# Kódgenerálás és GUI fejlesztés Matlab-Simulink környezetben

**A mérés célja:** A Simulink környezet egyes kódgenerálási szolgáltatásainak megismerése és elsajátítása, továbbá grafikus felhasználói felület létrehozása Matlab alatt, a GUIDE eszköz megismerése.

**Felhasznált eszközök:** (IK Cloud architektúra – <u>http://cloud.ik.bme.hu</u>), Matlab R2012a, Simulink, GUIDE, Matlab Compiler.

#### Rendelkezésre álló, letölthető állományok:

- Mérési útmutató (jelen leírás)
- <u>Holtidos szakasz.jpg</u> (szakasz működését szemléltető ábra)

## 1. feladat – Egyszerű alkalmazás grafikus felülettel

A feladat grafikus felhasználói felület (GUI) létrehozása, amely támogatja, hogy egy holtidős, egytárolós szakaszhoz PI szabályzóméretezést végezzünk<sup>1</sup>. A PI szabályzót egy fűtött folyadékot keringető berendezés (például dialízisgép) hőmérséklet szabályozójához kell megtervezni, amelynek sémája az ábrán látható.



A szakasz viselkedése egy adott hőmérséklet, mint munkaponti érték környékén jól jellemezhető egy átviteli függvénnyel:  $W(s) = \frac{A}{1+sT} \exp(-sT_h)$ . Ehhez az átviteli függvényhez tervezünk egy  $W_{PI}(s) = \frac{A_p}{T_i} \frac{1+sT_i}{s}$  átvitelű PI szabályzót.

Megoldás: A fejlesztést a GUIDE eszköz segítségével végezzük, amelyet parancssorból indíthatunk a guide utasítással.

GUIDE Quick Start Create New GUI Open Existing	g GUI
GUIDE templates	Preview
<ul> <li>Blank GUI (Default)</li> <li>GUI with Uicontrols</li> <li>GUI with Axes and Menu</li> <li>Modal Question Dialog</li> </ul>	BLANK
Save new figure as: C:\U	Jsers\Kiss Bálint\Documents\Oktatas\Gy Browse
	OK Cancel Help

<sup>&</sup>lt;sup>1</sup> PI szabályozó méretezése a Rendszerelmélet (VIHVAB00) tárgyból szerepelt

A későbbiekben mintákkal is dolgozhatunk, most egy üres felülettel (Blank GUI) indítsunk. A felhasználói felület mentésekor két állomány jön létre. Feltéve, hogy az ún. PI\_designer elnevezést adjuk a Matlab alatti alkalmazásnak, ezek az alábbiak:

- 2. PI\_designer.fig a GUI ablak Matlab alatti megjelenése
- 3. PI\_designer.m a GUI mögötti kód, amely részben generált, részben a felhasználó által módosított.

A GUI-hoz tartozó kódot a GUIDE generálja, abban a megfelelő helyekre beírt kód határozza meg a GUI viselkedését eseményvezérelt programozási paradigma szerint.

A szerkesztőfelület értelemszerű, a bal oldalon találhatók a nézeten elhelyezhető objektumok.



Az egyes objektumok és a teljes felület tagsági változóinak értékeit a Tulajdonság böngésző (Inspector) segítségével módosíthatjuk és itt ugorhatunk az adott objektum callback függvényire is.

Az Image\_frame objektum osztálya az Axes. A benne szereplő jpg képet az erre szolgáló imshow metódus meghívásával lehet beilleszteni, méghozzá az Axes objektum Create eseményéhez tartozó kezelőfüggvényébe. A kezelő függvény a 2006a verzióban úgy generálódik, hogy az %automatic szót kell beírni a függvény melletti mezőbe, majd elmenteni a GUI-t, de ezt a 2012a verzió már magától elvégzi.

•	<b>≜</b> ↓ <b>₩</b> ‡ <b>₩</b> ‡			
	CameraUpVectorMode		auto	+ <sup>1</sup>
	CameraViewAngle		6.608610360311923	Ø
	CameraViewAngleMode		auto	-
	Clipping		on	-
+	Color			
	ColorOrder			
	CreateFcn	4	ta,guidata(hObject))	ø
÷	CurrentPoint		[0 0]	
Ŧ	DataAspectRatio		[111]	
	DataAspectRatioMode		auto	•
	DeleteFcn	4		Ø
	DrawMode		normal	*
	FontAngle		normal	•
	FontName		Helvetica	Ø
	FontSize		10.0	Ø
	FontUnits		points	•
	FontWeight		normal	•
	GridLineStyle		:	•
	HandleVisibility		on	•
	HitTest		on	•

% Executes during object creation, after setting all properties.
<pre>function Image frame CreateFcn(hObject, eventdata, handles)</pre>
<pre>Limshow('Holtidos_szakasz.jpg');</pre>
<pre>% hObject handle to Image_frame (see GCBO)</pre>
<pre>% eventdata reserved - to be defined in a future version of MATLAB</pre>
<pre>% handles empty - handles not created until after all CreateFcns c</pre>

Az elkészült diagram értelemszerű, futtatáskor az alábbi felületet kapjuk:

科 PI_designer	
PI des	signer
hollin	
fülöszál	hömérséklet érzékelő
Szakasz paraméterek	PI paraméterek
A Edit Text	Ap <sup>n.a.</sup>
T Edit Text	Ti n.a.
Th Edit Text	Méretezés

A tervező eljárást egy különálló függvényben célszerű implementálni, hogy a fázistartalék például 60 fok. A méretezés a korábban tanultak alapján mindössze három sor.



Ha készen vagyunk, akkor már csak a Méretezés gomb eseménykezelő függvényét kell létrehoznunk és megírnunk a PI\_designer.m állományban:

155 156	% Executes on button press in design. [function design Callback(hObject, eventdata, handles)	
157 158 159 160 -	<pre>-% hObject handle to design (see GCBO) % eventdata reserved - to be defined in a future vers: -% handles structure with handles and user data (see hedit = findobj('Tag','Aval');</pre>	kiszedjük az értékeket a szövegdobozokból
161 - 162 - 163 - 164 - 165 - 166 -	<pre>A = str2double(get(hedit, 'String')) hedit = findobj('Tag', 'Tval'); T = str2double(get(hedit, 'String')) hedit = findobj('Tag', 'Thval'); Th = str2double(get(hedit, 'String')) [Ap,Ti]=myPI(A,T,Th);</pre>	itt hívjuk meg a tervezést végrehajtó függvényt ≡
167 - 168 - 169 - 170 -	<pre>htext = findobj('Tag', 'Ap'); set(htext, 'String', num2str(Ap)); htext = findobj('Tag', 'Ti'); set(htext, 'String', num2str(Ti));</pre>	az eredményeket a megfelelő szövegdobozba írjuk be

Ha rendelkezésünkre áll a Matlab Compiler eszköz, akkor lehetőségünk van különálló alkalmazás létrehozására a

mcc -m PI\_designer.m

paranccsal, amely hatására egy exe állomány keletkezik.

### 2. feladat - A PI méretező módosítása

Módosítsa az előző feladat nyomán keletkezett alkalmazást, hogy a fázistartalék értékét is a felhasználó adhassa meg!

#### 3. feladat – A PI méretező tesztelése szimulációval

Hozzon létre egy Simulink modellt a zárt szabályozási kör szimulációjához, amelynek munkaterében szerepelnek változóként a szakasz és a szabályzó paraméterei! Egészítse ki a felhasználói felületet, hogy a szabályzó méretezése nyomán a zárt kör szimulációjának eredménye a felületen megjelenjen (egységugrás alapjel esetén).

**Megoldás:** A Matlabon belül nem csak az 'alap' munkatér (Base Workspace) tartalmazhat változókat, minden Simulink modellnek is van saját munkatere. A szabályozási körünk paramétereit a modell munkaterében hozzuk létre a modellböngésző (Model Explorer) segítségével. Ehhez először egy modellt kell megnyitni.



A modellböngészőben a model munkaterében egyszerűen létrehozhatjuk és inicializálhatjuk a változókat az összes paraméter számára (szakasz és szabályzó). A változókra a modell blokkjainak párbeszéd ablakában hivatkozhatunk.

Image: Model Explorer         File       Edit       View       Tools       Adu         Image: Image	d Help I	) 🔘 🔳 🔶 🍽 🕀 🖽	- • × ≠ 2
Search: by Name Model Hierarchy Simulink Root Base Workspace Closed_loop	Name: Contents of: Model Workspace Column View: Data Obje      Sh Name Value	Filter Contents ow Details 5 object(s) T	Model Workspace Workspace data Data source: MDL-File
Model Workspace Code for Closed Advice for Closed Simulink Design V	Hance     Hance       ■     A       ■     Ap       0.0785398       ■     T       1.5       ■     Th       2       ■     Ti       1.5	double (auto) double (auto) double (auto) double (auto) double (auto)	Model arguments (for referencing this model):
4	Contents     Search	▶ Results	Revert Help Apply

A Simulink modellben a holtidőt is kezelő LTI blokkot használunk és a szimulált kimenetet egy mat fájlba mentjük.



A felhasználói felületet ki kell egészíteni egy Axes típusú objektummal, amelyben a szimulált tranzienst fogjuk megjeleníteni. természetesen ezt megintcsak a GUIDE segítségével tesszük meg.



Utolsó lépésként a felhasználó felületen korábban is szereplő egyetlen gomb kezelőfüggvényének kiegészítésére van szükség, amely a tervezésen kívül

- módosítja a modell munkaterében található változókat
- lefuttatja a szimulációt
- betölti az eredményeket tartalmazó állományt
- törli az aktuális tranzienst és a helyére kirajzolja az újat

Az alábbi utasítások állnak rendelkezésünkre:

- Simulink modellünk megnyitására a load\_system utasítás szolgál
- A munkatérhez tartozó handle-t a get\_param utasítás adja vissza
- Valamely munkatér változójához az assignin utasítással férhetünk hozzá
- Modellünk szimulációját az sim utasítással kezdeményezetjük

A keletkező kód alább látható

```
% --- Executes on button press in design.
function design Callback(hObject, eventdata, handles)
% eventdata reserved - to be defined in a future version of MATLAB
              structure with handles and user data (see GUIDATA)
 -% handles
 hedit = findobj('Tag','Aval');
 A = str2double(get(hedit, 'String'));
 hedit = findobj('Tag','Tval');
 T = str2double(get(hedit, 'String'));
 hedit = findobj('Tag','Thval');
 Th = str2double(get(hedit, 'String'));
 % méretezés
 [Ap,Ti]=myPI(A,T,Th);
 % eredmények kiírása
 htext = findobj('Tag', 'Ap');
 set(htext, 'String', num2str(Ap));
 htext = findobj('Tag','Ti');
 set(htext, 'String', num2str(Ti));
 % paraméterek áttöltése a modellbe
 load system('Closed loop');
 hws = get_param(bdroot, 'modelworkspace');
 hws.assignin('A',A);
 hws.assignin('T',T);
 hws.assignin('Th',Th);
 hws.assignin('Ap',Ap);
 hws.assignin('Ti',Ti);
 save system;
 % modell szimulálása
 sim('Closed loop');
 load output
 % eredmények kijelzése a results elnevezésű keretbe
 haxis = findobj('Tag', 'results');
 cla(haxis, 'reset');
 plot(y(1,:),y(2,:));
```

#### 4. feladat – A szimulációs idő változtatása

Egészítse ki az előző feladatban kapott alkalmazást, hogy a szimuláció időtartalma mindig automatikusan a beállított holtidő ötszöröse legyen! A szimulációs idő a Simulink modell egyik paramétere és közvetlenül beállítható a sim utasítás egy argumentumaként (v.ö. doc sim).

## 5. feladat – Kódgenerálás Simulink diagrammból

A feladat egy S-függvény C kódú megvalósítása egy Simulink modellben, majd C kódból annak kifordítása és futtatás normál üzemmódban.

📡 Simulink Library Browser		
File Edit View Help		
🗅 😅 🔹 Enter search term	- M 📺	
Libraries	Library: Simulink/User-Defined Functions Search Results: (none)	Most Frequently
Simulink Commonly Used Blocks	f(u)     Fon     Interpreted MA-     Interpreted MA-     TLAB Fon     TLAB Function	
Discrete	B S-Function	
Logic and Bit Operations = Lookup Tables Math Operations	> sydem > S-Function	
Model Verification     Model-Wide Utilities     Ports & Subsystems	S-Function Examples Examples	
Signal Attributes Signal Routing		
Sinks		
User-Defined Functions		
+ Aerospace Blockset		
E Communications System		
🛨 🕞 DSP System Toolbox 🔻		
Showing: Simulink/User-Defined Function	ins	.41

1. ábra: S-függvény C kódú megadását lehetővé tevő S-Funtion Builder a Simulink könyvtárban

A S-Function Builder a User-Defined Functions könyvtárban található. A modellbe behelyezve egy párbeszédablak segítségével adhatjuk meg az S-függvényekhez tartozó metódusok (kimenet számítása – Outputs, folytonos idejű állapotok deriváltjainak számítása – Continuous Derivatives, diszkrét idejű állapotok aktualizálása – Discrete Update) C kódját. A kód megadása előtt a bemenetek, kimenetek és állapotok dimenzióit és a paramétereket is meg kell adni, illetve létre kell hozni.

S-Function Builder: un	titled/S-Function Builder			
S-function name:				Build
S-function parameters—				
Name	Data ty	/pe	Value	
				X
Port/Parameter	Initialization Data Properties	Libraries Outputs Continu	ous Derivatives Discrete Upo	date Build Info
Input Ports	Description			
🕀 👿 Output Ports	The S-Function Builder block of norts, output norts, and a varia	creates a wrapper C-MEX S-fu able number of scalar, vector	nction from your supplied C c or matrix parameters. The inn	ode with multiple input ut and output norts
Parameters	can propagate Simulink built-i	in data types, fixed-point data	types, complex, frame, 1-D, a	nd 2-D signals. This
	TLC file to be used with Simuli	nd continuous states of type re ink Coder for code generation	eai. You can optionally have ti	he block generate a
	S-function settings			
	Number of discrete states:	0	Sample mode:	Inherited 👻
	Discrete states IC:	0	Sample time value:	
	Number of continuous states:	0		
	Continuous states IC:	0		
L	J [			
				Cancel Help

A kódrészleteket a megfelelő füleknél tudjuk megadni a C szintaxist követve. A fordítást külső fordítóval, vagy a Matlabbal járó lcc fordítóval hajthatjuk végre. Az mbuild –setup parancs segítségével jelölhetjük ki a fordítót. A kód bevitele nyomán a Build parancs állítja elő a Matlabon kívül végrehajtható kódot. Fontos, hogy a kód végrehajtását továbbra is a Matlab vezérli, azaz nem külső (External futtatásról van szó).

Válasszon egy rendszert az alábbi kettő közül és a megfelelő dinamikát kódolja C nyelven S függvény segítségével és szimulálja le az eredményt! Választhatja az előző mérésen bevezetett buszmodellt is.



A bal oldali ábrán szereplő két rugó lineráis, az egyenleteket az egyes tömegek gyorsulását leíró differenciálegyenletek adják. A jobb oldalon szereplő kiskocsi egyenletei pedig az alábbiak:

$$\dot{x} = \cos(\theta) u_1$$
$$\dot{y} = \sin(\theta) u_1$$
$$\dot{\theta} = u_2$$

#### 6. feladat – külső kód generálása Simulink Coder segítségével

A Simulink modellből történő kódgeneráláshoz a Configuration parameters dialógusablak Code Generation csoportját kell megfelelően beállítani. A Build parancs a modell eszközsorában is megtalálható.

Configuration Parameters: unt	itled/Configuration (Active)	8
Select:	Target selection	•
	System target file: grt.tlc Browse Language: C 🗸	
Hardware Implementation     Model Referencing     Simulation Target     Code Generation     Report     Comments     Symbols     Custom Code     Debug     Interface	Build process Compiler optimization level: Optimizations off (faster builds)  TLC options: Makefile configuration Generate makefile Make command: make_rtw Template makefile: grt_default_tmf	ш
⊕ HDL Code Generation	Code Generation Advisor         Select objective:       Unspecified         Check model before generating code:       Off         Check model only       Build	-
•	III	
0	OK Cancel Help App	ply

A HSZK gyakorlat során nem áll rendelkezésre külső, valós idejű target, a cloud megvalósítás miatt pedig a Windows alá telepíthető valós idejű mag sem használható (ennek egyik oka, hogy a virtuális gép 64 bites operációs rendszert futtat, ami nem kompatibilis a rendelkezésre álló RT maggal).

A fentiek miatt külső futtatásra az ún. Generic RT target (GRT) használata mutatkozik célszerűnek. Ehhez az ennek megfelelő System target állományt kell kiválasztani a Code Generation fülnél.

🚳 Configuration Parameters: unti	itled/Configuration (Active)
Select:	Target selection
	System target file: grt.tlc Browse Language: C
Diagnostics     Hardware Implementation	Build process
Model Referencing	Com System Target File Browser: untitled
- Code Generation	TLC System Target File: Description:
Report Comments Symbols Custom Code Debug Interface B-HDL Code Generation Global Settings Test Bench EDA Tool Scripts	Me arduino.tlc Arduino Target asap2.tlc ASAM-ASAP2 Data Definiti asap2.tlc ASAM-ASAP2 Data Definiti ert.tlc Embedded Coder ert.tlc Create Visual C/C++ Solu grt.tlc Generic Real-Time Target grt.tlc Generic Real-Time Target grt.tlc Create Visual C/C++ Solu grt_malloc.tlc Generic Real-Time Target Se grt_malloc.tlc Generic Real-Time Target idelink_ert.tlc IDE Link ERT ch idelink grt.tlc IDE Link GRT , Check model
4	G Full Name: D:\MATLAB\R2012a\rtw\c\grt\grt.tlc Template Makefile: grt_default_tmf Make Command: make_rtw OK Cancel Help Apply
0	OK Cancel Help Apply

A fordítást elvégezve egy exe állományt kapunk, amelyet a Matlab parancsssorból a ! segítségével az operációs rendszernek elküldött paranccsal, vagy cmd ablakból indíthatunk el, immáron a Matlabtól függetlenül, azaz külső módban. A szimuláció eredménye egy .mat állományba kerül, ahonnan a Matlab munkaterébe egyszerű módon importálható.