

Univerzitet u Beogradu – Elektrotehnički fakultet

Uputstvo za merenje signala u programskom okruženju Matlab

- skripta za studente -

Nadica Miljković

u Beogradu, 2020. godine

Predgovor

Ovo uputstvo je nastalo sa ciljem da se pokrije deo gradiva na predmetu Praktikum iz softverskih alata ([PSA](#)) koji je bio posvećen merenju signala u Matlabu. Iako Matlab nije programsko okruženje koje se tipično koristi za merenje signala, korisno je znati da se merenje može izvesti i u Matlabu. Dodatno, gradivo koje je posvećeno Matlabu se može generalizovati i za druge programske jezike, pa je tako korisno za merenja generalno.

Od 2019/2020 godine, na predmetu PSA se ne drži deo posvećen merenju, ali se ova skripta ispostavila vrlo korisnom na predmetu Kliničko inženjerstvo ([KLI](#)) kao deo jedne vežbe koja je posvećena merenju signala u Kliničkom inženjerstvu.

Primeri u skripti su realizovani za LabVIEW verziju 2009, Matlab i Simulink verzije R2013b i na operativnom sistemu Windows 8. Tekst obiluje hiperlinkovima, pa iscrpljujuća lista referenci ne postoji, a čitaoci se takođe upućuju na zvaničnu literaturu pomenutih programskih jezika i paketa.

u Beogradu, 10. aprila 2020. godine

Autorka

Sadržaj

1 Merenje signala	4
2 Merenje audio signala	5
Prikaz merenih audio signala.....	6
Snimanje audio signala u fajl i čitanje iz fajla	7
Dodatni zadaci za rad sa audio signalima.....	8
Zadatak 1	8
Zadatak 2	8
Zadatak 3	8
3 Serijski port.....	9
Primer merenja signala sa UNO hardverom.....	9
4 Merenje signala primenom A/D kartice	13
<i>Foreground</i> merenje signala.....	17
Rezime	19
5 Merenje signala u programskom okruženju Simulink	20
Dodatak A – kopiranje i čuvanje figura u Matlabu	22
Dodatak B - Matlab kod.....	24
Primeri merenja audio signala.....	24
Merenje naponskih signala	26
Literatura	28

1 Merenje signala

U mernim sistemima koji su zasnovani na primeni računara od značaja je korišćenje pogodnog softverskog alata koji omogućava merenje signala. Najčešće se kod nas na Elektrotehničkom fakultetu Univerziteta u Beogradu za analizu merenih signala koristi Matlab (The Mathworks, Natick, USA) programsko okruženje. Kako bi se, radi jednostavnosti projektovanja aplikacija, koristio isti program i za analizu i za merenje signala, u Matlabu je moguće realizovati merenje audio i naponskih signala koji će biti objašnjeni u ovom priručniku.

Današnji računari opšte namene, mogu uz pomoć odgovarajućeg hardvera koji najčešće uključuje analogno/digitalne (A/D) konvertore i namenske softverske aplikacije, postati fleksibilni i integrisani merni sistemi.

Skup alati u Matlabu za akviziciju signala (eng. *Data Acquisition Toolbox*, <http://www.mathworks.com/products/daq/>, pristupljeno 9. aprila 2020.) omogućava povezivanje Matlab sa hardverom za akviziciju. I to hardvera različitih proizvođača kao što su National Instruments, Measurement Computing, Advantech, Data Translation, g.tec i drugih (za kompletnu listu pogledati Matlab sajt - *Connect Matlab and Simulink to Hardware*, http://www.mathworks.com/hardware-support/home.html?s_tid=srchtitle, pristupljeno 9. aprila 2020). Hardver ne uključuje samo A/D konvertore, već i digitalne osciloskope, LCR metre i druge merne uređaje. Ove alatke, osim merenja signala, omogućavaju i upis signala na analogne i digitalne izlaze u cilju upravljanja, signalizacije i dr.

Za instalirano 32-bitno programsko okruženje Matlab¹ koristi se *Legacy* interfejs, a na 64-bitnom programskom okruženju, koristi se *Session based* interfejs. Odnosno, način na koji se kreira kod koji omogućava merenje signala se razlikuje na 32-bitnom i na 64-bitnom Matlabu. U ovom priručniku objašnjen je samo način merenja signala na računaru sa 64-bitnim Matlabom i sa Windowsom 8. Za odabir odgovarajućeg interfejsa pogledati preporuku proizvođača na sajtu *Choose the Right Interface*, Mathworks, <http://www.mathworks.com/help/daq/choose-the-right-interface.html> (sada se koristi samo *Session based* interfejs, a ova stranica je bila dostupna februara 2019). Ukratko, *Session based* interfejs je noviji interfejs koji uključuje određena unapređenja u odnosu na *Legacy* interfejs i dužu listu komercijalno dostupnog hardvera koji se može povezati sa Matlabom, pa je zato njemu i posvećena posebna pažnja u ovoj skripti.

Postoje Matlab aplikacije, koje omogućavaju merenje signala, bez pisanja koda, samo uz pomoć podešavanja parametara akvizicije, a za koje je moguć automatski generisati kod (pogledati video uputstvo za automatsko generisanje koda na Matlab sajtu: *Connecting to DAQ Devices Without Writing Code*, od Renee Qian iz MathWorks <http://www.mathworks.com/videos/connecting-to-daq-devices-without-writing-code-98130.html>, pristupljeno 9. aprila 2020).

¹ 32-bitni Matlab se može instalirati i na 64-bitnom operativnom Windows sistemu.

2 Merenje audio signala

Jedna od osnovnih funkcija za merenje audio signala koja se koristi u Matlabu naziva se *audiorecorder* funkcija (<http://www.mathworks.com/help/matlab/ref/audiorecorder.html>, pristupljeno 9. aprila 2020). Ova funkcija kreira objekat² za snimanje audio signala i može (ali i ne mora) imati ulazne parametre (ako ih korisnica ili korisnik ne zadaju, onda se koriste podrazumevani parametri). Ulazni parametri su frekvencija odabiranja u Hz *Fs* (podrazumevana vrednost je 8000 Hz), broj bita kojim je predstavljen jedan odbirak *nBits* (podrazumevana vrednost je 8 bita), broj kanala (1- mono i 2- stereo), *nChannels* (podrazumevana vrednost je 1) i identifikacioni broj uređaja (za slučaj da ima više audio periferija na računaru) *ID* (podrazumevana vrednost je -1 odnosno eng. *default device*).

Da bi se proverilo koji su audio uređaji dostupni na računaru, moguće je iskoristiti funkciju *audiodevinfo* (<http://www.mathworks.com/help/matlab/ref/audiodevinfo.html>, pristupljeno 9. aprila 2020). Ako se ova funkcija pokrene u komandnom prozoru Matlabu, može se dobiti izveštaj o dostupnim uređajima u vidu strukture u kojoj se nalazi dostupni ulazi (mikrofoni) i izlazi (zvučnici) audio uređaja koji su povezani sa računarom. Ako se, u komandnom prozoru Matlabu, ukuca sledeći kod sa podrazumevanim ulaznim parametrima:

```
>> recObj = audiorecorder
```

dobiće se izveštaj kao na Sl. 1.

```
recObj =  
  
audiorecorder with properties:  
  
    SampleRate: 8000  
    BitsPerSample: 8  
    NumberOfChannels: 1  
    DeviceID: -1  
    CurrentSample: 1  
    TotalSamples: 0  
    Running: 'off'  
    StartFcn: []  
    StopFcn: []  
    TimerFcn: []  
    TimerPeriod: 0.0500  
    Tag: ''  
    UserData: []  
    Type: 'audiorecorder'
```

Sl. 1, Izveštaj u komandnom prozoru nakon poziva funkcije *audiorecorder*.

² Vrsta podataka u Matlabu.

Rezultati su isti (Sl. 1) ako se ova funkcija pozove sa podrazumevanim parametrima:

```
>> recObj = audiorecorder(8000, 8, 1, -1)
```

Dalje, moguće je sa ovako definisanim objektom *recObj* snimiti zvuk pozivom funkcije *recordblocking* (<http://www.mathworks.com/help/matlab/ref/audiorecorder.recordblocking.html>, pristupljeno 9. aprila 2020). Ova funkcija za ulazne parametre ima naziv kreiranog objekta (*recObj*) i dužinu snimanja koja se zadaje u sekundama. Za snimanje signala sa definisanim audio objektom kao na Sl. 1 u trajanju od 5 s, potrebno je u komandnom prozoru, pozvati sledeći kod:

```
>> recordblocking(recObj, 5);
```

Ovaj kod je moguće snimiti u .m skriptu i pokretati odabirom opcije *Run* iz padajućeg menija *Editor* glavnog Matlab prozora ili pritiskom tastera *F5* na tastaturi. Ako se kod pokreće preko .m skripta, uobičajeno se realizuje znak (signal namenjen korisniku) za početak i kraj snimanja koji se korisniku ispisuje na ekranu. Funkcija *disp* (<http://www.mathworks.com/help/matlab/ref/disp.html>, pristupljeno 6. aprila 2020.) se može koristiti za potrebe ispisa poruke korisniku na ekranu. Primer koda je:

```
>> disp('Snimanje počinje...')
>> recordblocking(recObj, 5);
>> disp('Snimanje je završeno.');
```

Primiti da je u prethodnom kodu tekst koji se ispisuje na ekranu komandnog prozora napisan pod jednostrukim navodima. To je rezultat činjenice da se radi o podatku tipa *string* (tekstualni podatak). U Matlabu su takvi podaci prikazani i odgovarajućom bojom.

Za preslušavanje audio signala³, moguće je iskoristiti funkciju *play*, sledećim pozivom:

```
>> play(recObj);
```

Dalje, mereni signal (tj. odbirke merenog signala), moguće je snimiti u matricu pozivom funkcije *getaudiodata* (<http://www.mathworks.com/help/matlab/ref/audiorecorder.getaudiodata.html>, pristupljeno 9. aprila 2020):

```
>> pom = getaudiodata(recObj);
```

Prikaz merenih audio signala

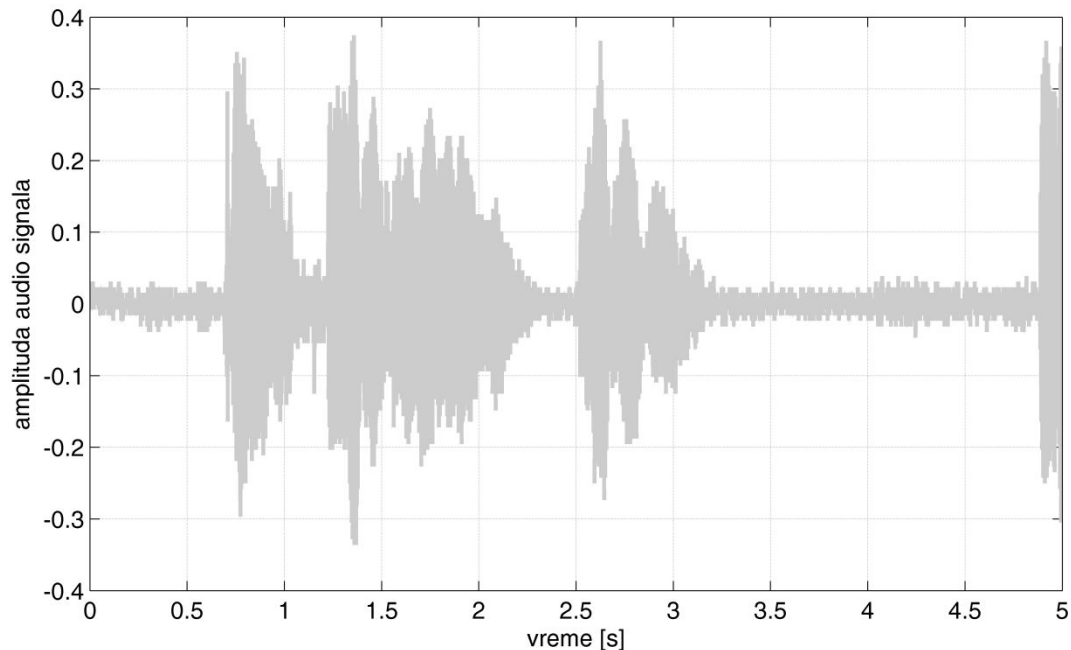
Nakon izvršavanja prethodne komande u *Workspace* prozoru Matlab pojavice se promenljiva *pom*. Duplim klikom na ovu promenljivu, otvara se prozor *Variables* u kome je moguće pregledati sadržaj ove matrice. Međutim, moguće je i prikazati ovaj signal na grafiku. Pre prikaza signala, potrebno je definisati

³ Ovu funkciju je moguće primeniti samo na podacima tipa strukture kao što je *recObj*, ali ne i na matricama koje sadrže odbirke signala.

vremensku osu. Za poznatu frekvenciju odabiranja (koja se nalazi u strukturi *recObj*, a moguće joj je pristupiti kao *recObj.sampleRate*), to se postiže, sledećim kodom:

```
>> time = 0: 1/recObj.sampleRate: (length(pom)-1) ./recObj.sampleRate;
```

Korišćenjem funkcije *plot* (<http://www.mathworks.com/help/matlab/ref/plot.html>, pristupljeno 9. aprila 2020), moguće je prikazati signal u vremenskom domenu kao na Sl. 2.



Sl. 2, Prikaz merenog audio signala sa vremenskom osom u sekundama.

Snimanje audio signala u fajl i čitanje iz fajla

Međutim, kako bi odbirci ovog signala bili dostupni i kada Matlab nije uključen, poželjno je snimiti signale u fajl. Najčešće korišćen format za audio signale je forma sa ekstenzijom *.wav* (eng. *Waveform Audio File Format*, <https://en.wikipedia.org/wiki/WAV>). Za snimanje matrice u *.wav* format, moguće je koristiti funkciju *audiowrite* (<http://www.mathworks.com/help/matlab/ref/audiowrite.html>, pristupljeno 9. aprila 2020.) funkciju. Ova funkcija može se realizovati sa tri ulazna parametra: *filename* - podatak tipa string koji definiše ime fajla, *y* - matrica u kojoj se nalazi audio signal i *Fs* -frekvencija odabiranja audio signala. U prethodnom primeru, to se postiže sledećim kodom⁴:

```
>> audiowrite('mojSignal.wav', pom, recObj.sampleRate);
```

U folderu koji je odabran kao tekući tj. *Current Folder* u Matlabu, može se uočiti novi fajl *mojSignal.wav* koji je moguće preslušati u nekom od programa za preslušavanje audio fajlova na računaru kao što je Windows Media Player (https://en.wikipedia.org/wiki/Windows_Media_Player) firme Microsoft ili VLC

⁴ Primititi da je prilikom definisanja imena fajla u kome bi funkcija trebalo da snimi promenljivu, potrebno uneti i ekstenziju. Ove je najčešće slučaj sa funkcijama koje služe za čitanje iz fajlova u Matlabu, jer najveći broj tih funkcija podržava i druge formate.

media player koji je deljen pod licencom slobodnog softvera neprofitne VideLAN organizacije (https://en.wikipedia.org/wiki/VLC_media_player) ili bilo koji po izboru korisnice ili korisnika.

Za snimljene audio fajlove, nekada je potrebno realizovati kod koji će omogućiti da se ovi fajlovi ponovo u Matlabu učitaju u matrice, kako bi se dalje prikazivali (kao na Sl. 2) ili analizirali.

Koristi se *wavread* funkcija (<http://www.mathworks.com/help/matlab/ref/wavread.html>, dostupno marta 2019.) da bi se signal učitao iz fajla, na sledeći način⁵:

```
>> [y, Fs] = wavread('mojSignal.wav');
```

U izlaznim promenljivim *y* i *Fs* smešteni su odbirci audio signala i frekvencija odabiranja u Hz, respektivno.

Dodatni zadaci za rad sa audio signalima

Zadatak 1

Realizovati kod za snimanje audio signala sa frekvencijom odabiranja koja ispunjava Nikvist-Šenonovu teoremu (https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem) odabiranja signala za snimanje govornog signala za mono audio signal u trajanju od 2 sekunde korišćenjem računarskog mikrofona. Prikazati signal primenom *plot* funkcije sa vremenskom osom u sekundama i snimiti signal u fajl sa ekstenzijom *.wav*. Označiti ose na grafiku i dodati naslov.

Zadatak 2

Realizovati kod za učitavanje 3 audio signala koji su snimljeni sa podrazumevanim parametrima u trajanju od 5 s u Matlabu sa ekstenzijom *.wav*. Prikazati sve signale na jednom grafiku primenom *subplot* (<https://www.mathworks.com/help/matlab/ref/subplot.html>) funkcije⁶ u Matlabu sa vremenskom osom u sekundama. Označiti ose na graficima i dodati naslove graficima.

Zadatak 3

Realizovati kod koji omogućava merenje 10 uzastopnih sekvenci audio signala (u trajanju od po 7 sekundi) i snimiti sve signale u *.wav* fajlove.

⁵ Prilikom korišćenja ove funkcije, nije potrebno uneti ekstenziju fajla iz koga se čitaju podaci. U novijim verzijama Matlab-a, koristi se *audioread* (<http://www.mathworks.com/help/matlab/ref/audioread.html>, pristupljeno 9. aprila 2020.) funkcija umesto *wavread* funkcije (ulazni i izlazni parametri su slični).

⁶ Funkcija *subplot* služi za podelu grafika za iscrtavanje signala na polja u kojima je moguće zasebno iscrtati signale.

3 Serijski port

Merenje preko serijskog porta se odnosi na serijsku komunikaciju koja omogućava prenos podataka (https://en.wikipedia.org/wiki/Serial_port). Uvek se prenosi jedan bit informacije u jednom trenutku za razliku od paralelnog porta. Ovaj vid komunikacije se najčešće koristi prilikom povezivanja računara sa periferijama.

Komunikacija između Matlaba (u opštem slučaju bilo kog programskog jezika) i nekog hardvera se može obaviti preko serijskog porta primenom funkcija za pristup serijskom portu, od kojih je najvažnija funkcija *serial*⁷ koja kreira objekat za komunikaciju (prenos podataka) sa serijskog porta (<https://www.mathworks.com/help/matlab/ref/serial.html>, pristupljeno aprila 2018).

Korišćenje ove opcije za merenje signala uključuje sledeće korake:

1. signal sa senzora se dovodi na ulaz A/D konvertora,
2. potom se vrši prenos digitalizovanog signala na računar,
3. odgovarajući softver omogućava da se digitalizovan signal štampa na serijskom portu i na kraju
4. Matlab pristupa serijskom portu i prikuplja odbirke signala.

Merenje koje je prikazano u ovom poglavlju je inspirisano materijalima koji su dati u [1].

Primer merenja signala sa UNO hardverom

Za merenje signala u Matlabu preko serijskog porta potrebno je najpre povezati hardver sa računarem. Potom, potrebno je pokrenuti odgovarajući kod (u ovom slučaju Arduino) koji omogućava ispis merenih odbiraka na serijskom portu u odgovarajućim vremenskim intervalima (koji su određeni željenom frekvencijom tj. periodom odabiranja). Arduino kod (<https://www.arduino.cc/en/Main/Software>, pristupljeno aprila 2018.) koji ima gore opisanu funkcionalnost pod nazivom *analogReadBezKasnjenja.ino* dat je u nastavku:

```
/*  
AnalogReadSerial + BlinkWithoutDelay: kombinacijom ova dva koda nastao je jedan, a ideja je bila da  
frekvencija odabiranja bude fiksna, odnosno da se na svakih "interval" milisekundi šalje po jedan odbirak  
na serijski port, a da se on kasnije snimi u Matlabu.  
*/  
unsigned long previousMillis = 0;
```

⁷ U novijim verzijama Matlaba preporučuje se korišćenje funkcije *serialport* umesto funkcije *serial* (<https://www.mathworks.com/help/matlab/ref/serialport.html>, pristupljeno aprila 2020).

```

const long interval = 10; // ako je 10 znači da je sampling rate 100 Hz

void setup() {
  Serial.begin(9600); }

void loop() {

  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {

    previousMillis = currentMillis;

    int sensorValue = analogRead(A0);

    Serial.println(sensorValue); } }

```

Primiti funkciju *Serial.println* na kraju prethodnog koda koja na serijski port ispisuje vrednosti koje su učitane sa senzora. Sada je moguće na relativno jednostavan način pristupiti ovim podacima u Matlabu. Najpre je potrebno proveriti o kom serijskom portu se radi: to se može videti u *Device Manager*-u u Windows operativnom sistemu (ovde je pretpostavljeno da je UNO R3 hardver povezan sa COM3 serijskim portom).

Prvi korak u Matlabu je kreiranje objekta za serijsku komunikaciju primenom *serial* funkcije i otvaranje tj. pristup serijskom portu primenom *fopen* funkcije:

```

s = serial('COM3');
fopen(s)

```

Sadržaju promenljive *s* koja je kreirana u prethodnim koracima, moguće je pristupiti jednostavnim štampanjem sadržaja ove promenljive u komandnom prozoru ili pozivom odgovarajućeg svojstva/ava sa tačkom nakon naziva promenljive *s*:

```

s
s.ValuesReceived

```

Sve vrednosti tj. parametre serijske komunikacije koji se nalaze u objektu *s* je moguće po potrebi izmeniti. Na primer, ako je potrebno promeniti brzinu ispisa na serijskom portu tzv. *baud rate*, može se koristiti sledeća komanda:

```
s.BaudRate = 9600;
```

Da bi se podaci pročitali, potrebno je primeniti *fread* funkciju koja, u ovom primeru, čita podrazumevanih 512 podataka koji se nalaze na serijskom portu:

```
pod = fread(s);  
length(pod)
```

Za kraj potrebno je obavezno zatvoriti serijsku konekciju primenom *fclose* funkcije i po želji, obrisati objekat *s*.

```
fclose(s)  
delete(s)  
clear s
```

Primer merenja 700 odbiraka signala sa serijskom porta je dat u nastavku ovog uputstva:

```
% PSA  
% ovaj dokument je inspirisan materijalima sa sajta:  
% https://www.mathworks.com/matlabcentral/fileexchange/26711-interacting-with-an-arduino-using-the-matlab-serial-commands  
close all; clear all; clc;  
  
s1 = serial('COM3'); % objekat za serijsku komunikaciju  
s1.BaudRate = 9600; % postavljanje parametara serijske komunikacije  
fopen(s1); % pocinje se sa komunikacijom  
  
%% Citanje sa serijskog porta  
ix = 1;  
w = fscanf(s1, '%s');  
while (ix < 701)  
    v(ix) = fscanf(s1, '%d');  
    ix = ix + 1;  
end  
  
%% Zatvoriti serijski port i nacrtati grafik izmerenih odbiraka signala  
fclose(s1);  
  
figure  
    plot(v)  
  
%% frekvencija odabiranja je 100, pa je potrebno kreirati vremensku osu  
fs = 100;  
t = 0:1/fs:(length(v)/fs - 1/fs);  
  
% napon (kako podatke sa UNO R3 kartice pretvoriti u napon)  
v = v * 5 / 1023;
```

```

figure
    plot(t, v, 'LineWidth', 2, 'Color', [0 .2 .5]);
    xlabel('vreme [s]');
    ylabel('napon [V]');
    title('Fotopletizmograf');
    grid on;
%% Snimiti podatke u txt
fileID = fopen('proba.txt', 'w'); % otvara datoteku za upis tj. write
fprintf(fileID, '%2.7f \r\n', v'); % definise nacin upisa u datoteku

fclose(fileID); % zatvara datoteku

```

Primititi u prethodnom primeru da je umesto *fread* funkcije korišćena *fscanf* funkcija za čitanje sa serijskog porta. Kada su izmereni podaci snimljeni u fajl primenom prethodnog koda za kasniju analizu tzv. *offline* analizu signala, potrebno je primeniti odgovarajuće funkcije za čitanje tih podataka iz fajla. U sledećem kodu prikazana su dva načina za čitanje tekstualnih podataka u Matlabu:

```

%% Pročitati podatke iz fajla

pod = load('proba.txt');
disp(['Srednja vrednost snimljenih signala je: ', num2str(mean(pod))]);
disp(['Standardna devijacija snimljenih signala je: ',
num2str(std(pod))]);

%% Drugi nacin za citanje podataka

fileID = fopen('pletizmograf.txt', 'r');
pod = fscanf(fileID, '%f');
fclose(fileID);
disp(['Medijana snimljenih podataka je: ', num2str(median(pod))]);

```

Korišćenje *disp* funkcije za ispis poruka i rezultata u komandnom prozoru je opciono. Dodatno, posebno je potrebno obratiti pažnju na to je potrebno da *Current folder* u Matlabu sadrži i kod i tekstualne datoteke sa snimljenim signalima. Za detaljne o *Current folder*-u u Matlabu pogledati u [2].

Važno je napomenuti da princip merenja podataka preko serijskog porta nije usko povezan sa UNO R3 hardverom tj. UNO R3 mikrokontrolerskom pločicom i Arduino softverom, a u opštem slučaju ni sa programskim okruženjem Matlab. Studenti i studentkinje mogu koristiti i neki drugi hardver koji bi povezali sa računarom, a signale prikupili i dalje analizirali u Matlabu ili koristili čitanje podataka sa serijskog porta u nekom drugom programskom jeziku, kao na primer u Python-u.

Jedna specifičnost korišćenog UNO R3 hardvera i Arduino softvera je što se podaci mere u opsegu od 0 do 1023 tj. u 1024 koraka koji odgovaraju opsegu od minimalnih 0 V do maksimalnih 5 V. Ovo je posledica činjenice da se radi o A/D konvertoru koji ima rezoluciju od 10 bita ($2^{10} = 1024$). Dalje, ovo znači da je najmanja razlika napona (odnosno rezolucija merenja) koja se na ovaj način može detektovati jednaka $(5 - 0) \text{ V} / 1024 = 4.88 \text{ mV}$.

4 Merenje signala primenom A/D kartice

U praksi se, vrlo često, koriste uređaji za akviziciju podataka (eng. *Data Acquisition*, https://en.wikipedia.org/wiki/Data_acquisition) specijalizovane namene. Rad ovih uređaja je zasnovan na primeni digitalnih procesora ili računara. Oni služe (najčešće) za merenje signala sa senzora. Ako je njihov rad zasnovan na primeni računara, onda je potrebno realizovati odgovarajući softver koji će omogućiti merenje, prikaz i analizu signala sa senzora. Postoje različiti proizvođači tih uređaja čija je osnovna funkcija A/D konverzija (eng. *analog to digital conversion*, https://en.wikipedia.org/wiki/Analog-to-digital_converter). Ti uređaji mogu biti i samo A/D konvertori, ali mogu sadržati i druge funkcije.

Ovde će biti, bez umanjenja opštosti, prikazan način merenja signala u Matlabu primenom A/D kartice proizvođača NI (National Instruments Inc., Austin, USA, <http://www.ni.com/>)

Da bi se pregledao spisak dostupnih NI uređaja na računaru koristi se `daq.getDevices` (<http://www.mathworks.com/help/daq/ref/daq.getdevices.html>, pristupljeno 9. aprila 2020.) komanda. Kucanjem ove komande u komandnom prozoru Matlaba dobija se izveštaj sa spiskom uređaja kao na Sl. 3. Prikaz na Sl. 3 se dobija u slučaju da ne postoji nijedan uređaj na računaru.

```
>> daq.getDevices

ans =

No data acquisition devices available.
Click here for a list of known vendors.
Click here for troubleshooting tips.
```

Sl. 3. Izgleda komandnog prozora programskog okruženja Matlab kao rezultat funkcije `daq.getDevices`.

Za potrebe testiranja programa, u slučaju kada hardver nije dostupan, moguće je simulirati NI uređaj na računaru. Pregled svih NI uređaja, ali i drugih periferija na računaru, je omogućen u programskom okruženju MAX⁸ (eng. *Measurement and Automation*, <http://www.ni.com/tutorial/4594/en/>, pristupljeno 9. aprila 2020). Izgled inicijalnog prozora MAX uređaja prikazan je na Sl. 4. MAX služi za konfiguraciju i podešavanje parametara uređaja i ovo okruženje se instalira zajedno sa drajverima NI uređaja.

MAX svakom instaliranom uređaju pridružuje određeni broj (eng. *device number*). Na primer, na Sl. 4 je prikazan uređaj sa identifikacijom "Dev1" kome je pridružen broj 1⁹. Uz ovaj ID uređaja se nalazi ikonica

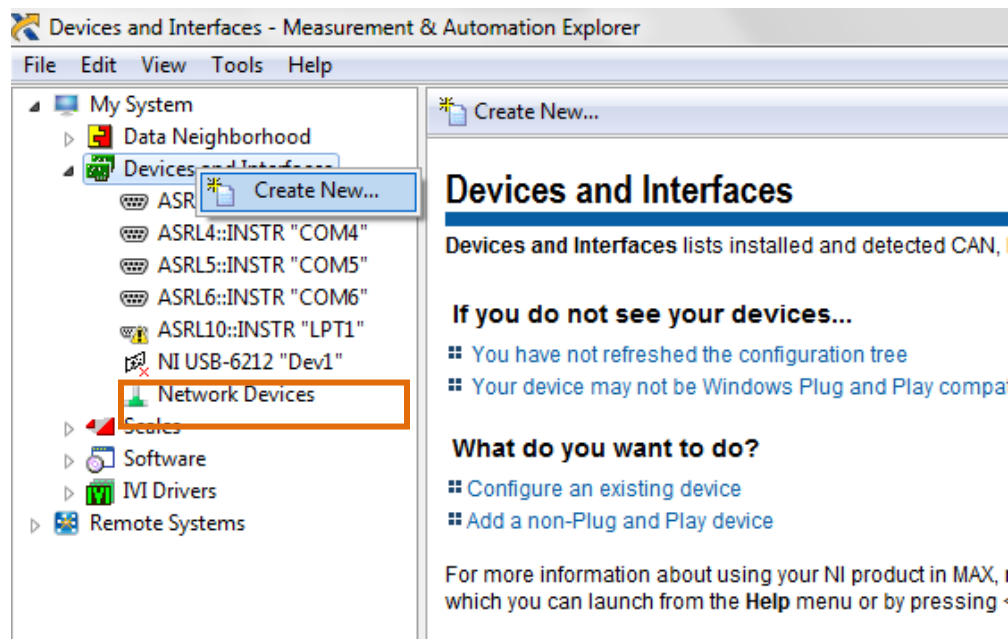
⁸ Da bi se instaliralo MAX programsko okruženje, potrebno je instalirati drajvere na računaru sa kojima u paketu dolazi MAX. Drajvere NI-DAQmx je moguće naći na sajtu: www.ni.com/en-rs/support/downloads/drivers.html, pristupljeno 9. aprila 2020.

⁹ Identifikacioni broj uređaja (ID) je od značaja, jer se preko ID-ja pristupa datom uređaju u Matlab kodu.

bele boje, koja označava da taj uređaj nije priključen na računar. Činjenica da uređaj ima svoj ID, znači da je uređaj bio priključen ranije na računaru. Uz uređaj je prikazan i njegov naziv NI USB 6212 (Sl. 4, www.ni.com/en-rs/support/model.usb-6212.html, pristupljeno 9. aprila 2020). MAX obuhvata ukupno 5 sledećih sekcija:

1. *Data Neighbourhood* - koja služi za definisanje posebnih merenja sa odgovarajućim hardverom.
2. *Devices and Interfaces* - prikazuje detektovane i instalirane NI uređaje i sastoji se iz sledećih elemenata:
 - a. *Properties* opcija (svojstva) koja služi za sinhronizaciju više uređaja i povezivanje sa konektorskim blokovima.
 - b. *Self Test* opcija za ispitivanje komunikacije između NI uređaja i računara.
 - c. *Test Panels* opcija za testiranje analognih i digitalnih ulaza i izlaza na uređaju.
3. *Scales* - omogućava skaliranje snimljenih signala sa senzora.
4. *Software* - prikazuje instaliran NI softver.
5. *IVI drivers* - omogućava da se softver koji je napisan za jedan uređaj koristi i za druge uređaje koji imaju isti skup drajvera.

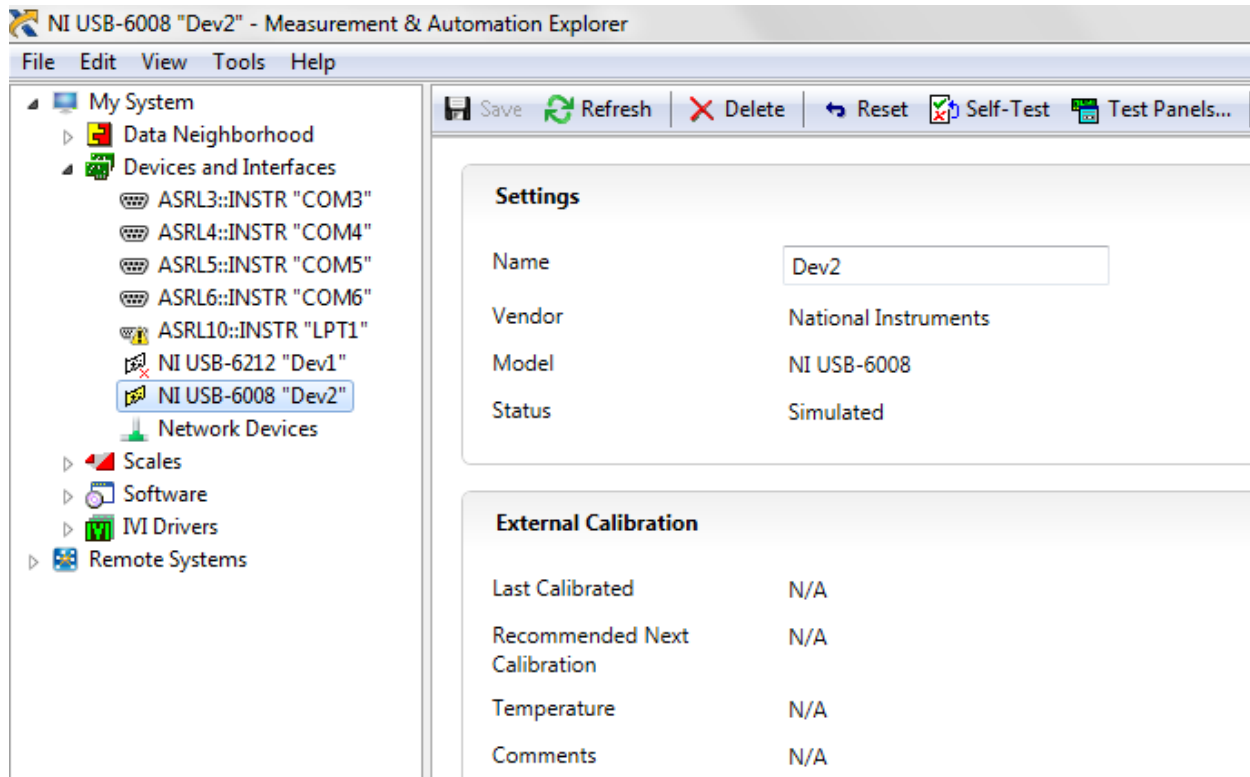
Od značaja za merenje signala u Matlab programskom okruženju signala sa NI uređajima je samo sekcija *Devices and Interfaces*, pa će na dalje, u tekstu, samo osnovna funkcionalnost ove sekcije biti objašnjena.



Sl. 4, Izgled prozora MAX uređaja sa proširenom sekcijom *Devices and Interfaces*. Desnim klikom računarskog miša na naziv ove sekcije omogućen je odabir opcije *Create New*. Na slici je uokviren NI uređaj koji nije priključen na računar, a čiji je ID "Dev1".

Ako se proširi kategorija *Devices and Interfaces*, mogu se uočiti instalirani NI uređaji (Sl. 5). Kako na Sl. 5 nije povezan nijedan uređaj, to će u sledećim koracima biti pokazano kako je moguće simulirati NI uređaj radi testiranja merenja u Matlabu.

Desnim klikom računarskog miša na *Devices and Interfaces* sekciju otvara se opcija *Create New* (Sl. 5). U otvorenom prozoru potrebno je odabrati opciju *Simulated NI-DAQmx Device or Modular Instrument*. Nakon toga, otvoriće se novi prozor u kome je potrebno odabrati kategoriju *USB DAQ* i odabrati uređaj NI USB 6008 (www.ni.com/en-rs/support/model.usb-6008.html, pristupljeno 9. aprila 2020.) uređaj¹⁰. Sada je u sekciji *Devices and Interfaces* moguće videti novi simulirani uređaj sa ikonicom žute boje čiji je ID "Dev2". Preimenovati (slično kao preimenovanje fajlova u Windows okruženju) naziv ovog uređaja u "Dev1" i naziv kartice koja je nekada bila priključena u "Dev11" (Sl. 5).

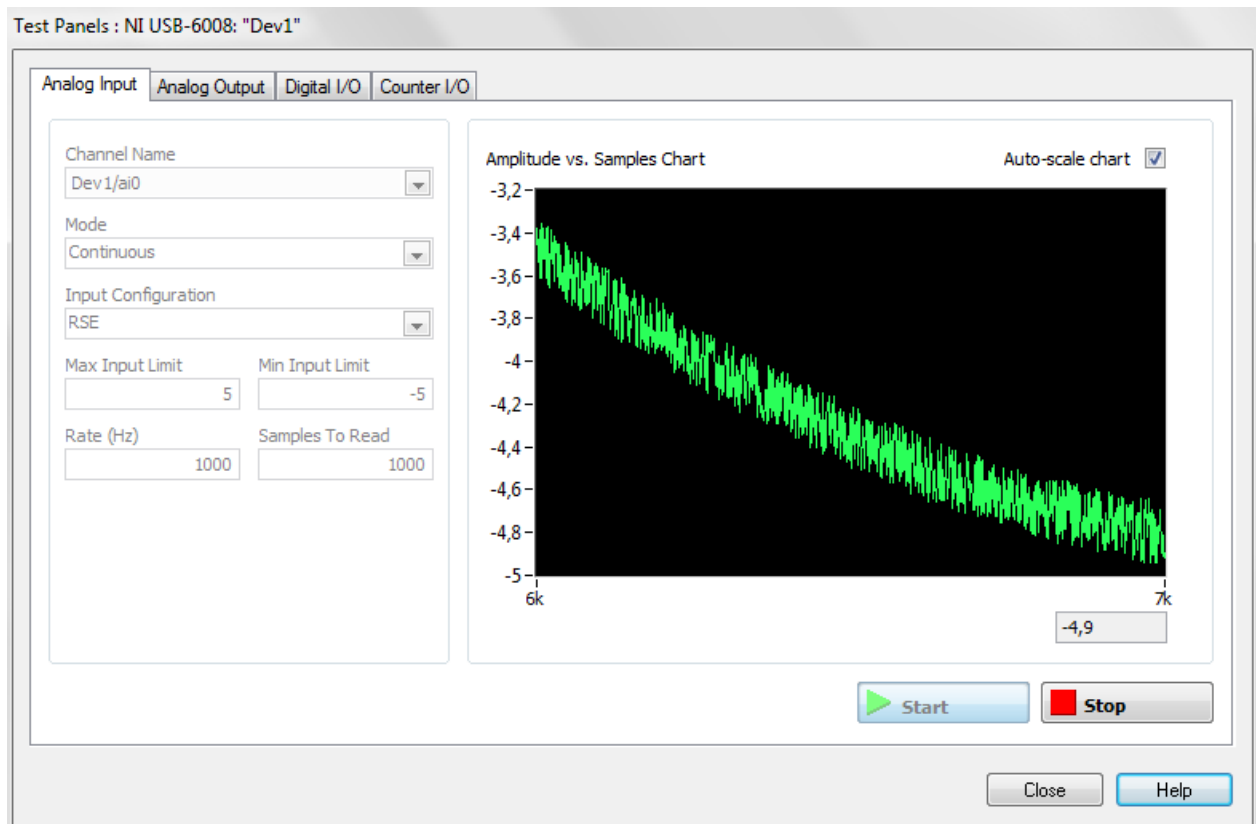


Sl. 5. Izgled MAX interfejsa u kome se nalazi simulirana kartica (ID = "Dev1") i kartica koja je nekada bila povezana sa računarom (ID = "Dev11").

Desnim klikom računarskog miša na naziv simulirane kartice otvoriti padajući meni i iz njega odabrati opciju *Test Panels*. Na kartici *Analog Input* odabrati prikaz 0-og kanala ("Dev1/ai0"¹¹), za *Mode* odabrati *Continuous*, konfiguraciju ulaza RSE (eng. *Reference Single Ended*), odnosno merenje u odnosu na referentnu tačku (masu). Frekvenciju odabiranja (eng. *Rate*) postaviti na 1000 Hz, broj odbiraka koji se prikazuje na ekranu (eng. *Samples To Read*) postaviti na 1000 i naponske granice ulaznog signala postaviti od -5 V do 5 V (eng. *Max Input Limit* i eng. *Min Input Limit*). Ovi parametri se biraju u zavisnosti od tipa signala koji se meri. Testiranje se započinje pritiskom na opciju *Start*. Primiti da je signal na simuliranoj kartici zašumljena sinusioda i da se kreće u zadatom opsegu od -5 V do 5 V (Sl. 6).

¹⁰ Primiti da je moguće simulirati relativno veliki broj komercijalno dostupnih uređaja.

¹¹ U zavisnosti od simuliranog hardvera, moguće je odabrati određen broj analognih ulaza. Pored analognih ulaza, moguće je birati i analogne izlaze, digitalne izlaze i dr, ali je ovde od značaja samo merenje analognih signala, pa ostale opcije neće biti razmatrane.



Sl. 6. Izgled ekrana *Test panel* alata za prikaz signala na simuliranom uređaju sa inicijalnim podešavanjem parametara merenja.

Isključiti prozor *Test Panel* na dugme *Stop* (Sl. 6). Ako je ovaj prozor aktivan, uređaju nije moguće pristupiti iz drugih programskih okruženja kao što su Matlab ili Simulink. Sada je potrebno otvoriti komandni prozor Matlaba i ponovo ukucati komandu *daq.getdevices*. Primititi da je sada simulirani uređaj dostupan i u Matlabu.

Rezultat komande *daq.getdevices*, moguće je dodeliti nekoj promenljivoj *d*:

```
>> d = daq.getDevices;
```

U tom slučaju, u promenljivoj *d* su smešteni podaci o uređajima koje je moguće koristiti za merenje signala u Matlabu. Da bi se pristupilo podacima uređaja po redosledu po kojem je njima dodeljen ID broj potrebno je iskoristiti sledeću liniju koda:

```
>> dev = d(1);
```

U ovom slučaju to nije bilo potrebno, jer postoji samo jedan uređaj, ali u slučaju da uređaja ima više njima bi se pristupalo redom *d(1)*, *d(2)*, ... Kako bi se započelo merenje, potrebno je kreirati sesiju *s*

primenom `daq.createSession` (<http://www.mathworks.com/help/daq/ref/daq.createsession.html>¹², pristupljeno 9. aprila 2020.) funkcije na sledeći način:

```
>> s = daq.createSession('ni');
```

Ulazni parametar ove funkcije je naziv skraćenica (tipa string) koja označava naziv kompanije čiji se uređaj povezuje sa Matlab kodom. Osim *ni* (<http://www.ni.com/>, pristupljeno 9. aprila 2020), validni su i *digilent* (<http://store.digilentinc.com/>, pristupljeno 9. aprila 2020.) i *directsound* ([https://msdn.microsoft.com/en-us/library/windows/desktop/ee416960\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ee416960(v=vs.85).aspx), pristupljeno 9. aprila 2020). Za konfiguraciju merenja (nalik onoj koja je realizovana u *Test Panel*-u MAX softverskog okruženja), potrebno je koristiti Matlab funkciju `addAnalogInputChannel` (<http://www.mathworks.com/help/daq/ref/addanaloginputchannel.html>¹³, pristupljeno 9. aprila 2020.) na sledeći način:

```
>> addAnalogInputChannel(s, 'Dev1', 'ai0', 'Voltage');
```

Ulazni parametri ove funkcije su promenljiva sesije *s*, ID uređaja (koji je pročitao iz MAX-a) u obliku string promenljive "Dev1", naziv kanala (kao u *Test Panel*-u MAX-a) "ai0" i tip signala koji se meri (u ovom slučaju naponski signal) "Voltage".

Podešavanje parametara za merenje je moguće uraditi na sledeći način:

1. merenje u odnosu na masu pomoću funkcije:

```
>> s.Channels.TerminalConfig = 'SingleEnded';
```

2. broj izmerenih odbiraka:

```
>> s.NumberOfScans = 1000;
```

3. frekvencija odabiranja u Hz:

```
>> s.Rate = 1000;
```

4. naponski opseg kanala u V:

```
>> ch.Range = [-5, 5];
```

Foreground merenje signala

Postoje i dodatne opcije za podešavanje merenja na kanalu u `addAnalogInputChannel` (<http://www.mathworks.com/help/daq/ref/addanaloginputchannel.html>¹⁴, pristupljeno 9. aprila 2020.)

¹² Primititi da se u novijim verzijama Matlaba, umesto ove i sličnih funkcija koristi *Data Acquisition Interface*, <https://www.mathworks.com/help/daq/daq.interfaces.dataacquisition.html>, pristupljeno 9. aprila 2020.

¹³ Pogledati fusnotu 12.

¹⁴ Pogledati fusnotu 12.

funkciji. Za započinjanje merenja u *Foreground* sesiji i smeštanje podataka u promenljivu *data*, moguće je iskoristiti kod:

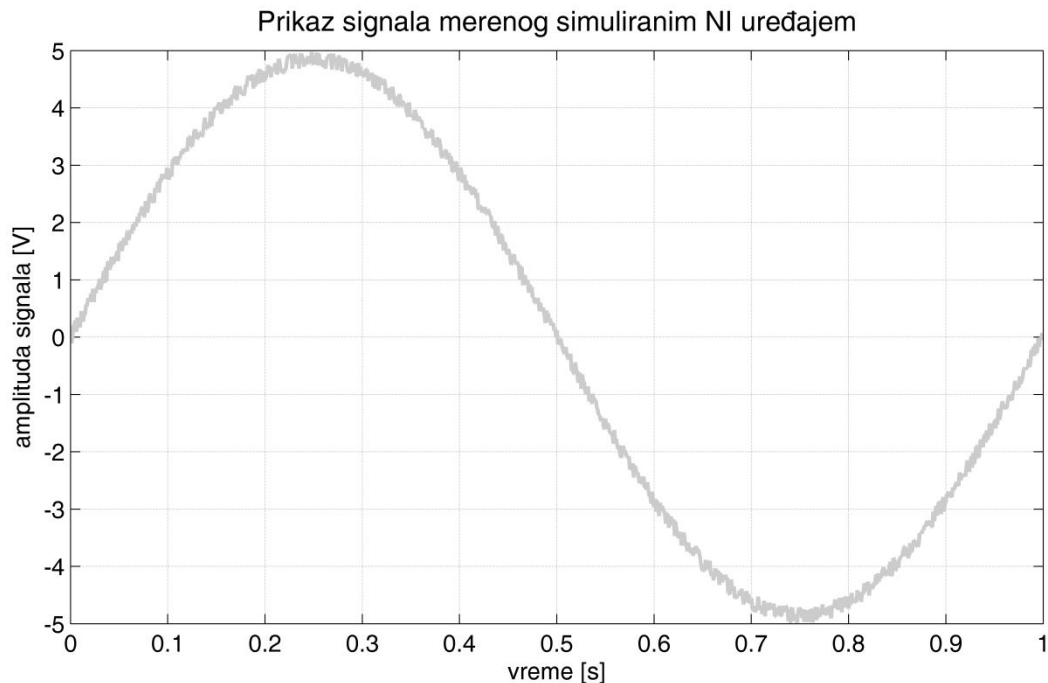
```
>> data = startForeground(s);
```

Ova komanda započinje *foreground* operaciju tj. operaciju "u prvom planu", gde je zauzeće procesora isključivo rezervisano za potrebe merenja signala. Osim *foreground* operacije, postoji i *background* merenje signala tokom kog je omogućeno da procesor obavlja i druge zadatke pored merenja signala.

Kako bi se prikazao signal u vremenskom domenu, potrebno je definisati vektor vremenske ose signala pomoću dostupnog podatka o broju odbiraka i frekvenciji odabiranja:

```
>> time = 0: 1/s.Rate: (length(data) - 1) ./s.Rate;
```

Izmeren naponski signal sa simuliranog uređaja se može prikazati, slično kao audio signal, pomoću *plot* funkcije i dobijen grafik je prikazan na Sl. 7.



Sl. 7, Prikaz naponskog signala koji je izmeren u Matlabu pomoću simuliranog NI uređaja.

Kao i na primeru audio signala, korisno je imati dostupne odbirke signala i nakon izlaska iz Matlab programskog okruženja. Signal je moguće sačuvati u raznim formatima i fajlovima. Na primer, za čuvanje signala u Excel fajlu (ekstenzija *.xlsx*), potrebno je iskoristiti funkciju *xlswrite* (<http://www.mathworks.com/help/matlab/ref/xlswrite.html>¹⁵, pristupljeno 9. aprila 2020), koja kao ulazne parametre ima naziv fajla (u obliku stringa) i promenljivu u kojoj se signal nalazi:

¹⁵ Primititi da se u novijim verzijama Matlab-a ova funkcija zamenjena drugim preporučenim, kao npr. *writetable*.

```
>> xlswrite('izmerenSignal', data);
```

Ova komanda će u aktivnom folderu Matlab programskog okruženja (*Curent folder*) kreirati fajl sa imenom *izmerenSignal.xlsx* i u njemu smestiti niz odbiraka signala iz promenljive *data*. Za čitanje signala iz Excel fajla koristi se funkcija *xlsread*¹⁶ (<http://www.mathworks.com/help/matlab/ref/xlsread.html>)¹⁷, pristupljeno 9. aprila 2020):

```
>> num = xlsread('izmerenSignal');
```

Rezime

Generalno, merenje signala u Matlabu sa *Session based* interfejsom se sastoji iz par koraka koji se neznatno razlikuju na primerima *Foreground* akvizicije i merenja audio signala:

1. kreiranje sesije odnosno povezivanje Matlaba sa odgovarajućim hardverom,
2. podešavanje parametara sesije,
3. definisanje kanala na odabranom uređaju,
4. podešavanje parametara kanala,
5. početak akvizicije,
6. opciono: prikaz signala tokom akvizicije,
7. opciono: smeštanje merenih odbiraka u matricu,
8. opciono: prikaz signala i
9. opciono: snimanje signala u fajl.

Primiti da su koraci za rad u *Session based* interfejsu isti kao i za snimanje audio signala ili za rad sa serijskim portom.

¹⁶ Ova funkcija smešta podatke u matricu *num*.

¹⁷ Pogledati fusnotu 14. Slično se primenjuje i za čitanje iz .xlsx fajla.

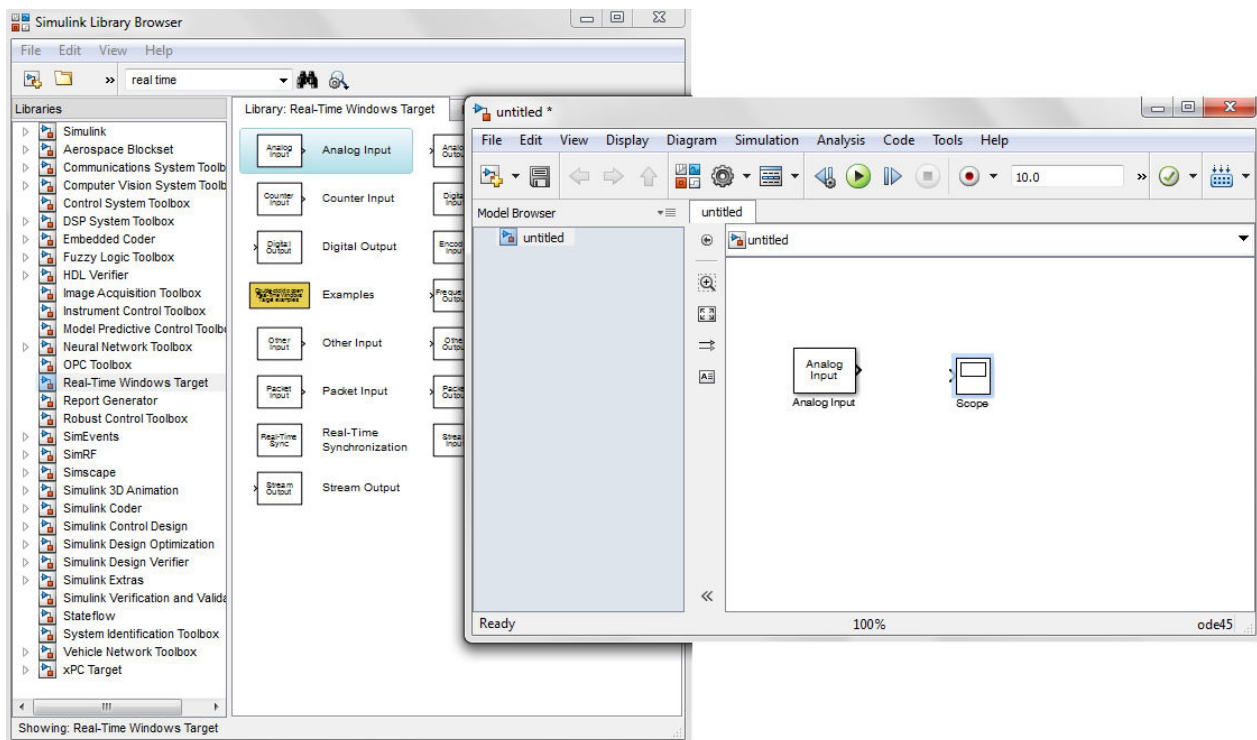
5 Merenje signala u programskom okruženju Simulink

U Simulinku je moguće realizovati merenje signala primenom tri skupa alati:

1. *Real Time Windows Target* skup alati,
2. *Simulink Desktop Real Time* i
3. *Data Acquisition* skup alati.

Za rad u *Real Time Windows Target*-u, potrebno je proveriti da li je kernel za rad u realnom vremenu instaliran na računaru unošenjem komande *rtwintgt* u komandni prozor Matlab programa. U slučaju da nije instaliran pokrenuti instalaciju komandom *rtwintgt -setup*, potom restartovati i ponovo pokrenuti Matlab¹⁸. Nakon toga pokrenuti Simulink i ovaj skup alati bi trebalo da bude dostupan.

Jednostavan primer akvizicije signala u Simulink-u može se realizovati odabirom bloka *Analog Input* (iz kategorije *Real Time Windows Target*) i povezivanjem ovog bloka na *Scope* (osciloskop) kao na Sl. 8.

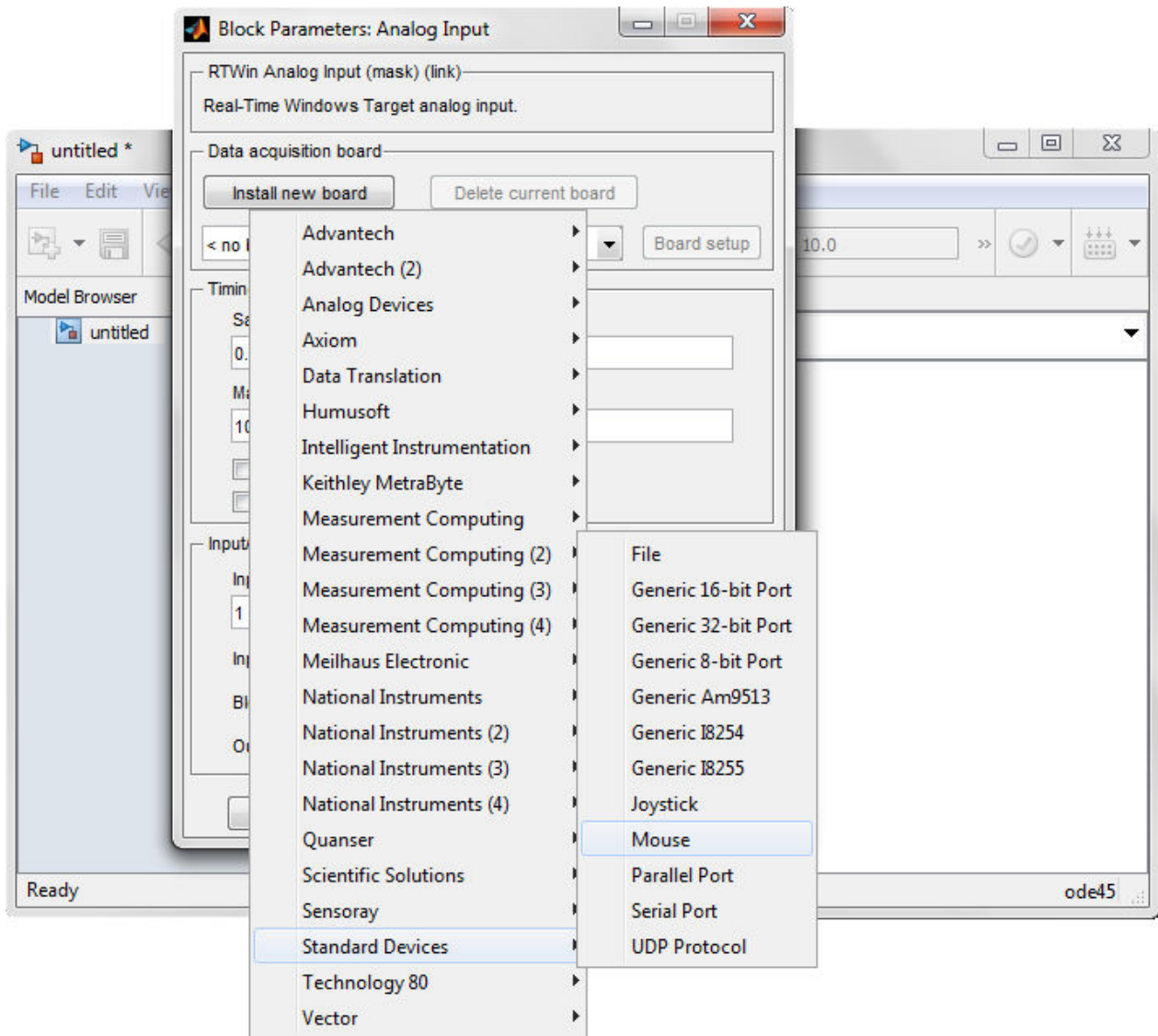


Sl. 8, Simulink biblioteka funkcija i jednostavan program sa analognim ulazom i osciloskopom.

Duplim klikom računarskog miša na *Analog Input* funkciju/blok, otvara se prozor kao na Sl. 9. Pritiskom na taster *Install new board* otvara se padajući meni u kome je po kategorijama moguće odabrati neki od uređaja sa koga će biti realizovano merenje. Iz kategorije *Standard Devices* odabrati opciju *Mouse* (računarski miš). Pokrenuti program i testirati rad programa.

¹⁸ Ova opcija neće raditi pod Windows-om 10, ako se koristi Matlab R2013b. Za detalje pogledatu Matlab dokumentaciju. Navedeni primer je funkcionalan za Windows 8 i za Matlab R2013b.

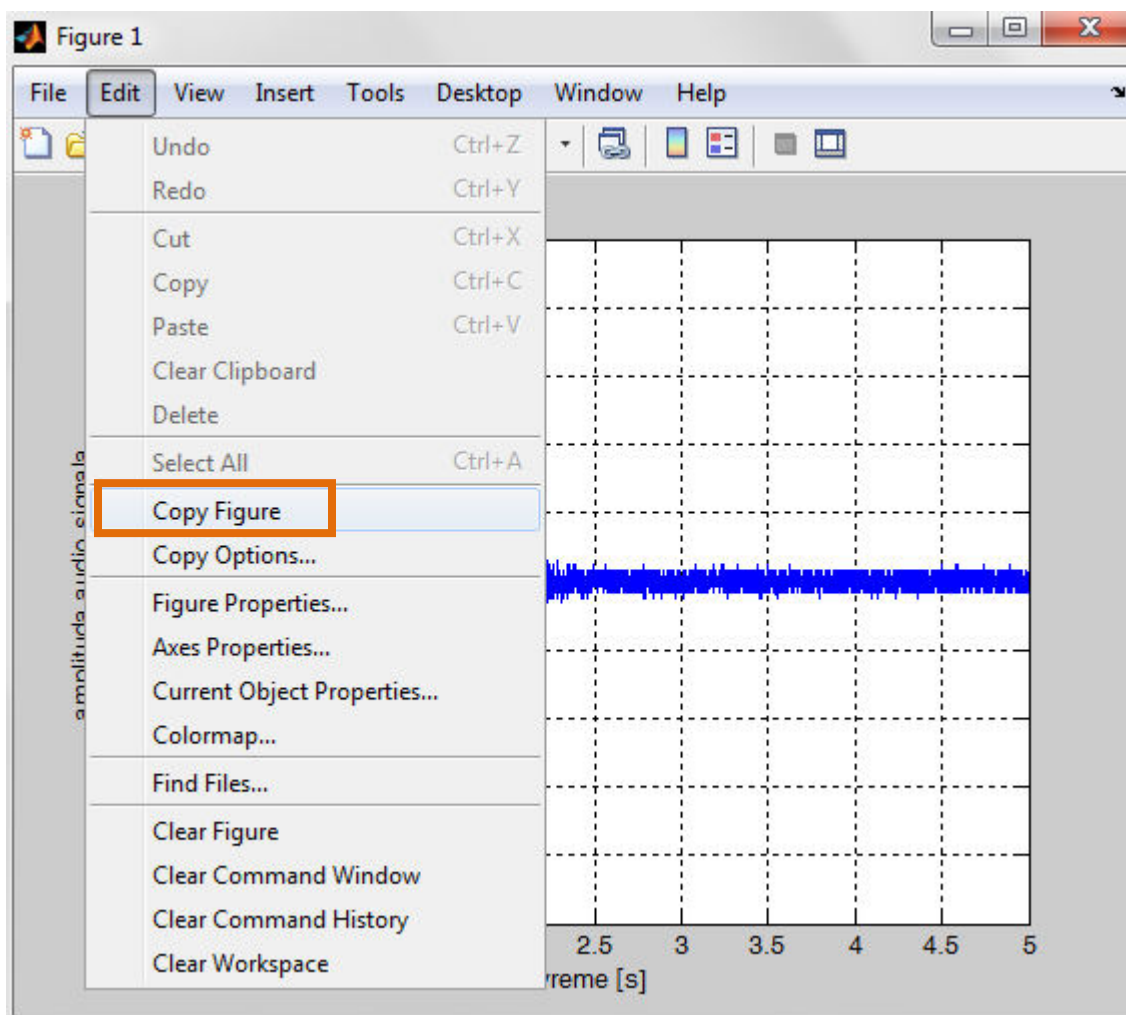
Odabirom opcije za prikaz dva kanala (1:2) umesto jednog (1) moguće je meriti u oba pravca kretanje računarskog miša na ravnoj površini.



Sl. 3, Odabir računarskog miša za merenje signala u Simulink programskom okruženju.

Dodatak A – kopiranje i čuvanje figura u Matlabu

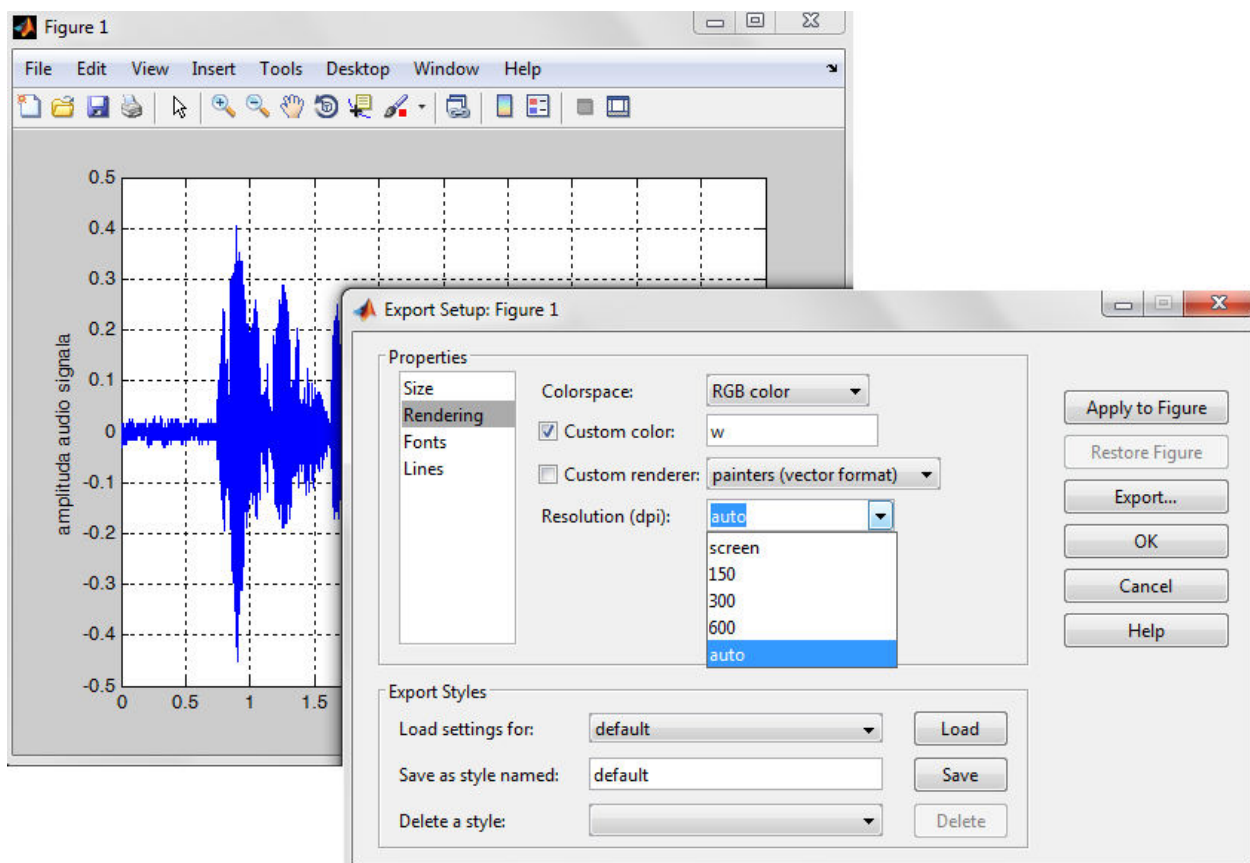
Za potrebe prikaza podataka i pisanja izveštaja, potrebno je na odgovarajući način prikazati podatke ili rezultate analize nekog signala. Kao primer, prikazan je način pisanja izveštaja u Word (Microsoft Office) dokumentu, ali slično se primenjuje i na ostale dokumente. Prvi način kopiranja grafika je odabirom opcije *Copy Figure* iz padajućeg menija *Edit Matlab figure* prozora (Sl. 10).



Sl. 104. Odabir opcije *Copy Figure* iz padajućeg menija *Edit*.

Nakon preslikavanja tj. kopiranja slike, u Word dokumentu je potrebno odabrati opciju *Paste* ili iz padajućeg menija ili prečicom sa tastature: istovremeno pritisnuti tastere CTRL (eng. *Control*) i V. Iz padajućeg menija *File* prozora *Figure* kao na Sl. 10, kada se odabere opcija *Save As*, moguće je definisati ekstenziju fajla odnosno tip fajla za snimanje signala .fig (od *Matlab figure*), .bmp (od *Bitmap file*), .jpg (od *JPEG image*) i druge. Nakon toga, potrebno je opcijom *Insert / Picture* (na kartici *Illustrations*) u Word-u odabrati odgovarajući fajl na računaru i uneti sliku u izveštaj.

Za Matlab grafike, podrazumevana rezolucija je automatska i najčešće je relativno mala¹⁹. Na jednostavan način, moguće je povećati rezoluciju Matlab grafika odabirom opcije *Export Setup* iz padajućeg menija *File* u *figure* prozoru.



Sl. 51, *Export Setup* prozor Matlab grafika.

Na kartici *Rendering*, u opciji *Resolution (dpi)* iz padajućeg menija odabrati na primer 600 dpi (eng. *dots per inch*, https://en.wikipedia.org/wiki/Dots_per_inch) i potvrditi izmenu (Sl. 11). Ako se grafik, nakon unošenja izmenjene rezolucije, kopira u izveštaj, onda će rezolucija biti veća i grafik kvalitetniji.

¹⁹ Potrebno je uzeti u obzir da je ova skripta realizovana za verziju Matlab R2013b, a da u novijim verzijama figure imaju veću inicijalnu rezoluciju.

Dodatak B - Matlab kod

Primeri merenja audio signala

Kod koji omogućava snimanje audio signala, prikaz signala, snimanje u fajl i čitanje iz fajla je:

```
close all;
clear all;
clc;

% definisanje objekta za merenje audio signala
% inicijalizacija hardware-a
recObj = audiorecorder;

% snimanje signala u trajanju od 5 s pomoću definisanog objekta
disp('Snimanje počinje.')
recordblocking(recObj, 5);
disp('Snimanje je završeno.');
```

```
% preslušavanje snimljenog signala
play(recObj);

% "prebacivanje" snimljenog audio signala u matricu pom
pom = getaudiodata(recObj);

% definisanje vremenske ose snimljenog signala
time = 0: 1/recObj.sampleRate: (length(pom)-1)./recObj.sampleRate;

% prikaz signala na grafiku
figure
plot(time, pom);
xlabel('vreme [s]');
ylabel('amplituda audio signala');
grid on;

% snimanje signala u fajl sa .wav ekstenzijom
audiowrite('mojSignal.wav', pom, recObj.sampleRate);

% Čitanje signala iz fajla sa .wav ekstenzijom
% [y, Fs] = wavread('mojSignal.wav');
```


Kod koji omogućava merenje audio signala za sekvencu koja se ponavlja onoliko puta koliko korisnik želi u trajanju od 5 s (dodato je i snimanje u fajl):

```
close all;
clear all;
clc;

% definisanje objekta za merenje audio signal
% inicijalizacija hardware-a
recObj = audiorecorder;
% može i recObj = audiorecorder(8000, 8, 1, -1);

% koliko puta je potrebno ponoviti merenje
br = input('Koliko puta želite da ponovite merenje? ');
name = input('Kako želite da se zove .wav fajl? ', 's');

% snimanje signala pomoću definisanog objekta
for ind = 1:br
    porukaStart = [num2str(ind) '. snimanje počinje...'];
    porukaStop = [num2str(ind) '. snimanje je završeno.'];
    disp(porukaStart)
    recordblocking(recObj, 5);
    disp(porukaStop);
    pom = getaudiodata(recObj);
    fileName = [name num2str(ind) '.wav'];
    audiowrite(fileName, pom, recObj.sampleRate);
    % clc; % opciono: da se tekst u komandnom prozoru obriše prilikom
    % snimanja u novi fajl
end
```

Merenje naponskih signala

Kod koji omogućava merenje jednog odbirka signala:

```
close all
clear all
clc

% provera uređaja koji su dostupni za merenje
daq.getVendors()

% kreiranje sesije
s = daq.createSession('ni');

% kreiranje kanala
s.addAnalogInputChannel('Dev1', 'ai0', 'Voltage');

% merenje jedne vrednosti signala
odbirak = s.inputSingleScan
```

Kod koji omogućava snimanje naponskog signala (*Foreground* metodom), prikaz signala, snimanje u fajl i čitanje iz fajla je:

```
close all;
clear all;
clc;

% provera uređaja i odabir uređaja na računaru (opciono)
% d = daq.getDevices;
% dev = d(1);

% kreiranje sesije za merenje sa određenim uređajem
s = daq.createSession('ni');

% definisanje kanala na uređaju
ch = addAnalogInputChannel(s, 'Dev1', 'ai0', 'Voltage');

% podešavanje parametara za merenje
s.Channels.TerminalConfig = 'SingleEnded'; % u odnosu na masu
s.NumberOfScans = 1000;
s.Rate = 1000;
ch.Range = [-5, 5];

% merenje i smeštanje podataka u promenljivu data
data = startForeground(s);

% definisanje vremenske ose
time = 0: 1/s.Rate: (length(data) - 1)./s.Rate;

% prikaz signala na grafiku
figure
plot(time, data);
xlabel('vreme [s]');
ylabel('amplituda signala [V]');
title('Prikaz signala merenog simuliranim NI uređajem');
grid on;

% snimanje signala u Excel fajl
xlswrite('izmerenSignal', data);

% čitanje signala iz Excel fajla
num = xlsread('izmerenSignal');
```

Literatura

1. Interacting with an Arduino using the Matlab serial commands, <https://www.mathworks.com/matlabcentral/fileexchange/26711-interacting-with-an-arduino-using-the-matlab-serial-commands>, pristupljeno aprila 2018.
2. Files and folders that Matlab accesses: Where does Matlab look for files?, https://www.mathworks.com/help/matlab/matlab_env/files-and-folders-that-matlab-accesses.html, pristupljeno aprila 2018.