізуін көрсетеді. Мысалда ішкі жүйенің басқарылатын кірісіне нөлдік емес сигнал берілсе ғана есептеу жүргізетін *EnabledSubsystem* ішкі жүйесінің басқарылатын блоктары қолданады. Бұл жерде ішкі жүйе ешқандай нақты бір есептеулер жасамайды, ол тек өзінің кірісінен шығысына қарай сигналды өткізеді. Сайып келгенде, *Merge* блогының шығысына ара тәріздес, не гармониялық сигналдар кезектесіп өтеді.



9.7.9-сурет. *Merge* блогын қолдану

9.7.8 Matrix Concatenation сигналдарды матрицаға біріктіру блогы

Қызметі: Блок кіріс векторларының немесе матрицаларының біріктірулерін (конкатенция) орындайды.

Параметрлері:

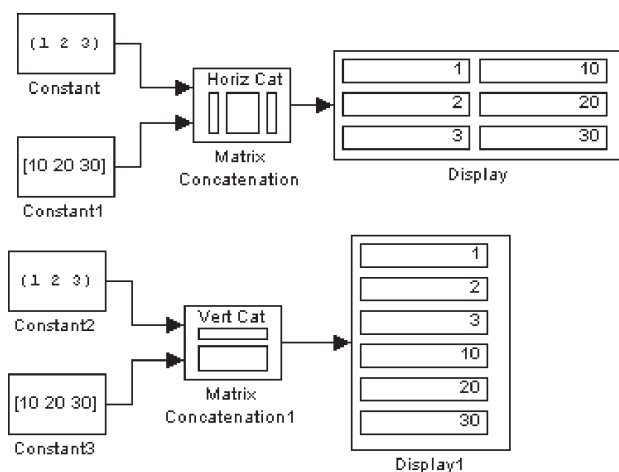
Number of inputs – кіріс саны.

Concatenation method – біріктіру әдісі. Ол келесі тізімнен таңдап алынады:

Horizontal – горизонталдық. Жиымдар жаңа жиымдарды оң жағына қосу арқылы бірігеді.

Vertical – вертикалдық. Жиымдар жаңа жиымдарды төмен жағына қосу арқылы бірігеді.

Matrix Concatenation блогын қолданудың мысалы 9.7.10-суретте көрсетілген.



9.7.10-сурет. Matrix Concatenation блогын қолдану

9.7.9 Goto сигналды беру блогы

Қызметі: Блок From блогына сигнал беруді орындайды.

Параметрлері:

Tag – сигнал идентификаторы.

Tag visibility – көріну белгісі. Ол келесі тізімнен таңдап алынады:

local – сигнал жергілікті ішкі жүйенің шегінде беріледі.

scoped – сигнал жергілікті және иерархияның төменгі деңгейіндегі ішкі жүйенің шегінде беріледі.

global – сигнал бүкіл модельдің шегінде беріледі.

Goto блогын *From* блогымен бірге қолдану байланыс сызығынсыз сигнал беруді қамтамасыз етеді. Ал берілетін сигнал типі кез келген бола береді.

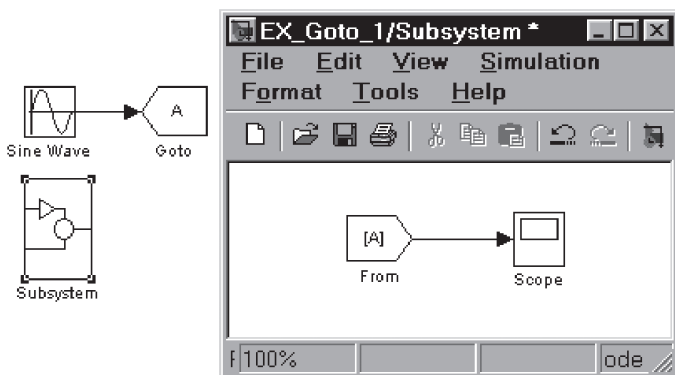
Блоктың сыртқы түрі таңдап алынған *Tag visibility* параметріне байланысты өзгереді:

Егер көріну белгісі *local* мәніне ие болса, сигналдың идентификаторы квадраттық жақшаның ішінде жазылады. Мысалы, $[A]$, A – сигнал идентификаторы.

Егер көріну белгісі *scoped* мәніне ие болса, сигнал идентификаторы фигуралық жақша ішінде жазылады. Мысалы, $\{A\}$.

Егер көріну белгісі *global* мәніне ие болса, сигнал идентификаторы блоктың пиктограммасында қосымша белгілерсіз көрінеді.

9.7.11.-суретте «сымсыз» сигнал беру тәсілі көрсетілген (синусоидалық сигнал).



9.7.11-сурет. *Goto* блогын қолдану

9.7.10 From сигнал қабылдау блогы

Қызметі: Блок *Goto* блогынан сигнал қабылдауды орындайды.

Параметрлері:

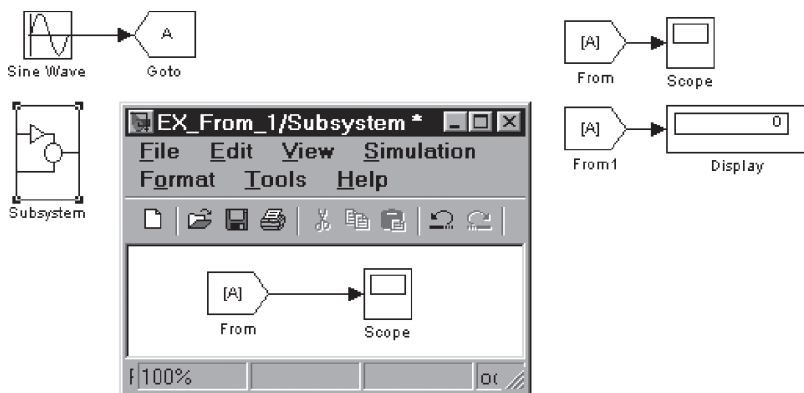
Goto tag – қабылданушы сигналдың идентификаторы. Ол *Goto* блогында көрсетілген идентификатормен сәйкес болуы тиіс.

From блогын *Goto* блогымен бірге қолдану сигналды байланыс сызығынсыз беруді қамтамасыз етеді.

Блок пиктограммасында сигналдың көріну белгісі *Goto* блогындағы оны қолдану тәсіліндегідей болады.

Модельде бір *Goto* блогынан сигнал қабылдайтын бірнеше *From* блоктары болуы мүмкін.

9.7.12-суретте *From* блогын қолданудың мысалы келтірілген. Бұл мысалда бір *Goto* блогы үш *From* блогына сигнал беруде (басты модельде – екеуіне, ал ішкі жүйеде – біреуіне).



9.7.12-сурет. *From* блогын қолдану

9.7.11 Goto Tag Visibility сигналдың көріну белгілерінің блогы

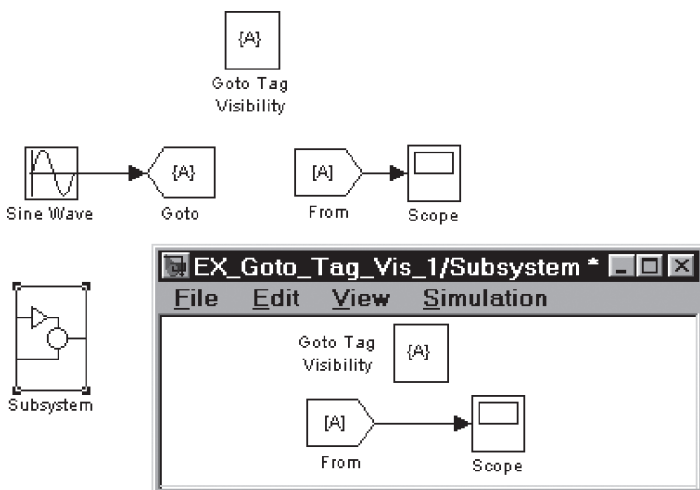
Қызметі: Блок *Goto* блогымен берілетін сигналдың көріну белгілерін береді.

Параметрлері:

Gototag–*Goto* блогымен берілетін сигналдың идентификаторы.

Егер берілетін сигналдар үшін оның көріну аймағы *scoped* берілетін болса, онда блокты модельдің немесе оның ішкі жүйесінің құрамына кіргізу керек болады. Блок берілетін деректердің көріну аймағы таралуын ішкі жүйелерінде орналастырылады. Блок сигнал беруге қатыспай, тек берілетін сигналдың атауын ғана көрсетеді.

Блокты қолданудың мысалы 9.7.13-суретте көрсетілген.



9.7.13-сурет. *Goto Tag Visibility* блогын қолдану

9.7.12 Data Store Memory жадтың ортақ аймағын құру блогы

Қызметі: Блок берілген мәліметтерді сақтау үшін жадтың атауы аталған аймақ құрады.

Параметрлері:

Data store name – жад аймағының атауы.

Initial value — бастапқы мән.

Interpret vector parameters as 1-D (жалалуша) – берілгендер векторының параметрлерін бірөлшемді вектор ретінде қабылдау.

Бұл блок *Data Store Write* (деректерді жазу) және *Data Store Read* (деректерді оқу) блоктарымен бірге қолданылады.

Initial value параметрі сигналдың бастапқы мәнімен және өлшеммен берілмейді. Мысалы, егер сигналдың бастапқы мәні $[0 \ 1; \ 2 \ 3]$ матрицасымен берілсе, онда сақталыныатын сигнал 2×2 матрицасы болуы тиіс.

Егер *Data Store Memory* блогы модельдің жоғарғы деңгейінде орналасса, онда оларға берілген жад аймағын модельдің өзінде де, иерархияның төменгі деңгейіндегі барлық ішкі жүйеде де қолдануға болады. Егер *Data Store Memory* блогы ішкі жүйеде орналасса, онда оларға берілген жад аймағын осы ішкі жүйеде

және иерархияның төменгі деңгейіндегі барлық ішкі жүйеде қолдануға болады.

Блок *double* типтегі нақты сигналдармен жұмыс істейді.

9.7.14-суретте *Data Store Memory* блогын *Data Store Write* және *Data Store Read* блоктарымен бірге қолданудың мысалдары көрсетілген.

9.7.13 Data Store Write деректерді жадтың ортақ аймағына жазу блогы

Қызметі: Блок атауы аталған жад аймағына берілген мәліметтерді жазады.

Параметрлері:

Data store name – жад аймағының атауы.

Sample time – модельдеу уақытының адымы.

Жазба амалы алдыңғы адымдағы есептеуден алынған сигналдың мәні үшін орындалады.

Модельде бір жад аймағына жазба орындап жатқан бірнеше *Data Store Write* блогын қолдануға болады. Алайда, егер жазба есептеудің бір адымында өтіп жатса, онда алынатын нәтиже нақты болмайды.

9.7.14-суретте *Data Store Write* блогын *Data Store Memory* және *Data Store Read* блоктарымен бірге қолданудың мысалдары көрсетілген.

9.7.14 Блок Data Store Read деректерді жадтың ортақ аймағынан оқу блогы

Қызметі: Блок атауы аталған жад аймағынан берілген мәліметтерді оқиды.

Параметрлері:

Data store name – жад аймағының атауы.

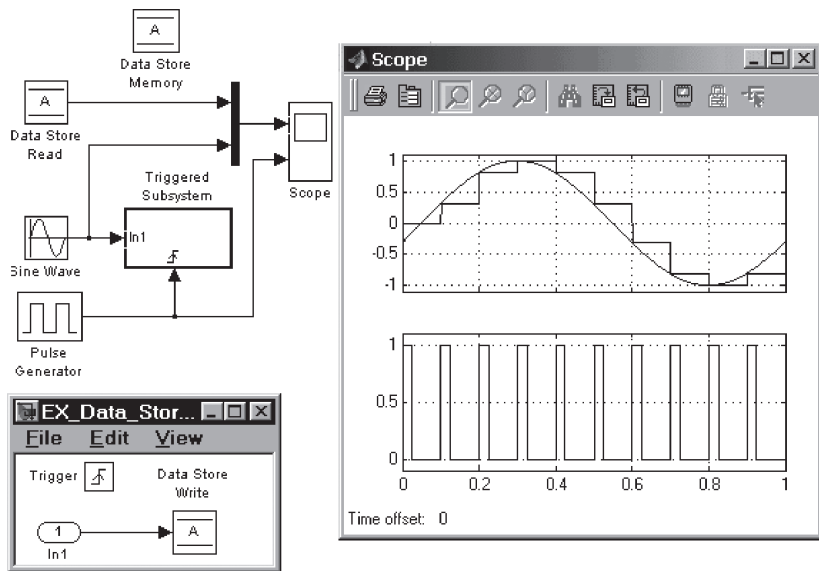
Sample time – модельдеу уақытының адымы.

Оқу амалы есептеудің әрбір адымында жүргізіледі.

Модельде бір жадтың деректерін оқитын бірнеше *Data Store Read* блоктары қолданылуы мүмкін.

9.7.14-суретте *Data Store Read* блогын *Data Store Memory*

және *Data Store Write* блоктарымен бірге қолданудың мысалдары көрсетілген. Мысалда басқарушы сигналдың алдыңғы шебі бойынша есептеу жүргізетін триггерлік ішкі жүйе қолданылған. Яғни, жад мәні дұрыс бағыттағы сигналды басқару кезінде ғана іске асады. Ал қалған уақытта жад аймағында өзгермейді.



9.7.14-сурет. *Data Store Memory*, *Data Store Write* және *Data Store Read* блоктарын қолдану

9.7.15 Data Type Conversion сигнал типін түрлендіру блогы

Қызметі: Блок кіріс сигналының типін түрлендіреді.

Параметрлері:

Data type – берілген шығыс сигналының түрі. Ол келесідей мәндерді қабылдай алады: *auto*, *double*, *single*, *int8*, *int16*, *int32*, *uint8*, *uint16*, *uint32* және *boolean*.

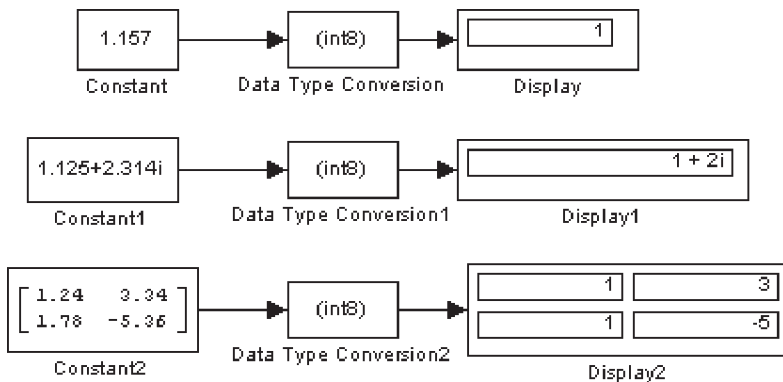
Saturate on integer overflow (жалауша) – бүтіннің артып кетуін басу. Жалауша тұрған жағдайда бүтін түрдегі сигналдарды шектеу дұрыс орындалады.

Егер блоктың кіріс портына берілген блоктан алынған сигналдың типіне тәріздес етіп қою керек болса, онда *Data type* параметріндегі *auto* мәні қолданылады.

Блоктың кіріс сигналы нақты немесе комплекстік болуы мүмкін. Кіріс сигналы комплекстік болған кезде шығыс сигналы да комплекстік болады.

Блок скалярлық, векторлық және матрицалық сигналдармен жұмыс істейді.

9.7.15-суретте *Data Type Conversion* блогын қолданудың мысалы көрсетілген.



9.7.15-сурет. *Data Type Conversion* блогын қолдану

9.7.16 Reshape сигнал өлшемін түрлендіру блогы

Қызметі: Блок векторлық немесе матрицалық сигналдардың өлшемдерін өзгертеді.

Параметрлері:

Output dimensionality – шығыс сигналы өлшемінің түрі. Ол келесідей тізімнен таңдап алынады:

1-D array – бір өлшемді жиым (вектор).

Column vector – вектор-баған.

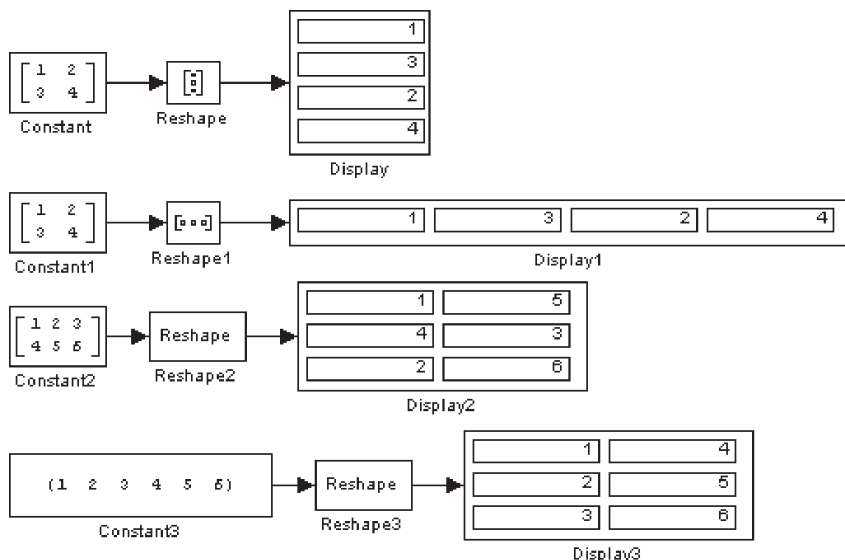
Row vector – вектор-жол.

Customize – берілген өлшемдегі матрица немесе вектор. Берілген векторлық шығыс сигналының параметрі ретінде шығыс вектордың элементтер санын анықтаушы скаляр беріледі. Ал матрицалық шығыс сигналының параметрі ретінде шығыс матрицасының бағаны мен жолының санын анықтаушы вектор беріледі. Параметр мәні кірістегі жиым элементінің санына

сәйкес болуы керек. Матрицалық сигнал болған кезде берілген деректер кіріс матрицаларының бағанынан таңдап алынады да, шығыс матрицаларының бағанына енгізіледі.

Output dimensions – шығыс сигналы өлшемінің мәні. Егер өлшем түрі *Customize* тәрізді орнатылса, онда параметр қолжетімді болады.

9.7.16-суретте *Reshape* блогын қолданудың мысалдары көрсетілген.



9.7.16-сурет. *Reshape* блогын қолдану

9.7.17 Width сигналдың өлшемін анықтау блогы

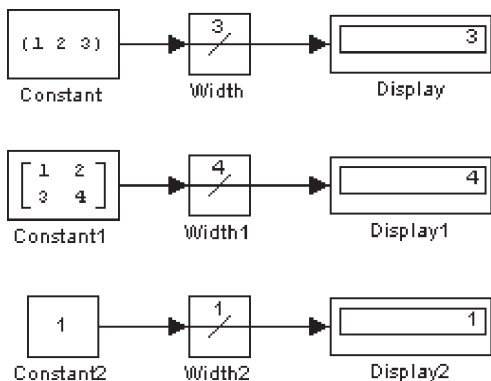
Қызметі: Кіріс сигналының өлшемін есептейді.

Параметрлері: Жоқ.

Блоктың кіріс сигналы кез келген типтегі нақты немесе комплекстік түрде болуы мүмкін.

Блоктың шығыс сигналы *double* түрінде болады.

Width блогының қолданылу мысалдары 9.7.17-суретінде көрсетілген:



9.7.17-сурет. *Width* блогының қолданылуы

9.7.18 Hit Crossing шектік мәннің қиылысу сәтін анықтайтын блок

Қызметі: Кіріс сигналының берілген шектік мәнімен қиылысқан кездегі уақыт моментін анықтайды.

Параметрлері:

Hit crossing offset – шек. Кіріс сигналының қиылысуымен анықталуы тиіс мән.

Hit crossing direction – қиылысу бағыты. Ол келесідей тізімнен таңдап алынады:

rising – өсу.

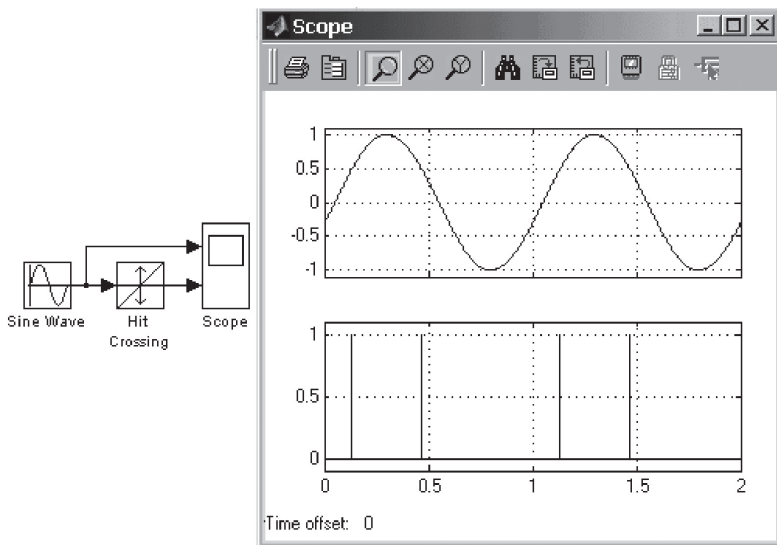
failing – кему.

either – екі бағытта.

Show output port (жалауша) – шығыс портын көрсету. Егер жалауша алынып тасталған болса, онда шектік деңгейдің сигналмен қиылысу нүктесі табылады, бірақ блокпен шығыс сигналы шығарылмайды.

Қиылысу сәтінде блок модельдеу уақытының бір адымына тең болатын жеке сигнал жасап шығарады.

Hit Crossing блогының қолданылу мысалы 9.7.18-суретінде көрсетілген. Блок 0.5-тік деңгейдегі тең синусоидалық сигналға тең екі бағыттағы қиылысу сәтін анықтайды.



9.7.18-сурет. *Hit Crossing* блогының қолданылуы

9.7.19 IC сигналдың бастапқы мәнін қою блогы

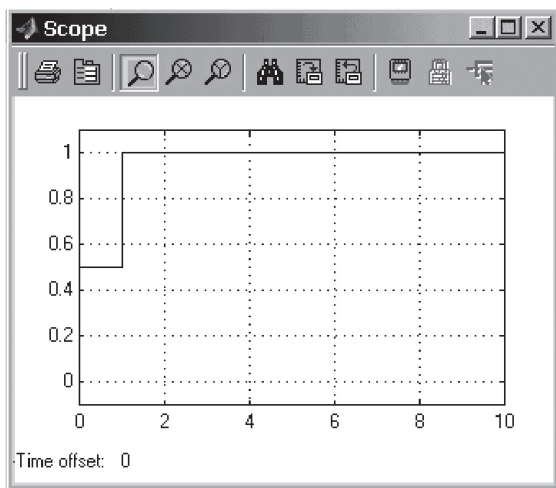
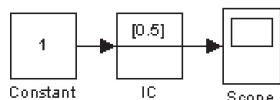
Қызметі: Сигналдың бастапқы мәнін береді.

Параметрлері:

Initial value – бастапқы мәні.

IC блогының шығыс сигналы кіріс сигналының шамасынан тәуелсіз есептеудің бірінші адымында *Initial value* параметрінің мәніне тең. Есептің қалған адымдарында кіріс сигналы блогтың шығысына ешқандай өзгеріссіз өтеді.

IC блогының қолданылу мысалы 9.7.19-суретте көрсетілген. Берілген мысалда сигналдың бастапқы мәні *0.5*-ке тең етіп берілген. Есептеу адымы *1c*-ке тең деп берілген.



9.7.19-сурет. IC блогының қолданылуы

9.7.20 Signal Specification сигналды тексеру блогы

Қызметі: Сигнал үшін берілген параметрлермен сәйкес сигналдың тексерілуін орындайды.

Параметрлері:

Dimension – сигнал өлшемі. Егер кіріс сигналы векторлық немесе матрица $[m \ n]$ түрлерінде болса (m – жол саны, n – баған саны), егер кіріс сигналы матрица болса, онда ол скалярмен беріледі. Егер параметр мәні -1 (азайту 1) түрінде берілсе, онда тексерілу жүрмейді.

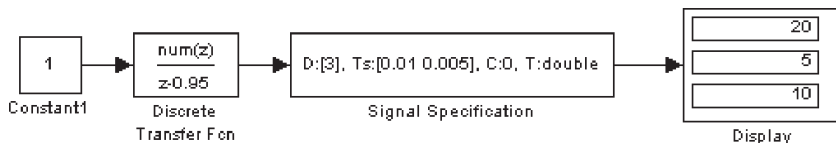
Sample time – модельдеу уақытының адымы. $[period \ offset]$ вектор түрінде беріледі, мұндағы *period* – модельдеу уақытының адымының мәні, *offset* – жылжу. Егер параметр мәні -1 (азайту 1) түрінде берілсе, онда тексерілу жүргізілмейді. Сонымен қатар -1 (азайту 1) мәнін *period* немесе *offset* параметрлеріне жеке-жеке беруге де болады. Бұл жағдайда, дәл осы параметрлерге тексеру жүргізілмейді.

Data Type – деректер типі. Ол келесідей тізімнен алынады: *auto* (тексеру жүргізілмейді), *double*, *single*, *int8*, *uint8*, *int16*, *uint16*, *int32*, *uint32* немесе *boolean*.

Signal type – сигнал типі. Ол келесідей тізімнен алынады: *auto* (тексеру жүргізілмейді), *real* немесе *complex*.

Блоктың пиктограммасында сигналдың тексерілетін

параметрлері мен оның мәндері кейіптеледі. *Signal Specification* блогын қолданудың мысалы 9.7.20-суретте көрсетілген.



9.7.20-сурет. *Signal Specification* блогының қолданылуы

9.7.21 Probe сигналдың қасиеттерінің датчигі

Қызметі: Блок сигнал параметрінің сандық мәнін алуға мүмкіндік береді.

Параметрлері:

Probe width (жалауша) – векторлық немесе матрицалық сигналдағы элементтер санының анықталуы.

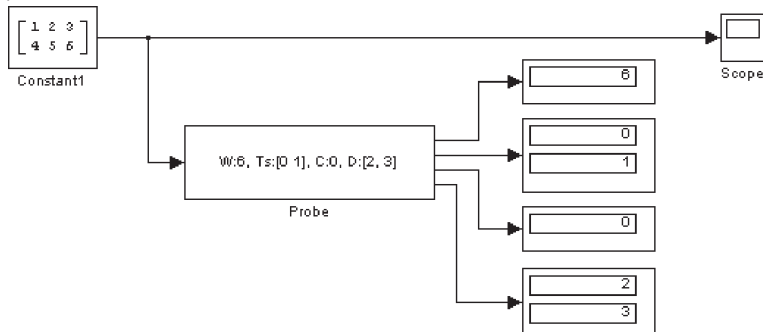
Probe Sample time (жалауша) – эталондық уақыт мәнінің анықталуы.

Probe Complex Signal (жалауша) – сигнал типінің анықталуы (егер сигнал комплекстік түрде берілсе 1-ді және кері жағдайда 0-ді қайтарады).

Probe signal dimension (жалауша) – сигнал өлшемінің анықталуы. Жалаушалар қойылған параметрлер қадағаланады. Блок шығысының саны белгіленген жалаушалар саны арқылы беріледі.

Кез келген параметр үшін жалаушаның қойылуы, берілген сигнал параметрінің мәнін есептеуге болатын порттың блок кескінінде пайда болуына алып келеді.

Probe блогының қолданылу мысалы 9.7.21-суретте көрсетілген.



9.7.21-сурет. *Probe* блогының қолданылуы

9.7.22 Function-Call Generator итерация санын беретін блок

Қызметі: Басқарылатын ішкі жүйе үшін модельдеу уақытының әр адымында итерация санын беруге мүмкіндік береді.

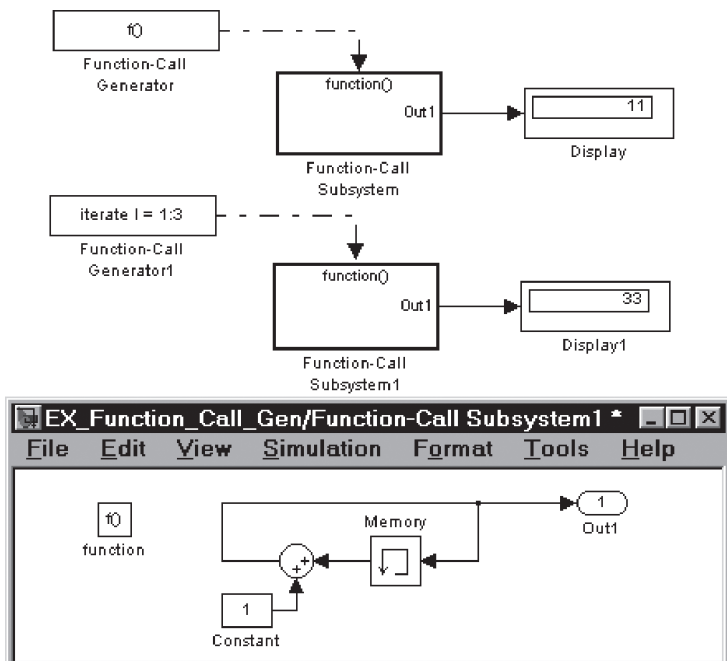
Параметрлері:

Sample time – модельдеу уақытының адымы.

Number of iterations – итерация саны.

Блок *Function-Call Subsystem* немесе *Triggered Subsystem* басқарылатын ішкі жүйелерімен бірге қолданылады. Басқарушы блоктар үшін бұл ішкі жүйелердің ішінде *Trigger type* параметрі *function-call* мәнін қабылдауы керек.

9.7.22-суретте *Function-Call Generator* блогының қолданылу мысалы көрсетілген. Бұл мысалда шығыс сигналы әрбір оның шақыруында бірге артатын, басқарылатын ішкі жүйе қолданылады. Бірінші ішкі жүйе үшін *Function-Call Generator* блогы әрбір адымы 1-ге тең, ал екіншісі үшін 3-ке тең итерация санын береді.



9.7.22-сурет. *Function-Call Generator* блогының қолданылуы

9.7.23 Model Info ақпараттық блогы

Қызметі: Блок модель туралы ақпаратты кескіндейді.

Параметрлері:

Model properties – модель қасиеттері:

Created – модельдің жасалған күні мен уақыты.

Creator – авторы туралы деректер.

Modified by – өзгерістер енгізген қолданушы туралы деректер.

ModifiedDate – өзгеріс уақыты.

ModifiedComment – өзгеріске сипаттама.

ModelVersion – модельдің нұсқасы.

Description – модельге сипаттама.

LastModificationDate – соңғы рет өзгертілген уақыт.


Horizontal text alignment – мәтінді көлденеңінен теңестіру тәсілі. Ол келесідей тізімнен алынады:

Center – ортасы бойынша.

Left – сол жақ шетімен.

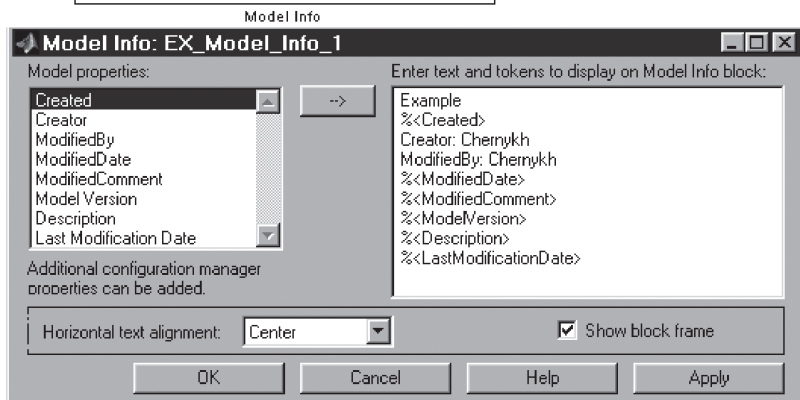
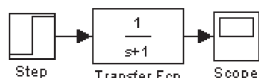
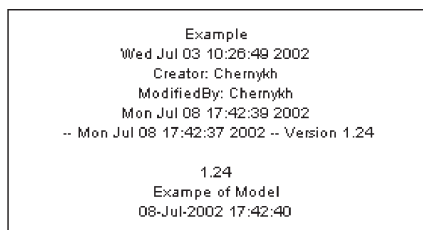
Right – оң жақ шетімен.

Show block frame (жалалуша) – блоктың рамкасын кескіндеу.

Деректерді блоктың пиктограммасында көрсету үшін  батырмасының көмегімен қажетті параметрді *Model properties* терезесінен редакторлау терезесіне қарай көшіреді. Блокта қолданушының өзі енгізетін статикалық ақпарат (мәселен, автор туралы дерек, модельдің сипаттамасы және т.б.) және динамикалық жаңаланатын ақпарат (мысалы, модельдің жасалған уақыты, түрленімнің соңғы уақыты және т.б.) кескінделуі мүмкін. Динамикалық жаңаланатын ақпарат блок терезесінде оны құрайтын өзгерістегі сілтеме түрінде көрсетіледі. Сілтеме `%<айнымалы_атауы>` түрінде болады. Мысалы, `%<LastModificationDate>` сілтемесі модельдің соңғы түрленім уақытын беретін *LastModificationDate* айнымалысының мәні талап етілген позицияда шығарылатынын түсіндіреді.

Сонымен қатар, модель терезесіндегі *File* мәзірінің *Model Properties* командасының көмегімен берілген ақпараттың бір бөлігі де блок пиктограммасында кескінделеді.

Model Info блогының қолданылу мысалы 9.7.23-суретте көрсетілген. Онда берілген блоктың параметр терезесі көрсетілген.



9.7.23-сурет. Model Info блогының қолданылуы

9.8 Function & Tables – функциялар мен кестелердің блоктары

9.8.1 Fcn функцияны беру блогы

Қызметі: C бағдарламалау тілінің стиліндегі өрнекті береді.

Параметрлері:

Expression – кіріс негізінде шығыс сигналын есептеу үшін блокпен қолданылатын өрнек. Бұл өрнек функцияны C тілінде сипаттау үшін қабылданған ереже бойынша құрылады.

Өрнекте келесідей компоненттерді қолдануға болады:

- Кіріс сигналы. Кіріс сигналы егер скаляр болса, онда ол өрнекте u түрінде беріледі. Егер кіріс сигналы вектор болса, онда вектор элементінің нөмірін дөңгелек жақшада көрсету керек. Мысалы, $u(1)$ және $u(3)$ –кіріс векторының бірінші және үшінші элементі.

- Тұрақтылар.

- Арифметикалық операторлар (+ - * /).
- Қатынас операторлары (= != > < >= <=).
- Логикалық операторлар (&& || !).
- Дөңгелек жақшалар.
- Математикалық функциялар: *abs, acos, asin, atan, atan2, ceil, cos, cosh, exp, fabs, floor, hypot, ln, log, log10, pow, power, rem, sgn, sin, sinh, sqrt, tan, және tanh.*

- Жұмысаймағындағы айнымалылар. Егер жұмыс аймағындағы айнымалы жиым болып табылса, онда оның элементтері дөңгелек жақшадағы индекстер арқылы көрсетілуі керек. Мысалы, $A(1,1)$ – A матрицасының бірінші элементі.

Қатынас операторлары мен логикалық операторларды логикалық нөл (*FALSE*) және логикалық бірлік (*TRUE*) түрінде қайтарады.

Өрнекте қолданылатын операторлардың басымдықтары келесідей ретпен орналастырылады (кему реті бойынша):

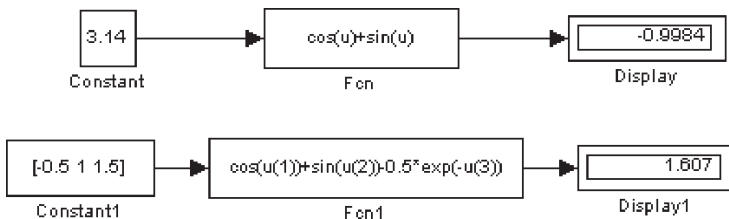
```

()
+ - (унарлық)
Дәрежелену
!
/
+ - (бинарлық)
> < <= >=
= !=
&&
||

```

Блок матрицасы және векторлы амалдарды қабылдамайды. Блоктың шығыс сигналы әрдайым скаляр болады.

Fcn блогының қолданылу мысалы 9.8.1-суретте көрсетілген.



9.8.1-сурет. *Fcn* блогының қолданылуы

9.8.2 MATLAB Fcn функцияны беру блогы

Қызметі: MATLAB бағдарламалау тілінің стиліндегі өрнекті береді.

Параметрлері:

MATLAB function – MATLAB тіліндегі өрнек.

Output dimensions – шығыс сигналының өлшемі. Параметр мәні –1 (азайту бір) блокта өлшемінің автоматты түрде анықталуын жазып береді.

Output signal type – шығыс сигналының типі. Ол келесідей тізімнен алынады:

Real – нақты сигнал.

Complex – комплекстік сигнал.

Auto – сигнал типінің автоматты түрде анықталуы.

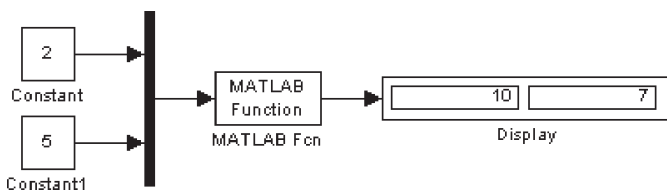
Collapse 2-D results to 1-D – шығыс сигналының екіөлшемін бірөлшемге түрлендіру.

Кіріс сигналы егер скаляр болса, онда ол өрнекте u түрінде беріледі. Егер кіріс сигналы вектор болса, онда вектор элементінің нөмірін дөңгелек жақшада көрсету керек. Мысалы, $u(1)$ және $u(3)$ – кіріс векторының бірінші және үшінші элементі. Егер өрнек бір функциядан тұрса, онда оны параметрлерін көрсетусіз-ақ беруге болады. Өрнек MATLAB тілінде жазылған және M -файлдар түрінде безендірілген қолданушының меншікті функциясын да құрайды. M -файлдың аты модельдің атымен (mdl -файлмен) сәйкес келмеуі керек.

9.8.2-суретте *MATLAB Fcn* блогының қолданылуы көрсетілген. Бұл мысалда кіріс векторының екі элементінің көбейтіндісі мен қосындысын есептеу үшін *My_Matlab_Fcn_1* функциясы қолданылады. Функцияның мәтіні (файл *My_Matlab_Fcn_1.m*) төменде келтірілген:

```
function y=My_Matlab_Fcn_1(x,k);  
y(1)=x*k;  
y(2)=x+k;
```

MATLAB function параметрімен берілген функцияны шақыру үшін кескіннің мынадай түрі болады: *My_Matlab_Fcn_1(u(1),u(2))*.



9.8.2-сурет. *MATLAB Fcn* блогының қолданылуы

9.8.3 Polynomial дәрежелік көпмүшені беру блогы

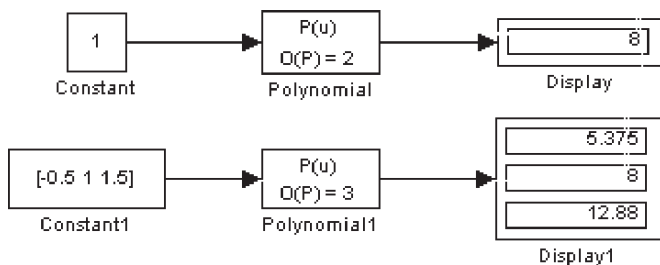
Қызметі: Дәрежелік көпмүшені береді.

Параметрлері:

Polynomial coefficients – полином коэффициенттерінің векторы. Коэффициенттер векторда тәуелсіз айнымалының кему дәрежелері бойынша орналасқан. Мысалы, x^2+2x+5 полиномы үшін коэффициенттерінің векторын $[1 \ 2 \ 5]$ түрінде беру керек. Коэффициенттердің типі нақты болу керек.

Блок полином мәнін оның коэффициенттері мен кіріс сигналының шамасы бойынша есептейді. Егер кіріс сигналы вектор немесе матрица болса, блок нәтижені әрбір жиым элементі үшін есептейді.

9.8.3-суретте *Polynomial* блогының қолданылу мысалы көрсетілген. Бұл мысалда бірінші полиномды блок үшін коэффициенттер $[1 \ 2 \ 5]$ векторымен, ал екіншісі үшін – $[1 \ 2 \ 0 \ 5]$ векторымен берілген.



9.8.3-сурет. *Polynomial* блогының қолданылуы

9.8.4 Look-Up Table бір өлшемді кесте блогы

Қызметі: Бір айнмалысы бар функцияны кесте түрінде береді.

Параметрлері:

Vector of input values – кіріс сигналының мән векторы. Мән дискреттік түрде берілуі мүмкін (мысалы, $[1\ 2\ 7\ 9]$) немесе үздіксіз мөлшер түрінде де берілуі мүмкін (мысалы, $[0:10]$). Вектор элементтері немесе мөлшер шекарасы есептелетін өрнек түрінде берілуі мүмкін, мысалы $[\tan(5)\ \sin(3)]$.

Vector of output values – кіріс мәніне сәйкес келетін шығыс мәнінің векторы.

Блок мына ережелерге сай жұмыс істейді:

Егер кіріс сигналы кіріс мәндерінің векторының қандай да бір элементіне тең болса (*Vector of input values*), онда блоктың шығыс мәні шығыс мәндерінің векторының лайықты элементіне (*Vector of output values*) тең болады. Мысалы, кіріс мәнінің векторы $[0\ 1\ 2\ 5]$ -ға тең болсын, ал шығыс мәнінің векторы $[-5\ -10\ 3\ 100]$ -ға, онда 1-ге тең кіріс сигналы кезінде шығыс сигналы -10 -ға тең болады.

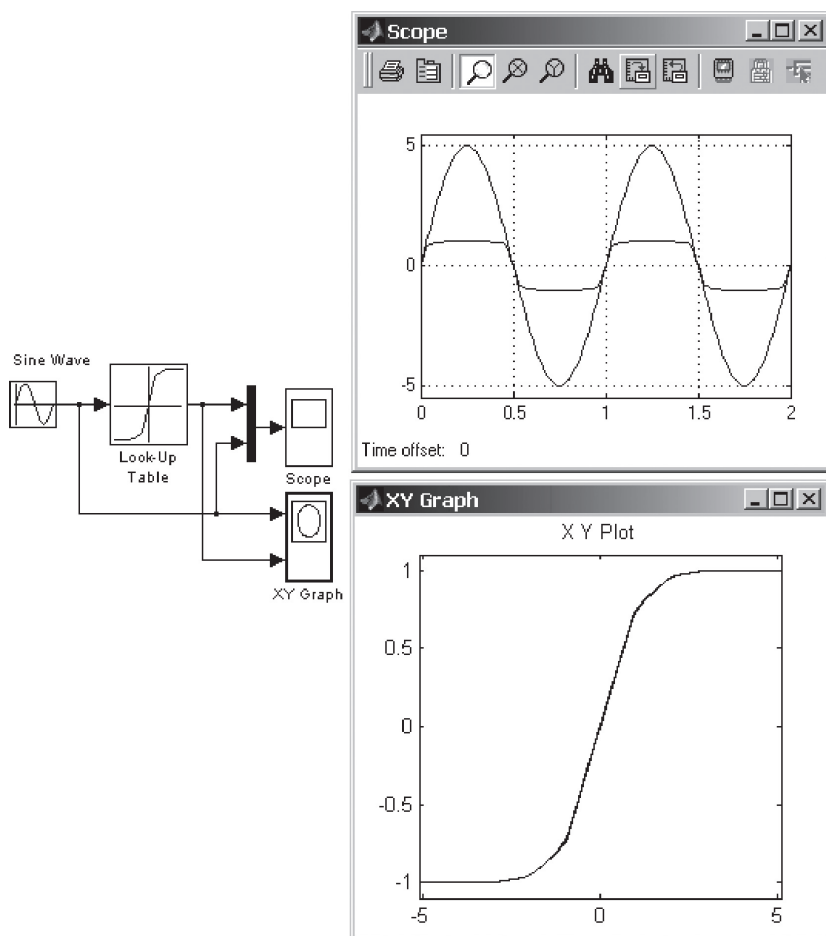
Егер кіріс сигналы кіріс мәндерінің векторының ешқандай элементіне сәйкес келмесе, онда блок оған екі жақын элементтің арасында сызықтық интерполяциялауды орындайды.

Егер кіріс сигналы кіріс мәндерінің векторының шекарасынан шығып кетсе, онда блок шеткі екі элементі бойынша сызықтық экстраполяциялауды орындайды.

Блоктың күйге келтірулерінің көмегімен берілген функцияның графигі оның пиктограммасында кескінделеді.

Блоктың кіріс сигналы векторлық болуы мүмкін. Мұндай кезде блок элементтер бойынша амалды орындайды.

9.8.4-суретте *Look-Up Table* блогының қолданылу мысалы көрсетілген. Мысалда кіріс мәндерінің векторы $[-5:5]$ -ке тең, ал шығыс мәндерінің векторы $\tanh([-5:5])$ тең.



9.8.4-сурет. Look-Up Table блогының қолданылуы

9.8.5 Look-Up Table(2D) екі өлшемді кесте блогы

Қызметі: Екі айнымалысы бар функцияны кесте түрінде береді.

Параметрлері:

Row – жол. Бірінші аргумент мәнінің векторы. Бір өлшемді кесте *Vector of input values* параметрі тәрізді беріледі. Вектор элементтері өсу реті бойынша орналасулары керек.

Column – баған. Екінші аргумент мәнінің векторы. Алдыңғы параметрге ұқсас беріледі.

Table – функция мәндерінің кестесі. Матрица түрінде беріледі. Жол саны *Row* вектор элементтерінің санына тең болу керек, ал баған саны – *Column* вектор элементтерінің санына тең.

Функция мәндерінің кестесін қалыптастыру ережесі 9.8.1-кестеде көрсетілген.

9.8.1-кесте.

		Екінші аргумент (Column)		
		3	7	9
Бірінші аргумент (Row)	2	10	20	30
	4	40	50	60
	8	70	80	90

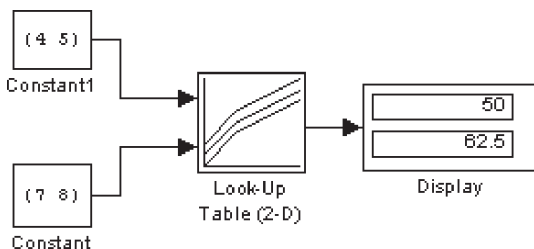
Келтірілген кестедегі блок параметрлерінің мәндері мынадай болады:

Row – [2 4 8],

Column – [3 7 9],

Table – [10 20 30; 40 50 60; 70 80 90].

Look-Up Table(2D) блогының қолданылу мысалдары 9.8.5-суретте көрсетілген. Блоктың параметрлері 9.8.1-кестесімен сәйкестендіріп берілген.



9.8.5-сурет. *Look-Up Table(2D)* блогының қолданылуы

9.8.6 Look-Up Table (n-D) көпөлшемді кестенің блогы

Қызметі: Көп айнымалысы бар функцияны кесте түрінде береді.

Параметрлері:

Number of table dimensions – кесте өлшемінің саны (функция

аргументтері). Параметр мәні келесідей тізімнен алынады: 1, 2, 3, 4, More...(көп).

First input (row) breakpoint set – бірінші аргумент мәнінің векторы (жол). Екі өлшемді кестедегі Row тәріздес беріледі.

Second (column) input breakpoint set – екінші аргумент мәнінің векторы (баған). Алдыңғы параметр тәріздес беріледі.

Third input breakpoint set – үшінші аргумент мәнінің векторы. Егер кесте өлшемінің саны 2-ден көп берілсе, онда бұл параметрді пайдалануға болады.

Fourth input breakpoint set – төртінші аргумент мәнінің векторы. Егер кесте өлшемінің саны 3-тен көп берілсе, онда бұл параметрді пайдалануға болады.

Fifth..Nth input breakpoint sets (cell array) – Бесінші және қалған аргумент мәндерінің жиымы (ұяшықтар жиымы). Егер кесте өлшемінің саны 4-тен көп берілсе, онда бұл параметрді пайдалануға болады.

Explicit number of dimensions – кесте өлшемінің нақты саны (функция аргументтері). *Егер Number of table dimensions* параметрінің More мәні болса, параметрді пайдалануға болады және оны беру керек.

Index search method – индекс бойынша іздеу амалы. Ол келесідей тізімнен мәндер қабылдайды:

Evenly Spaced Points – біркелкі артта қалушы индекстер үшін іздеу. Егер аргумент векторларының біркелкі артта қалушы мәндері бар болса, онда іздеу жылдамдық бойынша өте жақсы нәтиже береді (мысалы, [10 20 30]).

Linear Search – сызықтық іздеу. Егер ағымдағы есептеудің адымындағы кіріс сигналдарының мәндері алдыңғы мәндерден шамалы ғана айырмашылықта болса, өте жақсы нәтиже береді.

Binary Search – қосалқы іздеу. Егер ағымдағы есептеудің адымындағы кіріс сигналдарының мәндері алдыңғы мәндерден біршама айырмашылықта болса, өте жақсы нәтиже береді.

Begin index searches using previous index results (жалалуша) – алдыңғы іздеудің нәтижелерін қолдана отырып, іздеуді бастау.

Use one (vector) input port instead of N ports (жалалуша) – Бірнеше бір өлшемдінің орнына көпөлшемді кіріс портын қолдану.

Table data – функция мәндерінің кестесі. Көпөлшемді жиымдарды қалыптастыру ережесі бойынша беріледі.

Interpolation method – интерполяция әдісі. Ол келесідей тізімнен алынады:

None – интерполяция орындалмайды.

Linear – сызықтық интерполяция.

Cubic Spline – кубтық сплайн-интерполяция.

Extrapolation method – экстраполяция әдісі. Ол келесідей тізімнен алынады: *None*, *Linear* немесе *Cubic Spline*.

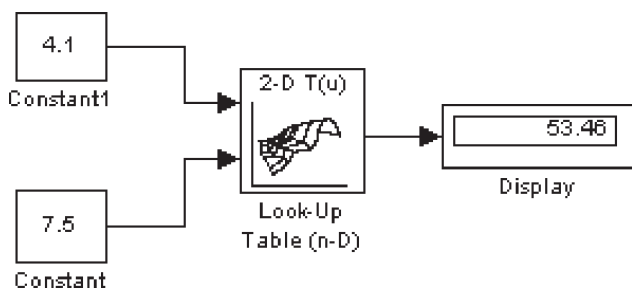
Action for out of range input – аргумент мәндері векторының шекарасынан шығуына кіріс сигналының реакциясы. Ол келесідей тізімнен алынады:

None – реакция жоқ.

Warning – MATLAB-тың командалық жолына ескерту хабарламасының қорытындысын шығару.

Error – MATLAB-тың командалық жолына қателік туралы хабарламасының қорытындысын шығару және есептеуді тоқтату.

Екі аргумент функциясының тапсырмасы үшін *Look-Up Table (n-D)* блогының қолданылу мысалы 9.8.6-суретінде көрсетілген. 9.8.1-кестесіне сәйкес блоктың параметрлері берілген. Шығыс мәндерін есептеу үшін кубтық сплайн-интерполяция берілген.



9.8.6-сурет. *Look-Up Table (n-D)* блогының қолданылуы

9.8.7 Direct Loop-Up Table (n-D) тікелей енуі бар кестенің блогы

Қызметі: Көпөлшемді кестені оның элементтеріне тура дерекпен береді. Элементтер индексациясы нөлден басталады.

Параметрлері:

Number of table dimensions – кесте өлшемінің саны (функция аргументтері). Параметр мәні келесідей тізімнен алынады: 1, 2, 3, 4, More...(көп).

Explicit number of dimensions – кесте өлшемінің нақты саны (функция аргументтері). Егер *Number of table dimensions* параметрінің *More* мәні болса, параметрді беруге болады.

Inputs select this object from table – шығыс сигналының түрін беру керек. Ол келесідей тізімнен алынады:

Element – элемент. Егер блоктың шығысында кестенің жеке элементін алу керек болса, онда блоктың кірісіне элементтің барлық индекс мәндері берілуі керек.

Column – жол. Егер шығыста баған алу керек болса, онда блоктың кірісіне алдыңғы нұсқамен салыстырғанда бір индекске кіші етіп беру керек.

D Matrix – матрица. Бұл жағдайда блоктың кірісіне бірінші нұсқамен салыстырғанда екі индекске кіші етіп беру керек.

Make table an input – функция мәнінің кестесі *Table data* параметрімен емес, блоктың жеке кірісі арқылы беріледі.

Table data — функция мәнінің кестесі. Көпөлшемді жиымдардың қалыптасу ережелері арқылы беріледі.

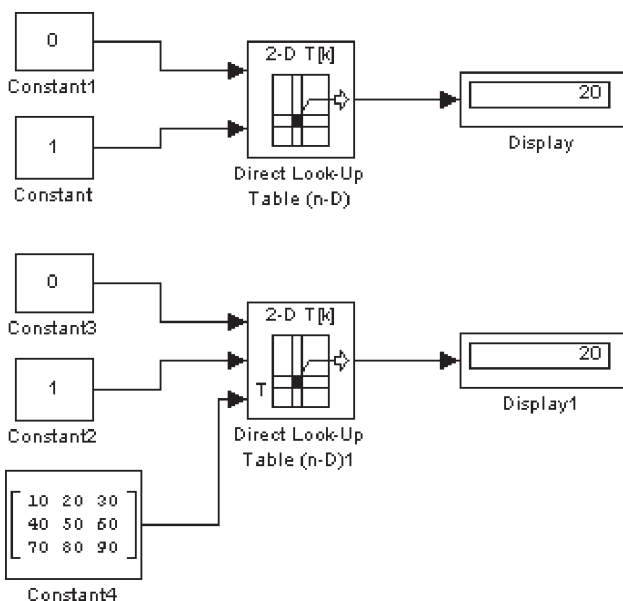
Action for out of range input – аргумент мәндері векторының шекарасынан шығуына кіріс сигналының реакциясы. Ол келесідей тізімнен алынады:

None – реакция жоқ.

Warning – MATLAB-тың командалық жолына ескерту хабарламасының қорытындысын шығару.

Error – MATLAB-тың командалық жолына қателік туралы хабарламасының қорытындысын шығару және есептеуді тоқтату.

Екі аргумент функциясы үшін *Look-Up Table (n-D)* блогының қолданылу мысалы 9.8.7-суретте көрсетілген. Функция мәнінің кестесі бірінші жағдайда блоктың параметрінде берілген ([10 20 30;40 50 60;70 80 90]), ал екіншісінде жеке кіріс арқылы берілген (*Make table an input* жалаушасы қойылған).



9.8.7-сурет. Direct Loop-Up Table (n-D) блогының қолданылуы

9.8.8 PreLook-Up Index Search индекстермен жұмыс істеу блогы

Қызметі: Кіріс сигналының қатынас шамасын және индекс мәнін есептейді. *Interpolation (n-D) using PreLook-Up* блогымен бірге қолданылады.

Параметрлері:

Breakpoint data – Түйіндік нүктелердің векторы. Берілген параметр кестелік функцияны беруші блоктың кіріс сигналының векторы тәріздес.

Index search method – индекстерді іздеу әдісі. Ол келесідей тізімнен алынады:

Evenly Spaced Points – біркелкі артта қалушы индекстер үшін іздеу.

Linear Search – сызықтық іздеу.

Binary Search – қосалқы іздеу.

Begin index search using previous index result (жалаяуша) – алдыңғы іздеудің нәтижелерін қолдана отырып, іздеуді бастау.

Output only the index (жалаяуша) – тек индекстерді шығару.

Process out of range input – кіріс сигналының берілген

шектерден шығуы кезіндегі үдеріс типі. Ол келесідей тізімнен алынады:

Clip to Range – шектік мәнмен шектеу.

Linear Extrapolation – сызықтық экстраполяция.

Action for out of range input – аргумент мәндері векторының шекарасынан шығуына кіріс сигналының реакциясы. Ол келесідей тізімнен алынады:

None – реакция жоқ.

Warning – MATLAB-тың командалық жолына ескерту хабарламасының қорытындысын шығару.

Error – MATLAB-тың командалық жолына қателік туралы хабарламасының қорытындысын шығару және есептеуді тоқтату.

Блоктың шығыс сигналы вектор болып табылады, вектордың бірінші элементі – табылған индекс, ал екіншісі – кіріс сигналының салыстырмалы шамасы. Блок мәні кіріс сигналының шамасынан аспайтын элементтің индексін табады. Мысалы, түйіндік нүктелердің $[0 \ 5 \ 10 \ 20 \ 50 \ 100]$ векторы және 55-ке тең кіріс сигналы үшін табылған индексі 4-ке тең.

Кіріс сигналының салыстырмалы шамасы

$$h = \frac{x - A(i)}{A(i+1) - A(i)},$$

өрнегіне сәйкес есептелінеді, мұндағы

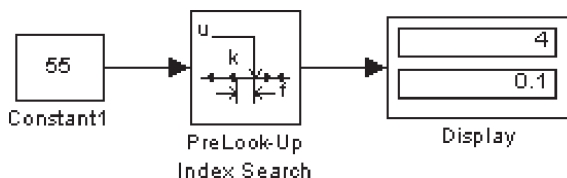
x – кіріс сигналы,

i – табылған индекс,

A – түйіндік нүктелердің векторы.

Жоғарыда келтірілген мысал үшін кіріс сигналының салыстырмалы шамасы 0.1-ге тең.

Блоктың жұмысын түсіндіретін мысал 9.8.8-суретте көрсетілген.



9.8.8-сурет. *PreLook-Up Index Search* блогын қолдану

9.8.9 Interpolation (n-D) using PreLook-Up кестелік функцияны интерполяциялау блогы

Қызметі: Индекс мәні үшін кестелік функцияның және кіріс сигналының қатынас шамасының мәнін есептейді. PreLook-Up Index Search блогымен бірге қолданылады.

Параметрлері:

Number of table dimensions – кесте өлшемінің саны (функция аргументтері). Параметр мәні келесідей тізімнен алынады: 1, 2, 3, 4, More...(көп).

Explicit number of dimensions – кесте өлшемінің нақты саны (функция аргументтері). Егер *Number of table dimensions* параметрінің *More* мәні болса, параметрді беруге болады.

Table data — функция мәнінің кестесі. Көпөлшемді жиымдардың қалыптасу ережелері арқылы беріледі.

Interpolation method – интерполяция әдісі. Ол келесідей тізімнен алынады:

None – интерполяция орындалмайды.

Linear – сызықтық интерполяция.

Extrapolation method – экстраполяция әдісі. Ол келесідей тізімнен алынады: *None*, *Linear* немесе *Cubic Spline*.

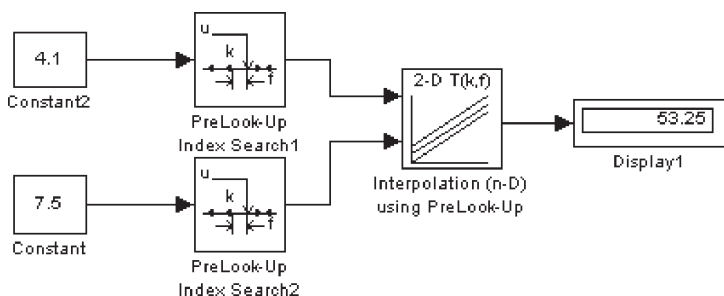
Action for out of range input – аргумент мәндері векторының шекарасынан шығуына кіріс сигналының реакциясы. Ол келесідей тізімнен алынады:

None – реакция жоқ.

Warning – MATLAB-тың командалық жолына ескерту хабарламасының қорытындысын шығару.

Error – MATLAB-тың командалық жолына қателік туралы хабарламасының қорытындысын шығару және есептеуді тоқтату.

Блоктың жұмысын көрсететін мысал 9.8.9-суретте көрсетілген. Функция мәндерінің кестесі матрицамен берілген [10 20 30;40 50 60;70 80 90].



9.8.9-сурет. *Interpolation (n-D) using PreLook-Up* блогының қолданылуы

9.9 Subsystem – ішкі жүйелер

Ішкі жүйе жеке блок түрінде рәсімделген *Simulink*-модельдің бөлігі. Ішкі жүйе модель құру кезіндегі қолданылуының келесідей пайдалы тұстары бар:

- экранда бір уақытта кейіптелген блоктардың санын азайтады, яғни модельдің қабылдануын жеңілдетеді (негізінде, модель монитордың экранында толығымен кейіптелуі керек);

- модельдің бөліктерін жеке-жеке құруға және дұрыстауға мүмкіндік береді, модель құрылуының технологиялылығын жоғарылатады;

- жеке кітапхана құруға мүмкіндік береді;

- ішкі жүйенің параллельді жұмыс істеуін синхронизациялауға мүмкіндік береді. үлгіге жеке анықтамалық деректерді қосуға мүмкіндік береді;

- ішкі жүйені қандай да бір *M*-файлмен байланыстырып, ішкі жүйенің ашылуы кезіндегі осы файлдың іске қосылуын қамтамасыз етуге мүмкіндік береді (ішкі жүйенің стандартсыз ашылуы);

- ішкі жүйені және оның блоктарының механизмін қолдануда өзінің стандартты рәсімделуі бойынша кем түспейтін блоктарды жасауға мүмкіндік береді. (блок параметрлерінің жеке терезесі, пиктограмма, анықтама және т.б.).

Модельде ішкі жүйенің саны шектелмеген және де ішкі жүйелер өзіне басқа ішкі жүйелерді қосып ала алады. Ішкі жүйенің деңгейі шектелмеген.

Ішкі жүйенің модельмен байланысы (немесе иерархияның жоғары деңгейіндегі ішкі жүйемен) кіріс (*Sources* кітапханасының *Inport* блогы) және шығыс (*Sinks* кітапханасының *Outport* блогы) порттарының көмегімен орындалады. Ішкі жүйеге кіріс немесе шығыс портын енгізсе, ішкі жүйенің кескінінде, сыртқы сигналдардың ішкі жүйенің ішіне берілуіне немесе негізгі модельге шығарылуына көмектесетін, порт белгілерінің пайда болуына алып келеді. *Inport* немесе *Outport* блоктарының атын өзгерту қолданушыға керек стандартты (*In* және *Out*) ішкі жүйенің пиктограммасында көрсетілген порттың белгілерін өзгертуіне алып келеді. Ішкі жүйелер *виртуалдық* (*Subsystem*) және *біртұтас* (*Atomic Subsystem*) болуы мүмкін. Ішкі жүйенің бұл түрлері есептеу уақытындағы блоктардың орындалу ретімен айрықшаланады. Егер ішкі жүйе *виртуалдық* болса, онда Simulink блоктың есептеу кезінде модельден мұндай ішкі жүйені бөлетін шекараны жояды. Басқаша айтқанда, виртуалдық жүйеде алғашқы болып бірнеше блоктардың шығыс сигналдары есептелуі мүмкін, содан соң негізгі модельдің блоктарының есептеулері орындалып, содан соң қайтадан ішкі жүйеге кіруші блоктардың есептеулері орындалуы мүмкін. *Біртұтас* жүйе тұтас (монолиттік бөлінбейтін) блок болып есептеледі және Simulink барлық блоктың есептеуін осындай ішкі жүйеде, негізгі модельдің басқа блоктарының есептеуіне ауыстырмай орындайды. Біртұтас жүйенің кескіні виртуалдық ішкі жүйемен салыстырғанда, сызығы жуанырақ болады.

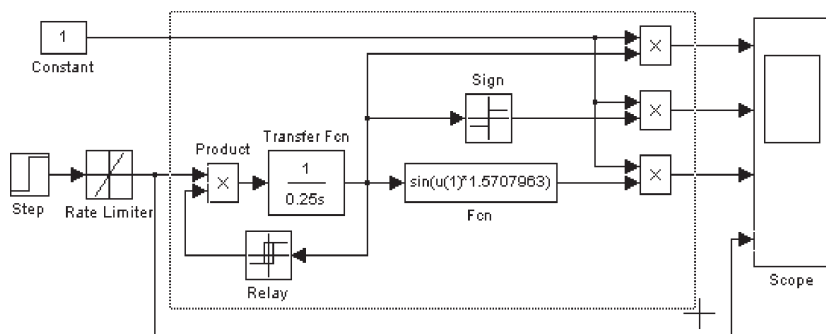
Ішкі жүйелердің бұдан басқа тағы да *басқарылатын* және *басқарылмайтын* түрлері болуы мүмкін. Басқарылатын ішкі жүйелер әрқашан біртұтас болып табылады. Басқарылатын ішкі жүйелерде берілген ішкі жүйені белсендендіретін сигналдар түсетін толықтауыш (басқаратын) кірістері болады. Басқаратын кірістер ішкі жүйенің жоғарғы немесе төменгі жағында орналасқан. Басқарылатын ішкі жүйе белсендендірілген болса–есептеуді орындайды. Ал егер басқарылатын ішкі жүйе бейтарап болса, онда ол есептеуді орындамайды, ал сигнал мәні оның шығысында шығыс портының күйін келтірумен анықталады.

Ішкі жүйенің моделін құру үшін келесі екі тәсілді қолдануға болады:

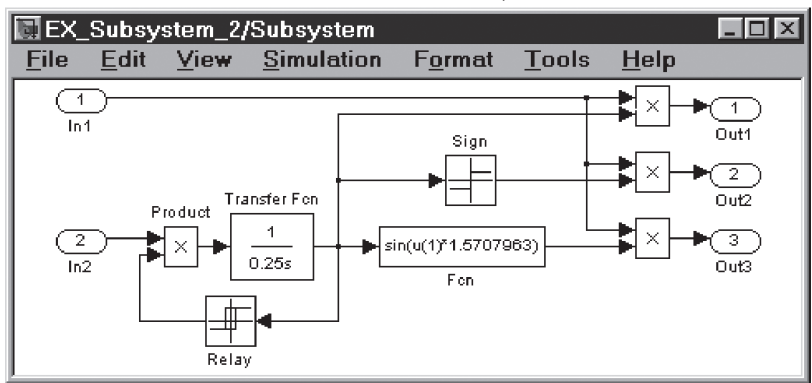
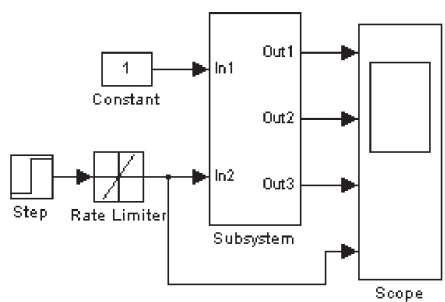
- *Subsystem* кітапханасынан қажетті ішкі жүйені модельге көшіру.

- Модельдің қажетті бөлігін тышқанның көмегімен бөліп алып, модель терезесіндегі *Edit* мәзірінен *Create Subsystem* командасын орындау. Бөлініп алынған бөлік ішкі жүйеге орналастырылады, ал ішкі жүйенің кірістері мен шығыстары сәйкес порттармен жабдыкталады. Берілген тәсіл виртуалдық басқарылмайтын ішкі жүйені құруға мүмкіндік береді. Алдыда, егер қажет болса, ішкі жүйені оның параметрлерін өзгерту арқылы біртұтас етуге немесе кітапханада болатын қажетті ішкі жүйеден басқаратын элемент қосып, басқарылатындай етуге болады. *Undo* командасы арқылы ішкі жүйедегі блоктарды топтастыруды болдырмауға болады.

9.9.1-суреті ішкі жүйенің екінші тәсілмен құрылу үдерісін көрсетеді. 9.9.2-суретте осы үдерістің нәтижесі көрсетілген. Мысалда басқарылатын функционалдық генератордың моделі қолданылған.



9.9.1-сурет. Ішкі жүйенің жасалуы



9.9.2-сурет. Ішкі жүйені қолданатын модель

ҚОЛДАНЫЛҒАН ТЕРМИНДЕРДІҢ ҚЫСҚАША ОРЫСША-ҚАЗАҚША СӨЗДІГІ

А

Активный – белсенді
Апостроф – дәйекше

Б

Библиотека – кітапхана

В

Вариант – нұсқа
Ввод данных – деректерді енгізу, беру
Ввод текста – мәтінді енгізу
Ветвь – тармақ
Вкладка – қосымша бет, қыстырма
Включатель – өшіріп-қосқыш
Вставка – кірістіру
Встроенный – кіріктірілген
Входной сигнал – кіріс сигналы
Вызов – шақыру
Выходной сигнал – шығыс сигналы
Вязкость – тұтқырлық

Г

Глобальные переменные – аумақты айнымалылар
Глобальный – аумақты
Группировка – топтастыру

Д

Данные – берілген деректер
Действие – іс-қимыл
Действительный - нақты
Диалог – сұхбат
Доступ – қолжетімдік дәреже, ену, рұқсат

З

Знак – таңба
Значение – мән, мағына

И

Изоляция – оқшаулау
Инвертирование – кері таңбалау

Интервал – аралык
Информация – ақпарат
Источник сигнала – сигнал көзі

К

Клавиша – перне
Ключ – кілт
Ключевое слово – түйінді сөз
Кнопка – батырма
Колебательное звено – тербелмелі буын
Комментарий – түсініктеме
Комплексно-сопряженный – комплекстік-түйіндес
Конечная разность – шектік айырым
Константа – тұрақты
Крутизна – құламалық, құлдылық
Курсор – мезгегіш

Л

Легенда – әфсана легенда (картаға, сызбаға берілген түсіндірме сөздер)
Локальный – жергілікті

М

Массив – жиым
Меню – мәзір
Метод – әдіс, тәсіл
Многочлен – көпмүше

Н

Наглядность – көрнекілік
Настройка – күйге келтіру

О

Область – аймақ
Обозреватель – шолушы
Окно команд – командалар терезесі
Окно подсказок – еске салу терезесі
Операция – іс-әрекет, амал
Отладчик – дұрыстаушы
Отношение – қатынас
Отображение – көріну, кейіптеу
Отрицание – терістеу
Ошибка – қате

П

Палитра – түс түрлері
Память – жад
Панель инструментов – құралдар жақтауы
Передачная функция – беріліс функциясы
Переменная – айнымалы
Пересечение – қиылысу
Подинтегральная величина – интеграласты өрнек
Подсистемы – ішкі жүйелер
Поле – өріс
Пользователь – қолданушы
Преобразование – түрлендіру
Приоритет – басымдық
Присвоение – меншіктеу
Проблемные ситуации – ахуалды мәселелер
Программа – бағдарлама
Пространство – кеңістік
Процент – пайыз
Процесс – үдеріс
Путь поиска – іздеу жолы

Р

Рабочее пространство – жұмыс кеңістігі, аймағы
Регистрирующие блоки – тіркеуші блоктар
Регулятор – реттегіш

С

Сброс – бастапқы күйге түсіру
Сдвиг – жылжу
Система – жүйе
Сообщение – мәлімет
Состояние – күй
Спираль – шиыршық
Ссылка – сілтеме
Столбец – баған
Стрелка – бағдар
Строка – жол
Строка состояния – күй жолы
Структура – құрылым
Ступенчатое воздействие – сатылық әсер
Сумматор – жиынтықтаушы
Сухой – құрғақ
Схема – сұлба
Счетчик – санағыш

Т

Таблица – кесте
Текст – мәтін
Текущий – ағымдық
Тип данных – деректер типі
Точка останова – тоқтау нүктесі
Транспонирование – тасымалдау

У

Узловая точка – түйіндік нүкте
Управление – басқару
Установка – орнату, қондыру, қою

Ф

Флажок – жалауша
Формат – қалып
Формирование – қалыптасу
Фронт – шеп

Ш

Шаг – адым, қадам

Я

Ячейка – ұяшық

КОЛДАНЫЛҒАН ӘДЕБИЕТТЕР

1. *Алексеев Е.Р., Чеснокова О.В.* MATLAB 7. Самоучитель. “НТ Пресс” 2006.
2. *Ануфриев И., Смирнов А., Смирнов Е.* MATLAB 7. Наиболее полное руководство. – СПб.: БХВ-Петербург, 2005.
3. *Бенькович Е.С.* и др. Практическое моделирование динамических систем. СПб.: БХВ-Петербург, 2002.
4. *В.Кондрашов, С.Королев.* Matlab как система программирования научно-технических расчетов. Мир. 2002.
5. *Веремей Е., Корчанов В., Коровкин М., Погожев С.* Компьютерное моделирование систем управления движением морских подвижных объектов. НИИ Химии СПбГУ. 2002.
6. *Гандер В., Гржебичек И.* Решение задач в научных вычислениях с применением Maple и MATLAB. «Вассамедина» 2005.
7. *Говорухин В., Цибулин В.* Компьютер в математическом исследовании: Maple, MATLAB, LaTeX. Питер. 2001.
8. *Гонсалес Р., Вудс Р., Эддинс С.* Цифровая обработка изображений в среде MATLAB. Москва: Техносфера, 2006.
9. *Гультияев А.* MATLAB 5.3. Имитационное моделирование в среде Windows. Корона принт. 2001.
10. *Гультияев А.* Визуальное моделирование в среде Matlab: Учебный курс. Питер. 2000.
11. *Дьяконов В. П.* Simulink 5/6/7. Самоучитель. ДМК Пресс. 2008.
12. *Дьяконов В.П.* Matlab 6.5 SP1/7 + Simulink 5/6 в математике и моделировании. М.: СОЛОН-Пресс, 2005.
13. *Дьяконов В.П.* Matlab 6.5 SP1/7 + Simulink 5/6. Основы применения. М.: СОЛОН-Пресс, 2005.
14. *Дьяконов В.П., Абраменкова И., Круглов В.* MATLAB с пакетами расширений. Нолидж. 2001.
15. *Дьяконов В.П., Круглов В.* MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник. Питер. 2001.
16. *Дьяконов В.П., Круглов В.* Математические пакеты расширения MATLAB. Специальный справочник. Питер. 2001.
17. *Дэбни Дж., Харман Т.* Simulink 4. Секреты мастерства.: Бином. Лаборатория знаний, 2003.
18. *Иглин С.П.* Математические расчеты на базе Matlab. Издательство «ВНУ-Санкт-Петербург» 2005.
19. *Карлащук В. И.* Электронная лаборатория на IBM PC. Лабораторный практикум на базе Electronics Workbench и MATLAB. СОЛОН-Пресс. 2004.
20. *Кривилев А.* Основы компьютерной математики с использованием системы MATLAB. Лекс-Книга, 2005.

21. Курбатова Е.А. Matlab 7. Самоучитель. – М.: Издательский дом «Вильямс», 2006.

22. Лазарев Ю.Ф. Начала программирования в среде MatLAB: Учебное пособие. - К.: НТУУ «КПИ», 2003.

23. Лазарев, Ю. Моделирование процессов и систем в MatLab: учебный курс / Ю. Лазарев. – СПб.: Питер; Киев: BHV, 2005.

24. Мартынов Н., Иванов А. Matlab 5.x. Вычисления, визуализация, программирование. Кудиц-образ. 2000.

25. Мартынов Н.Н., Иванов А. Matlab 5.x. Пособие по программированию в системе MATLAB. МГУ. 2000.

26. Мартынов Н.Н. Введение в MATLAB 6. Кудиц-образ. 2002.

27. Мэтьюз Д. и Финк К. Численные методы. Использование MATLAB. Вильямс. 2001.

28. Поршнев С. Компьютерное моделирование физических процессов в пакете MATLAB. Горячая Линия - Телеком. 2003.

29. Поршнев С.В. MATLAB 7. Основы работы и программирования. “Бином. Лаборатория знаний” 2006.

30. Потемкин В. Вычисления в среде MATLAB. Диалог-МИФИ. 2004.

31. Чен К., Джоблин П., Ирвинг А. MATLAB в математических исследованиях. Мир. 2001.

32. http://www.tspu.tula.ru/ivt/old_site/lcopy/Matlab_RU/simulink/book1/2.asp.htm. Черных И.В. «Simulink: Инструмент моделирования динамических систем»

33. Черных И.В. SIMULINK: Среда создания инженерных приложений. М.: Диалог-МИФИ.2004.

34. Черных И.В. Моделирование электротехнических устройств в MATLAB, SimPowerSystems и Simulink. 1-е издание, 2007.

35. Штовба С.Д. Проектирование нечетких систем средствами MATLAB.: Горячая Линия - Телеком, 2007.

МАЗМҰНЫ

КІРІСПЕ 4

1 MATLAB ОРТАСЫ. КОМАНДАЛЫҚ РЕЖИМДЕГІ ЖҰМЫС	5
1.1 MATLAB ортасы	5
1.2 Командалық режимдегі жұмыс.....	6
1.3 Математикалық өрнектер	9
1.3.1 Нақты сандар.....	9
1.3.2 Комплекстік сандар.....	9
1.3.3 Айнымалылар.....	10
1.3.4 Жүйелік айнымалылар	11
1.3.5 Арифметикалық амалдар	11
1.3.6 Қатынас амалдары	12
1.3.7 Логикалық амалдар.....	13
1.3.8 Элементар функциялар	14
1.4 Жұмыс аймағы	16
1.5 Күнделік жүргізу.....	19
1.6 Жүйе параметрлерін күйге келтіру	20
1.7 Анықтама жүйесін қолдану	22
2 ЖИЫМДАРМЕН ЖҰМЫС	23
2.1 Векторлар мен матрицалардың қалыптасуы.....	24
2.2 Жиым элементіне сілтеме	27
2.3 «Қос нүкте» операторын қолдану	28
2.4 Берілген жиымдағы жолды және бағанды өшіру	30
2.5 Векторлар мен матрицаларға элементар функцияларды қолдану	31
2.6 Матрицалық және векторлық амалдар	31
2.7 Жиымдарға элементтері бойынша орындалатын амалдар	35
2.8 Жиыммен жұмыс істеу функциялары.....	37
2.9 Сызықтық теңдеулер жүйелерін шешу.....	44
2.10 Жиымдарға қатынас және логикалық амалдарды қолдану- дың ерекшеліктері	45
2.11 Амалдарды орындаудағы басымдықтар	47
2.12 Матрицалармен жұмыс істеудің мысалдары.....	47
3 М-ФАЙЛДАР: СЦЕНАРИЙЛЕР ЖӘНЕ ФУНКЦИЯЛАР	51
3.1 Сценарий.....	52
3.2 Іздеу жолының қондырғысы.....	53
3.3 Функциялар	56

3.4 Аумақты айнымалылар.....	60
3.5 М-файлдың редактор/дұрыстауы. Бағдарламаны дұрыстау	62
4 ГРАФИК ТҰРҒЫЗУ	65
4.1 Екі өлшемдік графиктерді тұрғызу	65
4.1.1 Декарттық координат жүйесінде график тұрғызу	65
4.1.2 Бір терезеде бірнеше график тұрғызу	70
4.1.3 Полярлық координаттар жүйесінде график тұрғызу	70
4.2 Графиктік терезелермен жұмыс	71
4.3 Графиктік терезелердің шағын терезелерге бөлінуі.....	71
4.4 Үш өлшемді графиктің тұрғызылуы.....	72
4.5 Графиктік терезеге жазулар, тақырыптар және координата- лық кесте шығару.....	75
4.6 График масштабын өзгерту.....	78
4.7 Арнайы график.....	79
4.8 Қосымша мысалдар	84
5 БАҒДАРЛАМАЛАУ	89
5.1 Функцияны енгізу және шығару	89
5.2 Циклдік операторлар	91
5.2.1 For циклдік операторы	91
5.2.2 While циклдік операторы	93
5.3 Тармақталған оператор.....	94
5.3.1 Шартты оператор	94
5.3.2 Ауыстырып-қосқыш оператор.....	96
5.4 Басқа операторлар.....	99
5.5 Мысалдар.....	101
6 МАТЛАВ-ТАҒЫ ТИПТІК ЕСЕПТЕР.....	105
6.1 Сызықтық емес теңдеулерді және теңдеулер жүйесіні шешу	105
6.2 Бір немесе бірнеше айнымалысы бар функциялардың минимумын анықтау.....	108
6.3 Сандық дифференциалдау мен шектік айырымды есептеу	111
6.4 Сандық интегралдау	113
6.5 Көпмүшелерге қатысты есептерді шығару	116
6.6 Аппроксимациялау	119
6.7 Интерполяциялау	120
7 МАТЛАВ-ТАҒЫ ДИФФЕРЕНЦИАЛДЫҚ ТЕНДЕУЛЕР ЖҮЙЕСІ.....	124

8 SIMULINK.....	127
8.1 Simulink туралы жалпы деректер	127
8.2 Simulink-ті іске қосу	128
8.3 Simulink кітапханаларының тарауларын қарастыру.....	129
8.4 Модель құрастыру	131
8.5 Модель терезесі.....	134
8.5.1 Модельмен жұмыс істеу.....	135
8.6 Модель құрастыру мен оған өзгерістер енгізудің негізгі амалдары.....	137
8.6.1 Мәтіндік жазбаларды енгізу	137
8.6.2 Объектілерді белгілеу.....	137
8.6.3 Уақытаралық сақтау буферінен объектілерді көшіру және орын ауыстыру	138
8.6.4 Уақытаралық сақтау буферінен объектілерді кірістіру..	138
8.6.5 Объектілерді жою	139
8.6.6 Блоктардың жалғануы.....	139
8.6.7 Блоктардың өлшемдерін өзгерту.....	140
8.6.8 Блоктардың орын ауыстырылуы	141
8.6.9 Undo және Redo командаларының қолданылуы.....	141
8.6.10 Объектілерді қалыптау	142
8.7 Есептеу параметрлерін және оның орындалуын орнату...	143
8.7.1 Модельдің есептеу параметрлерін орнату.....	144
8.7.1.1 Simulation time (Модельдену аралығы немесе есептеу уақыты)	144
8.7.1.2 Solver options (есептеу параметрлері).....	144
8.7.1.3 Output options (шығыс параметрлері)	147
8.7.2 Жұмыс аймағымен алмасу параметрін қою.....	148
8.7.3 Модельді диагноздаудың параметрін орнату	149
8.7.4 Есептеуді жүргізу мен аяқтау	150
9. SIMULINK БЛОГЫНЫҢ КІТАПХАНАСЫ.....	151
9.1 Sources – сигнал көзі.....	151
9.1.1 Constant тұрақты сигнал көзі	151
9.1.2 Sine Wave синусоидалық сигнал көзі.....	151
9.1.2.1 Үздіксіз жүйе үшін ағымдағы уақыт мәні бойынша шығыс сигналының түзілуі.....	152
9.1.2.2 Дискреттік жүйе үшін ағымдағы уақыт мәні бойынша шығыс сигналының түзілуі.....	152

9.1.2.3 Модельдеу уақытының шамасы мен бір периодтағы есептеу адымының саны бойынша шығыс сигналының түзілуі.....	153
9.1.3 Ramp сызықтық өзгерістегі әсер көзі.....	154
9.1.4 Step сатылық сигнал генераторы.....	155
9.1.5 Signal Generator сигналдар генераторы	155
9.1.6 Uniform Random Number біркелкі таратылатын кездейсоқ сигнал көзі	156
9.1.8.7 Random Number қалыпты таратылатын кездейсоқ сигнал көзі.....	157
9.1.8 Pulse Generator импульстік сигнал көзі	158
9.1.9 Chirp Generator жиілігі сызықтық өзгертін генератор..	159
9.1.10 Band-Limited White Noise ақ шуыл генераторы.....	159
9.1.11 Clock уақыттық сигнал көзі	160
9.1.12 Digital Clock сандық уақыт көзі.....	161
9.1.13 From File файлынан берілгендерді оқу блогы.....	161
9.1.14 From Workspace жұмыс кеңістігінен берілгендерді оқу блогы	163
9.1.15 Ground нөлдік деңгейдегі сигналдар блогы	164
9.1.16 Repeating Sequence периодтық сигналдар блогы.....	164
9.1.17 Inport кіріс портының блогы.....	165
9.1.17.1 Ішкі жүйелерде Inport блогын қолдану.....	166
9.1.17.2 Жоғары дәрежедегі модельде Inport блогын қолдану	167
9.2 Sinks – сигнал қабылдаушылар	168
9.2.1 Scope осциллографы.....	168
9.2.2 Floating Scope осциллографы	174
9.2.3 XY Graph сызба құрастырушысы.....	176
9.2.4 Display сандық дисплейі	177
9.2.5 Stop Simulation модельдеуді тоқтату блогы.....	179
9.2.6 Деректерді To File файлында сақтау блогы.....	180
9.2.7 Деректерді To Workspace жұмыс аймағында сақтау блогы	181
9.2.8 Terminator соңғы қабылдағышы	182
9.2.9 Outport шығыс портының блогы	182
9.2.9.1 Ішкі жүйелерде Outport блогын қолдану.....	183
9.2.9.2 Outport блогын жоғары деңгейдегі модельде қолдану	185
9.3 Continuous – аналогтық блоктар.....	186

9.3.1 Derivative туынды есептеу блогы.....	186
9.3.2 Integrator интегралдаушы блогы.....	187
9.3.3 Memory блогы	192
9.3.4 Transport Delay сигналдың бекітілген бөгеу блогы	193
9.3.5 Variable Transport Delay бөгеуі басқарылатын сигналдың блогы	194
9.3.6 Transfer Fcn блогының функциясы.....	195
9.3.7 Zero-Pole беріліс функциясының блогы	198
9.3.8 State-Space динамикалық объекті моделінің блогы.....	199
9.4 Discrete – дискреттік блоктар	200
9.4.1 Unit Delay жеке дискреттік бөгеу блогы.....	200
9.4.2 Zero-Order Hold нөлінші реттік экстраполятордың блогы.....	201
9.4.3 First-Order Hold бірінші реттік экстраполятордың блогы	203
9.4.4 Discrete-Time Integrator дискретті интегратордың блогы.....	204
9.4.5 Discrete Transfer Fcn дискреттік беріліс функциясы.....	205
9.4.6 Discrete Zero-Pole дискреттік беріліс функциясының блогы	207
9.4.7 Discrete Filter дискреттік сүзгі блогы.....	208
9.4.8 Discrete State-Space динамикалық объекті моделінің блогы	209
9.5 Nonlinear – сызықсыз блоктар	210
9.5.1 Saturation шектеу блогы	210
9.5.2 Dead Zone сезімсіздік аймақ блогы.....	212
9.5.3 Relay релелік блогы	213
9.5.4 Rate Limiter сигналдың өзгеру жылдамдығын шектеу блогы	215
9.5.5 Quantizer деңгей бойынша кванттау блогы	216
9.5.6 Coulomb and Viscous Friction құрғақ және тұтқыр үйкеліс блогы	217
9.5.7 Backlash люфт блогы	218
9.5.8 Switch өшіріп-қосқыш блогы.....	219
9.5.9 Multiport Switch көпкірісті өшіріп-қосқыш блогы	219
9.5.10 Manual Switch қолмен өшіріп-қосқыш блогы	221
9.6 Math – математикалық амалдар блогы.....	222
9.6.1 Abs модуль есептеу блогы.....	222
9.6.2 Sum қосындыны есептеу блогы	223

9.6.3 Product көбейту блогы	224
9.6.4 Sign сигнал таңбасын анықтау блогы	226
9.6.5 Gain және Matrix Gain күшейткіштері	227
9.6.6 Slider Gain жылжымалы реттегіш	229
9.6.7 Dot Product скалярлық көбейту блогы	230
9.6.8 Math Function математикалық функцияларды есептеу блогы	231
9.6.9 Trigonometric Function тригонометриялық функцияларды есептеу блогы	233
9.6.10 Complex to Real-Imag комплекстік санның нақты және жорамал бөлігін есептеу блогы	234
9.6.11 Complex to Magnitude-Angle комплекстік санның аргументін және (немесе) модулін есептейтін блок....	235
9.6.12 Real-Imag to Complex комплекстік санды оның нақты және жорамал бөлігі бойынша есептеу блогы	236
9.6.13 Magnitude-Angle to Complex комплекстік санды оның модулі мен аргументі бойынша есептеу блогы.....	237
9.6.14 MinMax минималдық немесе максималдық мәнді анықтайтын блок.....	238
9.6.15 Rounding Function сандық мәнді дөңгелектеу блогы ...	239
9.6.16 Relational Operator қатынас амалдарын есептеу блогы	241
9.6.17 Logical Operation логикалық амалдар блогы.....	242
9.6.18 Bitwise Logical Operator биттік логикалық амалдар блогы	244
9.6.19 Combinatorial Logic комбинаторикалық логика блогы.....	245
9.6.20 Algebraic Constraint алгебралық контурының блогы	246
9.7 Signal&Systems – сигналдарды өңдеу блоктары мен көмекші блоктар	248
9.7.1 Mux мультиплексоры (араластырғыш)	248
9.7.2 Demux демультимплексоры (бөлгіш)	249
9.7.3 Bus Creator шиналық құраушының блогы.....	251
9.7.4 Bus Selector шиналық селектор блогы	253
9.7.5 Selector селектор блогы	254
9.7.6 Assignment жиым элементтеріне жаңа мәндер беру блогы	256
9.7.7 Merge сигналдарды біріктіру блогы.....	258
9.7.8 Matrix Concatenation сигналдарды матрицаға біріктіру блогы	260

9.7.9 Goto сигналды беру блогы	260
9.7.10 From сигнал қабылдау блогы.....	261
9.7.11 Goto Tag Visibility сигналдың көріну белгілерінің блогы	262
9.7.12 Data Store Memory жадтың ортақ аймағын құру блогы	263
9.7.13 Data Store Write деректерді жадтың ортақ аймағына жазу блогы	264
9.7.14 Блок Data Store Read деректерді жадтың ортақ аймағынан оқу блогы	264
9.7.15 Data Type Conversion сигнал типін түрлендіру блогы..	265
9.7.16 Reshape сигнал өлшемін түрлендіру блогы.....	266
9.7.17 Width сигналдың өлшемін анықтау блогы	267
9.7.18 Hit Crossing шектік мәнің қиылысу сәтін анықтайтын блок	268
9.7.19 IC сигналдың бастапқы мәнін қою блогы	269
9.7.20 Signal Specification сигналды тексеру блогы.....	270
9.7.21 Probe сигналдың қасиеттерінің датчигі	271
9.7.22 Function-Call Generator итерация санын беретін блок..	272
9.7.23 Model Info ақпараттық блогы.....	273
9.8 Function & Tables – функциялар мен кестелердің блоктары	274
9.8.1 Fcn функцияны беру блогы.....	274
9.8.2 MATLAB Fcn функцияны беру блогы	276
9.8.3 Polynomial дәрежелік көпмүшені беру блогы	277
9.8.4 Look-Up Table бір өлшемді кесте блогы.....	278
9.8.5 Look-Up Table(2D) екі өлшемді кесте блогы.....	279
9.8.6 Look-Up Table (n-D) көпөлшемді кестенің блогы.....	280
9.8.7 Direct Loop-Up Table (n-D) тікелей енуі бар кестенің блогы	282
9.8.8 PreLook-Up Index Search индекстермен жұмыс істеу блогы	284
9.8.9 Interpolation (n-D) using PreLook-Up кестелік функцияны интерполяциялау блогы.....	286
9.9 Subsystem – ішкі жүйелер	287
Қолданылған терминдердің қысқаша орысша-қазақша сөздігі	291
Қолданылған әдебиеттер.....	295
Мазмұны	297

Қ. Ә. Өжікенов

**ЖҮЙЕЛЕРДІ МОДЕЛЬДЕУДІҢ БАҒДАРЛАМАЛЫҚ
ҚҰРАЛДАРЫ (MATLAB/SIMULINK)**

Оқулық

Басуға 21.12.2012 қол қойылды. Қағазы офсеттік. Қаріп түрі «Times».
Пішіні 60x90/16. Офсеттік басылым. Баспа табағы 19,0.
Таралымы 1100 дана. Тапсырыс № 1242.

Тапсырыс берушінің дайын файлдарынан басылып шықты.



ЖШС РПБК «Дәуір», 050009,
Алматы қаласы, Гагарин д-лы, 93а.
E-mail: rpik-dauir81@mail.ru