

Pr. 9b Tvorba GUI v MATLABe pomocou App Designer

1. MATLAB App Designer – popis aplikácie

App Designer predstavuje interaktívne vývojové prostredie na navrhovanie aplikácie a programovanie jej správania. Poskytuje sadu interaktívnych komponentov používateľského rozhrania, ponúka možnosť automatického preformátovania, aby aplikácia reagovala na zmeny veľkosti obrazovky. Umožňuje distribuovať aplikácie ich zabalením do inštaláčnych súborov priamo z lišty nástrojov App Designer alebo vytvorením samostatnej pracovnej plochy alebo webovej aplikácie (vyžaduje kompilátor MATLAB™). Takýto softvér potom možno zdieľať na akomkoľvek počítači bez problémov s inštaláciou MATLABu alebo bez znalostí programovania na to, aby používateľ mohol ovládať softvér. Program zahŕňa širokú škálu štandardných grafických komponentov (prepínače, tabuľky, tlačidlá, začiarkavacie políčka, posúvače a mnoho ďalších).

Pre správne programovanie aplikácie treba dobre poznať vlastnosti komponentov, ktoré možno nájsť v lit. [8].

Po odštartovaní programu:

>> **appdesigner** alebo z menu: **APPS/Design App**

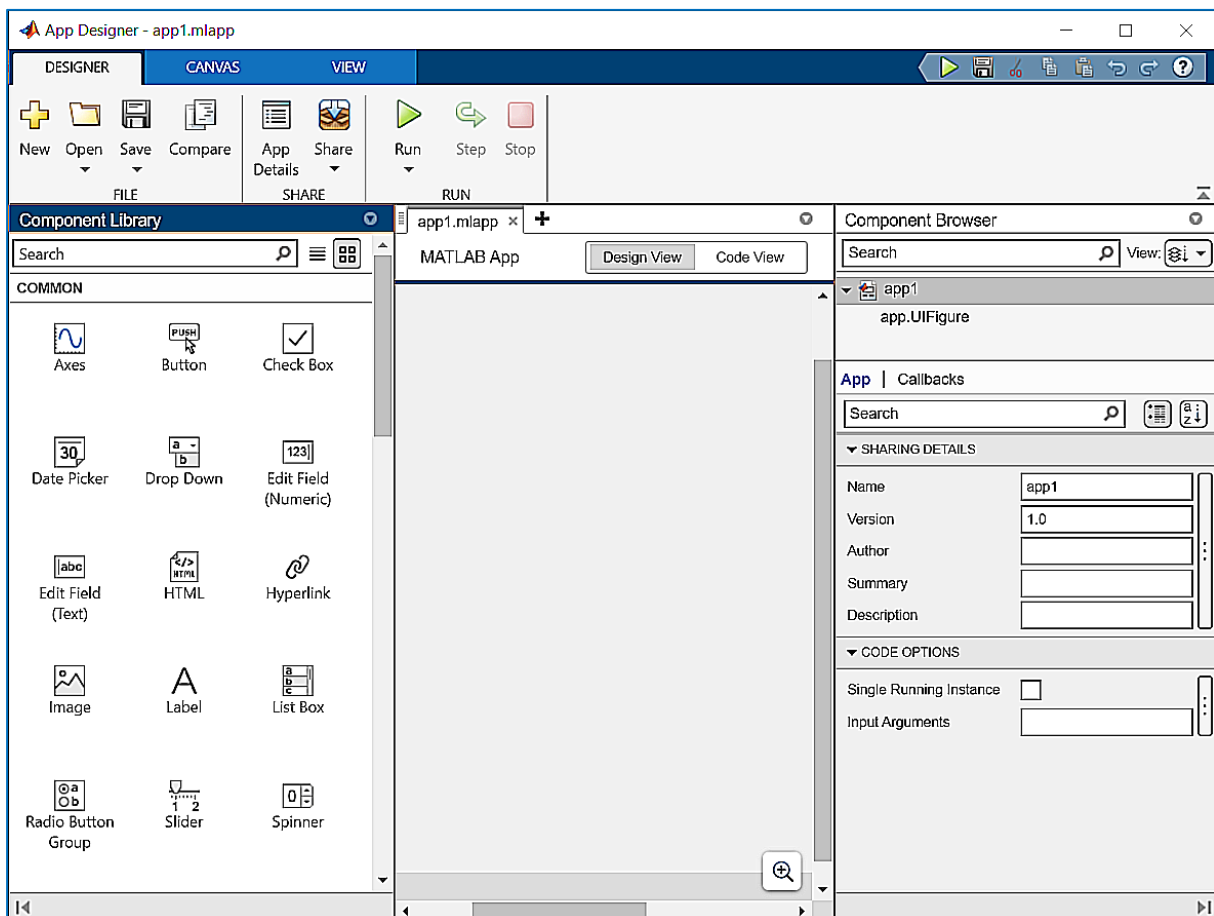
treba zvoliť **Blank App**:

Vzhľad a časti obrazovky

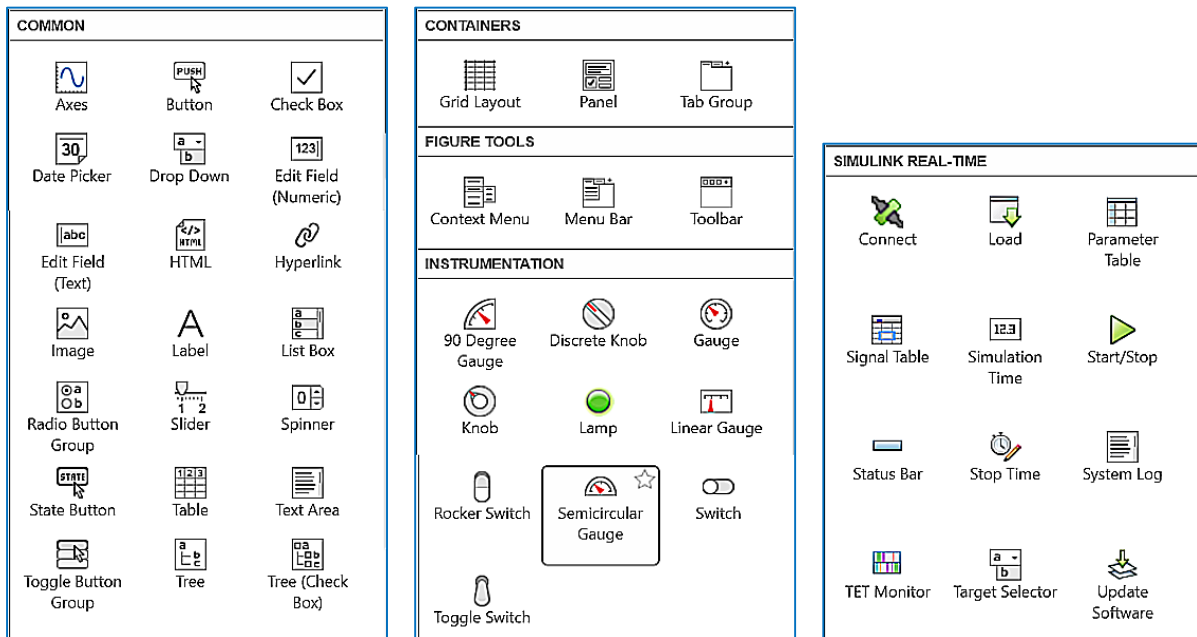
Knižnica komponentov
(Component Library)

Plocha pre uloženie komponentov
(Canvas)

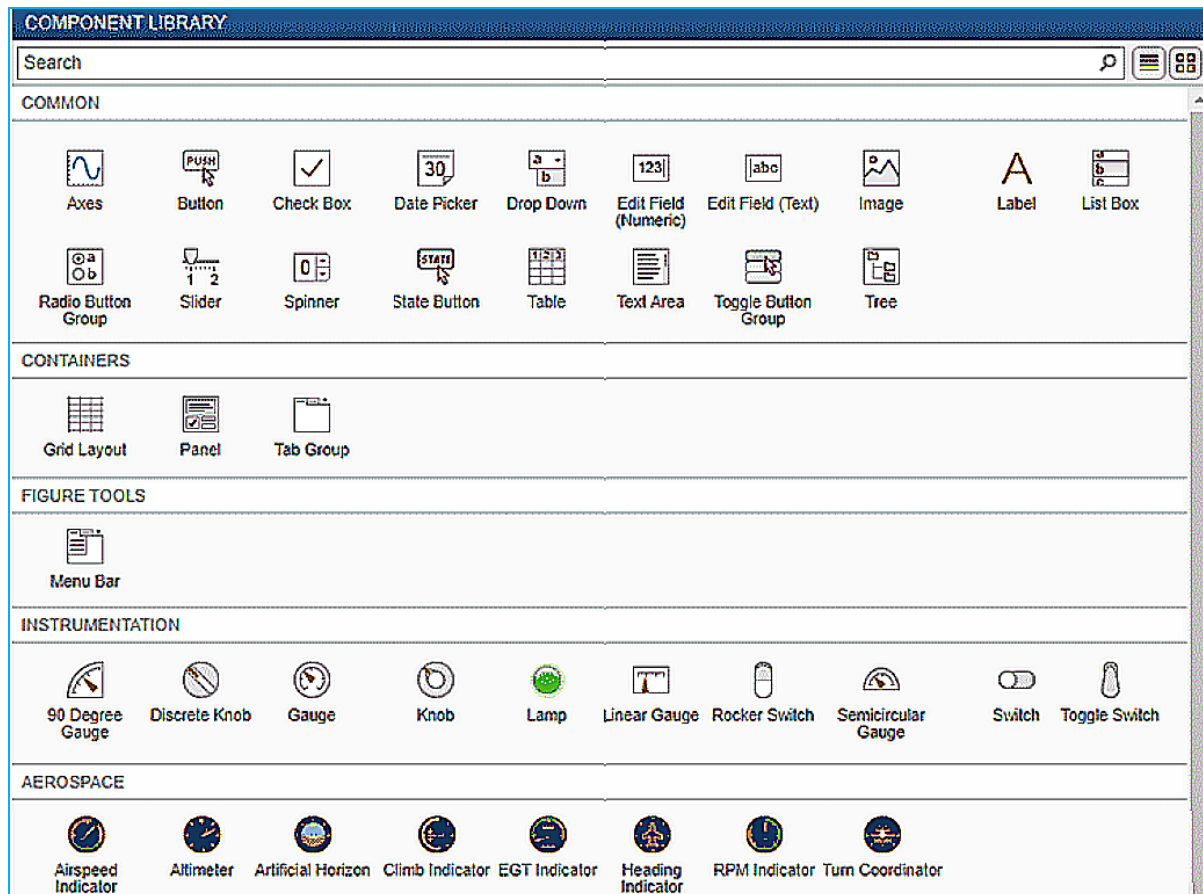
Prehliadač komponentov
(Component Browser)



2. Knižnica a programovanie komponentov GUI:



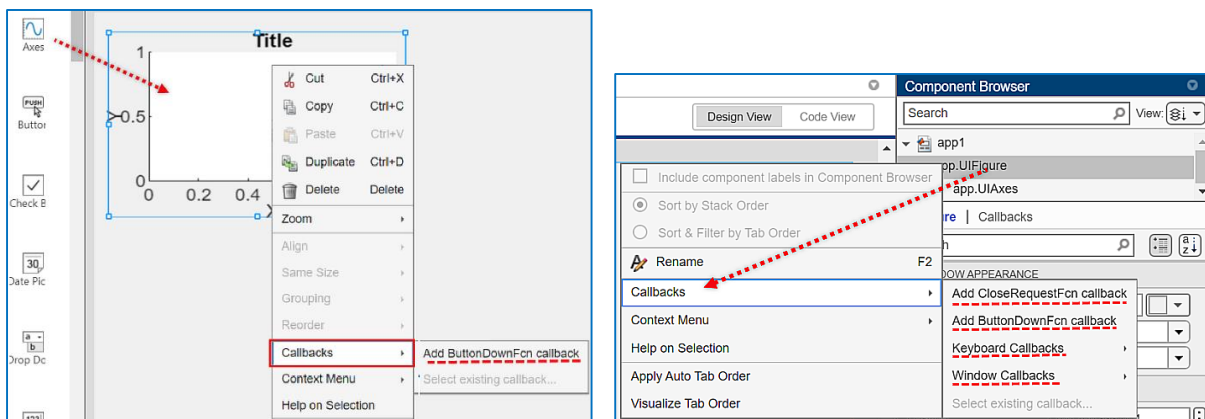
Roztiahnutím poľa Pavej časti (Component Library) dostávame prehľad všetkých komponentov.



3. Vytváranie spätných volaní, programovanie jednotlivých komponentov

Spätné volanie (callback) je funkcia, ktorá sa spustí, keď používateľ aktivuje komponent GUI. Pomocou spätných volaní možno naprogramovať správanie sa aplikácie. Možno napríklad napísať spätné volanie, ktoré vykreslí niektoré údaje, keď používateľ aplikácie klikne na tlačidlo, alebo spätné volanie, ktoré pohne ručičkou komponentu meradla, ak používateľ posúva posúvačom a pod.

Väčšina komponentov má aspoň jedno spätné volanie a každé spätné volanie je viazané na špecifickú interakciu s komponentom. Niektoré komponenty (labels – šítky, nápisy) však nemajú spätné volania, pretože zobrazujú iba informácie. Ak chcete zobrazit' zoznam spätných volaní, ktoré komponent podporuje, vyberte komponent a kliknite na kartu Spätné volania v Prehliadači komponentov (Component Browser) pravou myšou, zvolte Callbacks a ak sa tam nachádza viac spätných volaní, vyberte to, ktoré je vhodné pre vykonanie zvolenej operácie.



Naprogramovanie funkcie spätného volania

Po vytvorení spätného volania pre daný komponent App Designer vygeneruje funkciu spätného volania v zobrazení kódu a umiestni kurzor do tejto funkcie (biele riadky – tie sivé sa nedajú meniť). Napíšte kód do tejto funkcie spätného volania pre naprogramovanie správania spätného volania.

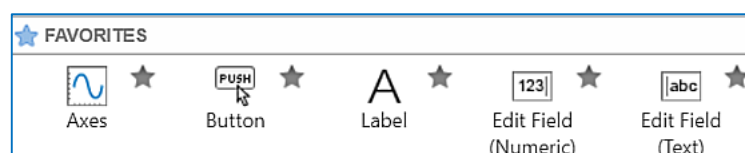
Vstupné argumenty spätného volania

Všetky funkcie spätného volania, ktoré vytvára App Designer, majú v popise funkcie tieto vstupné argumenty:

- **app** je objekt aplikácie, tento objekt sa používa na prístup ku komponentom GUI v aplikácii, ako aj k ďalším premenným uloženým ako vlastnosti.
- **event** (udalosť) je objekt, ktorý obsahuje špecifické informácie o interakcii používateľa aplikácie s komponentom používateľského rozhrania (napr. pri komponente slider - posúvač).

Viac informácií o programovaní callbackov je uvedených v lit. [6].

Pozn.: komponenty, ktoré budú použité v GUI možno vybrať na začiatku programovania GUI a to zaškrtnutím hviezdičky pri značke komponentu – tieto sa uložia pred všetkými ostatnými.



4. Programovanie GUI pre jednoduchú kalkulačku

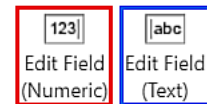
V tomto programe ide o programovanie komponentov **Edit Field**, **Label** a **Button**.

- 1) Odštartujeme >> **appdesigner** a uložíme ho do súboru **Kalkulacka.mlapp**
- 2) Rozmiestnime komponenty na ploche a súčasne ich označíme (Prvy, Druhy, Vysl, plus, ...). Použijeme komponenty **Button**, **EditField** a **Label**.

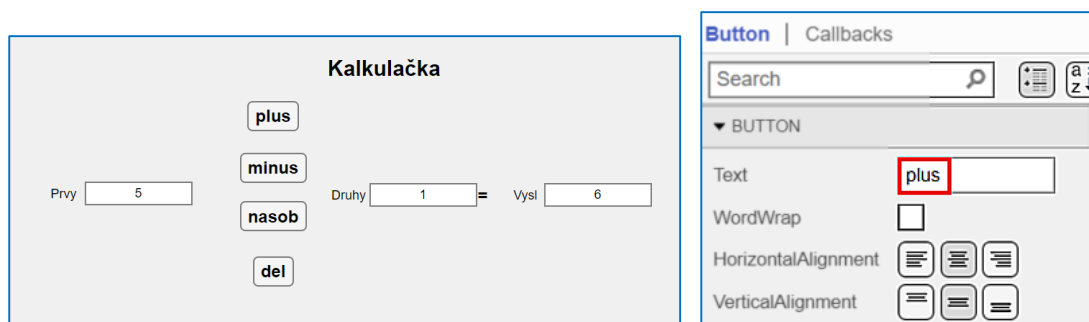
Pozor: Sú dva druhy komponentov **EditField**:

EditField Numeric – pre zadávanie čísiel, čím sa vyhneme potrebe konverzie reťazca na číslo a naopak (str2double, double2str)

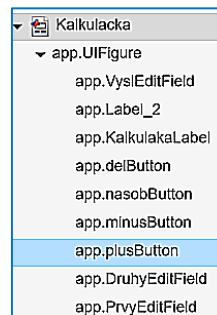
EditField Text (vstup reťazcov – textu..)



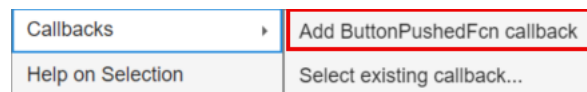
- 3) V jednotlivých komponentoch nastavíme vhodnú veľkosť písma (napr. 16 b.), jeho rez (bold, italics), horizontálne a vertikálne umiestnenie v danej oblasti



- 4) Tým sa vpravo hore, v Prehliadači komponentov (Component Browser), zobrazia názvy jednotlivých komponentov:



- 5) Zvolíme **callback pre plus** buď kliknutím pravou myšou na tlačidlo plus kliknutím alebo pravou myšou na **app.plusButton** a zvolíme operáciu:



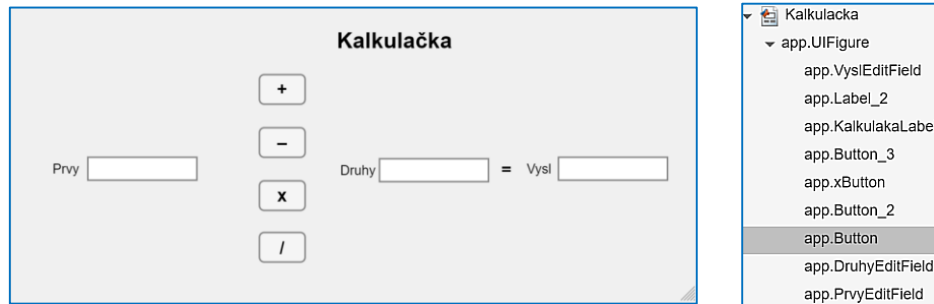
- 6) Doplníme text do časti **plusButton**. Do tlačidla sa zapisujú reťazce, ktoré musíme previesť na číslo **str2num**, vykonáme matematickú operáciu a výsledok sa zobrazí v poli **Vysl**:

```
% Button pushed function: plusButton
function plusButtonPushed(app, event)
    a = app.PrvyEditField.Value)
    b = app.DruhyEditField.Value)
    c = a+b
    app.VyslEditField.Value = c
end
```

- 7) Krok 3 a 4 opakujeme pre ostatné tlačidlá **minusButton**, **nasobButton**, **delButton**, zohľadňujúc príslušnú matematickú operáciu ($c=a-b$; $c=a*b$; $c=a/b$).
- 8) Súbor uložíme a aplikáciu odštartujeme tlačidlom **Run**.

Po zadaní čísiel do Edit boxov môžeme tlačiť jednotlivé tlačidlá matematických operácií a dostávame výsledok.

Na záver môžeme tlačidlá označiť matematickými symbolmi a MATLAB automaticky preznačí jednotlivé callbacks (nerobíme to na začiatku, lebo by sme mali problém pri rozlišovaní názvov (dostali by sme iba očíslované callbacks)).



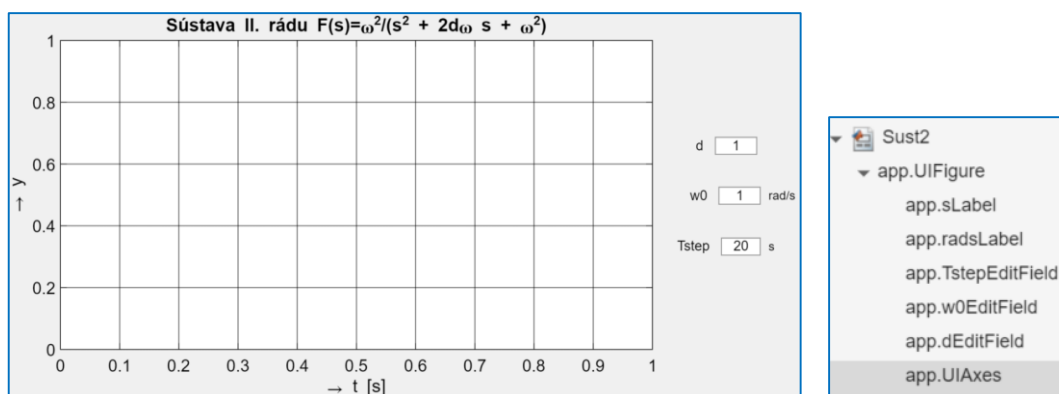
Pozn.: pri zostavovaní podprogramu pre ovládanie daného komponentu pri matematických operáciách na koniec riadka dávame bodkočiarku (ináč program hlási chybu).

5. Programovanie GUI pre sústavu II. rádu zadanú v štandardnom tvare

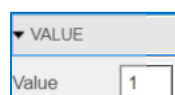
5.1 Vytvorenie grafu prechodovej charakteristiky

V tomto programe ide navyše o programovanie grafu (Axes). Zvolíme komponenty

Vzhľad GUI a zoznam vytvorených UI controls:



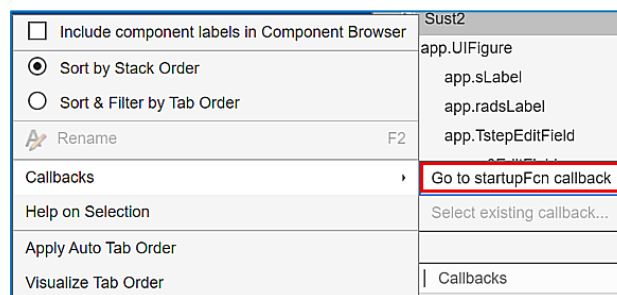
Pre **dEditField**, **w0EditField** a **TstepEditField** vytvoríme program, ktorý načíta tlačidlá, vytvorí prenosovú funkciu. Zadáme počiatkové hodnoty do jednotlivých edit Field – napr pre d:



Pre zobrazenie prechodovej charakteristiky vytvoríme vektor času **t** a vektor výstupnej veličiny **y** (pomocou funkcie **step**) a nakoniec výstup smerujeme na **app.UIAxes**.

```
% Value changed function: dEditField
function dEditFieldValueChanged(app, event)
    d = app.dEditField.Value;
    w0= app.w0EditField.Value;
    Tstep = app.TstepEditField.Value;
    % Prenosova funkcia
    num=1; den=[1/w0^2 2*d/w0 1];
    F=tf(num,den);
    % Zobrazenie priebeh step
    t=0:Tstep/1000:Tstep;
    y=step(F,t);
    plot(app.UIAxes,y)
end
```

Aby sme (v iných prípadoch, v tejto aplikácii to nie je nutné) graf mali k dispozícii už po odštartovaní aplikácie, zvolíme tiež callback pre Sust2 (celkom hore):

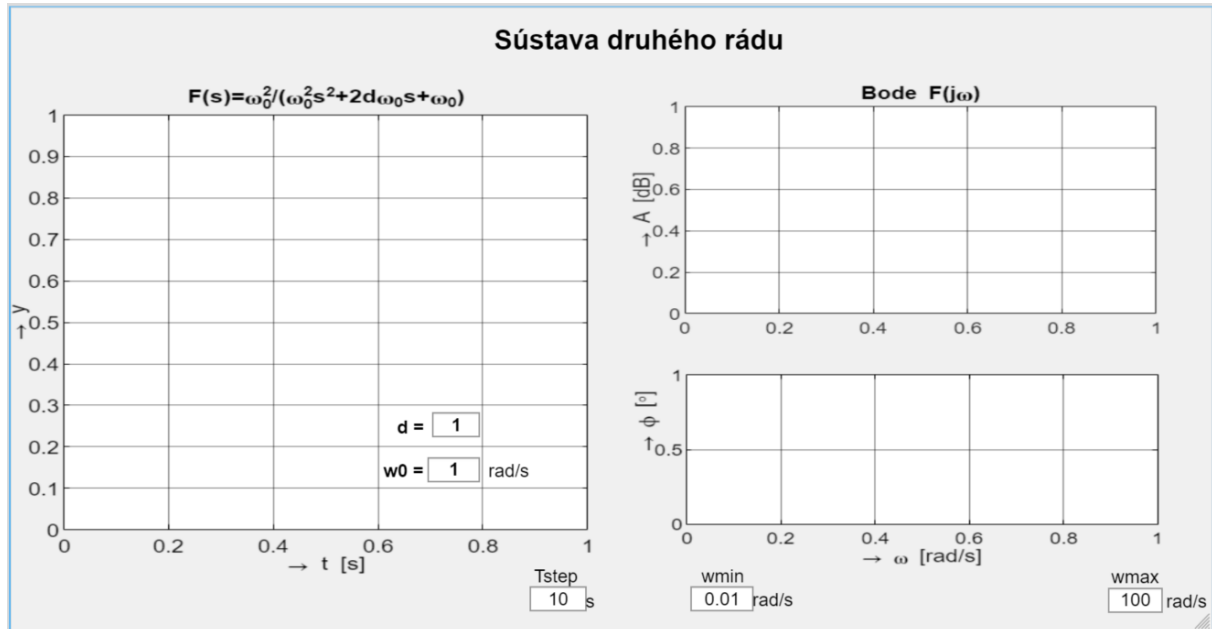


čím vytvoríme funkciu **function startupFcn(app)**, do ktorej skopírujeme obsah funkcie **dEditField**:

```
% Code that executes after component creation
function startupFcn(app)
    d=app.dEditField.Value;
    w0=app.w0EditField.Value;
    Tstep=app.TstepEditField.Value;
    % Prenosova funkcia
    num=1; den=[1/w0^2 2*d/w0 1];
    F=tf(num,den);
    % Zobrazenie priebehov step,bode
    t=0:Tstep/1000:Tstep;
    y=step(F,t);
    plot(app.UIAxes,y)
end
```

5.2 Vytvorenie grafu logaritmických frekvenčných charakteristík

Rozšírime pôvodné GUI vloženíím dvoch ďalších grafov (Axes) a tiež dvoma komponentami EditField – wmin a wmax pre výber pásma $< wmin; wmax >$ kruhovej frekvencie pre zvolenú časť LFCh, ktoré umiestnime na vhodné miesto pod grafy.



Podobným spôsobom ako pri výpočte PrCh vytvoríme súbory dát pre LFCh, t.j. vypočítame vektory hodnôt amplitúdy (magnitude – mag), fázy (phase) a odpovedajúcej kruhovej frekvencie wout. Aplikujeme teda výpočet **[mag, phase, wout] = bode(F)**

Na ľavej strane tejto rovnosti dostávame maticu s 3 stĺpcami. Pre výber príslušného stĺpca použijeme inštrukciu **squeeze**, t.j. **squeeze(mag)**, resp. **squeeze(phase)**.

Program pre jednotlivé EditField doplníme nasledovne:

```
% Graf pre bode
[mag, phase, wout] = bode(F, {wmin, wmax})
plot(app.BodeAmpUIAxes, 20*log10(squeeze(mag)))
plot(app.BodeFazaUIAxes, squeeze(phase))
```

Celý program každého callbacku EditField je nasledovný:

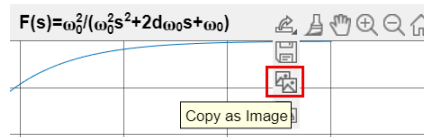
```
% Value changed function: dEditField
function wminEditFieldValueChanged(app, event)
    d = app.dEditField.Value;
    w0 = app.w0EditField.Value;
    Tstep=app.TstepEditField.Value;
    wmin =app.wminEditField.Value;
    wmax =app.wmaxEditField.Value;
% Prenosova funkcia
    num=w0^2; den=[1 2*d*w0 w0^2];
    F=tf(num,den);
% Graf pre step
    t=0:Tstep/1000:Tstep;
    y=step(F,t);
    plot(app.StepUIAxes,y);
```

% Graf pre bode

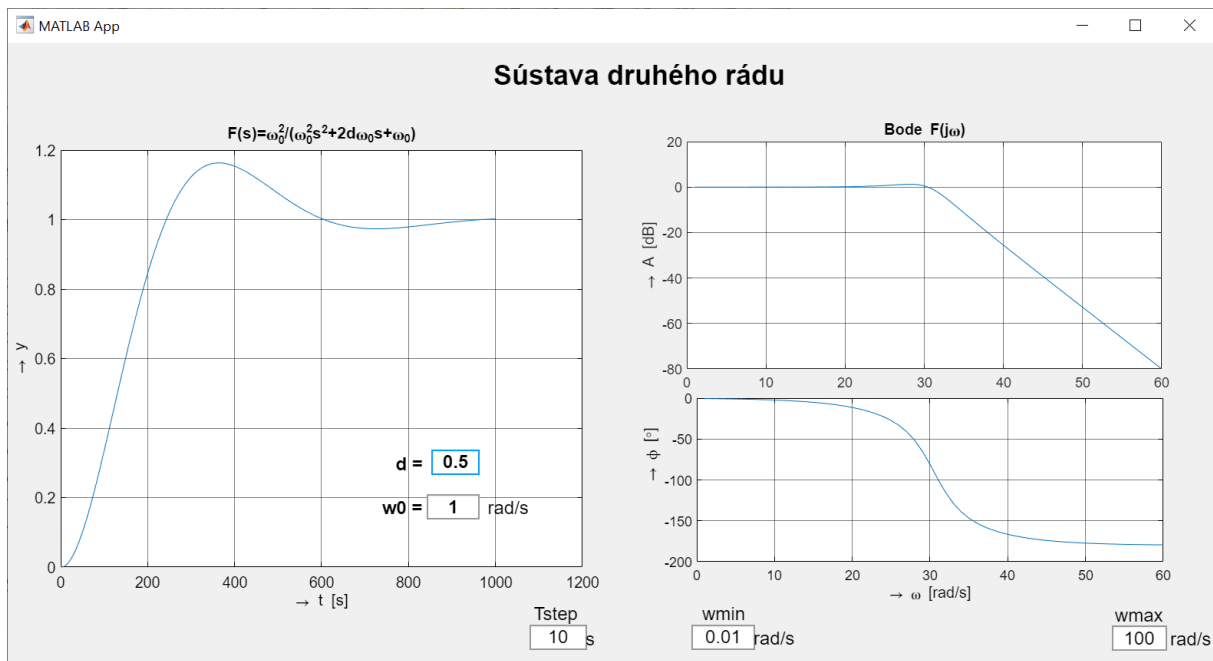
```
[mag,phase,wout] = bode(F,{wmin,wmax})
plot(app.BodeAmpUIAxes,20*log10(squeeze(mag)))
plot(app.BodeFazaUIAxes,squeeze(phase))
```

end

Každý graf vieme uložiť ako obrázok .png (zvolením vhodnej ikony na vrchu grafu v pravom hornom rohu), ktorý môžeme uložiť a potom vložiť do Wordu (oboznámte sa aj s ostatnými ikonami!).



Celkové GUI však musíme vložiť do medzipamäte pomocou tlačidla PrtScn.



Nedostatky – tvar PrCh a LFCh zodpovedá skutočnosti, avšak v oboch prípadoch nesúhlasia mierky na vodorovných osiach.

Ďalšie vlastnosti UIAxes možno vyhľadať v nápovede MATLABu (>> **doc UIAxes Properties**) alebo v lit. [6].

5.3 Zobrazenie výstupu znakov reťazcov (prenosová funkcia, nuly a póly prenosu)

Vytvorené GUI doplníme

- výpočtom núl **zeros=roots (num) ;**
- výpočtom pólov prenosovej funkcie **poles=roots (den) ;**
- a zobrazíme pomocou EditField (typu Numeric)

Prenosovú funkciu vytvoríme tak, že:

- pomocou label napíšeme $F(s) = \frac{\text{num}(s)}{\text{den}(s)}$ (zlomková čiara, nad ktorou a pod ktorou budú komponenty EditField (Text)
- zostavíme reťazec čitateľa: **[num2str (num (1))]**
- a reťazec menovateľa?
[num2str (den (1)) , 's^2+' , num2str (den (2)) , 's+' , num2str (den (3))]
- a vložíme do odpovedajúcich komponentov EditField (typu Text).

5.4 Doplnenie GUI zobrazením fázového posunu medzi vstupným a výstupným signálom pri harmonickom napájaní

Vložíme graf so sinusovkami generovanými pomocou gensig a lsim. Príslušnú frekvenciu načítame z EditText a podobne aj počet zvolených periód, príp. aj časy od – do.

5.5 Doplnenie GUI obrázkom schémy obvodu

Použi komponent Image. Použitie je jednoduché.

5.6 Vytvorenie samospustiteľnej aplikácie

Pomocou prekladača MATLAB Compiler možno vytvoriť samospustiteľnú aplikáciu na počítači, alebo na webe, ktorú možno spustiť aj na počítači, na ktorom nie je nainštalovaný program MATLAB. Postup a viac informácií je uvedených v [15 – 17].

Upozornenie! Pre úplný GUI, rovnocenný s GUI programovaným pomocou vývojového prostredia GUIDE je potrebné ešte vyriešiť nasledovné problémy:

1. Graf pre step: časovú os nezodpovedá skutočnosti – aplikáciou uvedeného zobrazenia takto možno získať iba tvar prechodovej charakteristiky.
2. Graf pre bode: os kruhovej frekvencie nezodpovedá skutočnosti.
3. Zmena hrúbky čiar priebehov.
4. Zobrazenie prenosovej funkcie (čitateľa a menovateľa).
5. Výpočet núl a pólov a ich zobrazenie.
6. Doplnenie ďalšieho grafu pre zobrazenie priebehu vstupného a výstupného signálu pri harmonickom napájaní pri zvolenej kruhovej frekvencii.
7. Vloženie obrázka so schémou obvodu.

6. Domáca úloha

Vytvoriť GUI pomocou APP Designer pre svoje zadanie. Editovacie polia majú obsahovať parametre elektrického obvodu. Korektný postup riešenia pri výpočte TF pomocou symbolického MATLABu spočíva v uložení programu so symbolickými výpočtami do externého súboru, ktorý voláme menom do **function startupFcn(app)**. Tu sa pred odštartovaním vlastnej aplikácie najprv vypočítajú koeficienty čitateľa a menovateľa prenosovej funkcie. Zjednodušený postup je, ak do GUI sa uložia iba jednotlivé koeficienty prenosovej funkcie (v symbolickom tvare) vypočítané pomocou symbolického MATLABu.

Odporúčaná literatúra

- [1] App Design in MATLAB
<https://www.youtube.com/watch?v=79tVxKArf7E>
- [2] How to Make an Interactive App in MATLAB,
<https://www.youtube.com/watch?v=SpAp7QACF34>
- [3] Getting Started with App Designer
<https://www.mathworks.com/videos/app-designer-overview-1510748719083.html>
- [4] App Designer. Create desktop and web apps in MATLAB
<https://nl.mathworks.com/products/matlab/app-designer.html>
- [5] Develop Apps Using App Designer
https://nl.mathworks.com/help/releases/R2022a/matlab/app-designer.html?s_tid=CRUX_lftnav
- [6] Callbacks in App Designer
https://nl.mathworks.com/help/matlab/creating_guis/write-callbacks-for-gui-in-app-designer.html
- [7] Display Graphics in App Designer
https://nl.mathworks.com/help/matlab/creating_guis/graphics-support-in-app-designer.html#responsive_offcanvas
- [8] UIAxes Properties
<https://nl.mathworks.com/help/matlab/ref/matlab.ui.control.uiaxes-properties.html>
- [9] Develop Apps Using App Designer — Examples
https://nl.mathworks.com/help/matlab/examples.html?category=app-designer&s_tid=CRUX_app-designer
- [10] MATLAB AppDesigner Beginner tutorials (1 – 20)
https://www.youtube.com/watch?v=iga-YS6VbyE&list=PLDzzooPYRJ_XNBukbCkOM-TBWFfeO5Ud4I
- [11] Migrate GUIDE Apps — Videos
https://nl.mathworks.com/help/matlab/creating_guis/guide-getting-started.html
- [12] GUIDE do APP Designer Migration Tool for MATLAB
https://nl.mathworks.com/videos/app-designer-overview-1510748719083.html?s_tid=pi_app_designer_R2019b_guide_banner_video

- [13] How to install 'Standalone Desktop App' in app designer
<https://nl.mathworks.com/matlabcentral/answers/517513-how-to-install-standalone-desktop-app-in-app-designer>
- [14] Teaching System Dynamics with MATLAB & Simulink
<https://nl.mathworks.com/videos/teaching-system-dynamics-with-matlab-simulink-81420.html>
- [15] Getting Started Standalone Applications using MATLAB Compiler
<https://www.youtube.com/watch?v=14FOQqCSGIM>
- [16] Simple Mortgage Calculator Web App
<https://nl.mathworks.com/help/compiler/webapps/simple-mortgage-calculator-web-app.html>
- [17] MATLAB Compiler
<https://nl.mathworks.com/help/compiler/>
- [18] Complete MATLAB App Design Tutorial
https://www.youtube.com/watch?v=cl0AcnN3Bmk&list=RDLVcl0AcnN3Bmk&start_radio=1&rv=cl0AcnN3Bmk&t=2216
a ďalšie, najvhodnejšie na YouTube.