**Դաս 1**

**1. Մոդելավորման ներածություն**

Մոդելավորումը օբյեկտի (համակարգի,կառույցի, պրոցեսի) հետազոտումն է ՙ դրա մոդելի միջոցով, մոդելի կառուցումը և ուսումնասիրումը այդ օբյեկտի մասին նոր գիտելիք ստանալու նպատակով։ Մոդելավորումը իմացաբանական կատեգորիա է՝ ճանաչողության կարևորագույն հնարներից ու մեթոդներից։  
Որպես իրակասության արտացոլման եղանակ, մոդելավորումը առաջացել է հին աշխարհում և կատարելագործվել գիտական առաջադիմությանը զուգընթաց։

Մոդելավորումը լայնորեն կիրառվել է Վերածննդի դարաշրջանում․ Բրունելլեսկին, Միքելանջելոն և ուրիշներ օգտվում էին կառուցվող օբյեկտների մոդելներից։ Գ. Գալիլեյի և Լեոնարդո դա Վինչիի տեսական հետազոտություններում տրված են մոդելավորման կիրառման սահմանները։ XIX դարի կեսից, պայմանավորված բնագիտության և տեխնիկայի բուռն զարգացմամբ, զգալիորեն ընդլայնվել է մոդելավորման կիրառությունը։ Մոդելավորումը դասակարգում են ըստ մոդելների, մոդելավորման օբյեկտների ու միջոցների, մոդելների կիրառման բնագավառների են։

Մոդելավորումը, կոչվում է առարկայական, եթե հետազոտությունը կատարվում է այնպիսի մոդելով, որը վերարտադրում է օբյեկտի հիմնական երկրաչափական, ֆիզիկական, դինամիկական և ֆունկցիոնալ բնութագրերը։ Կիրառվում է նշանային մոդելավորումը, երբ որպես մոդելներ օգտագործվում են գծագրեր, բանաձևեր, հավասարումներ, որևէ լեզվի այբուբենով գրված բառեր, նախադասություններ են։ Նշանային համակարգերն ու դրանց տարրերը դիտարկվում են միասնաբար որոշակի գործողությունների ու ձևափոխությունների հետ, որ կատարում է մարդը կամ մեքենան։ Որոշ հանգամանքներում նշանային մոդելավորումը իրականացվում է մտային-ակնառու պատկերման միջոցով և հավակնում մտովի՝ զգայական-ակնառու ձևերով պատկերել ուսումնասիրվող օբյեկտի կառուցվածքը, տարրերի կապն ու փոխազդեցությունը (օրինակ, գազերի կինետիկ տեսության՝ Մաքսվելի, ատոմի մոլորակային և այլ մոդելներ)։

Պատկերային-մտային մոդելավորումը ճանաչողության պրոցեսի անհրաժեշտ պայմանն է նրա ձևավորման շրջանում։ Այն հնարավորություն է տալիս սովորական (անմիջական) գիտափորձը փոխարինել մոդելային գիտափորձով։ Մոդելային գիտափորձի հատուկ տեսակ է մտային գիտափորձը, երբ հետազոտողը մտովի գործառում է նշանային (հիմնականում՝ պատկերային) մոդելներով։ Մտային գիտափորձի ճանաչողական նշանակությունն ակնհայտ է դառնում այն դեպքում, երբ անհնար է իրական փորձ կատարել կամ էլ իրագործել առարկայական(առարկայական մաթեմատիկական) մոդելավորումը։ Հայտնի է մտային գիտափորձի ճանաչողական դերը քվանտային մեխանիկայում, հարաբերականության տեսության մեջ և այլուր։  
թեմատիկայում մոդելավորման էությունն այն է, որ հետազոտության օբյեկտի մասին հայտնի փաստերը արտապատկերում են ինչ-որ հարաբերությամբ օբյեկտին իզոմորֆ (կամ հոմոմորֆ) որևէ մաթեմատիկական ձևով (բանաձև, դիֆերենցիալ հավասարում, բազմություն, խումբ են), իսկ օրինաչափությունների հետագա իմացությունը կատարվում է այդ ձևի (մոդելի) ձևափոխության ու վերլուծության միջոցով։ Գիտության տարբեր բնագավառներում մաթեմատիկայում մոդելավորման հնարավորությունները տարբեր են։

Մեխանիկայում, օպտիկայում,էլեկտրադինամիկայում են, որպես կանոն, կառուցվում են ուսումնասիրվող երևույթի ճշգրիտ մաթեմատիկական մոդելներ։ Իսկ քիմիայում, տնտեսագիտությունում, կենսաբանությունում հիմնական օրենքները մաթեմատիկական մոդելավորման չեն ենթարկվել, սակայն այն կարևոր դեր է խաղում մի շարք հարցեր ուսումնասիրելիս։

Սոցիոլոգիայում, հոգեբանությունում, մանկավարժությունում և հասարակական այլ գիտություններում մաթեմատիկական մոդելավորումը դեռևս գտնվում է ձևավորման փուլում։ Մաթեմատիկան (տրամաբանական) մոդելավորման լայն կիրառություն ունի նաև կիբեռնետիկայում և հաշվողական տեխնիկայում։ Մաթեմատիկական մոդելավորումը պետք է տարբերել մաթեմատիկայում կիրառվող մոդելավորումից։ Այստեղ առանձնապես կարևոր են մեկնաբանող մոդելները, որոնց հատկությունները ուսումնասիրում է մոդելների տեսությունը։ Այդ տեսությամբ ուսումնասիրվում են նաև աքսիոմացված դասերի ընդհանուր հատկությունները, այն կիրառություն է գտել մաթեմատիկայի մյուս ճյուղերում նույնպես։

Հաճախ որպես մեկնաբանող մոդել հանդես է գալիս օբյեկտների համախումբը, որոնց հատկությունները և դրանց միջև հարաբերությունները բավարարում են աքսիոմների տվյալ համակարգին։ Բացառված չէ, որ աքսիոմների համակարգի կամ տեսության մոդել հանդես գա մաթեմատիկական այլ տեսություն, որի նույնականությունը ապացուցված է գործնականում (օրինակ, Լոբաչևսկու երկրաչափության է․ Բելտրամիի ու Ֆ․ Կլայնի մոդելները)։

Մաթեմատիկական տրամաբանության մեջ որևէ բովանդակալից տեսության մոդել համարվում է այն ձևական համակարգը (հաշիվը), որի մեկնաբանումը այդ տեսությունն է։ Համանման բնույթ ունի մոդելի օգտագործումը լեզվաբանության մեջ։ Լեզվաբանական մոդելները կարևոր դեր են խաղում ինչպես տեսական լեզվաբանական հետազոտություններում (լեզվաբանական հասկացությունների ու դրանց միջև կապերի ճշտում, կառուցվածքների բացահայտում, որոնք կան լեզվական երևույթների անսահման բազմազանության մեջ են), այնպես էլ ինֆորմացիոն լեզուների կառուցման, մեքենայական թարգմանության մշակման և այլ խնդիրների լուծման գործում։ Կիբեռնետիկական մոդելավորումը սովորաբար իրացվում է ընդհատ կամ անընդհատ գործողության հաշվողական մեքենաների միջոցով և կիրառվում է հիմնականում բարդ համակարգերի հետազոտության համար։ Այս դեպքում որևէ պրոցեսի կամ երևույթի մոդել համարվում է ինչպես դրա գործողության ալգորիթմը, այնպես էլ էլեկտրոնային հաշվողական մեքենան՝ համապատասխան ծրագրավորումից հետո։

Կիբեռնետիկական մոդելավորման դեպքում սովորաբար վերացարկվում են համակարգի կառուցվածքից և այն դիտարկում որպես «սև արկղ»։ Կիբեռնետիկական մոդելավորման կարևոր առանձնահատկությունը մտային գործունեության մոդելավորում է, որը մարդուն անսահմանափակ հնարավորություններ է տալիս։ Մինչև 1940-ական թթ․ վերջը, երբ ստեղծվեցին թվանշանային հաշվողական մեքենաները, համակարգերի ուսումնասիրությունը հիմնականում կատարվում էր անալոգային հաշվողական մեքենաներով։ Ներկայումս անալոգային մոդելավորման դերը նվազել է, քանի որ թվանշանային հաշվողական մեքենաներով մոդելավորումը գերադասելի է ճշտության և համընդհանրության տեսակետից՛։

Ներկայումս մոդելավորումը հաջողությամբ կիրառվում է կենդանի և անկենդան բնության հետազոտություններում, տեխնիկայի ամենատարբեր բնագավառներում, մարդու և հասարակության մասին գիտություններում։

Հատկապես տարածում են գտել ֆիզիկական մոդելավորում, մոդելավորում կենսաբանության մեջ, տնտեսությունում:

Մաթեմատիկական մոդելավորում, նշանային մոդելավորման մի տեսակ, գիտական հետազոտության մեթոդներից։ Մաթեմատիկական մոդելավորման էությունն այն է, որ հետազոտության օբյեկտի մասին հայտնի փաստերը արտապատկերում են ինչ-որ հարաբերությամբ օբյեկտին իզոմորֆ (կամ հոմոմորֆ) որևէ մաթեմատիկական ձևով (բանաձևեր, դիֆերենցիալ հավասարում, բազմություն, խումբ և այլն), իսկ օրինաչափությունների հետագա իմացությունը կատարվում է այդ ձևի (մոդելի) ձևափոխության ու վերլուծության միջոցով։  
Գիտության տարբեր բնագավառներում մաթեմատիկական մոդելավորման հնարավորությունները տարբեր են։ մեխանիկաում, օպտիկայում,էլեկտրադ ինամիկայում և այլն, որպես կանոն, կառուցվում են ուսումնասիրվող երևույթի ճշգրիտ մաթեմատիկական մոդելներ։ Իսկ քիմիայում, տնտեսագիտությունում, կենսաբանությունում հիմնական օրենքները մաթեմատիկական մոդելավորման չեն ենթարկվել, սակայն այն կարևոր դեր է խաղում մի շարք հարցեր ուսումնասիրելիս։ Սոցիոլոգիայում, հոգեբանությունում, մանկավարժությունում և հասարակական այլ գիտություններում մաթեմատիկական մոդելավորումը դեռևս գտնվում է ձևավորման փուլում։  
Մաթեմատիկական (տրամաբանական) մոդելավորումը լայն կիրառություն ունի նաև կիբեռնետիկայում և հաշվողական տեխնիկայում։ Մաթեմատիկական մոդելավորումը պետք է տարբերել մաթեմատիկայում կիրառվող մոդելավորումից։ Այստեղ առանձնապես կարևոր են մեկնաբանող մոդելները, որոնց հատկությունները ուսումնասիրում է մոդելների տեսությունը։ Այդ տեսությամբ ուսումնասիրվում են նաև աքսիոմացված դասերի ընդհանուր հատկությունները, այն կիրառություն է գտել մաթեմատիկայի մյուս ճյուղերում նույնպես։ Հաճախ որպես մեկնաբանող մոդել հանդես է գալիս օբյեկտների համախումբը, որոնց հատկությունները և դրանց միջև հարաբերությունները բավարարում են աքսիոմների տվյալ համակարգին։ Բացառված չէ, որ |աքսիոմների համակարգի կամ տեսության մոդել հանդես գա մաթեմատիկական այլ տեսություն, որի նույնականությունը ապացուցված է գործնականում (օրինակ, Լոբաչևսկու երկրաչափություն, Բելտրամիի ու Ֆ․ Կլայնի մոդելները)։

Մաթեմատիկական տրամաբանության մեջ որևէ բովանդակալից տեսության մոդել համարվում է այն ձևական համակարգը (հաշիվը), որի մեկնաբանումը այդ տեսությունն է։ Համանման բնույթ ունի մոդելի օգտագործումը լեզվաբանության մեջ։ Լեզվաբանական մոդելները կարևոր դեր են խաղում ինչպես տեսական լեզվաբանական հետազոտություններում (լեզվաբանական հասկացությունների ու դրանց միջև կապերի ճշտում, կառուցվածքների բացահայտում, որոնք կան լեզվական երեվույթների անսահման բազմազանության մեջ և այլն), այնպես էլ ինֆորմացիոն լեզուների կառուցման, մեքենայական թարգմանության մշակման և այլ խնդիրների լուծման գործում։

Իրական օբյեկտի իդեալականացումը կոչվում է մոդելավորում, իսկ իդեալական օբյեկտն ինքը կոչվում է մոդել։

Մոդելավորումը մոդել կառուցելու պրոցեսն է, ինչը անհրաժեշտ է տարաբնույթ խնդիրներ լուծելու համար։ Մոդելը ստեղծվում է ելնելով տվյալ խնդրի պայմաններից։

Օբյեկտները կարող են լինել բազմազան և առընչվել տարբեր բնագավառների հետ։ Այժմ մոդելավորումը լայնորեն կիրառվում է տնտեսության և գիտության մի շարք ճյուղերում։ Գիտության մեջ „Կիրառական մաթեմատիկա” առարկան սկսվում է մաթեմատիկական մոդելների կառուցումով։ Դա նշանակում է, որ որևէ երևույթ կամ օբյեկտ ուսումնասիրելիս հաշվի է առնվում նրան բնորոշ առանձնահատկությունները, անտեսվում են նրա երկրորդային հատկությունները, և առաջնային հատկությունները բերվում են մաթեմատիկական տեսքի։ Բերել մաթեմատիկական տեսքի նշանակում է գրել այդ օբյեկտի մաթեմատիկական հավասարումները և անհրաժեշտ պայմանները։ Հաջորդ քայլը այդ հավասարումների լուծումն է կամ թվային իրականացումը։ Մաթեմատիկական մոդելը արտացոլում է տվյալ օբյեկտի գլխավոր բնորոշ հատկությունները և ամենևին էլ նույնական չէ այդ օբյեկտին։ Դա նշանակում է, որ տվյալ խնդրի պայմաններում այդ օբյեկտի որոշ հատկություններ մենք անտեսում ենք, կամ օբյեկտն ունի այնպիսի բնորոշ հատկություն, որը մեզ անհայտ է և կարող է բացահայտվել հետագայում։ Մաթեմատիկական մոդելների օրինակներ են՝ նյութական կետը, մաթեմատիկական ճոճանակը, ճկուն առաձգական թելը, իդեալական գազը, անդեֆեկտ նյութը և այլն։  
Մաթեմատիկական մոդելը ճշգրիտ չի համընկնում ուսումնասիրվող օբյեկտի հետ և բացի դրանից հնարավոր է որ օբյեկտի մասին մենք չունենք լրացուցիչ տեղեկություններ և մեզ անհրաժեշտ է լինում այդ մոդելը կիրառել պարամետրերի այլ արժեքների դեպքում։ Այդ դեպքում հնարավոր է, որ այդ մոդելը դադարի ճիշտ լինելուց։ Եթե օբյեկտի մասին ունենք քիչ ինֆորմացիա, այդ դեպքում օբյեկտի մասին անում ենք որոշակի վարկածներ, որոնց շրջանակներում է կառուցվում մաթեմատիկական մոդելը։ Այնուհետև փորձնական ճանապարհով ուսումնասիրվում է մոդելի ճշտությունը։ Եթե փորձարարական հետազոտողների տվյալները մոտիկ են կամ համընկնում են կառուցված մոդելով ստացված արդյունքների հետ, ապա այդպիսի մոդելը հաստատվում է և ստանում է գոյության իրավունք։ Ամեն մի մաթեմատիկական մոդել ունի իր ճշտության սահմանը։ Մաթեմատիկական մոդելի ճշտության աստիճանը բարձրացնելիս մոդելը հետզհետե բարդանում է։  
**Մոդելի և մոդելավորման հասկացությունները**

Գիտական ​​հետազոտություններում մոդելավորումը սկսեց օգտագործվել հին ժամանակներից և աստիճանաբար գրավել գիտական ​​ նոր ոլորտներ՝ տեխնիկական ձևավորում, շինարարություն և ճարտարապետություն, աստղագիտություն, ֆիզիկա, քիմիա, կենսաբանություն և վերջապես, սոցիալական գիտություններ: Մեծ հաջողությունն ու ճանաչումը ժամանակակից գիտության գրեթե բոլոր ճյուղերում բերեցին մոդելավորման նոր մեթոդների: Այնուամենայնիվ, մոդելավորման մեթոդաբանությունը վաղուց մշակվել է անհատական ​​գիտությունների կողմից միմյանցից անկախ: Չկար գաղափարների մի համակարգ մի տերմինաբանություն: Միայն աստիճանաբար ճանաչվեց մոդելավորման դերը՝ որպես գիտական ​​գիտելիքների համընդհանուր մեթոդ:

Մոդել» տերմինը լայնորեն օգտագործվում է մարդու գործունեության տարբեր բնագավառներում և ունի շատ իմաստաբանական իմաստներ: Այս բաժնում մենք կքննարկենք միայն այն մոդելները, որոնք գիտելիքներ ձեռք բերելու գործիք են:

Մոդելը այնպիսի նյութ է կամ մտավոր ներկայացված, այսինքն ՝ տեղեկատվական օբյեկտ, որը հետազոտության գործընթացում փոխարինում է բնօրինակը, տիրապետում է իր հիմնական տեղեկատվական հատկություններին (որակապես տրամաբանական և քանակականորեն մաթեմատիկական), այսինքն՝ ուսումնասիրված օբյեկտի տարրերի և դրա հարաբերությունների միջև փոխհարաբերությունների բնույթը: Ֆիզիկական իրականության այլ առարկաներ, այնպես որ մոդելի ուսումնասիրությունը տալիս է նոր գիտելիքներ բնօրինակ օբյեկտի մասին: Ավելի խստորեն, ըստ էության, մոդելը տեղեկատվական համակարգի մի տեսակ է, որը պատճենում է թիրախային համակարգերը (տեղեկատվություն, էներգիա, նյութեր), և նախատեսված է ուսումնասիրել վերջինիս հատկությունները: Ձևաչափով՝ մոդելը կարող է պարզեցվել ցանկացած ֆիզիկական միջավայրի վրա նյութական արտադրանք, համակարգչային ծրագիր:

**Մոդելավորումը** - մոդելների կառուցման, ուսումնասիրման և կիրառման գործընթաց է: Այն սերտորեն կապված է այնպիսի կատեգորիաների հետ, ինչպիսիք են աբստրակցիան, անալոգիան, վարկածը և այլն: Մոդելավորման հիմնական առանձնահատկությունն այն է, որ դա անուղղակի ճանաչման մեթոդ է օգտագործելով փոխարինող օբյեկտներ:

Մոդելը գործում է որպես մի տեսակ ճանաչողական գործիք, որը հետազոտողը դնում է իր և օբյեկտի միջև, և որի օգնությամբ նա ուսումնասիրում է իրեն հետաքրքրող առարկան: Դա մոդելավորման մեթոդի այն առանձնահատկությունն է, որը որոշում է աբստրակցիաների, անալոգիաների, վարկածների, այլ կատեգորիաների և ճանաչման օգտագործման հատուկ ձևերը:

Ամենատարածված դեպքում՝ մոդել կառուցելիս, հետազոտողը հեռացնում է այդ բնութագրերը, պարամետրերը, որոնք էական չեն օբյեկտի ուսումնասիրության համար: Բնօրինակ օբյեկտի բնութագրերի ընտրությունը, որոնք պահպանվում են և ընդգրկվելու են մոդելի մեջ, որոշվում է մոդելավորման նպատակներով: Սովորաբար, օբյեկտի ոչ էական պարամետրերից վերացվելու նման գործընթացը կոչվում է ֆորմալիզացիա: Ավելի ճիշտ ֆորմալացում:

**ֆորմալացումը** - իրական օբյեկտի կամ գործընթացի փոխարինումն է իր ֆորմալ նկարագրությամբ:

Մոդելների հիմնական պահանջը նրանց համարժեքությունն է իրական գործընթացներին կամ առարկաներին, որոնք մոդելը փոխարինում է:

Բնության գրեթե բոլոր գիտություններում գրեթե բոլոր գիտություններում, հասարակության մասին, մոդելների կառուցումն ու օգտագործումը գիտելիքների հզոր գործիք է: Իրական առարկաներն ու գործընթացներն այնքան բազմաբնույթ են և բարդ, որ դրանց ուսումնասիրման լավագույն (և երբեմն միակ) եղանակը հաճախ կառուցել և ուսումնասիրել այնպիսի մոդել, որը ցույց է տալիս իրականության միայն որոշ կողմեր ​​և հետևաբար, շատ ավելի պարզ է, քան այս իրականությունը: Գիտության զարգացման բազմադարյան փորձը գործնականում ապացուցեց այս մոտեցման պտղաբերությունը: Ավելի կոնկրետ՝ մոդելավորման մեթոդի օգտագործման անհրաժեշտությունը որոշվում է նրանով, որ շատ օբյեկտներ (կամ այդ օբյեկտների հետ կապված խնդիրներ) ընդհանրապես չեն կարող ուղղակիորեն ուսումնասիրվել, կամ այս ուսումնասիրությունը չափազանց շատ ժամանակ և միջոցներ է պահանջում:

Մոդելավորման հարցում կա երկու տարբեր մոտեցում: Սա լիարժեք և վերացական մոդելավորում է:

**Լիարժեք սանդղակը, ֆիզիկական մոդելը մոդել** է այլ նյութից պատրաստված օբյեկտի պատճենը, այլ սանդղակով, մի շարք մանրամասների բացակայությամբ: Օրինակ՝ սա խաղալիքի նավակ է, խորանարդի տուն, օդանավերի կառուցման մեջ օգտագործված օդանավի լրիվ չափի փայտե մոդել և այլն:

**Վերացական մոդելը, տեղեկատվական մոդելը** այն մոդելն է, որը իրականությունը ցուցադրում է ոչ թե նյութական, այլ տեղեկատվական կապերով բանավոր նկարագրություն, որը ձևակերպվում է որոշ կանոնների համաձայն, մաթեմատիկական հարաբերություններով և այլն:

**Համակարգչային մոդելի հասկացությունը**

Համակարգչային մոդելը համակարգչային ծրագիր է, որն աշխատում է առանձին համակարգչի, գերհամակարգիչի կամ փոխազդող համակարգիչների (հաշվողական հանգույցների) մի շարք, որը իրականացնում է վերացական, այսինքն՝ ինչ-որ համակարգի տեղեկատվական մոդել: Համակարգչային մոդելները դարձել են թվային և մաթեմատիկական մոդելավորման սովորական գործիք և օգտագործվում են ֆիզիկայի, աստղաֆիզիկայի, մեխանիկայի, քիմիայի, կենսաբանության, տնտեսագիտության, սոցիոլոգիայի, օդերևութաբանության, այլ գիտությունների և կիրառական խնդիրների մեջ՝ ռադիոէլեկտրոնիկայի, մեքենաշինության, ավտոմոբիլային արդյունաբերության և այլ ոլորտներում: Համակարգչային մոդելները օգտագործվում են մոդելավորված օբյեկտի վերաբերյալ նոր գիտելիքներ ձեռք բերելու կամ տրամաբանական-վերլուծական հետազոտությունների համար չափազանց բարդ համակարգերի վարքագծի հետազոտման համար:

Համակարգչային մոդելավորումը բարդ համակարգերի ուսումնասիրման արդյունավետ մեթոդներից մեկն է: Համակարգչային մոդելները ավելի հեշտ և հարմար են ուսումնասիրություն համար, այլ կերպ ասած, «հաշվարկային փորձեր» իրականացնելու համար, որոնք իրականում փորձեր չեն, քանի որ տեղեկատվությունը վերցված է ոչ թե ֆիզիկական փորձից, ֆիզիկական իրականությունից, այլ դրա դեպքերի մոդելային իրականությունից, այն դեպքում, երբ իրական փորձերը դժվար են ֆինանսական կամ ֆիզիկական խոչընդոտների պատճառով կամ կարող են անկանխատեսելի վտանգավոր արդյունքներ բերել: Համակարգչային մոդելների ստեղծման փուլում ճիշտ տրամաբանության և ճիշտ ձևակերպման դեպքում հնարավոր է բացահայտել հիմնական գործոնները, որոնք որոշում են ուսումնասիրված բնօրինակ օբյեկտի (կամ առարկաների մի ամբողջ դասի) քանակական հատկությունները, մասնավորապես՝ ուսումնասիրել մոդելավորված ֆիզիկական համակարգը դրա պարամետրերի և սկզբնական պայմանների ազդեցությունները:

Համակարգչային մոդելի կառուցումը հիմնված է երևույթների կամ կոնկրետ օբյեկտի ուսումնասիրմանհամար և այն բաղկացած է երկու փուլից` նախ ստեղծելով որակապես տրամաբանական, իսկ հետո քանակականորեն մաթեմատիկական մոդել: Մյուս կողմից, համակարգչային մոդելավորումը բաղկացած է համակարգչում մի շարք «հաշվարկային փորձեր» անցկացնելուց, որի նպատակը մոդելի ներքին վարքի վերլուծությունն է և քանակական տվյալներ ձեռք բերելը մոդելի վարքի վերաբերյալ:

**Թվային մաթեմատիկական մոդելավորման հասկացությունը**

Թվային մաթեմատիկական մոդելը օբյեկտի կամ գործընթացի հատկությունները բնութագրող մաթեմատիկական հավասարումներն են: Փաստորեն, մաթեմատիկան ինքնին պարտական ​​է իր գոյությանը այն, ինչ փորձում է հաշվարկել, այսինքն ՝ իր հատուկ լեզվով քանակականորեն արտացոլել շրջապատող աշխարհի քանակական օրենքները: Թվային և մաթեմատիկական մոդելավորման զարգացման հսկայական ազդակ տրվեց համակարգիչը, չնայած որ հաշվարկման մեթոդները ինքնին սկիզբ են առնում մաթեմատիկայից:

Թվային և մաթեմատիկական մոդելավորումը որպես այդպիսին միշտ չէ, որ պահանջում է համակարգչային աջակցություն: Յուրաքանչյուր մասնագետ, ով մասնագիտորեն զբաղվում է թվային և մաթեմատիկական մոդելավորմամբ, անում է հնարավոր ամեն ինչ մոդելի վերլուծական ուսումնասիրության համար: Վերլուծական լուծումները (այսինքն, ներկայացված են տրամաբանական կոնստրուկցիաներով, որոնք արտահայտում են ուսումնասիրության արդյունքները նախնական տվյալների միջոցով) սովորաբար ավելի հարմար և տեղեկատվական են, քան թվային: Այնուամենայնիվ, բարդ քանակական և մաթեմատիկական խնդիրները լուծելու տրամաբանական-վերլուծական մեթոդների հնարավորությունները շատ սահմանափակ են: Այս դասընթացում գերակշռում են համակարգիչներում իրականացվող թվային մեթոդները: Նկատի ունեցեք, որ «վերլուծական լուծում» և «համակարգչային լուծում» հասկացությունները ոչ մի դեպքում չեն հակադրվում միմյանց, քանի որ,

ա) մաթեմատիկական մոդելավորման մեջ ավելի ու ավելի հաճախ համակարգիչները օգտագործվում են ոչ միայն թվային հաշվարկների, այլև տրամաբանական և վերլուծական ձևափոխությունների համար,

բ) մաթեմատիկական մոդելի տրամաբանական-վերլուծական ուսումնասիրության արդյունքը հաճախ արտահայտվում է այնպիսի բարդ տրամաբանական բանաձևով, որ երբ նայում ես դրան, նրա նկարագրած գործընթացի տեսողական ընկալում գոյություն չունի:

**Դաս 2**

## Մոդելավորումը MATLAB & SIMULINK միջավայրում

## Simulink-ի նպատակը

Իմիտացիոն մոդելավորման Simulink փաթեթը նախատեսված է համակարգերի իմիտացիոն ձևավորման համար, որոնց վարքագիծը կախված է ժամանակից (դինամիկ համակարգեր)։ Այլ բառերով ասած, Simulink-ը թույլ է տալիս ստեղծել և խմբագրել դինամիկ համակարգերի քոմփյութերային մոդելներ և նրանց վրա կատարել քոմփյութերային փորձեր։ Մոդելավորված դինամիկ համակարգը ներկայացվում է ֆուկցիոնալ բլոկ-սխեմայի տեսքով, Simulink-մոդել, S-մոդել կամ ուղղակի մոդել անվանմամբ։

Simulink-ը թույլ է տալիս մոդելավորել ինչպես գծային, այնպես էլ ոչ գծային դինամիկ համակարգեր՝ անընդհատ կամ դիսկրետ ժամանակով։ Անընդհատ և դիսկրետ մոդելների հետ միասին, Simulink-ը մշակում է նաև իրադարձություններով կառավարվող հիբրիդային մոդելների համակարգ։ Simulink-ը ապահովում է S-մոդելի․

* նախագծման բոլոր անհրաժեշտ գործիքները,
* գործարկման և կատարման, մատուցման կամ առաջադրման հնարավորություն,
* գործունեության ընթացքում ստացված վերլուծական արդյունքների հավաքագրում և ցուցադրում։

### Simulink-ի աշխատանքային միջավայրը

Simulink փաթեթի աշխատանքային միջավայրը բաղկացած է գործիքներից և միջոցներից, որոնք օգնում են Simulink-ին օգտագործել իր բաղադրիչները, ինչպես նաև աջակցել նրան կապ հաստատելու օգտագոծողի և MATLAB միջավայրի միջև:

### Simulink-ի նշանակությունը և կառուցվածքը

Simulink-ի աշխատանքային միջավայրը իր մեջ ներառնում է բաղադրիչների համընդհանուր և մասնագիտացված Simulink Library Browuser գրադարանները: Բլոկները մշակված են.

* ըստ նշանակության գործիքների հավաքածուներից,
* ներկառուցված բլոկ-սխեմաների խմբագրիչից, որը հիմնված է գրաֆիկական ինտերֆեյսի վրա և հանդիսանում է տեսողականորեն ուղղվածությամբ ծրագրավորման բնորոշ միջոց,
* ստեղծվող S-մոդելի պատուհանից,
* մոդելային ներկայացումների վրա հիմնված նախագծային գործիքներից,
* S-մոդելի նախագծման գործիքներ,
* վերլուծության գործիքներ,
* արդյունքների և S-մոդելի անիմացիայի գրաֆիկական պատուհաններից,
* Simulink Help օգնության տեղեկատու համակարգից:

### Simulink ծրագրի գործարկումը

**Նոր S-մոդելի ստեղծումը, նախկինում ստեղծվածի գործարկումը:**

Simulink ծրագրի գործարկման համար անհրաժեշտ է նախապես բեռնավորել MATLAB փաթեթը: MATLAB համակարգից դուրս Simulink ծրագիրը չի աշխատում: Գոյություն ունի Simulink-ի գործարկման մի քանի տարբերակներ.

MATLAB համակարգի հրահանգի պատուհանի գործիքագոտում Home ► New ► Simulink կոճակը,

>>simulink հրամանի կատարումը MATLAB համակարգի հրամանի պատուհանի հրամանի տողում,

Նշված տարբերակներից յուրաքանչյուրում բեռնավորվում է բլոկային բաղադրիչների գրադարնը և ակտիվանում է Simulink Library Browuser պանակը: Տվյալ պարագայում գոյություն ունեցող S-մոդելի բեռնավորում կամ նոր S-մոդելի ստեղծում չի կատարվում:

Նոր S-մոդելի ստեղծման համար կարելի է կատարել հետևյալ գործողությունները. միաժամանակ սեղմել Ctrl+N կոճակները կամ գործիքների տիրույթում ընտրել մոդելի սետեղծման կոճակը:

Օգտվողի ինտերֆեյսը իրականացվում է հետևյալ ձևով` ընտրել File | New ► Model կամ MATLAB համակարգի գլխավոր մենյուից ընտրել Home ► New ► Simulink Model հրահաանգը:

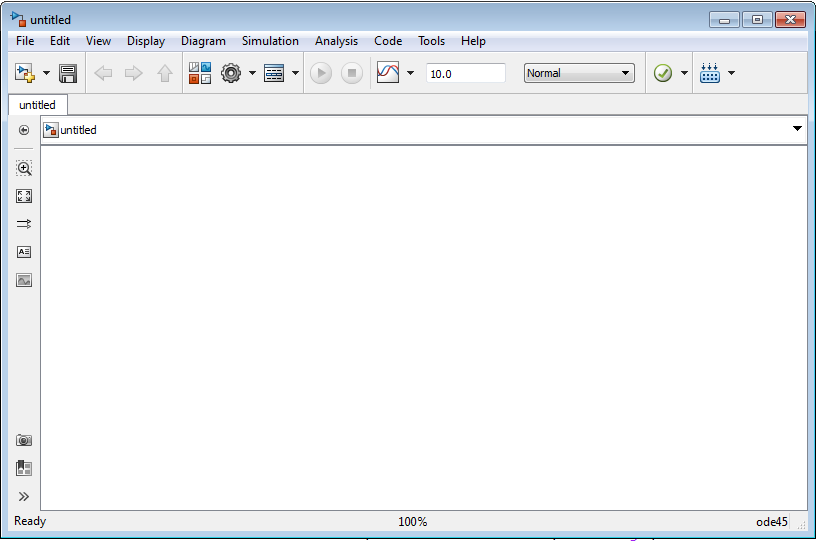
Նախկինում ստեղծված S-մոդելի բեռնավորման եղանակները.

* Ընտրել Home ► Open ► Open բացել MATLAB համակարգի հրամանների պատուհանից կամ Simulink-ի գլխավոր մենյուից File ► Open հրահանգով ընտրել գրադարանը և S-մոդել ֆայլի անվան վրա մկնիկի ձախ կոճակը կրկնակի սեղմել:
* S-MATLAB համակարգի հրամանային պատուհանում՝ Command Window-ի հրամանների տողում հավաքել >> simOut = sim('Model\_name') հրահանգը և առանց S-մոդելի բացման սկսում է S-մոդելի թողարկումը: Այս մեթոդը նախընտրելի է, երբ գործ ունես պատրաստի և աշխատող S-մոդելի հետ:

### Օգտագործողի ինտերֆեյսը Simulink-ում

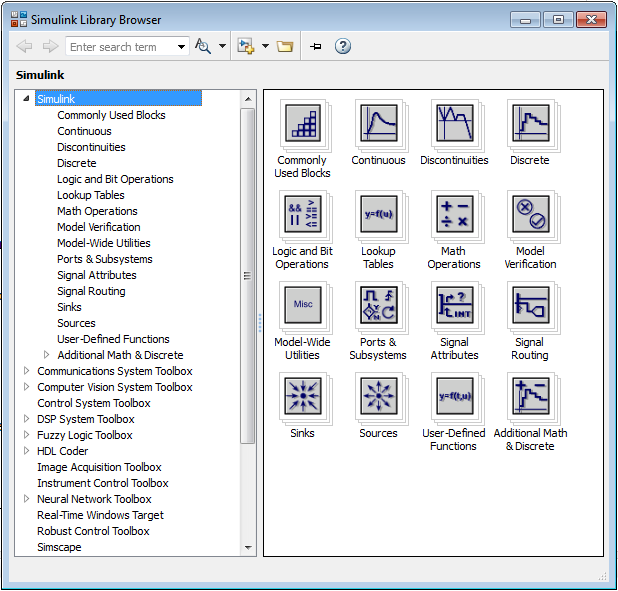
Simulink-ի և օգտագործողի փոխհամագործակցությունը իրականացվում է օգտագործողի գրաֆիկական ինտերֆեյսի միջոցով (Graphical User Interface, GUI), որի բաղադրիչ մասեր են հանդիսանում Simulink Library Browser գրադարանները և S-մոդելի պատուհանը։ S-մոդելի կառուցման ընթացքը իրականացվում է բլոկ-սխեմաների խմբագրիչի միջոցով։ Simulink-ի գործարկումից հետո էկրանին է բացվում բլոկ-սխեմաների խմբագրիչը (նկ․ 1), որում պարունակվող հիմնական բաղադրիչները, ըստ համարակալման բերված են նկ․ 5-ում

1. Պատուհանի անվանմամբ վերնագիր - Simulink Library Browser,
2. գլխավոր մենյու,
3. գործիքների վահանակ, հաճախակի օգտագործվող հրամանների պիկտոգրամաներով և որոնման պատուհանով:



Նկ․1. Simulink-ի բլոկ-սխեմաների խմբագրիչի պատուհանը

1. մեկնաբանությունների պատուհան, ընտրված բլոկի վերաբերյալ բացատրական տեղեկատվություն ցուցադրելու համար,
2. ծառի տեսքով իրականացված համընդհանուր և մասնագիտացված գրադարանների բաժինների ցուցակները,
3. գրադարանի բաժնի պարունակության պատուհան (գրադարանների բաժինների ներդրված ցուցակներով կամ ընտրված բլոկի բաժինների),
4. կատարվող գործողությունների ակնարկներով լրացված վիճակի տող։



Նկ․ 1. Simulink գրադարանային պատուհանը

### Simulink գրադարան

Simulink Library Browser պատուհանը ունի հետևյալ հնարավորությունները․

* բլոկների փնտրումը ըստ կոճակի անվանման Simulink-ի բլոկային գրադարաններում, գործիքների վահանակի վրա, ունեն հետևյալ նշանակությունը,
* ստեղծել նոր S-մոդել (բացել մոդելի նոր պատուհան) – new model,
* բացել գոյություն ունեցող S-մոդելը - Open model,
* փոխել բրաուզերի պատուհանի հատկությունները,
* անվանումով բլոկի փնտրում (անվանումների առաջին տառերով) – Find block,
* որոնվող տեղեկատվության մուտքագրման դաշտ,

Browser Toolbar The Simulink բրաուզերը ունի իր բլոկի որոնման համակարգը բլոկների անունով (նկ. 1) :

Իմանալով նպատակային բլոկի անգլերեն լեզվով անվանումը կարելի է ավտոմատ կերպով իրականացնել բլոկի որոնումը։ Եթե գործիքների վահանակի նշանից աջ տեքստային դաշտում մուտքագրենք որևիցէ բառ, օրինակ "demux", և սեղմել <Enter> ստեղնը, ապա համակարգը Simulink գրադարանում կսկսի որոնել այդ անվանմամբ բլոկը և հայտնաբերելու դեպքում կարտապատկերի առաջին հայտնաբերածը, որի անվանման մեջ կա փնրման դաշտում մուտքագրված բառը։ Դրա հետ միաժամանակ կբացվի գրադարանի այն բաժինը, որի մեջ գտվում է այդ բլոկը, իսկ գտված բլոկը կընգծվի։ Շարունակելով որոնումը կարելի է գտնել Simulink գրադարանի այն բոլոր բլոկները, որոնց անվանման մեջ կա "demux" բառը։ Իսկ եթե այդպիսի անվանմամբ բլոկը բացակայում է, ապա մեկնաբանությունների պատուհանում դուրս կբերվի՝ Not found "demux" ("demux" բլոկը բացակայում է)։

### Simulink-ի նշանակությունը և կառուցվածքը

Simulink-ի աշխատանքային միջավայրը իր մեջ ներառնում է բաղադրիչների համընդհանուր և մասնագիտացված գրադարանների Simulink Library Browuser բրաուզերը: Բլոկները կազմակերպված են.

* ըստ նշանակության բաղադրիչների հավաքածուներից,
* ներկառուցված բլոկ-սխեմաների խմբագրիչից, որը հիմնված է գրաֆիկական ինտերֆեյսի վրա և հանդիսանում է տեսողականորեն ուղղվածությամբ ծրագրավորման բնորոշ միջոց,
* ստեղծվող S -մոդելի պատուհանից,
* մոդելային ներկայացումների վրա հիմնված նախագծային գործիքներից,
* S-մոդելի գործիքներ,
* վերլուծության գործիքներ,
* արդյունքների և S-մոդելի անիմացիայի գրաֆիկական պատուհաններից,
* Simulink Help օգնության տեղեկատու համակարգից:

S-մոդելը կառուցելու համար, S-մոդելի պատուհանում տրամաբանորեն կարգաբերել բլոկները անհրաժեշտ կարգով, բլոկների հատկությունների օգնությամբ, նրանցից յուրաքնչյուրի համար սահմանել պարամետրեր, այնուհետև բլոկները միացնել գծերով, որոնց կապման ազդանշանը բլոկներին փոխանցվում է S-մոդելի պատուհանից։ Կարելի է նաև կազմակերպել S-մոդելի ավտոմատ վերահսկումը, մասնավորապես՝ S-մոդելի համար նախապես տալով կազմաձևման, ժամանակի սկզբի և ավարտի, ընտրիչի տեսակի, մուտքային/ելքային տվյալների պարամետրերը։

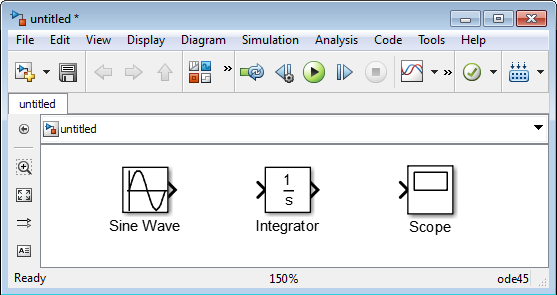
Simulink-ի նպատակն է սկսել և դադարեցնել S-մոդելի կատարումը, պահպանել S-մոդելի ֆայլը, ստանալ տպագրման բլոկ-սխեմա:

### S-մոդելի կառուցումը և կառուցման փուլերը

SIMULINK–ի միջավայրում նոր S-մոդելի կառուցման համար անհրաժեշտ է ստեղծել նոր S-մոդել, ինչպես նկարագրված է 1.3-ում։ Ստեղծվում է դատարկ, ինչպես նկ․1-ում է ցուցադրված, նոր S- մոդելի պատուհանը։

S-մոդելի կառուցման փուլերը․

**1․**Բլոկների տեղակայում S-մոդելի պատուհանում: Դրա համար անհրաժեշտ է բացել գրադարանի համապատասխան բաժինը, կուրսորի միջոցով նշել պահանջվող բլոկը, սեղմած պահելով մկնիկի ձախ կոճակը բլոկը սահեցնել S-մոդելի պատուհանի մեջ, կամ բլոկի վրա կատարել մկնիկի աջ հարված և բացվող մենյուից **Add block to model vib** կամ **Ctrl+I** հրահանգով բլոկը տեղափոխել աշխատանքային դաշտ։



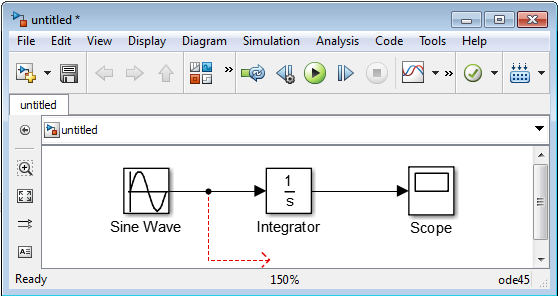
Նկ․ 2. S- մոդելի բլոկների տեղակայում S- մոդելի պատուհանում

**2․** Տեղեկատվական կապերի անցկացում, յուրաքանչյուր բլոկ ունի մեկ կամ մի քանի **>** տեսքի մուտքեր և ելքեր (տես նկ․ 3):



Նկ․ 3. S-մոդելի բլոկների տեղակայում S-մոդելի պատուհանում

> սիմվոլը, որը ցույց է տալիս մուտք դեպի բլոկ, կոչվում է մուտքային հանգույց կամ մուտք։ > սիմվոլը, որը ցույց է տալիս ելք բլոկից, կոչվում է ելքային հանգույց կամ ելք։ S-մոդելի բլոկները S-մոդելի պատուհանում տեղակայելուց հետո անհրաժեշտ է տալ բլոկ-սխեմայի էլեմենտների կապերը։ Դրա համար մկնիկի կուրսորի միջոցով պետք է նշել բլոկի մուտքը, այնուհետև սեղմել ձախ կոճակը և առանձ բաց թողնելու մկնիկի ձախ կոճակը, տանել ուղիղ մինչև մյուս բլոկի մուտքը, որից հետո բաց թողնել կոճակը։ Ճիշտ կապ ստեղծելու դեպքում կապի ուղիղ գիծը ամբողջությամ դառնում է սև, իսկ սխալի դեպքում կետավոր – կարմիր գիծ։ Բլոկների միջև միացումներ կատարված S-մոդելի սխեմայի օրինակ բերված է նկ․ 4-ում։



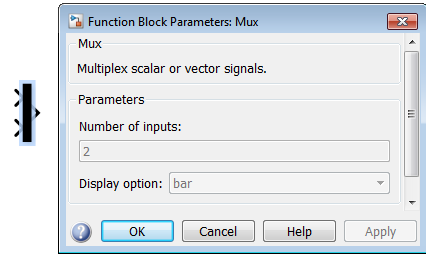
Նկ․ 4. Բլոկների միջև միացումներ կատարված S-մոդելի

սխեմայի օրինակ

**3․**Բլոկների պարամետրերի կարգավորումը, համակարգով տեղադրված լռելյայն պարամետրերը փոփոխելու համար, անհրաժեշտ մկնիկի մարկերը կանգնեցնել բլոկի գծապատկերի վրա, հետո սեղմել մկնիկի ձախ կոճակը երկու անգամ։ Կբացվի տվյալ բլոկի պարամետրերի խմբագրման պատուհանը (նկ․ 10):

Տասնորդական թիվ մուտքագրելու դեպքում, որպես բաժանիչ օգտագործվում է կետը, այլ ոչ ստորակետը։ Փոփոխություններ կատարելուց հետո պատուհանը փակում ենք <OK>-ով կամ փոփոխությունները պահում ենք շարունակելով խմբագրման աշխատանքը <Apply> կոճակը սեղմելով։

**4․**Մոդելավորման պարամետրերի կարգավորում – կարգավորման պատուհանի կանչը կատարվում է խմբագրվող S-մոդելի ակտիվ պատուհանի առկայության ժամանակ, մենյուի Simulation Configurations Parameters… ճյուղի ընրությամբ կամ Ctrl+E միաժամանակ ստեղների սեղմմաբ (նկ. 5)։



Նկ․ 5. Բլոկի պարամետրերի խմբագրման պատուհանը

**5․** S-մոդելի ֆայլի պահպանումը – ստեղծելով S-մոդելային սխեման և ընտրելով համապատասխան պարամետրերը, մոդելը պահպանում ենք ընտրելով File / Save as … : Ֆայլի անվանումը չպետք է սկսվի թվով և պարունակի ազգային այբուբենի որևիցէ տառ։ S-մոդելը պահպանվում է ․mdl կամ slx ընդլայնմամբ ֆայլերում։

### S-մոդելի աշխատանքի կառավարում

S-մոդելի գործարկումը իրականացվում է մենյուի Simulation | Start ճյուղի կամ գործիքների վահանակից  գծապատկերի ընտրմամբ կամ Ctrl+T միաժամանակ ստեղների սեղմմաբ։ S-մոդելի աշխատանքը կշարունակվի այնքան ժամանակ մինչև մոդելավորման ժամանակի լրանալը։ Աշխատանքի պրոցեսը կարելի է ավարտել ընտրելով մենյուի Simulation / Stop ճյուղի կամ  գծապատկերի ընտրմամբ։ Աշխատանքի պրոցեսը կարելի է ավարտել նաև ընտրելով մենյուի Simulation | Pause ճյուղը և հետո շարունակել աշխատանքը ընտրելով մենյուի Simulation | Continue ճյուղը կամ  գծապատկերը:

Աշխատանքը ավարտելու համար կարելի է սպասել նրա ավարտին կամ կանգնեցնել կատարումը, պահել S-մոդելը ֆայլում, փակել S-մոդելի պատուհանը, փակել բաղադրիչների համընդհանուր և մասնագիտացված բլոկների գրադարանների Simulink Library Browuser բրաուզերի պատուհանը, ինչպես նաև MATLAB համակարգի գլխավոր պատուհանը։

**S-մոդելի խմբագրման հիմնական գործիքները**

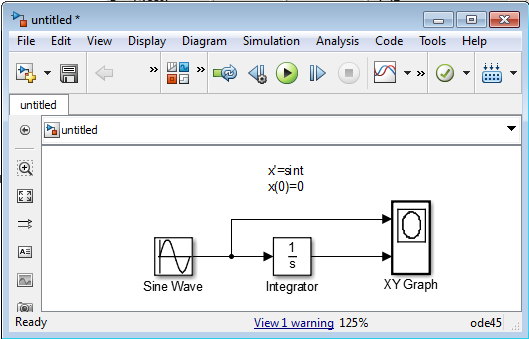
Բաժնի նպատակն է, տիրապետել․

* Simulink-ի միջավայրում գրաֆիկական խմբագրիչի աշխատանքի հիմնական հմտություններին,
* S-մոդելի բլոկ-սխեմաների ստեղծման և խմբագրման հիմնական հնարքներին։

**Գործողություններ օբյեկտների հետ**

**Օբյեկտների ընտրում։**  S-մոդելի օբյեկտների վրա որևիցէ գործողություն իրականացնելու համար (բլոկ, միավորող գծեր, գրառում) պահանջվում է տվյալ օբյեկտը կամ օբյեկտների խումբը ընտրել։

* Մեկ առանձին օբյեկտին ընտրելու համար, անհրաժեշտ է, մկնիկի կուրսորը տեղակայել օբյեկտի վրա և սեղմել մկնիկի ձախ կոճակը։ Օբյեկտի անկյուններում առաջացած մարկերները մեզ հուշում են, որ օբյեկտը ընտրված է (օրինակ, նկ․6-ում մեր կողմից առանձնացված է Sine Wave բլոկը):



Նկ․6. Sine Wave բլոկի ընտրում

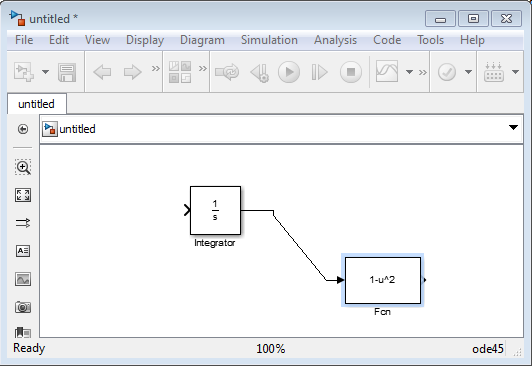
* Մի քանի հարևան օբյեկտների ընտրելու համար, անհրաժեշտ է, մկնիկի կուրսորը տեղակայել օբյեկտի վրա և սեղմել մկնիկի ձախ կոճակը և կոճակը սեղմած պահելով մկնիկը սահեցնել։ Առաջացած կետգծերով ուղղանկյուն շրջանակը կփոփոխվի մկնիկի սահեցման ժամանակ։ Մկնիկի սեղմած կոճակը բաց թողնելուց հետո, շրջանակով ծածկված օբյեկտները դառնում են ընտրված:
* Մի քանի առանձին տեղակայված օբյեկտների ընտրելու համար, անհրաժեշտ է, սեղմած պահել < Shift > ստեղնը, մկնիկի կուրսորը տեղակայելով անհրաժեշտ օբյեկտներիի վրա և սեղմել մկնիկի ձախ կոճակը։ Այդ գործողությաունները նշված օբյեկտները դառնում են ընտրված:
* S-մոդելի ակտիվ պատուհանի առկայության դեպքում, S-մոդելի բոլոր օբյեկտները նշելու համար, օգտագործում ենք մենյուի Edit / Select All ճյուղը կամ միաժամանակ սեղմում ենք Ctrl+A ստեղները։

**Բլոկի չափսի փոփոխում։** Բլոկի չափսերը փոփոխելու համար, անհրաժեշտ է ընտրել նրան, հետո տեղափոխել մկնիկի մարկերը բլոկի անկյունների վրա առաջացած մարկերներից մեկի վրա։ Մարկերի տեսքի փոփոխելու (երկկողմանի սլաք դարձնելու) համար անհրաժեշտ է, սեղմել մկնիկի ձախ կոճակը։ Այնուհետև բլոկը մեծացնելու համար մարկերը ձգել, բլոկը փոքրացնելու համար մարկերը սեղմել։ Բլոկի չափսը փոփոխելուց փոխվում է միայն գծապատկերի (բլոկի գրաֆիկական պատկերի) չափսերը, ընդ որում բլոկին վերաբերող տեքստային գրառումների (անվանումների) չափսերը չեն փոխվում։

**Բլոկների միավորում։** Բլոկների միավորումը նշում է ազդանշանի փոխանցման ուղղությունը և արտապատկերվում է միավորող ուղիղը, սև կետգծերով կառուցման ընթացքում և հաջող ավարտի դեպքում՝ սև հոծ գծով, ծայրակետում տեղակայված գունալցված սլաքով։ Սխալ միավորում կատարելու դեպքում միավորող գիծը մնում է կետագիծ և վերափոխվում է կարմիր գույնի։ Միավորող գծերի ստեղծման ձևերը․

* մկնիկի կուրսորը տեղակայել բլոկի ելքի վրա, որտեղից պետք է դուրս գա բլոկների միավորումը։ Այդ դեպքում մկնիկի մարկերը ձևափոխվում է բարակ կրկնակի գծերով խաչի։ Այնուհետև սեղմել մկնիկի ձախ կոճակը, և առանց բաց թողնելու, մոտեցնել մկնիկի կուրսորը հաջորդ բլոկի մուտքին։ Մկնիկի տեղաշարժման ժամանակ առաջանում է միավորող կետգծիկներով ուղիղը։ Բլոկի մուտքին մկնիկի կուրսորի մոտեցման ժամանակ այն նորից ձևափոխվում է՝ բարակ կրկնակի գծերով խաչի: Այժմ մկնիկ ձախ կոճակը բաց ենք թողնում։
* Նշել բլոկ-սկզբնաղբյուրը, սեղմել <Ctrl> ստեղնը, այն սեղմած պահել և ընտրել բլոկ-ընդունիչը։

Թեք կապող գծեր ստեղծելու համար, ինչպես ցուցադրված է նկ․6-ում, անհրաժեշտ է կապերը ստեղծել < Shift > ստեղնը սեղմած պահելով։

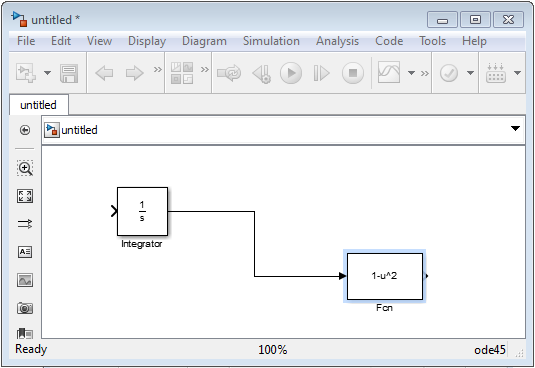


Նկ․7. Թեք կապող գծերով բլոկ-սխեմայի օրինակ։

### Կապող տեսակի միավորման գծի ստեղծումը

Կապող տեսակի միավորման գիծ ստեղծելու համար անհրաժեշտ է մկնիկի կուրսորը տեղակայել բլոկ-սխեմայի աղդանշանի ենթադրյալ հանգուցային կետի տեղում և սեղմելով մկնիկի աջ ստեղնը (կամ ընտրելով <Ctrl> և սեղմած պահելով մկնիկի ձախ կոճակը) տանել ուղիղ։ Ստեղծված կապող տեսակի միավորման գծի արդյունքը բերված է նկ․ 5-ում։

Միավորման գծերի խմբագրում։ Միավորման գիծը նշվում է մկնիկի ձախ կոճակի սեղմմամբ։ Ընտրված միավորման գծի հատկանիշ են հանդիսանում հանգուցային կետերում սև ուղղանկյունների առաջացումը, ինչպես ցույց է տրված նկ․ 8-ում։

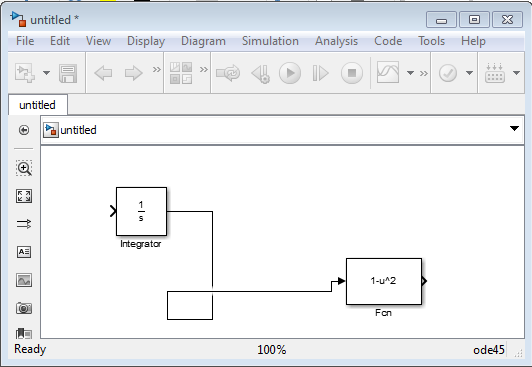


Նկ․8. բլոկ-սխեմայում ընտրված միավորման գծի օրինակ։

Միավորման գծի ձևի փոփոխման համար անհրաժեշտ է ընտրել գծի ցանկալի մասը և տեղաշարժել մկնիկը, սեղմած պահելով նրա ձախ կոճակը, ձևափոխմանը անհրաժեշտ ուղղությամբ։ Միավորման գծի խմբագրումը ավարտվում է մկնիկի ձախ կոճակի բաց թողնելով։

### 1.8.2 Բլոկների և միավորման գծերի հեռացում

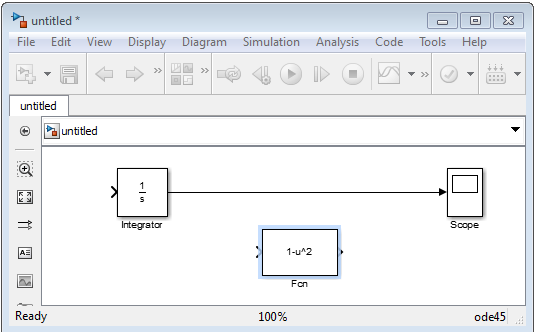
Բլոկի կամ միավորման գծի հեռացման համար անհրաժեշտ է նշել օբյեկտը և կատարել S-մոդելի գլխավոր պատուհանի կամ կոնտեքստային մենյուի Edit ► Cut или Edit ► Delete հրամանը կամ սեղմել ստեղնաշարի < Delete > ստեղնը։ Այս գործողությունը կարելի է չեղարկել մենյուի Edit ► Undo հրամանով։

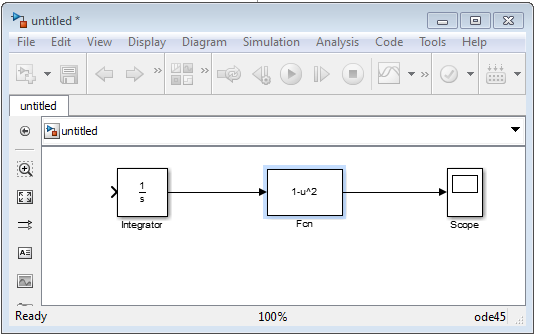


Նկ․9. բլոկ-սխեմայում միավորման գծի հեռացման օրինակ

### Միավորումների մեջ բլոկների տեղադրում

Միմյանց հաջորդաբար միավորված բլոկների միջև կարելի է տեղադրել, մեկ մուտք և մեկ ելք ունեցող բլոկ։ Դրա համար անհրաժեշտ է, մկնիկի օգնությամբ, տեղափոխել բլոկը միավորման գծի պահանջվող տեղը։ Նկ․10-ում ցույց է տրված Fcn բլոկի տեղադրումը Integrator և Scope բլոկների միջև։





Նկ․ 10. բլոկ-սխեմայում միավորուների մեջ բլոկի տեղադրման օրինակ

Եթե ցանկանում եք տեղադրել բլոկ, որը ունի երկու և ավելի մուտք կամ ելք, ապա պետք է հեռացնել միավորման գիծը, մկնիկի օգնությամբ, տեղափոխել բլոկը պահանջվող տեղը և նորից կառուցել միավորման գիծը։

Բլոկների տեղաշարժում։ S-մոդելը բլոկը կարելի է տեղափոխել, նրան ընտրելով և մկնիկի ձախ կոճակը սեղմած պահելով։ Եթե տեղաշարժվող բլոկը ունի միավորման գծեր, ապա նրանք տեղաշարժվում են օբյեկտի հետ միասին (չի անջատվում օբյեկտից)։

### Տեքստային գրառումների տեղադրում

S-մոդելը առավել դիտարժան դարձնելու համար, բլոկ-սխեմաներում օգտագործում են տեքստային գրառումներ։ Գրառում ստեղծելու համար անհրաժեշտ է մկնիկի օգնությամբ մարկերը տեղափոխել գրառման տեղը, կրկնակի անգամ սեղմել մկնիկի ձախ կոճակը։ Կհայտնվի ուղղանկյուն շրջանակ մուտքի կուրսորով, որտեղ և կատարվում է տեքստի գրառումը։ Ընտրելով մուտքագրված տեքստը, կարելի է կարգավորել տեքստային գրառումը․

* ընտրելով խմբագրվող S-մոդելի պատուհանի գլխավոր ընտրանում Diagram ► Format ► Font… ճյուղը կամ կոնտեքստային ընտրանու Font… ճյուղը, կարողանում ենք փոխել շրիֆտը, տառերի եզրագիծը և տեքստային գրառման չափը,
* ընտրելով խմբագրվող S-մոդելի պատուհանի գլխավոր ընտրանու Diagram ► Format ► Foreground Color … ճյուղը կամ Format ► Background Color ․․․ ճյուղը կամ կոնտեքստային ընտրանու համապատասխան ճյուղերը, կարողանում ենք փոխել ֆոնի և տեքստի գույնը,
* ընտրելով խմբագրվող S-մոդելի պատուհանի գլխավոր ընտրանու կամ կոնտեքստային ընտրանու Diagram ► Format ► Text alignment … ճյուղը, կարողանում ենք փոխել տեքստային գրառումների ուղվածության ձևերը։

### Օբյեկտների անվանակոչումը

S-մոդելի տեղադրման ժամանակ նրան վերագրվում է անուն, որը բաղկացած է բլոկի անվանումից և հերթական համարից։ S-մոդելում բոլոր անունները պետք է լինեն ունիկալ` առնվազն պարունակեն մեկ սիմվոլ։ Լռելյայնորեն բլոկի անվանումը հայտնվում է բլոկի ներքևում, եթե նրա կապի հանգույցները տեղակայված են բլոկի կողային կողմերին, և բլոկից աջ, եթե կապի հանգույցները տեղակայված են բլոկի հորիզոնական կողմերին։ Օբյեկտի անվանումը փոխելու համար անհրաժեշտ է կրկնակի սեղմել մկնիկի ձախ կոճակը S-բլոկի անվանումը պարունակող գրառման վրա։ Կհայտնվի, մուտքի մարկերով ուղղանկյուն շրջանակ, որտեղ և մուտքագրվում է բլոկի նոր անունը։ Եթե բլոկը ունի միավորման հանգույցներ, ապա նրանք նույնպես կանվանակոչվեն։

### Պատկերների մաշտաբի փոփոխում

Բլոկ-սխեմայի պատկերի մաշտաբը փոփոխելու համար անհրաժեշտ է S-մոդելի պատուհանը ակտիվացնել, այնուհետև ընտրել գլխավոր ընտրանու View ճյուղը և հետևյալ հրահանգներից որևէ մեկը․

* Zoom In - պատկերը մեծացնելու համար,
* Zoom Out - պատկերը փոքրացնելու համար,
* Normal Viev 100% - բլոկ-սխեմայի չափերը բերել նորմալ չափերի 100%,
* Fit To View - բլոկ-սխեմայի չափերը S-մոդելի պատուհանի չափի դարձնելու համար:

Բլոկ-սխեմայի պատկերի մասշտաբը տոկոսներով ցուցադրվում է S-մոդելի պատուհանի վիճակի ռեգիստրի տողում։ S-մոդելի պահելու ժամանակ պահվում են նաև S-մոդելի բլոկ-սխեմայի պատկերի մաշտաբի պարամետրերը։

### Օբյեկտների ֆորմատավորում

S-մոդելի պատուհանի օբյեկտի վրա մկնիկի աջ կոճակի սեղմմամբ բացված գլխավոր մենյուի Format և Rotateat & Flip հրահանգներով բացված ենթամենյուի հրամաններով կարգավորվում են ընտրված օբյեկտի` անվանման արտապատկերումը, ցուցադրման գույների ընտրություն և դիրքի փոփոխումը:

Ֆորմատավորման հրամաննները բաժանվում են մի քանի խմբերի։ Դրանցից են․

1․ Բլոկի անվանման արտապատկերման և ցուցադրման գույների ընտրության հրամանները․

* Font Style – ֆորմատավորել տառատեսակը,
* Foreground Color – ընտրված բլոկի եզրագծի գույնի ընտրում,
* Background Color - ընտրված բլոկի ֆոնային գույնի ընտրում,
* Block Shadow – ցուցադրել/թաքցնել բլոկից առաջացող ստվերը,
* Show Block Name ցուցադրել/թաքցնել բլոկի անունը,

2․ Բլոկի դիրքի փոփոխման և ցուցադրման ձևերը(Rotateat & Flip)․

* Rotateat & Flip | Clockwise - շրջել բլոկը 900–ով ժամսլաքի ուղղությամբ,
* Rotateat & Flip | CounterClockwise - շրջել բլոկը 900 –ով ժամսլաքի հակառակ ուղղությամբ,
* Rotateat & Flip | Flip block - շրջել բլոկը 1800 –ով,

Rotateat & Flip | Flip Block name - բլոկի անվան հայելային արտապատկերում:

Ֆորմատավորման որոշ հրամանների կատարման օրինակներ բերված են աղյուակ 1-ում։

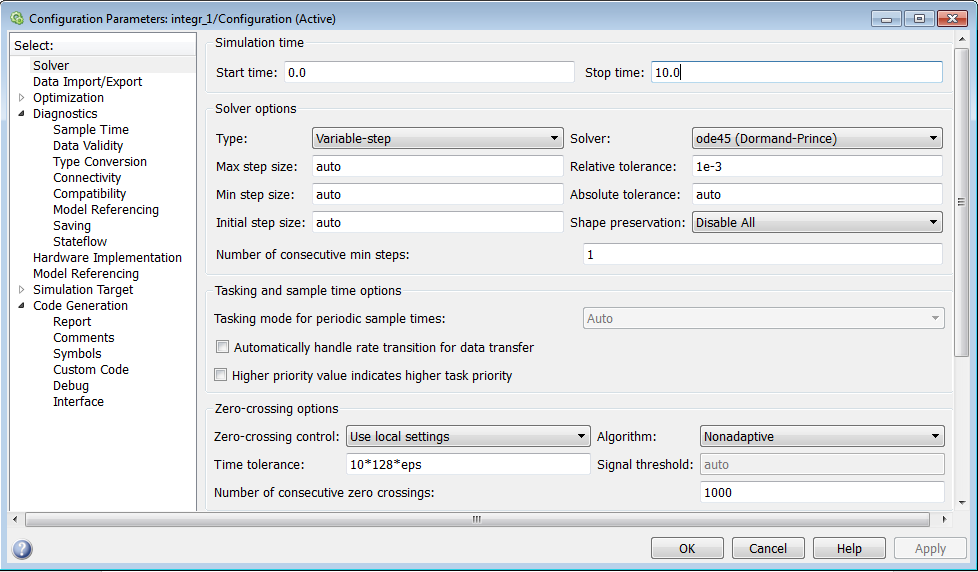
|  |  |  |
| --- | --- | --- |
| **Հրաման** | **Նշանակությունը** | **Օրինակ** |
| Rotateat & Flip | Clockwise | Շրջել բլոկը 900–ով ժամսլաքի ուղղությամբ |  |
| Rotateat & Flip | CounterClockwise | Շրջել բլոկը 900–ով ժամսլաքի հակառակ ուղղությամբ |  |
| Rotateat & Flip | Flip block | Շրջել բլոկը 1800 -ով |  |
| Rotateat & Flip | Flip Block name | Բլոկի անվան հայելային  արտապատկերում |  |

Աղյուսակ 1. Ֆորմատավորման որոշ հրամանների կատարման օրինակներ։

### մոդելի բլոկների պարամետրերի կարգավորումը

S-մոդելի բլոկների պարամետրերի կարգավորումը ունի մի քանի բաժիներ ներկայացված ծառի տեսքով: Ամեն բաժին իր մեջ ներառում է տարրերի խումբ բլոկների պարամետրերի կարգավորման համար:

Դիտարկենք դրանցից մի քանիսը.



11. S- մոդելի պարամետրերի կարգավորումը

Մոդելի հաշվարկային պարամետրերի կարգավորումը կատարվում է Solver բաժնում տեղադրված տարրերի միջոցով: Այդ տարրերը բաժանվում են 3 խմբի (Նկար 4.2):

* **Simulation time** (մոդելավորման ժամանակը),
* **Solver options** (հաշվարկային պարամետրեր),
* **Solver diagnostic controls** (hաշվարկի վերահսկման գործիքներ).

Դիտարկենք առաջին երկու խմբերը:

### Մոդելավորման ժամանակը

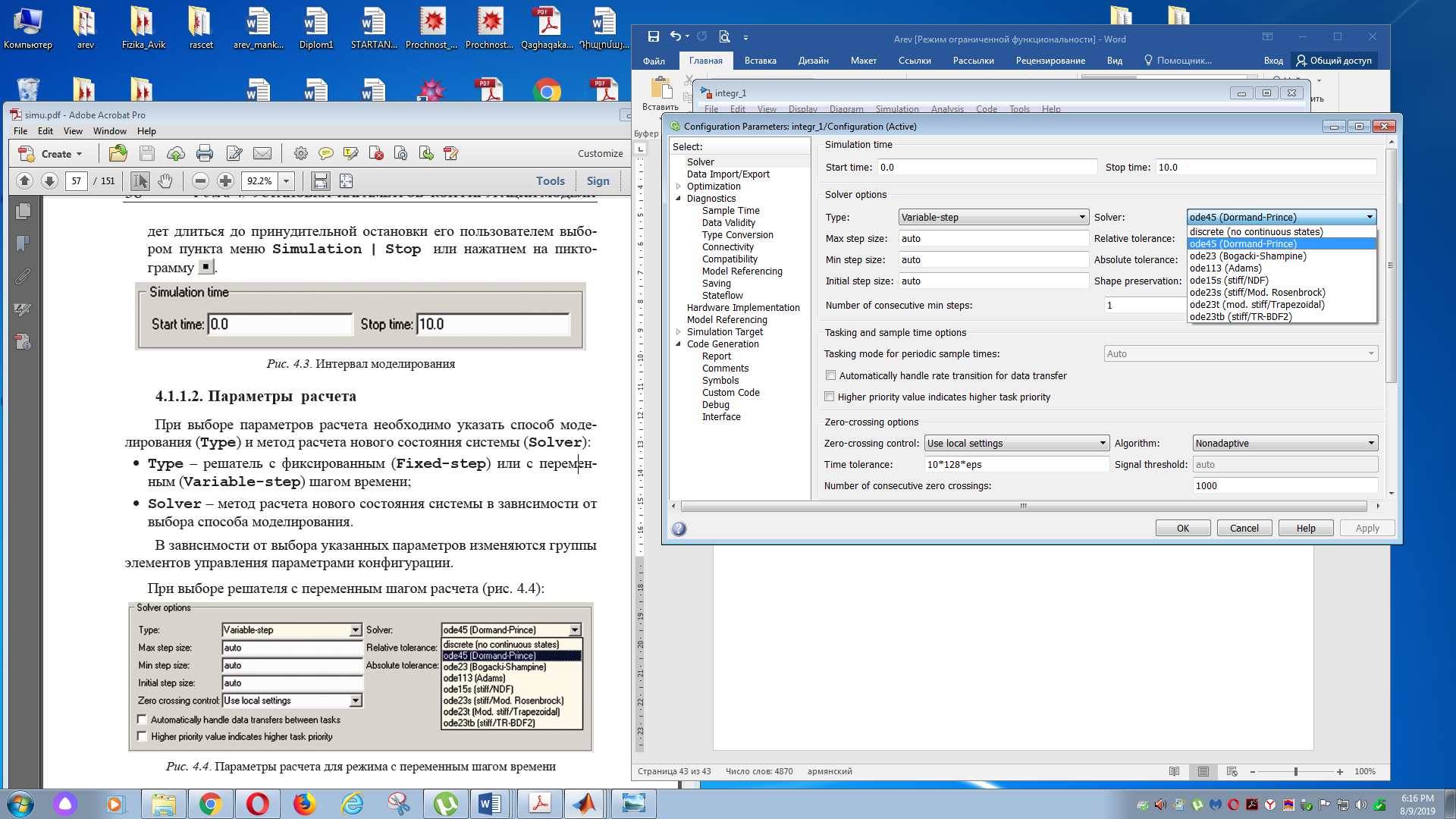
* Մոդելավորման ժամանակի սկիզբը և վերջը տրվում է **Start time** և **Stop time** պարամետրերի արժեքներով (Նկար 11):
* Սկզբնական արժեք որպես կանոն տրվում է զրո:
* Վերջնական ժամանակի արժեքը տրվում է օգտագործողի կողմից՝ կախված խնդրի լուծման պայմաններից, և կարող է լինել վորջավոր ոչբասական թիվ կամ անվերջություն inf: Վերջինիս ժամանակ մոդելավորման պրոցեսը կկատարվի այնքան ժամանակ քանի դեռ օգտագործողի կողմից չի կատարվել Simulation | Stop հրահանգը կամ չի սեղմվել  կոճակը:

### Հաշվարկման պարամետրերը

Հաշվարկման պարամետրերը ընտրելիս անհրաժեշտ է նշել մոդելավորման եղանակը(**Type**) և նոր համակարգի վիճակի հաշվարկման մեթոդը (**Solver**).

* **Type** - Լուծող **Solver** ֆիկսված(**Fixed-step**) կամ փոփոխական (**Variable-step**) ժամանակի քայլով,
* **Solver** - համակարգի նոր վիճակը հաշվելու մեթոդը կախված մոդելավորման մեթոդի ընտրությունից:

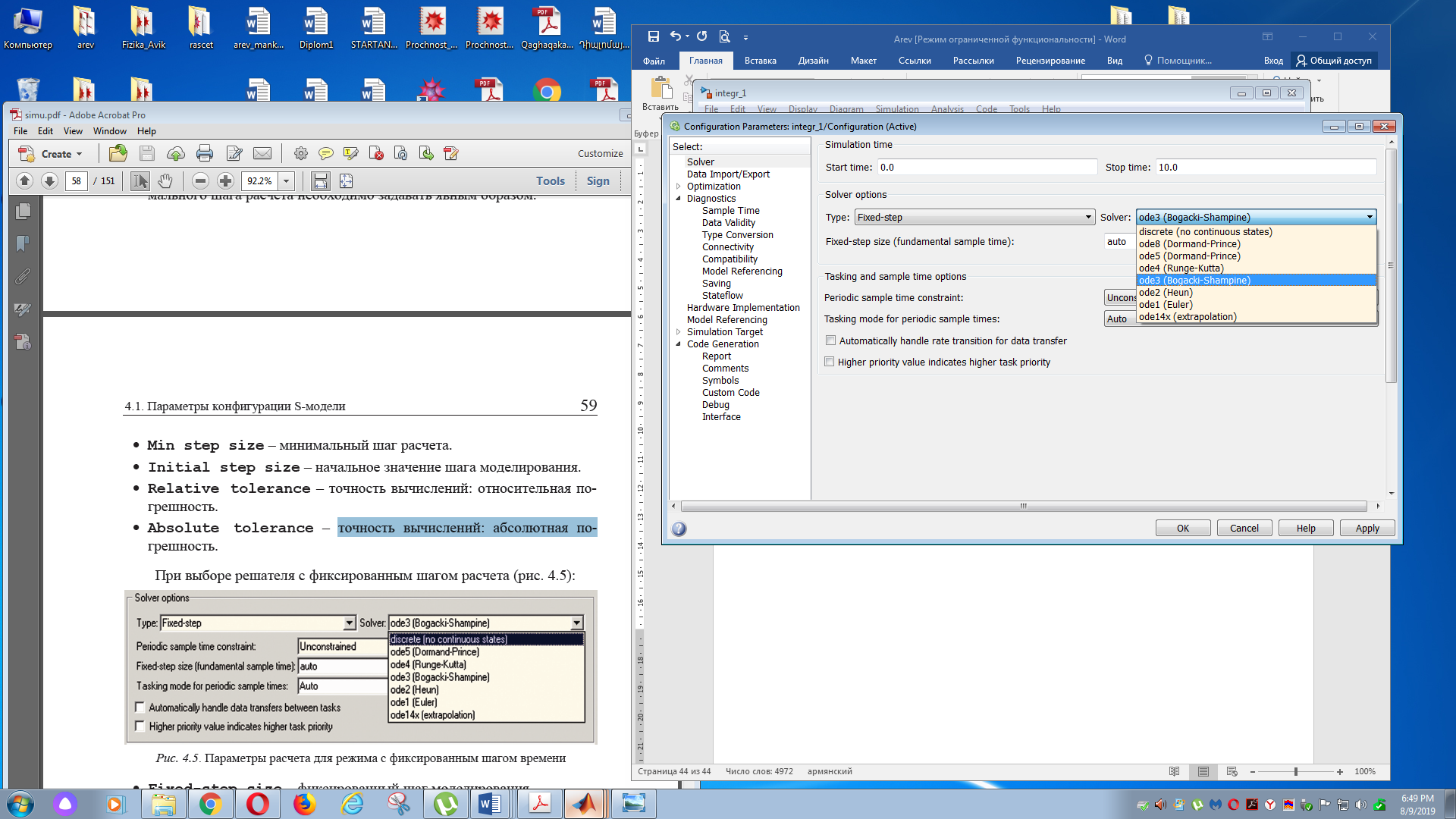
Փոփոխական քայլովSolver-ի պարամետրերի ընտրությունը բերված է նկար (Նկար 12):



Նկ. 12 Փոփոխական քայլովSolver-ի պարամետրերը

* **Max step size** – հաշվարկի մաքսիմալ քայլը, որը ազդում է արդյունքի ճշտության վրա: Լռելյայն ընդունված է հավասար (StopTime - StartTime)/ 50: Եթե այս արժեքը չափազանց մեծ է, ապա դիտարկված գրաֆիկները կտոր առ կտոր գծեր են (այլ ոչ թե հարթ): Այս դեպքում հաշվարկման առավելագույն քայլի արժեքը պետք է հստակ սահմանվի:
* **Мin step size** – հաշվարկման նվազագույն քայլ:
* **Initial step size** – Մոդելավորման քայլի սկզբնական արժեքը:
* **Relative tolerance** - հաշվարկի ճշգրտությունը. հարաբերական սխալանք:
* **Absolute tolerance -** հաշվարկի ճշգրտությունը. բացարձակ սխալանք:

Ֆիկսված քայլովSolver-ի պարամետրերի ընտրությունը բերված է նկար (Նկար 12).



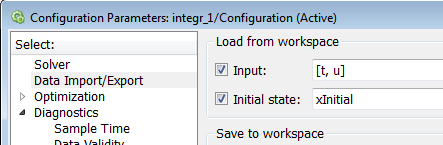
Նկ.13. Ֆիկսված քայլովSolver-ի պարամետրերը

* Fixed-step size - Մոդելավորման ֆիկսված քայլ:
* Tasking Mode for periodic sample times – հաշվարկման ռեժիմները` ավտոմատ, նպատակային և բազմանպատակ:

**Դաս 3**

### Աշխատանքային տարածքի տվյալների փոխանակում

Մոդելավորման մուտքի, ելքի միջանկյալ տվյալների և արդյունքների MATLAB միջավայրի աշխատանքային Workspace տարածքից տեղափոխումը կատարվում է **Data Import/Export** բաժնում (Նկար 11):



Նկ.13 Ղեկավարման տարրերի խումբը Load from Workspace

### Աշխատանքային տարածքից տվյալների բեռնում

Ղեկավարման տարրերը, որոնք մոդելավորման արդյունքները բեռնում են Workspace MATLAB միջավայր, բերված են Նկ. 13:

* Input (մուտքային տվյալներ) – երբ դրոշակը դրված է, թույլ է տալիս տեքստային դաշտում մուտքագրել տվյալների ֆորմատը և անունը, որոնք պետք է բեռնվեն Workspace MATLAB միջավայից:
* Լռելյայն, տեքստային դաշտում գրված է [t, u], որտեղ t-ն ժամանակի, u-ն տվյալների վեկտոր սյուն է:
* Մուտքի **Input** դաշտում նշված տվյալները փոխանցվում են մոդելին

մեկ կամ մի քանի **Sources | In** բլոկների:

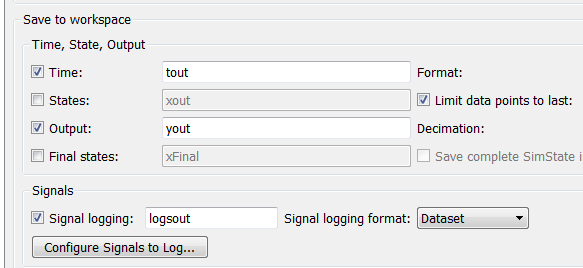
* Լռելյայն, **In** բլոկը **u** զանգվածը ընդունում է տվյալների վեկտոր սյուն:
* Եթե պահանջվում է փոխանցել երկու կամ ավելի վեկտոր uյուներ, ապա In բլոկում, պետք է փոխել **Port dimensions** պարամետրի չափերը, այն դնելով 2 կամ ավելի, կամ տեղադրել մոդելի վրա համապատասխան **In** քանակի բլոկներ:
* [t, u] վեկտորի փոխարեն կարող եք մուտքագրման դաշտում մուտքագրել տվյալների անվանումը օրինակ՝ **data:** data զանգվածը կունենա հետևյալ տեսքը.

:

* **Initial State** (Սկզբնական վիճակ) – սեթե դրված է դրոշակը, ապա իրեն կից տեքստային դաշտում կարող ենք մուտքագրել մոդելի սկզբնական վիճակի պարամետրերը պարունակող փոփոխականի անունը:

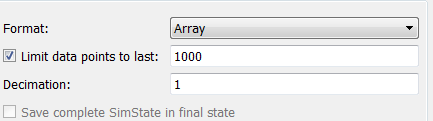
### Տվյալների գրանցումը MATLAB Workspace միջավայր

Ղեկավարման տարրերի խումբը **Save to workspace** (բեռնել workspace աշխատանքային տարածք) թույլ են տալիս ուղարկել տվյալները և ազդզանշանները (**Time** - ժամանակ **tout**, **States** - **xout** վիճակ, **Output yout** ազդանշաններ, **Final states** - վերջնական վիճակը **xFinal**, **Signal logging** ազդանշանի գրանցում **logsout**) **workspace** աշխատանքային տարածք S-մոդելի կատարման ընթացքում (Նկար 14):



Նկ. 14 Save to Workspace ղեկավարման տարրը

**Գրանցման տարբերակները:** Save options ղեկավարման տարրերի խաւմբը թույլ են տալիս տարբեր ձևերով գրանցել մադելավորման արդյունքները աշխատանքային տարածք (Նկար 11):



* Եթե **Limit data points to last** դրոշակը դրված է, ապա կից դաշտում տրվում է հաշվարկային կետերի քանակը(կետերի քանակը որոշվում է հաշվարկի ավարտի պահին): Եթե դրոշակը դրված չէ, ապա փոխանցվում են բոլոր տվյալները:
* **Decimation** (տվյալների գրանցման քայլը) տաիս հաշվարկված կետերի գրանցման քայլը աշխատանքային տարածք: Լռելյայն, Decimation արժեքը 1 է, այսինքն՝ գրանցումը կատարվում է մոդելի յուրաքանչյուր ազդանշանի արժեքի համար: Եթե Decimation-ը դնենք 2, ապա գրանցվելու է յուրաքանչյուրը երկրորդ ազդանշանը և այն:
* **Format** (տվյալների ֆորմատը) սահմանում է այն ֆորմատը, որով պետք է տեղափոխվել տվյալները աշխատանքային տարածքը: Այն կարող է ընդունել հետևյալ արժեքները.
* Array - զանգված է: Տվյալները պահվում են որպես զանգված, որում տողերի քանակը որոշվում է հաշվարկման կետերի քանակով, իսկ սյուների քանակը` ըստ բլոկի մուտքերի քանակի: Եթե բլոկի մուտք տրվում է մեկ ազդանշան վեկտոր է, ապա մատրիցը պարունակում է ընդամենը մեկ սյուն,
* **Structure** – կառուցվածք: Տվյալները պահվում են կառուցվածքի տեսքով, որն ունի երեք դաշտ՝ **time** - ժամանակ (մնում է դատարկ), **signals** - ազդանշանների գրանցվող արժեքներ, blockName - S-մոդելի և բլոկի անվանումը, որից գրանցվում է ինֆորմացիան: **signals** երեք դաշտեր ունեցող կառուցվածք է` values՝ ազդանշանային արժեքների զանգված, dimensions - ազդանշանի զանգվածի չափ, label - մուտքային գծի պիտակ,
* **Structure with Time** – կառուցվածք լրացուցիչ «ժամանակ» դաշտով: Այս դաշտում գրանցվում է ժամանակը: Տվյալների գրանցումը կատարվում է հետևյալ հրամանների միջոցով`**yout.time** և **yout.signals.values,**
* **Output options -** փոփոխական քայլով մոդելավորելու դեպքում գեներացնում է լրացուցիչ ելքային տվյալներ ժամանակի տրված պահերին:

## Simulink գրադարանի կառուցվածքը

Simulik գրադարանը, որի կառուցվածքը ներկայացված է Նկ.1-ում, պարունակում է հետևյալը բաժինները.

* + **Commonly Used Blocks** - լայնորեն օգտագործվող բլոկներ,
  + **Continuous** - բլոկներ, որոնք գործում են անընդհատ ժամանակում,
  + **Discontinuous** - բլոկներ, որոնք կառուցում են տարբեր տեսակի ոչ գծային գծեր,
  + **Discrete** - դիսկրետ ժամանակով գործող բլոկներ,
  + **Logic and Bit Operations** - տրամաբանական և բիթային գործողությունների բլոկներ,
  + **Lookup Tables** - աղյուսակների բլոկներ,
  + **Math Operations** - մաթեմատիկական գործողությունների բլոկներ,
  + **Model Verifications** - ազդանշանի հատկությունների ստուգման բլոկներ,
  + **Model-Wide Utilities** - բլոկներ որոնք ընդլայնում են մոդելի հնարավորությունները,
  + **Ports & Subsystems** - մուտքերի և ենթահամակարգերի բլոկներ,
  + **Signal Attributes** - ազդանշանային հատկանիշների և հատկությունների կարգաորման բլոկներ,
  + **Signal Routings** – ազդանշանային երթուղղման բլոկներ,
  + **Sinks** –ազդանշանային ընդունման և ցուցադրման բլոկներ,
  + **Sources** – ազդանշանային աղբյուրի բլոկներ,
  + **User-Defined Functions** – օգտագործողի գործառույթի բլոկներ,
  + **Additional Math & Discrete** – մաթեմատիկական և դիսկրետ աջակցության լրացուցիչ բլոկներ:

### Որոշիչ էլեմենտներ

Simulink փաթեթը գործարկվում է MatLab ծրագրային միջավայրից։ Ծրագրը աշխատեցնող համապատասխան կոճակը տեղակայված է գործիքների վահանակի վրա։ Աշխատանքը սկսելու համար պետք է նոր \*.mdl ֆայլ (Simulink model), որում մկնիկի օգնությամբ «քաշելու» միջոցով SIMULINK գրադարանից կարելի է տեղակայել բլոկներ։ Անհրաժեշտ է նշել, որ մոդելի գործիքների հավաքածուն գրադարանում ընտրվում են օգտագործողի կողմից մոդելի ստեղծման ժամանակ։

Գլխավոր մենյույում է գտնվում **Simulation/Model Configuration Parametrs** ներդիրը, որտեղ կարելի է փոփոխել մոդելավորման պրոցեսի պարամետրերը, ինչպիսիք են մոդելավորման սկզբի և ավարտի ժամանակը, լուծման մեթոդի ընտրությունը, քայլը որով կատարվում է հաշվարկը և այլն։

Դիտարկենք SIMULINK–ում հաճախ հանդիպող բլոկները։ Յուրաքանչյուր բլոկի համար մկնիկի կրկնակի սեղմումով բացվում է պարամետրերի Function Block Parameters պատուհանը, որտեղ կարելի է փոփոխել համապատասխան գործիքի արժեքները։

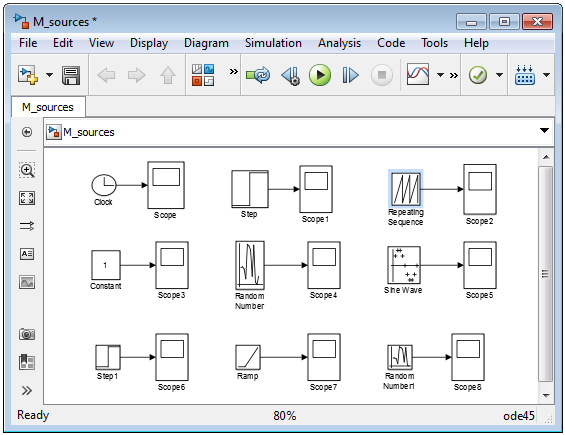
## Ազդանշանների աղբյուրներ

### Sources ազդանշանային աղբյուրի բլոկների ակնարկ

**Sources** պանակի բլոկները ազդանշանների աղբյուր են:

Օրինակ 3.1 Կատարեք հետևյալը.

* + ընտրելով File | New ► Model ստեղծեք S-մոդել,
  + Sources պանակից ազդանշանային բլոկներ տեղափոխեք S-մոդելի պատուհան,
  + Sinks պանակից Scope օսցիլօգրաֆը տեղափոխեք S-մոդելի պատուհան և բազմացրեք,
  + յուրաքանչյուր ազդանշանի աղբյուր միացնել օսցիլօգրաֆի հետ,
  + գրանցեք S- մոդելը M\_ Sources.mdl անունով ֆայլում,
  + գործարկել մոդելը և մկնիկի ցուցիչով կրկնակի հարված կատարելով Scope բլոկին տեսնել առաջացած ազդանշանների տեսքերը:



Նկ․ 15 Ազդանշանների ստեղծման աղբյուրներ

### Բլոկները ըստ նշանակության

**Constant** - հաստատուն ազդանշանի բլոկ,

**Sine Wave** – ձևավորում է տրված հաճախականությամբ, ամպլիտուդայով, ֆազայով և տեղաշարժով սինուսոիդային ազդանշան,

**Signal Generator** – ձևավորում է պարբերական` սինուսոդային, ուղղանկյուն, սղոցաձև, պատահական ազդանշաններ,

**Ramp** – ձևավորում է գծային ազդանշան,

**Step** – ձևավորում է աստիճանաձև ազդանշան,

**Random Number** - ձևավորում է պատահական ազդանշան նորմալ բաշխված մակարդակով,

**Sources** - պանակում ազդանշանային գեներատորներից բացի կան նաև բլոկներ, որոնք ապահովում են.

* **From Workspace** – Simulink ծրագրային միջավայրից տվյալների MatLab Workspace միջավայր փոխանցման բլոկ,
* **From File**  – ֆայլից տվյալներ ստանալու բլոկ,
* **To File**  – ֆայլում տվյալների գրանցման բլոկ:

Մկնիկի կրկնակի հարված կատարել Sine Wave բլոկի վրա և ուսումնասիրել հնարավորությունները:

Sine Wave սինուսոիդալ ազդանշանի գեներատոր է տրված  հաճախականությամբ,  ամպլիտուդայով,  ֆազայով և շեղումով: Այն կարող է աշխատել երկու ռեժիմով՝ գեներացնել անընդհատ ազդանշան (Time based) և կտոր առ կտոր հաստատուն ազդանշան հաստատուն քայլով (Sample based), որը որոշվում է Sine type պարամետրով և Sample time ոչ զրոյական արժեքով:

Time based անընդհատ ռեժիմում Sine Wave բլոկը գեներացնում է հետևյալ տեսքի տատանումներ : Լռելյայն, պարամետրերի արժեքներն են՝ a = 1, ω = 1, θ = 0, u0 = 0, և բլոկը գեներացնում է  տիպիազդանշան: Այս դեպքում ժամանակի պարամետրը Sample time պետք է լինի զրո:

Sampled base ռեժիմում կտոր առ կտոր հաստատուն ազդանշանը տրվում է դիսկրետացման ալգորիթմով հետևյալ բանաձևով.

,

որտեղ -ն և -ն համապատասխանաբար Samples period և Number of offset samples պարամետրերի արժեքներն են: Պարամետր Sample time-ը պետք է լինի դրական:

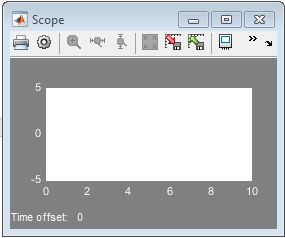
Պարամետր Time (t) ընդունում է երկու արժեք Use simulation time կամ Use external signal: Վերջինիս դեպքում Sine Wave բլոկի վրա ավելանում է մուտքի հանգույցին, որին անհրաժեշտ է միացնել ժամանակի աղբյուրը:

### Sinks բաժնի բլոկների ակնարկ

Sinks բլոկները օգտագործվում են ազդանշաններ ստանալու և արտածելու համար,

* **Display** - ցույց է տալիս մուտքային ազդանշանի թվային արժեքը,
* **Outport** - ստեղծում է ելքային կապ ենթահամակարգի կամ հիերարխիայի բարձր մակարդակի մոդելի համար,
* **Scope, Floating Scope** – կառուցում է ուսումնասիրվող ազդանշանների գրաֆիկը կախված ժամանակից: դիտարկեք ազդանշանի փոփոխությունները մոդելավորման ընթացքում,
* **Stop Simulation** - ընդհատում է S-մոդելի կատարումը, եթե բլոկի մուտքային ազդանշանը հավասար չէ զրո,
* **Terminator** - արգելափակում է որոշ բլոկի ելքը, որը միացված չէ այլ բլոկի մուտքին,
* **To File** - մուտքային տվյալները գրանցում է ֆայլում,
* **Workspace** - տվյալներ է գրանցում MATLAB աշխատանքային տարածքում,
* **XY Graph** - MATLAB միջավայրի Figure պատուհանում XY հարթության վրա ստեղծում է մուտքային ազդանշանի գրաֆիկը:
* **Scope -** օսցիլոգրաֆ է, որը ցույց է տալիս ուսումնասիրվող ազդանշանների գրաֆիկը կախված ժամանակից:

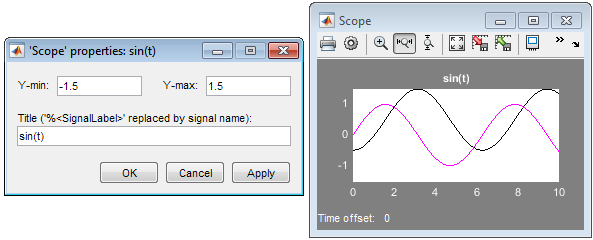
Ազդանշանի դիտման համար անհրաժեշտ է կրկնակի սեղմել մկնիկի ձախ կոճակը Scope բլոկի վրա: Դա կարելի է անել մոդելավորման ցանկացած փուլում: Օսցիլոգրաֆի ազդանշանները դիտելու համար պատուհանը կարգավորվում կատարվում է վահանակի գործիքներով (նկար 16):



Նկ. 16 Ազդանշանի դիտման պատուհան Sinks | Scope

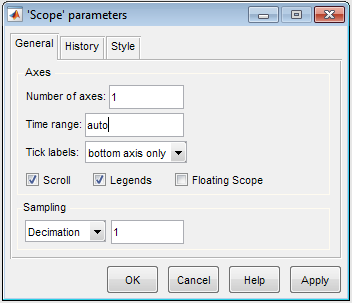
Եթե մկնիկի աջ ստեղնով սեղմեք Scope պատուհանի **Autoscale** հրամանի վրա, ապա տեղի կունենա գրաֆիկի առանցքների ավտոմատ մասշտաբավորում:

Գրաֆիկի ուղղահայաց առանցքի երկայնքով սահմանային արժեքները Y-min և Y-max կարող ենք փոփոխել հետևյալ կերպ` մկնիկի աջ հարված կատարել գրաֆիկի դաշտում և բացվող մենյուից ընտրել **Axes Propertis** ենթամենյուն: Կբացվի պատուհան, որտեղ կարող ենք փոխել Y-min և Y-max արժեքները և գրաֆիկի Title անվանումը: Ստորև բերված է նկարը:



Scope բլոկի պարամետրերը տեղադրված են «Scope» պարամետրերի պատուհանում, որը բացվում է գործիքագոտու վրատեղադրվածParameters հրամանով:

### Պարամետրերի պատուհանը ունի երեք ներդիր



* **General** - ընդհանուր պարամետրեր,
* **History** **-** ազդանշանների պարամետրեր արտածում և դրանց գրանցումը MATLAB միջավայրի Workspace աշխատանքային տարածքում:
* **Style** – գրաֆիկի տիրույթի ձևավորում:

### General ներդիրում տեղադրված պարամետրերը

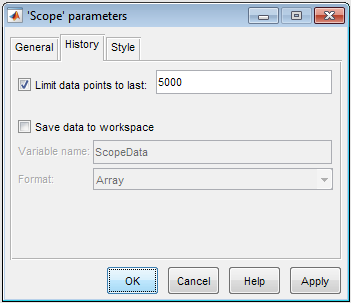
* **Number of axes -**  այս պարամետրի փոփոխմամբ կարող ենք ավելացնել օսցիլոգրաֆին լրացուցիչ մուտքեր:
* **Time range** - ժամանակիայն ինտերվալն որի ընթացքում արտածվում է գրաֆիկը:

Եթե մոդելի հաշվարկային ժամանակը դուրս է տրված պարամետրով սահմանված միջակայքից, ապա գրաֆիկը արտածվում է մաս մաս: Այս դեպքում գրաֆիկի մաս մաս արտածման ժամանակային ինտերվալը հավասար է Time range-ում սահմանված միջակայքի թվային արժեքին:

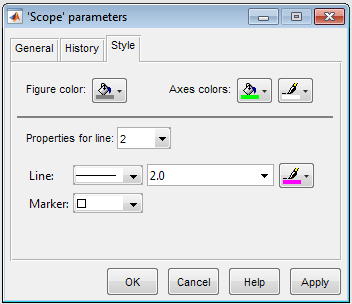
* **Tick labels -** Ցույց տալ / թաքցնել առանցքները և առանցքների պիտակները,
* **Sampling**  - պատուհանում գրաֆիկի ցուցադրման պարամետր: Այս պարամետրը սահմանում է էկրանին հաշվարկված կետերը ցուցադրելու ռեժիմը: Երբ Decimation-ն ընտրվում է, ելքային կետերի բազմությունը սահմանվում է մի թվով, որը հանդիսանում ելքային կետերի արտածման քայլը:

### History ներդիրում տեղադրված պարամետրերը

* **Limit data points to last** – Գրաֆիկի արտածվող առավելագույն կետերի քանակը: Եթե այս թիվը գերազանցում է, ապա գրաֆիկի սկզբնական մասը կկտրվի: Եթե Limit data points to lastդրոշակը դրված չէ, ապա Simulink-ը ավտոմատ կարտածի բոլոր հաշվարկված կետերը:
* **Save data to workspace –** եթե դրված է դրոշակը հաշվակային արժեքներըկպահպանվեն MATLAB միջավայրի աշխատանքային տարածքում:

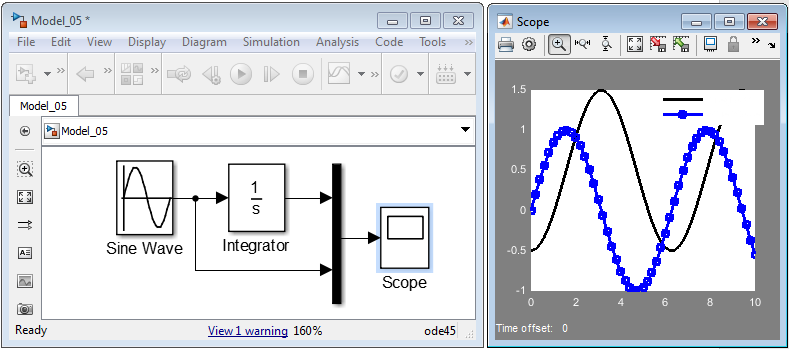


### Style ներդիրում տեղադրված պարամետրերը



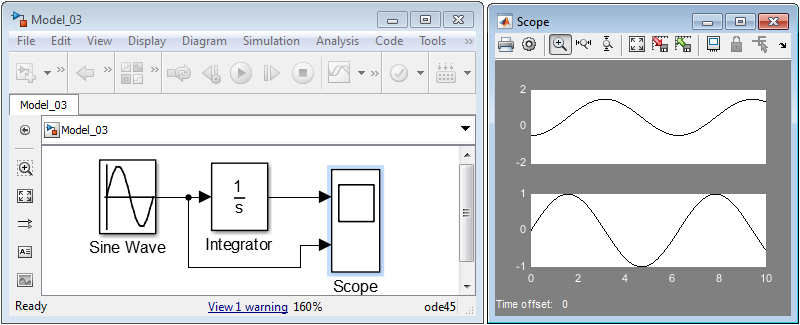
* **Figure Color** – գրաֆիկի գծի գույնը,
* **Axes Colors –** առաջին գործիքով կարող ենք փոխելգրաֆիկի պատուհանի գույնը, իսկ երկրորդով առանցքների գույնը,
* **Properties for line –** ընտրել գրաֆիկական պատուհանի գրաֆիկի համարը ձևափոխություն կատարելու համար,
* **Line –** ընտրել գծի տեսակը, հաստությունը և գույնը,
* **Marker –** ընտրել մարկերի տեսքը գրաֆիկում նշումներ կատարելու համար:

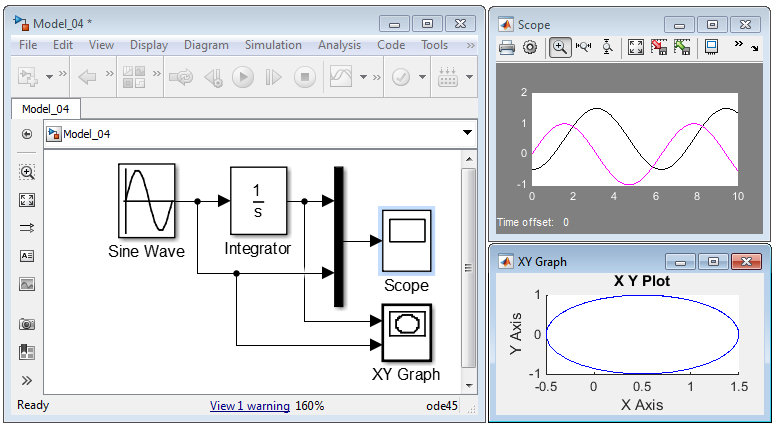
Մի քանի ազդանշանների գրաֆիկները համեմատելու համար կարող ենք նույն կոորդինատական հարթության վրա արտածել գրաֆիկները օգտագործելով ուժեղացուցիչ բլոկը (Mux բլոկ, ազդանշանային երթուղղման բաժին **Signal Routing**), որը միավորում է մի քանի ազդանշաններ վեկտորի մեջ: Ստորև բերված է սխեման:



Եթե վեկտորի ազդանշանը հասնում է Scope բլոկի մուտքին, ապա վեկտորի յուրաքանչյուր տարրի գրաֆիկը արտածվում է առանձին գույնով:

Արտածենք ֆունկցիայի  և առաջին կարգի ածանցյալի  գրաֆիկները նույն կոորդինատական հարթության վրա, օգտագործելով (ազդանշանային երթուղղավորման մուլտիպլեքսերը) Signal Routing | Mux, միացնելով մի քանի ազդանշաններ վեկտորի մեջ: Կառուցենք նաև ֆունկցիայի ֆազային հետագծի գրաֆիկը  հարության մեջ, օգտագործելով Sinks պանակի XYGraph բլոկը:





## Math Operations, Logic և Bit Operations բաժինների ակնարկ

Մաթեմատիկական, տրամաբանական և բիթային գործողություններ կատարելու համար անհրաժեշտ բլոկները տեղակայված են հետևյալ բաժիններում.

* **Math Operations -** մաթեմատիկական գործողությունների բլոկներ,
* **Logic and Bit Operations –** տրամաբանական և բիտային գործողությունների բլոկներ:

Ստորև բերված են մի քանի բլոկների համառոտ մեկնաբանությունը.

**Gain** – բլոկը բազմապատկում է մուտքի ազդանշանը ըստ տրված գործակցի և ընդունվում է, որպես ուժեղացուցիչ,

**Product -**  բլոկը բազմապատկում է երկու կամ ավելի ազդանշաններ,

**Math Function -** բլոկը ձևափոխում է մուտքի ազդանշանը ըստ տրված ֆունկցիոնալ տեսքի (օրինակ՝ բարձրացնել քառակուսի, հանել քառակուսի արմատ, հաշվել լոգարիթմը և այլն):

**MinMax** - էքստրեմումի հաշվարկման բլոկ,

**Rounding Function** – թվերի կլորացման բլոկ,

**Sign** – թվի նշանի որոշման բլոկ,

**Sum** – գումարման բլոկ, կարող է լինել շրջանի տեսքով, կամ ուղանկուն: Տեսքը փոխելու համար մկնիկի կրկնակի հարված կատարել բլոկի վրա, բացված պատուհանում փոխել Icon Shape:

**Trigonometric Function** - եռանկյունաչափական ֆունկցիաների բլոկ,

**Matrix Gain** - մատրիցային բազմակատկման բլոկ,

**Abs** - մոդուլի բլոկ,

**Algebraic Constraint** - ոչ գծային հավասարումների համակարգի լուծման բլոկ,

**Logical Operator(And)** - Տրամաբանական գործողությունների բլոկ,

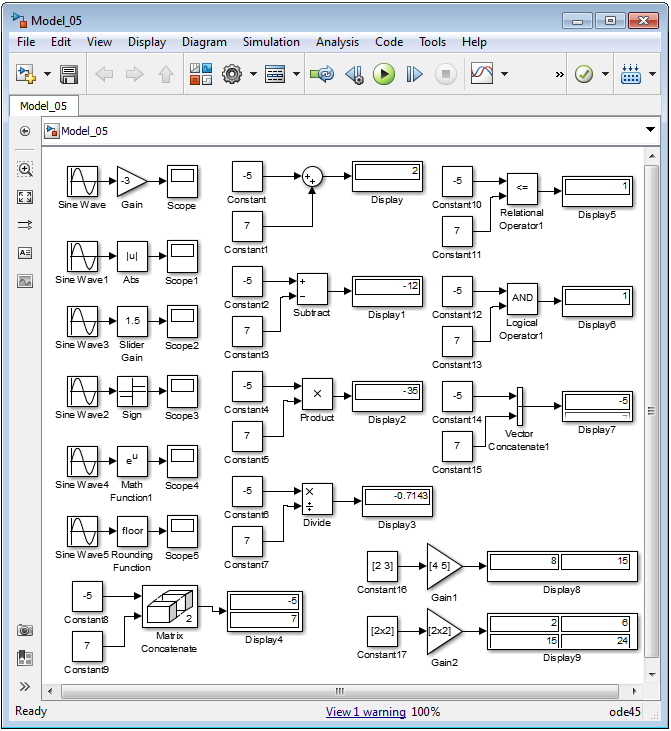
**Complex to Real-Imaag** – կոմպլեքս թվի իրական և կեղծ մասի բաժանման բլոկ,

**Real-Image to Complex** – դեկարտյան կոորդինատներից կոմպլեքսի անցման բլոկ,

**Slider Gein** – տրված քայլով միջակայքի թվերի փոխանցման բլոկ:

Բլոկների դիրքերի շրջման համար, մկնիկի աջ հարվածով բացել բլոկի հատկությունների Explore պատուհանը և Rotate&Flip մենյուից ընտրել դիրքը: Բլոկների անվանումների ցուցադրման համար մկնիկի աջ հարվածով բացել բլոկի հատկությունների Explore պատուհանը և

Format մենյույում ակտիվացնել Show Block Name-ը:



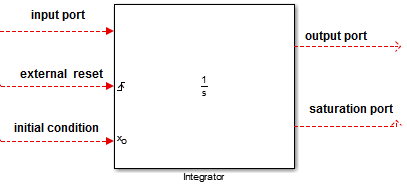
Մաթեմատիկական գործողությունների աղբյուրներ

### 

**Դաս 4**

### Բլոկ Integrator

Continuous բաժնի բլոկները ձևափոխում են անընդհատ ազդանշանները(**continuous states**): Աայս բաժնի Integratorբլոկը ինտեգրում է մուտքային ազդանշանը (նկար 16):



նկ.16. Բլոկ Continuous | Integrator

* + **External reset** – արտաքին ղեկավարվող ազդանշան է, որը ինտեգրատորը վերադարձնում սկզբնական վիճակին: Ստորև բերված են այն պարամետրերը, որը կարող է ընդունել External reset-ը.
* **none** –վերադարձ չի կատարվում,
* **rising** – աճող ազդանշան (վերբեռնման ազդանշանը կառավարման մուտքում զրոյից բարձրանում է դրական կամ բացասականից դեպի դրական արժեք),
* **falling** – նվազող ազդանշան (վերբեռնման ազդանշանը կառավարման մուտքում նվազում է դրականից միջև զրո կամ դրականից բացասական արժեք ),
* **either** – աճող կամ նվազող ազդանշան (վերբեռնման ազդանշանը կառավարման մուտքում փոփոխվում է զրոյականից ոչ զրոյական արժեք ),
* **level** – ոչ զրոյական ազդանշան (վերբեռնումը կատարվում է, եթե ազդանշանը կառավարման մուտքում զրոյական է ժամանակի ընթացիկ պահին կամ ժամանակի նախորդ քայլում ոչ զրոյական արժեքից փոփոխվում է զրոյի ընթացիկ ժամանակում  ),
* **level hold** – ոչ զրոյական ազդանշան (վերբեռնումը կատարվում է, եթե վերաբեռնման ազդանշանը կառավարման մուտքում ոչզրոյական է ժամանակի ընթացիկ պահին  ),

Նշենք, որ եթե ընտրված է կառավարման ազդանշանը, բացառությամբ **none** ընտրության, բլոկի պատկերի վրա հայտնվում է լրացուցիչ մուտք, որի կողքին ցուցադրվում է կառավարման ազդանշանի պայմանական տեսքը` ,,, կամ :

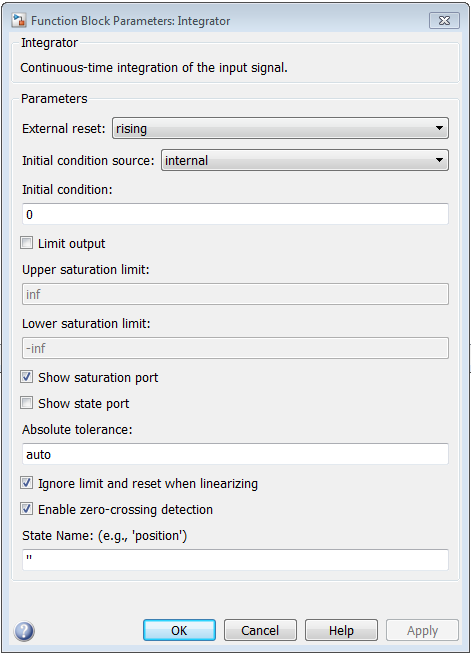
* **Initial condition source -** ելքային ազդանշանի սկզբնական արժեքի աղբյուրը, որը ընտրվում է ցուցակից.
* **Initial** – ներքին,
* **External –** արտաքին: Բլոկի վրա հայտնվում է լրացուցիչ մուտք x0 նշանակումով, որին տրվում է ազդանշան, նշելով ինտեգրատորի ելքի սկզբնական արժեքը:
* **Initial condition** – սկզբնական պայման: Տրվում է ինտեգրատորի ելքային ազդանշանի x0 սկզբնական արժեքը: Պարամետրը հասանելի է, եթե ընտրված է ելքային ազդանշանի սկզբնական արժեքի ներքին աղբյուրը:
* **Limit output (դրոշակ)** - ելքային ազդանշանի սահմանափակում: Եթե դրոշակը դրված է, ապա հասանելի են դառնում հետևյալ երկու պարամետրերը:
* **Upper saturation limit -** ելքային ազդանշանի վերին սահման: Այն կարող է տրվել թվով կամ inf ( լռելյայն + ∞):
* **Lower saturation limit** -ելքային ազդանշանի ներքին սահման: Այն կարող է տրվել թվով կամ inf ( լռելյայն - ∞):
* **Show saturation port** – ղեկավարում է ելքային ազդանշանի հանգույցը.
* 0, եթե ինտեգրատորը արգելափակված չէ,
* +1, եթե ինտեգրատորի ելքային ազդանշանը հասել է վերին սահմանին,
* -1, եթե ինտեգրատորի ելքային ազդանշանը հասել է ներքին սահմանին:
* **Show state port** (դրոշակ) - ցույց տալ / թաքցնել բլոկի ելքային հանգույցի վիճակը: Ելքային հանգույցի վիճակի պարամետրը օգտագործվում է, եթե ինտեգրատորի ելքային ազդանշանը հանդիսանում է նույն ինտեգրատորի մուտքային ազդանշան:
* **Absolute tolerance** - բացարձակ սխալ:

Integrator բլոկը հաշվում է



մեծությունը: Լռելյայն :

Ստորև բերված է ինտեգրատորի պարամետրերի պատուհանը:



## . Գրաֆիկների կառուցումը Simulink միջավայրում

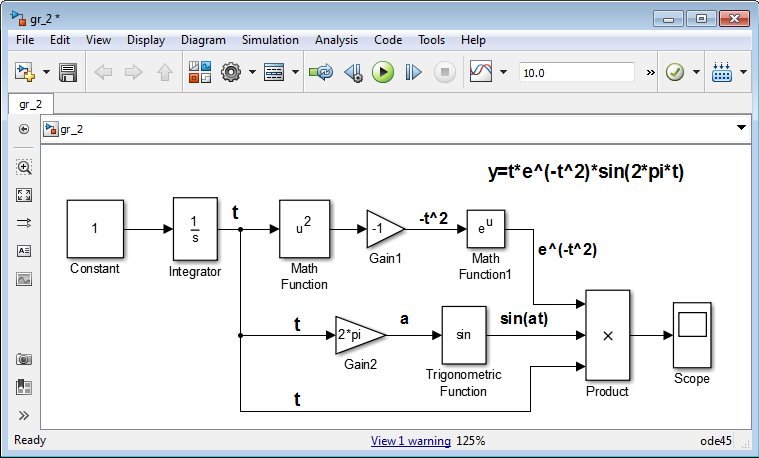
**Օրինակ 1:** Կառուցել ֆունկցիայի գրաֆիկը:

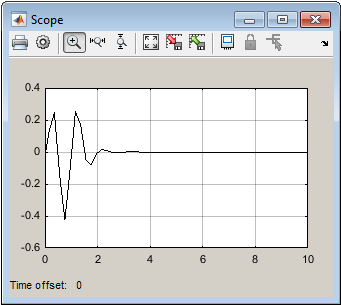
Բացեք Simulink գրադարանի պատուհանը, որտեղից կվերցնենք մոդելի բլոկները: Ստեղծեք նոր S-մոդել առանձին պատուհանում և այն գրանցենք gr\_2.mdl անունով ֆայլում:

Գրաֆիկի կառուցման սխեման.

* տեղադրենք **Sources | Constant** բլոկը հաստատունի արժեքը ընդունելով 1,
* տեղադրենք **Continuous | Integrator** ինտեգրման բլոկը, նշենք, որ հաստատունը ինտեգրելով կստանանք t ազդանշանը,
* առաջին ճյուղով ստանանք  ազդանշանը,
* **Math Operations | Math Function** բլոկի կիրառմամբ ստանանք,
* ստացված արդյունքը **Math Operations | Gain1** ուժեղացուցիչի միջոցով բազմապատկենք **-**1-ով**,**
* **Math Operations | Math Function1** բլոկի կիրառմամբ ստանանք ,
* երկրորդ ճյուղով ստանանք  ազդանշանը,
* տեղադրենք **Math Operations | Gain2** ուժեղացուցիչը, որի միջոցով ստանանք 2pi թիվը,
* **Math Operations | Trigonometric Function** բլոկի կիրառմամբ հաշվենք,
* երկրորդ ճյուղով ստանանք t ազդանշանը,
* **Commonly Used Blocks | Product** բլոկի միջոցով բազմապատկենք երեք ազդանշանները,
* Արդյունքը փոխանցենք **Sinks |** **Scope** բլոկներին գրաֆիկի արտածման համար:

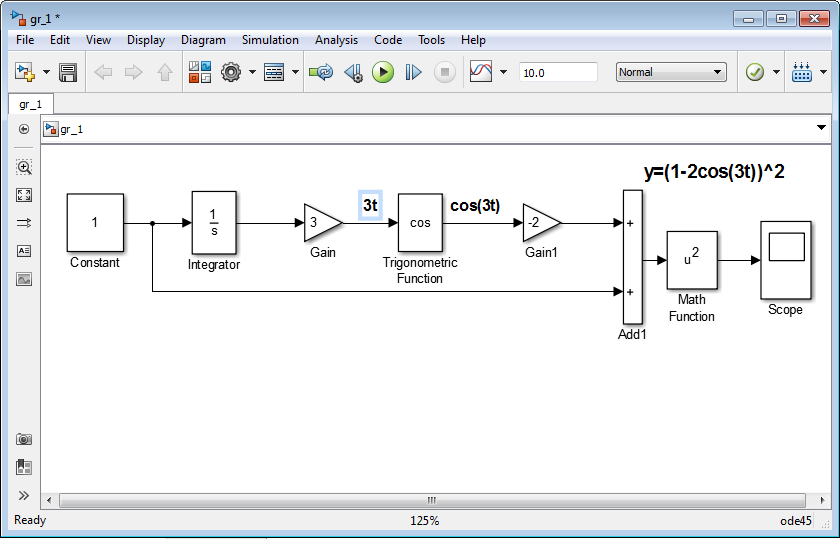
Օգտագործել հետևյալ բլոկները` Constant, Integrator, Gain, Trigonometric Function, Math Function, Product և Scope:

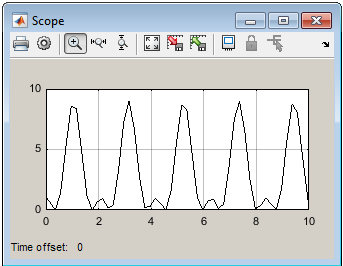




**Օրինակ 2**: Կառուցել  ֆունկցիայի գրաֆիկը:

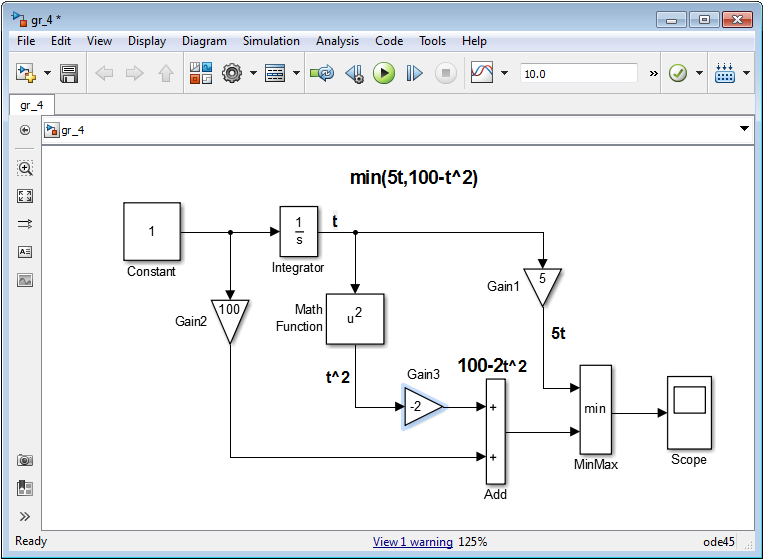
**Ցուցում:** Օգտագործել հետևյալ բլոկները` Constant, Integrator, Gain, Trigonometric Function, Math Function, Sum և Scope:

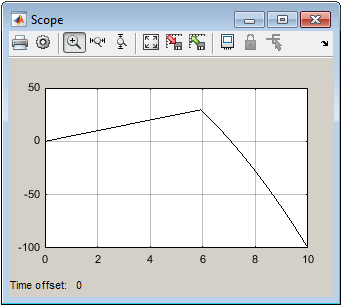




**Օրինակ 3:** Գրաֆիկորեն որոշել` և ֆունկցիաների մինիմումը:

**Ցուցում:** Օգտագործել հետևյալ բլոկները` Constant, Integrator, Gain, Math Function, Sum, MinMax և Scope:





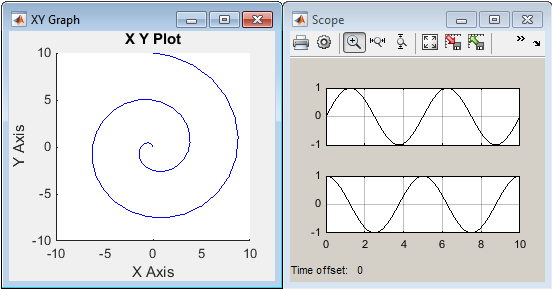
**Օրինակ 4:** Կառուցել   ֆունկցիաների գրաֆիկը:

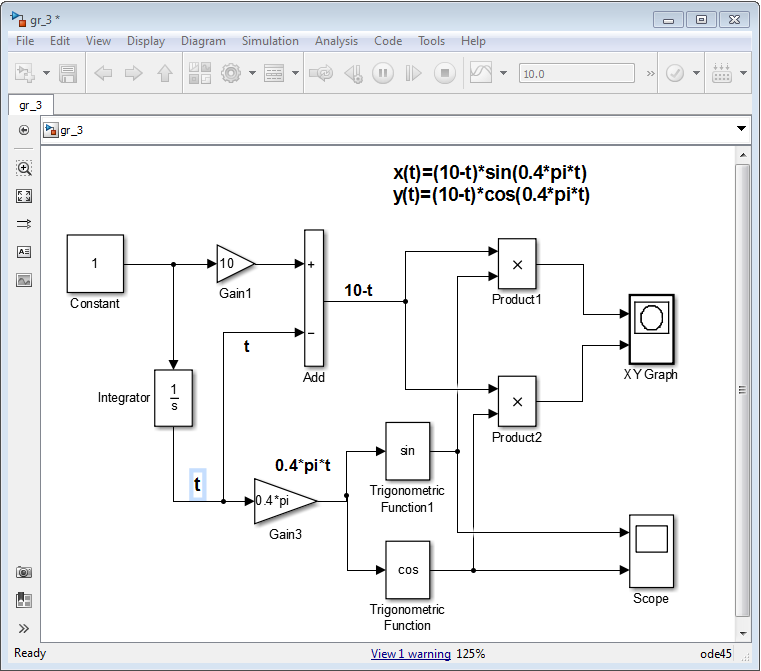
Օգտագործենք հետևյալ բլոկները` Сonstant, Integrator, Gain, Trigonometric Function, Product, Scope, XY Graph:

Բացեք Simulink գրադարանի պատուհանը, որտեղից կվերցնենք մոդելի բլոկները: Ստեղծեք նոր S-մոդել առանձին պատուհանում և այն գրանցենք gr\_3.mdl անունով ֆայլում:

Գրաֆիկի կառուցման սխեման.

* տեղադրենք **Sources | Constant** բլոկը հաստատունի արժեքը ընդունելով 1-ը,
* սխեմայի վերի ճյուղում **Math Operations | Gain1** ուժեղացուցիչի կիրառմամ ազդանշանը ուժեղացնենք 10 անգամ, իսկ երկրորդ ներքևի ճյուղում **Continuous | Integrator** ինտեգրման բլոկի կիրառմամբ ինտեգրելով հաստատունը ստանանք t ազդանշանը,
* **Math Operations | Add** բլոկի կիրառմամբ ստանանք 10**-** t ազդանշանը,
* **Math Operations | Gain3** ուժեղացուցիչի կիրառմամ հաշվենք 0.4pit ազդանշանը և այն փոխանցենք **Math Operations | Trigonometric Function** բլոկներին sin0.4pit և cos0.4pit արտահայտությունները հաշվելու համար:
* **Commonly Used Blocks | Product1 և Product2** բլոկների միջոցով հաշվենք (10-t)sin0.4pit և (10-t)cos0.4pit-ն և այն փոխանցենք XYGraph-ին գրաֆիկի կառուցելու համար,
* sin0.4pit և cos0.4pit-ն արժեքները փոխանցենք **Sinks |** **Scope-**ին նրանց կախվածության գրաֆիկները կառուցելու համար,





**Դաս 5**

## Առաջին կարգի դիֆերենցիալ հավասարման և հավասարումների համակարգի լուծումը Simulink միջավայրում

### 1. Առաջին կարգի դիֆերենցիալ հավասարման լուծումը

MatLab համակարգը գործարկելուց հետո ակտիվացրեք Simulink կոճակը, այնուհետև բացվող պատուհանում ստեղծեք նոր մոդել: Բացվող ֆայլում ստեղծեք հավասարման լուծման սխեմա` Simulink Library Brouser պատուհանից մկնիկի ձախ կոճակը սեղմելով անհրաժեշտ բլոկները տեղափոխել աշխատանքային պատուհան:

Մեկ փոփոխականով առաջին կարգի սովորական դիֆերենցիալ հավասարումը Կոշի խնդիրն է, որի տեսքն է.



**Օրրինակ**: Լուծել դիֆերենցիալ հավասարումը.

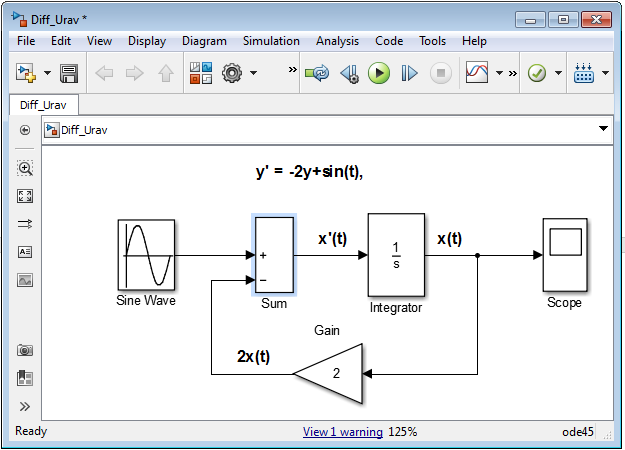


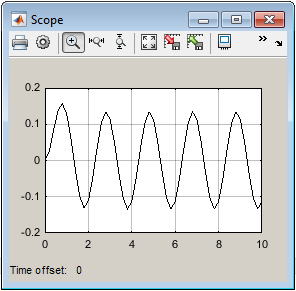
**Simulink**-ում հավասարման լուծման սխեման կառուցելու համար օգտագործվում է **Integrator** բլոկը(**Continuos** դաս):

Մուտքի վրա տրվում է y’ ածանցյալը, իսկ ելքի վրա ստացվում է y-ը:

Օգտագործենք **Blocks Sum**-ը (գումարման) և **Gain** (ուժեղացուցիչ) **Math** դասից ձևավորելու համար x' ածանցյալը:

sin(t) ազդանշանը ստանալու համար օգտագործենք **Sine Wave** բլոկը (**Sources** դաս), որում անհրաժեշտ է իրականացնել խնդրին համապատասխան պարամետրերի փոփոխություն, բլոկի վրա մկնիկի կրկնակի սեղմումով կամ ընտրելով «**Block Parameters**»-ը սեղմելով մկնիկի աջ կոճակ: Արդյունքում ստացված y արժեքները տրվում են **Scope** բլոկի մուտքին: Այս բլոկի վրա մկնիկի սեղմումով կբացվի լուծման գրաֆիկը: Կարող ենք առանցքի մասշտաբները փոփոխել **Autoscale** կոճակը սեղմելով:





### 2 Վան Դեր Պոլի հավասարման Simulink մոդելը

Այժմ օգտվելով **ode45** սոլվերից լուծենք Վան Դեր Պոլի 2-րդ կարգի դիֆերենցիալ հավասարումը, որն ունի հետևյալ տեսքը,

,

որտեղ -ն պարամետր է:

Դիֆերենցիալ հավասարումը բերենք առաջին կարգի դիֆերենցիալ հավասարումների համակարգի:

Նշանակենք՝ ;  կստանանք`

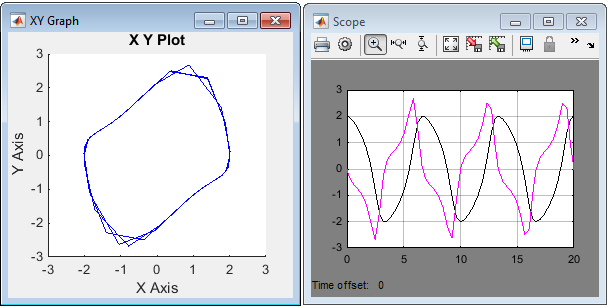
, որտեղ` 

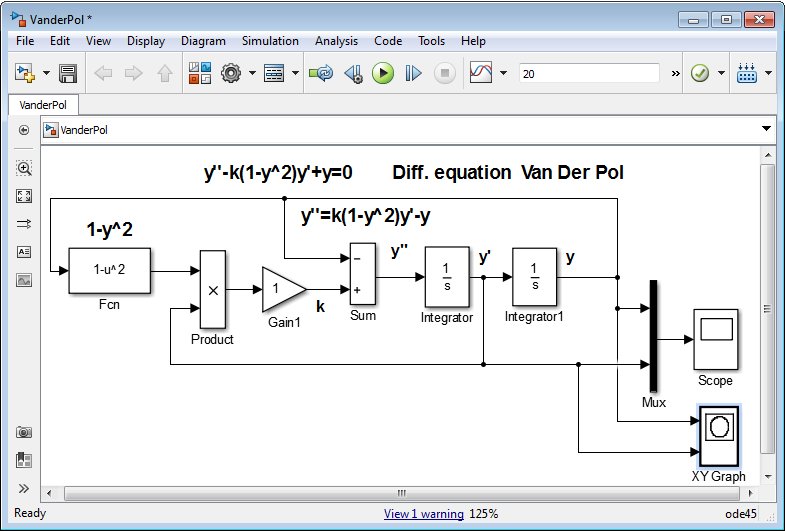
Բացեք Simulink գրադարանի պատուհանը, որտեղից կվերցնենք մոդելի բլոկները: Ստեղծեք նոր S-մոդել առանձին պատուհանում և այն գրանցենք Vander\_Pol.mdl անունով ֆայլում:

* Տեղադրենք ինտեգրման երկու բլոկր, որոնք կհաշվեն y-ի առաջին և երկրորդ կարգի ածանցյալները: Օգտագործեք **Continuous | Integrator** բլոկը: Մկնիկի կրկնակի հարվածով բացենք ինտեգրատորների պարամետրերը և **Initial condition** դաշտում մոտքագրել սկզբնական պայմանի արժեքները **(**Integrator. Initial condition**=**0, , Integrator1. Initial condition**=**2, ):Առաջինի ելքը միացնենք երկրորդի մուտքին:
* Օգտագործենք **User-Defined | Fcn** բլոկը, որի **Expression** պարամետրի դաշտում ներմուծենք  արտահայտությոնը:
* **Commonly Used Blocks | Product** բլոկը, որի միջոցով կհաշվենքարտահայտությունը:
* Ստացված արդյունքը **Math Operations | Gain** ուժեղացուցիչի միջոցով բազմապատկենք-ով**, կստանանք`** **:**
* **Math Operations | Add** կամ **Math Operations | Sum** բլոկի կիրառմամբ հաշվենք  արտահայտության արժեքը:

Արդյունքները փոխանցվում **Signal Routing | Mux** բլոկին, որը ձևավորում է մատրից` կազմված ֆունկցիայի, ածանցյալ ֆունկցիայի և t-իարժեքներից, որը փոխանցվում է **Sinks |** **Scope** բլոկներին գրաֆիկի արտածման համար: **XYGraph** փոխանցվում են ֆունկցիայի, ածանցյալ ֆունկցիայի արժեքների վետորները ֆունկցիայի ֆազային գրաֆիկը կառուցելու համար:

* Գործարկեք S- մոդելը և ուսումնասիրենք արդյունքները:





Այժմ բերենք խնդրի լուծումը ode45 և ode15s սոլվերների կիրառմամբ:

Ստեղծենք diffeqVand.m (function) ֆայլ ֆունկցիան, որում կգրենք համակարգի աջ մասի վեկտորը:

function F = diffeqVand(t,x)

global k

F = [x(2); k\*x(2)\*(1-x(1)^2)-x(1)];

end

Ստեղծենք Vandfun.m (script) ֆայլը և գրենք ծրագրային կոդը, նշենք որ խնդիրը կարող ենք լուծել տարբեր սոլվերներով:

clc; clear;

global k

y0=[2 0]; k=1;

[t, y]=ode45(@diffeqVand,[0 20],y0);

subplot(2,1,1)

plot(t, y(:,1),'k','linewidth',1)

grid on, xlabel('t'), ylabel('y')

[t, y] = ode15s(@diffeqVand, [0 20], y0);

subplot(2,1,2),

plot(t, y(:,1),'k-.','linewidth',1)

grid on

xlabel('t')

ylabel('y')

### 3 Գծային հավասարումների և դիֆերենցիալ հավասարումների համակարգի

### լուծումը Simulink միջավայրում

Որոշ կիրառական խնդիրների մոդալավորումը հանգում է սովորական դիֆերենցիալ հավասարումների համակարգի, որի լուծումը տալիս նաև հանրահաշվական խնդրի լուծումը:

Դիցուք ոչ իներցիոն օբեկտը բնութագրվում է հետևյալ հավասարումների համակարգով.

: (c)

Օբյեկտի մոդելը կառուցելու համար, տրված համակարգը ներկայացնենք դիֆերենցիալ հավասարումների հետևյալ համարժեք համակարգով.

: (a)

Հենց բոլոր ածանցյալները  մարում են, արդյունքում կստանանք լուծման  ազդանշանը:

Այս երկու հավասարումների համակարգերի համարժեքության համար պետք է որոշվի դիֆերենցիալ հավասարումների համակարգի զրոյական լուծումը: Մարող լուծման համար բավարար պայման է հավասարումների գծային համակարգի գործակիցային մատրիցայի հետևյալ պայմանը`



Մասնավոր դեպքում դիտարկենք հավասարումների հետևյալ համակարգը.

:

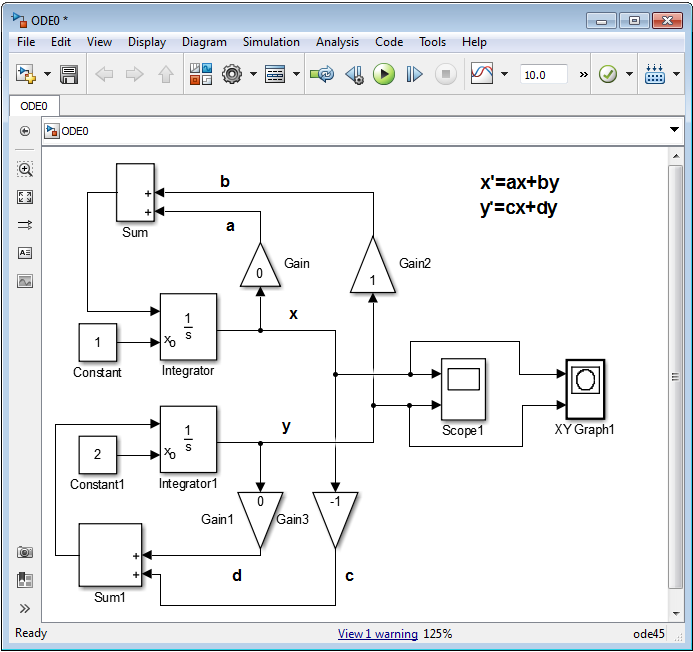
Բացեք Simulink գրադարանի պատուհանը, որտեղից կվերցնենք մոդելի բլոկները: Ստեղծեք նոր S-մոդել առանձին պատուհանում և այն գրանցենք System\_1.mdl անունով ֆայլում:

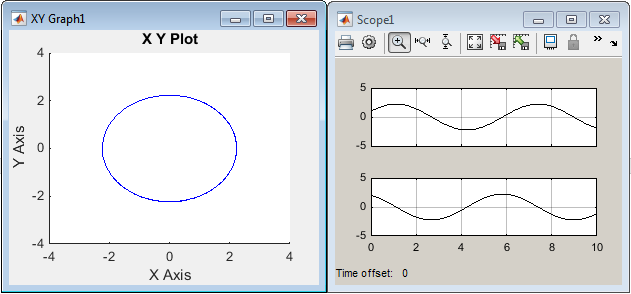
**Լուծում:** Տեղադրենք ինտեգրման **Continuous | Integrator** երկու բլոկ, որոնք կհաշվեն x և y-ի առաջին կարգի ածանցյալները: Առաջինը անվանենք **Integrator\_x**, երկրորդը` **Integrator\_y**: Ինտեգրման բլոկների պարամետրերում նշենք, որ սկզբնական պայմանը ըստ x և y-ի սահմանվելու է դրսից: Դա համար **Initial condition source** պարամետրը դնենք **external**: Այս դեպքում բլոկները կունենան լրացուցիչ x0 մուտք: Առաջին ինտեգրատորի x0 մուտքից մուտքագրենք սկզբնական պայմանը ըստ x-ի, իսկ երկրորդով ըստ y-ի:

Ավելացնել երկու բլոկ **Sources | Constant** բլոկներ x0 և y0 անուներով: Ընդունենք սկզբնական պայմանները x (0) =1 և y (0) =2:

Միացնենք x0 և y0 բլոկների ելքերը ինտեգրատորների մուտքերին:

* **Math Operations | Gain1** ուժեղացուցիչի միջոցով ստանանք **ax** անդամը և **Integrator\_y** բլոկի y ազդանշանը բազմապատկենք **Gain2**-ի արժեքով և ստացված արդյունքները գումարենք իրար **Math Operations | Sum1** բլոկի կիրառմամբ կստանանք **x'=ax+by** արդյունքը**:**
* **Math Operations | Gain3** ուժեղացուցիչի միջոցով ստանանք **dy** անդամը և Integrator\_x բլոկի x ազդանշանը բազմապատկենք **Gain4**-ի արժեքով և ստացված արդյունքները գումարենք իրար **Math Operations | Sum2** բլոկի կիրառմամբ կստանանք **y'=cx+dy** արդյունքը**:**
* **Sinks |** **Scope** բլոկի կիրառմամբկառուցենք x(t) և y(t) լուծումների գրաֆիկները կախված t փոփոխականից:
* **Sinks | XY Graph1** բլոկի կիրառմամբկառուցենք x(t)-ի կախվածության գրաֆիկը y(t)-ից:



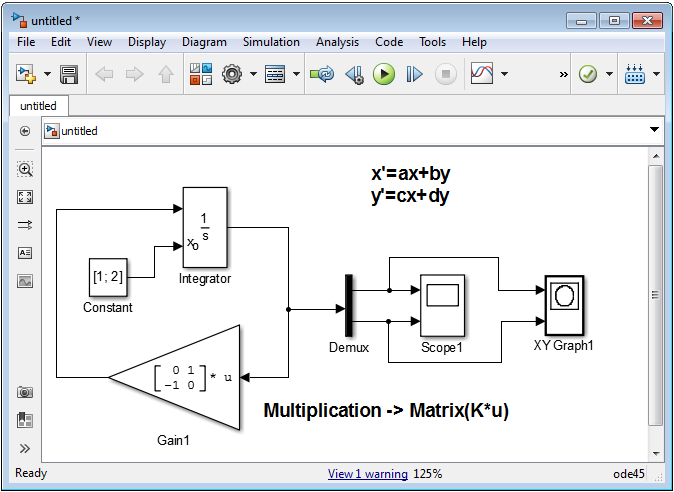


**Խնդրի լուծումը մատրիցային եղանակով**

Բացեք Simulink գրադարանի պատուհանը, որտեղից կվերցնենք մոդելի բլոկները: Ստեղծեք նոր S-մոդել առանձին պատուհանում և այն գրանցենք System\_1.mdl անունով ֆայլում:

**Լուծում:**

* Տեղադրենք ինտեգրման **Continuous | Integrator** երկու բլոկ, որը կհաշվի x և y-ի առաջին կարգի ածանցյալները: Ինտեգրատորի պարամետրերում նշենք, որ սկզբնական պայմանը ըստ x և y-ի սահմանվելու է դրսից: Դա համար **Initial condition source** պարամետրը դնենք **external**: Ավելացնենք **Sources | Constant** բլոկըսկզբնակաան պայմանի վեկտորը ներմուծելու համար [1; 2]:
* **Math Operations | Gain1** ուժեղացուցիչի միջոցով մուտքագրենք համակարգի աջ մասի փոփոխականների գործակիցների [a,b;c,d] մատրիցը: **Gain1** ուժեղացուցիչի **Multiplication** պարամետրը դնել **Matrix(K\*u):** Միացնել **Integrator-**ի ելքը **Gain1**-ի մուտքին, **Gain1**-ի ելքը **Integratorի** մուտքին: Արդյունքները փոխանցենք **Demux**-ին, այնուհետև **Scope** և **XY Graph1** բլոկներին գրաֆիկը կառուցելու համար:



**Օրինակ:** Գտնել գծային հավասարումների համակարգի լուծումը.

:

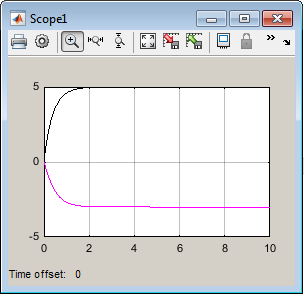
Լուծում: Համակարգը ներկայացնենք (a) տեսքով.

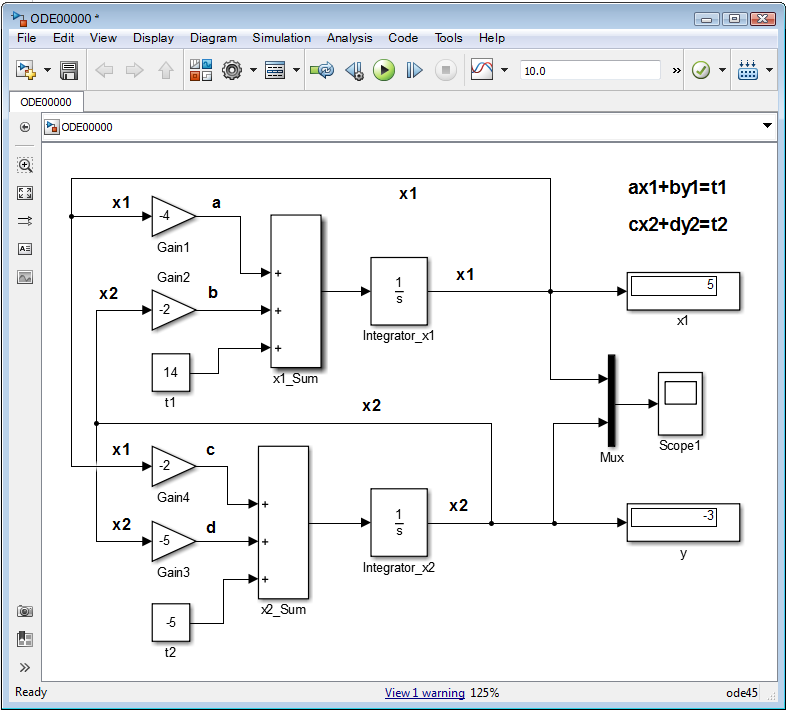
,

որտեղից` կստանանք.

:

**Լուծում:** Ստորև բերված է համակարգի լուծումը և լուծման սխեման: Օգտագործվել է երկու Display բլոկներ հավասարման լուծումները արտածելու համար, ինտեգրման երկու բլոկ, գումարի հաշվման երկու բլոկ, ազդանշանի չորս ուժեղացուցիչ և Mux բլոկը ազդանշանի բաժանման համար:





**Օրինակ:** Գծային հավասարումների (c) համակարգը ներկայացնենք հետևյալ տեսքով.



և մասնավոր դեպքում լուծենք համակարգը օգտագործելով **Algebraic Constrain** բլոկը.

 :

Բացեք Simulink գրադարանի պատուհանը, որտեղից կվերցնենք մոդելի բլոկները: Ստեղծեք նոր S-մոդել առանձին պատուհանում և այն գրանցենք Solve\_3.mdl անունով ֆայլում:

Համակարգի ընդհանուր լուծման սխեմայի կառուցման կամար կատարենք հետևյալ քայլերը.

* Տեղադրենք երեք **MathOperations | Algebraic Constraint** բլոկները` համակարգի հավասարումների լուծման համար,
* **Gain1-Gain4** բլոկների միջոցով ձևավորենք համակարգի առաջին հավասարումը,
* **Gain4-Gain8** բլոկների միջոցով ձևավորենք համակարգի երկրորդ հավասարումը,
* **Gain9-Gain12** բլոկների միջոցով ձևավորենք համակարգի երրորդ հավասարումը,
* **Sum1-Sum3** բլոկների ելքում կստանանք համակարգի առաջին`, երկրորդ և երրորդ հավասարումների ազդանշանները, որոնք փոխանցվում են համապատասխանաբար` **Algebraic Constraint1-Algebraic Constraint2** և **Algebraic Constraint3**-ին համակարգի լուծման համար,
* արդյունքում, լուծումները փոխանցվում են **Mux** բլոկին, որը տարանջատում է արմատները **Display** բլոկում արտածման համար:

Ստորև բերված է լուծման սխեման և արդյունքը:

