CV. 10 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 inštalačných súborov priamo z lišty nástrojov App Designer alebo vytvorením samostatnej pracovnej plochy alebo webovej aplikácie (vyžaduje kompilátor MATLABTM). Takýto softvér potom možno zdieľať na akomkoľvek počítači bez problémov s inštaláciou MATLABu alebo bez znalostí programovania na to, aby používateľ mohol ovládať softvér. P
- rogram 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, alebo Home/New/A zvoľ Blank App:

Vzhľad a časti obrazovky

Knižnica komponentovPlocha pre uloženie komponentovPrehliadač komponentov(Component Library)(Canvas)(Component Browser)

📣 App Designer -	app1.mlapp									-		×
DESIGNER	CANVAS	VIEW								4 唱 唱	5 ¢	?
New Open Save	Compare	App Share Details SHARE	Run	Step Stop								Ā
Component Libra	ry) 🛛 🔤 ar	op1.mlapp × 🕂				0	Component Browser			0
Search		פ ≣	N	IATLAB App	Desigr	View	Code View		Search		P View	: 🕃 🔻
	Rised Second							•	✓ app1 app.UIFigure			
Axes	Button	Check Box							App Callbacks			
									Search		6 (a: z↓
30	a - b	123							▼ SHARING DETAILS			
Date Picker	Drop Down	Edit Field (Numeric)							Name	app1		
		(Version	1.0		
abc		N							Author			!!
Edit Field	HTML	Hyperlink							Summary			
(Text)									Description			
	^	a iil							▼ CODE OPTIONS			
imaga i	A	List Pox							Single Running Instance			
intage	Laber	LIST DOX							Input Arguments			Ľ
ा ठिमे Radio Button Group	∏ 1 2 Slider	이는 Spinner					æ					
14			▼				٩	/ - }				ÞI

2. Knižnica a programovanie komponentov GUI



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

COMPONENT	LIBRARY				çanınınının ere		nassannanassa:	vananaran	
Search									89 🗐 🤇
COMMON									<u>م</u>
Axes	िम्हम् दूरे Button	Check Box	30 Date Picker	a • b Drop Down	123 Edit Field (Numeric)	abc Edit Field (Text)	image	A Label	List Box
©a Ob Radio Button Group	1 2 Slider	0 -	State Button	Table	Text Area	Toggle Button Group	E Tree		
CONTAINERS									
Grid Layout	Panel	Tab Group							
FIGURE TOOLS	5								
Menu Bar									
INSTRUMENTA	TION								
90 Degree Gauge	Discrete Knob	Gauge	(D) Knob	Eamp	Linear Gauge	Rocker Switch	Semicircular Gauge	C) Switch	O Toggle Switch
AEROSPACE									
Airspeed Indicator	Altimeter	Artificial Horizon	Climb Indicator	EGT Indicator	Heading Indicator	RPM Indicator	Coordinator		E

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 <u>aspoň jedno spätné volanie</u> a každé <u>spätné volanie je viazané na</u> <u>špecifickú interakciu s komponentom</u>.

Niektoré komponenty (**Labels** – šítky, nápisy) však nemajú spätné volania, pretože **zobrazujú iba** informácie.

Ak chceme zobraziť zoznam spätných volaní, ktoré komponent podporuje, vyberime komponent a kliknite na kartu Spätné volania v Prehliadači komponentov (Component Browser) pravou myšou, zvoľte Callbacks, Ak ja tam viac spätných volaní, vyberte to, ktoré je vhodné pre vykonanie zvolenej operácie.



Prepínanie Návrh/Kód pre zobrazenie návrhu, alebo programu callbacku priamo z plochy pre uloženie komponentov (Canvas):



Naprogramovanie funkcie callbacku (spätného volania)

Po vytvorení spätného volania pre daný komponent **App Designer vygeneruje funkciu** spätného volania v zobrazení kódu a <u>umiestni kurzor do tejto funkcie</u> (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 <u>prístup ku komponentom GUI v aplikácii,</u> <u>ako aj k ďalším premenným uloženým ako vlastnosti</u>.
- event (udalosť) je objekt, ktorý <u>obsahuje špecifické informácie o interakcii používateľa</u>: aplikácie s komponentom používateľského rozhrania (napr. pri komponente slider posúvač).

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

Pozn.: komponenty, ktoré sa často používajú pre tvorbu 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
- Rozmiestnime komponenty na ploche a súčasne ich označíme (Prvy, Druhy, Vysl, plus, ...). Použijeme komponenty Button, EditField a Label.
 Bozor: Sú dva druhy komponentov EditField:

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 ret'azcov – textu..

123	abc
Edit Field	Edit Field
(Numeric)	(Text)

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

	Kalkulačka		Button Callbacks	\$
nlus		Search	₽ : a:	
	minus Druhy 1 = Vysl 6		▼ BUTTON	
Prvy 5		Vysl 6	Text	plus
			WordWrap	
	del		HorizontalAlignment	
			VerticalAlignment	

- 4) Tým sa vpravo hore, v Prehliadači komponentov (**Component Browser**), zobrazia názvy jednotlivých komponentov:
 - Kalkulacka
 app.UIFigure

 app.VyslEditField
 app.Label_2
 app.KalkulakaLabel
 app.delButton
 app.minusButton
 app.lusButton
 app.DruhyEditField
 app.PrvyEditField
- 5) Zvolíme **callback pre plus** buď kliknutím pravou myšou na tlačidlo plus kliknutím alebo pravou myšou na **app.plusButton** a zvolíme operáciu:

Callbacks >	Add ButtonPushedFcn callback
Help on Selection	Select existing callback

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.VysledokEditField.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 aplikáciou 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 callbacky (<u>nerobíme to na začiatku</u>, lebo by sme mali problém pri rozlišovaní názvov (dostali by sme iba očíslované callbacky).



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

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

5.1 Vytvorenie grafu prechodovej charakteristiky

V tomto programe ide naviac 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 počiatočné hodnoty do jednotlivých **edit Field** – napr. pre d:

▼ VALUE	
Value	1

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 mali graf k dispozícii už po odštartovaní aplikácie, zvolíme tiež calback pre Sust2

Sustava2radu
 app.UIFigure
 app.TstepEditField
 app.w0EditField
 app.dEditField
 app.UIAxes

Include component labels in Component Browser	Sust2			
	app.UIFigure			
 Sort by Stack Order 	app.sLabel			
O Sort & Filter by Tab Order	app.radsLabel			
A Rename F2	app.TstepEditField			
Callbacks >	Go to startupFcn callback			
Help on Selection	Select existing callback			
Apply Auto Tab Order				
Visualize Tab Order	Callbacks			

čím vytvoríme funkciu **function startupFcn(app)**, do ktorej <u>skopírujeme obsah</u> 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
```

Pozn. pre postup pri vkladaní grafu výsledkov zo schémy v Simulinku pozri video [11],

5.2 Vytvorenie grafu logaritmických frekvenčných charakteristík

Rozšírime pôvodné GUI vložením 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 l'avej 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!

	$F(s)=\omega_0^2/(\omega_0^2s)$	²+2d _{@0} s+ _{@0})	e, j	⊕£Q6;
1		Copy a	as Image	

Celkové GUI však musíme vložiť do medzipamäte pomocou tlačidla PrtScn.



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

Ďalšie vlastnosti UIAxes možno vyhľadať MATLABe v nápovede: >> doc UIAxes Properties alebo v lit. [6].

5.3 Vloženie výstupu do grafu zo schémy v Simulinku

Úlohou je zostaviť jednoduchú schému v Simulinku – generátor sínusového signálu a výstup cez blok To Workspace do grafu na ploche GUI (pre detaily pozri [11]). Zvolíme zadávanie doby riešenia a amplitúdy pomocou posúvača (možno nastavovať aj frekvenciu napr. pomocou ďalšieho posúvača, či EditField – tu nerealizované).

- 1) Prenesieme vhodné komponenty Axes, Slider, Button
- 2) Nastavíme základné parametre (popisy, veľkosti a pod.) napr. v tvare:



 V MATLABe odštartujeme Simulink (cez: >> simulink alebo Home/Simulink, zvolíme Blank Model). Vytvoríme jednoduchú schému:



Schému uložíme pod vhodným menom – tu SineVawe_S.slx

V blokoch nastavíme:

- SineWave: amplitúdu Ampl
- To Workspace: výstupnú premennú napr. SineWaveValue
- nezabudnúť v Simulinku nastaviť vhodnú toleranciu presnosti výpočtu (Ctrl + E)
- 4) V tlačidle Start zapíšeme klikom na Calbacks meno callbacku:

app.StartButton					
app.UIAxes					
✓ app.Conte	extMenu				
app.M	enu				
app.M	enu2				
Button Callbaci	ks				
ButtonPushedFcn StartButtonPushed 💌 🖨					
<no callback=""></no>					
StartButtonPushed					

- 5) Zapíšeme program, ktorý sa má vykonať:
 - (1) nastavenie amplitúdy podľa hodnoty nastavenej na posúvačoch Ampl a Time
 - pozri doc assignin (syntax <u>assignin(ws,var,val)</u>)
 - (2) simulácia modelu doba riešenia a StopTime (+ prevod hodnoty reťazca zo str to num)
 - (3) vykreslenie priebehu: nasmerovanie kam app.UIAxes a čo:
 SineWaveValue.Time simout.SineWaveValue.Data).
 Možno nastaviť aj parametre čiary: typ, hrúbku farbu.
 - Odporúčanie: najprv si vyskúšať kreslenie v MATLABe:

plot (out.SineWaveValue.Time, out.SineWaveValue.Data) Konkrétne:

```
% Button pushed function: StartButton
function StartButtonPushed(app,event)
    assignin('base','Ampl',app.AmplSlider.Value);
    assignin('base','StopTime',app.TimeSlider.Value);
    simout = sim('SineWave_S, 'StopTime', num2str(app.TimeSlider.Value));
    plot(app.UIAxes, simout.SineWaveValue.Time, simout.SineWaveValue.Data);
end
```

Pozor: Treba nastaviť presnosť výpočtu v Simulinku (obmedziť veľkosť kroku (nie je integrátor). Aby sa zmeny na posúvači okamžite prejavili v grafe, treba pridať funkciu tohto callbacku na posuvnú lištu. V tejto funkcii spätného volania sa tento program prepíše tiež do Axes. Tým sa graf okamžite obnoví pri akejkoľvek zmene zadanej hodnoty parametra.

5.4 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 zobrazíme tak, že:

- pomocou komponentu Label napíšeme F(s) = ----- je to zlomková čiara vhodnej dĺžky, nad ktorou (polynóm čitateľa vo forme reťazca) a pod ktorou (polynóm menovateľa vo forme reťazca) budú komponenty EditField (Text)
- zostavíme reťazec čitateľa: [num2str(num(1))]
- a ret'azec menovatel'a?
 [num2str(den(1)), 's^2+', num2str(den(2)), 's+', num2str(den(3))]
- a vložíme do odpovedajúcich komponentov EditField (typu Text).

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

Vložíme graf so sínusovkami generovanými napr. pomocou **gensig** a **lsim** (príp. je to možné generovať aj zo Simulinku). <u>Príslušnú frekvenciu</u> načítame z **EdiťTex**t a podobne aj <u>počet</u> zvolených periód, príp. aj časy od – do.

5.6 Doplnenie GUI obrázkom schémy obvodu

Použite komponent Image. Použitie je jednoduché. Obrázok, ktorý sa má zobraziť na ploche, musí byť v tom istom adresári.

5.7 Vytvorenie samospustiteľ nej aplikácie

Pomocou prekladača **MATAB 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].

6. Domáca úloha

Vytvorit' GUI pomocou APP Designer pre svoje zadanie:

- Editovacie polia majú obsahovať parametre 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 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/appdesigner.html?s_tid=CRUX_lftnay_
- [6] **Callbacks in App Designer,** <u>https://nl.mathworks.com/help/matlab/creating_guis/write-callbacks-for-gui-in-app-designer.html</u>
- [7] **Display Graphics in App Designer,** <u>https://nl.mathworks.com/help/matlab/creating_guis/graphics-support-in-app-designer.html#responsive_offcanvas</u>
- [8] UIAxes Properties, https://nl.mathworks.com/help/matlab/ref/matlab.ui.control.uiaxes-properties.html
- [9] Develop Apps Using App Designer Examples, <u>https://nl.mathworks.com/help/matlab/examples.html?category=app-designer&s_tid=CRUX_app-designer</u>
- [10] MATLAB AppDesigner Beginer tutorials (1 20) 20 vidíe, <u>https://www.youtube.com/watch?v=iga-YS6VbyE&list=PLDzzooPYRJ_XNBUkbCkOM-TBWFeO5Ud4I</u>
- [11] How to simulate a model through App Designer, set parameters and plot outputs in App Designer?, <u>https://www.youtube.com/watch?v=ThS6xllOy9Y</u> video zo série MATLAB App Designer Tutorials project videos: <u>https://www.youtube.com/playlist?list=PLFfOLCbwIxbCnCcfK-vjXSYao3DZJ66eR</u>
- [12] Complete MATLAB App Design Tutorial, https://www.youtube.com/watch?v=cl0AcnN3Bmk&t=2254s
- [13] Migrate GUIDE Apps Videos, https://nl.mathworks.com/help/matlab/creating_guis/guide-getting-started.html
- [14] GUIDE do APP Desginer Migration Tool for MATLAB, <u>https://nl.mathworks.com/videos/app-designer-overview-</u> <u>1510748719083.html?s_tid=pi_app_designer_R2019b_guide_banner_video</u>
- [15] How to install 'Standalone Desktop App' in app designer, <u>https://nl.mathworks.com/matlabcentral/answers/517513-how-to-install-standalone-desktop-app-in-app-designer</u>

[16]	Getting Started Standalone Applications	18	sing MA'	LAB Compiler,
	https://www.youtube.com/watch?v=14FOC	20	qCSGlM	

- [17] Simple Mortgage Calculator Web App, https://nl.mathworks.com/help/compiler/webapps/simple-mortgage-calculator-webapp.html
- [18] MATLAB Compiler, https://nl.mathworks.com/help/compiler/

[19] Complete MATLAB App Design Tutorial, <u>https://www.youtube.com/watch?v=cl0AcnN3Bmk&list=RDLVcl0AcnN3Bmk&start_radio=1&rv=cl0AcnN3Bmk&t=2216</u> zo série MATLAB TutorialsMATLAB Tutorials (10 videî), <u>https://www.youtube.com/playlist?list=PLYmlEoSHldN4GZ6SrT1ckBTzAwpTFdshr</u>

a d'alšie, najvhodnejšie na YouTube.