

UNIVERZA V NOVI GORICI  
POSLOVNO-TEHNIŠKA FAKULTETA

DIPLOMSKO DELO

**UPORABA PROGRAMSKEGA MODULA SCICOS ZA GOSPODARSKEGA  
INŽENIRJA**

Aleš Bajc

Mentor: prof. dr. Juš Kocijan

Nova Gorica, 2007



## **IZVLEČEK**

Diplomsko delo vsebuje priročnik programskega modula Scicos za uporabnika začetnika. Scicos je modul, ki je del programa odprte kode Scilab in ga lahko dobimo brezplačno na internetu. Scicos je zelo napredno orodje za dinamično simulacijo sistemov.

Priročnik je napisan za uporabnika začetnika in ne zahteva nikakršnega programerskega predznanja. V priročniku so opisani koraki, ki bodo vodili študente skozi postopek izvedbe simulacije dinamičnih sistemov. Priročnik najprej opisuje, kako pridobiti in namestiti program, nato pa sledi opis funkcij modula Scicos. V nadaljevanju smo z uporabo konkretnih primerov sestavili projekte, jih modelirali in simulirali. Z uporabo tega priročnika bo študent sposoben modelirati in simulirati raznovrstne dinamične sisteme.

## **KLJUČNE BESEDE**

Scicos modul, programski paket Scilab, simulacija dinamičnih sistemov, simulacijsko orodje

## **ABSTRACT**

Scicos manual for beginners is described in the diploma thesis. Scicos is the part of Scilab open code program which can be downloaded from the internet free of charge. Scicos is a very advanced tool for dynamic system simulation.

This manual is made for users that do not have prior programming skills. Step by step from the first chapter to the end, this manual guides students through the procedure of dynamic process simulation. At the beginning the manual describes how to obtain Scilab from the internet and the installation procedure of the Scicos module. A brief introduction of the program is followed by a short description of each function. Next it is explained how to build projects on selected examples and how to simulate them. Using this manual, students will be able to construct dynamic system models and simulate them.

## **KEY WORDS**

Scicos module, programming package Scilab, dynamic system simulation, simulation tool

## KAZALO VSEBINE

IZVLEČEK .....	III
KLJUČNE BESEDE .....	III
ABSTRACT .....	IV
KEY WORDS .....	IV
UVOD .....	1
1. PROGRAMSKI MODUL SCICOS .....	3
2. ZAGON PROGRAMA .....	5
3. MENIJI PROGRAMA .....	7
4. OSNOVNI GRADNIKI .....	9
4.1 Bloki signalnih virov .....	9
4.2 Bloki signalnih ponorov .....	10
4.3 Bloki, ki predstavljajo linearne gradnike sistemov .....	11
4.4 Bloki, ki predstavljajo nelinearne gradnike sistemov .....	12
5. OSNOVE DELA S PROGRAMOM .....	15
5.1 Postavitev gradnikov .....	15
5.2 Povezovanje gradnikov .....	18
5.3 Spreminjanje parametrov gradnikov .....	19
5.4 Simulacija .....	20
5.5 Odpiranje in shranjevanje projektov .....	23
5.5.1 Shranjevanje projekta .....	23
5.5.2 Shranjevanje projekta kot slike .....	25
5.5.3 Odpiranje projekta .....	28
5.6 Super Block (Nadblok) in Scifunc .....	28
5.6.1 Super Block (nadblok) .....	28
5.6.2 Scifunc blok .....	33
6. SIMULACIJA RAZLIČNIH VRST MODELOV .....	37
6.1 Linearna diferencialna enačba .....	37
6.2 Nelinearna diferencialna enačba .....	40
6.3 Sistem diferencialnih enačb .....	42
6.4 Prenosna funkcija .....	46
6.5 Modularno sestavljeni sistemi .....	49
7. PRIMERI UPORABE .....	53
7.1 Elektrotehnika: Odziv RC vezja .....	53
7.2 Mehanika: Poenostavljeno avtomobilsko vzmetenje .....	59
7.3 Marketing: Simulacija propagandne akcije .....	65
7.4 Avtomatsko vodenje: Simulacija sistema regulacije temperature .....	69
8. ZAKLJUČEK .....	75
9. LITERATURA .....	77

## KAZALO SLIK

Slika 1: Namizje .....	5
Slika 2: Zagon programa Scilab .....	5
Slika 3: Osnovno okno programa Scicos.....	6
Slika 4: Bloki signalnih virov .....	9
Slika 5: Bloki signalnih ponorov .....	10
Slika 6: Bloki linearnih gradnikov.....	11
Slika 7: Bloki nelinearnih gradnikov.....	12
Slika 8: Izbira objektov .....	15
Slika 9: Izbira izvorov .....	15
Slika 10: Postavitev izvorov na delovno površino .....	16
Slika 11: Izbira ponorov .....	16
Slika 12: Postavitev gradnikov .....	17
Slika 13: Hitri meni nad gradnikom .....	17
Slika 14: Povezovanje gradnikov .....	18
Slika 15: Nastavitev parametrov gradnika.....	19
Slika 16: Parametri gradnika "Sinusni generator" .....	20
Slika 17: Nastavitev parametrov simulacije .....	20
Slika 18: Parametri nastavitve simulacije.....	21
Slika 19: Zagon simulacije .....	22
Slika 20: Izhod prikazan na prikazovalniku .....	22
Slika 21: Izris potem, ko smo spremenili parametre sinusnega generatorja .....	23
Slika 22: Prikazovalnik potem, ko smo mu spremenili parametre .....	24
Slika 23: Shranjevanje projekta.....	24
Slika 24: Okno za shranjevanje .....	25
Slika 25: Odpiranje projekta.....	25
Slika 26: Izvoz projekta kot slike .....	26
Slika 27: Okno za urejanje načina izvoza slike .....	26
Slika 28: Kopiranje slike v odložišče .....	27
Slika 29: Super Block (nadblok) .....	28
Slika 30: Primer za prikaz uporabe nadbloka.....	29
Slika 31: Izberemo področje nadbloka .....	30
Slika 32: Prikaz določanja področja za nadblok.....	30
Slika 33: Vsi izbrani gradniki znotraj nadbloka .....	31
Slika 34: Vsebina nadbloka .....	31
Slika 35: Odpiranje lastnosti bloka.....	32
Slika 36: Popravljen vsebina, ki preimenuje blok v Nadblok.....	33
Slika 37: Ikona nadbloka se je spremenila .....	33
Slika 38: Scifunc blok .....	34
Slika 39: Nastavljanje parametrov Scifunc bloku .....	34
Slika 40: Okno za vpis Scilab programske kode .....	35
Slika 41: Uporaba Scifunc bloka v simulacijski shemi .....	36
Slika 42: Odziv sistema na rampo .....	36
Slika 43: Prilagoditev Sum bloka (blok vsote).....	37
Slika 44: Prilagoditev izhodnega modula »Mscope« (večkratni prikazovalnik).....	38

Slika 45: Nastavitev bloka »Generator stopnice«.....	38
Slika 46: Simulacijska shema linearne diferencialne enačbe .....	39
Slika 47: Rezultat linearne diferencialne enačbe .....	39
Slika 48: Fizikalni problem nihanja uteži po podlagi .....	40
Slika 49: Primer simulacijske sheme nelinearne diferencialne enačbe v Scicos-u.....	41
Slika 50: Grafični prikaz spremenljivk nelinearne diferencialne enačbe .....	42
Slika 51: Izberemo tri integratorje za tri diferencialne enačbe .....	43
Slika 52: Rösslerjev kaotični model v Scicos-u.....	45
Slika 53: Grafični prikaz rešitve sistema (atraktor kaotičnega sistema).....	45
Slika 54: Simulacija reaktivnega letala.....	47
Slika 55: Vnašanje parametrov v blok.....	47
Slika 56: Nastavitev prikazovalnika .....	48
Slika 57: Grafični prikaz odziva letala na premik višinskega krmila .....	48
Slika 58: Princip delovanje spremembe globine podmornice .....	49
Slika 59: Bločni diagram vodenja za spreminjanje globine podmornice .....	49
Slika 60: Simulacijska shema vodenja podmornice .....	50
Slika 61: Parametri bloka »Num(s)/Den(s)« .....	50
Slika 62: Sprememba globine podmornice glede na željeno globino.....	51
Slika 63: RC vezje .....	53
Slika 64: Simulacijska shema modela R-C vezja .....	54
Slika 65: Vpis parametrov v blok Num(s)/Den(s) .....	54
Slika 66: Nastavitev stopnice.....	55
Slika 67: Nastavitev prikazovalnika .....	55
Slika 68: Nastavitev parametrov ure za sinhronizacijo .....	56
Slika 69: Krivulja izhodne napetosti v odvisnosti od vhodnega signala .....	56
Slika 70: Simulacijska shema RC vezja z generatorjem pravokotnega signala na vhodu .....	57
Slika 71: Izris odziva z generatorjem pravokotnega signala na vhodu.....	57
Slika 72: Prikaz izhodnega signala, če imamo na vhodu generator pravokotnega signala .....	58
Slika 73: Poenostavljen model avtomobilskega vzmetenja.....	59
Slika 74: Potek risanje simulacijske sheme za $\ddot{y}_1$ .....	61
Slika 75: Potek risanja simulacijske sheme za $\ddot{y}_2$ .....	62
Slika 76: Simulacijska shema modela avtomobilskega vzmetenja.....	63
Slika 77: Nastavitev prikazovalnika .....	64
Slika 78: Odziv karoserije ( $y_1$ ) in kolesa ( $y_2$ ) pri trku s pločnikom.....	64
Slika 79: Nastavitev parametrov integratorja .....	66
Slika 80: Prikaz risanja simulacijske sheme .....	67
Slika 81: Simulacijska shema propagandne akcije .....	67
Slika 82: Nastavitev parametrov večkratnega prikazovalnika.....	68
Slika 83: Rezultat za vsak segment kupcev posebej, prikazan na večkratnem prikazovalniku .....	69
Slika 84: Blokovna shema sistema za regulacijo temperature.....	69
Slika 85: Simulacijska shema sistema za temperaturno regulacijo .....	70
Slika 86: Odziv sistema za temperaturno regulacijo na enotino stopnico .....	70
Slika 87: Simulacijska shema sistema za temperaturno regulacijo s PID regulatorjem..	72

Slika 88: Odziv sistema za temperaturno regulacijo s PID regulatorjem.....	72
Slika 89: Simulacijska shema sistema za temperaturno regulacijo s PID regulatorjem in prefiltrom.....	73
Slika 90: Odziv sistema temperaturne regulacije s PID regulatorjem in prefiltrom .....	74



## UVOD

Beseda simulacija izhaja iz latinske besede simulatio, kar pomeni oponašanje ali posnemanje. V ožjem pomenu besede je simulacija eksperimentiranje z abstraktnim modelom. Simulacija je v tem pomenu mišljena kot proces, ki je v nekem smislu kopija ali paralela realnega procesa.

Simulacija dinamičnih sistemov je ena najpomembnejših metod pri razvoju sodobnih tehnologij. Predstavlja učinkovito orodje pri reševanju problemov, ki jih lahko opišemo z matematičnim modelom, na primer z diferencialnimi oziroma diferenčnimi enačbami. Simulacija se uporablja na vseh področjih, ki temeljijo na sistemskem pristopu obravnave procesa. Naj naštejemo le nekaj področij, kjer se računalniška simulacija uporablja: vodenje sistemov, robotika, računalništvo, fizika, biologija, kemija, medicina, farmakokinetika.

Strokovnjaki so razvili razna programska orodja, s katerimi lahko simuliramo dinamične sisteme. Eno izmed orodij je tudi programski paket Scilab, ki ga lahko iz interneta brezplačno prenesemo na osebni računalnik. V diplomski nalogi bomo opisali programski modul Scicos, ki je del programskega paketa Scilab. Scicos je zelo napredno orodje, s katerim lahko simuliramo domala vse bolj ali manj zahtevne dinamične sisteme. Orodje je namenjeno začetnikom in zahtevnim uporabnikom. Namen priročnika je približati orodje začetnikom, ki nimajo programerskega predznanja in bi radi uporabljali programski modul Scicos.

Priročnik je sestavljen tako, da se zahtevnost uporabe programskega modula Scicos stopnjuje skozi poglavja priročnika. Začeli bomo z nasvetom, kako pridobiti program, ter opisali njegovo namestitev in zagon. Sledi opis menijev in bistvenih gradnikov za začetniško nivo uporabnika. Na preprostem primeru bomo prikazali uporabo programa, od postavitve gradnikov pa do simulacije. Primer bo enostaven in pregleden, zato se bomo lahko bolj posvetili opisovanju osnovnih nalog, kot so postavitve in povezovanje gradnikov, shranjevanje in odpiranje projekta, prenašanje projekta kot slike in nenazadnje simulaciji projekta. Ko uporabnik osvoji osnove dela z modulom Scicos, lahko prične s simulacijo različnih vrst modelov. V priročniku bomo prikazali simulacijo linearne diferencialne enačbe, nelinearne diferencialne enačbe, sistema diferencialnih enačb, prenosne funkcije in modularno

sestavljene sistema. Vsem modelom bo sledil podroben opis in postopek simulacije. Na tej točki bo imel bralec vse potrebno znanje za simulacijo dinamičnih sistemov in to znanje priročnik na koncu dodatno obogati z različnimi ilustrativnimi primeri.

## 1. PROGRAMSKI MODUL SCICOS

Scilab je programski paket za numerično računanje matematičnih funkcij, ki so ga razvili na inštitutu INRIA v Franciji, in je odprto programsko okolje, v katerem je lahko izdelava funkcij in njihovih knjižnic v rokah uporabnika. Uporaben je za računanje z različnimi oblikami matematičnih zapisov in struktur. Računa lahko z vsemi znanimi oblikami števil, z matrikami, polinomi, objekti, ki prikazujejo modele dinamičnih sistemov, itd.

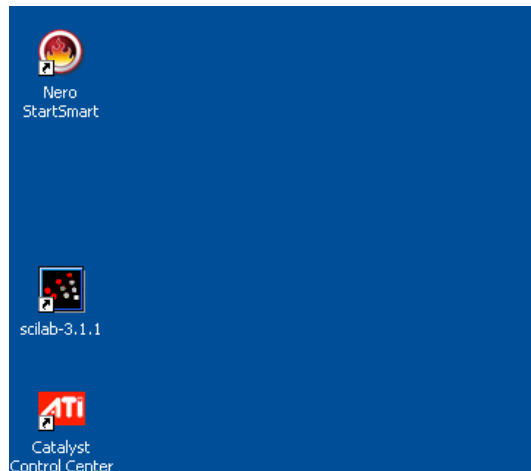
V okviru programskega paketa Scilab je programski modul Scicos, ki se uporablja za modeliranje in simuliranje dinamičnih sistemov. Scicos omogoča na modularen in grafičen način sestaviti bločne simulacijske sheme za zelo kompleksne dinamične sisteme, pri čemer ni potrebno imeti programerskega predznanja. V Scicos-u so sistemi prikazani v povezanih blokih in podsistemih ali nadblokih. Vrednosti, ki vsebujejo decimalna števila se ločijo z decimalno piko. Grafično okolje je prijazno do uporabnika in zagotavlja hitro ter učinkovito analizo dinamičnih sistemov.

Uporabnik programskega modula Scicos lahko najde veliko uporabne literature, ki bo koristila predvsem tistim, ki želijo nadgraditi znanje iz simulacije dinamičnih sistemov. V ta namen priporočamo izvrstno knjigo (Campbel et al., 2006). Knjiga je napisana v angleškem jeziku in uporabniku predstavi uporabo Scilab programskega orodja ter Scicos modula. Za naprednejše uporabnike je tudi knjiga (Scilab, 1999), veliko literature pa najdemo tudi na svetovnem spletu (Scilab, 2007), (Scilab/Scicos, 2007), kjer lahko najdemo v celoti izdelane simulacijske projekte in njihov opis.

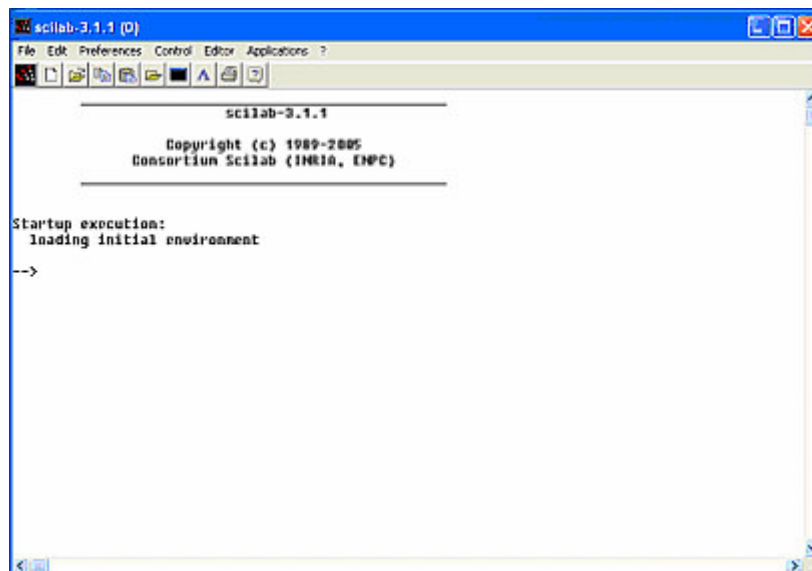


## 2. ZAGON PROGRAMA

Na računalnik si namestimo programski paket Scilab, ki ga brezplačno najdemo na spletni strani <http://www.scilab.org>. V priročniku je uporabljena inačica 3.1.1 programskega paketa Scilab za operacijski sistem Windows. Po uspešno končani namestitvi se na namizju pokaže ikona bližnjice (slika 1) do programskega paketa Scilab. Z dvoklikom na ikono se zažene program Scilab, kot je prikazano na sliki 2.

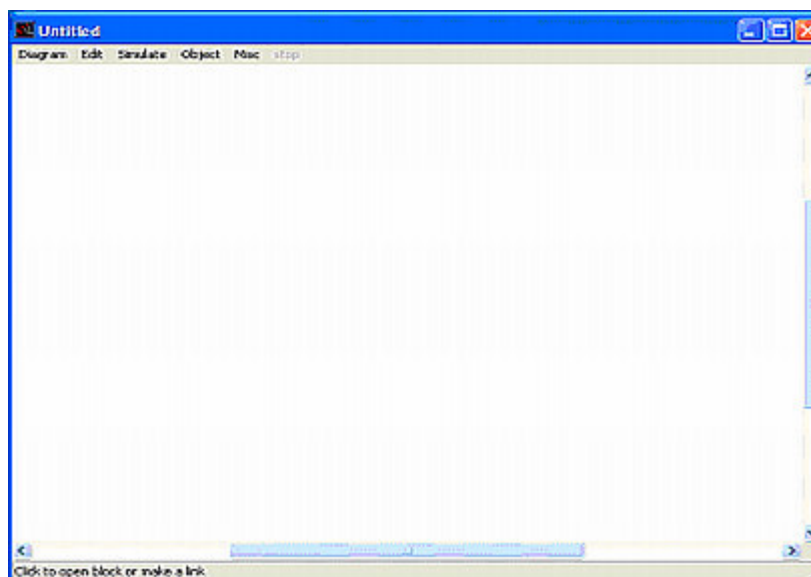


Slika 1: Namizje



Slika 2: Zagon programa Scilab

Za zagon programskega modula Scicos v ukazno vrstico vpišemo `scicos` ali kliknemo na orodno vrstico Applications (aplikacije) ter iz menija izberemo možnost Scicos. Odpre se novo okno (slika 3) z imenom Untitled (neimenovan), to je delovno okolje programskega modula Scicos, ki ga bomo podrobneje spoznali v nadaljevanju.



Slika 3: Osnovno okno programa Scicos

### 3. MENIJI PROGRAMA

Meniji programa se nahajajo v zgornjem predelu okna. S klikom na eno od ponujenih možnosti se odpre padajoči meni, ki ponuja uporabniku nove možnosti. Osnovni meni je sestavljen iz smiselnih skupin in omogoča uporabniku hitro iskanje ter uporabo funkcij, ki jih ponuja program. Smiselne skupine so: Diagram (shema), Edit (urejanje), Simulate (simulacija), Object (predmet), Misc (razno).

- Diagram (shema)

V tej skupini se nahajajo ukazi za upravljanje z datotekami: odpiranje, shranjevanje, preimenovanje in izvažanje projektov v druge oblike. Poleg tega je zelo koristen tudi ukaz Replot, ki deluje kot osvežitev oziroma ponoven izris sheme, s čimer pridobimo na preglednosti.

- Edit (urejanje)

Edit ali urejanje je skupina, ki nudi vrsto ukazov za urejanje simulacijskih shem. Ukaz Palettes (knjižnice) se nahaja čisto na začetku in ne brez razloga. Ta ukaz bomo pogosto uporabljali, saj kliče knjižnice, v katerih so spravljene vsi gradniki, ki jih lahko najdemo v programskem modulu Scicos. Ostali ukazi v tej skupini so namenjeni kopiranju, premikanju, povezovanju predmetov, brisanju, skratka urejanju vsebine projekta.

- Simulate (simulacija)

Skupina za simulacijo je smiselno na tretjem mestu v meniju, saj jo uporabimo šele takrat, ko je projekt narejen in pripravljen za zagon. V skupini se nahajajo ukazi za nastavitev simulacije z ukazom Setup (nastavitev) ter zagon simulacije z ukazom Run (zaženi). Če se pri izvedbi kaj zalomi ali ne gre po predvidenih načrtih, uporabimo ukaz Debug level za odkrivanje napak oziroma »razhroščevanje«.

- Object (predmet)

Skupina za urejanje predmetov, nameščenih v projektu, je namenjena predvsem vizualnim popravkom in je zato manj pomembna za samo simulacijo. Z ukazi v tej skupini lahko tako na primer povečamo predmete ali jih preimenujemo ter jim določimo oznako in barvo. Prav

gotovo to niso odvečne funkcije kadar se lotimo obsežnejšega projekta in je preglednost zelo pomembna. Če se pa zgodi, da simulacija ne steče po pričakovanjih, je v tako urejenem projektu lažje poiskati ter odpraviti napako.

- Misc (razno)

V skupini ukazov, ki se nahajajo pod razno, najdemo manj pogosto uporabljane ukaze. Z njimi lahko zamenjamo barvo ozadja, če nam bela ni všeč, izberemo vrsto pisave in podobno.



## 4. OSNOVNI GRADNIKI

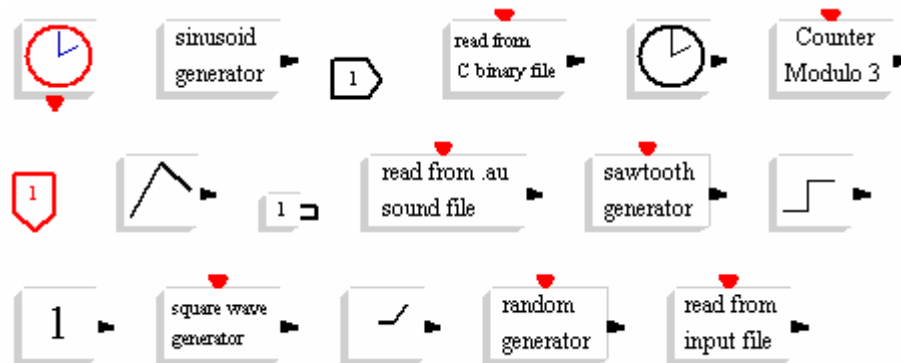
Da bi sestavili simulacijsko shemo v Scicosu, moramo dostopiti do osnovnih gradnikov, ki jih ponuja orodje. Scicos ponuja pisano paleto osnovnih gradnikov, ki so organizirani v skupinah, do katerih lahko dostopimo z uporabo ukaza Palettes (knjižnice). Ukaz se nahaja v osnovnem meniju Edit (urejanje) in povzroči, da se odpre novo okno, ki vsebuje spisek razpoložljivih skupin blokov. Pogosto uporabljane skupine blokov so: Sources (izvori), Sinks (ponori), Linear (linearni gradniki), Non linear (nelinearni gradniki) in Others (ostalo).

Z izbiro skupine se odpre novo okno, ki vsebuje razpoložljive bloke v tej skupini gradnikov. Na sliki 4 lahko vidimo primer gradnikov ali blokov iz skupine Sources (bloki signalnih virov).

Bloke lahko vnesemo v osnovno okno s klikom na željeni gradnik in nato v osnovnem oknu ponovno kliknemo tam, kamor želimo izbrani gradnik postaviti.

### 4.1 Bloki signalnih virov

Bloki signalnih virov vsebujejo le izhode in se uporabljajo za tvorjenje signalov, kot je razvidno iz slike 4.



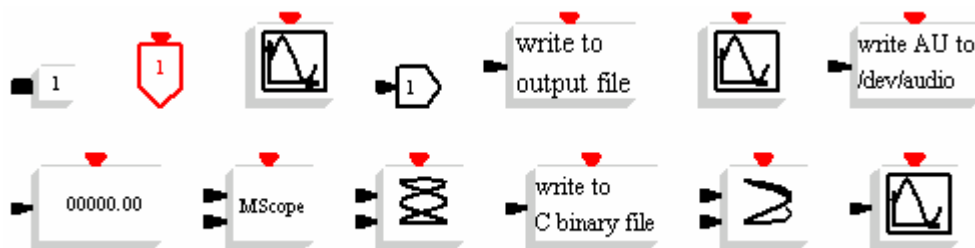
Slika 4: Bloki signalnih virov

Opis nekaterih pomembnejših blokov:

- Ura se uporablja za aktiviranje določenih blokov periodično z željeno frekvenco. Bloki, ki potrebujejo uro za sinhronizacijo poteka signala, imajo na vrhu bloka rdečo puščico, na katero jo povežemo.
- Sinusoid generator (sinusni generator) tvori signal sinusne oblike. Blok ne potrebuje sinhronizacijske ure kot vhod, ampak nastavimo parametre in deluje samostojno.
- Square wave generator (generator pravokotnega signala) proizvaja signal pravokotne oblike s frekvenco, ki jo določa sinhronizacijska ura na vhodu.
- Stopničast funkcijski generator zagotovi stopnico med dvema nivojema v časovnem intervalu, ki ga določimo.
- Konstanta je vrednost, ki jo določimo na izhodu bloka oziroma na vhodu nekega sistema.
- Časovni blok na izhodu daje podatke trenutnega časa računalnika.

#### 4.2 Bloki signalnih ponorov

Bloki signalnih ponorov (slika 5) so gradniki, ki imajo le vhode. Uporabljajo se za prikaz vrednosti na izhodu nekega sistema v grafični ali numerični obliki. S signalnimi ponori lahko spremljamo odziv sistema med simulacijo ter primerjamo željene ter dejanske podatke.



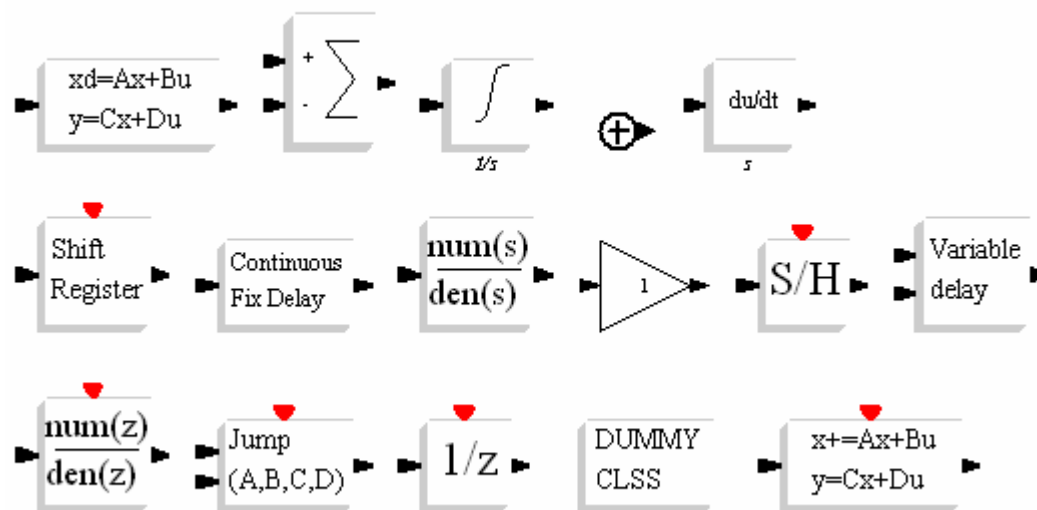
Slika 5: Bloki signalnih ponorov

Opis nekaterih pomembnejših blokov:

- Scope (prikazovalnik) je grafični prikazovalnik, ki na podlagi izhodne vrednosti simulacije izriše krivuljo.
- Mscope (večkratni prikazovalnik) ima podobno funkcijo kot Scope, le da lahko definiramo število vhodov.
- XY Scope (XY prikazovalnik) omogoča vizualizacijo poteka dveh vhodnih signalov, pri čemer je drugi vhod funkcija prvega.

### 4.3 Bloki, ki predstavljajo linearne gradnike sistemov

Bloki linearnih gradnikov (slika 6) predstavljajo gradnike za opis linearnih dinamičnih sistemov.



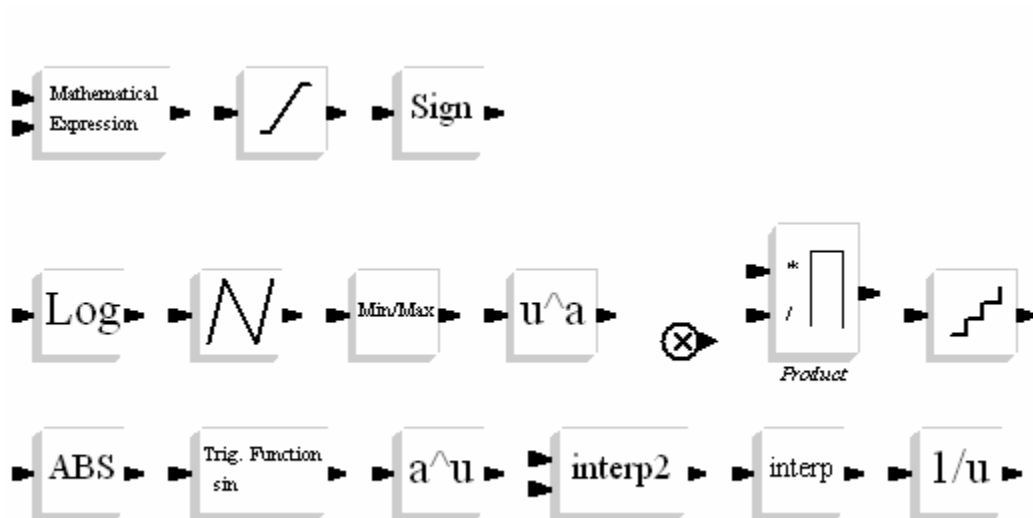
Slika 6: Bloki linearnih gradnikov

Opis nekaterih pomembnejših blokov:

- Sum (seštevalnik) je blok, ki da na izhodu vsoto vseh vhodov. Bloku lahko določimo poljubno število vhodov, njihove uteži ter njihov predznak. Poleg osnovnega bloka pravokotne oblike obstaja tudi inačica okrogle oblike, pri kateri pa ne moremo spreminjati predznaka in uteži vhodov.
- Integrator. Izhod tega gradnika je integral vhoda.
- Numerator/Denominator (num(s)/den(s)) je blok, v katerega vpišemo prenosno funkcijo.
- Gain (ojačevalnik) uporabimo takrat, kadar moramo funkcijo množiti s konstanto. Na izhodu tako dobimo vhod z ojačenjem, ki ga določimo.

#### 4.4 Bloki, ki predstavljajo nelinearne gradnike sistemov

Bloki nelinearnih gradnikov (slika 7) predstavljajo gradnike za opis nelinearnih dinamičnih sistemov.



Slika 7: Bloki nelinearnih gradnikov

Opis nekaterih pomembnejših blokov:

- Mathematical expression (matematični izraz) uporabimo, kadar želimo direktno zapisati matematično enačbo v Scicos. Bloku lahko nastavimo število vhodov, ki predstavljajo spremenljivke v enačbi.
- Multiplication (množenje). Izhod gradnika je produkt vhodov.



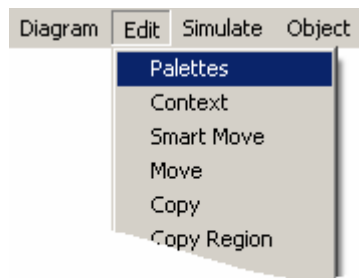
## 5. OSNOVE DELA S PROGRAMOM

V tem poglavju bomo na preprostem primeru prikazali, kako začeti z delom. Za prikaz bomo uporabili le izvore in ponore, torej tiste gradnike, ki imajo le izhode ali vhode. Naučili se bomo postaviti in povezati gradnike, zagnati simulacijo ter shraniti projekt.

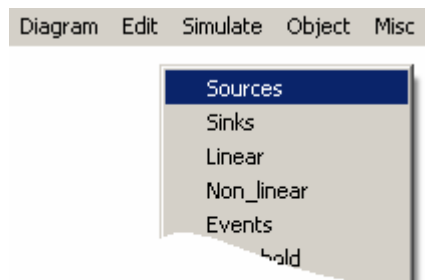
### 5.1 Postavitev gradnikov

Predpostavimo, da je program Scicos že zagnan in smo pripravljeni na prvi projekt. V ta namen bomo tvorili izhodni signal sinusnega generatorja in ga prikazali na prikazovalniku.

V meniju Edit (urejanje) izberemo Palettes (knjižnice) (slika 8) in prikaže se paleta možnosti, ki smo jih že opisali v prejšnjem poglavju. Najprej izberemo Sources (viri) (slika 9).

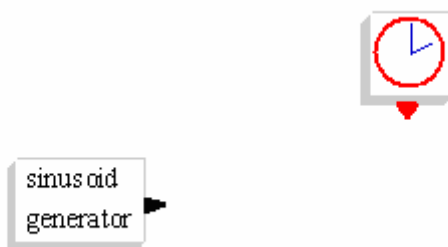


Slika 8: Izbira objektov



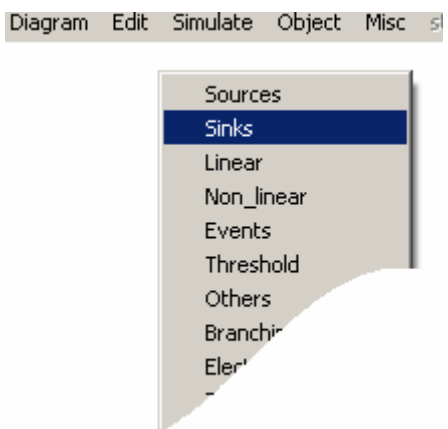
Slika 9: Izbira izvorov

Odpre se okno izvorov in med njimi izberemo blok Sinusoid generator (sinusni generator). Kliknemo nanj in ga prenesemo na delovno površino tako, da na željenem mestu ponovno kliknemo z levim gumbom na miški. Nato izberemo še Event clock (ura za sinhronizacijo) in jo na isti način postavimo na delovno površino (slika 10).



Slika 10: Postavitev izvorov na delovno površino

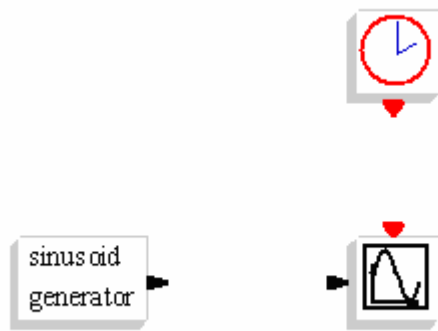
Ponovno izberemo v meniju Edit (urejanje) in Palettes (knjižnice), nato pa Sinks (ponori) (slika 11). Izberemo Scope (prikazovalnik) ter ga postavimo na delovno površino.



Slika 11: Izbira ponorov

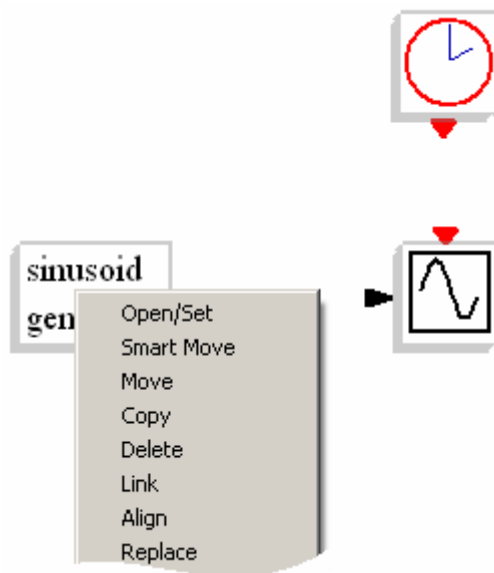
Postavili smo vse gradnike za naš projekt. Preden začnemo gradnike povezovati, jih je potrebno, za boljšo preglednost, primerno poravnati (slika 12). Pri obsežnejših projektih je preglednost zelo pomembna.





Slika 12: Postavitev gradnikov

Zgodi se namreč lahko, da so gradniki postavljeni na napačno mesto, še posebej takrat, ko je potrebnih bistveno več gradnikov, kakor smo predvidevali na začetku. V ta namen uporabimo hitri meni, ki ga prikličemo z desno tipko na miški. Za premikanje določenega gradnika se postavimo z miškinim kazalcem nanj in pritisnemo desno tipko. Prikaže se meni, ki omogoča delo s posameznim gradnikom (slika 13).



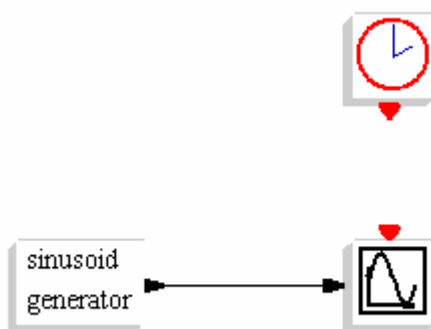
Slika 13: Hitri meni nad gradnikom

Opis nekaterih ukazov hitrega menija:

- **Move (premakni)**: Če hočemo gradnik premakniti, uporabimo ukaz Move. Gradnik se »odlepi« iz delovne površine in postavimo ga lahko tja, kjer nam bolj ustreza.
- **Copy (kopiraj)**: Če želimo v projektu uporabiti več enakih gradnikov je zelo zamudno, da vsakega posebej poiščemo na način, ki smo ga prej opisali. Ker je gradnik že na površini, ga lahko enostavno preslikamo z ukazom Copy. Kopijo gradnika nato postavimo na željeno mesto.
- **Delete (izbriši)**: Ukaz za odstranjevanje gradnika.
- **Align (poravnaj)**: Ukaz uporabimo kadar želimo postaviti gradnik v vrsto, kar kasneje omogoča lepše povezovanje. Ko izberemo ukaz Align, kliknemo na vstopno oziroma izstopno točko gradnika, ki ga želimo postaviti v vrsto z izbranim gradnikom. Ko so gradniki enkrat povezani, jih ni več mogoče premikati s tem ukazom.
- **Flip (obrni)**: Ukaz obrne izbrani gradnik za kot  $180^\circ$ .

## 5.2 Povezovanje gradnikov

Gradnike povezujemo vedno od izhoda proti vhodu. Za povezovanje lahko uporabimo hitri meni in ukaz Link (poveži) tako, da izberemo gradnik, katerega izhod želimo povezati z vhodom drugega gradnika. Ko potrdimo ukaz Link, se prikaže povezovalna črta (slika 14) in naša naloga je, da jo spojimo z vhodom željenega gradnika.

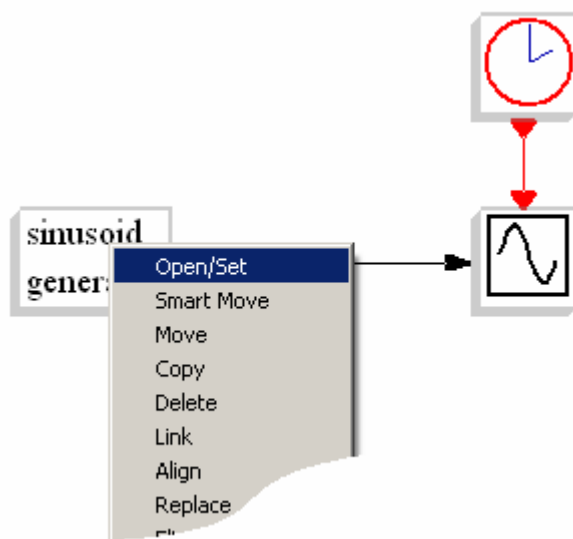


Slika 14: Povezovanje gradnikov

Namesto hitrega menija lahko enostavneje povežemo gradnike tako, da z levim gumbom miške kliknemo na izhod gradnika in povlečemo do vhoda željenega gradnika ter ponovno kliknemo.

### 5.3 Spreminjanje parametrov gradnikov

Do sedaj smo gradnike postavili na namizje in jih povezali. Preden zaženemo simulacijo, je potrebno nastaviti nekatere parametre gradnikov, odvisno od zahteve problema. Do lastnosti posameznega gradnika pridemo tako, da kliknemo na gradnik z desno miškino tipko ter na meniju, ki se prikaže, izberemo Open/set (odpri/nastavi) (slika 15) ali pa enostavneje dvakrat kliknemo na gradnik z levo tipko na miški.



Slika 15: Nastavitev parametrov gradnika

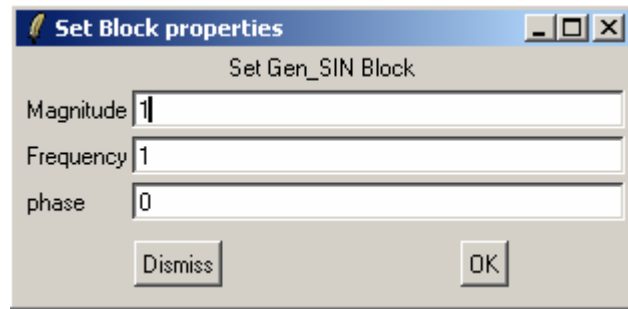
Prikaže se okno, v katerem se nahajajo parametri gradnika, ki jih lahko spreminjamo. V našem primeru smo odprli gradnik Sinusoid generator (sinusni generator) (slika 13). V gradniku Sinusoid generator lahko spreminjamo tri parametre (slika 16):

Magnitude (amplituda) določa amplitudo izhoda sinusnega generatorja.

Frequency (frekvenca) določa frekvenco ali število nihajev na sekundo.

Phase (fazni zamik) določa fazni zamik signala od  $0^\circ$  do  $360^\circ$ .

Da bi bil vhodni signal malo večji, lahko Magnitude povečamo na 5 in fazo zamaknemo za 90°.

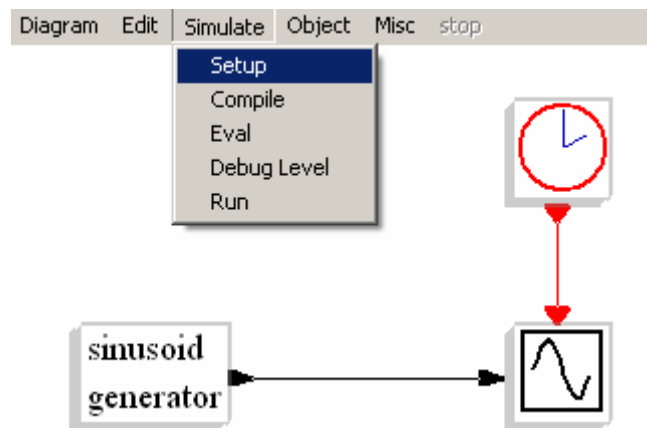


Slika 16: Parametri gradnika "Sinusni generator"

Tudi ostalim gradnikom lahko spremenimo parametre, vendar za naš primer to ni potrebno. Ko gradnikom nastavimo vse potrebne parametre, se lahko lotimo simulacije. Smisel računalniške simulacije je v tem, da lahko spreminjamo parametre gradnikov tako, da dosežemo željeni učinek.

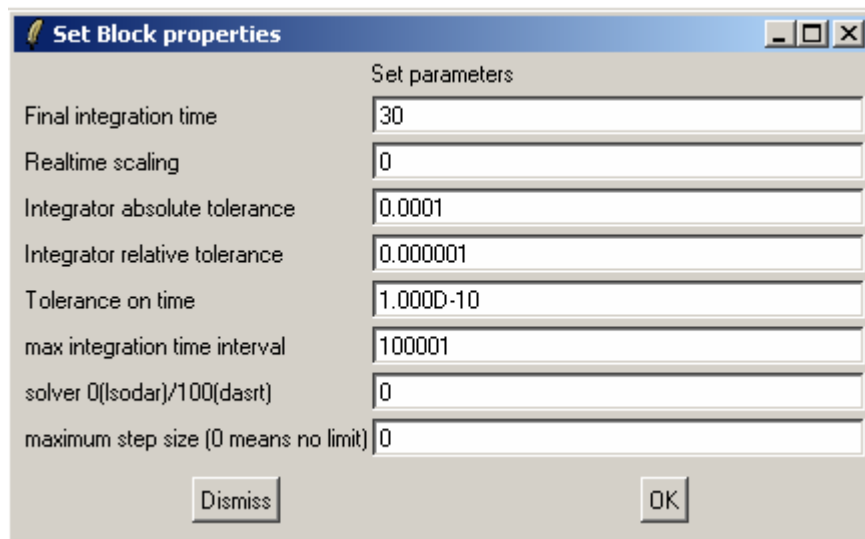
## 5.4 Simulacija

Ko so vsi gradniki povezani in so njihovi parametri ustrezno določeni, se lahko posvetimo simulaciji. Da bo simulacija kar najbolj ustrezala razmeram našega projekta, je potrebno nastaviti nekatere parametre. V glavnem meniju izberemo gumb Simulate (simulacija) in nato Setup (nastavitev) (slika 17).



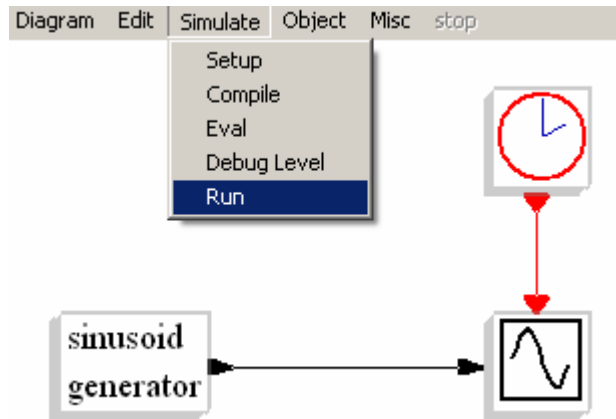
Slika 17: Nastavitev parametrov simulacije

Prikaže se okno (slika 18), v katerem lahko nastavljam navedene parametre. Za naš primer bomo nastavili parameter Final integration time (končni čas integracije). V osnovi je nastavljen na 100000, vendar ga bomo za naš primer nastavili na primernejšo vrednost. Izberemo vrednost 30. Smiselno je nastaviti čas na prikazovalniku in čas trajanja simulacije tako, da se dogodek zgodi na enem oknu prikazovalnika.

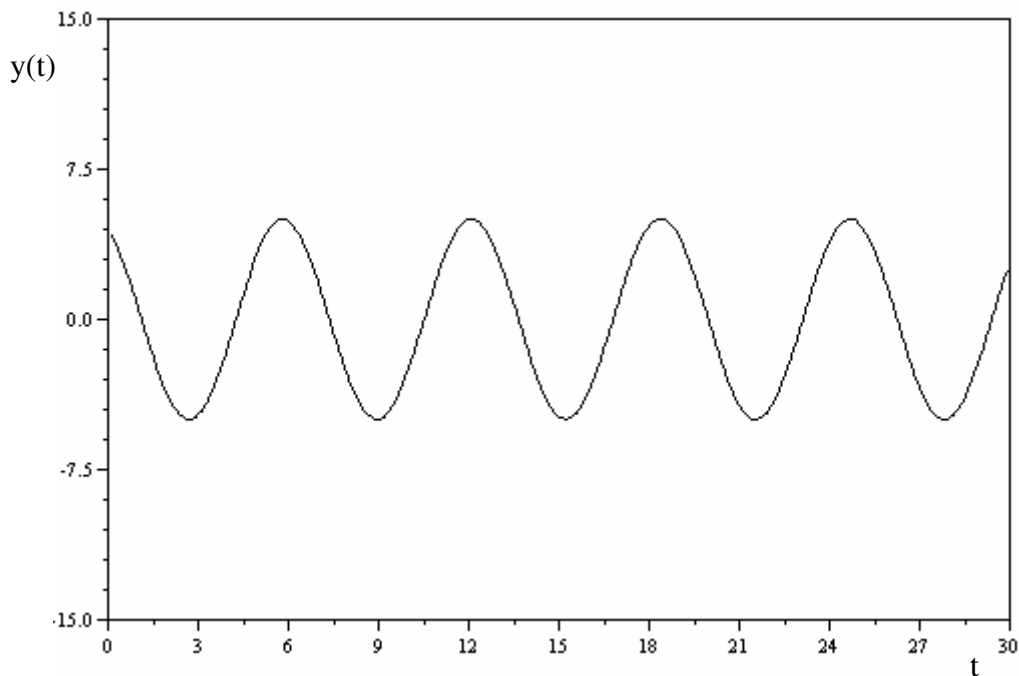


Slika 18: Parametri nastavitv simulacije

Ko smo nastavili parametre simulacije, jo lahko zaženemo. V glavnem meniju kliknemo gumb Simulate (simulacija) in nato Run (zagon) (slika 19). Na zaslonu se kot rezultat prikaže sinusna krivulja (slika 20), ki prikazuje izhod generatorja.

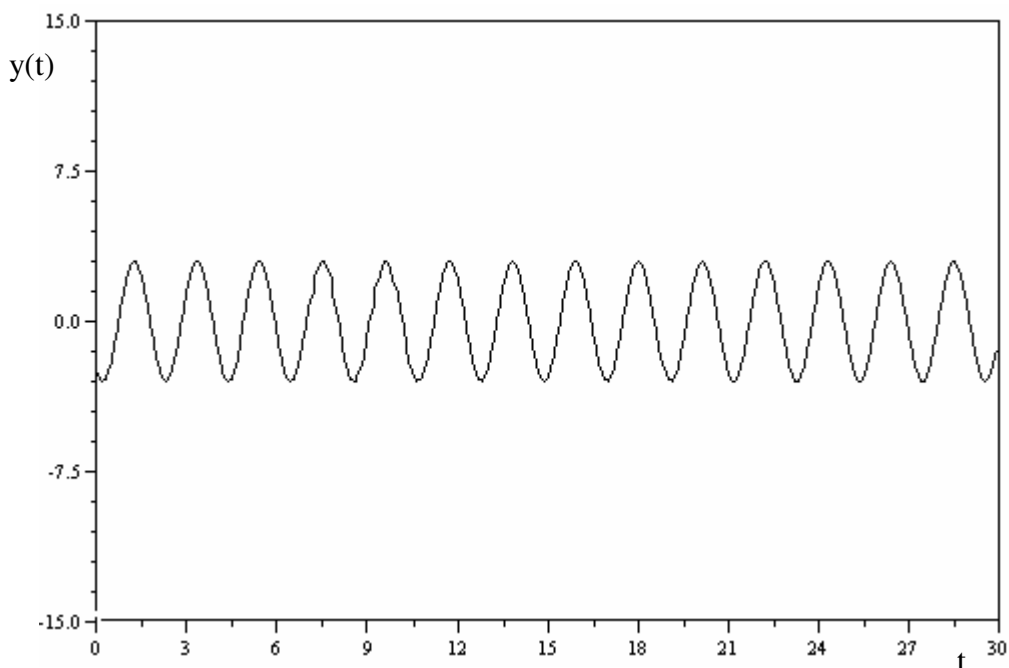


Slika 19: Zagon simulacije



Slika 20: Izhod prikazan na prikazovalniku

Sedaj lahko spreminjamo parametre gradnikov in opazujemo, kako se spreminja izhod. Za primer vzemimo sinusni generator. Nastavimo lahko različno frekvenco, amplitudo in fazni zamik. Če nastavimo Magnitude (amplituda): 3, Frequency (frekvenca): 3 in Phase (fazni zamik):  $180^\circ$ , vidimo, da se krivulja spremeni (slika 21).



Slika 21: Izris potem, ko smo spremenili parametre sinusnega generatorja

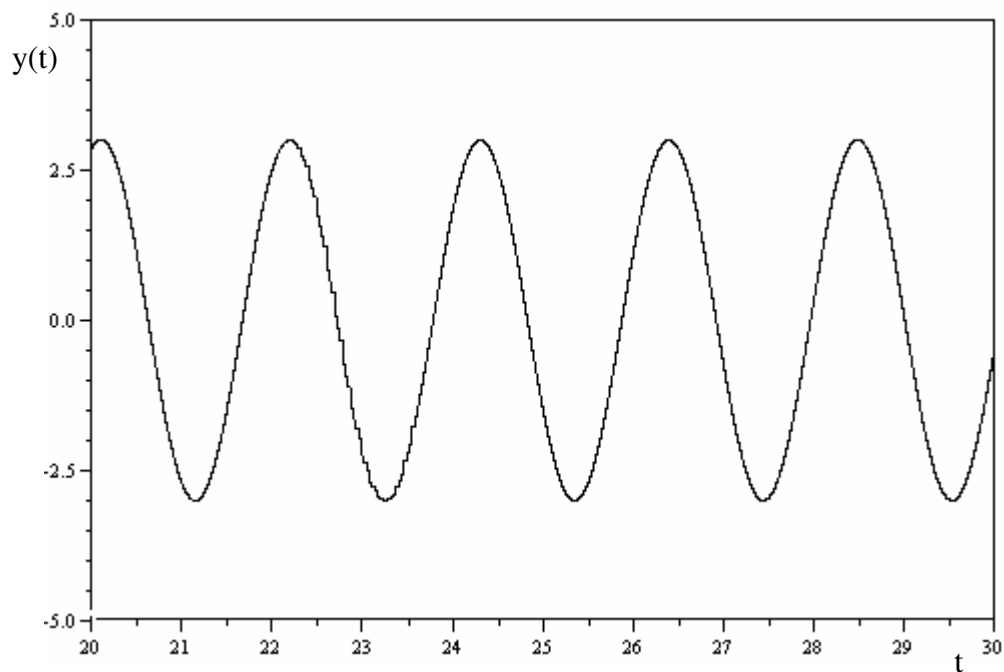
Če želimo, da prikazovalnik prikaže krivuljo bolj podrobno, moramo primerno spremeniti parametre prikazovalnika. Če vemo, da je amplituda signal generatorja 3, potem je smiselno nastaviti prikazovalnik na manjše območje, na primer ( $Y_{\min} = -5$ ,  $Y_{\max} = 5$ ) in Refresh period (čas osveževanja) tako, da bo izrisana krivulja čimbolj pregledna. Mi smo izbrali čas osveževanja 10 in pognali simulacijo. Krivulja je ista (slika 22), le prikazovalnik jo prikaže drugače.

## 5.5 Odpiranje in shranjevanje projektov

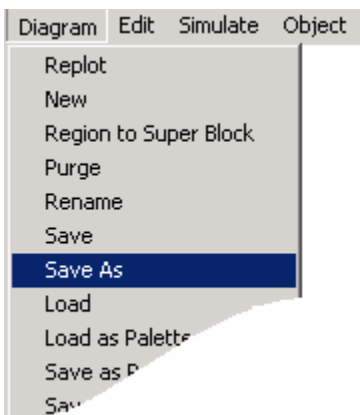
### 5.5.1 Shranjevanje projekta

Ko je projekt zaključen, ga moramo shraniti. To storimo tako, da v osnovnem meniju kliknemo na gumb Diagram (shema) in nato v padajočem meniju izberemo ukaz Save as (shrani kot) (slika 23). Odpre se Windows okno (slika 24) Save in ali Shrani v, odvisno ali imamo nameščeno slovensko ali angleško različico Windows operacijskega sistema, v katerem izberemo lokacijo, kamor želimo shraniti projekt. V spodnjem delu okna je okence

File name ali Ime datoteke. Tu vpišemo ime za naš projekt in obvezno končnico »cos«. Na primer: projekt1.cos in kliknemo na gumb Save ali Shrani.

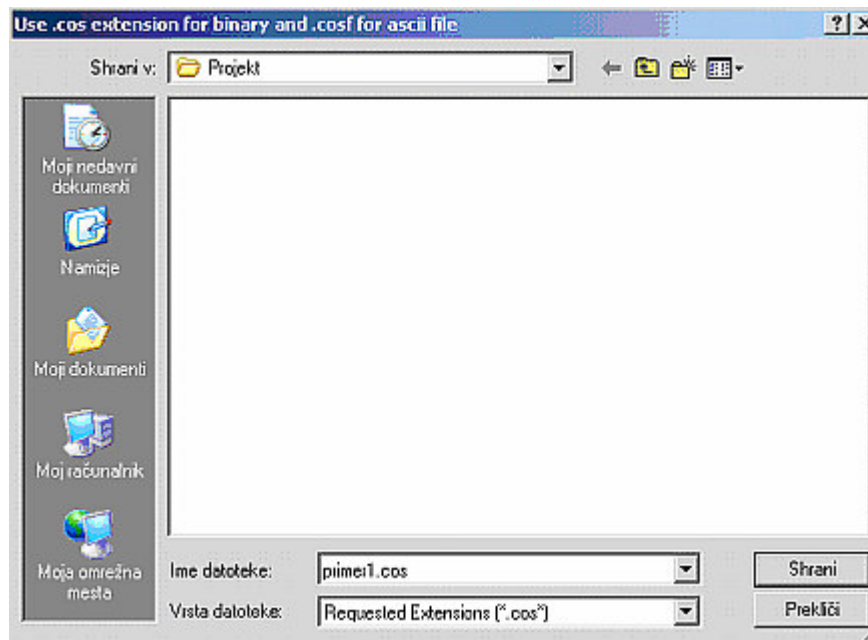


Slika 22: Prikazovalnik potem, ko smo mu spremenili parametre



Slika 23: Shranjevanje projekta

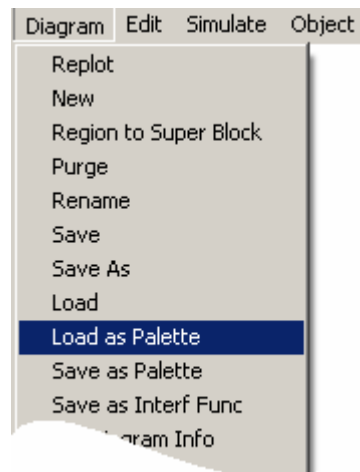




Slika 24: Okno za shranjevanje

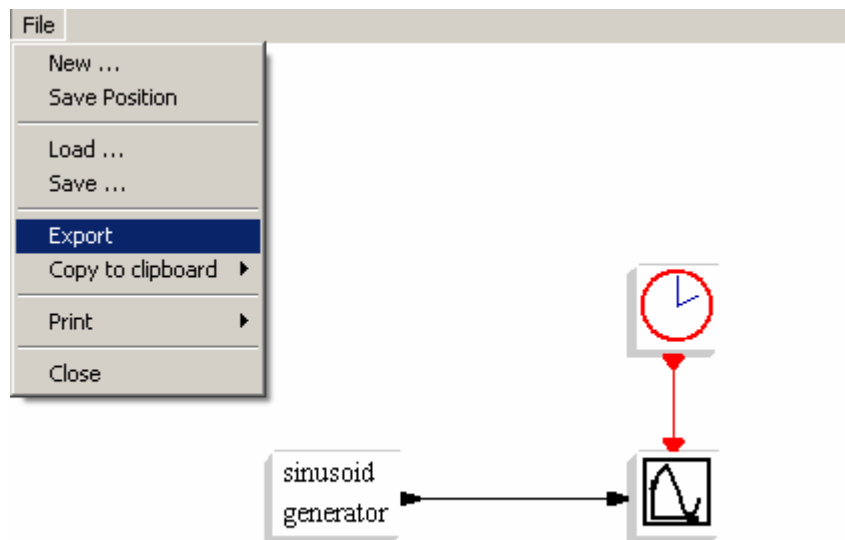
### 5.5.2 Shranjevanje projekta kot slike

Če hočemo projekt shraniti kot sliko, ga moramo najprej odpreti tako, da kliknemo v osnovnem meniju na Digram (shema) in nato Load as palette (naloži kot knjižnico) (slika 25).



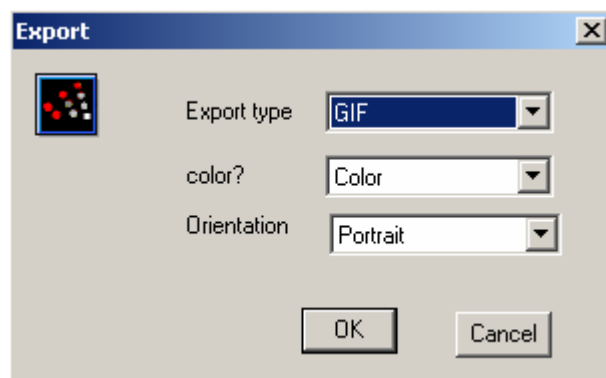
Slika 25: Odpiranje projekta

Odpre se okno in v zgornjem okencu izberemo lokacijo, kjer se nahaja shranjeni projekt. Nato izberemo datoteko, ki jo želimo odpreti in kliknemo na gumb Open (odpri). Odpre se novo okno, v katerem lahko vidimo projekt, vendar gradniki niso aktivni. Kliknemo na gumb File (datoteka) v osnovnem meniju in nato na gumb Export (izvozi) (slika 26).



Slika 26: Izvoz projekta kot slike

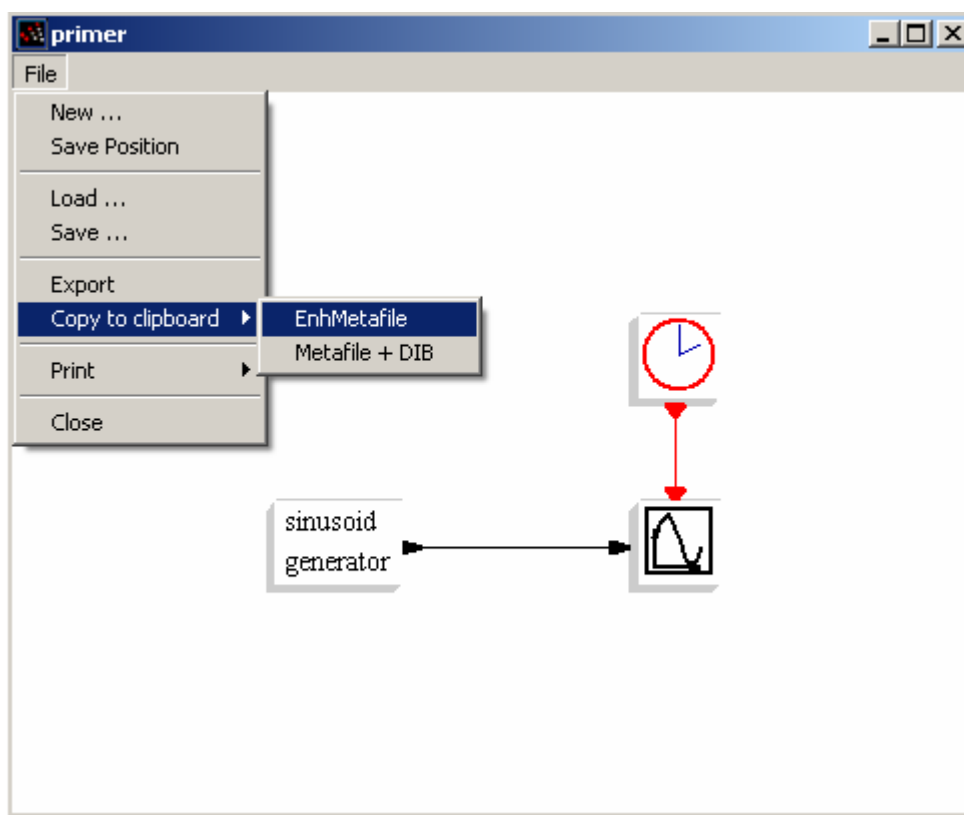
V oknu, ki se prikaže, izberemo format izvoza datoteke. Mi bomo izbrali GIF, lahko pa izbiramo med: Post script, Xfig, GIF, PPM, BMP in Enhanced meta file. V drugem okencu izberemo, ali bo izvožena slika barvna ali črno bela in nazadnje še, ali bo slika postavljena pokončno ali ležeče (slika 27).



Slika 27: Okno za urejanje načina izvoza slike

Ko izberemo gumb OK, se odpre okno za shranjevanje. V zgornje okno Save in (shrani v) vpišemo lokacijo, kamor želimo shraniti sliko. V spodnjem okencu File name (ime datoteke) izberemo ime slike in dodamo končnico GIF, na primer: *Slika.gif*.

Ko je projekt odprt kot slika, lahko izberemo tudi neposredno prenašanje slike v drugi dokument. V padajočem meniju File izberemo ukaz Copy to clipboard (preslikaj v odložišče). Program ponudi dve možnosti kopiranja slike: EnhMetafile in Metafile + DIB. Metafile je datoteka, ki vsebuje le podatke o originalni sliki, s katerimi obnovi sliko v Windows okolju. EnhMetafile (Enhanced Metafile) je Metafile s kompresirano vsebino in Metafile + DIB je metafile, ki v urejevalniku besedila prilepi tudi bitno sliko. Mi izberemo EnhMetafile in slika se preslika v odložišče (slika 28).



Slika 28: Kopiranje slike v odložišče

Sedaj lahko odpremo urejevalnik besedila kot je na primer Word dokument in izberemo ukaz Paste (prilepi). Slika se prenese v dokument.

### 5.5.3 Odpiranje projekta

Če hočemo odpreti projekt tako, da ga lahko spreminjamo ali zaženemo simulacijo, izberemo ukaz Load (naloži). Kliknemo v osnovnem meniju na gumb Diagram (shema) in nato v padajočem meniju izberemo ukaz Load (naloži). Prikaže se okno z imenom Open (odpri), v katerem izberemo projekt, ki ga želimo odpreti.

## 5.6 Super Block (Nadblok) in Scifunc

Poleg blokov, ki jih najdemo v samem modulu Scicos, lahko ustvarimo dodatne bloke po meri tako, da uporabimo že obstoječe bloke, ki jih najdemo v knjižnicah Palettes (knjižnice). Opisali bomo dva načina (Campbell et al, 2006):

- z uporabo pripomočka Super Block (nadblok) ali
- z uporabo Scifunc bloka

### 5.6.1 Super Block (nadblok)

Za izgradnjo nadbloka moramo vstaviti blok, ki ga najdemo v meniju Palettes (knjižnice) v skupini Others (ostalo) (slika 29).



Slika 29: Super Block (nadblok)

Ko ga prenesemo v projekt in kliknemo nanj, se odpre novo Scicos okno, v katerega vstavimo željene bloke in jih povežemo v simulacijsko shemo, ki jo želimo združiti v en blok.

Nadblok lahko naredimo tudi iz že obstoječega projekta. V meniju Diagram (shema) izberemo ukaz Region to Super Block (predel v nadblok). Nato z miško izberemo področje, ki naj se preslika v nadblok in nadblok je narejen.

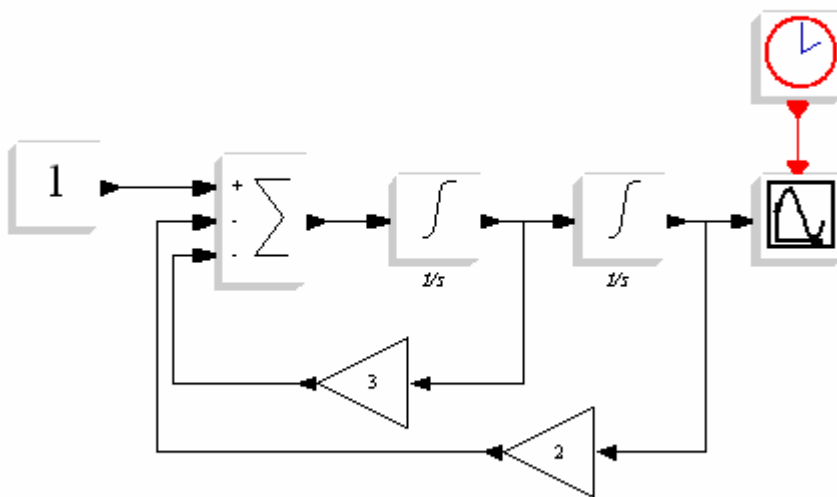
Za prikaz bomo naredili simulacijsko shemo za diferencialno enačbo drugega reda (5.1):

$$\ddot{y} + 3\dot{y} + 2y = 1 \tag{5.1}$$

Enačbo preoblikujemo v

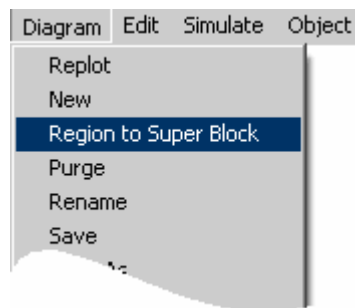
$$\ddot{y} = 1 - 3\dot{y} - 2y \tag{5.2}$$

in sestavimo simulacijsko shemo po postopku, ki ga bomo opisali v naslednjem poglavju (slika 30).



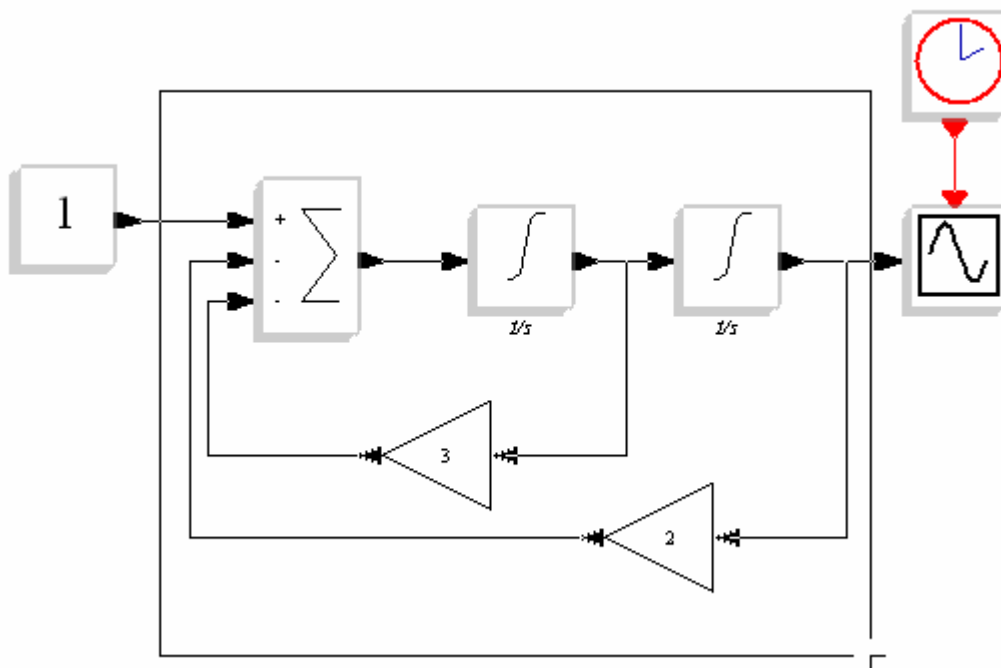
Slika 30: Primer za prikaz uporabe nadbloka

Sedaj izberemo ukaz Region to Super Block (slika 31) in izberemo področje, ki ga želimo vključiti v nadblok.



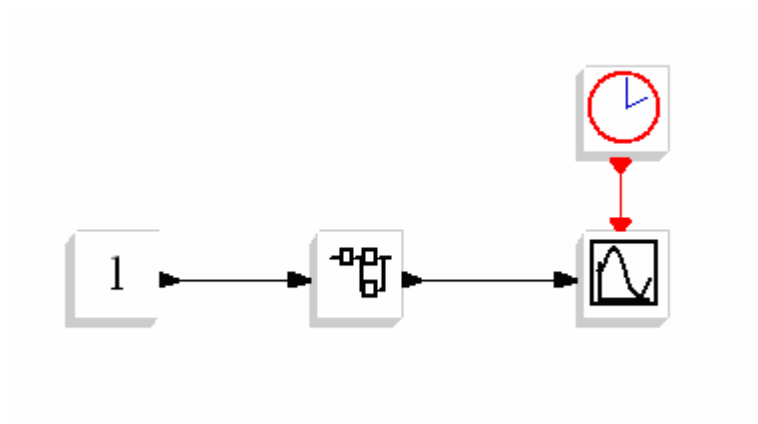
Slika 31: Izberemo področje nadbloka

Pritisnemo levi gumb na miški takrat, ko smo tam, kjer želimo začeti risati področje nadbloka. Z miško zarišemo okvir (slika 32) tako, da so vsi željeni gradniki znotraj okvirja in ponovno pritisnemo levo tipko.



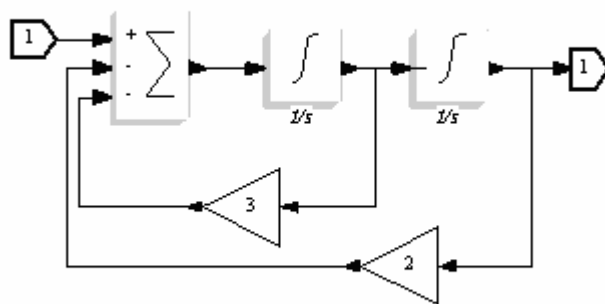
Slika 32: Prikaz določanja področja za nadblok

Slika se spremeni tako, da so se vsi bloki znotraj začrtanega okvirja premaknili v en sam blok (slika 33).



Slika 33: Vsi izbrani gradniki znotraj nadbloka

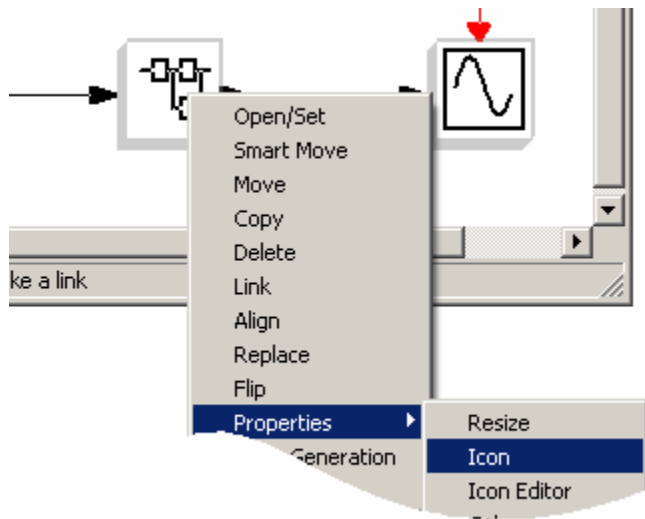
Če želimo pogledati, kaj vsebuje nadblok, moramo klikniti nanj in prikaže se njegova vsebina (slika 34). Kot vidimo, imamo na levi in desni strani novo označbo, ki ponazarja vhod oziroma izhod nadbloka.



Slika 34: Vsebina nadbloka

V večjem projektu lahko naredimo več nadblokov, zato bi bilo dobro bloke označiti tako, da bomo vedeli, kaj vsebujejo. V ta namen bomo preoblikovali ikono in na blok napisali na primer besedo Nadblok.

Z desnim miškinim kazalcem kliknemo na ikono nadbloka ter iz padajočega menija izberemo Properties (lastnosti) in nato Icon (ikona) (slika 35).



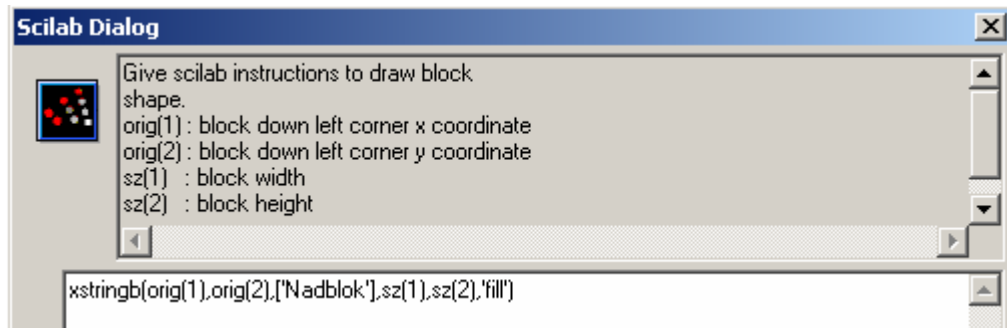
Slika 35: Odpiranje lastnosti bloka

Odpre se okno, v katerem je programska koda, ki opisuje blok. Vsebino izbrišemo in vpišemo ukazno vrstico (slika 36):

```
xstringb(orig(1), orig(2), ['Nadblok'], sz(1), sz(2), 'fill')
```

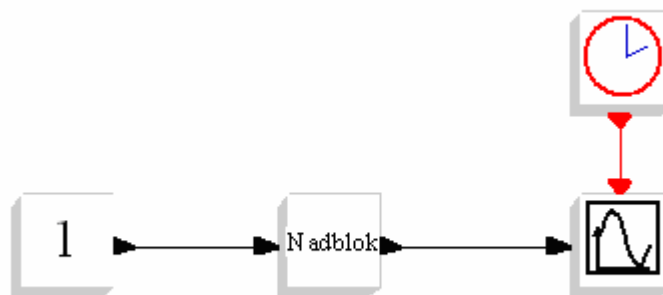
V sredini vrstice se nahaja naš napis Nadblok, ki se bo izpisal na bloku.





Slika 36: Popravljena vsebina, ki preimenuje blok v Nadblok

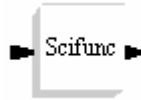
Kot lahko vidimo na sliki 37, se je standardna ikona nadbloka spremenila v napis Nadblok. To isto vrstico lahko uporabimo za različne napise, če spremenimo besedilo.



Slika 37: Ikona nadbloka se je spremenila

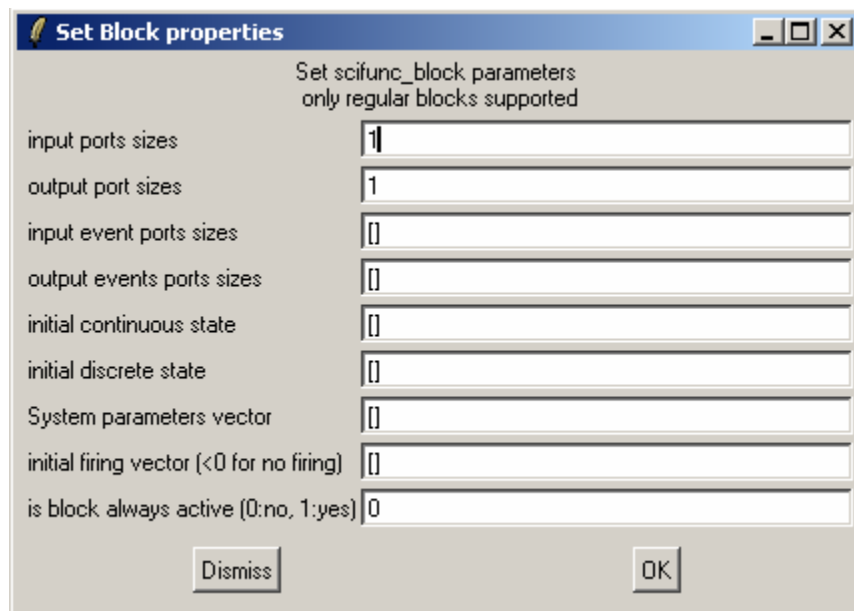
### 5.6.2 Scifunc blok

Scifunc blok uporabljamo za izdelavo novega Scicos bloka z uporabo programskega jezika Scilab. Bloku lahko nastavljamo parametre, kot so število vhodov in izhodov, začetne vrednosti in ostale parametre, ki jih mora vnesti uporabnik in se nanašajo na obnašanje bloka med simulacijo. Za izgradnjo Scifunc bloka moramo vstaviti gradnik, ki ga najdemo v meniju Palettes (knjižnice) v skupini Others (ostalo) (slika 38).



Slika 38: Scifunc blok

Blok prenesemo v projekt in s klikom nanj se odpre okno (slika 39), v katerem nastavimo parametre. Za naš primer ne bomo v tem oknu storili ničesar in bomo le potrdili z gumbom OK.



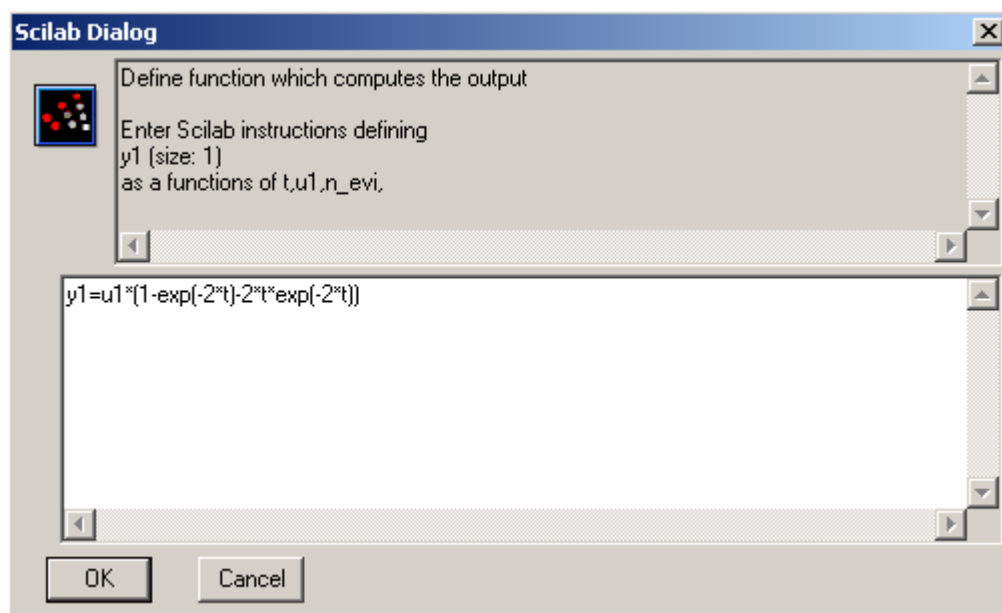
Slika 39: Nastavljanje parametrov Scifunc bloku

Pokaže se naslednje okno (slika 40), v katerega vpišemo Scilab programsko kodo. Z  $y$  definiramo izhod in z  $u$  definiramo vhod.

Za naš primer bomo naredili blok za simulacijo enačbe (5.3):

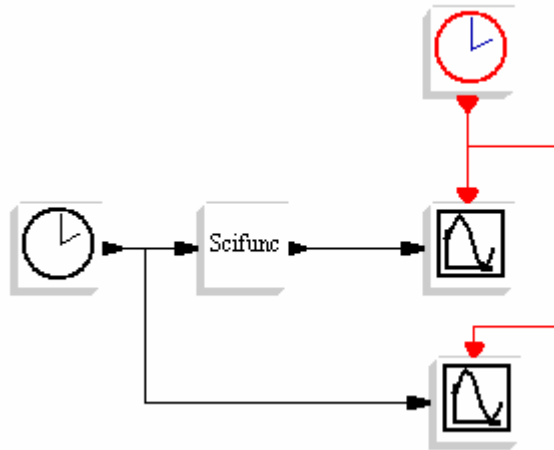
$$y = x(1 - e^{-2t} - 2te^{-2t}) \tag{5.3}$$

Enačbo (5.3) vpišemo v Scifunc okno in potrdimo z gumbom OK. Prikaže se naslednje okno, ki nas sprašuje, ali želimo, da program zažene kakšno datoteko ali grafični vmesnik. Za to dodatno funkcijo je potrebno dodatno znanje Scilab programskega jezika, zato bomo pustili ukazno okno prazno in nadaljevali s potrditvijo na gumb OK. Naslednje okno je podobno kot prejšnje in nas sprašuje, ali želimo, da se morda katera datoteka ali grafični vmesnik zaključi. Ukazno okno pustimo prazno in nadaljujemo z OK. Zopet se pojavi ukazno okno, ki nas sprašuje, ali želimo spremeniti razmerje med vhomom in izhodom. Če ne spreminjamo že vpisane ukazne vrstice, bo razmerje izhod:vhod 1:1, oziroma če želimo spremeniti razmerje, vpišemo vrednost v oglati oklepaj.



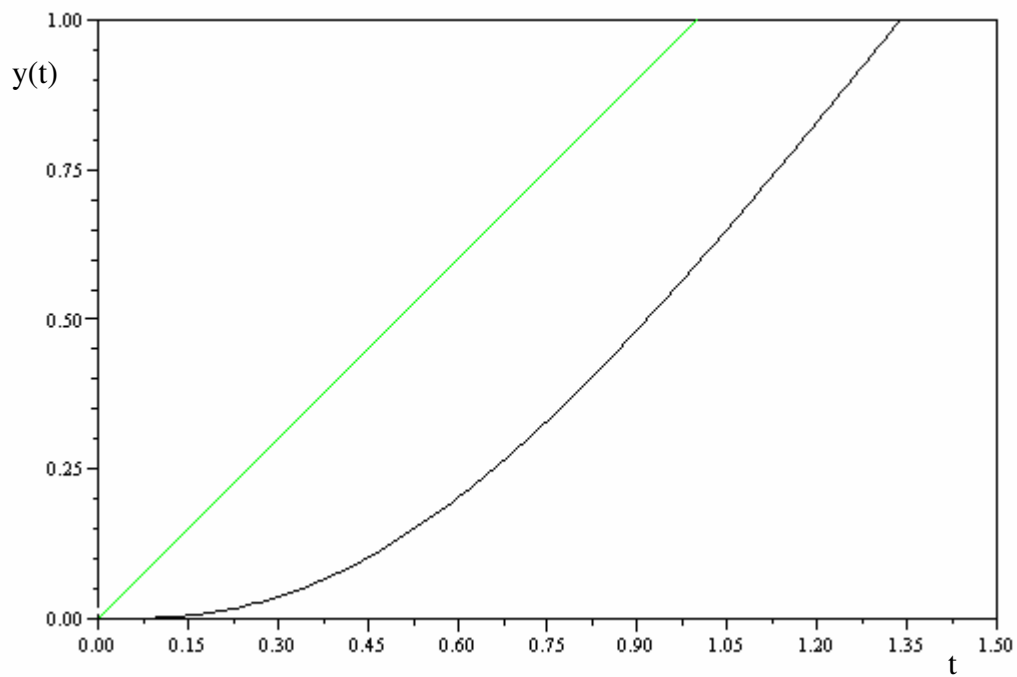
Slika 40: Okno za vpis Scilab programske kode

Tako pripravljen Scifunc blok povežemo s prikazovalnikom (slika 41) ter ga simuliramo.



Slika 41: Uporaba Scifunc bloka v simulacijski shemi

Na prikazovalniku (slika 41) opazujemo, kako se obnaša izhodni signal v odvisnosti od vhodnega.



Slika 42: Odziv sistema na rampo

## 6. SIMULACIJA RAZLIČNIH VRST MODELOV

### 6.1 Linearna diferencialna enačba

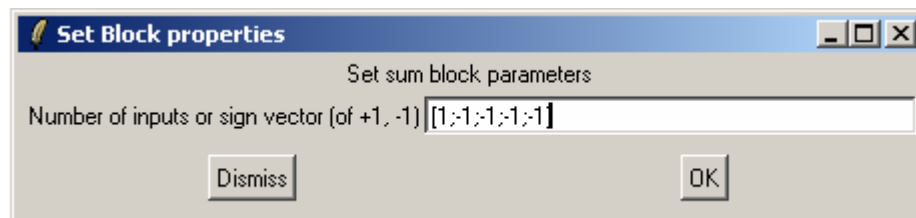
Za primer smo uporabili linearno diferencialno enačbo četrtega reda (6.1) (Scilab/Scicos, 2007).

$$y''''+y'''+y''+0.5y'+0.1y = u \quad ; \quad u = 0.1 \quad ; \quad t \geq 0 \quad (6.1)$$

Enačbo (6.1) preoblikujemo tako, da element najvišjega reda postavimo na levo stran, ostale elemente pa na desno stran.

$$y'''' = -y'''-y''-0.5y'-0.1y+u \quad ; \quad u = 0.1 \quad ; \quad t \geq 0 \quad (6.2)$$

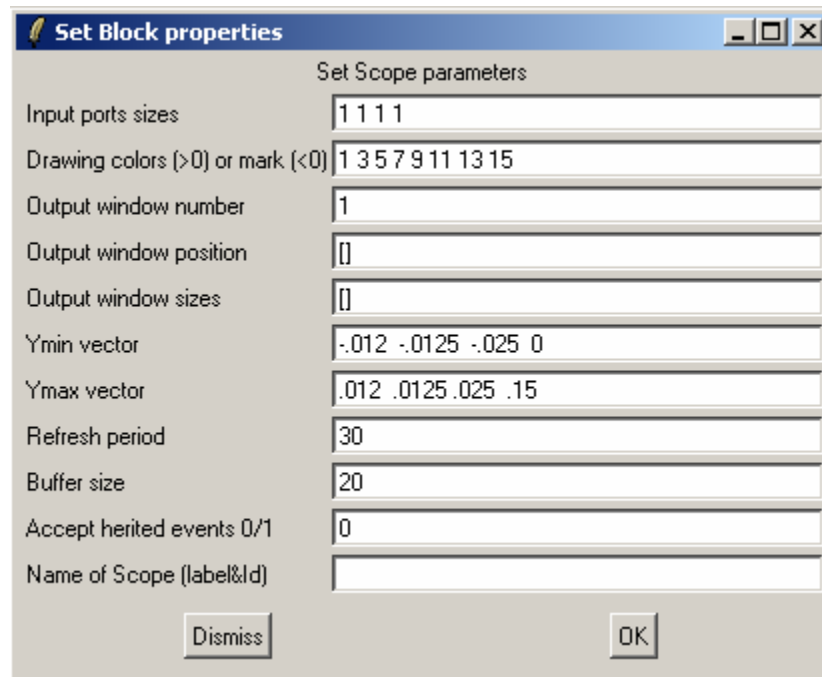
Diferencialno enačbo (6.2) sedaj prenesemo v Scicos tako, da najprej določimo število integratorjev. Število integratorjev je enako velikosti reda najvišjega odvoda rešitve enačbe. V našem primeru postavimo štiri integratorje enega za drugim v ravni vrsti. Pred integratorjem, ki leži na skrajni levi, je najvišji odvod in nato z vsakim integriranjem pada red odvoda do zadnjega. Pred integratorje postavimo sumator ter določimo število in predznak vhodov, zato pogledamo enačbo (6.2) in vidimo, da imamo štiri negativne predznake ter enega pozitivnega. Dvakrat kliknemo na blok in vpišemo vhode (slika 43) v okence.



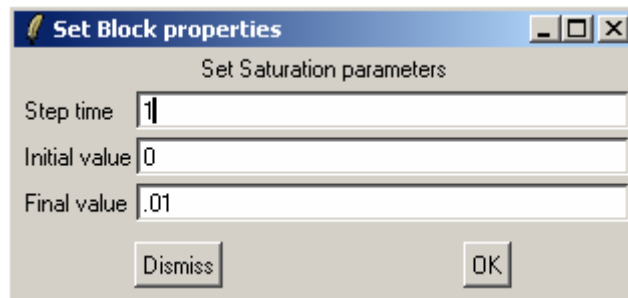
Slika 43: Prilagoditev Sum bloka (blok vsote)

Na delovno površino postavimo še gradnike Gain (ojačenje), v katere vpišemo faktor, ki stoji pred neznanko. V našem primeru je to 0.5 in 0.1. Kot lahko vidimo na sliki 44, smo povezali

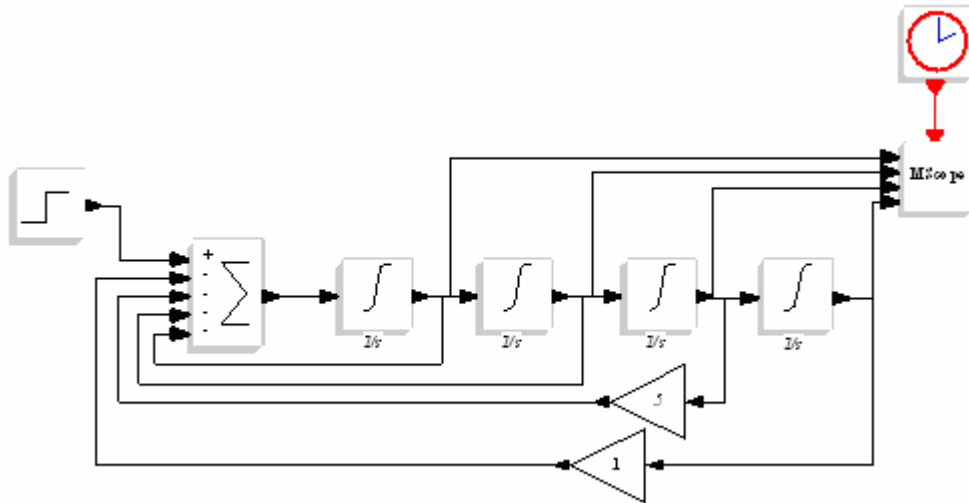
blok Gain (ojačenje) z vrednostjo 0.5 tam, kjer je prvi odvod spremenljivke  $y$  in blok z vrednostjo 0.1 tam, kjer je spremenljivka  $y$ . Na delovno površino vstavimo še blok MScope (večkratni prikazovalnik) in v lastnostih (slika 44) v okencu Input ports size (vnos števila vhodov) vpišemo štiri enice, ki pomenijo štiri vhode na prikazovalniku. Sistem bomo simulirali z enotino stopnico, zato izberemo še gradnik Step function generator (generator stopnice) in nastavimo vrednosti (slika 45). Vse gradnike povežemo (slika 46), nastavimo parametre simulacije ter jo zaženemo.



Slika 44: Prilagoditev izhodnega modula »Mscope« (večkratni prikazovalnik)

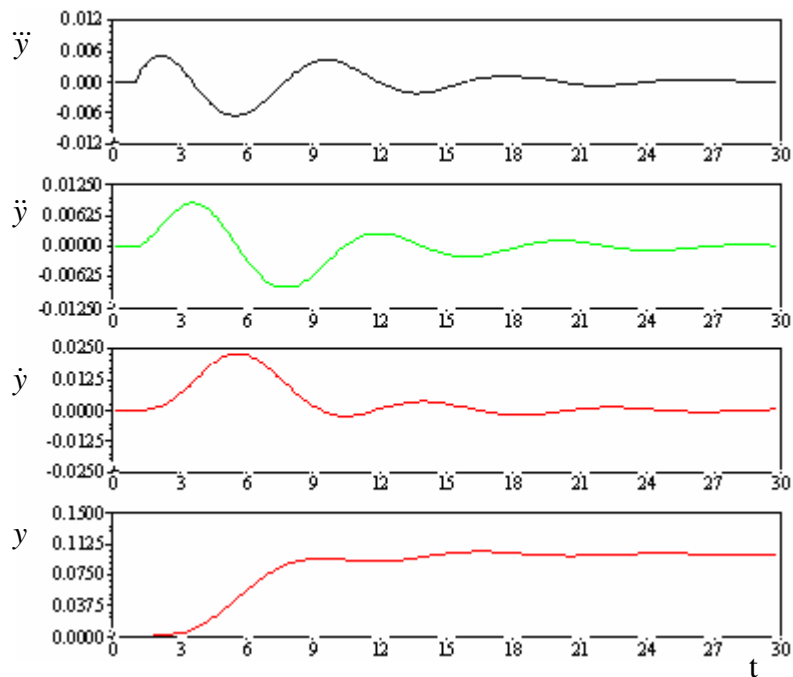


Slika 45: Nastavitev bloka »Generator stopnice«



Slika 46: Simulacijska shema linearne diferencialne enačbe

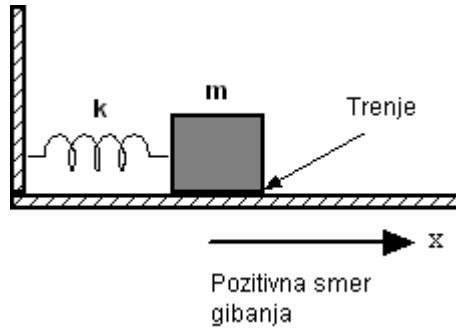
Rezultat simulacije je prikazan na sliki 47. Vsak graf prikazuje signal na izhodu enega integrala.



Slika 47: Rezultat linearne diferencialne enačbe

## 6.2 Nelinearna diferencialna enačba

Za prikaz reševanja nelinearne diferencialne enačbe smo uporabil primer iz fizike (Scilab/Scicos, 2007). Telo z maso  $m$  potegnemo v desno za  $x = 1m$  in izpustimo (slika 48). Pri tem bomo upoštevali koeficient trenja  $f$  in koeficient vzmeti  $k$ , ki se upirata spremembi. Zanima nas obnašanje sistema.



Slika 48: Fizikalni problem nihanja uteži po podlagi

Diferencialna enačba, ki opisuje primer je:

$$m\ddot{x} = -kx - fmg \quad \text{pri } \dot{x} > 0$$

$$m\ddot{x} = -kx + fmg \quad \text{pri } \dot{x} < 0$$

(6.3)

Enačbo preoblikujemo:

$$m\ddot{x} + kx + fmg = 0 \quad \dot{x} > 0$$

$$m\ddot{x} + kx - fmg = 0 \quad \dot{x} < 0$$

$$\ddot{x} + \frac{k}{m}x + fg = 0 \quad \text{pri } \dot{x} > 0$$

$$\ddot{x} + \frac{k}{m}\left(x + \frac{fmg}{k}\right) = 0 \quad \text{pri } \dot{x} > 0$$

$$\ddot{x} + \frac{k}{m}x - fg = 0 \quad \text{pri } \dot{x} < 0 \quad \text{ali}$$

$$\ddot{x} + \frac{k}{m}\left(x - \frac{fmg}{m}\right) = 0 \quad \text{pri } \dot{x} < 0$$

$$\ddot{x} = -\frac{k}{m}\left(x + \frac{fmg}{k}\right) \quad \text{pri } \dot{x} > 0$$



$$\ddot{x} = -\frac{k}{m}\left(x - \frac{fmg}{k}\right) \text{ pri } \dot{x} < 0 \quad (6.4)$$

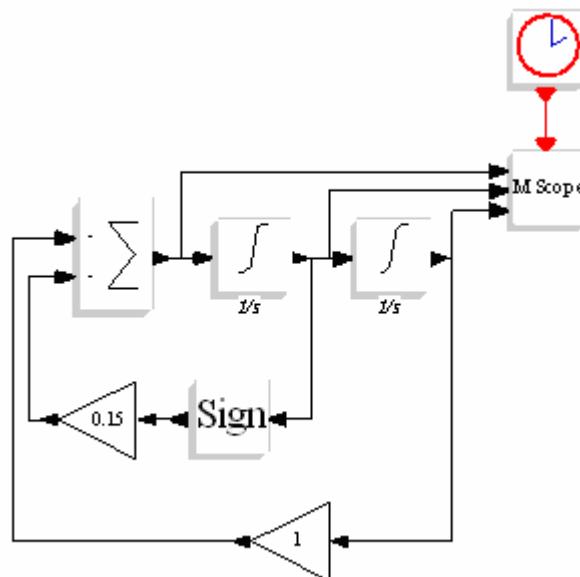
Za vrednosti parametrov vzamemo naslednje vrednosti:

$$\begin{aligned} \frac{k}{m} &= 1 \\ \frac{fmg}{k} &= 0.15 \end{aligned} \quad (6.5)$$

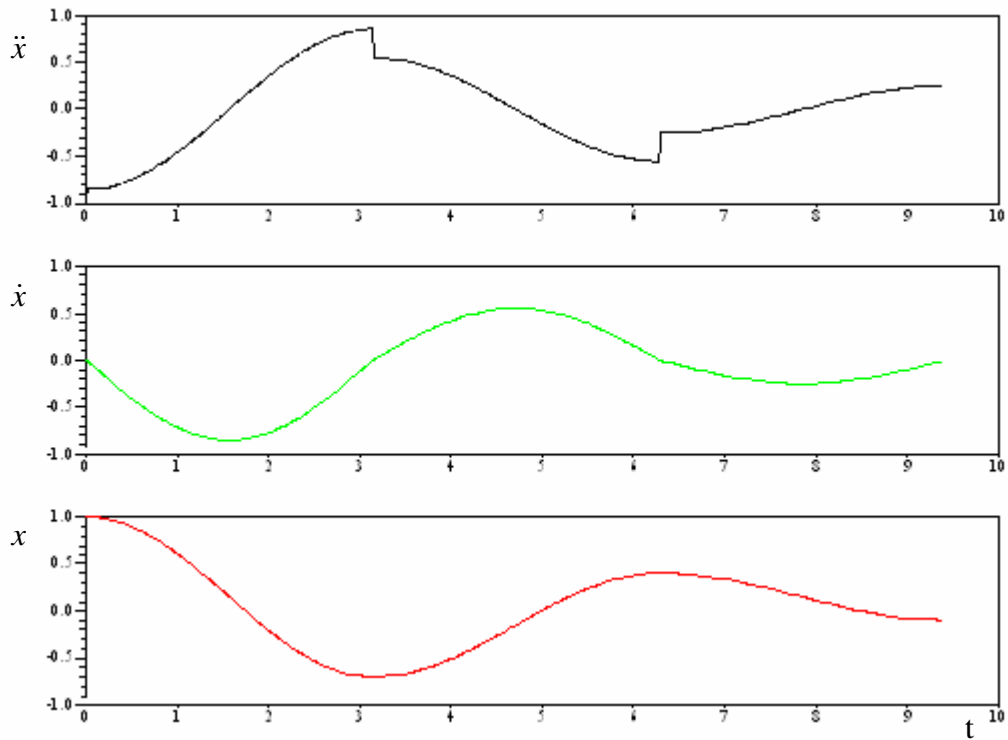
Sedaj zapišemo diferencialno enačbo:

$$\ddot{x} = -x - 0.15\text{sign}(\dot{x}) \quad (6.6)$$

Sign je funkcija, ki da na izhodu vrednost 1, ko je na vhodu več kot 0, -1, ko je na vhodu manj kot 0 in 0, ko je na vhodu vrednost natanko 0. Diferencialno enačbo (6.6) prenesemo v simulacijsko shemo kakor kaže slika 49, nastavimo ustrezne parametre gradnikov in simulacije. Začetna vrednost simulacije je  $x = 1$ . Zaženemo simulacijo in na prikazovalniku dobimo vrednosti, ki so prikazane na sliki 50.



Slika 49: Primer simulacijske sheme nelinearne diferencialne enačbe v Scicos-u



Slika 50: Grafični prikaz spremenljivk nelinearne diferencialne enačbe

### 6.3 Sistem diferencialnih enačb

Poglejmo si, kako naredimo simulacijsko shemo za prikaz rešitve sistema diferencialnih enačb (Kocijan, 1996).

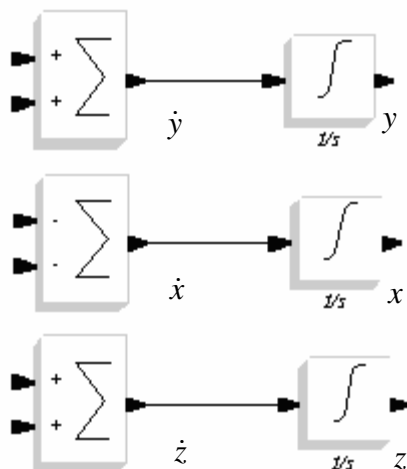
Rösslerjev kaotični model je definiran s sistemom diferencialnih enačb:

$$\begin{aligned}\dot{x} &= -(y + z) \\ \dot{y} &= x + ay \\ \dot{z} &= b + z(x - c)\end{aligned}$$

(6.7)

Simulacijsko shemo začnemo risati tako, da najprej določimo, koliko integratorjev potrebujemo. Ker imamo tri diferencialne enačbe prvega reda (6.7), bomo v simulacijsko

shemo vnesli tri integratorje. V vsaki diferencialni enačbi se vrednosti seštevajo ali odštevajo, zato bomo pred integratorje dodali še seštevalnike, ki jim ustrezno spremenimo predznak vhodov. Seštevalnike in integratorje nato povežemo (slika 51).



Slika 51: Izberemo tri integratorje za tri diferencialne enačbe

Če sedaj pogledamo vsako diferencialno enačbo posebej, lahko dopolnimo simulacijsko shemo.

$$\dot{x} = -(y + z) \tag{6.8}$$

Iz enačbe (6.8) vidimo, da  $\dot{x}$  dobimo tako, da seštejemo med seboj  $-y$  in  $-z$ . Zato smo določili seštevalniku pri  $\dot{x}$  dva negativna vhoda. Izhod iz  $y$  integratorja povežemo z negativnim vhomom na  $x$  seštevalniku in izhod  $z$  integratorja povežemo z drugim negativnim vhomom seštevalnika.

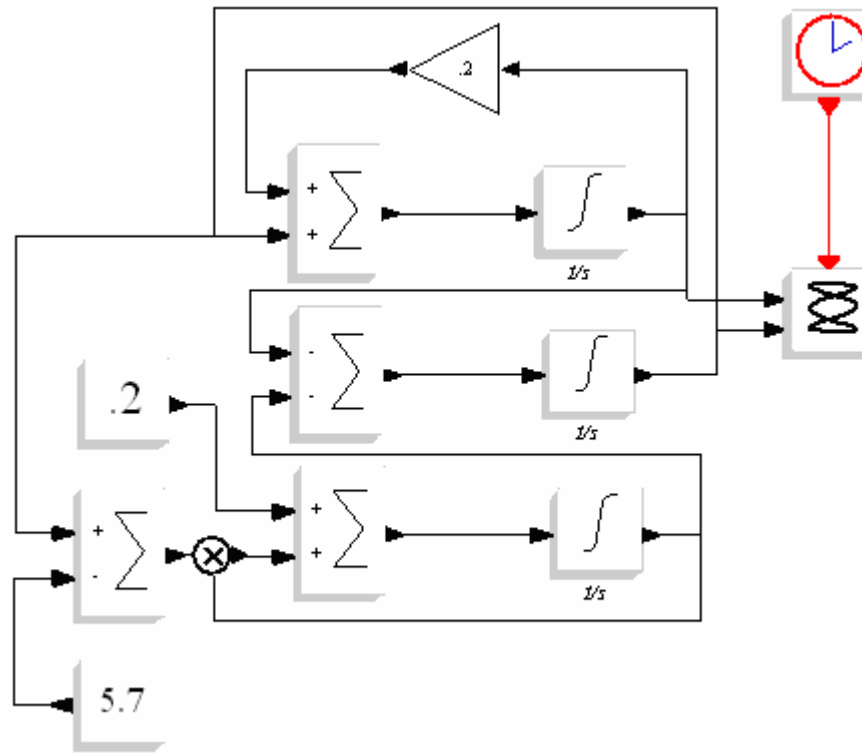
$$\dot{y} = x + ay \tag{6.9}$$

Iz enačbe (6.9) vidimo, da dobimo  $\dot{y}$  tako, da seštejemo  $x$  in  $ay$ . Izhod  $x$  iz integratorja povežemo s pozitivnim vhodom na seštevalniku in izhod  $y$  integratorja povežemo z drugim pozitivnim vhodom seštevalnika z ojačenjem  $a = 0.2$ .

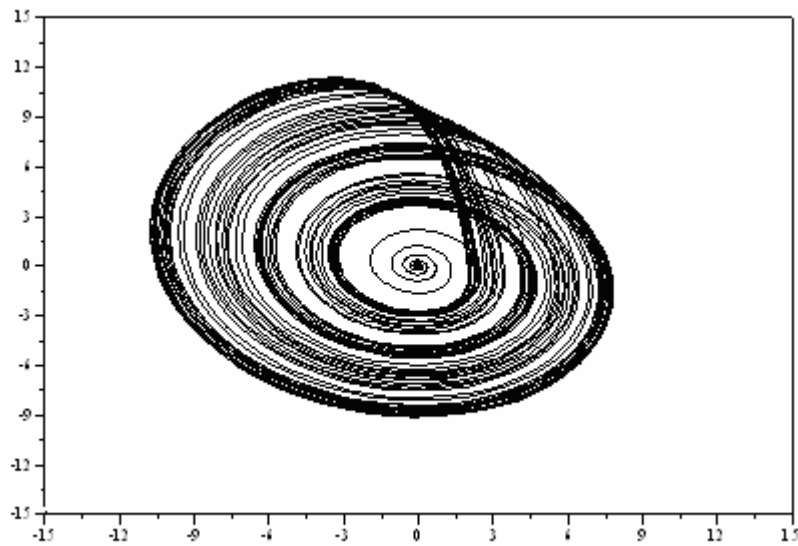
$$\dot{z} = b + z(x - c) \tag{6.10}$$

Za diferencialno enačbo (6.10) moramo v simulacijsko shemo vnesti še en sumator in gradnik za množenje. Najprej rešimo  $z(x - c)$ . Izhod  $x$  iz integratorja povežemo s pozitivnim vhodom dodatnega seštevalnika in izbrano konstanto  $c = 5.7$  povežemo na drugi negativni vhod seštevalnika. Ta seštevalnik povežemo na gradnik za množenje ter nanj povežemo še izhod iz  $z$  integratorja. Izrazu  $z(x - c)$  prištejemo še izbrano konstanto  $b = 0.2$  tako, da povežemo izhod iz gradnika za množenje na pozitivni vhod  $z$  seštevalnika in konstanto  $b = 0.2$  na drugi pozitivni vhod iz seštevalnika.

Končana simulacijska shema diferencialnih enačb je prikazana na sliki 52, s parametri  $a = b = 0.2$  in  $c = 5.7$ . Začetne vrednosti so nastavljene na nič. Za prikaz vrednosti  $x$  ter  $y$  je uporabljen XY prikazovalnik. Rezultat je prikazan na sliki 53.



Slika 52: Rösslerjev kaotični model v Scicos-u



Slika 53: Grafični prikaz rešitve sistema (atraktor kaotičnega sistema)

## 6.4 Prenosna funkcija

Za simulacijo dinamičnega sistema, zapisanega s prenosno funkcijo (6.11) v Scicos-u, si bomo pogledali model dviganja in spuščanja reaktivnega letala (Kocijan, 1996). Prenosna funkcija (6.11) opisuje spremembo kota dviganja oziroma spuščanja letala glede na položaj višinskega krmila.

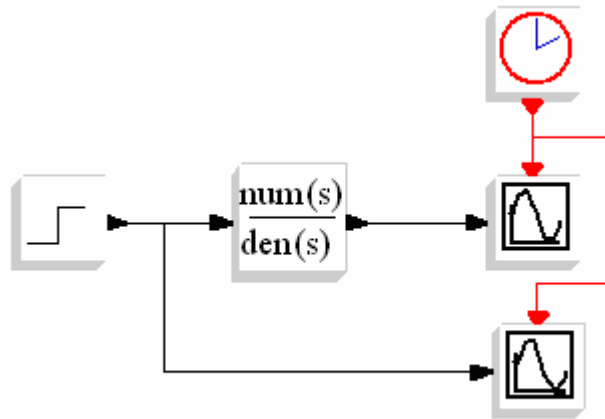
$$G(s) = \frac{-18(s+0.015)(s+0.45)}{(s^2+1.2s+12)(s^2+0.01s+0.0025)} \quad (6.11)$$

S programskim modulom Scicos bomo simulirali odziv reaktivnega letala na stopničasto spremembo položaja višinskega krmila za eno enoto  $u(t) = 1 ; t \geq 0$  oziroma  $U(s) = \frac{1}{s}$ .

Prenosno funkcijo lahko najprej preoblikujemo (6.12):

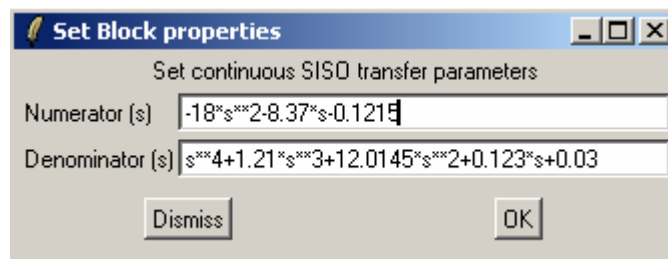
$$G(s) = \frac{Y(s)}{U(s)} = \frac{-18s^2 - 8.37s - 0.1215}{s^4 + 1.21s^3 + 12.0145s^2 + 0.123s + 0.03} \quad (6.12)$$

Zaženemo programski modul Scicos in v knjižnici Linear izberemo blok za simulacijo prenosnih funkcij in ga postavimo na delovno površino. Za vhod izberemo stopničasti signal ter za izhod prikazovalnik. Vse skupaj povežemo tako, kot je prikazano na sliki 54.



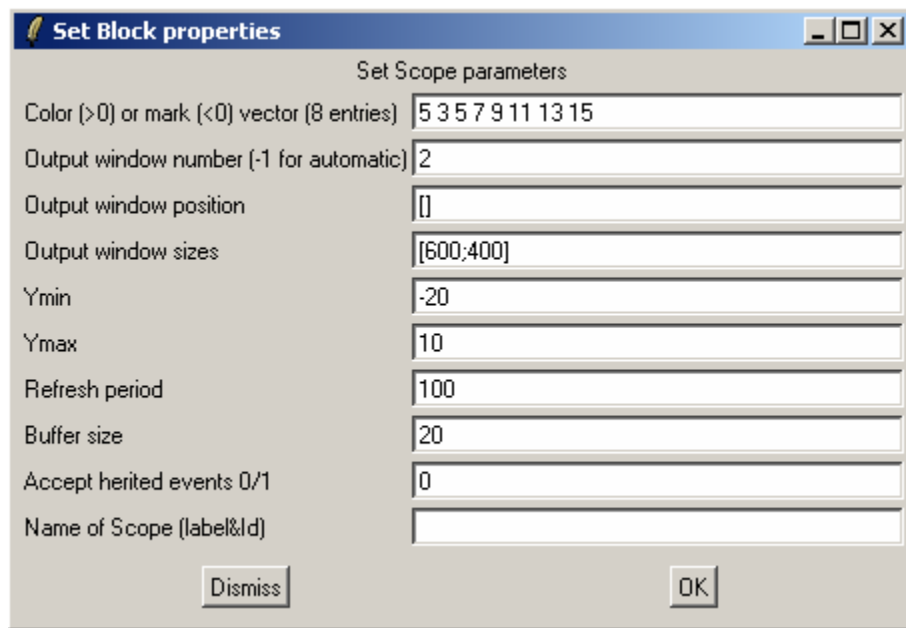
Slika 54: Simulacija reaktivnega letala

V blok za simulacijo prenosnih funkcij vnesemo enačbo tako, kot je prikazano na sliki 55. Pri tem moramo biti pazljivi na to, da pri množenju uporabimo zvezdico in pri potenciranju dve zvezdici (\*\*) ali strešico (^). Za ločilo decimalnih števil uporabimo decimalno piko.



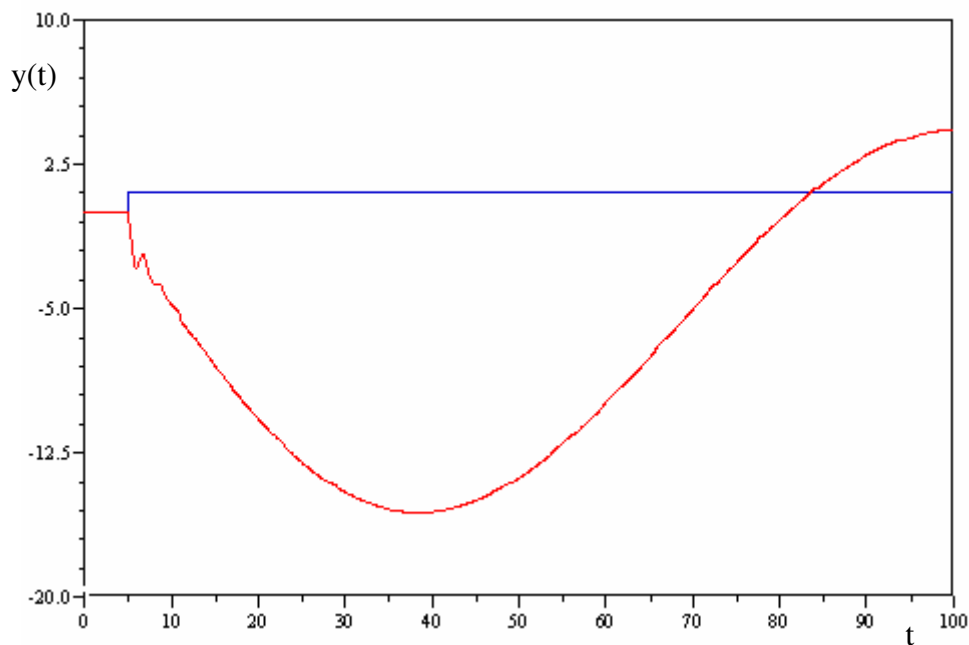
Slika 55: Vnašanje parametrov v blok

Simulaciji nastavimo še čas trajanja, ki smo ga v našem primeru nastavili na vrednost 100, ter na prikazovalniku pod točko Refresh period prav tako nastavimo vrednost 100. Ker imamo v projektu dva prikazovalnika, se pri zagonu pojavita dve okni, vsako za svoj prikazovalnik. Da bi izrisali obe krivulji v istem oknu, nastavimo na obeh prikazovalnikih (slika 56) pod točko Output window number (številka izhodnega okna) na primer 2. Tako se bo graf izrisal v obeh primerih v okno 2.



Slika 56: Nastavitev prikazovalnika

Ko zaženemo simulacijo, se izriše krivulja (slika 57), ki opisuje odziv prenosne funkcije dviganja in spuščanja reaktivnega letala, kadar premaknemo krmilno palico za dviganje in spuščanje. Premik krmilne palice ponazarja stopničast signal na vhodu.



Slika 57: Grafični prikaz odziva letala na premik višinskega krmila

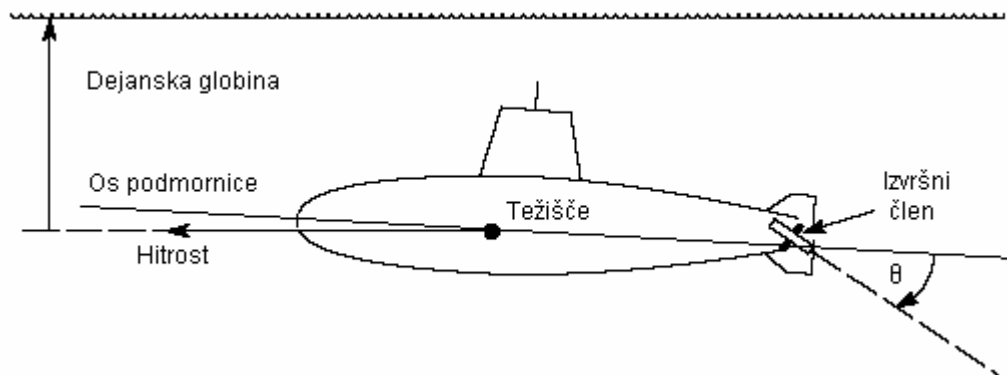


## 6.5 Modularno sestavljeni sistemi

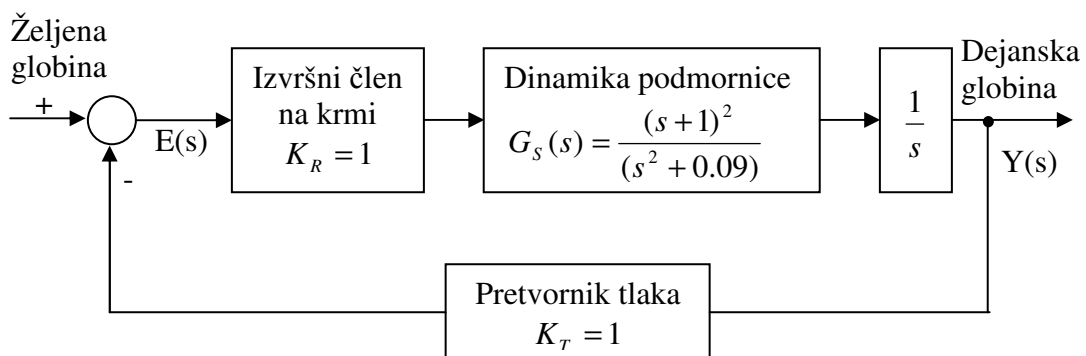
Za prikaz modularno sestavljenega sistema bomo uporabili primer avtomatskega sistema vodenja za spreminjanje globine podmornice (slika 58) (Kocijan, 1996). Globino podmornice merimo z merilnikom tlaka. Signal pretvornika tlaka primerjamo z vrednostjo željene globine. Ko regulator ugotovi razliko med željeno in dejansko vrednostjo signala, nastavi kot krme  $\theta$ . Bločni diagram je prikazan na sliki 59.

Proces je opisan s prenosno funkcijo (6.12):

$$G_S(s) = \frac{(s+1)^2}{(s^2+0.09)} \quad (6.12)$$

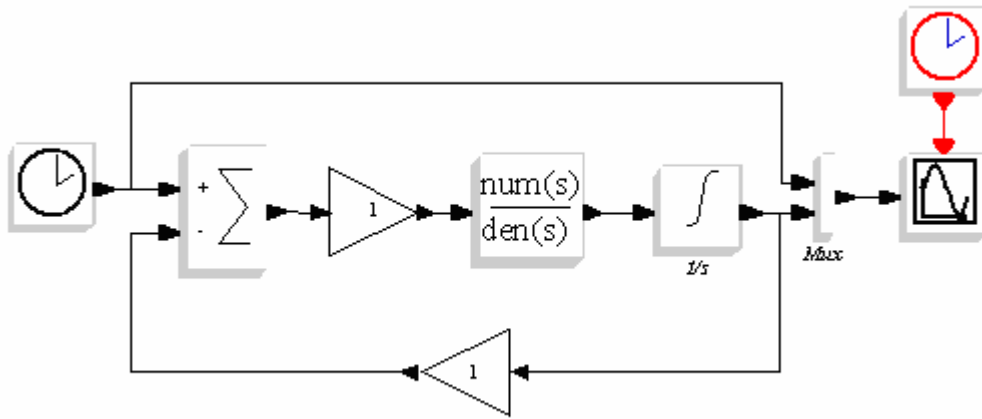


Slika 58: Princip delovanje spremembe globine podmornice

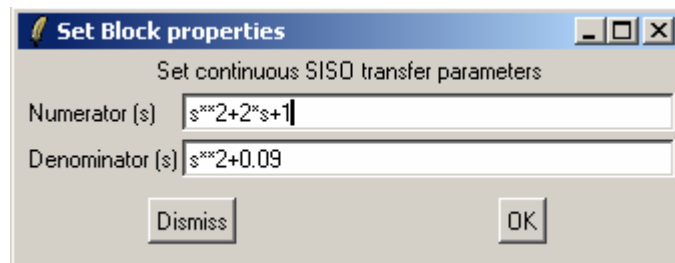


Slika 59: Bločni diagram vodenja za spreminjanje globine podmornice

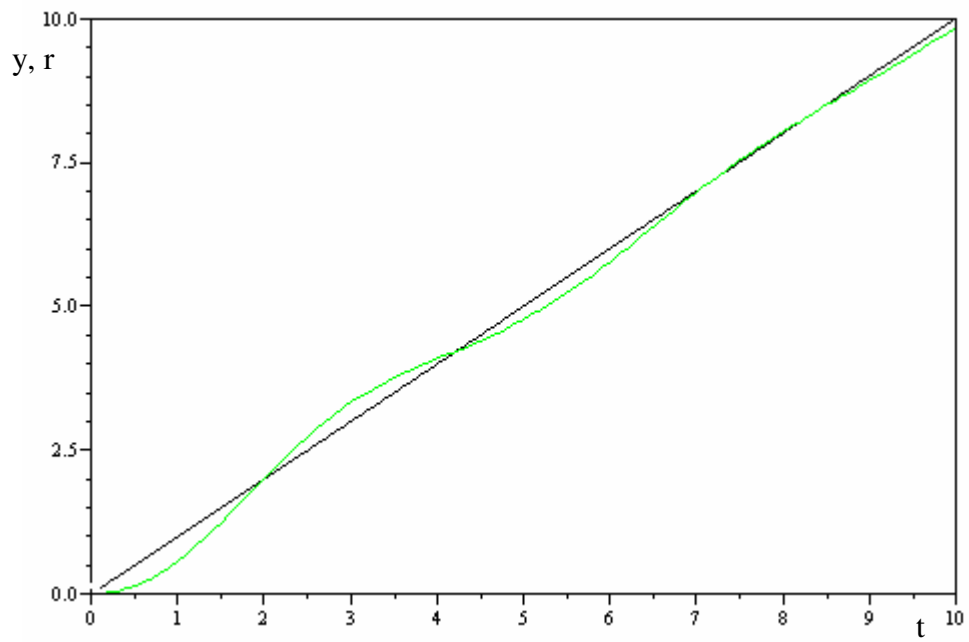
Bločni diagram prenesemo v simulacijsko shemo (slika 60) ter v bloke vpišemo ustrezne parametre (slika 61). Sistem simuliramo in na izhodu dobimo grafični prikaz (slika 62) odziva podmornice glede na željeno globino.



Slika 60: Simulacijska shema vodjenja podmornice



Slika 61: Parametri bloka »Num(s)/Den(s)«



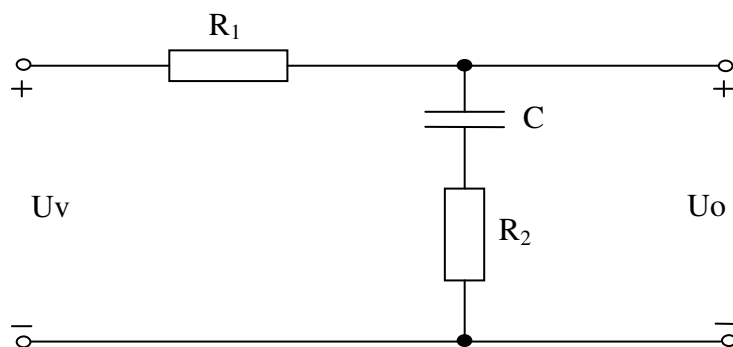
Slika 62: Sprememba globine podmornice glede na željeno globino



## 7. PRIMERI UPORABE

### 7.1 Elektrotehnika: Odziv RC vezja

Za primer bomo vzeli RC vezje (slika 63) ter ugotavljali, kaj se dogaja na izhodu  $U_o$ , če damo na vhod  $U_v$  pravokotni napetostni impulz.



Slika 63: RC vezje

Zančni enačbi:

$$U_i(t) = U_{R_1}(t) + U_c(t) + U_{R_2}(t)$$

$$U_o(t) = U_c(t) + U_{R_2}(t)$$

(7.1)

Obe enačbi transformiramo (7.2) z Laplaceovo transformacijo:

$$U_i(s) = U_v = R_1 I(s) + \frac{1}{sC} I(s) + R_2 I(s)$$

$$U_o = \frac{1}{sC} I(s) + R_2 I(s)$$

(7.2)

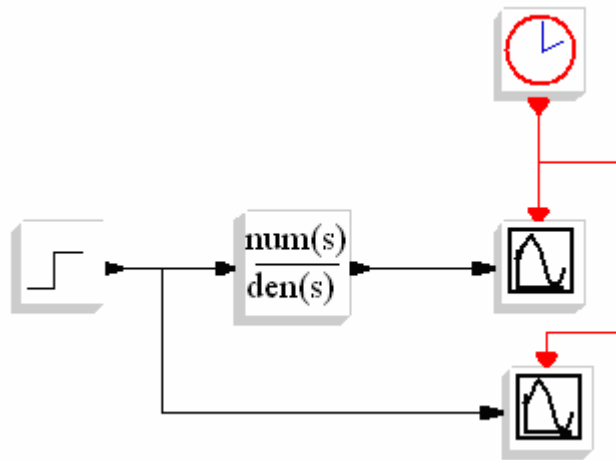
Dobimo prenosno funkcijo:

$$G_s = \frac{U_o}{U_v} = \frac{\frac{1}{sC} + R_2}{R_1 + R_2 + \frac{1}{sC}} = \frac{1 + sCR_2}{1 + sCR_1 + sCR_2}$$

(7.3)

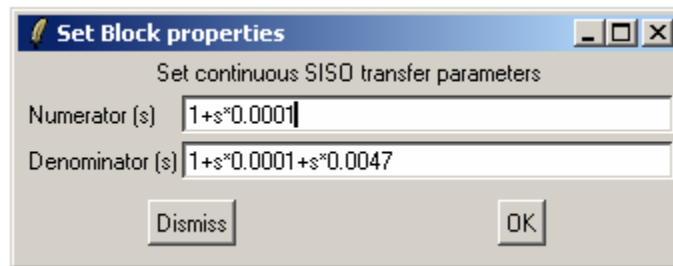
Za ta primer so bile vrednosti gradnikov:  $R_1=47k\Omega$ ,  $R_2=1k\Omega$ ,  $C=100nF$

V Scicosu-u postavimo gradnike in jih povežemo tako, kot je prikazano na sliki 64.



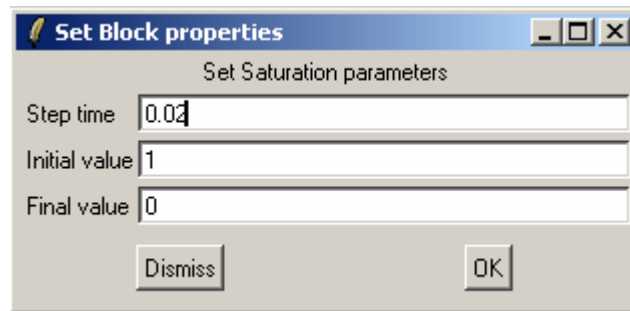
Slika 64: Simulacijska shema modela R-C vezja

Uporabili smo blok za prenosno funkcijo, ki ga najdemo v knjižnici linearnih gradnikov. V blok vpišemo izračunane vrednosti tako, kot je prikazano na sliki 65 in potrdimo.



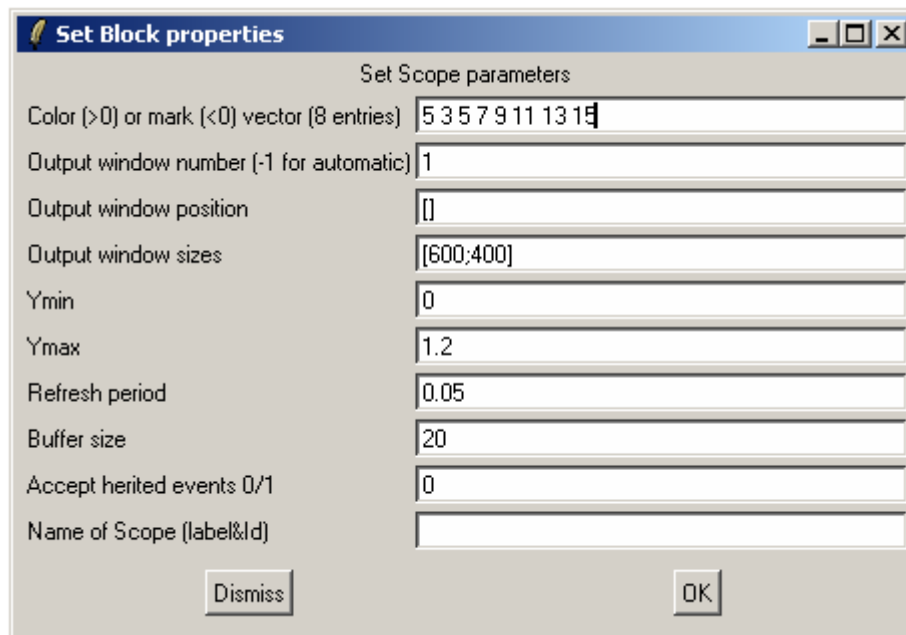
Slika 65: Vpis parametrov v blok Num(s)/Den(s)

Impulz vhodne napetosti smo izvedli z zakasnjeno stopnico na vhodu tako, da smo nastavili (slika 66) Initial value (začetna vrednost) 1 ter Final value (končno vrednost) 0. V simulaciji bomo videli, da na ta način lahko opazujemo, kako na vhodu napetost narašča, dokler je na vhodu 1 in kako napetost pada, kadar je na vhodu 0. Zaradi tega smo povečali step time (čas trajanja impulza) in tako dosegli na izhodu višjo napetost.



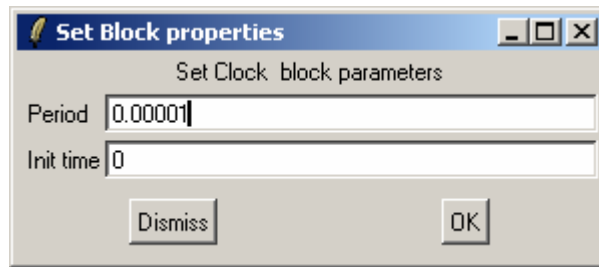
Slika 66: Nastavitev stopnice

Pred začetkom simulacije nastavimo še vrednosti prikazovalnika tako, kot je prikazano na sliki 67. Ker imamo v primeru dva prikazovalnika, je potrebno oba nastaviti popolnoma enako, razen pri izbiri barve izrisa, kjer izberemo različne vrednosti.



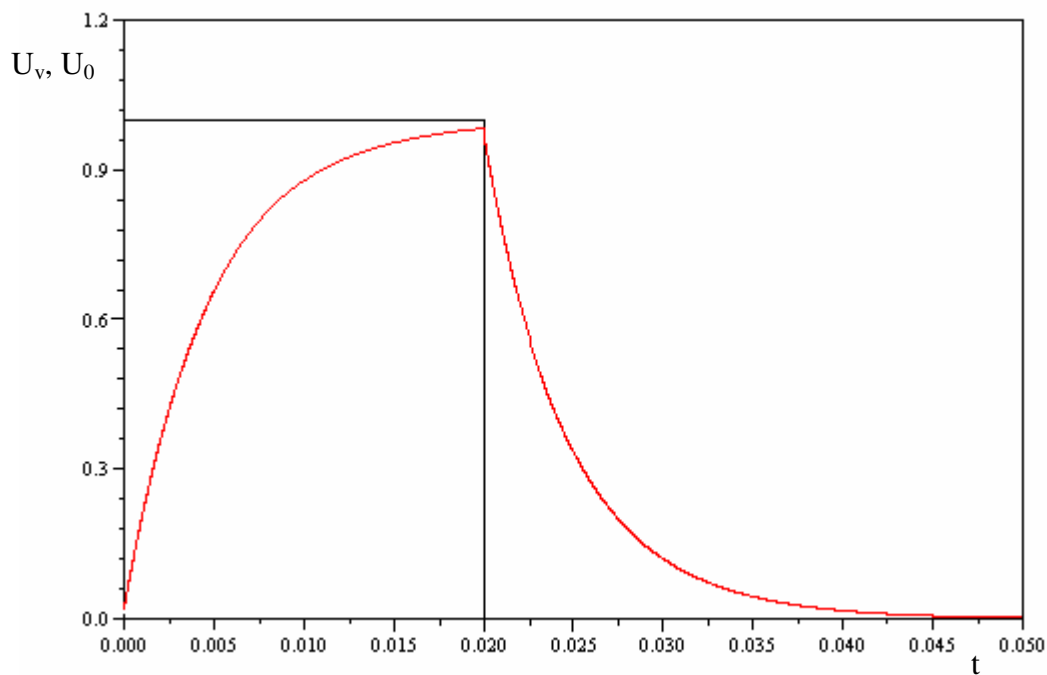
Slika 67: Nastavitev prikazovalnika

Ker je čas simulacije zelo majhen (približno 50ms), je potrebno nastaviti tudi uro za sinhronizacijo sistema (slika 68). Manjša kot je vrednost, bolj natančen bo izris na prikazovalniku, zaradi večje gostote točk. Z večanjem števila točk se povečuje čas, potreben za izris grafa.



Slika 68: Nastavitev parametrov ure za sinhronizacijo

Na prikazovalniku (slika 69) lahko vidimo, kako se spreminja izhodna napetost glede na vhodno. V prvi fazi, dokler je na vhodu 1, se kondenzator polni in izhodna napetost narašča. V drugem delu je vhodna napetost enaka 0 in kondenzator se začne prazniti.

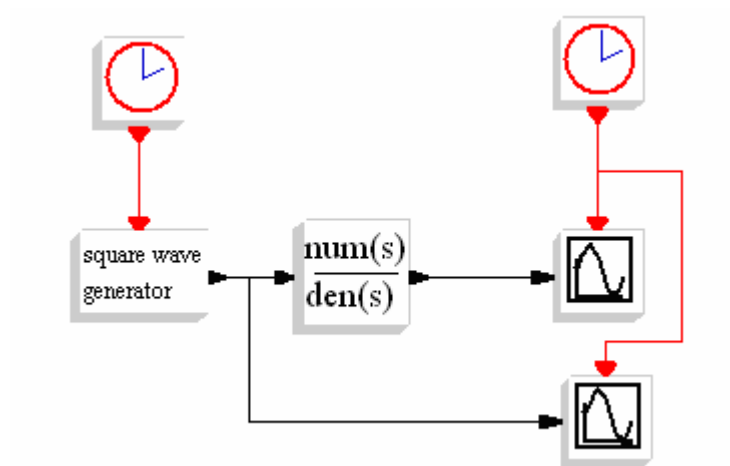


Slika 69: Krivulja izhodne napetosti v odvisnosti od vhodnega signala

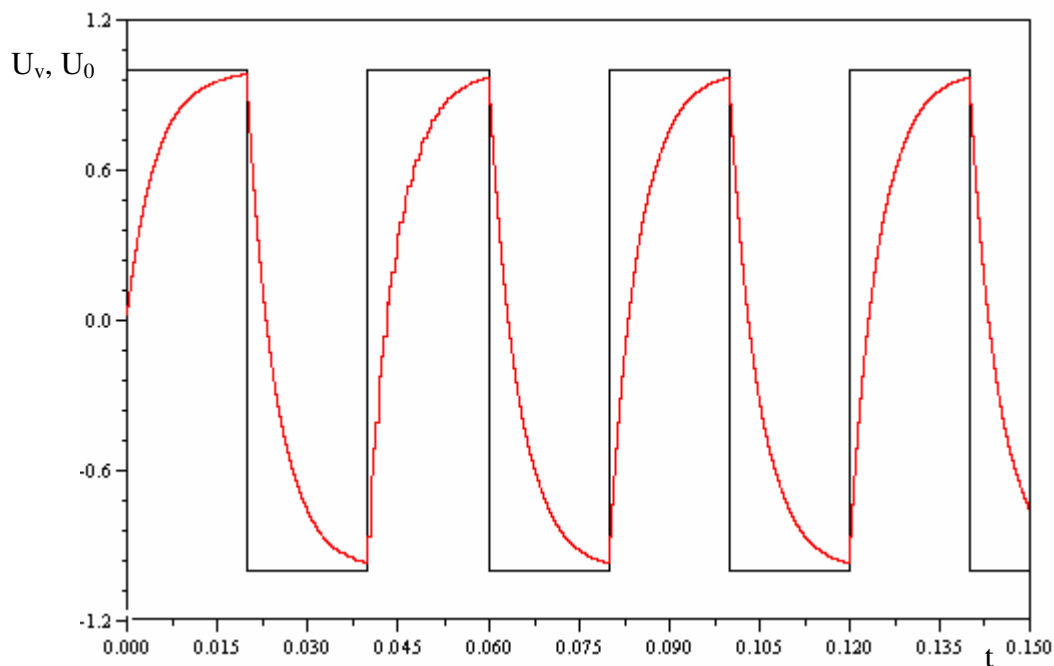
Če na vhod vezja priključimo generator pravokotnega signala (slika 70), lahko opazujemo, kako se spreminja napetost na izhodu glede na napetost in frekvenco vhodnega signala (slika 71).



Na generator pravokotnega signala priključimo sinhronizacijsko uro, s katero določimo frekvenco signala. Ko imamo na vohodu vezja pozitiven signal, se kondenzator polni in napetost na izhodu narašča. Če bi bil čas pozitivnega signala daljši, bi lahko opazili, da napetost na izhodu ne bi več naraščala. V nadaljevanju dobimo na vohodu signal negativne vrednosti in kondenzator se prične prazniti in nato polniti, ampak tokrat je nasprotno polariziran.

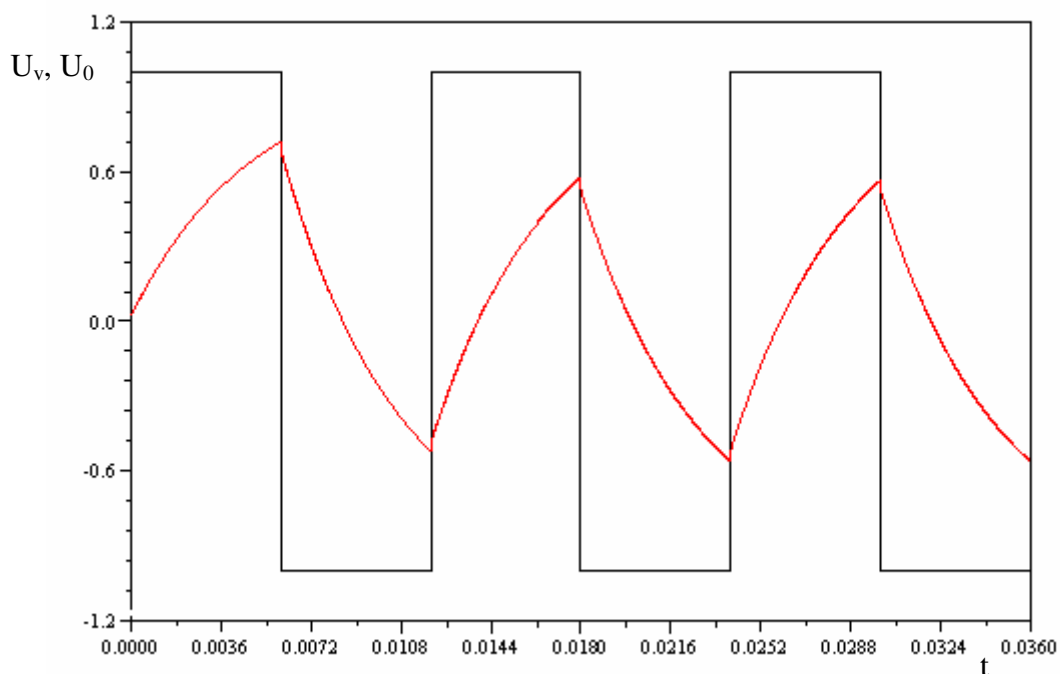


Slika 70: Simulacijska shema RC vezja z generatorjem pravokotnega signala na vohodu



Slika 71: Izris odziva z generatorjem pravokotnega signala na vohodu

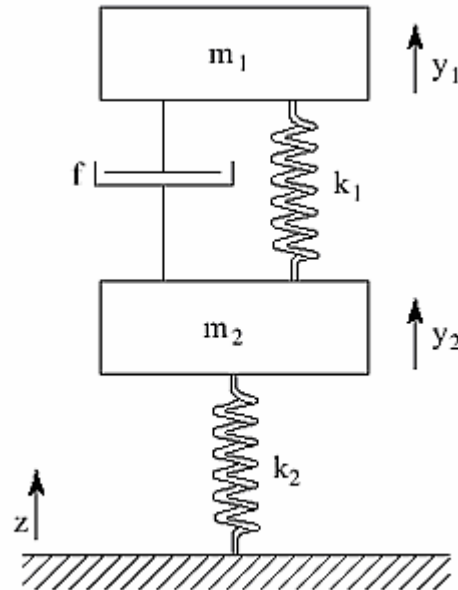
Kaj pa se zgodi, če skrajšamo čas polnjenja? To dosežemo tako, da povečamo frekvenco signala na vhodu. Uro za sinhronizacijo nastavimo na 6ms in parametre simulacije ter prikazovalnika nastavimo tako, da je graf čimbolj pregleden. Na izrisanem grafu (slika 72) lahko vidimo, da je napetost na izhodu v tem času narasla le na približno 0.7V in nato začela padati. Isto, le z nasprotnim predznakom, se zgodi tudi v drugi polperiodi. Signal na izhodu ni pravokoten temveč žagast, zato takšno vezje lahko uporabimo za preoblikovanje signala.



Slika 72: Prikaz izhodnega signala, če imamo na vhodu generator pravokotnega signala

## 7.2 Mehanika: Poenostavljeno avtomobilsko vzmetenje

V tem podpoglavju bomo simulirali odziv poenostavljenega avtomobilskega vzmetenja (slika 73), če avtomobil trči ob 0.1m visok robnik.



Slika 73: Poenostavljen model avtomobilskega vzmetenja

- $m_1$  – masa četrtine avtomobila [kg],
- $m_2$  – masa avtomobilske polosi [kg],
- $k_1$  – konstanta vzmetenja  $\left[\frac{\text{N}}{\text{m}}\right]$  avtomobila,
- $f$  – konstanta dušilnika  $\left[\frac{\text{Ns}}{\text{m}}\right]$ ,
- $k_2$  – konstanta amortizerja kolesa  $\left[\frac{\text{N}}{\text{m}}\right]$ ,
- $y_1$  – premik avtomobilske karoserije [m],
- $y_2$  – premik kolesa [m],
- $z$  – profil ceste [m]

Poenostavljen model avtomobilskega vzmetenja lahko opišemo s sistemom dveh enačb (7.4):

$$\begin{aligned} m_1 \ddot{y}_1 + f(\dot{y}_1 - \dot{y}_2) + k_1(y_1 - y_2) &= 0 \\ m_2 \ddot{y}_2 + f(\dot{y}_2 - \dot{y}_1) + k_1(y_2 - y_1) + k_2(y_2 - z) &= 0 \end{aligned} \quad (7.4)$$

Enačbe preoblikujemo(7.5):

$$\begin{aligned} \ddot{y}_1 &= -\frac{f}{m_1}(\dot{y}_1 - \dot{y}_2) - \frac{k_1}{m_1}(y_1 - y_2) \\ \ddot{y}_2 &= \frac{f}{m_2}(\dot{y}_1 - \dot{y}_2) + \frac{k_1}{m_2}(y_1 - y_2) + \frac{k_2}{m_2}(z - y_2) \end{aligned} \quad (7.5)$$

Vrednost konstant:

$$\begin{aligned} m_1 &= 250 \text{ kg} & m_2 &= 20 \text{ kg} \\ k_2 &= 5 \cdot 10^4 \text{ N/m} & k_1 &= 10^4 \text{ N/m} \\ f &= 1000 \text{ Ns/m} & z &= 0.1 \text{ m} \end{aligned}$$

Simulacijsko shemo pričnemo risati tako, da najprej določimo, koliko integratorjev potrebujemo. Ker imamo dve diferencialni enačbi drugega reda, bomo potrebovali za vsako dva integratorja, torej skupaj štiri integratorje.

Začnemo risati diferencialno enačbo (7.6):

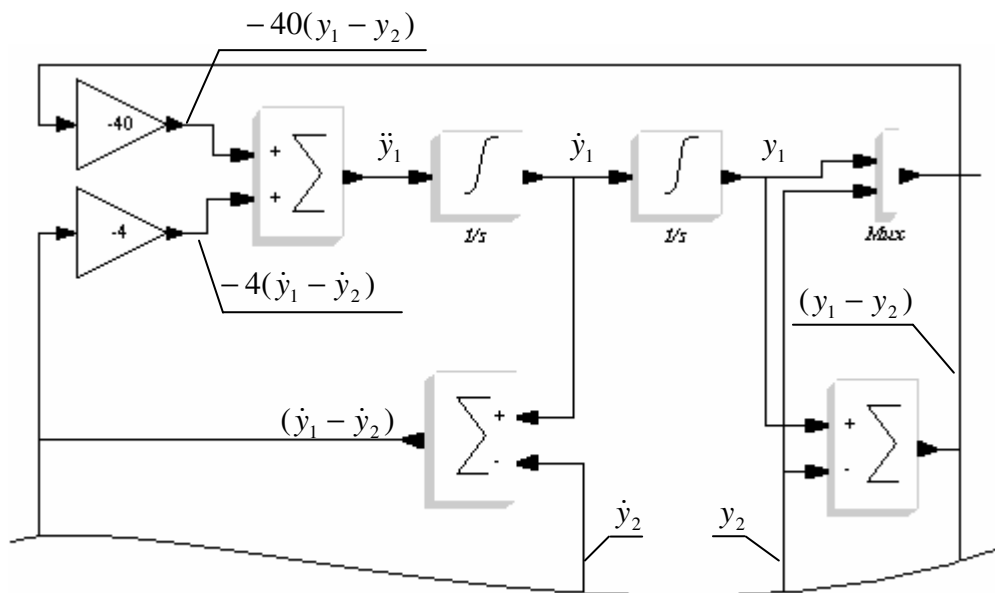
$$\ddot{y}_1 = -\frac{f}{m_1}(\dot{y}_1 - \dot{y}_2) - \frac{k_1}{m_1}(y_1 - y_2) \quad (7.6)$$

Izračunamo ojačenja:

$$\frac{f}{m_1} = 4$$

$$\frac{k_1}{m_1} = 40$$

Sedaj rešimo izraze v oklepajih. Vzamemo izraz  $(\dot{y}_1 - \dot{y}_2)$  in ga prenesemo v simulacijsko shemo. Potrebujemo seštevalnik z dvema vhodoma različnih predznakov in ju povežemo. Enako naredimo tudi z izrazom v drugem oklepaju. Izrazoma dodamo ojačenja, ki imata v obeh primerih negativen predznak. Postopek je prikazan na sliki 74. Vse izhode sistema nato priključimo na multiplekser. To je še en način zajemanja več izhodnih signalov, ko dva signala vodimo po eni povezavi. Namesto multiplekserja bi lahko uporabili dva prikazovalnika ali blok Mscope (prikazovalnik z več vhodi).



Slika 74: Potek risanje simulacijske sheme za  $\ddot{y}_1$

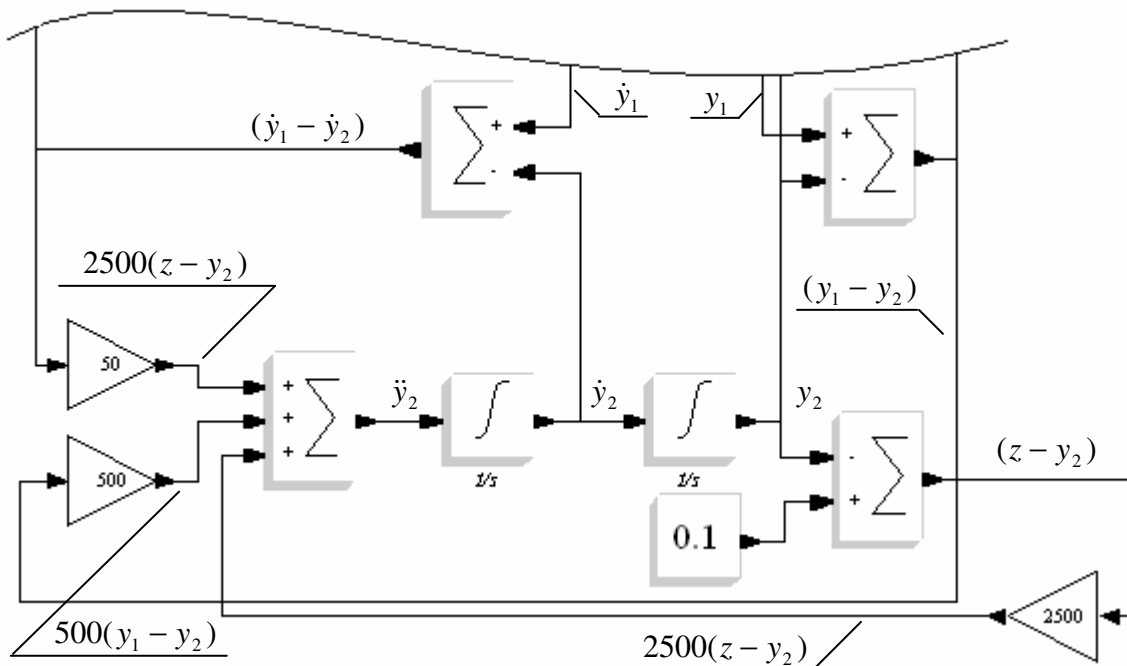
Nadaljujemo z diferencialno enačbo (7.7):

$$\ddot{y}_2 = \frac{f}{m_2}(\dot{y}_1 - \dot{y}_2) + \frac{k_1}{m_2}(y_1 - y_2) + \frac{k_2}{m_2}(z - y_2) \quad (7.7)$$

Izračunamo ojačenja:

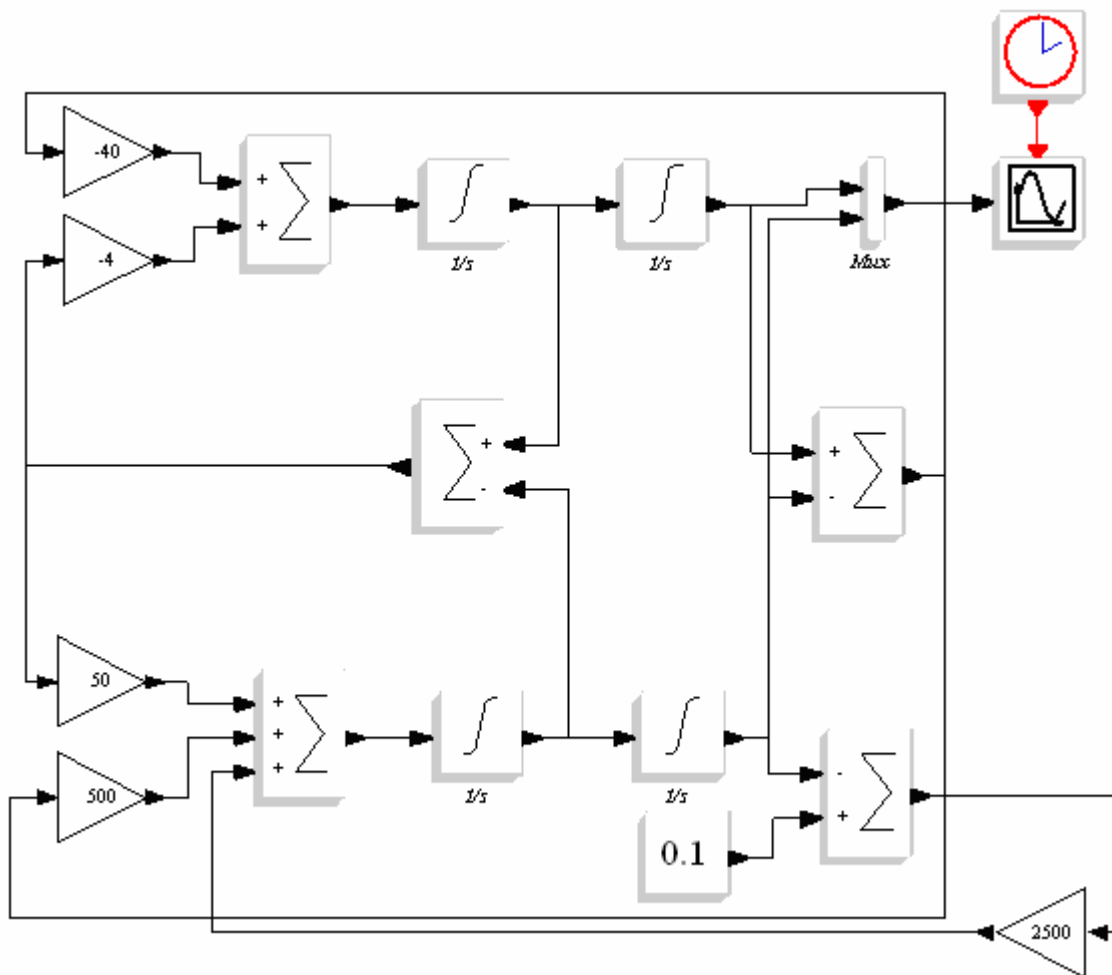
$$\frac{f}{m_2} = 50, \quad \frac{k_1}{m_2} = 500, \quad \frac{k_2}{m_2} = 2500.$$

Podobno kot v prejšnjem primeru začnemo reševati izraze v oklepajih. Izraz  $(\dot{y}_1 - \dot{y}_2)$  rešimo tako, da v simulacijsko shemo vnesemo seštevalnik z dvema vhodoma različnih predznakov in povežemo z vrednostjo  $\dot{y}_1$  in  $\dot{y}_2$ . Podobno naredimo tudi z ostalimi izrazi v oklepajih. Izraze v oklepajih ojačimo in povežemo s seštevalnikom, ki ima tri pozitivne vhode. Izhod  $y_2$  povežemo z multiplekserjem. Ta izhod bo prikazal odziv koles po trku s pločnikom. Shema z označenimi signali je prikazana na sliki 75.



Slika 75: Potek risanja simulacijske sheme za  $\ddot{y}_2$

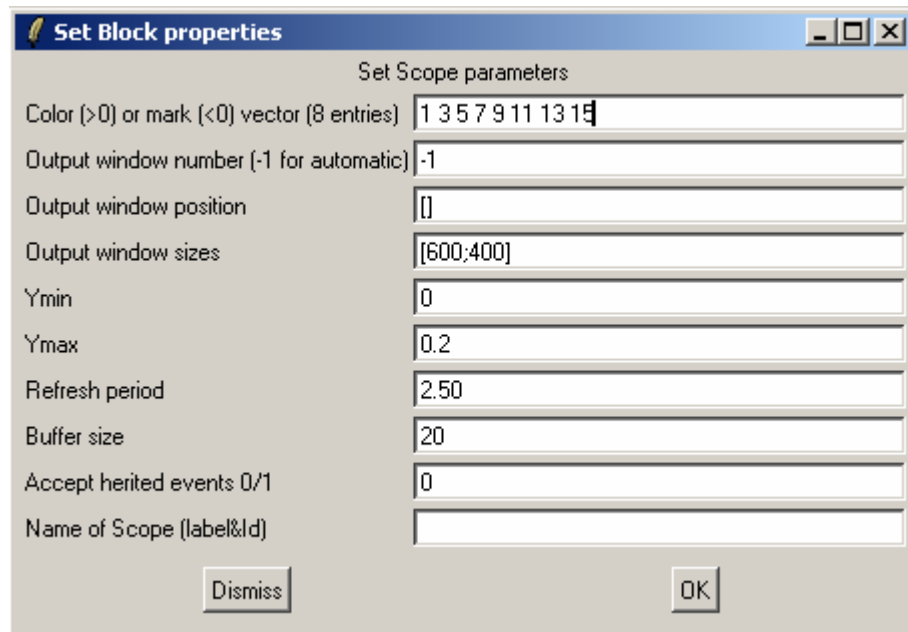
Obe diferencialni enačbi smo povezali preko multiplekserja na prikazovalnik. Celotna simulacijska shema avtomobilskega vzmetenja je prikazana na sliki 76.



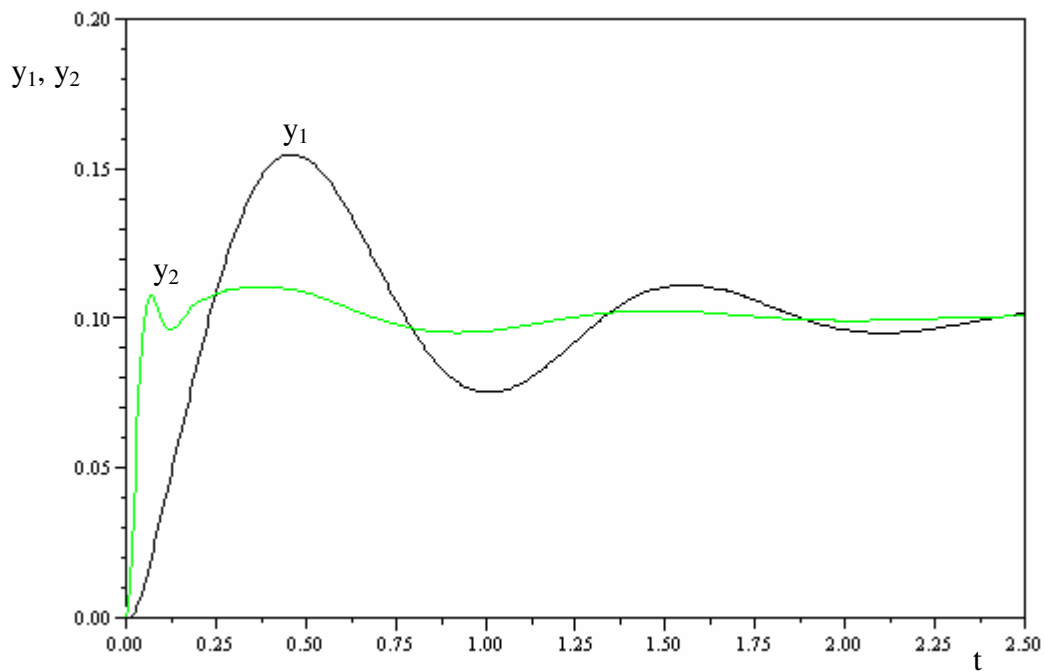
Slika 76: Simulacijska shema modela avtomobilskega vzmetenja

Preden zaženemo simulacijo moramo nastaviti nekatere parametre. Dolžino simulacije nastavimo na 2.5s. Parametre v prikazovalniku nastavimo tako, kot je prikazano na sliki 77. Ker se dogodek zgodi v zelo kratkem časovnem intervalu, je potrebno nastaviti tudi sinhronizacijsko uro sistema. Nastavimo jo na 0.005s.

Poženemo simulacijo in na prikazovalniku vidimo dve krivulji (slika 78), ki ponazarjata odziv kolesa avtomobila in odziv karoserije ob trku s pločnikom višine 0.1m.



Slika 77: Nastavitev prikazovalnika



Slika 78: Odziv karoserije ( $y_1$ ) in kolesa ( $y_2$ ) pri trku s pločnikom



### 7.3 Marketing: Simulacija propagandne akcije

S simulacijo lahko napovemo tudi učinke propagandne akcije. Za primer smo vzeli promocijo novega avtomobila prek različnih medijev, kot so televizija, radio, časopis in revije (Kocijan, 1996).

Upoštevali smo, da se prebivalstvo deli v naslednje tri skupine:

- $x_1(t)$  predstavlja ljudi, ki bi se lahko zanimali za nabavo novega avtomobila,
- $x_2(t)$  predstavlja ljudi, ki kupujejo avtomobile,
- $x_3(t)$  predstavlja ljudi, ki se zaradi različnih vzrokov (npr. zdravja) ne bodo odločili za nakup novega avtomobila.

Stopnja, s katero se večja število potencialnih kupcev, je enaka  $u_1(t)$  in stopnja, s katero se večja število kupcev, je enaka  $u_2(t)$ . Model tega sistema lahko opišemo z naslednjim sistemom enačb (7.8).

$$\begin{aligned}\dot{x}_1(t) &= -mx_1(t) - nx_2(t) + u_1(t) \\ \dot{x}_2(t) &= nx_1(t) - px_2(t) + u_2(t) \\ \dot{x}_3(t) &= mx_1(t) + px_2(t)\end{aligned}\tag{7.8}$$

Vrednost konstant:

$$m = 4, \quad n = 2 \quad \text{in} \quad p = 1$$

Predpostavimo zaprt model prebivalcev ( $u_1(t) = u_2(t) = 0$ ). Naša naloga je, da napovemo, kako uspešna bo prodaja avtomobilov in ali bo propagandna akcija uspešna. Odziv  $x_2(t)$  poiščemo pri naslednjih začetnih pogojih:

$$x_1(0) = 100000, \quad x_2(0) = 20000 \quad \text{in} \quad x_3(0) = 2000$$

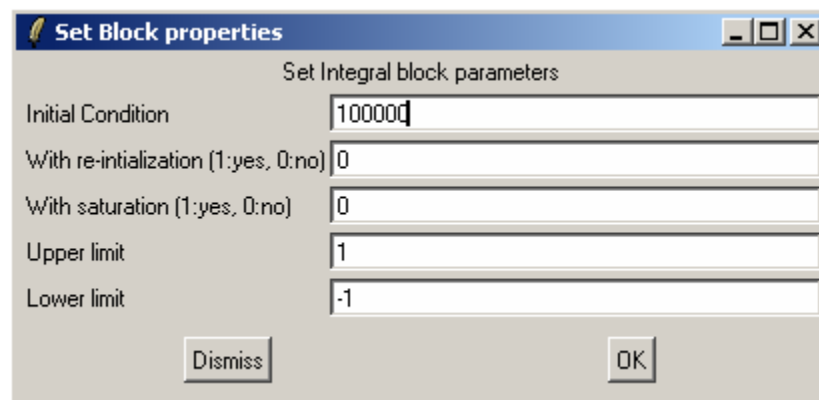
Časovna enota so tedni.

Sistem diferencialnih enačb je razdeljen na tri dele in tako bomo tudi začeli z risanjem simulacijske sheme. Najprej določimo število integratorjev, ki so potrebni za rešitev sistema diferencialnih enačb. Ker imamo tri diferencialne enačbe prvega reda, bomo potrebovali po en integrator za vsako diferencialno enačbo.

Za prikaz vzemimo diferencialno enačbo (7.9):

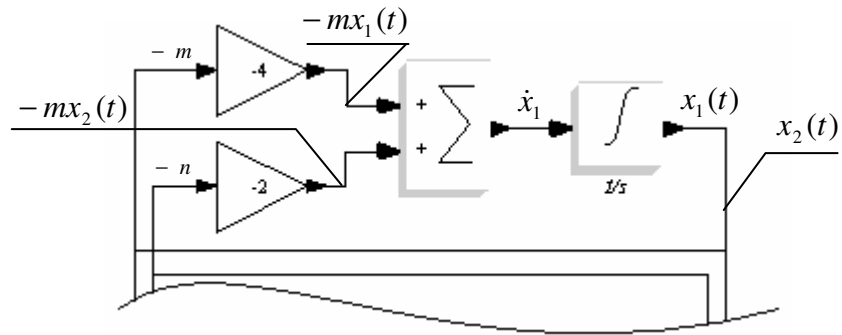
$$\dot{x}_1(t) = -mx_1(t) - nx_2(t) + u_1(t) \quad (7.9)$$

Diferencialna enačba (7.9) je sestavljena iz treh delov:  $-mx_1(t)$ ,  $-nx_2(t)$  in  $u_1(t)$ . V simulacijsko shemo vnesemo dva gradnika za ojačenje. V prvega vpišemo ojačenje  $-m$  in na ojačevalnik pripeljemo signal  $x_1$ , v drugega pa vpišemo ojačenje  $-n$  ter ojačevalnik povežemo s signalom  $x_2$ . Vrednost  $x_1(t)$  dobimo na izhodu integratorja, kateremu podamo začetno vrednost (slika 79), ki znaša 100000. Vrednost  $x_2(t)$  bomo dobili na isti način kot  $x_1(t)$ , le da je to druga diferencialna enačba.



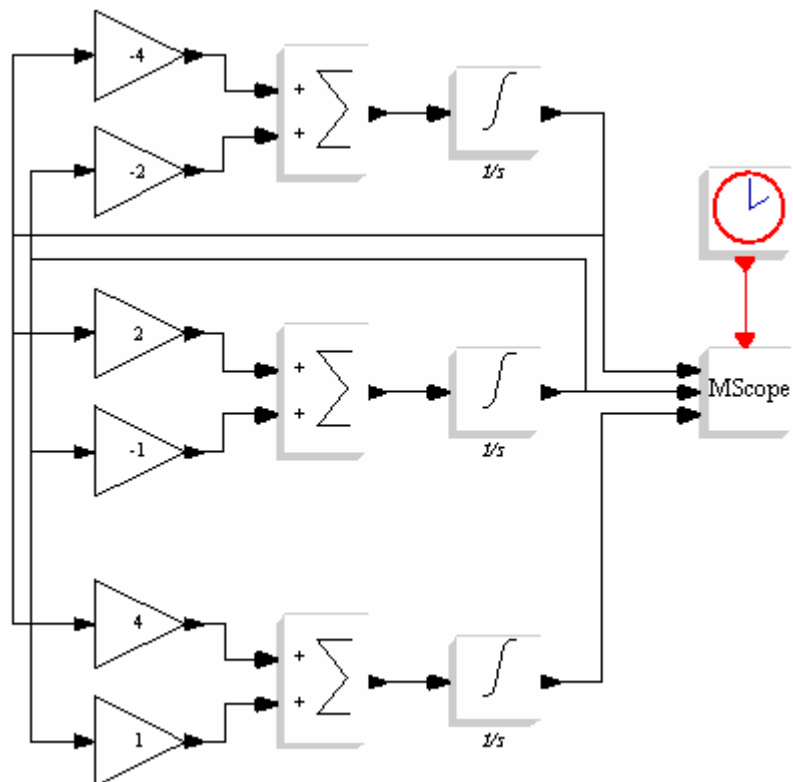
Slika 79: Nastavitev parametrov integratorja

Postopek risanja simulacijske sheme je prikazan na sliki 80.



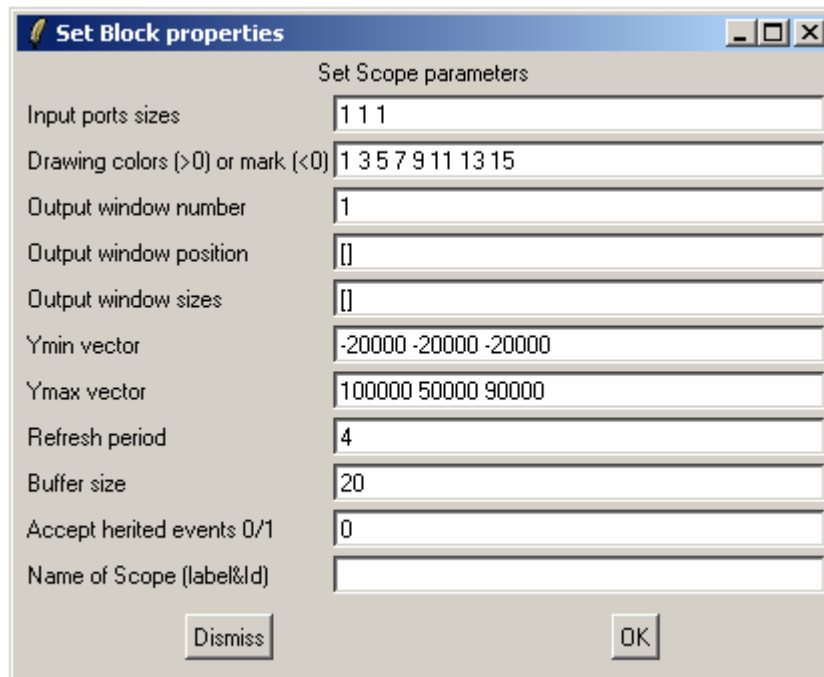
Slika 80: Prikaz risanja simulacijske sheme

Enako naredimo tudi z ostalimi diferencialnimi enačbami. Vse tri izhode priključimo na prikazovalnik Mscope (večkratni prikazovalnik). Poleg do sedaj opisanih načinov prikazovanja signalov, je Mscope (večkratni prikazovalnik) namenjen zajemanju več signalov naenkrat. Dokončna simulacijska shema je prikazana na sliki 81.

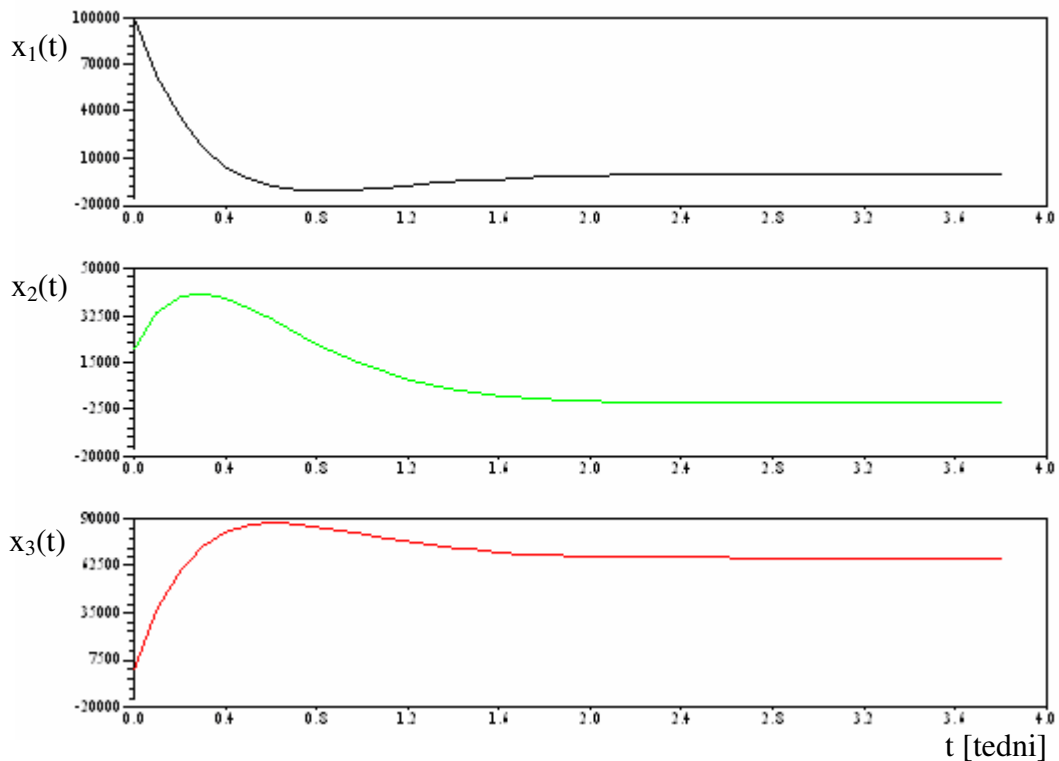


Slika 81: Simulacijska shema propagandne akcije

Preden zaženemo simulacijo, je potrebno nastaviti še parametre prikazovalnika (slika 82). Refresh period (čas osveževanja) nastavimo na 4, kar pomeni 4 tedne. Ymin vector (minimalna vrednost osi Y) in Ymax vector (maksimalna vrednost osi Y) nastavimo različno za vsak graf posebej in vrednosti ločimo s presledkom tako, kot je prikazano na sliki 82. Barve izrisa grafa so določene v okencu Drawing colors (barve izrisa). V našem primeru imamo tri vhode, zato so uporabljena prva tri števila. Rezultat je izrisan na prikazovalniku (slika 83). Rezultat nam pove, kako bo propaganda akcija vplivala na potencialne kupce in kdaj bo dosegla vrhunec pri posamezni ciljni skupini. Vidimo lahko tudi, da po dveh tednih, propagandna akcija nima več vpliva, zato lahko takrat prenehamo s propagando novega avtomobila.



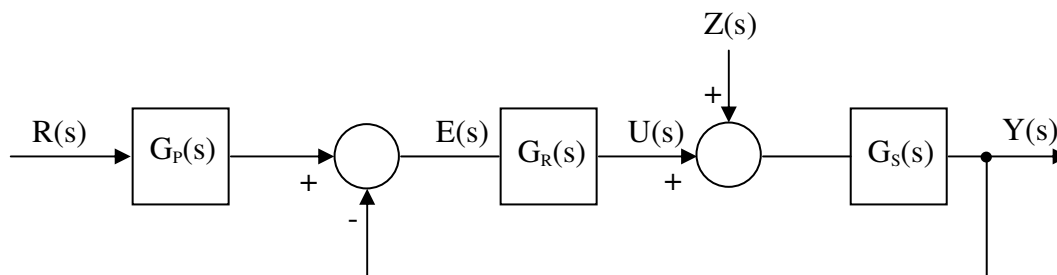
Slika 82: Nastavitev parametrov večkratnega prikazovalnika



Slika 83: Rezultat za vsak segment kupcev posebej, prikazan na večkratnem prikazovalniku

#### 7.4 Avtomatsko vodenje: Simulacija sistema regulacije temperature

Simulirali bomo zaprtozančni sistem vodenja, kot je prikazan na sliki 84 (Kocijan, 1996). Vhod  $R(s)$  je željena temperatura,  $G_S(s)$  je prenosna funkcija sistema,  $G_R(s)$  je prenosna funkcija regulatorja ter  $G_P(s)$  je prenosna funkcija predfiltra.

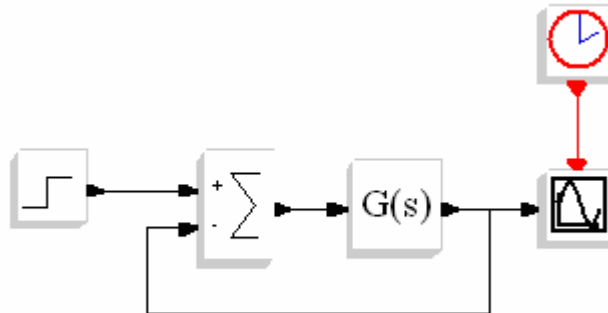


Slika 84: Blokovna shema sistema za regulacijo temperature

Proces opišemo s prenosno funkcijo (7.10):

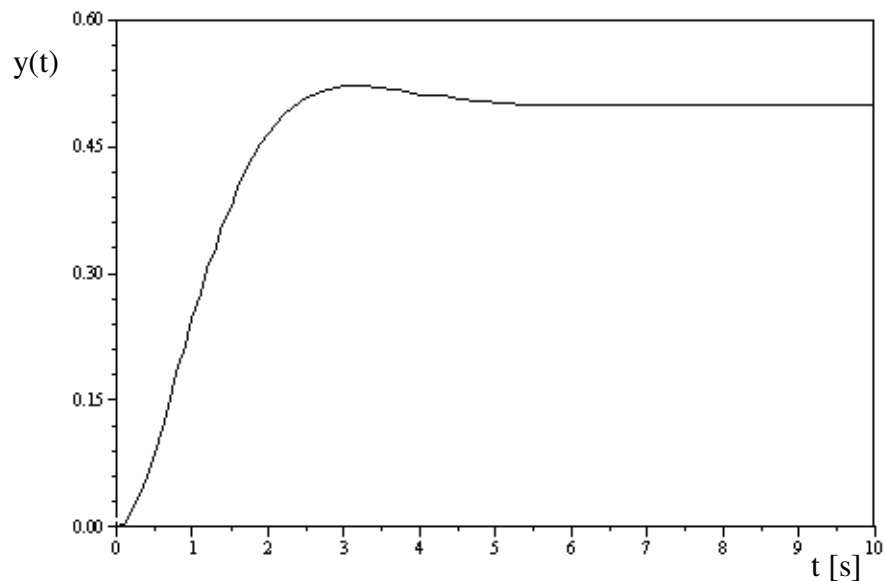
$$G(s) = \frac{1}{(s+1)^2} \quad (7.10)$$

Najprej si pogledjmo, kaj se zgodi, če predpostavimo, da je prenosna funkcija regulatorja  $G_R(s) = 1$ . Simulacijska shema takšnega sistema je prikazana na sliki 85.



Slika 85: Simulacijska shema sistema za temperaturno regulacijo

V simulacijski shemi smo zaradi preglednosti blok Num(s)/Den(s) preimenovali v G(s). Iz odziva na sliki 86 vidimo, da dobimo napako v ustaljenem stanju, ki znaša 50% ustaljene vrednosti in izravnalni čas 3.2 sekundi pri odzivu na enotino stopnico na referenčnem vходу.



Slika 86: Odziv sistema za temperaturno regulacijo na enotino stopnico

Da odpravimo napake zaprtizančnega sistema za regulacijo temperature, uporabimo PID regulator tako, da bo znašal izravnalni čas sistema 0.5 sekunde.

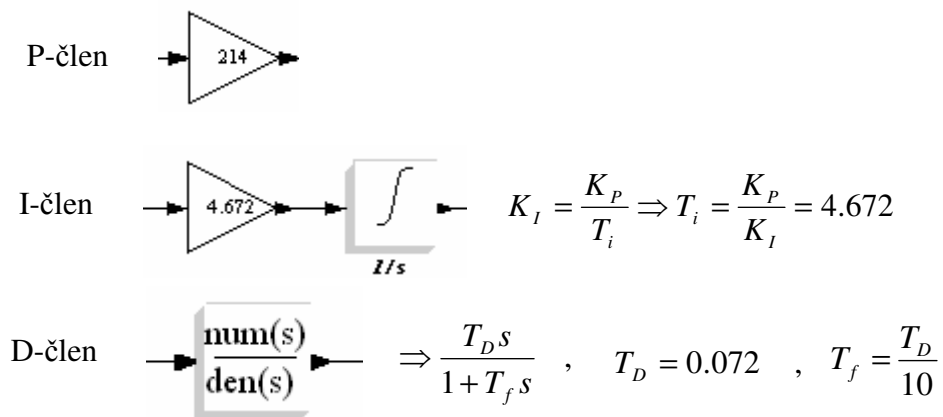
Uporabili bomo PID regulator s prenosno funkcijo:

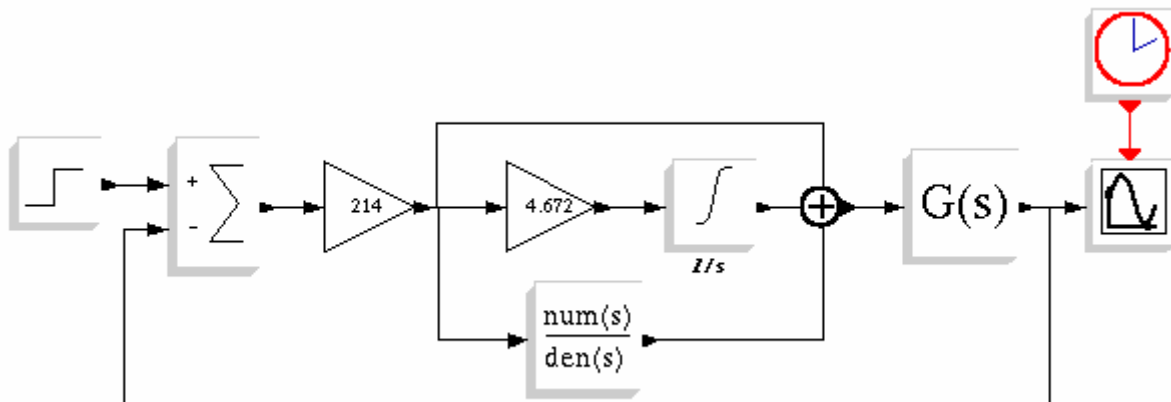
$$G_R(s) = K_p \left( 1 + \frac{1}{T_i s} + \frac{T_D s}{T_f s + 1} \right) \quad (7.11)$$

Parametri:  $K_p = 1000$ ,  $T_i = 4.672$  in  $T_D = 0.72$ . Izračun parametrov se nahaja v knjigi (Kocijan, (1996).

Narišemo simulacijsko shemo s PID regulatorjem, slika 87.

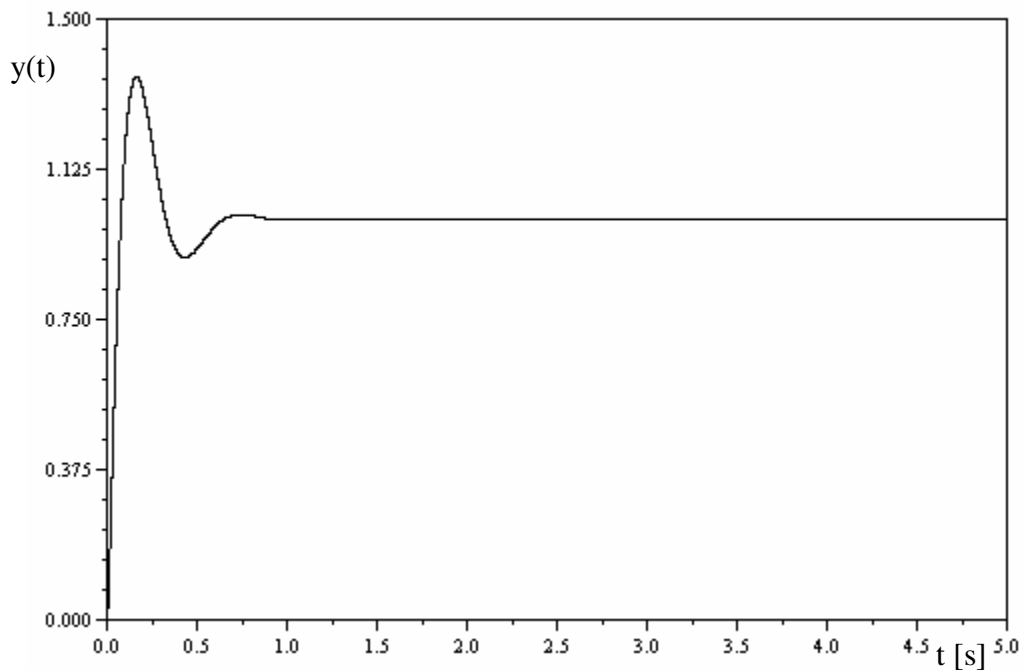
Izvedba P I D členov:





Slika 87: Simulacijska shema sistema za temperaturno regulacijo s PID regulatorjem

Sistem simuliramo z odzivom na stopničast signal velikosti 1 na vhodu in na prikazovalniku (slika 88) opazujemo odziv.



Slika 88: Odziv sistema za temperaturno regulacijo s PID regulatorjem

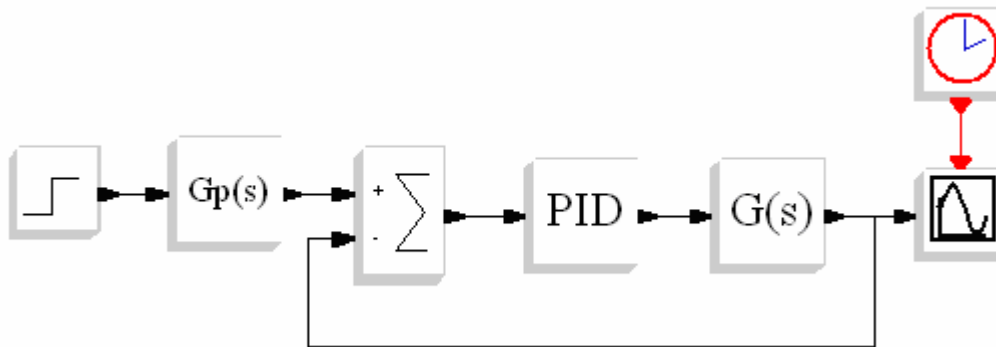
Vidimo, da se je čas izravnave zmanjšal in je približno 0.5 sekunde, obenem pa se je povečal prenehaj, ki je lahko slabost sistema. Zaradi tega bomo v sistem uvedli še predfilter  $G_p(s)$ .



Izbrani predfilter (Kocijan, 1996), s katerim dosežemo želeni odziv, ima prenosno funkcijo (7.12):

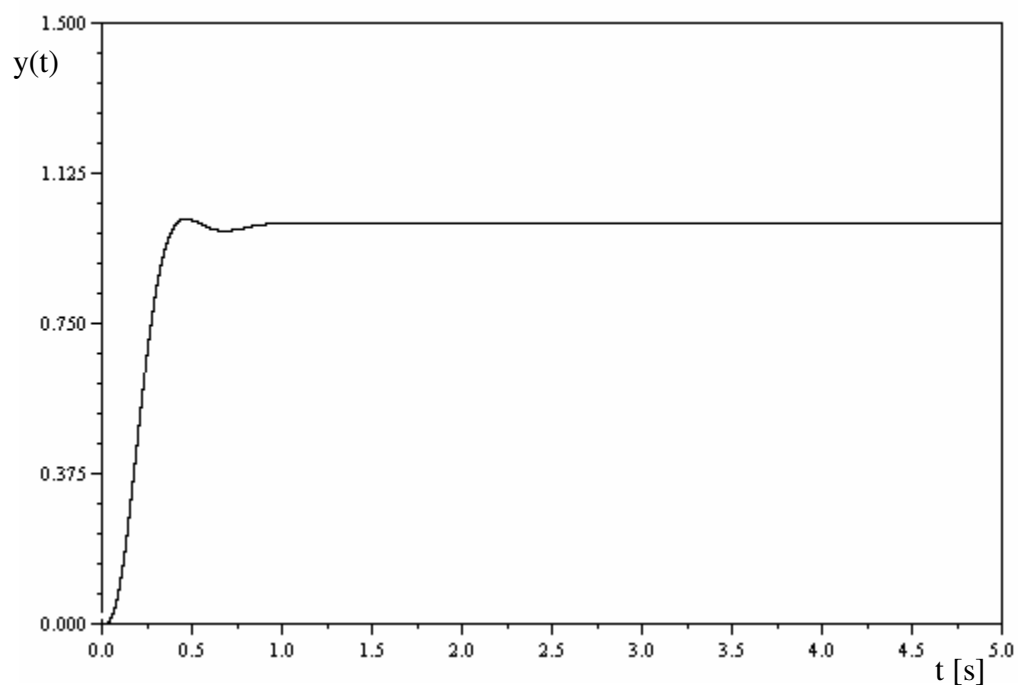
$$G_p(s) = \frac{64.5}{s^2 + 13.8s + 64.5} \quad (7.12)$$

Prenosno funkcijo  $G_p(s)$  vpišemo v blok, ki je priključen na vходу pred regulatorjem. Simulacijska shema je prikazana na sliki 89. Zaradi preglednosti smo PID regulator vstavili v nadblok in prenosno funkcijo predfiltra v blok  $G_p(s)$ .



Slika 89: Simulacijska shema sistema za temperaturno regulacijo s PID regulatorjem in predfiltrom

Na sliki 90 lahko vidimo, da je čas izravnave še vedno 0.5 sekunde in prenihaj se je zmanjšal.



Slika 90: Odziv sistema temperaturne regulacije s PID regulatorjem in predfiltrom

## 8. ZAKLJUČEK

Priročnik za programski modul Scicos smo napisali na način, ki je primeren za uporabnika začetnika. Za začetek smo programski modul Scicos predstavili uporabniku in mu tako nakazali, kakšne so njegove zmogljivosti. Iz poglavja v poglavje smo stopnjevali zahtevnost uporabe od postavitve gradnikov do simulacije različnih vrst dinamičnih sistemov. Vse primere smo simulirali in natančno opisali postopek izdelave simulacijske sheme ter način, kako izvedemo simulacijo.

Naš namen je bil izdelati priročnik, ki bi služil kot pripomoček študentom Poslovno-tehniške fakultete. V ta namen je priročnik grajen funkcionalno in omogoča uporabniku, da osvoji osnove dela s programskim modulom Scicos. Uporabnika vodi skozi postopke simulacije dinamičnih sistemov s slikami in natančnim opisom vseh pomembnejših korakov.

Programski modul Scicos je napredno orodje za simulacijo dinamičnih sistemov, primeren tudi za zahtevne uporabnike. Priročnik je resda sestavljen na nivoju uporabnika začetnika, vendar so podane smernice tudi tistim, ki želijo preiti na višji nivo uporabe. Podana je literatura, ki bo uporabniku v pomoč.

Uporabniku predlagamo, da snov iz priročnika osvaja postopoma in ne preskakuje poglavja, saj se teme dopolnjujejo. Zelo pomembno je primere razumeti in jih preizkusiti v programskem modulu Scicos. Vse primere lahko uporabnik spreminja in preizkuša na različne načine ter tako ugotovi, kakšni so odzivi na spremembo. Simuliranje dinamičnih sistemov je lahko zelo zabavno.



## 9. LITERATURA

**Campbell S. L., Chancelier J.-P., Nikoukhah R.** (2006), Springer, New York, London.

**Kocijan, J.** (1996). Načrtovanje vodenja dinamičnih sistemov, zbirka nalog, ZAFER, Ljubljana.

**Scilab** (2007), Documentation and support , pridobljeno s svetovnega spleta,

[http://www.scilab.org/product/index\\_product.php](http://www.scilab.org/product/index_product.php) (25.1.2007)

**Scilab/Scicos** (2007), Some simple worked examples and uses of the programs, pridobljeno s svetovnega spleta, <http://www.wolffdata.se> (3.2.2007)

**Scilab** (1999), Engineering and scientific computing with Scilab, Birkhäuser