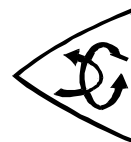




Univerza v Mariboru,
Fakulteta za elektrotehniko, računalništvo in
informatiko



INŠTITUT ZA ROBOTIKO

Izdelava in uporaba S- funkcij

Navodila za delo v orodju Matlab/Simulink

Maribor, avgust, 2007

Avtor: Miran Rodič

Podatki o avtorju

Avtor: dr. Miran Rodič, dipl. inž. el.
Naslov: Inštitut za robotiko
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
Smetanova ulica 17, SI-2000 Maribor, Slovenija, EU
Tel.: (+386 2) 220 7308
Fax: (+386 2) 220 7315
e-mail: miran.rodic@uni-mb.si
www: <http://www.ro.feri.uni-mb.si/~miranro/>

Kazalo

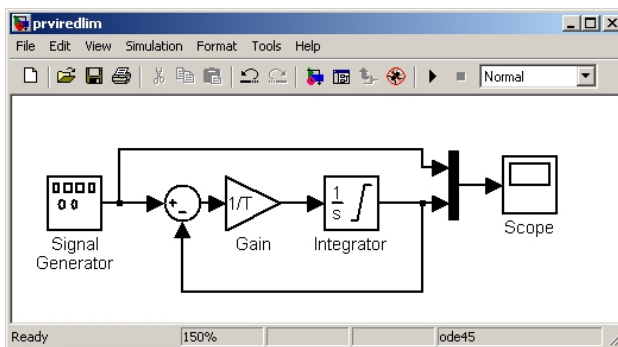
1. Uvod.....	4
2. Zagon orodja MATLAB/Simulink	5
3. Priprava za preizkušanje S-funkcij	6
4. Izdelava nove S-funkcije.....	9
5. Uporaba obstoječih S-funkcij	14
6. Literatura.....	16
7. Varstvo avtorskih pravic	17

1. Uvod

MATLAB je programsko orodje, ki se uporablja za matematično računanje, ki ga povezuje z vizualizacijo in močnim programskim jezikom. Sam proizvajalec ga predstavlja kot programski jezik za tehnično računanje. Ena njegovih najpomembnejših lastnosti je odprta struktura, ki uporabniku omogoča preprosto delo in z uporabo obstoječih funkcij razvoj novih, ki jih sestavi v skladu s svojimi potrebami.

Simulink je razširitev MATLABa, ki ponuja modeliranje, simulacije in analizo dinamičnih sistemov v grafičnem (GUI) okolju. Model sestavimo s pomočjo miške, s katero izberemo ustrezni blok in ga postavimo na željeno mesto. Na voljo so knjižnice elementov, ki omogočajo linearno in nelinearno analizo sistema. Ker je Simulink del MATLABa, je mogoče med uporabniškima vmesnikoma prosto prehajati, kar omogoča uporabniku izkoriščanje prednosti obeh okolij.

Primer preproste analize kaže Slika 1, kjer lahko izvedemo simulacijo časovnega odziva člena 1. reda (npr. RC člen) z omejitvijo na različne vhodne signale.



Slika 1: Člen 1. reda z omejitvijo v Simulinku

Poleg že obstoječih lahko v Simulink dodajamo tudi lastne uporabniške funkcije. Ene od njih so tudi S-funkcije, ki jih lahko programiramo v programskem jeziku C. Način izdelave bo prikazan v tem dokumentu.

Opis je izdelan za *MATLAB 6.5 (R13)* in *Simulink 5.0*. Pri uporabi drugih verzij lahko pride do sprememb.

Poleg predstavljenih so na voljo tudi druge opcije, ki so podrobneje opisane v literaturi.

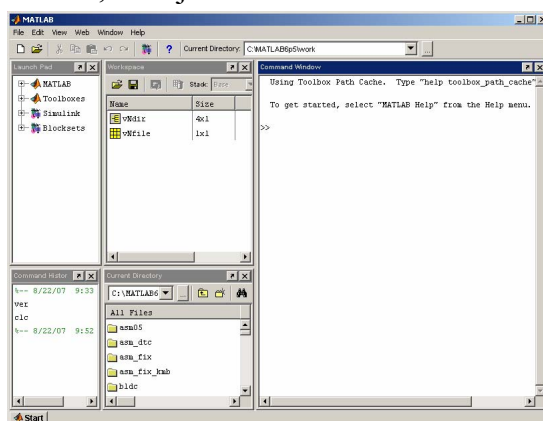
2. Zagon orodja *MATLAB/Simulink*

Program zaženemo tako, da na namizju dvakrat kliknemo na ikono *MATLAB 6.5* (Slika 2).




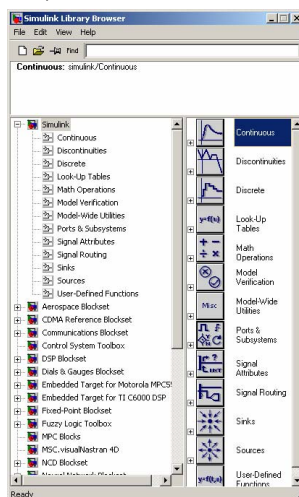
Slika 2: Ikona *MATLAB 6.5*

Zažene se program *MATLAB*, verzija 6.5.




Slika 3: Glavno okno programa *MATLAB*

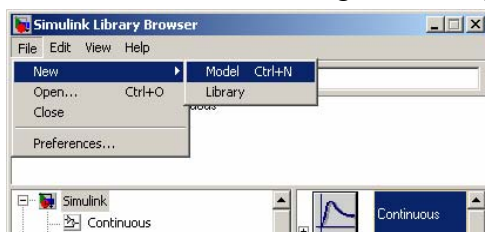
V naslednjem koraku poženemo *Simulink*. To lahko izvedemo tako, da v ukazno okno (*Command Window*) vpišemo ukaz *simulink* ali kliknemo na ikono . Odpre se okno *Simulink Library Browser* (Slika 4).



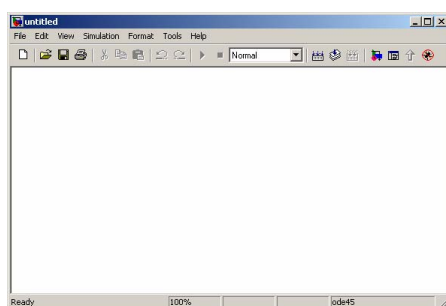
Slika 4: Okno *Simulink Library Browser*

3. Priprava za preizkušanje S-funkcij

Z izbiro **File** → **New** → **Model**, tipkama <Ctrl> in <N> ali klikom na ikono  ustvarimo nov model. Dobimo začetno okno novega modela (Slika 6).

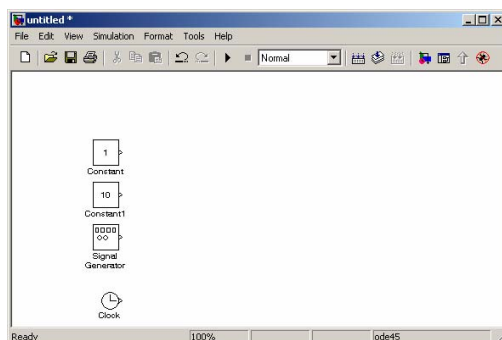


Slika 5: Izbira opcije za ustvarjanje novega modela



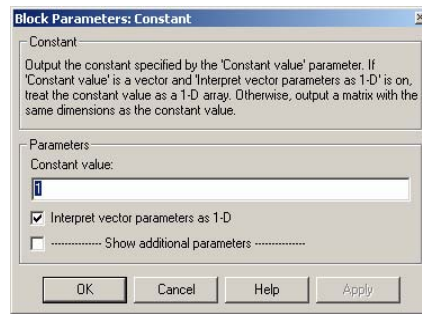
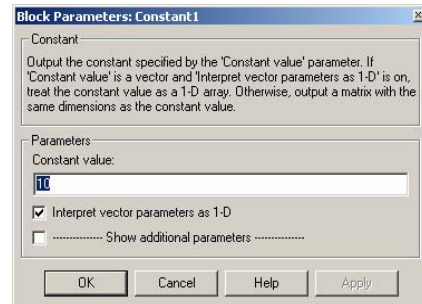
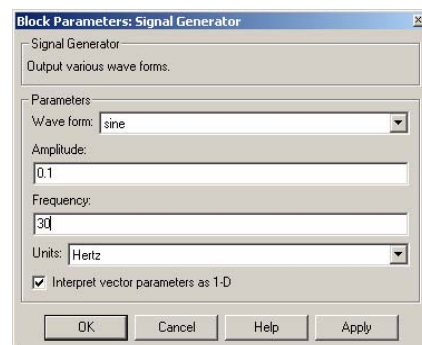
Slika 6: Začetno okno novega modela

Zdaj najprej izdelamo shemo, ki jo bomo uporabili za preizkušanje delovanja. Na začetku vstavimo elemente za vhode. V ta namen iz knjižnice **Simulink** → **Sources** izberemo elemente **Constant**, **Signal Generator** in **Clock**. To izvedemo tako, da jih z uporabo miške preprosto primemo in prenesemo iz okna *Simulink Library Browser* v naše okno (na začetku *untitled*, po shranjevanju s **Save** → **As** s poljubnim imenom). Element **Constant** prenesemo dva krat.

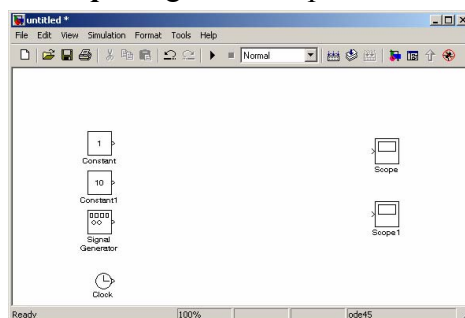


Slika 7: Delovno okno z vnešenimi bloki za vhode

Parametre elementom spremenimo tako, da jih odpremo z dvojnim klikom nanje. Za začetek jih postavimo na vrednosti, ki jih kažejo Slika 8, Slika 9 in Slika 10. Parametrov elementa **Clock** ni potrebno nastavljati.

Slika 8: Parametri bloka *Constant*Slika 9: Parametri bloka *Constant1*Slika 10: Parametri bloka *Signal Generator*

V naslednjem koraku si pripravimo bloka za izris. V ta namen iz knjižnice *Simulink* → *Sinks* izberemo element *Scope* in ga dvakrat prenesemo v naše delovno okno.



Slika 11: Delovno okno z vnešenimi bloki za izhode

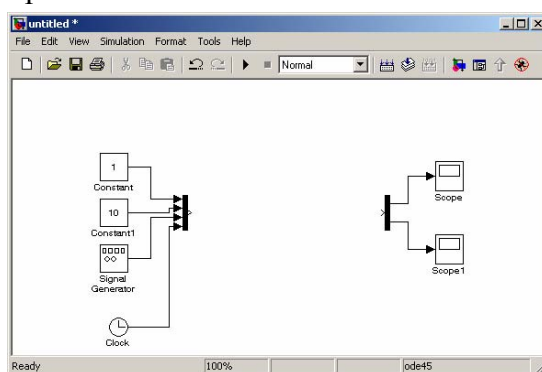
Iz knjižnice *Simulink* → *Signal Routing* prenesemo še bloka za združevanje in razdruževanje (*Mux* in *Demux*). Bloku *Mux* spremenimo parametre, kot to kaže Slika 12.



Slika 12: Parametri bloka *Mux*

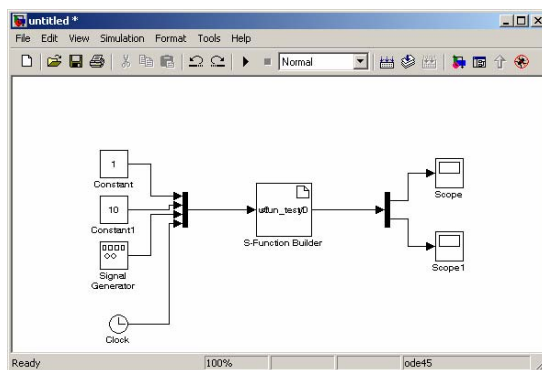
Izvedemo še povezave, kar naredimo tako:

- z miško se postavimo na izhod prvega elementa,
- kliknemo levo tipko na miški in jo držimo,
- potegnemo kurzor do vhoda drugega elementa in
- spustimo levo tipko na miški.



Slika 13: Delovno okno po izvedbi povezav

Zdaj iz knjižnice *Simulink* → *User-Defined Functions* prenesemo blok *S-Function Builder* ter ga povežemo z vhodi in izhodi.

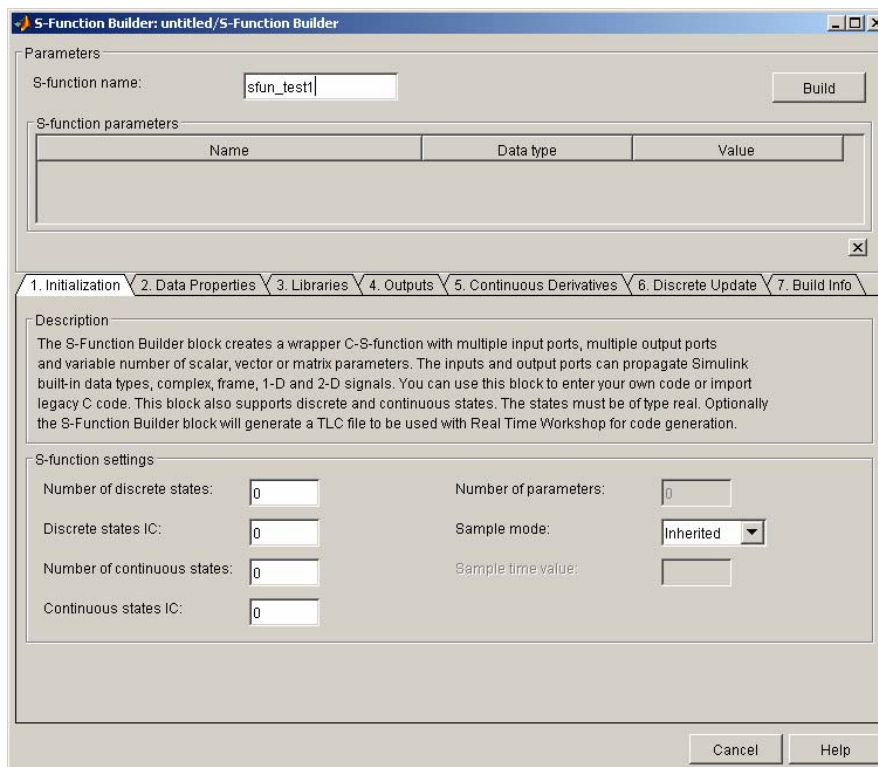


Slika 14: Delovno okno po vstavljanju bloka *S-Function Builder*





Naslednji korak je ustvarjanje nove S-funkcije, ki jo izvedemo s konfiguracijo bloka *S-Function Builder*.

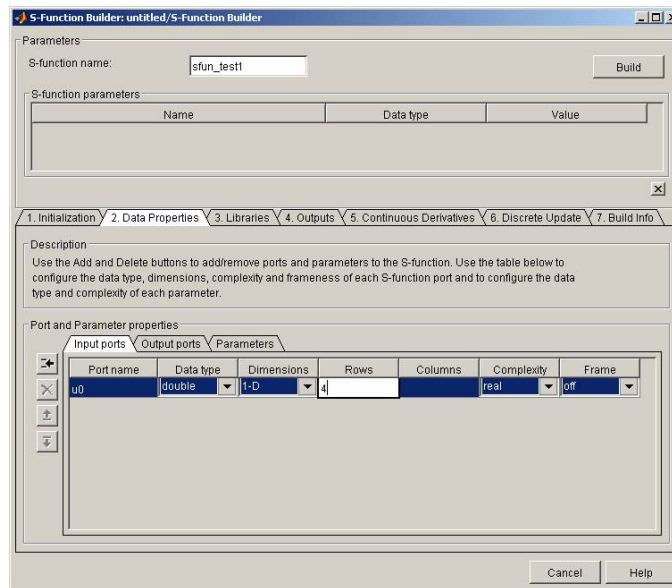
4. Izdelava nove S-funkcije

Konfiguracijo bloka *S-Function Builder* začnemo z dvojnim klikom na blok. Odpre se okno *S-Function Builder:untitled/S-Function Builder* (Slika 15). V okno *S-function name* vstavimo ime funkcije, npr. *sfun_test1*. V spodnjem delu je odprt zavihek *Initialization*, ki mu parametr nastavimo tako, kot to kaže slika.

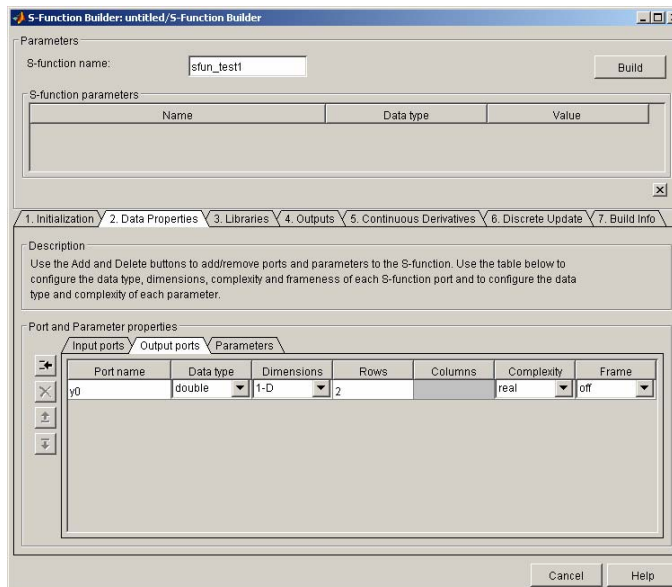


Slika 15: Okno *S-Function Builder:untitled/S-Function Builder* – zavihek *Initialization*

V naslednjem koraku v zavihku *Data Properties* spremenimo vhode in izhode s-funkcije. Za začetek imejmo en vhod in en izhod. Oba sta vektorja, vhodni ima tri elemente, izhodni pa dva. V zavihku *Data Properties* v oknu *Ports and Parameter properties* spremenimo parametra *Rows* elementa *u1* v zavihku *Input ports* (nova vrednost je 4) in *y1* v zavihku *Output ports* (nova vrednost je 2). Nov vhod ali izhod dodamo s klikom na ikono , brišemo z uporabo ikone  ter premikamo navzgor in navzdol z ikonama  in . Okni in njihove vsebine po spremembah kažeta Slika 16 in Slika 17.

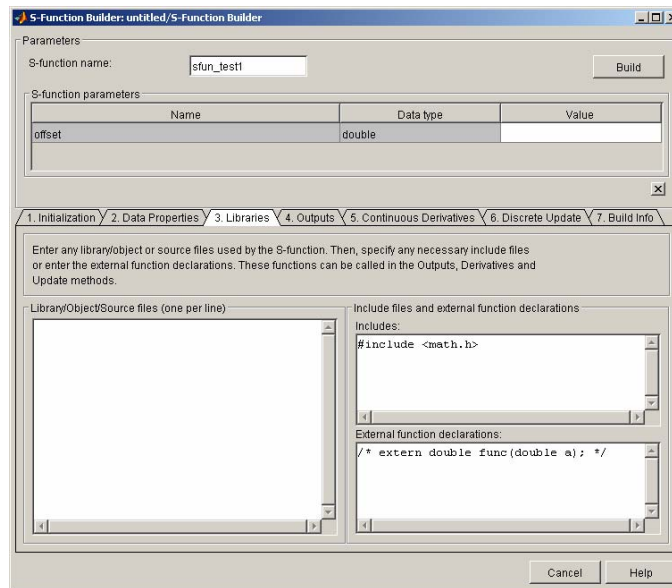


Slika 16: Okno *S-Function Builder:untitled/S-Function Builder* – zavihek *Data Properties*, okno *Ports and Parameter properties* – zavihek *Input ports*



Slika 17: Okno *S-Function Builder:untitled/S-Function Builder* – zavihek *Data Properties*, okno *Ports and Parameter properties* – zavihek *Output ports*

V zavihku **Libraries** vstavljamo potrebna zaglavja (v našem primeru *math.h*), zunanje deklaracije funkcij ter dodatne knjižnice, izvorne in objektne datoteke.



Slika 18: Okno *S-Function Builder:untitled/S-Function Builder* – zavihek *Libraries*
 V zavihek *Outputs* vstavimo naslednjo kodo:

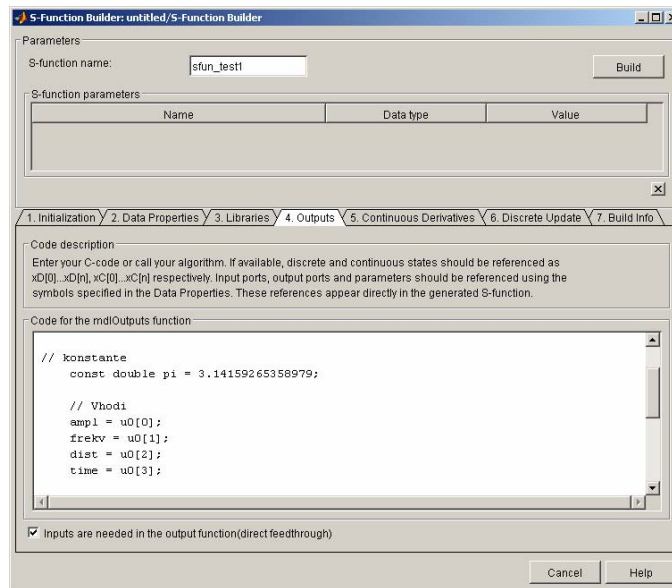
```
// Lokalne spremenljivke
    double ampl, frekv, dist, time;
    double out1;
    double out2;

// konstante
    const double pi = 3.14159265358979;

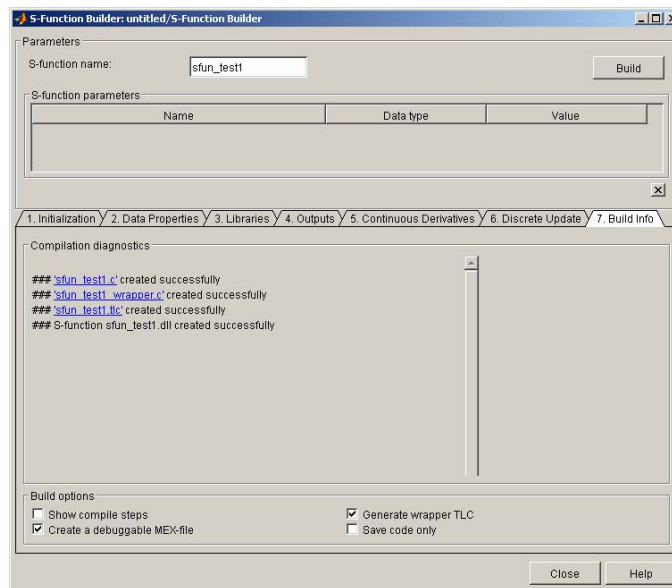
// Vhodi
    ampl = u0[0];
    frekv = u0[1];
    dist = u0[2];
    time = u0[3];

// Funkcija
    out1 = ampl * sin(2.0*pi*frekv*time) + dist;
    out2 = dist;

// Izhodi
    y0[0] = out1;
    y0[1] = out2;
```

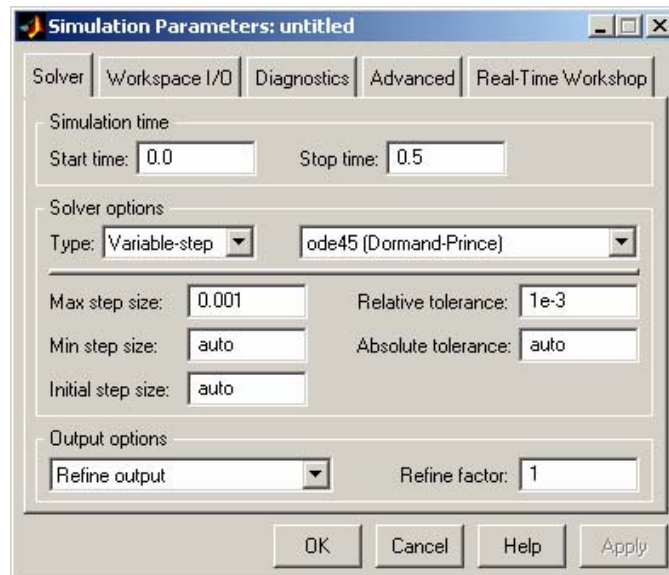


Slika 19: Okno *S-Function Builder:untitled/S-Function Builder* – zavihek *Outputs*
 Zdaj kliknemo na tipko **Build** v zgornjem desnem kotu okna *S-Function Builder:untitled/S-Function Builder* in sprožimo generiranje s-funkcije. Če je bilo prevajanje uspešno, se prikaže okno, ki ga kaže




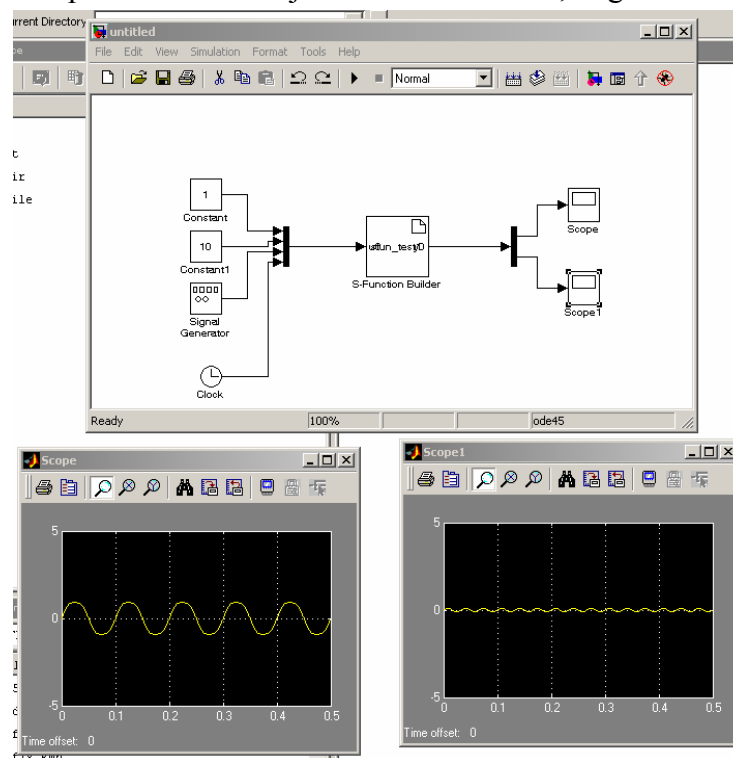
Slika 20: Okno *S-Function Builder:untitled/S-Function Builder* po uspešnem generiranju s-funkcije

Kliknemo na tipko **Close** in se vrnemo v *Simulink*. Z izbiro **Simulation** → **Simulation Parameters ...** nastavimo parametre simulacije. Nastavitve kaže Slika 21.




Slika 21: Nastavitev parametrov simulacije

Kliknemo **OK** in se vrnemo na glavno okno. Za prikaz rezultatov kliknemo na elementa **Scope** in **Scope1**. Z izbiro **Simulation** → **Start**, kombinacijo tipk <Ctrl> in <T> ali ikono  sprožimo simulacijo in dobimo rezultat, ki ga kaže Slika 22.



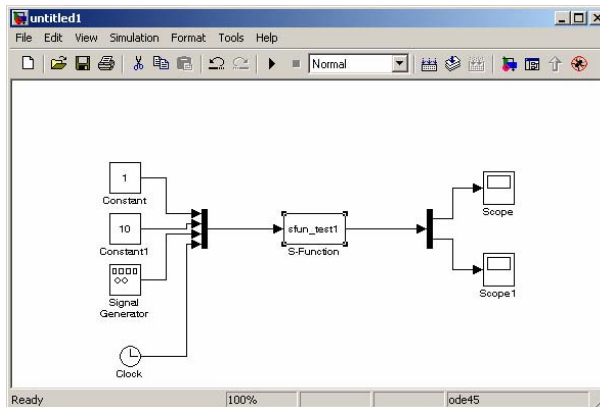
Slika 22: Zaslone izvedbe simulacije

Če želimo prikaz v oknih **Scope** in **Scope1** prilagoditi velikosti okna, kliknemo ikono .

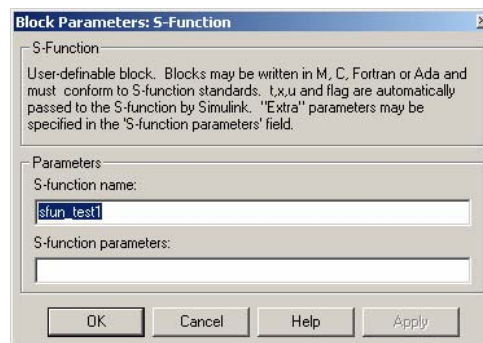
5. Uporaba obstoječih S-funkcij

Če želimo uporabiti že obstoječo s-funkcijo, imamo v knjižnici *Simulink* → *User-Defined Functions* na razpolago blok *S-Function*.

Uporabimo enak okvir kot v primeru ustvarjanja nove s-funkcije (Slika 13). Namesto bloka *S-Function Builder* uporabimo blok *S-Function* (Slika 23), ki mu nastavimo parametre, kot jih kaže Slika 24.

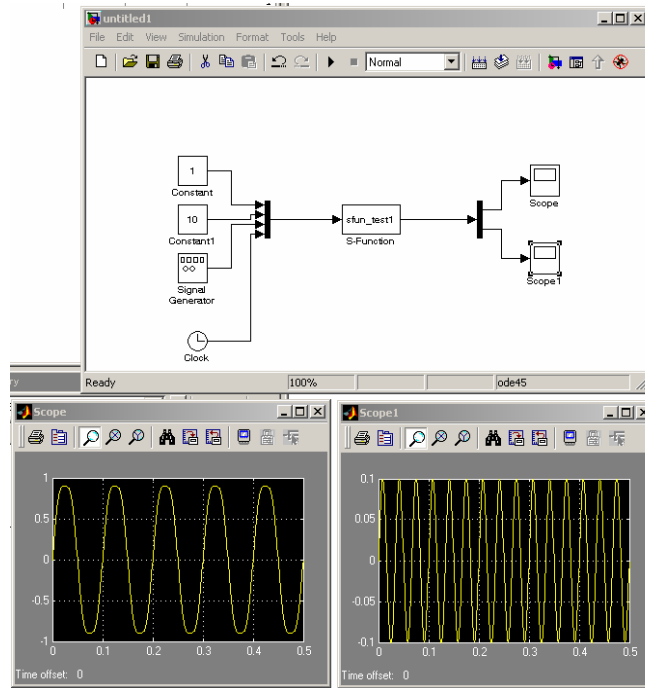


Slika 23: Delovno okno po vstavljanju bloka *S-Function*



Slika 24: Parametri bloka *S-Function*

Spet poženemo simulacijo (postopek je enak kot v prejšnjem poglavju) in dobimo rezultat, kakršnega kaže Slika 25.



Slika 25: Zaslون po izvedbi simulacije z blokom *S-Function*

6. Literatura

- [1] Darko Hercog, Andreja Rojko, *Introduction to DSP-2 control systems*, Februar 2007, http://www.ro.feri.uni-mb.si/predmeti/int_reg/Predavanja/Lab_exercises/taret_intro_to_dsp2_full.pdf
- [2] Andreja Rojko, *Navodila za uporabo programskega paketa Matlab*, http://www.ro.feri.uni-mb.si/predmeti/navodila/Matlab_nav/matlab_65_and.htm
- [3] The MathWorks, Inc.: Real-Time Workshop User's guide (rtw_ug.pdf), verzija 3, januar 1999, <http://www.mathworks.com>.
- [4] The MathWorks, Inc.: Writing S-Functions (sfunctions.pdf), verzija 3, oktober 1998, <http://www.mathworks.com>.
- [5] The MathWorks, Inc.: Using Simulink (sl_using.pdf), verzija 3, oktober 1999, <http://www.mathworks.com>.

7. Varstvo avtorskih pravic

Vse pravice so pridržane. Noben del gradiv ne sme biti reproduciran, spremenjen ali distribuiran brez predhodnega dovoljenja lastnikov avtorskih pravic. Gradivo je last Inštituta za robotiko, Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, in njegovih avtorjev.