Kódgenerálás és GUI fejlesztés Matlab-Simulink környezetben

A mérés célja: Grafikus felhasználói felülettel rendelkező alkalmazás létrehozása a Matlab alatt. Az App Designer eszköz és elemkönyvtárainak megismerése. Simulink modell létrehozása és vezérlése az alkalmazásból.

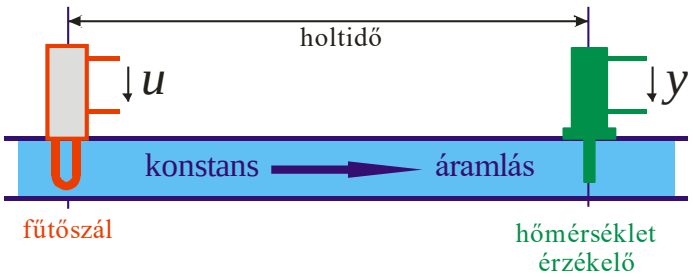
Felhasznált eszközök: (KIFÜ-NIIF felhő architektúra – https://niif.cloud.bme.hu/dashboard/), Matlab R2018a, Simulink, App Designer, Matlab Compiler.

Rendelkezésre álló, letölthető állományok:

- Mérési útmutató (jelen leírás)
- Holtidos_szakasz.jpg (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[1]. A PI szabályzót egy fűtött folyadékot keringető berendezés (például dialízisgép) hőmérséklet szabályozójához kell megtervezni, amelynek sémája az ábrán látható.



1. ábra. A holtidős szakasz működésének vázlata (a Holtidos_szakasz.jpg ábra).

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.

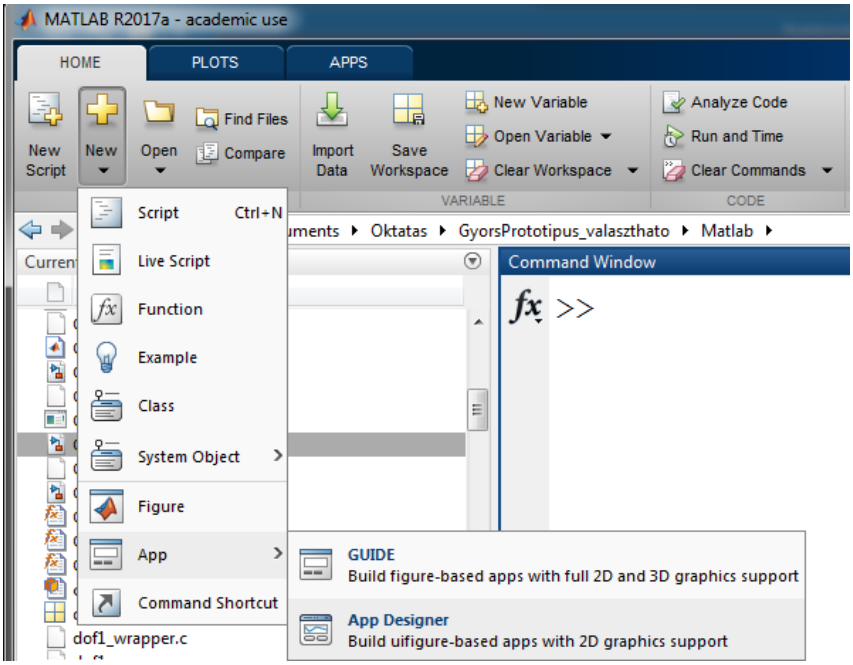
Megoldás: A fejlesztést a `GUIDE` vagy az `App Designer` eszköz segítségével végezhetjük. A `GUIDE` összetettebb eszköz, ugyanakkor egyszerűbb applikációk (Appok) fejlesztéséhez az egyszerűsített és letisztultabb `App Designer` javasolható. A feladatban szereplő eszköz esetében az `App Designer` alkalmazása elegendő lesz, az alkalmazásmodell a `GUIDE` esetében is hasonló.

A fejlesztő eszköz indításakor egy újabb felület jelenik meg. Az alkalmazás felépítése egyszerű. Külön szerkesztői nézetben helyezhetjük el a felhasználói felület elemeit (`Design View`). Az programozási modell eseményvezérelt, az egyes kezelőelemek eseményeihez rendelt kódot is egy külön nézetben szerkeszthetjük (`Code View`). A kód egy részét a fejlesztőkörnyezet generálja, ennek szerkesztése nem tanácsos.

A fejlesztőkörnyezet további elemeinek használata is értelemszerű. Az alkalmazható kezelőszervek könyvtára a bal oldalon található, ezeket *drag & drop* művelettel példányosíthatjuk az alkalmazásunkban. Az egyes elemekhez

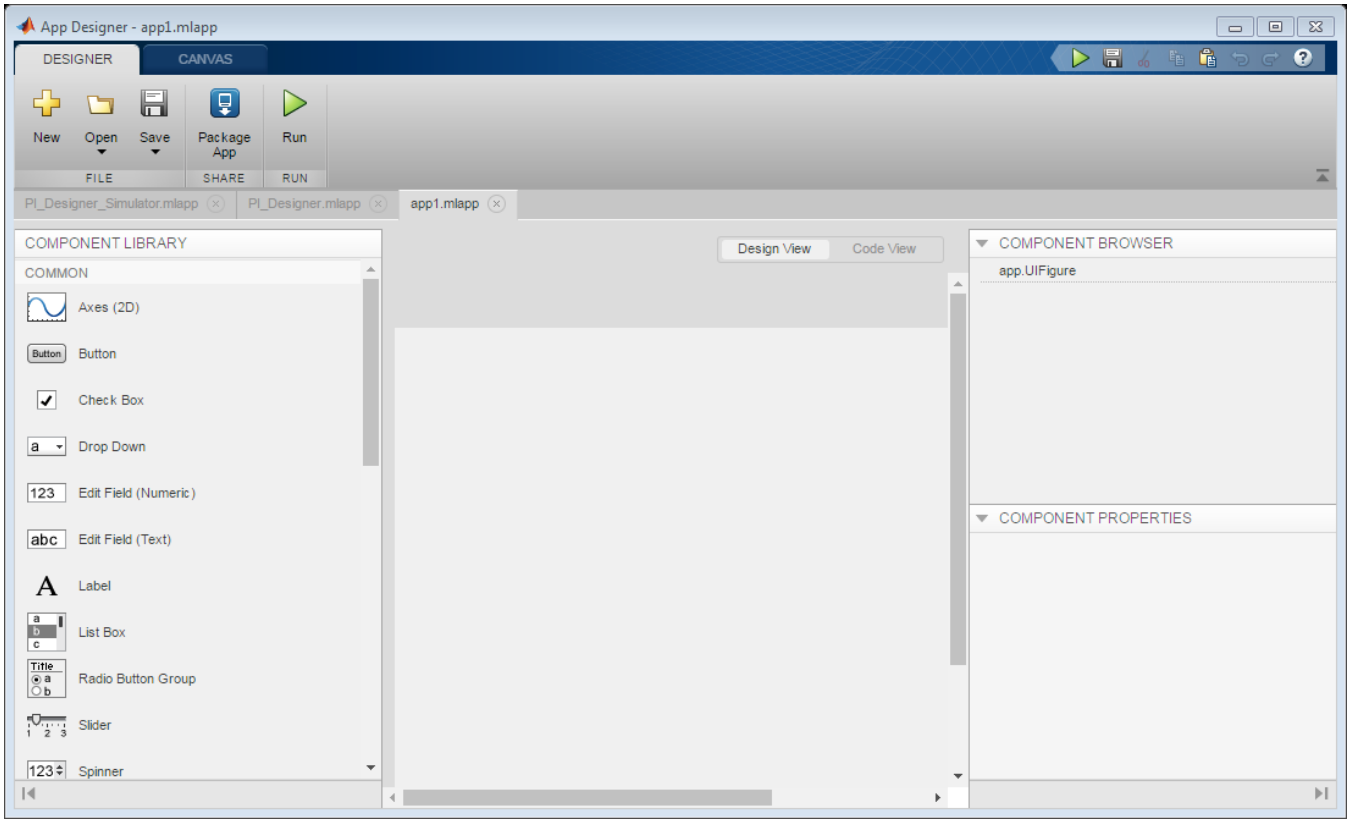
[1] PI szabályozó méretezése a Rendszerelmélet (VIHVAB00) tárgyból szerepelt

eseménykezelő függvényeket a jobb oldali menüben rendelhetünk és itt állíthatók a felület elemeinek tulajdonságai is.



2. ábra. Alkalmazások fejlesztése (GUIDE és App Designer)

Az alkalmazást a `Run` paranccsal indíthatjuk, a futtatás értelemszerűen a Matlab keretrendszerben zajlik.



3. ábra. Az App Designer felülete.

Kezdjük a kép elhelyezésével. Képet a `Axes (2D)`, `UIAxes` osztályú objektum esetében mutathatunk a felhasználónak. A `jpg` képet az erre szolgáló `imshow` metódus meghívásával lehet megjeleníteni. A metódus hívását ugyanakkor be kell illeszteni egy eseménykezelő függvénybe, viszont az `UIAxes` objektumokhoz nem

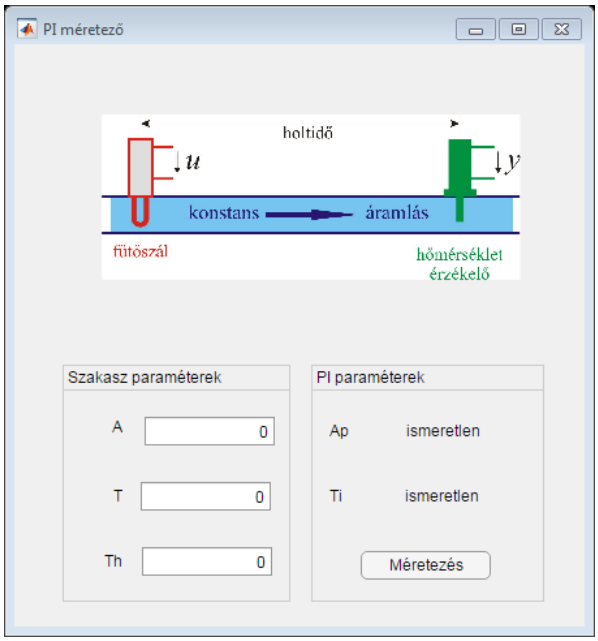
tartozik eseménykezelő. Ezért az alkalmazás ablakának (`UIFigure`) egyik eseménykezelőét használjuk (`startupFcn`), amely annak létrehozásához tartozó esemény nyomán kerül végrehajtásra. Ennek kódját mutatja a 4. ábra.

```
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)
        imshow('Holtidos_szakasz.jpg','parent',app.UIAxes);
    end
```

4. ábra. Kép mutatása UIAxes osztályú objektummal.

A PI szabályozó tervezéséhez a fejlesztői felületen például a 5. ábra szerinti módon helyezhetjük el az elemeket. A szakasz paramétereinek megadásához numerikus szerkesztő mezőket (`Edit Field (Numeric)`), a számított paraméterek megjelenítéséhez pedig `Label` elemeket használhatunk.



5. ábra. A PI méretező felülete. Az elrendezésben a baloldali csoportban találhatók a tervezés bemeneti paraméterei, jobb oldalon pedig a méretezés eredménye.

A tervező eljárást egy különálló függvényben is lehet implementálni, hogy a méretezést például 60 fokos fázistartalékra végezzük. A méretezés a korábban tanultak alapján mindössze három sor, amit a felület `Méretezés` nyomógombjának eseményéhez (gomb megnyomása) rendelünk.

```
    % Button pushed function: DesignButton
    function DesignButtonPushed(app, event)
        A = app.PlantGain.Value;
        T = app.PlantTimeCte.Value;
        Th = app.PlantTimeLag.Value;
        % controller design calculations for 60 degrees phase margin
        Ti = T;
        wc = pi/6/Th;
        Ap = Ti*wc/A;
        app.TiValue.Text = num2str(Ti);
        app.ApValue.Text = num2str(Ap);
    end
```

6. ábra. A Méretezés gomb megnyomásához tartozó eseménykezelő függvény kódja.

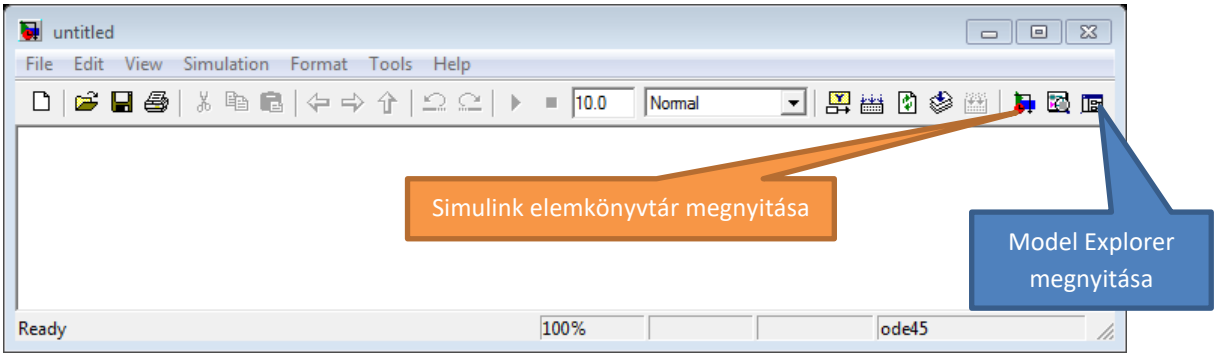
2. feladat – A PI méretező módosítása

Módosítsa az előző feladat nyomán keletkezett alkalmazást, hogy a fázistartalék értékét is a felhasználó adhassa meg! Ehhez illesszen be egy új mezőt az 5. ábra szerinti felület bal oldalára (a pereméterek közé) és módosítsa a 6. ábra kódját. Ügyeljen rá, hogy a számításokban a fázistartaléknak radiánban kell szerepelnie.

3. feladat – A PI méretező tesztelése szimulációval

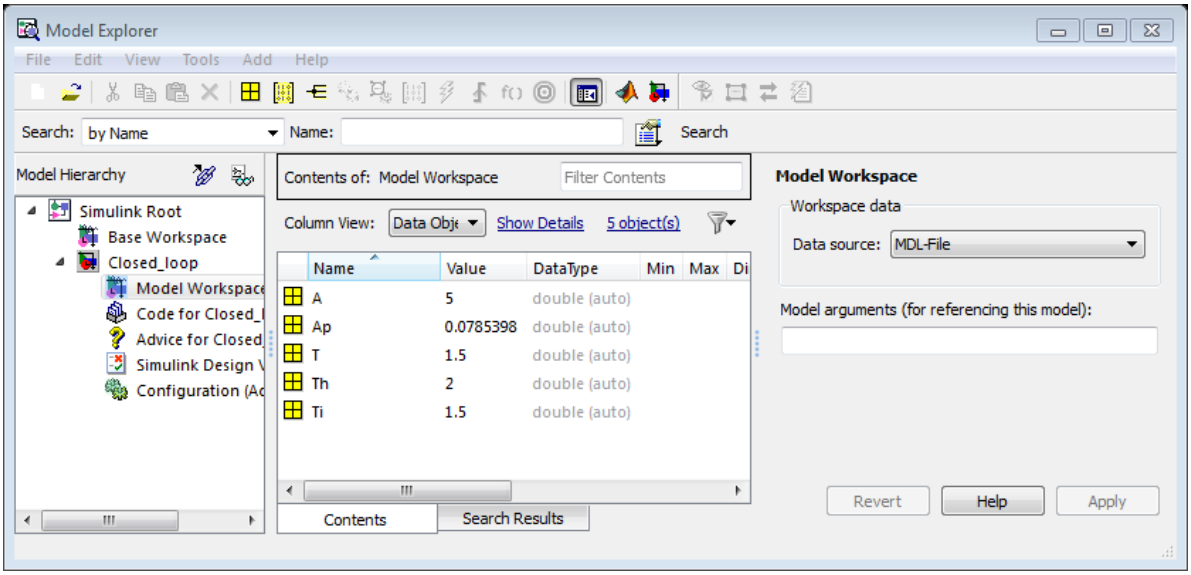
Hozzon létre egy Simulink modellt a zárt szabályozási kör szimulációjához. A Simulink modell munkaterében szerepelnek változóként a szakasz és a szabályzó paraméterei! Egészítse ki a felhasználói felületet, hogy a szabályzó méretezése nyomán a zárt kör szimulációjának eredménye a felületen megjelenjen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



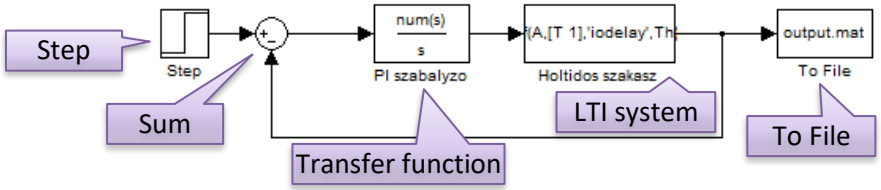
7. ábra. A modell böngésző és az elemkönyvtár megnyitása (újabb Simulink verzióknál a felület parancssorának megjelenése eltérő lehet)

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 is. Fotos megjegyezni, hogy amennyben azonos nevű változó szerepel a modell és a alap munkatérben, akkor a blokk a modellben szereplő értéket fogja használni, azaz a modell munkaterében szereplő változó „eltakarja" az azonos nevű, alapmunkateri példányt.



8. ábra. A modellböngésző felülete

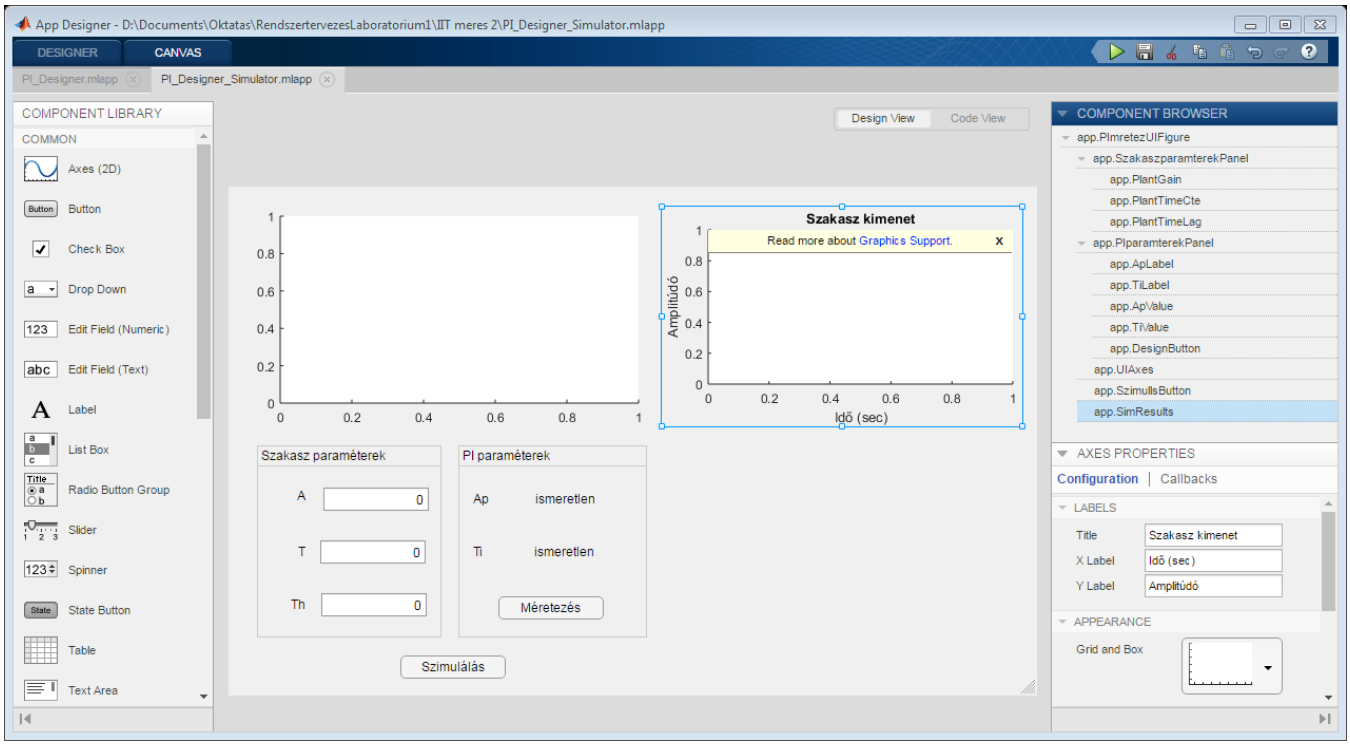
A Simulink modellben a holtidőt is kezelő LTI blokkot használunk és a szimulált kimenetet egy `mat` fájlba mentjük. A modell elemeit vagy gyorskeresővel illesztjük be, vagy az ún. Simulink elemkönyvtárból (utóbbi a Simulink parancssorából megnyitható). A gyorskereső használatakor elegendő a keresett modell nevét a modellben elkezdeni begépelni, melynek nyomán megjelenik egy találati lista. A kiválasztott elem közvetlenül a modellbe kerül. A 9. ábra a hatásvázlaton kívül mutatja az egyes elemek kereshető könyvtári neveit is.



9. ábra. A holtidőt tartalmazó szabályozási kör modellje Simulink-ben (az elemek könyvtári megnevezésével)

Az összeköttetések létrehozása értelemszerű, ugyanakkor csak ki és bementek köthetők össze.

A felhasználói felületet ki kell egészíteni egy UI`Axes` típusú objektummal, amelyben a szimulált tranzienst fogjuk megjeleníteni. Természetesen ezt megintcsak az `App Designer` segítségével tesszük meg.



10. ábra. Újabb UIAxes objektum hozzáadása a felülethez a szimuláció eredményének megjelenítéséhez

Utolsó lépésként a szimulációhoz tartozó gomb eseménykezelőjének megadására van szükség, amely az alábbi feladatokat végzi el:

1. módosítja a modell munkaterében található változókat a felületről beolvasott értékek szerint;
2. lefuttatja a szimulációt;
3. betölti az eredményeket tartalmazó állományt;
4. törli az aktuális tranzienst és a helyére kirajzolja az újat.

Ehhez 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

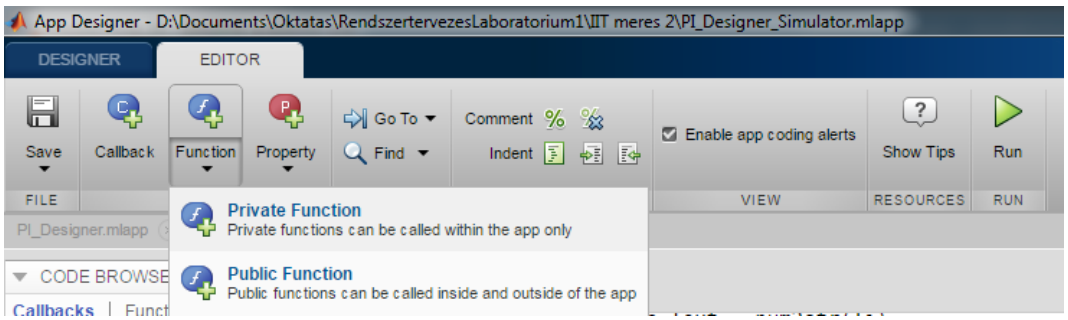
A keletkező kódot a 11. ábra mutatja.

```
    % Button pushed function: SzimullsButton
    function SzimullsButtonPushed(app, event)
        load_system('Closed_loop_PI');
        hws = get_param(bdroot,'modelworkspace');
        hws.assignin('A',app.PlantGain.Value);
        hws.assignin('T',app.PlantTimeCte.Value);
        hws.assignin('Th',app.PlantTimeLag.Value);
        hws.assignin('Ap',str2num(app.ApValue.Text));
        hws.assignin('Ti',str2num(app.TiValue.Text));
        save_system;
        sim('Closed_loop_PI');
        load output;
        plot(app.SimResults,y(1,:),y(2,:));
    end
```

11. ábra. Szimuláció futtatása eseménykezelő függvényből.

4. feladat – Az alkalmazás kiegészítése

1. Egészítse ki az előző feladatban kapott alkalmazást, hogy a szimuláció időtartalma mindig automatikusan a beállított holtidő legalább ötszöröse, de az egységugrás alapjel hatására a kimeneten jelentkező tranziens lezajlásának megfigyeléséhez is elegendő legyen! A szimulációs idő a Simulink modell egyik paramétere és közvetlenül beállítható a `sim` utasítás egy argumentumaként (v.ö. `doc sim`).
2. Egészítse ki az előző feladatban kapott alkalmazást, hogy a felületen a beavatkozó jel is megjelenjen!
3. Egészítse ki az alkalmazást, hogy a szabályozó paramétereit egy belső (`private` láthatósági osztályú) függvény számolja. Ilyen függvény létrehozásához nyújt segítséget a 12. ábra.



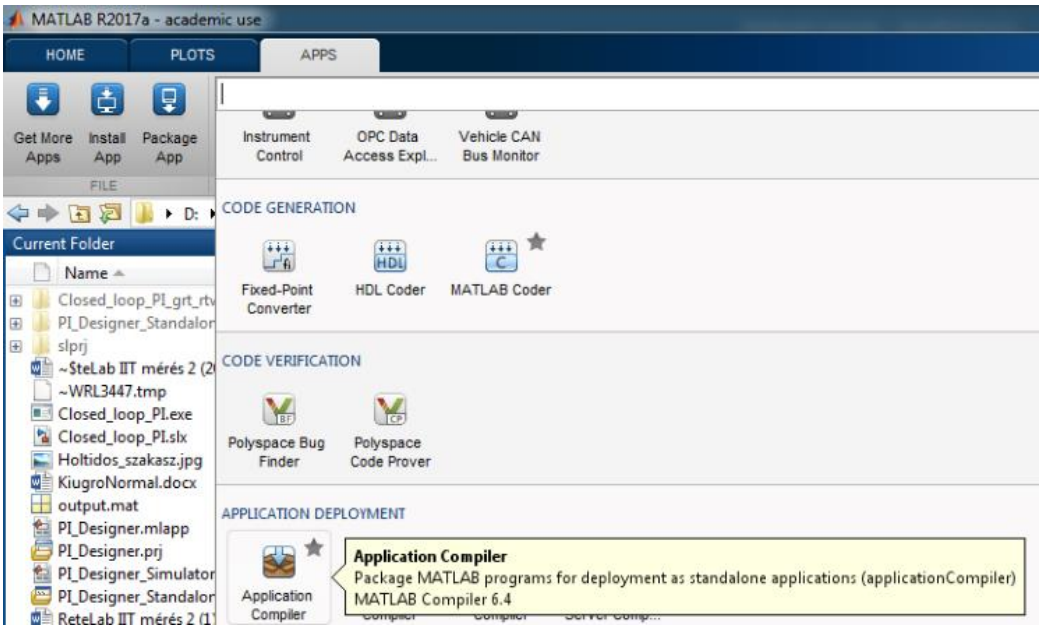
12. ábra. Függvények (`private` vagy `public` láthatósági osztályú) létrehozása

5. feladat – Önálló alkalmazás létrehozása az elkészített App-ből

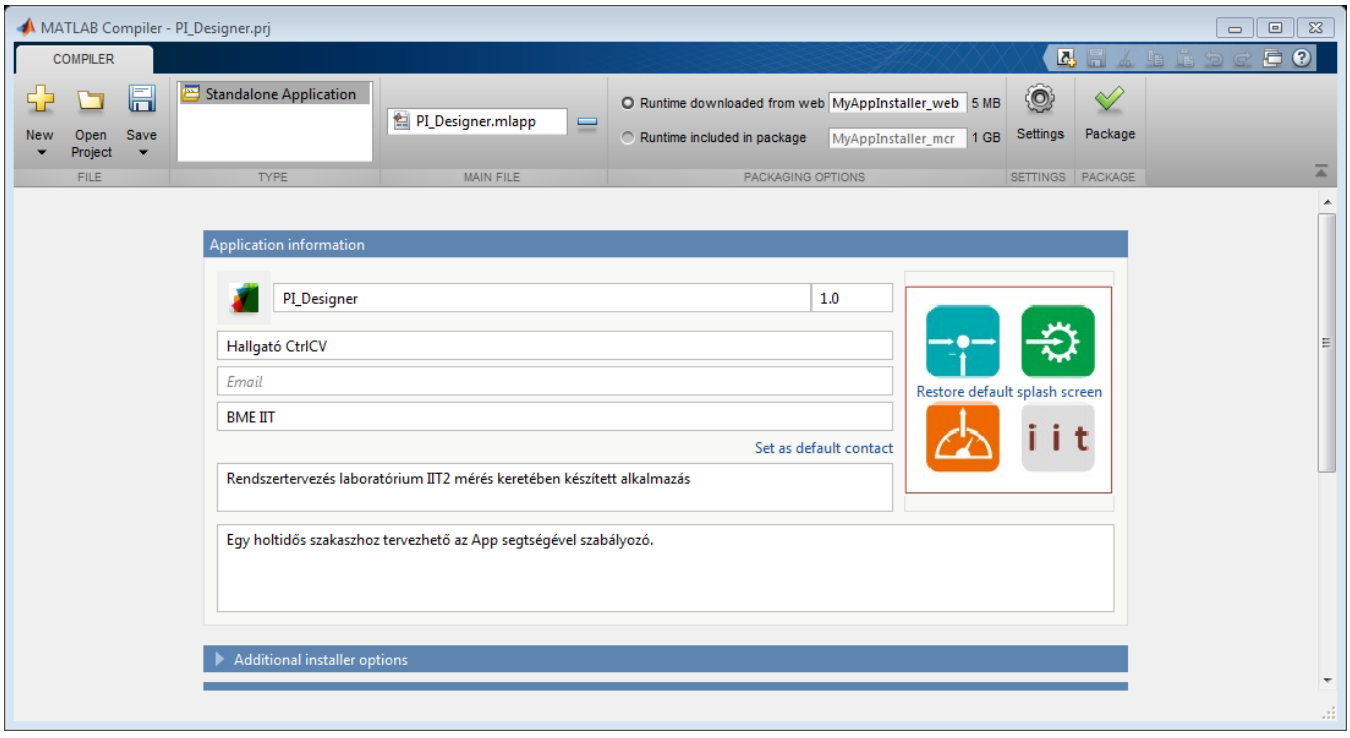
Az eddig elkészült Appok futtatásához a Matlab indítása szükséges volt. Ugyanakkor a Matlab fordító eszközének (`Matlab Compiler`) segítségével lehetőség van olyan önálló (standalone) alkalmazás előállítására is, amelynek futtatásához nincsen szükség a Matlab indítására. Fontos megjegyezni ugyanakkor, hogy a lefordított alkalmazáshoz vagy hozzácsomagoljuk a Matlab futtató környezetet (ez akár 1Gb is lehet) vagy lehetővé tesszük, hogy ehhez az alkalmazásunk a hálózaton keresztül férjen hozzá. Utóbbira is csak akkor van szükség, ha futtató környezet nem található meg lokálisan. A későbbiekben mindig a hálózaton keresztüli hozzáférést biztosító opciót válasszuk! A fordító is egy App persze, ezt mutatja a 13. ábra.

Az önálló alkalmazást előállító felület használata is értelemszerű, esetünkben a szükséges állományok kiválasztása is egyszerű. A kiválasztandó `MAIN FILE` most a `PI_Designer.mlapp`, további függvényekre és szkriptekre nincsen szükség. Az alkalmazás létrehozásához a `Package` parancsot kell használni. Sajnos a kód analízise nem elég fejlett ahhoz, hogy az `imshow` utasításnál megadott `Holtidos_szakasz.jpg` fájlt is a csomagolni kívánt állományok közé illessze a keretrendszer, így ezt kézzel kell megtenni a `Files installed for your end user` mezőnél.

A hosszas fordítás után a végrehajtható állományok a `PI_Designer` könyvtárban keletkeznek. A Matlab három különböző változatot is előállít, próbálkozzunk a `for_redistribution_files_only` könyvtárban található `PI_Designer.exe` futtatásával.



13. ábra. A Matlab fordító felületének indítása



14. ábra. A Matlab fordító felülete

IMSc feladat

Az IMSc feladatra két IMSc pont szerezhető.

Egészítse ki a hőmérsékletszabályozási kör hatásvázlatát egy szakasz bementén ható, additív (egységugrás alakú) zavaró jellel. A zavaró jel értékét és a zavarás érkezésének idejét (azaz az egységugrás idejét) a felhasználó az App felhasználói felületén állíthatja. A szimuláció idejének beállításakor az alkalmazás figyelembe veszi a zavarás idejét, hogy a zavarás okozta tranziens is megjelenjen a kimeneten, illetve a beavatkozó jelben.

A feladatok leadása a kari Moodle segítségével

Az IIT2 méréshez létrehozott feladatnál tölthetők fel a megoldások a kari Moodle Portálon (https://edu.vik.bme.hu). A feltöltött megoldásnak az alábbiakat kell tartalmaznia:

1. Az 1-4 feladatokat megoldó, működőképes és stabil zárt szabályozási kört méretező Matlab alkalmazás (APP) a szükséges állományokkal (kép, Smulink modell), tömörítve.
2. Az 1-2 feladatokat megoldó, működőképes és stabil zárt szabályozási kört méretező, önállóamfutó alkalmazás.

A feltöltés határideje a mérést követő hét péntek 23:59.