

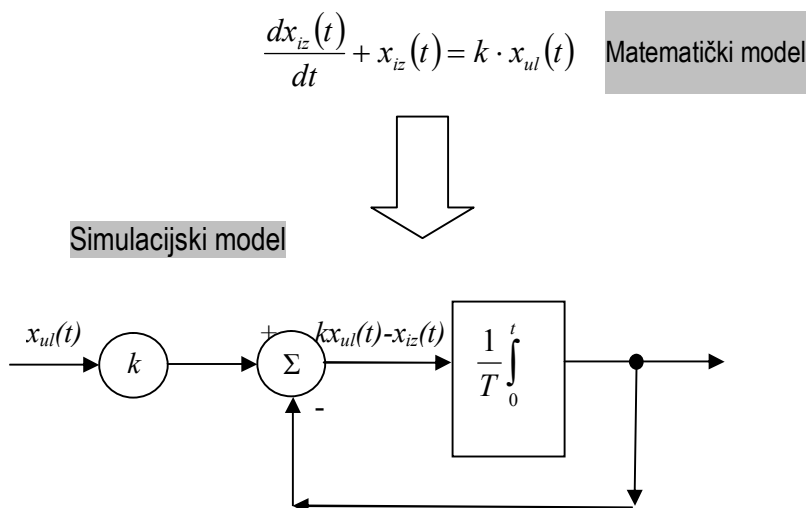
ELEKTROTEHNIČKI FAKULTET OSIJEK		Električni strojevi
Vježba br. 4:	MATLAB – Simulink: Modeliranje i simulacija 1	
Student:		
Grupa:		

4.1. CILJ VJEŽBE

Cilj je ove vježbe upoznavanje studenata s osnovama modeliranja i simuliranja u MATLAB – Simulinku. Na kraju ove vježbe studenti će biti u mogućnosti na jednostavnim primjerima analizirati Simulink modele, grupirati blokove, dokumentirati i pokretati simulaciju, vrednovati rezultate simulacije.

4.2. Što je MATLAB - Simulink ?

Simulink je programski alat razvijen u Matlab grafičkom okruženju koji omogućuje modeliranje, simulaciju i analizu raznih kontinuiranih i diskretnih dinamičkih sustava. Za razliku od Matlab oblik opisivanja sustava nije tekstualan nego grafički. Promatrani sustav razlaže se u pojedinačne blokove koji se zatim povezuje tokovima signala u jednu funkcionalnu cjelinu. U suštini predstavlja numerički rješavač diferencijalni jednačini.



4.3. Kako funkcionira MATLAB – Simulink?

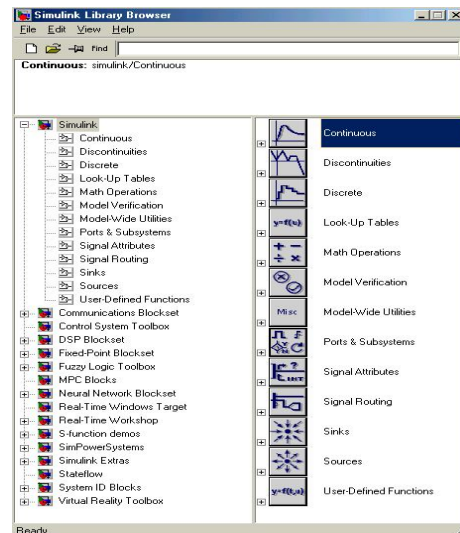
Kod modeliranja u Simulinku koristi se grafičko sučelje te "click – and- drag" operacije mišem za crtanje modela u obliku blok dijagrama. Pri tome simulink sadrži bogatu biblioteku generatora ulaznih pobuda, prikazivača izlaznih varijabli (varijabli stanja), linearnih i nelinearnih komponenti sustava te konektora. Biblioteku možete vidjeti aktiviranjem naloga *simulink*. Simulink biblioteka može se aktivirati i direktno klikom na simbolični nalog *Simulink* smješten u traci s nalogima.

Nakon što je sustav modeliran moguće ga je simulirati na više različitih načina (pomoću simulink menu-a ili pomoću naloga unesenih u komadni prostor Matlab). Korištenjem različitih blokova za prikaz varijabli stanja, mogu se vidjeti varijable sustava i za vrijeme trajanja simulacije. Dana je mogućnost mijenjanja parametara simulacije kao i simultano praćenje rezultata simulacije. Rezultati simulacije mogu se spremiti u Matlab *workspace* ili u *.m fil-ove s ciljem naknadne obrade i vizualizacije.

Dakle, simulacijski model izvodi se u sljedećim etapama:

- Definicija problema
- Postavljanje matematičkog modela
- Razvoj simulacijskog programa
- Test simulacijskog modela
- Izvođenje simulacijskog postupka
- Vrednovanje rezultata

Kod definicije problema bitna je odluka želi li se promatrati točno ponašanje jednog motora ili funkcionalnost cjelokupnog sustava. Oba cilja rezultiraju s potpuno različitim modelskim strukturama.



4.4. Postupak konstruiranja jednostavnog simulacijskog blok dijagrama

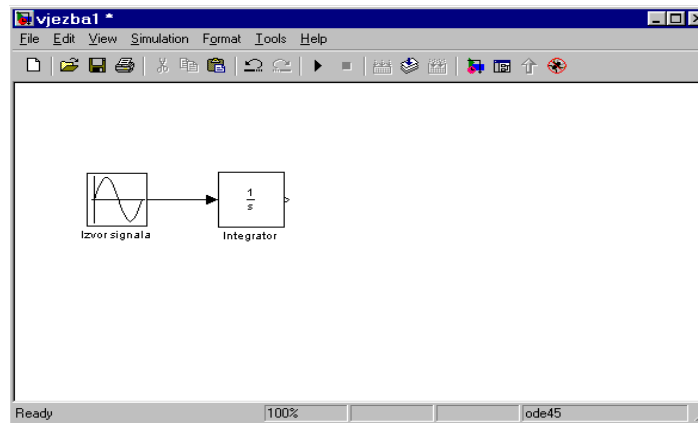
Želimo li pomoću blokovske biblioteke konstruirati vlastiti simulacijski sustav, moramo najprije otvoriti Simulink prozor nalogom *File/New/Model Ctrl+N* u traci s nalogima. Dio navedenog naloga govori da se Simulink prozor može otvoriti i kombinacijom tipki *Ctrl+N* u tipkovnici računala. Ne smijemo zaboraviti, da je kod mnogo otvorenih prozora aktivan samo onaj čiji je naziv zatamnjen.

Preporučljivo je otvorenom prozoru odmah pridodati odgovarajući naziv. Načinimo to nalogom *File/Save as...* i pohranimo ga pod nazivom *vjezba1.mdl*. Nastavak *.mdl* dolazi od «model» i specifičan je za Simulink.

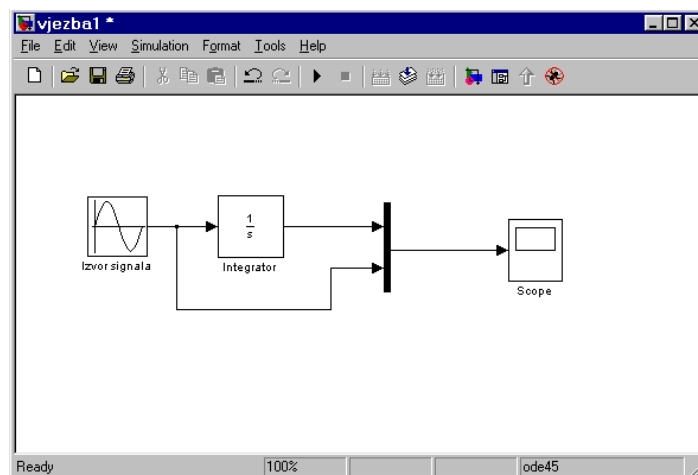
Otvorimo blok Sources, pronađimo u njegovom sadržaju blok Sine Wave, prevucimo ga klikom miša u prozor Simulinka i tamo pozicionirajmo kao na donjoj slici. Nismo li zadovoljni njegovim nazivom, kliknimo na njega i promijenimo ga u Izvor signala. Čini li nam se blok premalen kliknimo na njega i povećajmo ga na jednom od njegovih uglova.

Slijedećim blokom želimo dobiveni signal integrirati. Otvorimo u tu svrhu osnovni blok Continuous i u njemu blok Integrator, te i njega prevucimo u prozor Simulinka i nešto povećajmo, kao i prethodni. Spojimo zatim pomoću miša izlaz prvoga bloka s ulazom drugoga. Tu je potrebno imati nešto više strpljenja, a izgleda da je to lakše provesti ako se spajanje obavi u smjeru suprotnom od smjera toka signala. U ovom slučaju kliknimo mišem na ulaz Integratora, vucimo crtu prema izlazu izvora signala i tu ju pustimo. Ostvarili smo željeni spoj i dobili prikaz kao na donjoj slici.

Oznaka $1/s$ u bloku integratora ukazuje da se tu radi o integraciji u Laplaceovom području. Više je blokova koji nose oznake Laplaceove transformacije.



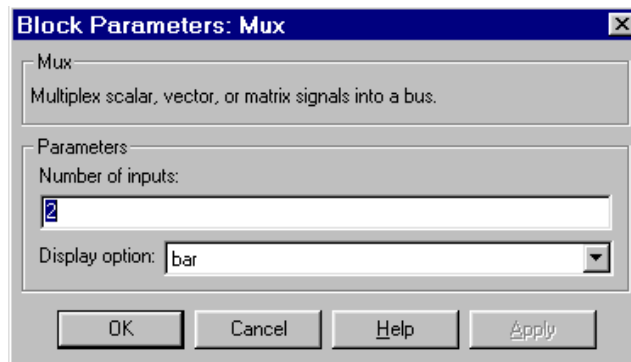
Proširimo naš prikaz prozorom u kojem ćemo moći pogledati tokove sinusnog signala i njegove integracije. Izaberimo u biblioteci blok Signals&Systems i u njemu blok Mux te ga prevucimo i pozicionirajmo u prozor Simulinka, a zatim malo produljimo kao u doljnoj slici. Blok Mux je višepolni prekidač ili multiplekser, čija je zadaća povezivanje više signala u jedan vektorski signal. Izaberimo zatim u biblioteci Sinks blok Scope i isto tako ga prevucimo u prozor Simulinka, pozicionirajmo i malo povećajmo. Spojimo sada izvorni signal, klikom na spojnu crtu obaju prvih blokova, kao i izlaz integratora s dva ulaza multipleksera, a izlaz multipleksera s ulazom bloka Scope. Kod prvog spajanja moramo zaobići blok Integrator. To ćemo učiniti tako da ćemo pri svakoj promjeni smjera kratko pustiti tipku miša i zatim nastaviti s povlačenjem crte. Spoj s jednom postojećom vezom, kao u ovom drugom slučaju biti će korektan tek onda kad se na mjestu spoja pojavi mala pravokutna točka. Naš prikaz sad ima izgled kao na donjoj slici.



Prije no što s ovim blokovskim prikazom izvedemo prvu simulaciju, moramo načiniti još nekoliko radnih koraka. Kao prvo u primjenjenim blokovima moramo podesiti njihove parametre, jer kako će primjerice Simulink znati koju frekvenciju treba imati sinusni signal. A drugo, moramo utvrditi i parametre simulacije, primjerice vrijeme trajanja, postupak numeričkog rješavanja itd. Uz to su nam potrebni još i neki opisi u prozoru Simulinka, jer želimo našu simulaciju i dokumentirati.

4.5. Parametriranje blokova

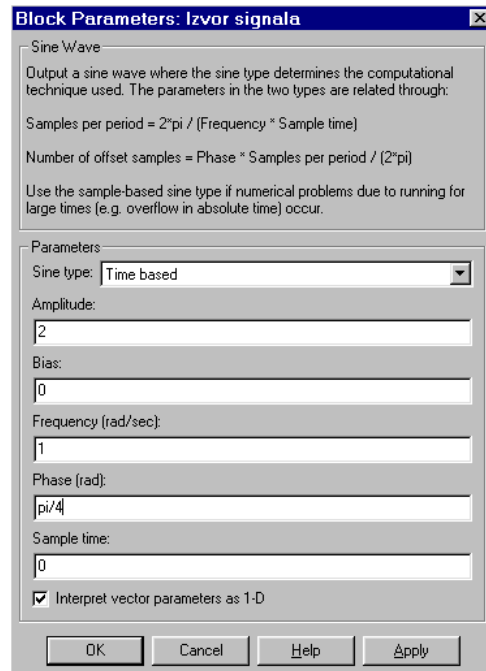
Primjenjeni blok Mux ima predpodešena dva ulaza, što u našem primjeru slučajno i odgovara. Broj se ulaza međutim može i promijeniti. Dvostrukim klikom na odgovarajući blok otvara se njegova lista s parametrima, a u primjeru bloka Mux ona ima izgled kao u donjoj slici. U njoj se broj ulaza zadaje u prozorčiću *Number of inputs*.



Parametar *Display option* omogućava promjenu grafičkog prikaza multipleksera. Pretpodešen je prikaz *bar*, koji bloku daje izgled jedne debele okomite crte i koji je pogodan za slučaj velikog broja ulaza. Ako su potrebna samo dva ulaza, kao u našem slučaju, pogodniji je prikaz *signals*, jer se malim povećanjem ovog bloka pojavljuju i nazivi signala (ovdje *signal1* i *signal2*). Time se kod većih simulacijskih sustava postiže veća preglednost tokova signala.

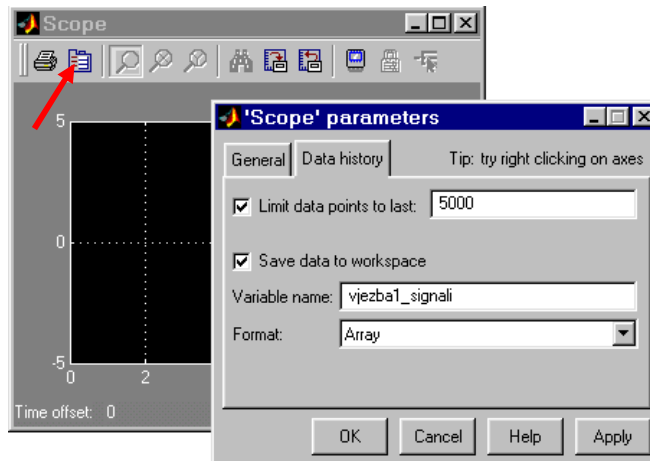
Sinusni se signal parametrira na sličan način. Dvostrukim klikom na blok Izvor signala otvara se prikaz kao u bočnoj slici. Tu se može podesiti tip sinusnog signala, amplituda, frekvencija (u rad/s), faza itd. Parametar *Sine type* i *Bias* ostavimo na pretpodešenjima. Parametar *Sample Time* ostavimo također na pretpodešenju 0, jer želimo izvesti vremenski kontinuiranu simulaciju. Promijenit ćemo ga onda ako želimo eksplicitno izvesti vremenski diskretnu simulaciju, unosom vremena otipkavanja.

Za naš slučaj promijenimo parametar amplitude na 2, a parametar frekvencije ostavimo na 1 rad/sec, što odgovara $1/2\pi=0.16$ Hz (unos 2π odgovarao bi 1 Hz). Simulink isto kao i MATLAB prepoznaje simbol pi i time egzaktan vrijednost ove konstante. Za fazni pomak izaberimo $\pi/4$.



Blok Integrator zahtijeva između ostalog i početnu vrijednost integracije (parametar *Initial Condition*). Ostavimo tu predpodešenu vrijednost 0. Ostali se parametri odnose na specijalne oblike Integratora, pa ih također ostavimo na pretpodešenim vrijednostima.

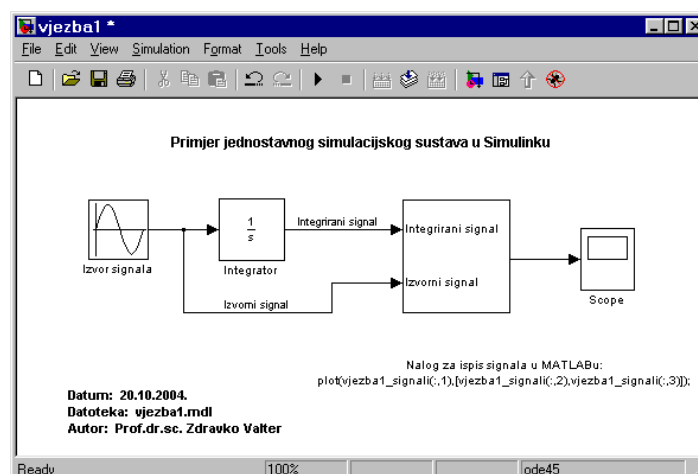
Dvostrukim klikom na blok Scope otvara se prozor kao na slici dolje, a klikom na simbol *Parameters*, lijevo do simbola štampača u tom prozoru, i prozor s parametrima. Puno se toga tu da podesiti, no pređimo u karticu *Data history*, kao na slici, i pohranimo prikazane signale u *Workspace* MATLAB-a. Upišimo naziv varijable u polje *Variable name* kao u slici i u polje *Format* ponudu *Array* koja nije pretpodešena. Želimo li naime preći u MATLAB, moramo uzeti matrični format koji odgovara varijabli MATLAB-a.



Nakon izvedene simulacije i pojave signala u bloku Scope, njihov se prikaz u gornjem prozoru može još podosta izmijeniti.

Posvetimo se na kraju spomenutim opisima za dokumentiranje simulacije. Dvostrukim klikom na slobodni dio radne plohe Simulinka pojavljuje se žmigavi kursor. Njim se može upisati odabrani tekst i po želji doraditi preko naloga *Format/Font...* u traci s nalozima. Tako se primjerice može upisati naziv simulacije s datumom i slično. Korisno je i opisati tokove signala. Dvostrukim klikom na strelicu, signalima se može pridodati naziv. Ovaj se naziv čak pomiče, ako se pomiče strelica ili pripadajući blok. Uz to se automatski promjene i nazivi ulaza blokova, primjerice bloka Mux. U tu se svrhu dotični blok mora ili malo povećati, ili mu se mora promijeniti položaj.

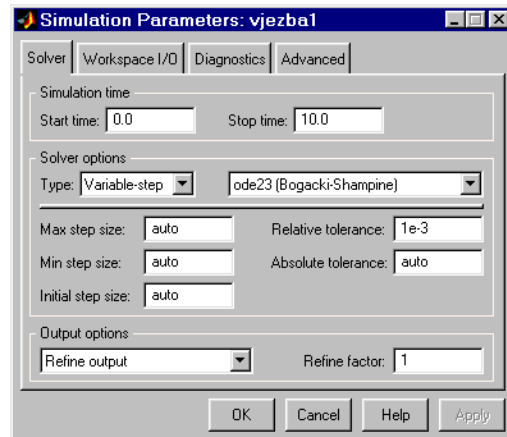
Korisno je upisati i nalog kojim se može pozvati pripadajući dijagramski prikaz u MATLAB-u, radi kasnije analize. Iduća slika prikazuje izabrani način opisa blokovskog prikaza *vjezba1.mdl*. Time je završeno parametriranje blokova, pa se sve može zaokružiti pohranjivanjem preko naloga *File/Save*.



4.6. Parametriranje i izvođenje simulacije

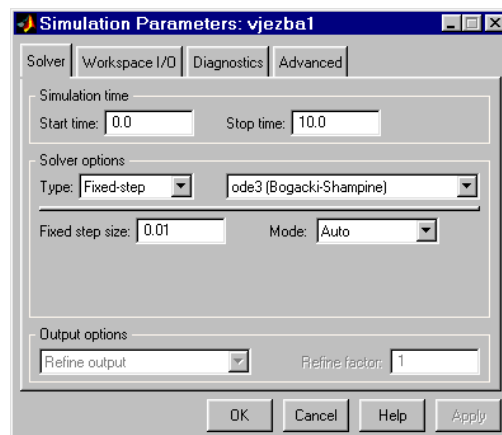
Simulacijski parametri podešavaju se nalogom *Simulation/Simulation parameters...* u traci s nalozima. Aktiviranjem naloga otvara se prozor za dijalog kao u slici dolje.

Parametri *Start time* i *Stop time* su sami po sebi jasni. Pogledajmo stoga oba parametra uz naziv *Solver options*. Pod nazivom *Solver* podrazumijeva se numerički postupak rješavanja diferencijalnih jednadžbi, koji se želi primjeniti u ovoj simulaciji. U obzir dolaze postupci upravljanja sa i bez promjenljive duljine koraka, a utvrđuju se izborom *Variable-step* ili *Fixed-step* u prvom otvoru. Ovi postupci počivaju na Runge-Kutta metodi i kriju se pod oznakama *ode3*, *ode23*, *ode45* itd.



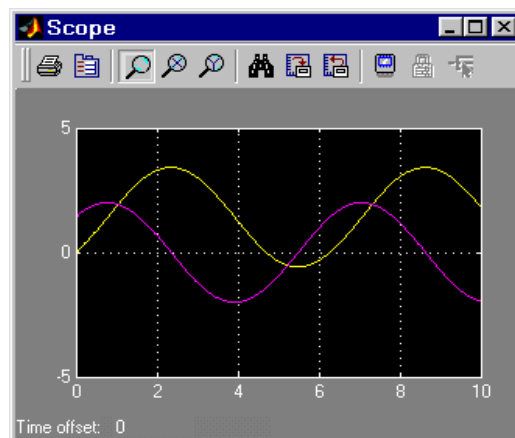
Izborom naloga *Variable-step*, kao u prozorčiću na slici, izrazili bismo želju da se duljina koraka numeričkog postupka rješavanja diferencijalne jednadžbe mijenja s dinamičkim ponašanjem rješenja. Mijenja li se rješenje vrlo malo, duljina koraka bi se povećavala, a mijenja li se puno duljina koraka bi se automatski smanjivala. Koliko se duljina koraka smije mijenjati utvrđuje se zadavanjem vrijednosti u donjih pet otvora.

Izabrom nalog *Fixed-step*, smanjio bi se opseg unosa predviđenih parametara. Tako su primjerice unosi u donja dva otvora pod nazivom *Output options* vezani samo uz izbor *Variable-step*, a služe za dodatnu interpolaciju izabranih točaka koje treba matematički obraditi kako bi se dobilo još glađe grafičko rješenje.



Ostavimo za sada ova složena objašnjenja i orijentirajmo se na nalog *Fixed-step*, na postupak *ode3* i podesimo parametar *Fixed step size* na 0.01, kao u donjoj slici.

Kako je time sve spremno za izvođenje simulacije, možemo ju pokrenuti nalogom *Simulation/Start* u traci s nalozima. Druga je mogućnost kliknuti na simbol trokuta u traci sa simbolima. Rezultat simulacije, prikazan u bloku *Scope*, otvara se dvostrukim klikom na taj blok i izgleda kao na donjoj slici. Vide se obje krivulje, i izvorni signal i njegov integral.



Budući da smo signale u bloku Scope pohranili u MATLAB-u pod nazivom *vjezba1_signali*, gornji dijagram možemo pogledati i u MATLAB-u. Pogledajmo prije toga sadržaj *Workspace*.

» whos

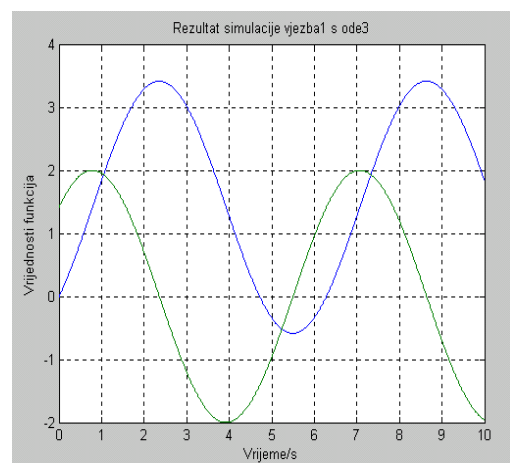
Name	Size	Bytes	Class
tout	1000x1	8000	double array
vjezba1_signali	1001x3	24024	double array

Grand total is 4003 elements using 32024 bytes

U *Workspace* su smještena tri vektora u stupcima, a u prvom stupcu je vremenski vektor. A sad pogledajmo dijagram unosom slijedećih naloga:

```
» plot(vjezba1_signali(:,1),[vjezba1_signali(:,2),  
    vjezba1_signali(:,3)])  
» grid  
» title('Rezultat simulacije vjezba1 s ode3')  
» xlabel('Vrijeme/s')  
» ylabel('Vrijednosti funkcija')
```

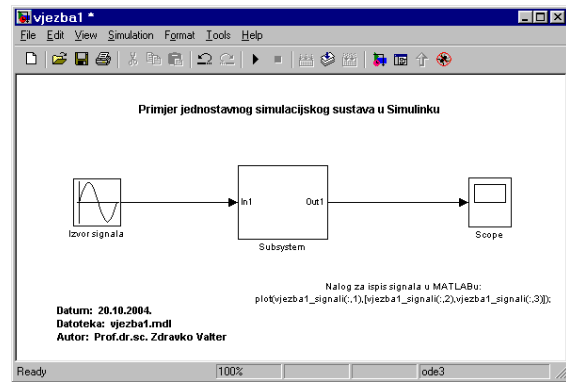
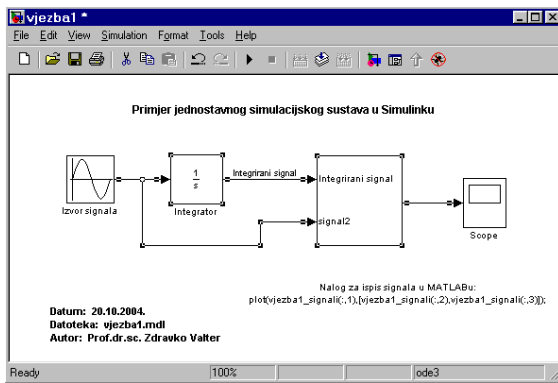
Dijagram ima izgled kao na bočnoj slici i da se usporediti s onim u Simulinku..



Simulink sustavi mogu sadržavati puno blokova i biti nepregledni. Za njihovo pojednostavljenje nudi se mogućnost grupiranja nekoliko blokova, koji čine jedan podsustav, u jedan jedini blok. Takvo modulariziranje složenijih sustava je često puta i neizbježno.

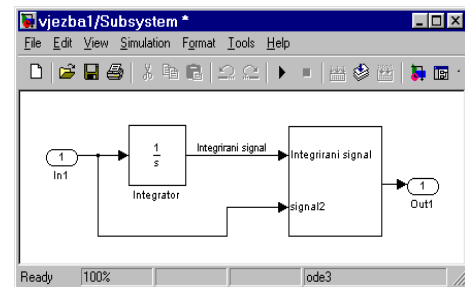
Prvu mogućnost grupiranja nudi primjena tzv. Fcn bloka. Ona omogućuje sprezanje cijelih jednadžbi u jednu cjelinu, tako da se ne moraju unositi pojedinačni koraci ili pojedinačni blokovi niže razine već se u njegov prozor za parametriranje upisuje ili cjelokupna jednadžba, ili naziv *m-datoteke* s jednadžbama pisanim u MATLAB-u. Blok Fcn nalazi se u grupi blokova User-Defined Functions, a spomenuta jednadžba ili naziv upisuje se u prozoru za parametriranje pod *Expression*. Oznaka *u* u toj jednadžbi uvijek označuje ulazni signal kojeg jednadžbom treba obraditi, bez obzira kako on bio označen u promatranom sustavu.

Kod složenijih slučajeva rješenje je u vlastitom definiranju blokova koji se daju povezati u jednu cjelinu. Probajmo u svrhu vježbe pojednostavniti sustav prikazan na stranici 15. Grupirajmo dva bloka pozicionirana u sredini. Uokvirimo mišem te blokove i njihove signalne linije ili prijelazom miša preko njih, ili držanjem tipke *Shift* u tipkovnici te klikovima na pojedinačne elemente u blokovskom prikazu koje želimo međusobno povezati. Dobili smo prikaz kao na slici dolje lijevo. Elementi koji se trebaju povezati označeni su malim kvadratićima.



Kad smo ih odabrali, kliknimo na nalog *Edit/Create Subsystem* u traci s nalozima. Pojavio se prikaz kao na gornjoj slici desno. Dobili smo jedan podsustav s nazivom *Subsystem* koji ima jedan ulaz i jedan izlaz. Naš je blokovski prikaz postao jednostavniji.

Dvostrukim klikom na blok podsustava *Subsystem* otvara se novi prozor u kojem možemo vidjeti njegov sadržaj. On u ovom slučaju ima izgled kao na slici desno. Dva nova bloka s oznakama *In1* i *Out1* nastala su automatski i predstavljaju ulaz i izlaz podsustava. Ovi se blokovi inače mogu naći u biblioteci *Ports&Subsystems*.



Nalogom *View/Model browser options/Model browser* u traci s nalozima, otvara se tzv. *Model Browser* s hijerarhijskom strukturom sustava, u kojoj se vidi da model *vjezba1* posjeduje podsustav *Subsystem*. Novonastali podsustavi mogu se svrstati u vlastitu grupu ili biblioteku, i koristiti u programiranju kao ostali blokovi Simulinka.

4.7. Prikazivanje karakteristika

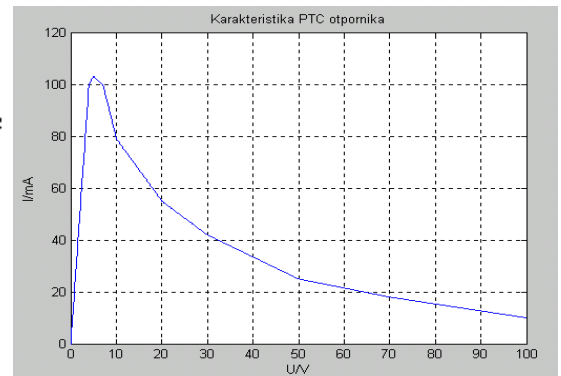
Često se javlja potreba za prikazivanjem funkcionalnih ovisnosti među diskretno izmjerenim parametrima koje se ne daju izraziti jednadžbom. Ove se ovisnosti u Simulinku daju grafički prikazati pomoću blokova *Look-Up Table* i *Look-Up Table (2-D)* u biblioteci *Functions&Tables*.

U cilju definiranja jedne karakteristike moraju se zadati dva jednako duga vektora, vektor ulaznih i vektor izlaznih vrijednosti. Ako jedna ulazna vrijednost nije sadržana u ulaznom vektoru, ona će biti linearno interpolirana ili ekstrapolirana. Obradimo i ovdje jedan primjer. Odaberimo ovisnost $I=f(U)$ jednog PTC otpornika, čiji su parovi izmjerenih vrijednosti svrstani u donju tablicu.

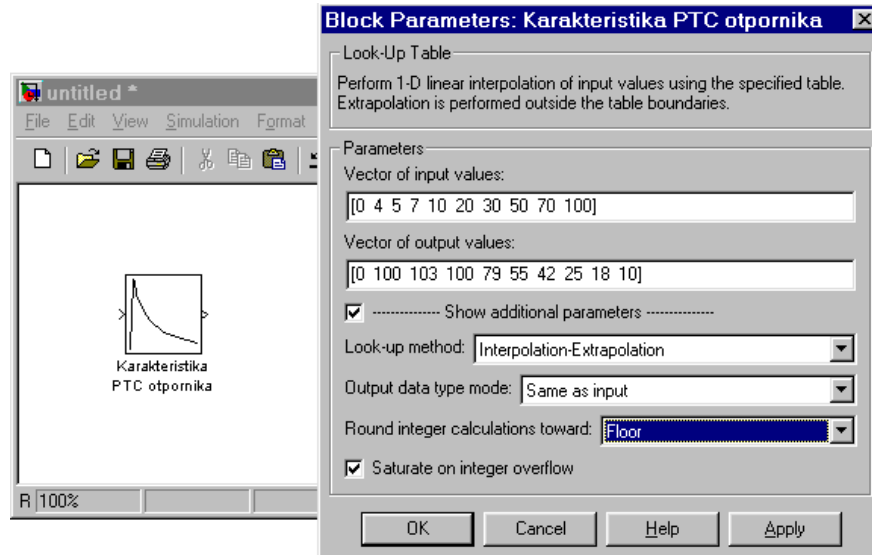
U/V	0	4	5	7	10	20	30	50	70	100
I/mA	0	100	103	100	79	55	42	25	18	10

Programirajmo prikaz ove karakteristike u MATLAB-u, kako bismo joj mogli vidjeti izgled, a za bolji grafički prikaz primjenimo MATLAB-ov nalog `interp1` koji interpolira vrijednosti u razmacima od 0.1 V. Izgled i redoslijed naloga u MATLAB-u prikazan je dolje lijevo, a programirane karakteristike desno pored njega.

```
>> volt1=[0 4 5 7 10 20 30 50 70 100];
>> mamper1=[0 100 103 100 79 55 42 25 18 10];
>> vl=(0:0.1:100);
>> krivuljal=interp1(volt1,mamper1,vl,'linear');
>> plot(vl,krivuljal)
>> grid
>> xlabel('U/V')
>> ylabel('I/mA')
>> title('Karakteristika PTC otpornika')
>>
```



Za unos karakteristike u Simulink koristimo blok Look-Up Table i dodajmo mu odgovarajući naziv, kao u slici dolje lijevo, a zatim otvorimo njegov prozor za parametrisiranje i unesimo izmjerene vrijednosti za oba vektora kao u slici dolje desno. Vidimo da je karakteristika poprimila oblik koji odgovara onom u dijagramu iz MATLAB-a. Time smo dobili potvrdu da su vrijednosti u prozoru za parametrisiranje korektno unesene.



Sada se na ulaz Simulink bloka Karakteristika PTC otpornika može priključiti odgovarajući izvor napona, a na izlazu se dobiju pripadajuće vrijednosti struja, koje se kao signal mogu voditi dalje.