



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar

Mérnöki Intézet



DIPLOMAMUNKA

Beltéri hőmérséklet szabályozása fuzzy logikával

OE-AMK

Hallgató neve: Bekk Tímea

2024

Hallgató törzskönyvi száma: T001476/FI12904/A-M



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Alba Regia Műszaki Kar
Mérnöki Intézet

DIPLOMAMUNKA FELADATLAP

Hallgató neve: Bekk Tímea

Diplomamunka száma: SZD2306210941496778

Törzskönyvi száma: T001476/FI12904/A-M

Neptun kódja:

Szak: mechatronikai mérnöki

Specializáció:

A dolgozat címe: Beltéri hőmérséklet szabályozása fuzzy logikával

A dolgozat címe angolul: Fuzzy Logic Controller for Regulating the Indoor Temperature

A feladat részletezése:

A hallgató feladata beltéri hőmérséklet szabályozásának megvalósítása fuzzy szabályozóval, amelynek elkészítéséhez a Matlab Fuzzy Logic Toolbox-ot használja. A javasolt módszer validálásához és teszteléséhez készítse el a vizsgált rendszer MATLAB-Simulink modelljét, amelybe integrálja a fuzzy szabályozót. Hasonlítsa össze a javasolt fuzzy szabályozót a leggyakrabban használt szabályozó eszközökkel (pl. On/Off szabályozóval) és értékelje a kapott eredményeket.

Intézményi konzulens neve: Piglerné Dr. Lakner Rozália

A kiadott téma elévülési határideje: 2026. 07. 10.

Beadási határidő: 2024. 05. 15.

A diplomamunka: Nem titkos.

Kiadva: Székesfehérvár, 2024. 03. 20.



Intézetigazgató

A dolgozatot beadásra alkalmasnak találom: 2024. 05. 15.

R. Lakner

belső konzulens

külső konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar
Mérnöki Intézet

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a dolgozat saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült dolgozatban található eredményeket az Óbudai Egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Kelt: Székesfehérvár, 2024.05.22.

.....
hallgató aláírása

Tartalomjegyzék

1.	Bevezetés	1
2.	A felhasznált szakirodalom bemutatása.....	2
2.1	Rendszerek modellezése és szabályozása.....	2
2.1.1	Rendszerről általánosságban.....	2
2.1.2	Rendszer modellje.....	3
2.1.3	Szabályozás.....	4
2.1.4	PID szabályozó	5
2.2	Fuzzy rendszerek	8
2.2.1	Fuzzy halmazok és alapvető típusaik.....	8
2.2.2	Műveletek fuzzy halmazokkal	12
2.2.3	Fuzzy szabályok.....	14
2.2.4	Fuzzy szakértő rendszer.....	16
3.	Felhasznált eszközök	22
3.1	Matlab Simulink	22
3.1.1	A Simulink használata	22
3.1.2	Simulink Blokkok.....	25
3.1.3	A szimuláció és paraméterei	27
3.2	Matlab Fuzzy Logic Toolbox	29
3.2.1	Fuzzy következtető rendszer szerkesztő	31
3.2.2	Tagsági függvény szerkesztő	32
3.2.3	Szabály szerkesztő	34
3.2.4	Szabály megjelenítő	37
3.2.5	Felület megjelenítő	39

3.2.6	A modell mentése	40
4.	A hőmérsékletszabályozás megvalósítása	41
4.1	A vizsgált rendszer bemutatása.....	41
4.2	A vizsgált rendszer Simulink modellje	42
4.3	A terem részrendszer	45
4.4	Külső hőmérséklet	49
4.5	Hőmérsékletek megjelenítése	50
4.6	A szabályozó részrendszer.....	51
4.6.1	ON/OFF szabályzó	53
4.6.2	PID szabályozó	58
4.6.3	Fuzzy szabályozó	62
5.	Értékelés, továbbfejlesztési lehetőségek.....	77
6.	Összefoglaló.....	78
7.	Summary.....	79
8.	Irodalomjegyzék	80
9.	Ábrajegyzék	82

1. BEVEZETÉS

A diplomamunkám során szeretném bemutatni egy terem hőmérsékletszabályozásának különböző megvalósítási lehetőségeit, kiemelten a nyelvi kifejezésekkel működő fuzzy szabályozót. Egy konkrét termet vizsgáltam, amely a Székesfehérvár Pirosalma utca 1-3. épületében található 1. emeleti 103 terem, a modelljeimben ennek a paramétereit használtam.

Azért választottam ezt a témát, mert napjainkban kiemelkedő szerepet tölt be a mindennapi életünkben a klímaváltozás. Átalakulnak az évszakok, a telek hőmérséklete szélsőséesebb, vagy túl enyhe idő van, vagy az átlagosnál sokkal hidegebb.

Az elmúlt években megfigyelhető a rezsicsökkentés fontossága is. A közintézményekben kötelezően elrendelt hőmérsékletet kell tartanunk. Dolgozatomban megvizsgálom, hogy ezt különböző módszerekkel hogyan lehet hatékonyan fenntartani.

Először általánoságban fogom leírni a rendszerek modellezését és szabályozását. Majd a Fuzzy rendszer alapjairól fogok írni.

A dolgozatomban részletesen be fogom mutatni a fűtésrendszer programjához felhasznált eszközöket, melyek a következők: Matlab Simulink, Matlab Fuzzy Logic Toolbox.

A Simulink programban létrehozok egy modellt a fűtés szabályozásához, amelyen belül háromféle szabályozási módszert fogok kialakítani. Ezek a következők: ON/OFF szabályozó, PID szabályozó és Fuzzy szabályozó. A modell kialakításánál figyelembe fogom venni a terem fizikai méretét, beleértve a falak és az ablakok felületét. Vizsgálom az elérni kívánt hőmérséklet (alapjel) beállítási lehetőségét, a kinti hőmérsékletingadozás határát.

Vizsgálni fogom, a háromféle megvalósított szabályozó rendszer szimulációjának eredményét, majd ezeket összehasonlítva értékelem ezeket.

2. A FELHASZNÁLT SZAKIRODALOM BEMUTATÁSA

2.1 Rendszerek modellezése és szabályozása

2.1.1 Rendszerről általánosságban

A rendszerek részei a mindennapi életünknek. Ezeket irányítani kell, hogy úgy viselkedjenek, hogy az elvárásoknak megfeleljenek.

A rendszer egy olyan halmaz, melyben összefüggő részek vagy komponensek tartoznak. Ezek az elemek együttműködnek egy közös cél elérése érdekében. Beszélhetünk fizikai, biológiai vagy elméleti rendszerekről. Napjainkban fontos szerepet tölt be az automatizálásban, de ha a mindennapi életünket nézzük, ott is egy irányítási folyamat megy végbe a szervezetünk működésének fenntartása érdekében. [1]

A rendszer elemei a következők:

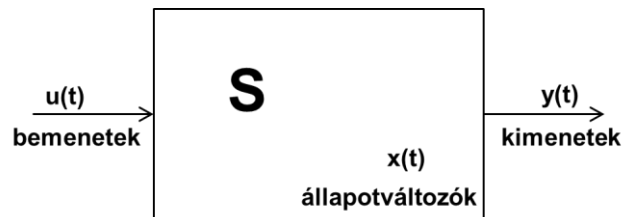
- Komponensek: olyan alkotóelemek, melyek a rendszer részei.
- Kapcsolatok: meghatározzák, hogy hogyan működjenek együtt a különböző alkotóelemek.
- Határok: elválasztják a rendszert a környezetüktől.
- Bemenetek és kimenetek: a környezet és a rendszer közötti interakció. A bemenetek jelek vagy erőforrások lehetnek, melyek hatnak rendszerre, a kimenetek is jelek és erőforrásokból állnak, ezek viszont a rendszer hatását írják le.

A rendszerek jellemzői:

- Célorientáltság: minden rendszernek van egy célja és a rendszer feladata, hogy ez a cél megvalósuljon. Például egy gyártósor működése, algoritmusok vagy szoftverek működése.
- Rendezettség: a rendszerek valamilyen hierarchiával, struktúrával rendelkeznek, melyek meghatározzák a rendszer elemei közötti kapcsolatot.
- Komplexitás: a rendszerek összetettsége függ a komponensek számától és a kapcsolatok bonyolultságától.

- Dinamika: A rendszer időbeni változása és a bemenetek reakcióinak változása folyamatosan alakítják a rendszer állapotát.

Egy dinamikus (időben változó) rendszer (system) sematikus vázlatát mutatja be a 2.1. ábra.



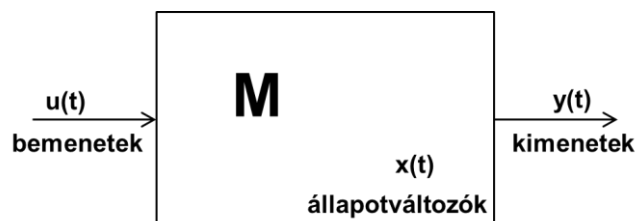
2.1. ábra Rendszer és elemei

A rendszer jelei időfüggő jelek, amelyek a bemenetek (u), kimenetek (y) és az állapotváltozók (x). A rendszer tulajdonképpen egy átalakítást végez, amely során a bemenő jeleit feldolgozva kimenő jeleket állít elő.

A rendszerek lehetnek egyszerű és összetett rendszerek is. Az összetett rendszerek több egyszerű rendszerből állnak, melyek kapcsolódnak egymáshoz. Például a rendszer egyik kimenő eleme lehet a másik rendszer bemenő eleme.

2.1.2 Rendszer modellje

A vizsgálat céljának megadása után következik a modell megválasztása. A modell az általunk elképzelt, észlelt valóságot ábrázolja. Feladatuk, hogy a valóságot tükrözve, a rendszerek irányítása úgy működjön, hogy jó döntéseket hozzanak. A modellek matematikai formulákkal is megadhatók. Fontos, hogy a modell kapcsolódjon a környezetéhez. Egy dinamikus rendszer modellje egy speciális célra készített leképezés, amely a bemenő jelek és a kimenő jelek közötti kapcsolatot adja meg, ahogy ezt a 2.2. ábra mutatja. [1]



2.2. ábra Egy rendszer modellje

Egy modell gyakran egy rendszer valamilyen matematikai konstrukcióval leírt reprezentációja egy megadott célra.

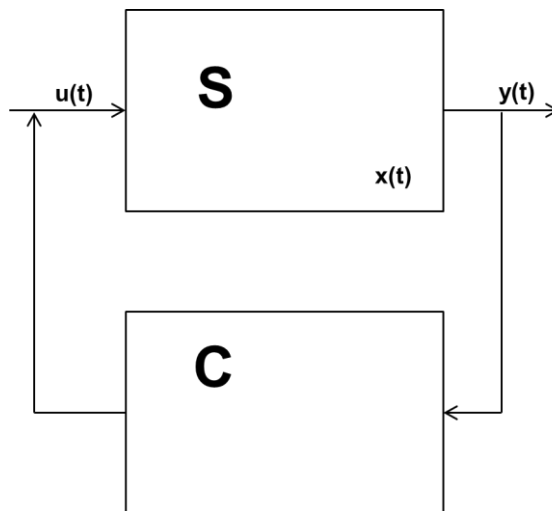
Minden rendszernek van határa. A vizsgálat célja határozza meg, hol legyen ez a határ.

A tervezési, üzemeltetési feladatok megoldása differenciálegyenletek segítségével valószínűsíthető meg. Taylor és Fourier sorok és azok különböző transzformáltjaival írhatók fel az időbeni és gyakorisági elemzések. Az időtartomány ábrázolása Descartes koordináta rendszerben történik, míg a frekvenciák szerinti ábrázolás a komplex változók vektoreiben történik. [1]

2.1.3 Szabályozás

A szabályozás egy olyan folyamat, mely során egy rendszer viselkedését befolyásoljuk, úgy, hogy követelményeket definiálunk, annak érdekében, hogy a meghatározott célt elérjük. A rendszer összehasonlítja az alapjelet és a mért jelet és az eltérés alapján dönti el, hogy be kell-e avatkozni a folyamatba, attól függően, hogy mi a kimenőjel. Ha nagyobb az eltérés az alapjelhez képest, mint a beállított, akkor reagál a rendszer. A beavatkozó hatásra a rendszer általában késleltetve reagál. A szabályozási folyamat stabilitását nekünk kell biztosítanunk, figyelembe véve a rendszer statikus és dinamikus tulajdonságait. [1]

A leggyakrabban használt szabályozási struktúra a visszacsatolás, amelyet a 2.3. ábra mutat be. Ebben a struktúrában a (nyílt) rendszerünket kiegészítjük egy szabályozóval (controller), amelynek eredményeképpen egy zárt, összetett rendszert kapunk.



2.3. ábra Rendszer és szabályozó

A szabályozástechnikának több területe van. Ilyen például az adaptív szabályozás, mely során a rendszer változásaihoz alkalmazkodik a szabályozás úgy, hogy a szabályozót újra kell hangolni, vagy a robosztus szabályozás, mely során elfogadható viselkedést kell biztosítani a rendszernek úgy, hogy rendszer paraméterei adott tartományban változhassanak, ezt a szabályozó tervezésénél kell figyelembe vennünk.

2.1.4 PID szabályozó

Egy gyakran használt szabályozó a PID szabályzó, amely neve a Proporcionális, Integráló és Deriváló szavak rövidítését tartalmazza, ami azt jelenti, hogy ez a szabályozó rendszer képes a múltbeli, jelen és jövőbeni hibákra is reagálni.

A múltbeli hibát a hiba integráljával, a jövőbeni hibát pedig a hiba deriváltjával tudja módosítani. A proporcionális rész a jelenlegi hibára fókuszál, vagyis a kimenet és a célváltozó közötti különbségre. Ez a rész felelős azért, hogy az aktuális hiba nagysága arányos legyen a szabályzó által meghatározott korrekció mértékével. Így a hibát három csatornán keresztül tudja kezelni. Ezek a következők: proporcionális, integráló, valamint deriváló csatornán.

Ez a szabályzó napjainkban az egyik legelterjedtebb. Ez annak köszönhető, hogy egyszerű a struktúrája és az iparban előírt szabályozási követelményeket képes teljesíteni. Legelterjedtebb alkalmazási területei: a robotika, ipari vezérlések és automatizálás.

A PID szabályzó célja, hogy szabályozza és stabilizálja egy rendszer kimenetét egy előre meghatározott célváltozó értékhez képest.

A PID szabályzó egy szabályozott mért értéket összehasonlít egy alapjellel. A két mennyiség különbségeként hibajel az „e” keletkezik. Ennek segítségével számítható ki az új beavatkozó jel, melynek a jele az „u”. A beavatkozó jel hatására a szabályozott mennyiség értéke közelebb fog kerülni az alapjelhez. A PID szabályzó feladata, hogy előállítsa a beavatkozó jelet „e” függvényben. A szabályozott mennyiség minél gyorsabban és pontosabban kövesse le a megadott alapjel értékét. [2]

A P tag az arányosságot veszi figyelembe, a P szabályzó az aktuális hiba arányosságát használja a kimenet korrekciójára. Minél nagyobb a hiba, annál nagyobb a korrekció mértéke és fordítva is igaz.

Matematikai formában a következőképpen írható fel:

$$u(n) = K_P e(n)$$

ahol:

u : a szabályzó kimeneti értéke

e : a mindenkori hibajel, azt mutatja meg hogy mekkora az eltérés az alapjeltől

K_P : a P tag együtthatója, amely meghatározza a korrekció mértékét.

Az egyenletből felírható a P tag átviteli függvénye, amely

$$H(s) = \frac{u(n)}{e(n)} = K_P$$

A „ K_P ” tagot az egyenletben erősítésnek is hívhatjuk, mivel a vezérlő jel a hibajel „ K_P ”-szereze lesz. [2] A P tag beállítása nagyon fontos a rendszer optimális működéséhez. A megfelelő értéket próbálkozásokkal vagy hiba kereséssel lehet megtalálni. A túl kicsi érték lassú, esetleg túl kevés korrekciót eredményezhet, amíg a túl nagy értékek túllövéseket vagy instabilitást okozhatnak. Azonban a P szabályzó nem képes megszüntetni a hibát, sokkal jobban működik a rendszer, ha az integráló és a deriváló részeket is figyelembe vesszük.

A szabályozó második tagja az integráló tag. Hibának nevezzük azt a különbséget, mely a referencia jel és a kimenő jel állandósult különbsége. Ez egy nagyon kicsi érték, a P tag nem képes kezelni, ezért kell bevezetni egy olyan járulékos tagot, ahol az idő során összegyűlt kis hibajel ily módon javíthatóvá válik. Ezt integrálással tudjuk megoldani, ezért figyelembe kell vennünk az I tagot. Ez a tag olyan járulékos vezérlőjel, amely a hibák korábbi időbeli összegétől függ.

Az integráló tag átviteli függvénye a következőképpen írható fel:

$$H(s) = \frac{1}{T_I s}$$

ahol T_I az integráló tag időállandója.

Az integráló tagnak fontos szerepe van a PID szabályzóban a rendszer stabilitásában és a pontos célváltozások követésében.

A PID szabályzó utolsó tagja a deriváló tag. A hibajel változási sebességétől függően a deriváló tag egy járulékos vezérlő jelet küld a rendszernek. A D tag a követési tulajdonságokat javítja a rendszer állapotában bekövetkezett gyors változások esetén.

Minél gyorsabban változik az aktuális hiba, annál nagyobb a javítás értéke a D tagban. A D tagot magában nem használjuk, főleg, ha túl nagy, mert a rendszernek instabilitást okozhat. PD vagy PID szabályzóként használható csak. A derivatív tagra jellemző, hogy zajérzékeny, ami azt jelenti, hogy zajforrások esetén nagyobb lehet a szabályzó kimenetének ingadozása. A D tag segít csökkenteni a rendszerben a túllövéseket, mivel gyorsan tud reagálni a hiba változására és ezáltal javítja a kimenetet.

A deriváló tag átviteli függvénye:

$$H(s) = T_D s$$

ahol T_D a differenciáló tag időállandója.

A három tagból álló PID szabályozó átviteli függvénye pedig a következő:

$$H(s) = K_P \left(1 + \frac{1}{T_I s} + T_D s \right)$$

ahol a szabályozó paraméterei a K_P erősítés, a T_I integrálási idő és a T_D differenciálási idő.

2.2 Fuzzy rendszerek

A fuzzy logikai rendszerek olyan számítási módszereket alkalmaznak, melyek nemcsak a hagyományos igaz/hamis logikát használják, hanem a nyelvi változókat, az ezekben rejlő bizonytalanságot is kezelik. A fuzzy rendszerek a fuzzy halmazokból épülnek fel, melyek elemei nemcsak 0 vagy 1 értékeket vehetnek fel, hanem bármilyen köztes értéket is a halmazhoz tartozás erősségének megadásához.

A fuzzy logika bevezetésének az egyik fő célja az volt, hogy jobban tudjuk modellezni a bizonytalan fogalmakat, a természetes nyelvi fogalmakat és az emberi gondolkodást. A nem éles határokkal jellemezhető fuzzy halmazokat Zadeh írta le egy 1965-ben megjelent cikkében [3], majd 1973-ban ő alkotta meg a nyelvi változók használatával az első fuzzy következtető algoritmust. Ez még nagyon bonyolult volt és nagy volt a számításigénye, azonban 1975-ben Mamdani újra fogalmazta a nyelvi változókat és az új megoldásnak köszönhetően drasztikusan csökkent a számításigény. [4]

2.2.1 Fuzzy halmazok és alapvető típusaik

Gyakran találkozhatunk olyan kifejezésekkel, amiről nem lehet egyértelműen meghatározni, hogy igazak vagy hamisak, elemei vagy nem elemei egy adott halmaznak. Ha például szeretnénk leírni a „kellemes szobahőmérséklet” halmazát, a klasszikus halmazelmélet szerint kellene definiálnunk egy éles halmazt, például egy minimum és egy maximum érték megadásával, amelyek által meghatározott intervallumba eső hőmérséklet értékek beletartoznak a halmazba, az intervallumon kívül eső hőmérséklet értékek pedig nem. Itt egyrészt kérdés lehet, hogy melyek legyenek ezek a kitüntetett értékek. Ha egyáltalán sikerült ebben megállapodásra jutni, akkor felmerülhet egy további kérdés, hogy a megadott minimum értéknél például 0.1 értékkel kisebb, vagy a megadott maximum értéknél 0.1 értékkel nagyobb hőmérséklet érték már nem tartozik a „kellemes szobahőmérséklet” halmazába.

Az ilyen típusú kifejezések matematikai leírására használhatók a fuzzy halmazok, amelyekben a halmazba tartozás egy tagsági függvénnyel írható le:

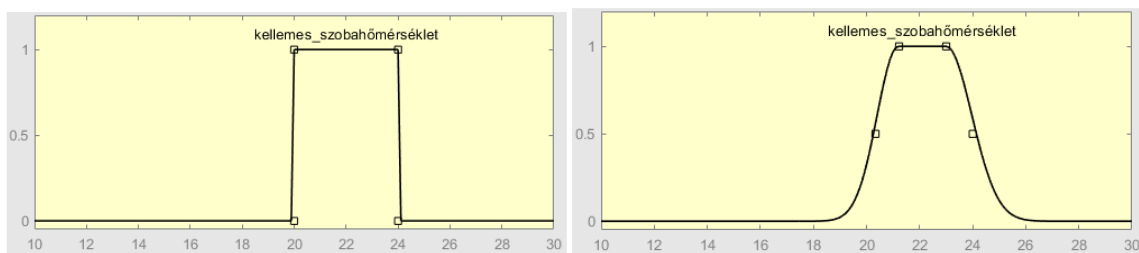
$$\mu_A(x): U \rightarrow [0, 1]$$

ahol μ jelenti a tagsági függvényt, A a fuzzy halmazt, U az univerzumot, x pedig az univerzum egy elemét.

A tagsági függvény definíciójában használt fogalmak a következők:

1. Alaphalmaz: Az alaphalmaz (vagy univerzum) az a halmaz, amelyeken a fuzzy halmaz(oka)t értelmezzük. Az univerzum mindig éles halmaz. Ez esetünkben a hőmérséklet univerzum.
2. Fuzzy halmaz: A tagsági függvény által definiált halmaz, amely lehetővé teszi az értelmezési határok „életlenítését” egy adott halmazban. Ez a példában szereplő „kellemes szobahőmérséklet” halmaza.
3. Tagsági függvény: Egy olyan függvény, amely az alaphalmaz minden eleméhez egy tagsági értéket rendel. A tagsági függvények legtöbbször használt értéktartománya a $[0, 1]$ intervallum közé eső érték.

A tagsági függvénnyel leírt fuzzy halmaz és a klasszikus halmazelmélet éles halmaza közötti különbséget mutatja a 2.4. ábra egy konkrét példán, a hőmérséklet univerzumon értelmezett „kellemes szobahőmérséklet” halmazok definiálásával.

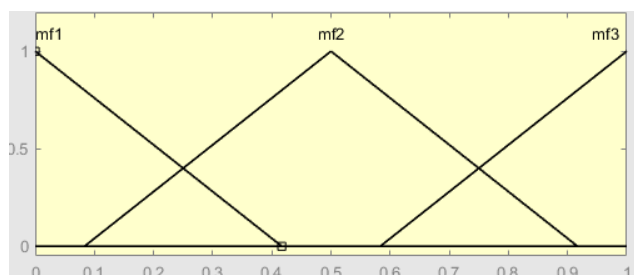


2.4. ábra Éles halmaz és fuzzy halmaz

Természetesen sokféleképpen lehetne leírni a „kellemes szobahőmérséklet” fuzzy halmazt, ez is azt jelenti, hogy ez a nyelvi fogalom bizonytalan, életlen, amolyan fuzzy.

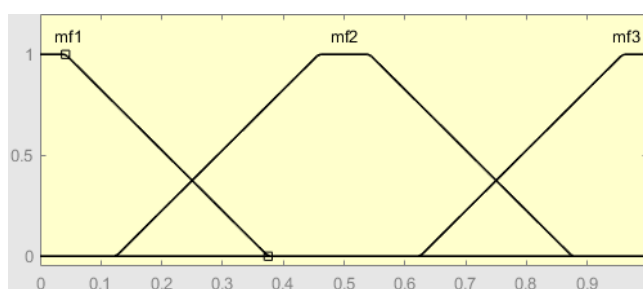
A fuzzy halmazok különböző típusú tagsági függvényekkel írhatók le, amelyek közül a leggyakrabban használtak a következők:

- háromszög alakú (2.5. ábra)



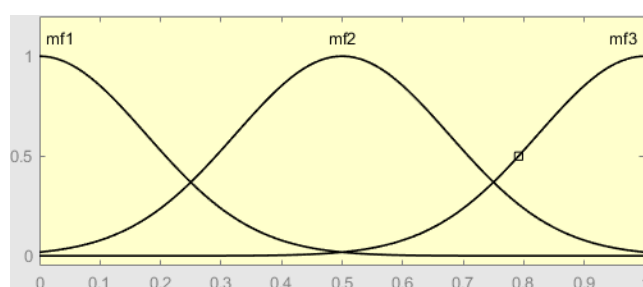
2.5. ábra Háromszög alakú tagsági függvények

- trapéz alakú (2.6. ábra)



2.6. ábra Trapéz alakú tagsági függvények

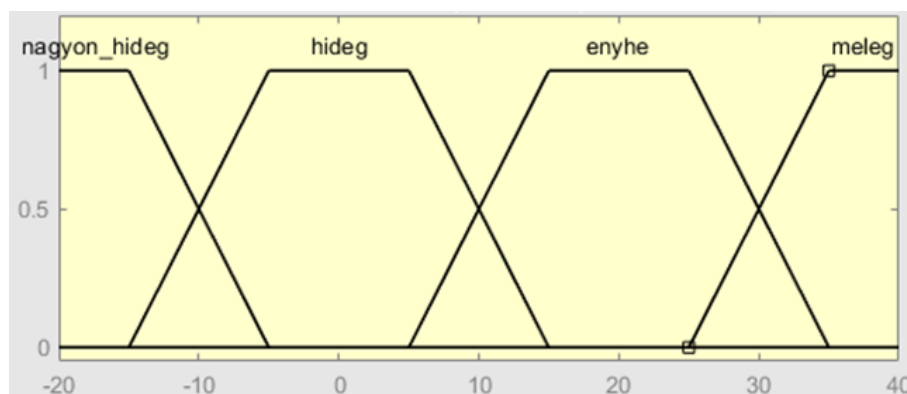
- Gauss görbe (2.7. ábra)



2.7. ábra Gauss görbe tagsági függvények

A tagsági függvények további típusai lehetnek a haranggörbék, szigmoid tagsági függvények, szakaszonként lineáris függvények, valamint diszkrét értékű univerzum esetén singleton-ok. [5]

Egy univerzumon több fuzzy halmaz is definiálható, amelyre a 2.8. ábra mutat be példát. A hőmérséklet univerzumon értelmezett négy fuzzy halmaz a példában a következő: nagyon_hideg (A1), hideg (A2), enyhe (A3) és meleg (A4).



2.8. ábra Hőmérséklet univerzumon értelmezett fuzzy halmazok

A példa fuzzy halmazai trapéz alakú tagsági függvényekkel reprezentáltak és az alábbi formulákkal adhatók meg:

$$A1 = \begin{cases} 1, & \text{ha } x \leq -15^{\circ}\text{C} \\ (-5^{\circ}\text{C} - x)/10^{\circ}\text{C}, & \text{ha } -15^{\circ}\text{C} < x < -5^{\circ}\text{C} \\ 0, & \text{ha } x \geq -5^{\circ}\text{C} \end{cases}$$

$$A2 = \begin{cases} 0, & \text{ha } x \leq -15^{\circ}\text{C} \text{ vagy } x \geq 15^{\circ}\text{C} \\ (x - (-15^{\circ}\text{C}))/10^{\circ}\text{C}, & \text{ha } -15^{\circ}\text{C} < x < -5^{\circ}\text{C} \\ 1, & \text{ha } -5^{\circ}\text{C} \leq x \leq 5^{\circ}\text{C} \\ (15^{\circ}\text{C} - x)/10^{\circ}\text{C}, & \text{ha } 5^{\circ}\text{C} < x < 15^{\circ}\text{C} \end{cases}$$

$$A3 = \begin{cases} 0, & \text{ha } x \leq 5^{\circ}\text{C} \text{ vagy } x \geq 35^{\circ}\text{C} \\ (x - 5^{\circ}\text{C})/10^{\circ}\text{C}, & \text{ha } 5^{\circ}\text{C} < x < 15^{\circ}\text{C} \\ 1, & \text{ha } 15^{\circ}\text{C} \leq x \leq 25^{\circ}\text{C} \\ (35^{\circ}\text{C} - x)/10^{\circ}\text{C}, & \text{ha } 25^{\circ}\text{C} < x < 35^{\circ}\text{C} \end{cases}$$

$$A4 = \begin{cases} 0, & \text{ha } x \leq 25^{\circ}\text{C} \\ (x - 25^{\circ}\text{C})/10^{\circ}\text{C}, & \text{ha } 25^{\circ}\text{C} < x < 35^{\circ}\text{C} \\ 1, & \text{ha } x \geq 35^{\circ}\text{C} \end{cases}$$

2.2.2 Műveletek fuzzy halmazokkal

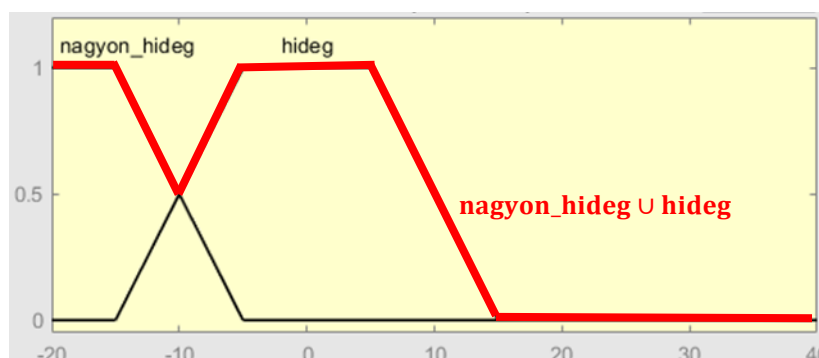
A fuzzy halmazokra vonatkozóan többféleképpen lehet a hagyományos halmazokon alkalmazott három alpműveletet (unió, metszet, komplementer képzés) elvégezni a tagsági függvények segítségével. A gyakorlatban a legtöbbet alkalmazott az úgynevezett Zadeh-féle, amelyet a következőkben részletesen bemutatok.

Unió (egyesítés, diszjunkció)

Az unió a halmazelméletben két vagy több halmazból képez egy olyan új halmazt, melyben a kiindulási halmazok összes eleme benne van. Fuzzy halmazok esetén az azonos univerzumon értelmezett halmazok uniója a tagsági függvényekből elemenkénti maximumképzéssel határozható meg. Például a „nagyon hideg” unió „hideg” fuzzy halmaz a

$$\mu_{nagyon_hideg \cup hideg}(T) = \max \{ \mu_{nagyon_hideg}(T), \mu_{hideg}(T) \}$$

összefüggéssel határozható meg, amelyet a 2.9. ábra szemléltet.



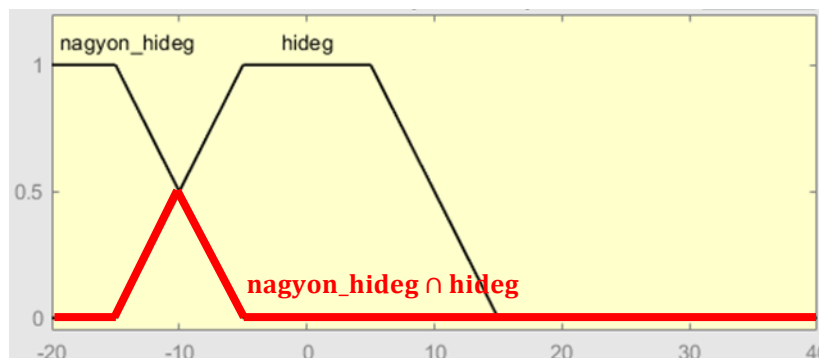
2.9. ábra Fuzzy halmazok uniója

Metszet (konjunkció)

A halmazelméletnek egy olyan művelete a metszetképzés, mely során két vagy több halmazból úgy képzünk egy új halmazt, hogy a létrejövő halmaz csak azokat az elemeket tartalmazza, amelyek mindegyik kiinduló halmaznak az elemei voltak. Azonos univerzumon értelmezett fuzzy halmazok metszetének meghatározása a tagsági függvényekből minimumképzéssel határozható meg. Például a „nagyon hideg” unió „hideg” fuzzy halmaz a

$$\mu_{\text{nagyon_hideg} \cap \text{hideg}}(T) = \min \{ \mu_{\text{nagyon_hideg}}(T), \mu_{\text{hideg}}(T) \}$$

összefüggéssel határozható meg, amelyet a 2.10. ábra mutat be.



2.10. ábra Fuzzy halmazok metszete

Komplementer

Egy A halmaz komplementere (kiegészítő halmaza) olyan halmaz, amely tartalmazza az univerzumban szereplő, de az A halmazban nem szereplő elemeket.

Egy fuzzy halmaz komplementerének meghatározását a μ_A tagsági függvényből az elemenkénti $1 - \mu_A$ érték számításával határozható meg. Például a „hideg” fuzzy halmaz komplementere:

$$\mu_{\neg \text{hideg}}(T) = 1 - \mu_{\text{hideg}}(T)$$

amelyet a 2.11. ábra szemléltet.



2.11. ábra Fuzzy halmaz komplementere

A halmazelméleti alapműveleteken kívül további műveleteket (például reláció, kompozíció) is értelmezhetünk fuzzy halmazokon. [4]

2.2.3 Fuzzy szabályok

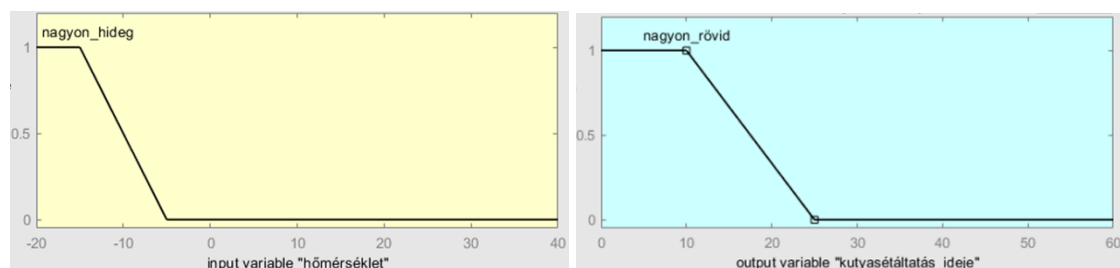
A fuzzy halmazok használatával fuzzy szabályok készíthetők, amelyek

HA $x = A$

AKKOR $y = B$

alakú feltételes állításokkal fogalmazhatók meg, ahol A és B fuzzy halmazok, x és y pedig a megfelelő univerzumok értékei.

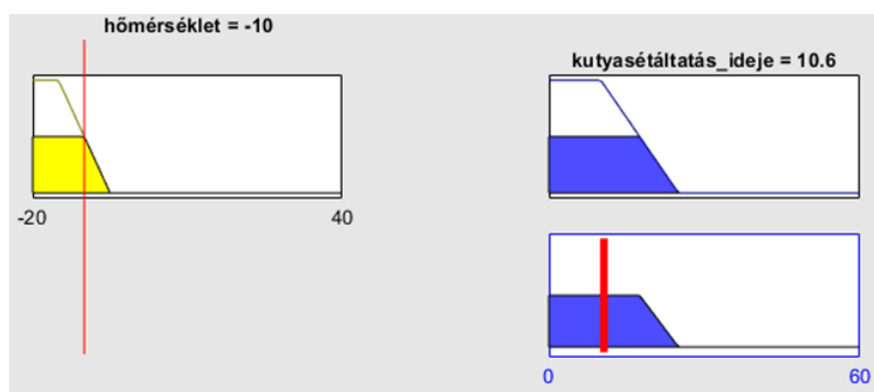
Például a „HA a hőmérséklet nagyon hideg, AKKOR nagyon rövid ideig sétáltassuk a kutyát.” szabály esetén x a hőmérsékletet, y a kutyasétáltatás idejét adja meg, A a „nagyon hideg”, B pedig a „nagyon rövid” fuzzy halmazokat jelenti. A feltételi és következmény részével definiált szabályt a 2.12. ábra mutatja.



2.12. ábra Fuzzy szabályt definiáló fuzzy halmazok

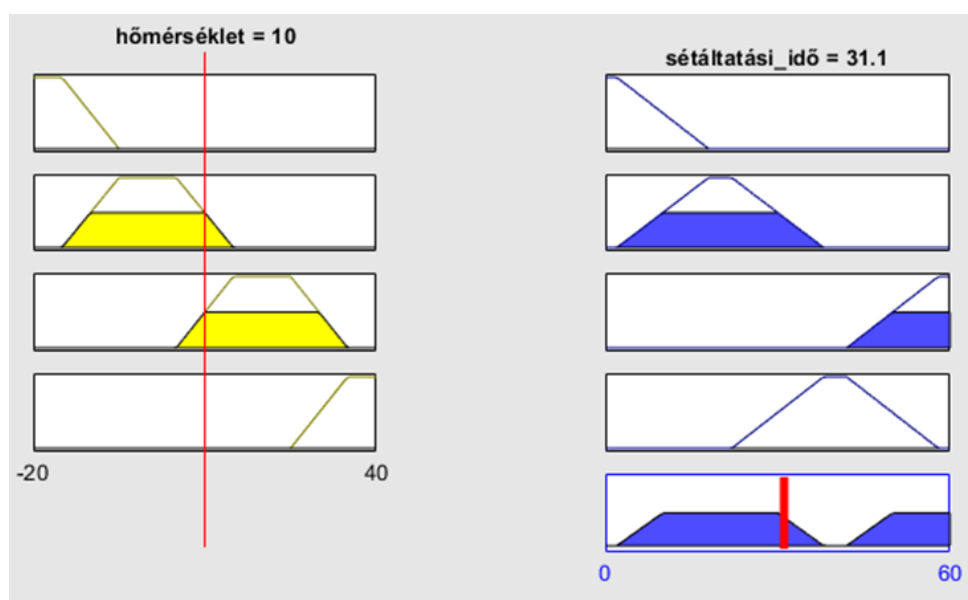
Egy „közönséges” szabály akkor végrehajtható (tüzelőképes), amennyiben a feltételi rész igaz, s a tüzelés eredményeképpen a következmény rész végrehajtásra kerül (értéke igaz lesz).

Egy fuzzy szabály csak bizonyos mértékig, azaz részben tüzel, ezért a következmény rész is csak bizonyos mértékben lesz igaz. Ennek oka, hogy egy fuzzy szabály esetén a feltételi rész egy 0 és 1 közötti értékkel lesz igaz, amelyet az aktuális bemeneti érték határoz meg. Ez az úgynevezett illeszkedés mértéke. Egy szabály tüzelésének feltétele, hogy az illeszkedés mértéke és a feltételi részben levő fuzzy halmaz metszete ne legyen üres, a következmény értéke pedig a következmény részben levő fuzzy halmaz illeszkedés mértékével kombinált mértéke (minimum képzés). Egy fuzzy szabály működését mutatja a 2.13. ábra.



2.13. ábra Fuzzy szabály végrehajtása

Több fuzzy szabállyal történő következtetés esetén előállhat, hogy egy adott bemenetre több szabály is tüzel. Ebben az esetben a szabályok kiértékelését egyesével végezzük el, majd az így kapott fuzzy következtetéseket összevonjuk, amelyhez például a fuzzy unió műveletét használhatjuk. Ezt mutatja be a 2.14. ábra, ahol a korábbi példa folytatásaként három további szabályt készítettem („HA a hőmérséklet hideg, AKKOR rövid ideig sétáltassuk a kutyát.”, „HA a hőmérséklet enyhe, AKKOR hosszú ideig sétáltassuk a kutyát.” és „HA a hőmérséklet meleg, AKKOR átlagos ideig sétáltassuk a kutyát.”).



2.14. ábra Következtetés fuzzy szabályhalmazon

A szabályok egy adott feladatra vonatkozó szakértői ismereteket, gyakorlati tapasztalatokat is megfogalmazhatnak. Egy adott feladatra alkotott szabályok halmaza egy szabálybázist alkot, amelynek használatával, fuzzy következtetések segítségével az adott rendszer viselkedése meghatározható. Egy ilyen fuzzy modell előnye, hogy akár rendszerelméleti vagy irányításelméleti ismeretek nélkül, az emberi gondolkodásmódhoz hasonlóan, nyelvi kifejezésekkel írható le.

2.2.4 Fuzzy szakértő rendszer

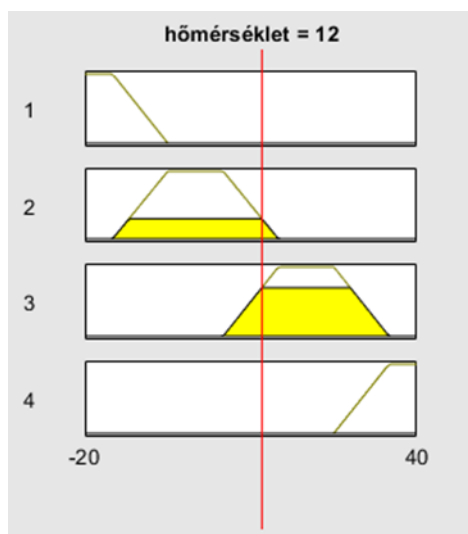
A fuzzy szakértő rendszer egy olyan szakértő rendszer, amelynek tudásbázisát fuzzy halmazokon értelmezett fuzzy szabályok alkotják, a következtető gépe pedig fuzzy bemeneteket képez le fuzzy kimenetekké a fuzzy szabályok használatával, amelyben a fuzzy következtetések olyan logikai folyamatok, amelyek a fuzzy logika alapelveire épülnek, valamint lehetővé teszik a nem-lineáris, ezen kívül bizonytalan összefüggések modellezését.

Amennyiben ezt a rendszert valós bemenetekkel és kimenetekkel működtetjük, akkor a valós (éles) bemenetet fuzzy bemenetté átalakító fuzzifikáló, illetve a fuzzy kimenetet valós (éles) kimenetté átalakító defuzzifikáló egységekkel kell kiegészíteni.

Ennek megfelelően a fuzzy szakértő rendszerek jellemzően a következő komponensekből állnak:

1. Fuzzifikáló komponens

A fuzzifikáció a valós, éles bemenetek fuzzy tagsági értékké transzformálását jelenti.



2.15. ábra Fuzzifikáció

A 2.15. ábra szemlélteti egy éles hőmérséklet érték fuzzy értékké történő átalakítását. Ahogy az ábrán látható, a hőmérséklet = 12°C input érték a „hideg” (2) és az „enyhe” (3) fuzzy halmazokba nullától különböző mértékkel tartozik (a „nagyon_hideg” (1) és a „meleg” (4) halmazba tartozás mértéke 0). A „hideg” halmazba tartozás meghatározása a $(15^{\circ}\text{C} - x)/10^{\circ}\text{C}$ összefüggéssel történik, ahol $x = 12^{\circ}\text{C}$, így $\mu_{hideg}(12^{\circ}\text{C}) = 0.3$. Az „enyhe” fuzzy halmazba tartozás számítását a $(x - 5^{\circ}\text{C})/10^{\circ}\text{C}$ képlettel végezhetjük el, amelynek számításával $\mu_{enyhe}(12^{\circ}\text{C}) = 0.7$ értéket kapunk. Tehát a fuzzifikálás eredménye a vizsgált 12°C hőmérsékletnél

$$\mu_{nagyon_hideg}(12^{\circ}\text{C}) = 0$$

$$\mu_{hideg}(12^{\circ}\text{C}) = 0.3$$

$$\mu_{enyhe}(12^{\circ}\text{C}) = 0.7$$

$$\mu_{meleg}(12^{\circ}\text{C}) = 0$$

2. Szabálybázis

A szabálybázis tartalmazza az általunk megadott szabályokat. Egy fuzzy szabály HA-AKKOR szabályként jelenik meg, amely tartalmaz egy feltételt és egy következtetést. A szabálybázis a fuzzy következtetési rendszer fontos részeként a kimeneti változókat határozza meg.

Például a 2.14. ábra szabályhalmaza a következő:

HA hőmérséklet = nagyon_hideg AKKOR sétáltatási_idő = nagyon_rövid

HA hőmérséklet = hideg AKKOR sétáltatási_idő = rövid

HA hőmérséklet = enyhe AKKOR sétáltatási_idő = hosszú

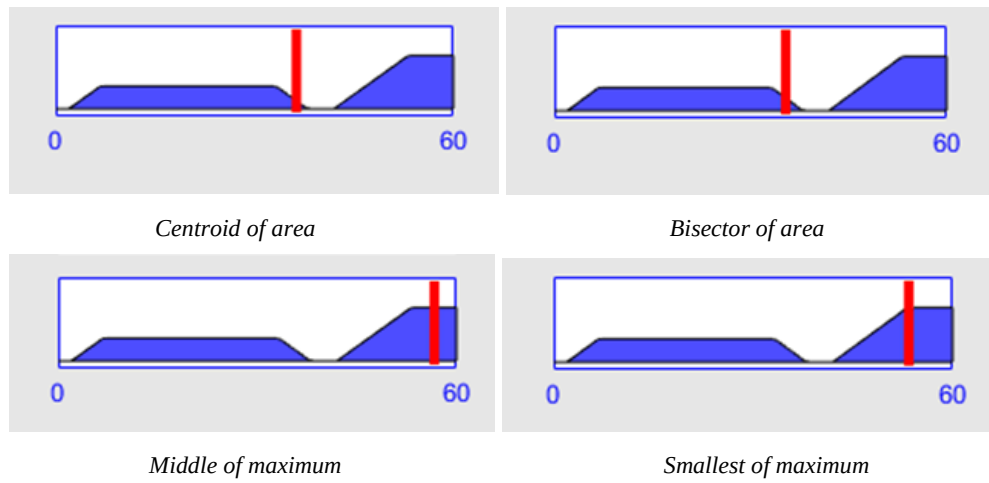
HA hőmérséklet = meleg AKKOR sétáltatási_idő = átlagos

3. Következtető komponens

A fuzzy szabályalapú következtetés három fő részből áll, ezek a fuzzy illesztés, a fuzzy következtetés és a fuzzy kombináció. A fuzzy illesztés azt határozza meg, hogy a bemeneti adatok milyen mértékben egyeznek meg a fuzzy szabályok feltételi részeivel. Ha több „és” – sel összekapcsolt feltétel szerepel az előfeltételben, akkor Mamdani módszer esetén a metszetet (konjunkciót) használjuk a feltételi részek kombinálására.. A fuzzy következtetés a megadott bemenetről a kimenetre való leképezés, amely tagsági függvényeket, fuzzy logikai operátorokat és HA-AKKOR szabályokat használ a végső eredmény megadásához, amelyek az egyezés mértékétől függnnek. Mivel minden definiált szabály kiértékelésre kerül, a szabályok sorrendje nem releváns. A fuzzy kombináció összesíti a fuzzy szabályokból származó összes kimenetet azok egyetlen fuzzy halmazba egyesítésével, amelyhez a fuzzy unió (diszjunkció) operátort alkalmazzák.

4. Defuzzifikáló komponens

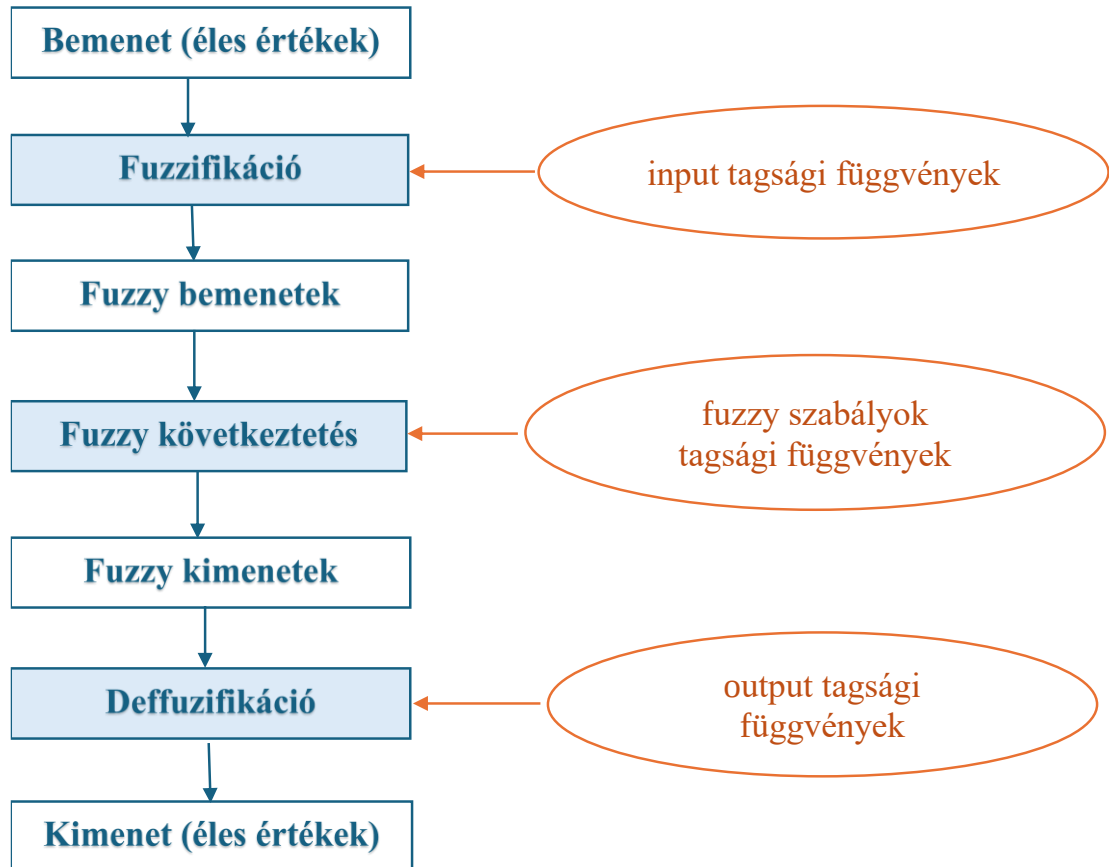
A defuzzifikálás a fuzzy rendszerek éles kimeneteinek előállítását jelenti az összesített fuzzy érték felhasználásával. Számos különböző defuzzifikációs módszer létezik. Ilyen például a Centroid of area (COA), Bisector of area (BOA), Middle of maximum (MOM), Smallest of maximum (SOM), amelyet a 2.16. ábra mutat be.



2.16. ábra Defuzzifikációs módszerek

Defuzzifikációra akkor van szükség, ha a fuzzy szakértő rendszer kimenetén éles jelet várunk, például ahogy a fuzzy szabályozó rendszerek esetén is.

Egy fuzzy szakértői rendszer felépítését a 2.17. ábra szemlélteti.



2.17. ábra Fuzzy szakértői rendszer

A fuzzy rendszerek használatának előnyei a következők:

- Emberi kifejezésekhez hasonló, nyelvi kifejezéseket használ.
- Szabálybázist használ, amelyben a szabályok és a fuzzy halmazok könnyen módosíthatók.
- A kimenet és bemenet közötti kapcsolat könnyen felépíthető és érthető, mert nyelvi kifejezésekkel adja meg.
- Bármely adott rendszerrel könnyen bővíthető, alakítható, nem kell előlről kezdeni.
- Nagyon egyszerű matematikai műveleteket használ.
- A pontatlan, zajos bemenetek esetében is könnyen használhatók.
- Tetszőleges bonyolultságú nemlineáris függvényeket is tud modellezni.

- Keverhető a hagyományos szabályozási technikákkal. A fuzzy rendszerek nem feltétlenül helyettesítik a hagyományos vezérlési módszereket, több esetben kiegészítik azt. [6]

A fuzzy logika nehézségei közé tartozik, hogy csak akkor érthető könnyen, ha egyszerű, valamint nincs szisztematikus módszer a fuzzy rendszerek kialakításához. [5]

A fuzzy szakértői rendszer alkalmazási területei szerte ágazóak. Használják az iparban, hőmérséklet szabályozására, sebességszabályozásra. Orvosi szakterületen találkozhatunk cukorbetegség, rák diagnosztizálásuk és kezelésük során, vérnyomás szabályozásnál. Elektronikai cikkeknel – háztartási gépek vezérlésénél, környezeti vezérlésnél. Otthoni fűtésrendszer vagy párástók vezérlésénél.

3. FELHASZNÁLT ESZKÖZÖK

3.1 Matlab Simulink

A Matlab Simulink egy vizuális programozási felület, mely dinamikus és beágyazott rendszerek szimulációs és modellalapú tervezési környezetben található. Az eszköz lehetőséget biztosít egyenletek numerikus megoldására grafikus felhasználói felület segítségével, melynek megvalósítása nem igényel kódolást. [7]

Interaktív grafikus környezetet és testre szabható blokk könyvtárakat biztosít, amelyek segítségével számos időben változó rendszer tervezhető, szimulálható, megvalósítható és tesztelhető, többek között kommunikációs, vezérlési, jelfeldolgozási, videófeldolgozási és képfeldolgozási rendszerek.

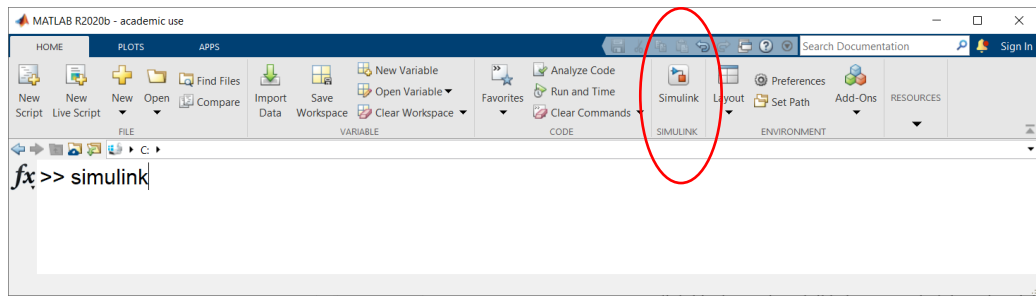
A Simulink modellekből épül fel. A modellek blokkokat, jeleket tartalmaznak. A blokkok matematikai függvényekből épülnek fel, különböző számú bemenettel és kimenettel rendelkeznek. A blokkokat összekötő vonalakat jeleknek nevezzük, melyek értékeket közvetítenek. Ezek a jelek különböző adatstruktúrákat például: mátrixokat, vektorokat, numerikus értékeket tudnak kezelni.

A modellhez megjegyzéseket lehet hozzáadni, ami lehet képi vagy szöveges. Ezek a megjegyzések nem használhatók fel a számításokhoz, de mások számára megkönnyíthetik a modellben szereplő műveletek megértését.

3.1.1 A Simulink használata

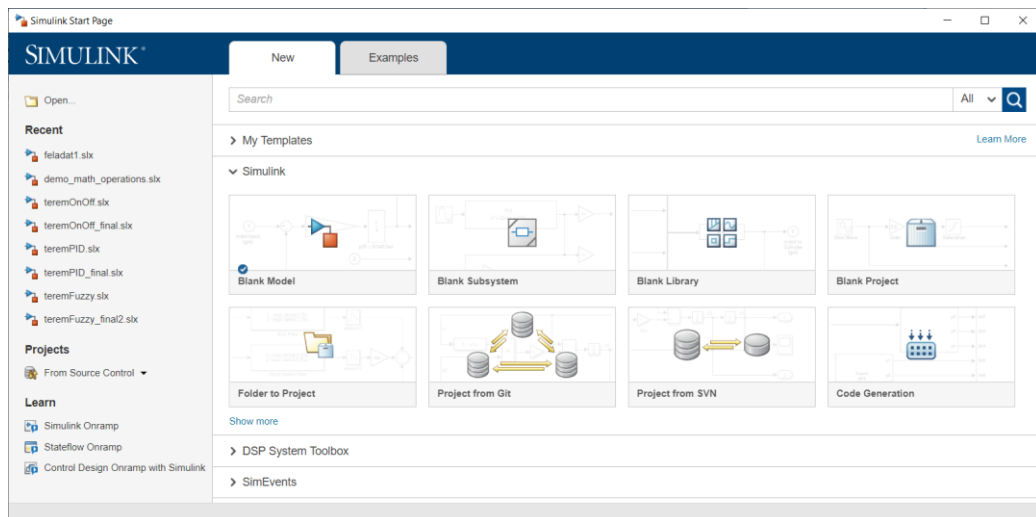
Simulink használatának alapvető lépései a következők:

- Simulink indítása: A Matlab megnyitása után két lehetőségünk van elindítani a programot. a Simulink ikonja segítségével vagy a Matlab parancssorába beírva a „simulink” szót, ahogy ezt a 3.1. ábra mutatja.



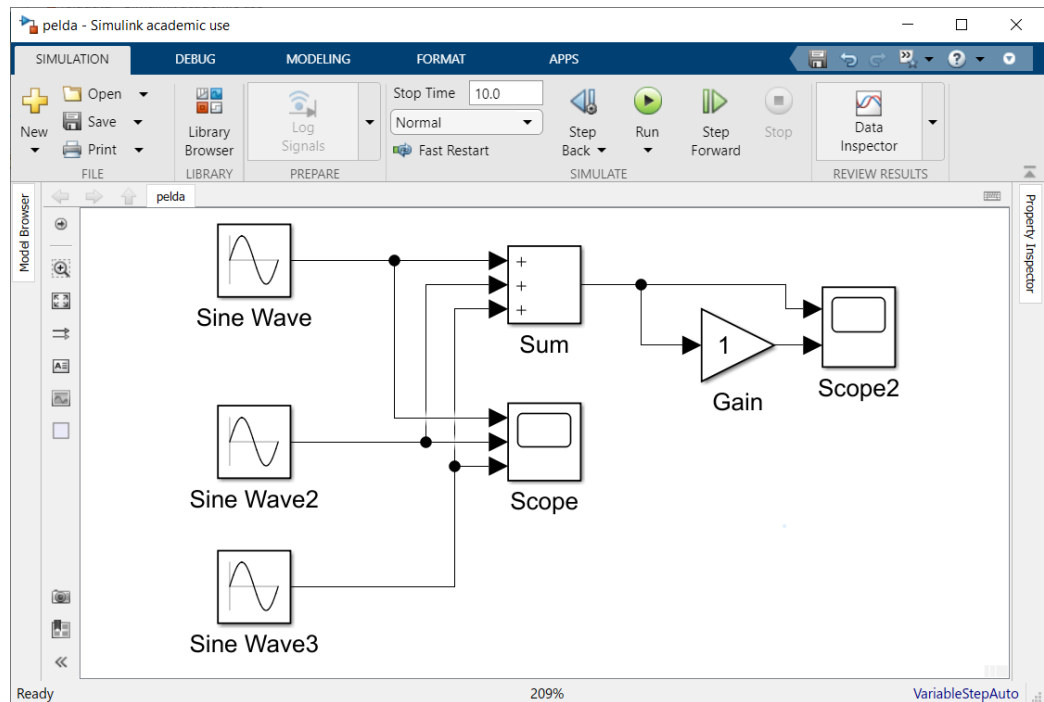
3.1. ábra Simulink indítása

- Modellek megnyitása: A Simulink nyitó oldalán (3.2. ábra) a bal oldali menüsorból lehet kiválasztani a legutóbb használt modellek vagy a könyvtárakban elmentett modellek közül azt, melyikkel szeretnénk dolgozni. Amennyiben új Simulink modellt, részrendszert, könyvtárat, projektet stb. szeretnénk létrehozni, a jobb oldalon található elemek közül a megfelelő kiválasztásával indíthatjuk azt.



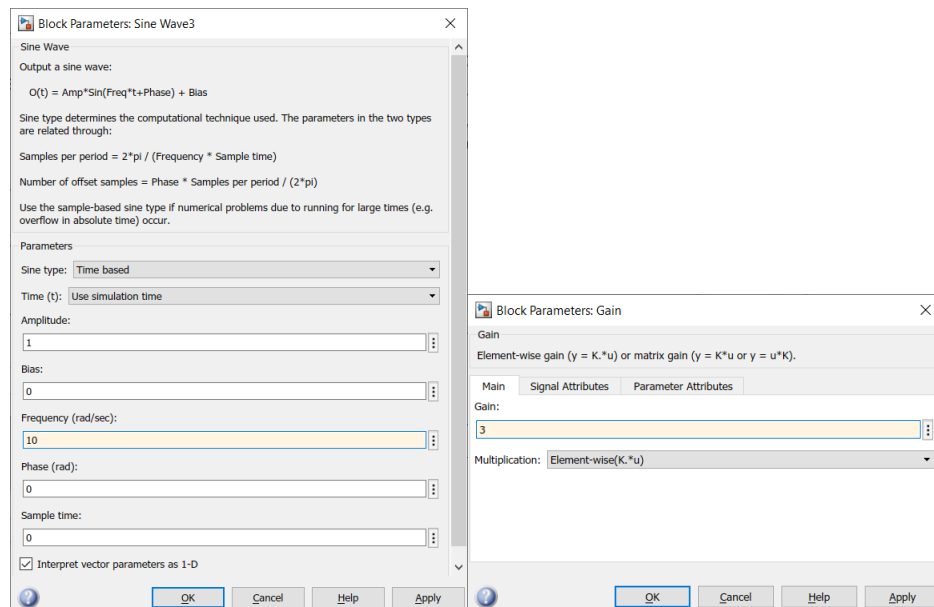
3.2. ábra A Simulink nyitó oldala

- Modell készítése: A Simulink Editorban (3.3. ábra) blokkok elhelyezésével és azok összekapcsolásával tudunk modellt készíteni. Az ábrán látható egyszerű modell három szinusz jelet tartalmaz, amelyek a „Scope”, valamint a „Sum” blokkok inputjai. Az „Sum” blokk összeadja a három bemenő jel értékét, ami egyrészt a „Gain”, másrészt a „Scope2” egyik bemenete. A „Gain” outputja pedig a „Scope2” másik bemenete.



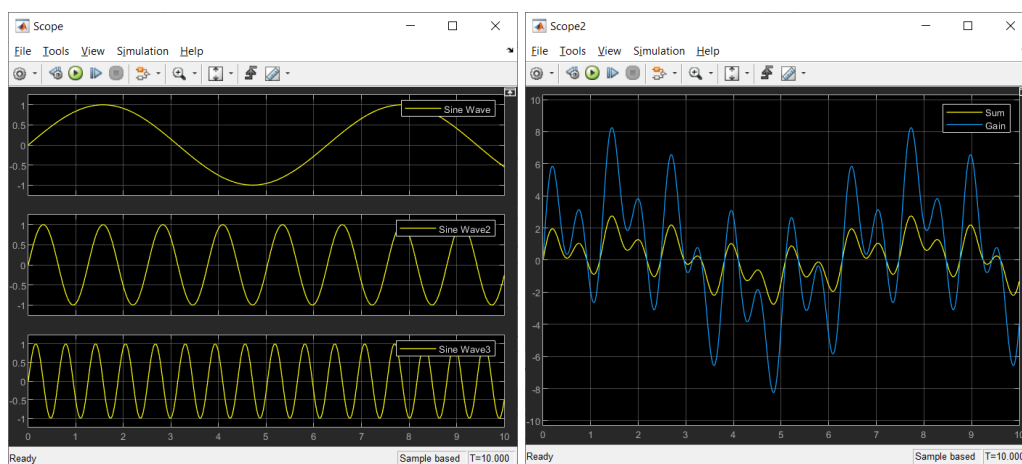
3.3. ábra Egyszerű Simulink modell

- Konfigurálás: Az adatok megadása, a blokkok paramétereinek beállítása történik ebben a szakaszban. Itt adhatjuk meg például a szinusz függvények tulajdonságait vagy az erősítés értékét, ahogy azt a 3.4. ábra mutatja.



3.4. ábra Blokkok paramétereinek megadása

- Szimuláció: Az elkészült modellünket tudjuk ebben a fázisban tesztelni. Itt tudjuk tesztelni, hogy az elkészített modell, hogy viselkedik az idő függvényében. Emellett vizsgálni tudjuk a kimeneti változók értékeit is.
- Analízis során az eredmények vizsgálata és elemzése történik. Grafikonon tudjuk megjeleníteni a kimeneti változók értékeit és elemzéseket tudunk végezni a modellünk viselkedéséről. A példa modell jeleinek időbeli viselkedését mutatja a 3.5. ábra.

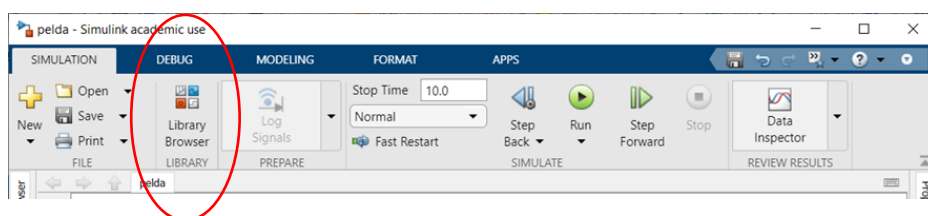


3.5. ábra Szimuláció eredménye

3.1.2 Simulink Blokkok

A következő fejezetben röviden bemutatom a blokkok készítésének lehetőségét egy modellen belül.

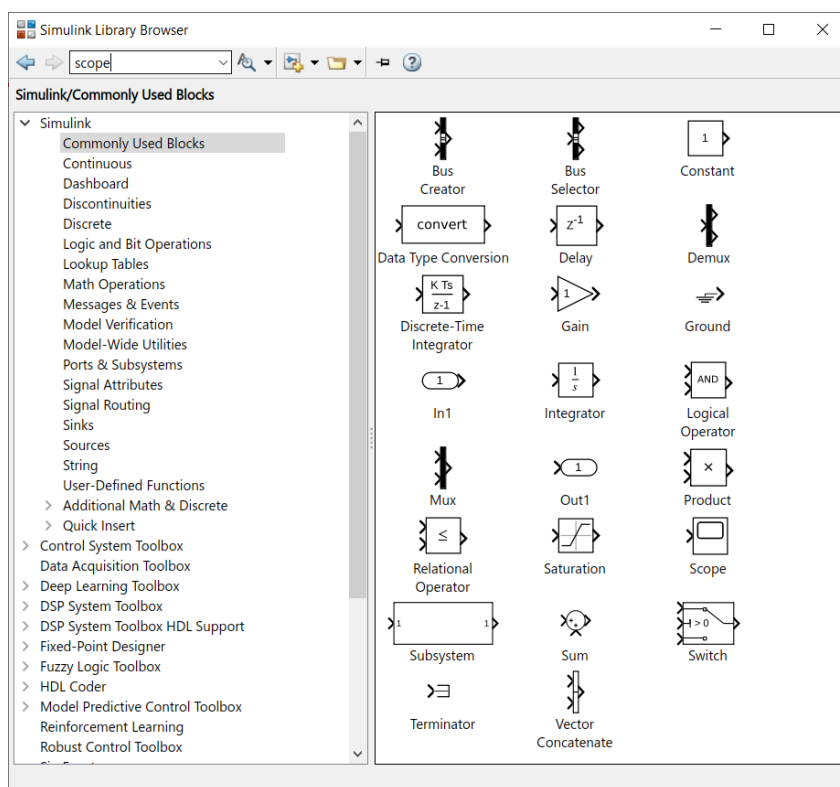
Egy blokkot a modellhez könyvtárböngészőből (Library Browser) történő kiválasztással lehet hozzáadni (3.6. ábra Library Browser megnyitása).



3.6. ábra Library Browser megnyitása

A böngésző megjeleníti a Simulinkben elérhető összes blokkot, mappák szerint csoportosítva. Ezt a 3.7. ábra szemlélteti. A böngésző bal felső sarkában található egy

gyorskeresősáv, ha név szerint szeretnénk rákeresni egy adott blokkra. Lehetőségünk van úgy is keresni, hogy lenyitjuk a könyvtárak bal oldalán található fület és ott megjelenik az adott mappába tartozó összes blokk és az alapján át tudjuk húzni a Simulink Editorba a kiválasztott blokkot.



3.7. ábra Simulink Library Browser

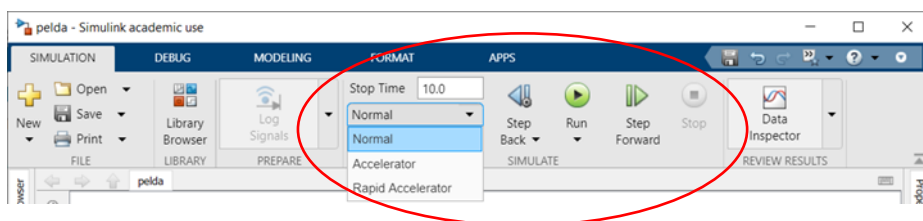
Az elhelyezett blokkok mozgathatók, forgathatók. Az összekapcsolásuk úgy történik, hogy az egyik blokk kimenetére kattintunk és azt egy másik blokk bemenetére húzzuk. A jelek, melyek két blokkot kötnek össze, kattintással és húzással áthelyezhetők. A meglévő jelekből ágakat hozhatunk létre a CTRL billentyű lenyomásával. Az ágak olyan elemek a Simulink modellben, amelyek lehetővé teszik a jelutak megosztását vagy összekapcsolását. Az ágak segítségével létrehozhatunk több útvonalat a modellben. Ezzel lehetővé tesszük a jelek szétválasztását, összekapcsolását vagy átirányítását különböző blokkok között. Ennek köszönhetően a modell struktúrája és működése hatékonyabb és rugalmasabb lesz.

Minden blokknak saját beállítási lehetőségei vannak. Ezeket a blokk paraméterek és a blokk tulajdonságai között találjuk meg. A Simulink modellek tartalmazhatnak Matlab változókat is, de arra ügyelni kell, hogy ha ilyet szeretnénk, ezt a Matlab munkaterületen kell definiálnunk.

3.1.3 A szimuláció és paraméterei

A Simulink modell futtatásával kapcsolatos paraméterek és azok beállítása a következő:

- Megoldó és beállításai: A modell egyenleteinek numerikus megoldásához, az állapotváltozók számításához különböző megoldókat használhatunk. Ezek kiválasztása és paramétereinek beállítása befolyásolja többek között a megoldás pontosságát és időigényét. A megoldók egyik fontos paramétere a lépéshossz (step size).
- Modell futtatása: A szimuláció a „Run” gombbal indítható el, a „Stop Time” beállításával pedig megadhatjuk másodpercben, hogy mennyi ideig történjen a szimuláció. (3.8. ábra)



3.8. ábra Szimuláció indítása

Különböző szimulációs üzemmódokat van lehetőségünk beállítani. Ezek a következők:

- Normal: A normál mód az alapértelmezett módja a Simulink-ben annak, hogy egy modellt szimuláljunk. Ez az alapértelmezett mód, amelyben a modellt feldolgozzuk és futtatjuk Matlab környezetben interpretált műveletek használatával. Ez a mód általában lassabb lehet, mint a további pontokban bemutatott másik kettő, legfőképp nagyobb és összetettebb modellek esetén. A normál szimulációs mód használatakor nem kell külön beállításokat véggezni a Simulink-ben.
- Accelerator: másnéven gyorsító, a Simulink olyan szimulációs módja, amely lehetővé teszi a szimulációk gyorsabb végrehajtását, elsősorban

nagyobb és bonyolultabb modellek esetén. A gyorsító módban a modellek által generált kódot futtatja, ami hatékonyabb és gyorsabb, mint a normál mód.

Az Accelerator mód előnyei a következők:

- Gyorsabb szimulációk: A gyorsító mód jelentősen felgyorsíthatja a szimulációk futását, különösen nagyobb és összetettebb modellek esetén.
- Hatékonyabb erőforrásfelhasználás: Az Accelerator lehetővé teszi a számítógép hardveres erőforrásainak hatékonyabb kihasználását a modellek szimulálása során.
- Valós idejű szimulációk: A gyorsító mód alkalmazása lehetővé teszi a valós idejű szimulációkat olyan alkalmazásokban, ahol az időkritikus válaszidők fontosak.

A gyorsító módban a Simulink előzetesen generálja a modellből származó C kódot, majd ezt a kódot futtatja. Ez általában hatékonyabb, mint az interpretált műveletek használata, amelyek jellemzőek a normál szimulációs módban.

- Rapid Accelerator: egy speciális mód a Simulinkben, amely még hatékonyabbá teszi a szimulációs folyamatot, különösen a gyorsító módban. A Rapid Accelerator lehetővé teszi a modell generált kódjának előzetes fordítását, majd futtatását újrahasznosítható formában. Ezáltal még tovább nő a szimuláció sebessége, valamint hatékonysága.

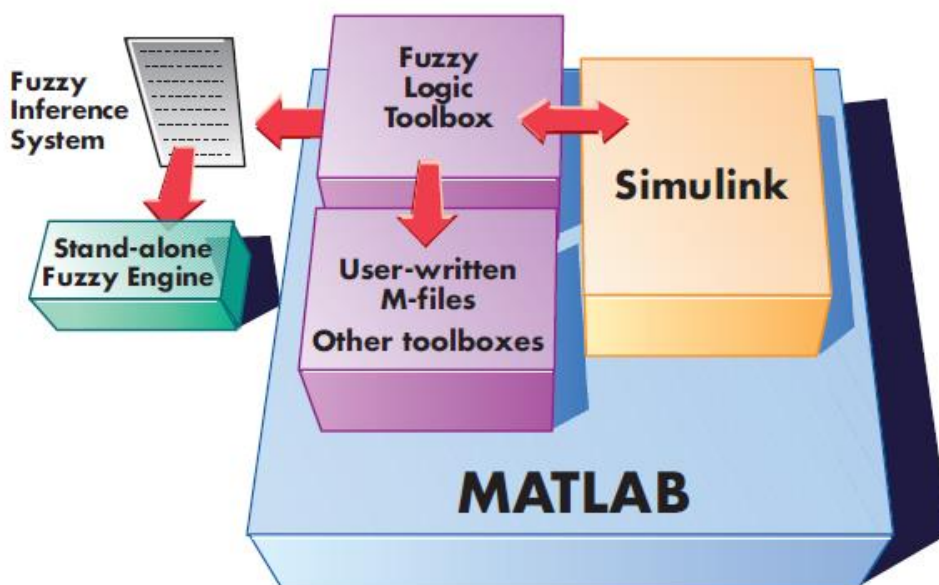
A Rapid Accelerator alkalmazásának folyamata:

1. Fordítás előtt: A Simulink először előállítja a modell által generált kódot. Ez a kód tartalmazza a modell működését, viselkedését és a szükséges adatstruktúrákat.
2. Fordítás: Bináris kódra fordítja le először a Rapid Accelerator ezt a kódot, ami hatékonyabban futtatható.
3. Futtatás: A Simulink ezt a bináris kódot használja a futtatáshoz, ezáltal sokkal gyorsabb és hatékonyabb lesz a szimuláció.

Az előnyei közé tartozik a hatékonyabb erőforrás felhasználás, a gyorsabb szimuláció, valamint az alkalmazások valós idejű szimulációja. Fontos megjegyezni, az előnyök mellett, azt, hogy meg tudjuk tartani a modell rugalmasságát és az adott probléma részleteit, nem könnyű a módosítás vagy a hiba keresés. Ezért a Rapid Accelerator-t olyan helyzetekben érdemes használni, ahol a gyorsított futási sebesség és hatékonyság fontosabb, mint a modell módosíthatósága és hibakeresése.

3.2 Matlab Fuzzy Logic Toolbox

Szakdolgozatom fő témája beltéri hőmérséklet szabályozásának megvalósítása fuzzy szabályozóval, ezért a következőkben szeretném bemutatni a fuzzy szabályozó megvalósításához használt eszköz, a Fuzzy Logic Toolbox működését. A Fuzzy Logic Toolbox [8] a Matlab programnak az egyik eszköztára. A 3.9. ábra szemlélteti a Matlab által kínált programozási lehetőségeket, amelyen látható, hogy a Fuzzy Logic Toolbox önálló használata mellett a Matlab számos toolbox-ával, így a Simulink-kel is összekapcsolható.

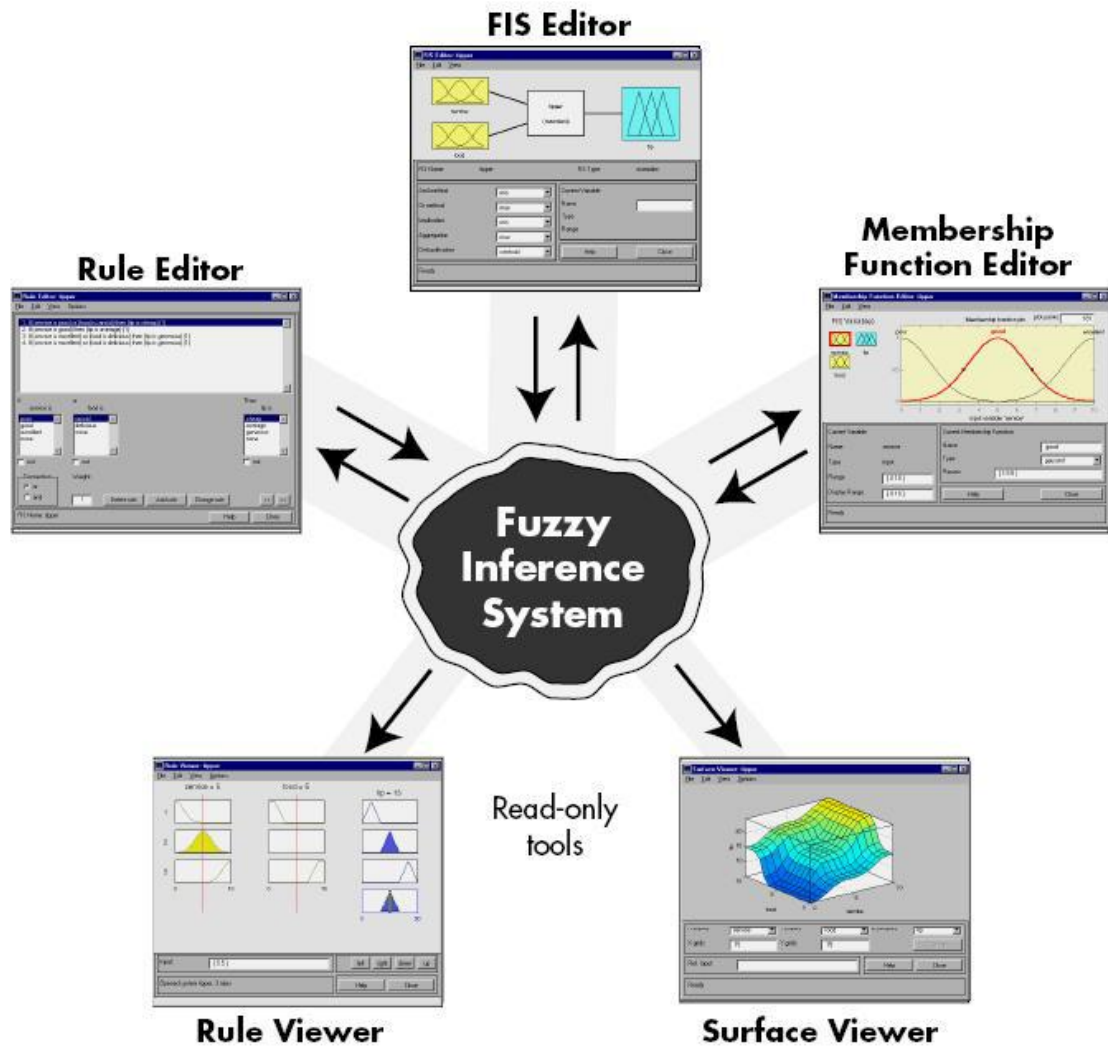


3.9. ábra Matlab - Fuzzy Logic Toolbox interakciós képességei [6]

A fuzzy logika lényege, ahogy azt a 2.1.1 fejezetben ismertettem, hogy egy bemeneti teret egy kimeneti térre képezzen le. Ennek mechanizmusa a szabályok által alkotott „ha-

akkor” utasítások definiálása és azok használata, a fuzzy következtető rendszer (angolul Fuzzy Inference System, FIS) működése. [6], [9]

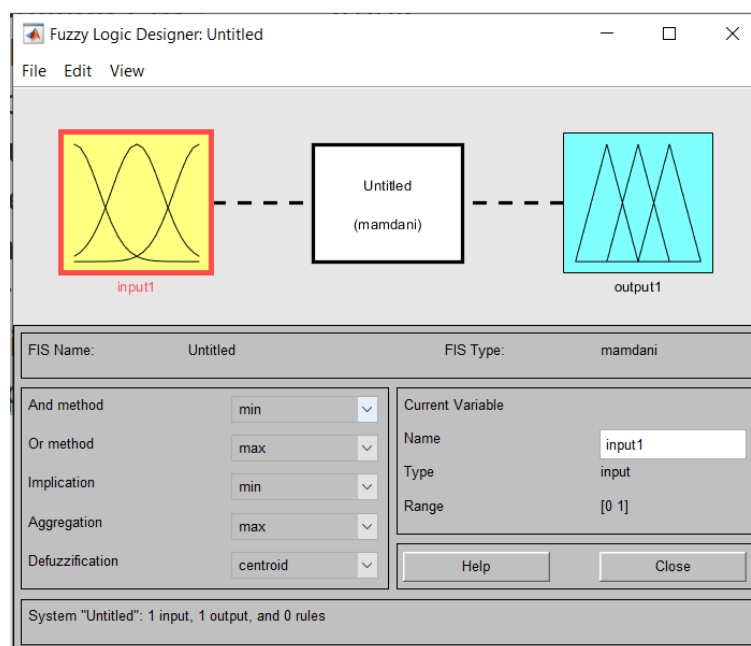
Ezeknek a feladatoknak az elvégzéséhez a Matlab Fuzzy Logic Toolbox grafikus felhasználói felülettel rendelkezik, amelynek elemeit a 3.10. ábra mutatja.



3.10. ábra A Fuzzy Logic Toolbox grafikus felülete [6]

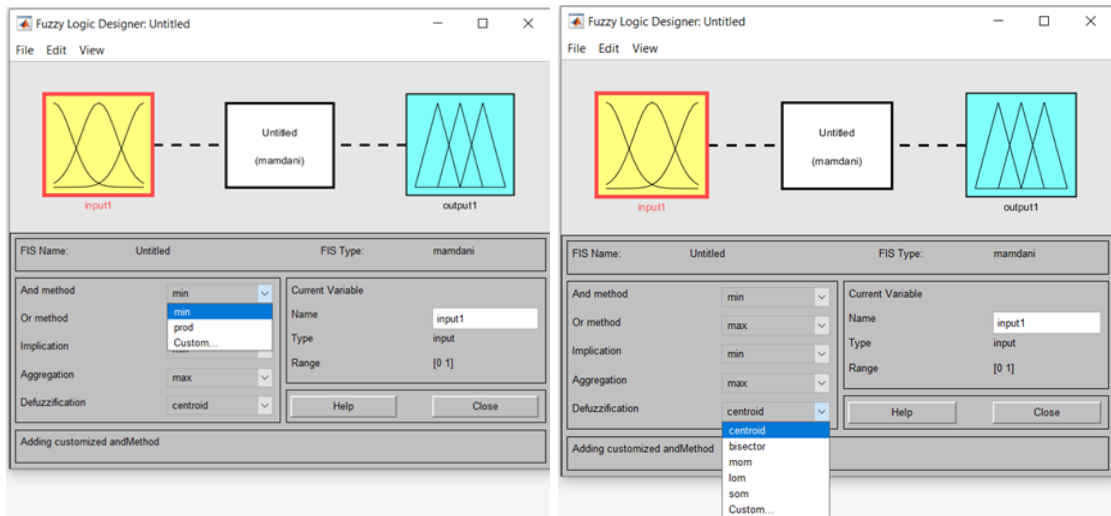
3.2.1 Fuzzy következtető rendszer szerkesztő

A Fuzzy Toolbox használatát a Matlab megnyitása után, a parancs ablakban a „fuzzy” szó beírásával tudjuk megkezdeni. Ekkor megnyílik egy általános szerkesztő, az úgynevezett FIS Editor (3.11. ábra). Itt láthatók a bemenetek és a kimenetek. Ha ezek valamelyikére kattintunk, szerkeszthetővé válik a hozzárendelt változó neve (Name) és láthatóvá válik a típusa (Type), ami input vagy output lehet, valamint a tartománya (Range), azaz az univerzum. Mindezek a FIS Editor jobb oldalán láthatók. Célszerű olyan változónevet adni, amiről könnyen felismerjük a tartalmát. Amennyiben további bemeneteket vagy kimeneteket szeretnénk megadni, akkor a menüben az Edit/Add Variable.../Input-ot vagy Output-ot kell kiválasztani. Annyi be- és kimenetet tudunk definiálni, amennyit a feladatunk megkövetel.



3.11. ábra Fuzzy Logix Toolbox általános szerkesztő felülete

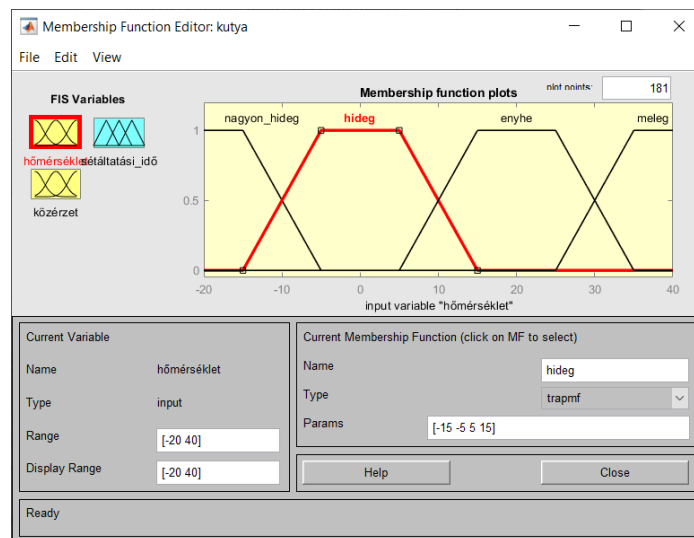
A FIS Editor bal oldalán találhatók a fuzzy műveletek elvégzésének módszerei, amelyek a Mamdani módszer által definiált alapbeállítások [4], amint ezt a 3.11. ábra mutatja. Természetesen lehetőség van ezek bármelyikének megváltoztatására akár további beépített függvények kiválasztásával, akár saját függvények definiálásával (3.12. ábra).



3.12. ábra Fuzzy műveletek módosítása

3.2.2 Tagsági függvény szerkesztő

Az input és output változókhoz tagsági függvények készíthetők a tagsági függvény szerkesztő használatával. [10] A tagsági függvények definiálásához kiválasztjuk azt a változót, amihez fuzzy halmazokat szeretnénk készíteni. Ezt dupla kattintással tudjuk megtenni, amellyel a tagsági függvény szerkesztő felület nyílik meg (3.13. ábra). Az ábrán egy input változó (hőmérséklet) tagsági függvényének szerkesztése látható.

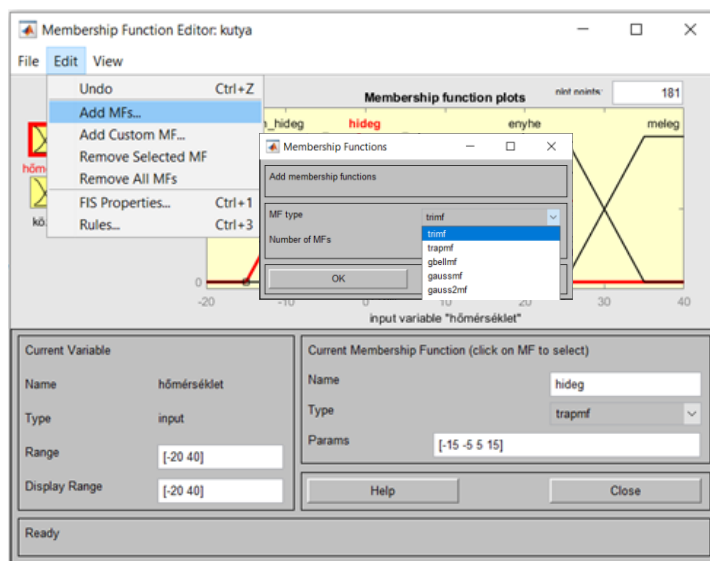


3.13. ábra Tagsági függvény szerkesztő – input változók

A megjelenő ablak három részre osztható. A felső részen a tagsági függvények „grafikus rajza” látszik, az ábrán négy tagsági függvény (nagyon_hideg, hideg, enyhe, meleg) található. Ez alatt, az editor bal oldalán található az aktuális változó (Current Variable). A változó neve (Name), típusa (Type), értelmezési tartománya, azaz az univerzum (Range) és a megjelenített tartomány (Display Range). Utóbbi kettő szerkeszthető, a Display Range-ben beállított tartományi értékek jelennek meg a tagsági függvények grafikus rajzán. A jobb oldali részen található a kijelölt tagsági függvény (Current Membership Function), amely a példában a „hideg”. A kijelölést mindig piros szín jelöli a grafikus részen, ott tudjuk kiválasztani melyik elemet szeretnénk változtatni. Itt megadhatjuk a tagsági függvény nevét (Name), típusát (Type), és a hozzá tartozó paramétereket (Params). A típusoknál tudjuk kiválasztani, hogy a tagsági függvény milyen alakú legyen: háromszög, trapéz, Gauss görbe...stb. A paraméternél állítjuk be a függvény alakját meghatározó tulajdonságokat, amelyet kétféleképpen tudunk megadni: vagy beírjuk ezeket az értékeket a Params cellába, vagy a grafikus rajzon, a görbe kitüntetett pontjait az egér segítségével arra az értékre mozgatjuk, ahova szeretnénk, hogy tartozzon.

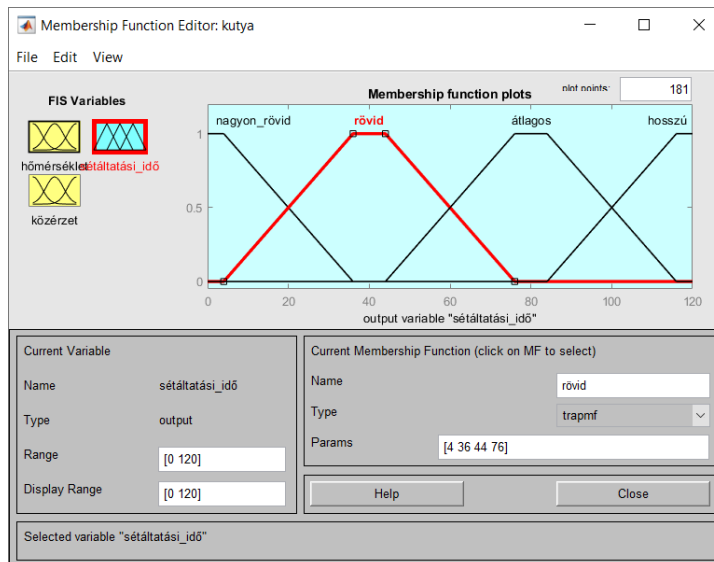
A tagsági függvények készítésénél fontos, hogy két egymás mellett elhelyezkedő tagsági függvény között legyen átfedés.

Ha növelni szeretnénk egy változó tagsági függvényeinek számát, akkor az Edit/Add MFs menüpontok kiválasztásával tehetjük meg. Több tagsági függvényt is tudunk egyszerre beszúrni és itt tudjuk megadni annak típusát is (3.14. ábra). Természetesen van lehetőségünk a tagsági függvények tulajdonságainak változtatásra is, a korábban leírtak szerint.



3.14. ábra Tagsági függvények hozzáadása

A bemeneti és a kimeneti változók tagsági függvényeinek definiálása azonos módon történik a tagsági függvény szerkesztő használatával. A 3.15. ábra egy output változó tagsági függvényének szerkesztését mutatja be.

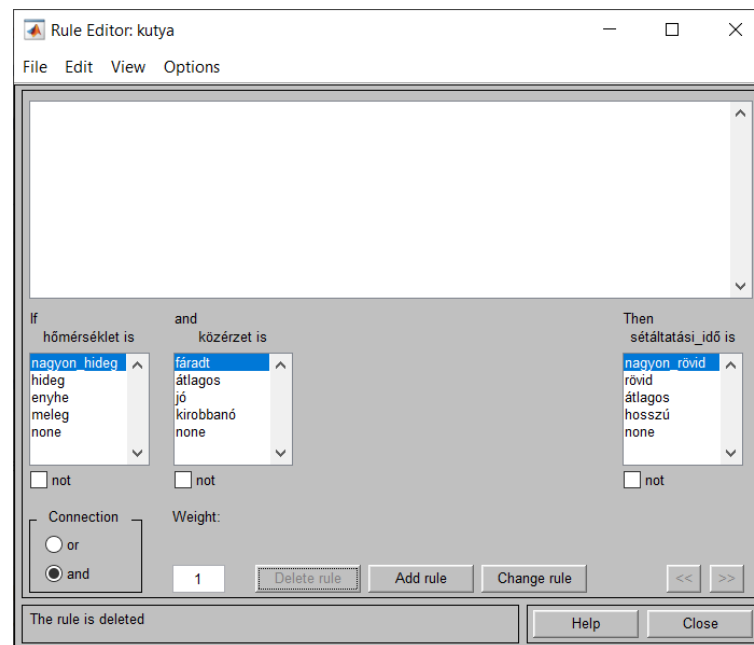


3.15. ábra Tagsági függvény szerkesztő - kimeneti változó

3.2.3 Szabály szerkesztő

Az input és output változók és a hozzájuk tartozó fuzzy halmazok definiálása utáni lépés a szabályok létrehozása. Ezt a szabály szerkesztő (Rule Editor) segítségével végezhetjük

el, amelynek aktiválása az Edit/Rules... menüpont kiválasztásával történik. A szabály szerkesztő ablakot a 3.16. ábra mutatja be.



3.16. ábra Szabály szerkesztő

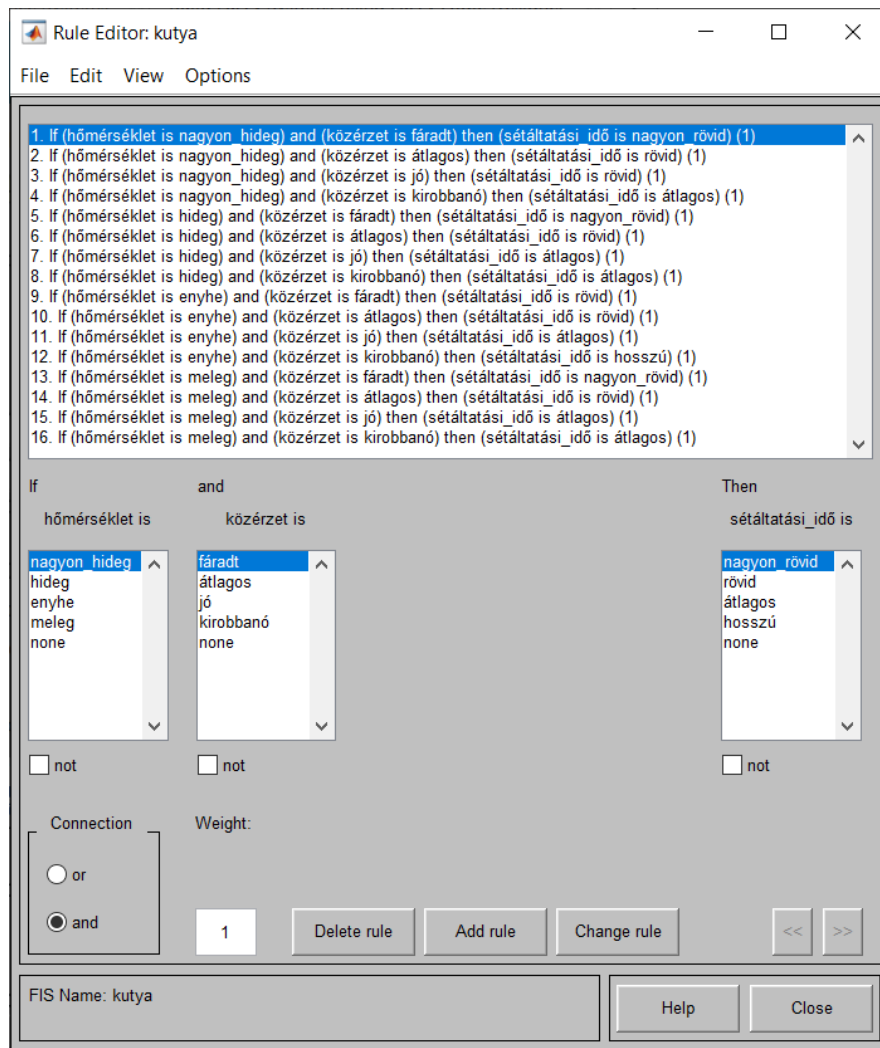
Az ablak felső részében az általunk írt szabályok jelennek meg, amely kezdetben természetesen üres. Alatta található a bemeneti és kimeneti változók és azok tagsági függvényei. A szabály feltételi részét jelölő „If” az ábra bal oldalán a bemeneteknél, a következmény részt definiáló „Then” pedig a bal oldalon a kimenetnél látszik. Manuálisan mi adjuk meg a szabályokat annak függvényében, hogy milyen feltételek mellett milyen következményt szeretnénk előállítani.

A bemeneti és a kimeneti változók alatt látszik egy „not” felirat. Ez azt jelenti, hogy ennek kiválasztása esetén a felette levő fuzzy tagsági függvény „ellentettét”, ami a fuzzy komplementer, szeretnénk figyelembe venni a szabályban.

A „Connection” lehetőséget ad a szabály feltételi részében szereplő tagsági függvények között értelmezett művelet beállítására. Az „or” és az „and” közül tudunk választani.

A „Weight” beállításnál tudjuk megadni a szabály súlyozásának mértékét. Mi választjuk meg, hogy az adott szabályunk mekkora súllyal számítson be az összesített következtetés eredményébe.

A szabály szerkesztő segítségével lehetőségünk van szabályokat létrehozni, törölni vagy egy korábban létrehozott szabályt megváltoztatni. Egy elkészített szabályhalmazt mutat be a 3.17. ábra.

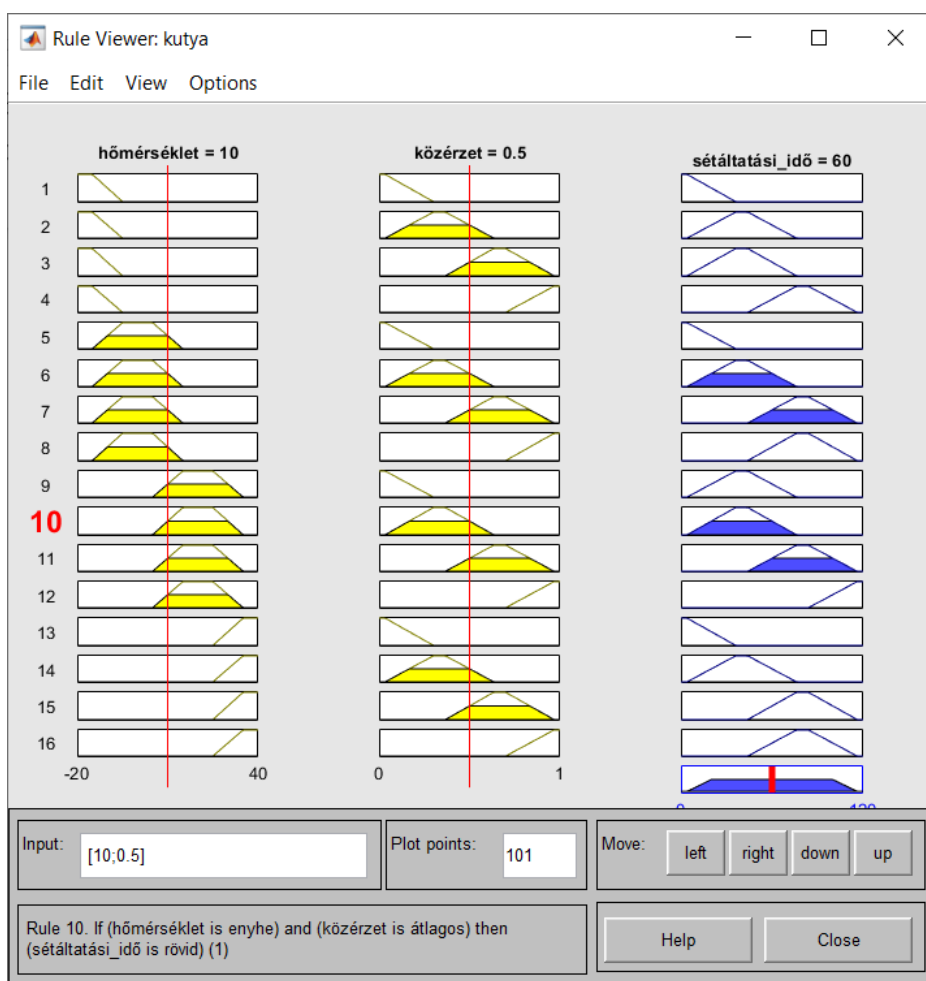


3.17. ábra Szabály szerkesztővel létrehozott szabályhalmaz

3.2.4 Szabály megjelenítő

A szabályok megadása után következik a Mamdani típusú fuzzy következtetés, amely során a bemeneti változókhoz tartozó jelértékek alapján a kimeneti változók értékeit meghatározzuk. A következtetés során minden definiált szabály kiértékelésre kerül, ahol a szabályok sorrendje nem releváns.

A View/Rules menüpont használatával megnyíló szabály megjelenítő (Rule Viewer) segítségével tudjuk megtekinteni a bemeneti és kimeneti tagsági függvényekből felépített szabályokat, azok tüzelését éles bemeneti változó értékek esetében, valamint az egyes szabályokkal és a teljes szabályhalmazzal végzett következtetés eredményét. Ezt a teljes fuzzy következtetési folyamatot szemlélteti a 3.18. ábra.



3.18. ábra Szabály megjelenítő

A szabály megjelenítő a következő részekből épül fel:

Az első, legnagyobb területű rész a grafikai rész. Itt jelennek meg a szabályok, az azokat alkotó tagsági függvények formájában. Az oszlopokban az input és output változók neve látszik, a sorokban pedig a szabályhalmaz szabályai jelennek meg. Egy szabály számára kattintva meg tudjuk nézni a szabályt a megjelenítő alján látható állapotsorban. A 3.18. ábra kiválasztott szabálya a 10. számú, amely az „If (hőmérséklet is enyhe) and (közérzet is átlagos) then (sétáltatási_idő is rövid)” szabály 1 súllyal. Ezeknek a lekérdezéseknek a segítségével lehetőség van az információk rugalmasabb és az emberi gondolkodáshoz hasonlóbb kezelésére, főleg akkor, ha az adatok nem pontosak vagy nem egyértelműek.

A példában az első két oszlopban az input változókhoz tartozó a tagsági függvények láthatók (szabályok feltételi része, „if” rész), ezek sárga színnel jelzettek. A harmadik, egyben az utolsó oszlopban az output változó tagsági függvénye (szabályok következmény része, „then” rész) található kék színnel jelölve. A változók neve és az aktuális „éles” értékek az oszlop tetején jelennek meg. Az utolsó oszlop legalsó „téglalapja” mutatja a szabályhalmaz következményeinek egyesítését. Az így kapott fuzzy tagsági függvény defuzzifikált kimeneti értékét az aggregált fuzzy halmazon áthaladó vastag piros vonal mutatja.

A második részen, a bal alsó sarokban az input változók éles értékei találhatók. Itt bemeneti értékeket tudunk megadni. A bemenetek megadására lehetőségünk van még úgy is, hogy magára a diagramra kattintva, vagy a függőleges piros vonalak húzásával állítjuk be az aktuális értékeket. Az input értékek megadása után végbemegy a fuzzy következtetés, a szabályok tüzelését az aktuális tagsági függvény „görbéje” alatti sárga rész mutatja, a következtetéssel kapott fuzzy tagsági függvényeket pedig a kék területek.

A harmadik részen a szabályhalmaz mozgatót tudom állítani a munkaterületen. Erre főleg akkor van szükség, ha a változók száma nagy, illetve a szabályhalmaz nagy méretű.

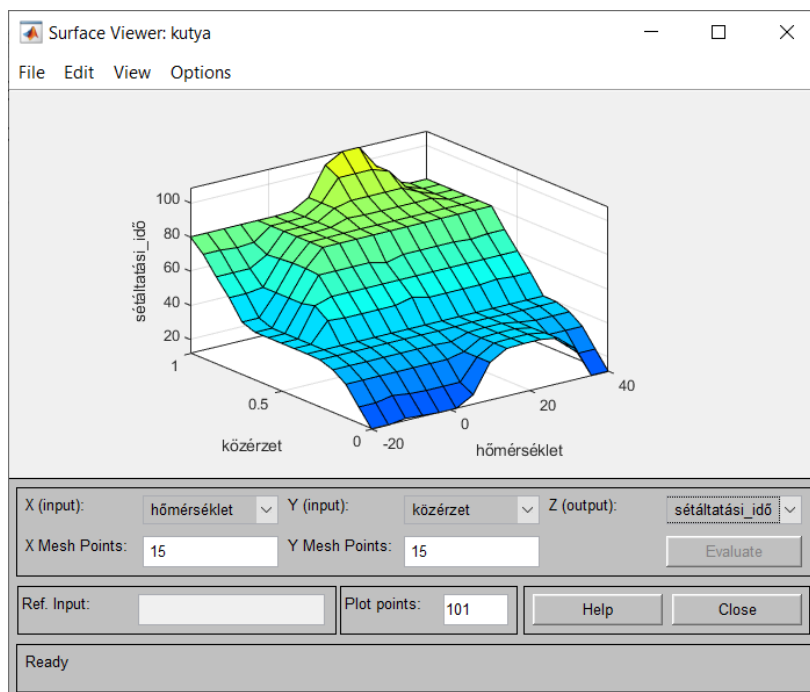
Összefoglalva a Rule Viewer lehetővé teszi a teljes fuzzy következtetési folyamat egyszerre történő értelmezését. Megmutatja, hogy az szabályok, az azokban szereplő tagsági függvények hogyan befolyásolják az általános eredményt. Minden szabály minden részét

ábrázolja. Ezt a vizualizációt kisebb méretű szabályhalmazok esetén célszerű használni, ha túl nagy rendszerben használjuk nehézkessé válhat a kezelése és átláthatósága. [6]

3.2.5 Felület megjelenítő

A View/Surface menüpont segítségével egy úgynevezett felület megjelenítő (Surface Viewer) indítható el, amely a fuzzy modell két input és egy output változója közötti összefüggést jeleníti meg egy háromdimenziós felülettel, ahol mi tudjuk kiválasztani a tengelyeken megjelenítendő változókat. Az „X” és „Y” tengely egy-egy input változót tartalmazhat, a „Z” tengely pedig egy output változót (3.19. ábra). Ha több bemenet vagy kimenet van, akkor mi tudjuk kiválasztani, melyiket szeretnénk megjeleníteni.

A grafikus megjelenítés alatt találhatóak a beállítási lehetőségek. A Plot pointsnál fontos, hogy ahhoz, hogy meg tudjuk jeleníteni a modellünket minimum 101 pontra van szükség. Az X és Y Mesh Points beállításnál a rácsháló sűrűségét tudjuk változtatni. A felület megjelenítő nagy előnye, hogy az egér segítségével a tengelyek körül el tudjuk forgatni a felületet és így a háromdimenziós felület különböző szögekből is megtekinthető.



3.19. ábra Felület megjelenítő

3.2.6 A modell mentése

A fuzzy rendszerek mentése az exportálással történik (File/Export menüpont). Ezt kétféleképpen tehetjük meg. To Workspace... és To File... attól függően, hogy mire szeretnénk használni a továbbiakban az állományt. Fájlba történő mentéskor egy .fis kiterjesztésű fájlt kapunk, ami egy szöveges fájl. A To Workspace... Matlab munkaterületre történő mentést jelent, ami létrehoz egy változót, amely a fuzzy rendszer MATLAB-objektumaként működik.

A beolvasáshoz a File/Import menüpont segítségével van lehetőség. Mind a két fájlformátumot vissza lehet olvasni.

4. A HŐMÉRSEKLETSZABÁLYOZÁS MEGVALÓSÍTÁSA

4.1 A vizsgált rendszer bemutatása

A diplomamunkám hőmérsékletszabályozási feladatának vizsgálatához az egyetem egyik tantermét választottam. A kiválasztott tanterem egy viszonylag nagy méretű előadóterem, a Pirosalma utcai épület 1. emeletének 103. előadóterme, amelyben körülbelül 60 hallgató fér el. A terem főbb paraméterei a következők:

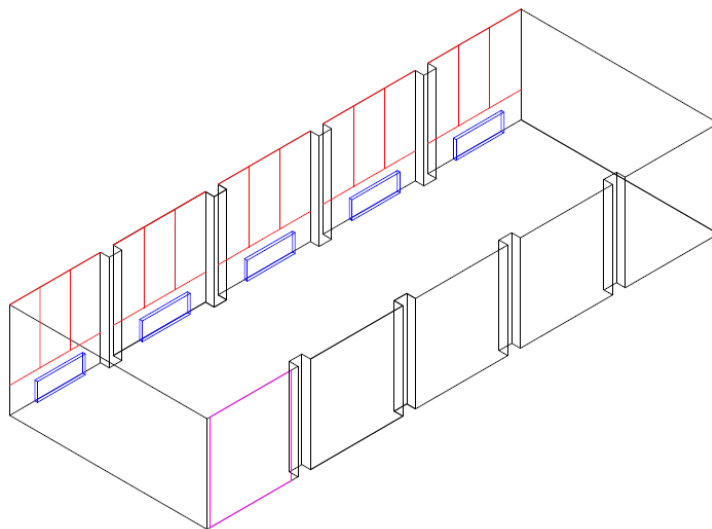
- A terem alapterülete: 112.56 m^2
- Belmagassága: 3.20 m
- Belső térfogata: $112.56 \text{ m}^2 * 3.20 \text{ m} = 360.19 \text{ m}^3$
- Ablakok száma: 5 db
- Radiátorok típusa: lapradiátor
- Radiátorok száma: 5 db
- Radiátorok mérete: $1.60 \text{ m} * 0.60 \text{ m} * 0.09 \text{ m}$
- A terem tájolása: Dél, Délkelet

A terem fényképét a 4.1. ábra mutatja.



4.1. ábra A 103. előadóterem fényképe (saját fotó)

A fénykép mellett szeretném szemléltetni a termet egy AutoCad-ben szerkesztett metszeten is, amelyet a 4.2. ábra szemléltet. A rajzon a falak fekete színnel lettek megszerkesztve, kék színnel látszanak a radiátorok, az ablakokat piros színnel ábrázoltam, az ajtó színe magenta. Az ablakok között oszlopok találhatók. Az ablakok előtt függöny nem található, csak sötétítő függöny, amely a napsütötte órákban árnyékot ad.



4.2. ábra A 103. előadóterem metszete (saját ábra)

4.2 A vizsgált rendszer Simulink modellje

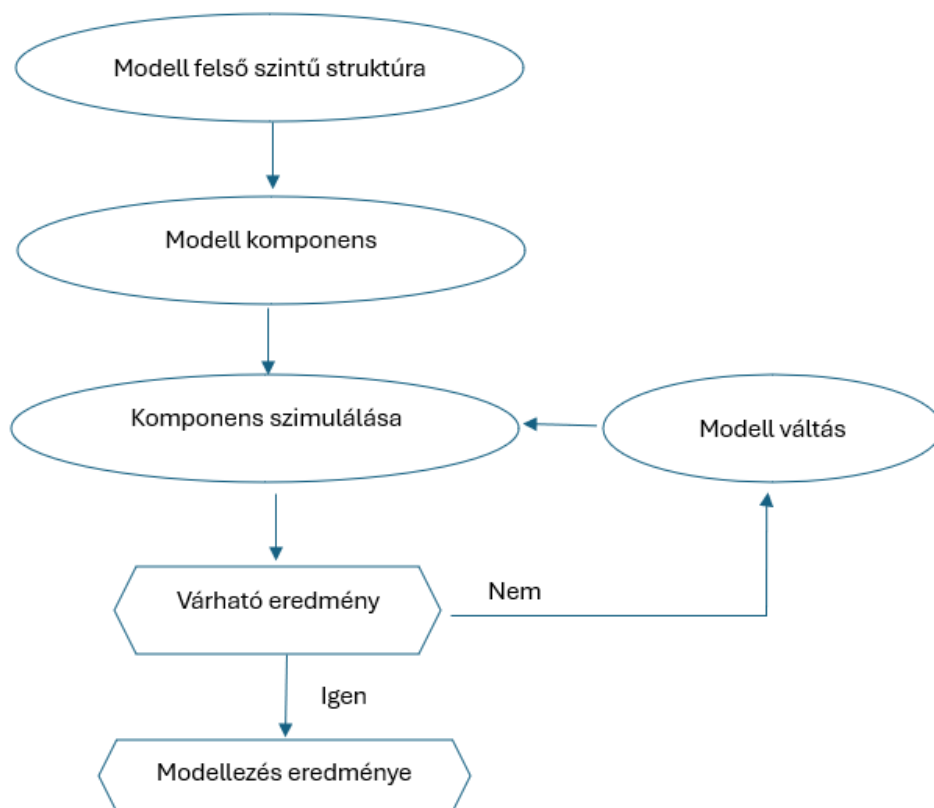
A programok elkészítése előtt megfogalmaztam a modellezési célokat. Fontos szempont, hogy a kitűzött célt milyen követelményrendszer mellett tudom elérni?

A fűtésrendszer kialakításánál a célom az, hogy megfigyeljem, hogy a külső hőmérséklet változása hogyan befolyásolja a terem hőmérsékletét. A terem elvárt hőmérséklet értékét (alapjel érték) én állítom be. A beállított hőmérséklet mellett szeretném megvizsgálni a változó paraméterek hatását a beltéri hőmérsékletre.

A tervezés következő lépése, hogy milyen szempontokat fogok figyelembe venni, ehhez milyen rendszerelemeket fogok használni. Ezeket milyen matematikai konstrukciókkal lehet leírni, a modellt hogyan lehet megoldani? Milyen paraméterekre lesz szükségem?

Meg kellett határoznom továbbá, hogy a hőmérséklet szabályozásához milyen szabályozót vagy szabályozókat használok. Három különböző típusú szabályozót vizsgáltam, ON/OFF szabályozót, PID szabályozót, illetve fuzzy szabályozót, amelyek közös jellemzője, hogy a rendszeresen ellenőrzött terem hőmérséklet és a beállított hőmérséklet közötti különbségtől függően határozza meg a beavatkozó jelet.

A következőkben egy folyamatábrán szemléltetem egy modell elkészítésének a folyamatát (4.3. ábra).



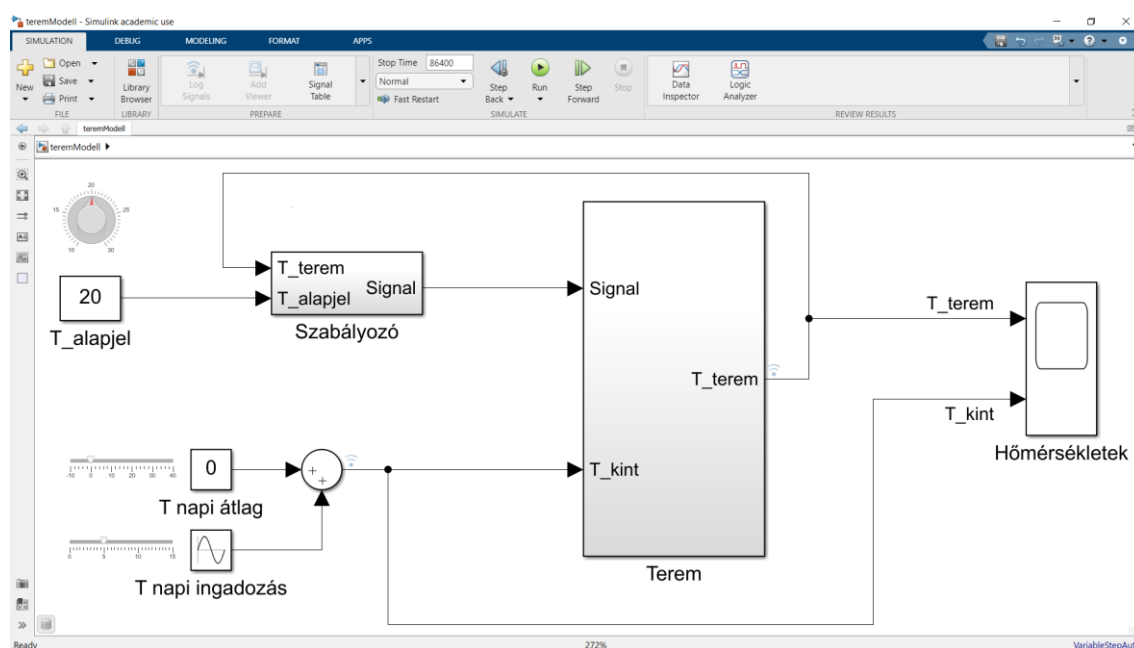
4.3. ábra Modell elkészítésének folyamata

A modell felső szintű struktúrájában az alrendszer blokkokat tervezem meg és használom, a modell szervezéséhez és a struktúra kialakításához. A modell felső szintű struktúrája tartalmazni fogja az alapjelet, a külső hőmérséklet meghatározásához kapcsolódó napi átlaghőmérsékletet és a napi ingadozást, a szabályozó egységet, valamint magát a terem részrendszert.

A modell komponense részben megvizsgálom a fűtésszabályozó rendszer komponensének modelljét, ahol bemenetként figyelembe veszem a terem aktuális hőmérsékletét és az alapjelet. Ez a részrendszer a háromféle szabályozó esetén különböző módon viselkedik, ezért különböző modellekkel írható le. A terem részrendszer bemenő jelei a külső hőmérséklet, valamint a szabályozó kimenő jele, amelyek ismeretében határozza meg a terem aktuális hőmérsékletét, amely egyúttal a részrendszer kimenő jele.

A részrendszerek bemeneti és kimeneti portokat tartalmaznak (inport és outport blokkok), amelyek összekapcsolják a részrendszert a modell más részrendszereivel.

A vizsgált rendszer Simulink modelljét, valamint annak részrendszereit a 4.4. ábra mutatja be.

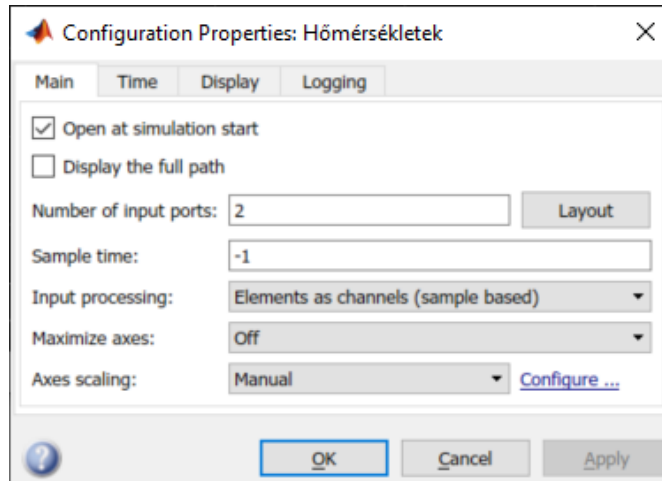


4.4. ábra A vizsgált rendszer Simulink modellje

Miután elkészültem a modell komponenseinek definiálásával, következik a modell szimulálása. Ebben a munkafázisban végig kell gondolni a modellünk várható viselkedését, és azt, hogy hogyan tudjuk ezt a viselkedést szimulációval tesztelni.

A szimuláció eredményeit szemléletesen „Scope” blokkok segítségével tudjuk nyomon követni, amely segítségével változók időbeli viselkedése jeleníthető meg ahogy ezt a 3.5.

ábra is mutatta. A szimuláció elindítása („Run” gomb megnyomása) után dupla kattintással tudjuk megjeleníteni a „Scope” blokk tartalmát, de lehetőség van a blokk konfigurációs paramétereinek beállításával a megjelenítés a szimulációval egyidőben történő, automatikus elindítására is (4.5. ábra).



4.5. ábra „Scope” blokk konfigurációs paramétereinek beállítása

A „Scope” blokk használatával ki fog rajzolódni az eredmény, miközben fut a szimuláció a háttérben. Ekkor lehetőségünk van eldönteni, hogy megfelel-e az eredmény az elvárásunknak. Ha nem felel meg, a modell módosítását szükséges elvégezni és megismételni utána a komponens szimulálását. Mindezt addig végezzük el, míg nem kapjuk meg a szükséges eredményt, azaz az elvárt viselkedést leíró modellt.

A következőkben bemutatom az elkészített részrendszerek modelljét.

4.3 A terem részrendszer

A terem részrendszer vizsgálata esetén a terem hőmérsékletének időbeli viselkedésére vagyunk kíváncsiak. Így a terem modelljének elkészítése során a benti hőmérsékletet, mint állapotváltozót vizsgáljuk, amelynek megváltozását egyrészt a fűtés során átadott hőenergia, másrészt a környezetnek átadott energiaveszteség befolyásolja. Ez a következő állapotegyenlettel fogalmazható meg:

$$\frac{dT_{\text{terem}} V \rho c_p}{dt} = Q_{\text{fűtés}} - Q_{\text{veszteség}}$$

ahol

T_{terem} : a terem hőmérséklete [$^{\circ}\text{C}$]

V : a terem belső térfogata [m^3]

ρ : a teremben levő levegő sűrűsége [kg/m^3]

c_p : a teremben levő levegő fajlagos hőkapacitása [$\text{J}/\text{kg}^{\circ}\text{C}$]

$Q_{fűtés}$: az átadott hőenergia mértéke [J/s]

$Q_{veszteség}$: az energiaveszteség mértéke [J/s]

Konstans térfogatot, sűrűséget és fajlagos hőkapacitást feltételezve az egyenlet átalakítható a következőképpen:

$$\frac{dT_{terem}}{dt} = \frac{Q_{fűtés}}{V\rho c_p} - \frac{Q_{veszteség}}{V\rho c_p}$$

Az átadott hőenergia mértéke függ a fűtőberendezés és a terem hőmérsékletének különbségétől

$$Q_{fűtés} = Q(T_{fűtő} - T_{terem})$$

az energiaveszteség pedig a terem és a külső hőmérséklet különbségétől

$$Q_{veszteség} = U_w A_w (T_{terem} - T_{kint})$$

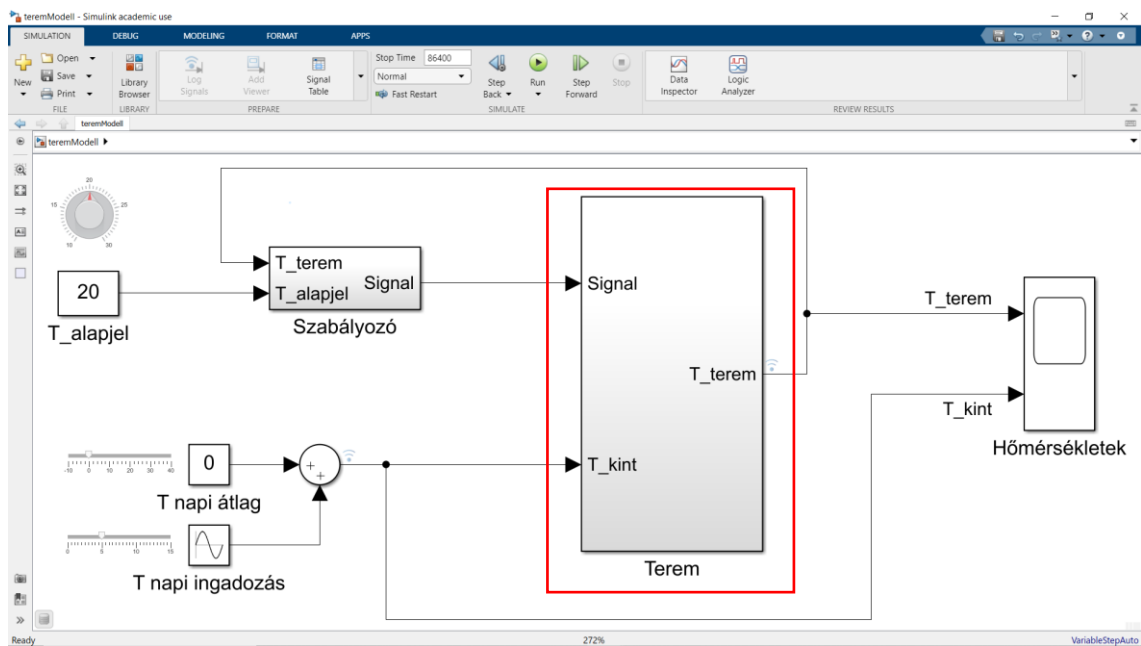
ahol

$T_{fűtő}$: a fűtőberendezés hőmérséklete [$^{\circ}\text{C}$]

U_w : a hőátadási tényező [$\text{W}/\text{m}^2^{\circ}\text{C}$]

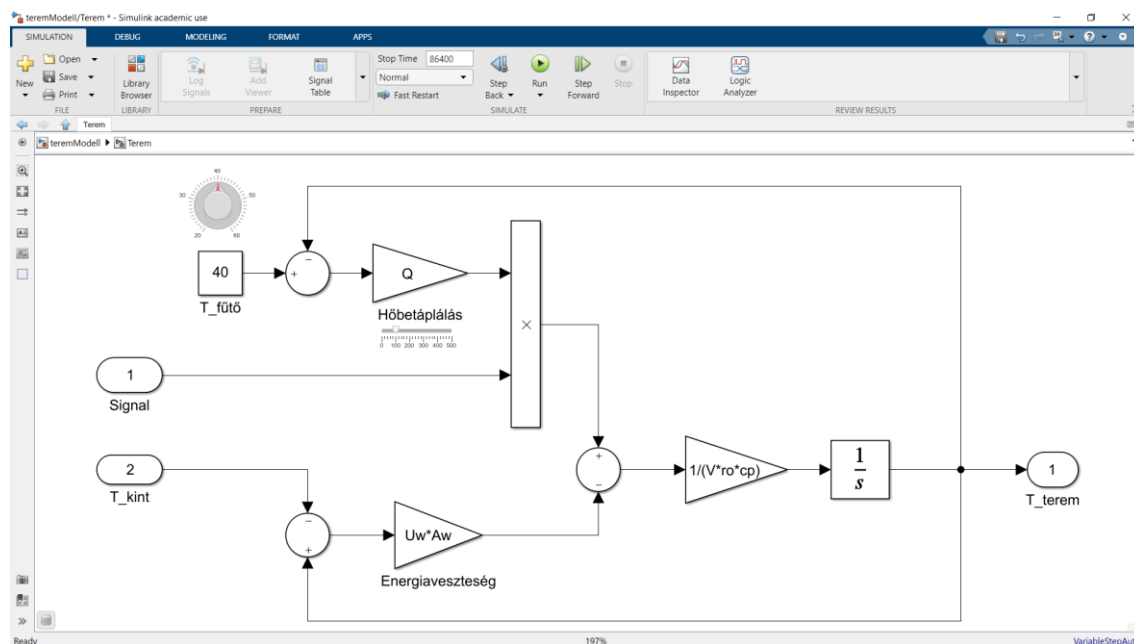
A_w : a hőátadási felület [m^2]

T_{kint} : a külső hőmérséklet [$^{\circ}\text{C}$]



4.6. ábra Simulink modell - Terem részrendszer

Az így kapott egyenleteket használtam a Terem részrendszer (4.6. ábra) Simulink modelljének elkészítéséhez. A részrendszerhez két bemeneti port (a szabályozó kimeneti értéke, „Signal” és a külső hőmérséklet, „T_kint”) és egy kimeneti port (a terem hőmérséklete, „T_terem”) lett kialakítva. A részrendszert és annak elemeit a 4.7. ábra mutatja.



4.7. ábra Terem részrendszer és annak blokkjai

Az ábra bal felső sarkában található „T_{fűtő}” blokk értéke a fűtőberendezés (radiátor) hőmérsékletét mutatja. A szimuláció során konstans értéket feltételeztem, de az érték megváltoztatásával lehetőség van különböző radiátor-hőfok beállítás hatásának vizsgálatára is. Az ábrán beállított érték 40 °C, amelynek megváltoztatása vagy a blokk felett található „Knob” blokk segítségével manuálisan, vagy közvetlenül a „T_{fűtő}” blokk konstans paraméterének módosításával történhet. A „Knob” blokkal történő beállításhoz 20 és 60 °C közötti értékeket definiáltam.

A következő nagyobb egységem a „Hőbetáplálás” nevű „Gain” blokk, amellyel a fűtőberendezésből a terembe átadott hőenergia értéke határozható meg, ami arányos a fűtőberendezés és a helyiség közötti hőmérsékletkülönbséggel. A „Hőbetáplálás” blokk alatt található egy skála, melyen manuálisan is lehet állítani Q értékét. A ténylegesen átadott hőenergia értéket befolyásolja még a szabályozó output jele („Signal”).

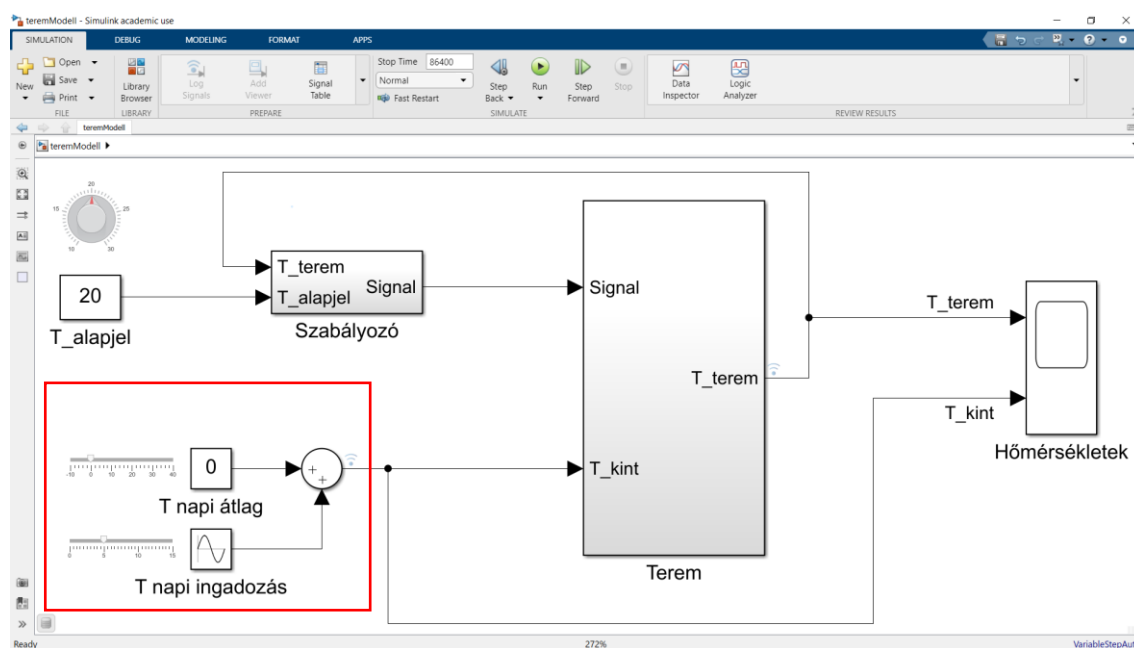
Az ábra bal alsó részén látható az energiaveszteség meghatározása („Energiaveszteség” nevű „Gain” blokk), amely a vizsgált terem homlokzati falán és ablakain keresztül eltávozó hőenergiát jelenti. Ez a veszteség arányos a terem hőmérséklete („T_{terem}”) és a külső hőmérséklet („T_{kint}”) közötti különbséggel. Befolyásolja még a hőátadási felület

(A_w), amely esetünkben 45.02 m^2 , valamint a hőátadási tényező (U_w). A terem homlokzati falán $1/3$ arányban található fal és $2/3$ arányban ablak, ennek figyelembevételével átlagos szigetelést feltételezve ez utóbbinak az értéke $0.94 \text{ W/m}^2 \text{ } ^\circ\text{C}$.

A fűtéssel átadott hőenergia és a hőenergia-veszteség különbsége adja a hőenergia-változás értékét. Ebből a hőmérséklet-változás egy újabb „Gain” blokkal számítható ki, amelynek erősítési tényezője $\frac{1}{V\rho c_p}$, ahol a terem belső térfogata 360.19 m^3 , ρ (ro) a levegő sűrűsége 1.205 kg/m^3 , c_p a levegő fajhője $1013 \text{ J/kg } ^\circ\text{C}$.

Azt, hogy vizsgálni tudjam, hogy egységnyi idő alatt mennyivel nő a hőmérséklet az $\frac{1}{s}$ jelű „Integrator” blokk végzi el, amely a bemeneti jel időbeli integráljának értékét adja ki, jelen esetben a kimenete a terem hőmérséklete („T_terem”).

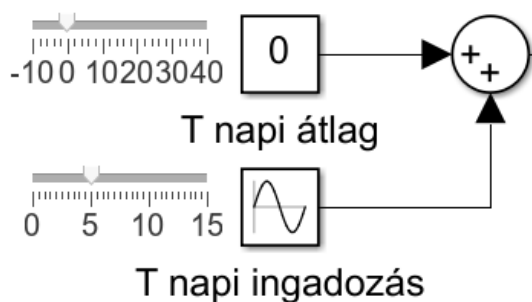
4.4 Külső hőmérséklet



4.8. ábra Simulink modell - Külső hőmérséklet szimulálása

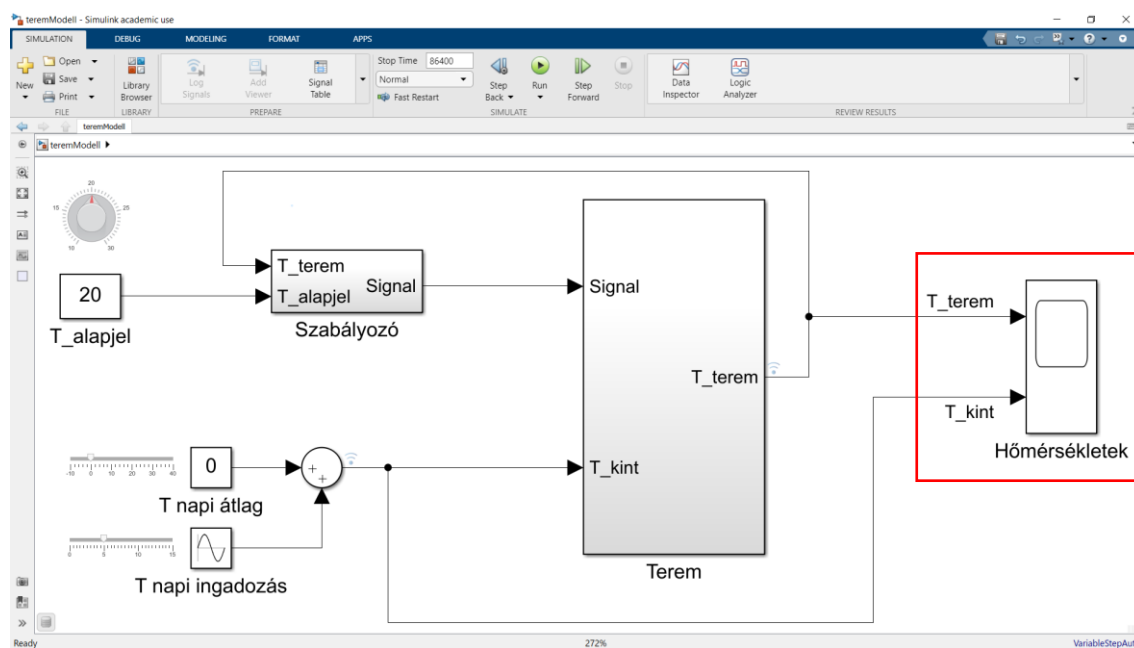
A Simulink modell bal alsó sarkában található a kinti hőmérséklet szimulációjához készített részrendszer (4.8. ábra). Ez két blokból épül fel. Az egyikkel beállítható a napi átlaghőmérséklet („T napi átlag”), amely az ábrán $0 \text{ } ^\circ\text{C}$ értékre van beállítva.

A második blokk a napi ingadozást írja le egy szinusz függvény segítségével. Az ingadozás értéke az ábra esetében ± 5 °C. Ezek konstans értékek, azonban ezeket beállíthatjuk egyrészt a mellettük elhelyezett „Slider” blokkok segítségével, másrészt a „Constant” blokkokra kattintva az értékek beállításával (4.9. ábra).



4.9. ábra Külső hőmérséklet

4.5 Hőmérsékletek megjelenítése



4.10. ábra Simulink modell – Hőmérsékletek megjelenítése

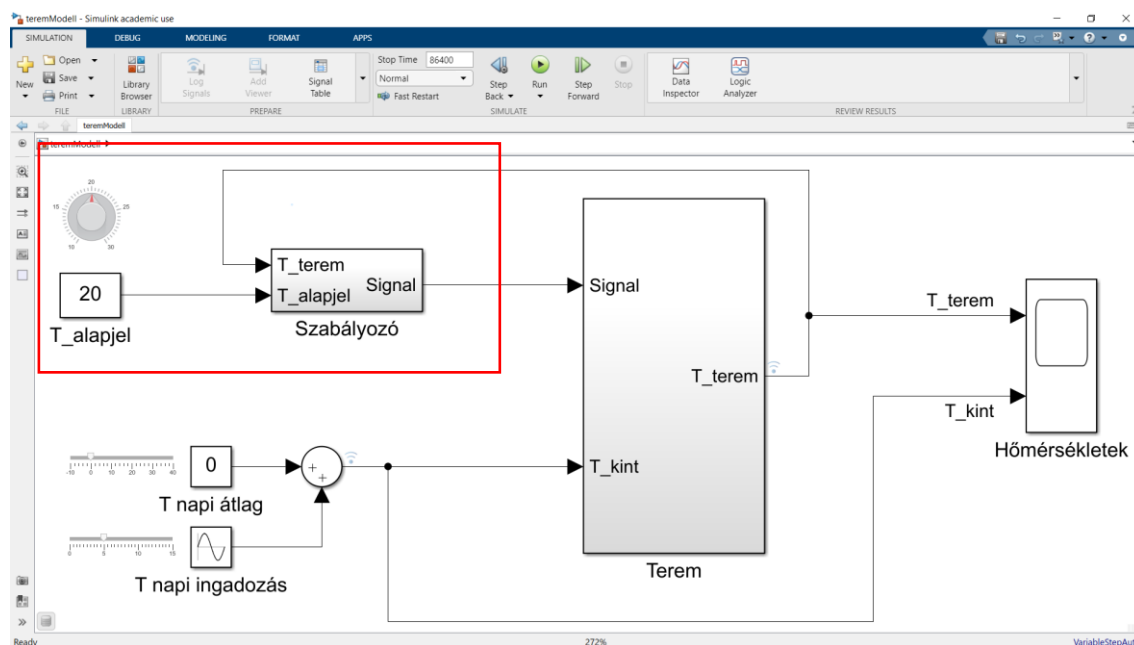
A szimuláció megjelenítéséhez a Simulink modellben elhelyeztem egy „Scope” blokkot „Hőmérsékletek” névvel (4.10. ábra). Ez egy olyan blokk, mely lehetővé teszi a jelek grafikus megjelenítését és monitorozását a tesztelés során. A „Scope” blokk lehetőséget

biztosít a megjelenítő jellemzők konfigurálására, be tudjuk állítani a megjelenített jelek számát, az időtartományt és a jelek színét is. Mivel több jelet tudunk egyszerre megjeleníteni, lehetőségünk van könnyen elvégezni az összehasonlítást is.

Az ábrán látható „Scope” blokk által megjelenített grafikon vízszintes tengelyén az időt mutatja, a függőleges tengelyén pedig a hőmérsékletet. A vizsgálat során lehetőség van állítani az időtartományon, ha nem szeretném a teljes szimulációs intervallumot egyszerre látni, hanem egy bizonyos részét szeretném nézni a vizsgált intervallumnak. A szimuláció során módunkban áll a jellemzők módosítására, így nyomon követhető a változás. A jellemzők mentése egy lényeges funkció, mert a későbbiekben praktikus, hogy nem kell az egész programot lefuttatni, ha szeretnénk visszanézni valamit a rendszerünk-ből vagy hibát keresni.

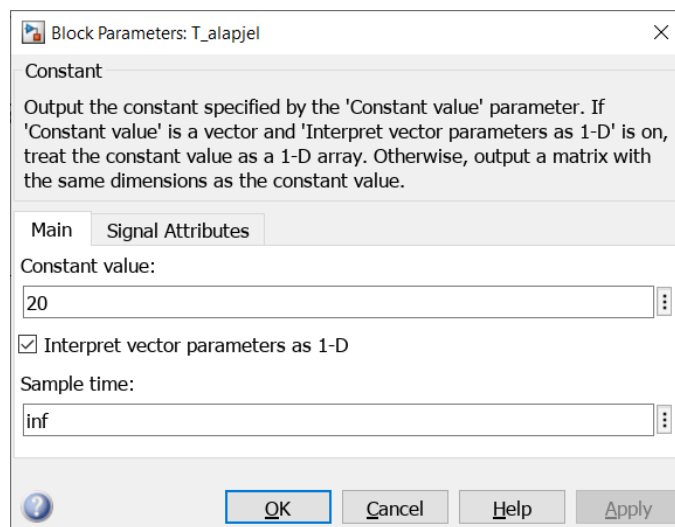
A modellemhez tartozó „Scope” blokknak az input jelei a terem hőmérséklete („T_terem”) és a kinti hőmérséklet („T_kint”).

4.6 A szabályozó részrendszer



4.11. ábra Simulink modell – Szabályozó részrendszer az alapjellel

A Simulink modell bal felső részén található a szabályozó részrendszer, valamint annak egyik bemenete, az alapjel (4.11. ábra). A „T_alapjel” nevű blokk az elvárt teremhőmérsékletet jelenti, mely az ábrán látható esetben 20 °C-ra állítottam be. Ezt kétféleképpen lehet változtatni. Az alapjel felett található „Knob” blokkal, melynek intervallum értéke 10-30 °C között definiált. Itt manuálisan tudjuk az egér segítségével beállítani az alapjelet. A másik lehetőség, hogy „T_alapjel” blokk paramétereit között mi tudjuk átírni ezt a konstans értéket (4.12. ábra).



4.12. ábra T_alapjel beállítási paramétereit

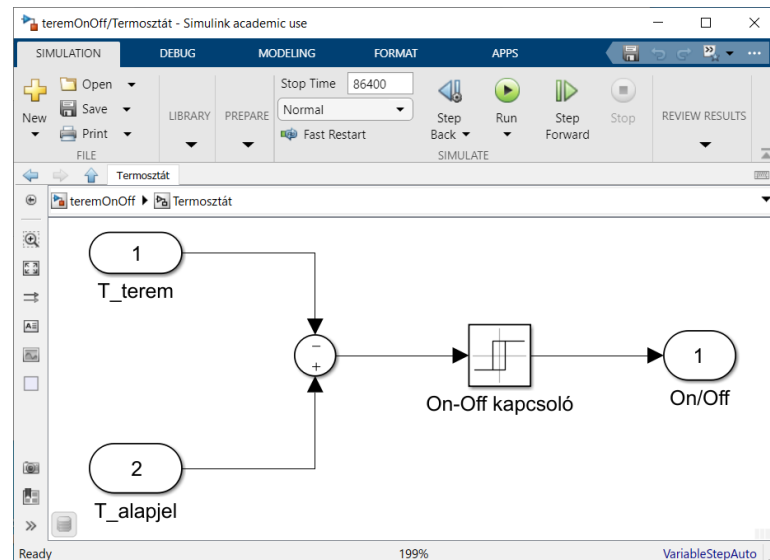
A 4.11. ábra „Szabályozó” részrendszere a „T_terem” és a „T_alapjel” bemeneti értékek ismeretében meghatározza a szabályozó kimenetét („Signal”). A következőkben három különböző típusú szabályozó részrendszert és az ezek használatával megvalósított terem hőmérséklet-szabályozó rendszer viselkedését mutatom be.

A vizsgált szabályozó részrendszerek a következők:

1. ON/OFF szabályozó
2. PID szabályzó
3. Fuzzy szabályozó

4.6.1 ON/OFF szabályzó

Először az ON/OFF szabályozó részrendszert (Termosztát) mutatja be a 4.13. ábra, amelynek fő eleme az „On-Off kapcsoló” nevű „Relay” blokk.

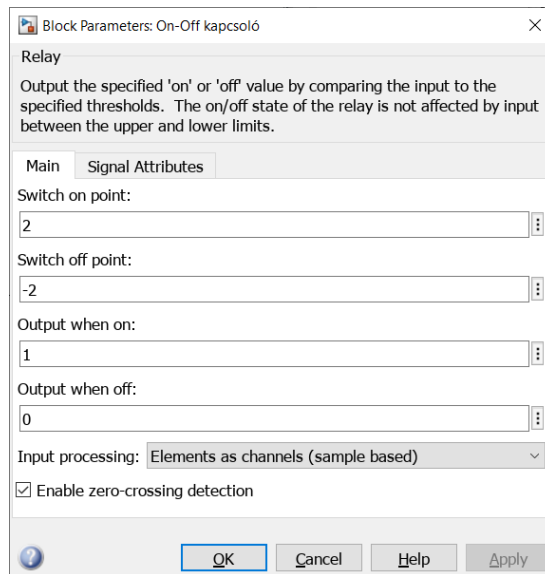


4.13. ábra Az ON/OFF szabályozó részrendszer

Az „On-Off kapcsoló” bemenetének meghatározásához két változót használok:

1. a „T_terem” – terem aktuális hőmérséklete
2. a „T_alapjel” – a beállított hőmérséklet

A kettő különbsége („T_alapjel” – „T_terem”) az „On-Off kapcsoló” inputja, amely meghatározza, hogy bekapcsoljon-e a termosztát, attól függően, hogy hidegebb vagy melegebb van az alapjelhez képest. Az „On-Off kapcsoló” paraméterbeállításait a 4.14. ábra mutatja, ahol látható a „Relay” blokk működésének leírása is: az output egy „on” vagy „off” érték, amely megadott küszöbértékekkel történő összehasonlítással határozható meg. Az „on/off” állapotot nem befolyásolja a határértékek közötti bemenet. A blokk működése a példában: ha a terem hőmérséklete a beállított hőmérséklet alatt van, a vezérlőjel egyenlő 1-gyel (a fűtés bekapcsol). Ha a terem hőmérséklete a beállított hőmérséklet felett van, a vezérlőjel 0 (a fűtés kikapcsol).

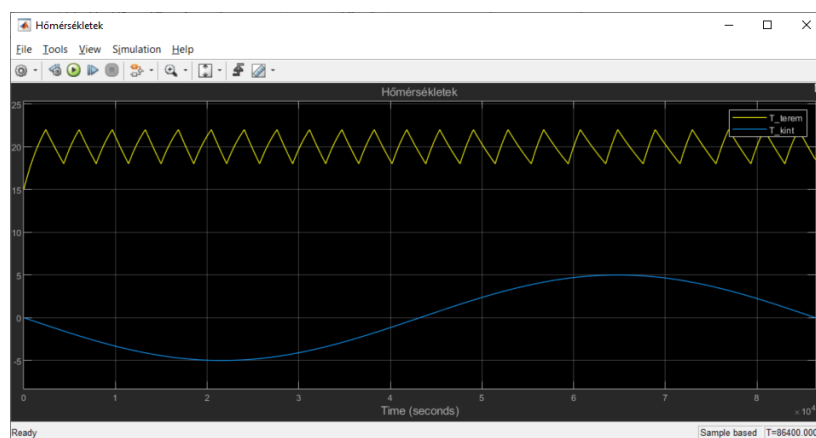


4.14. ábra Az „On-Off kapcsoló” paraméterei

A beállított hőmérséklet körüli ismételt kapcsolás elkerülése érdekében a termosztát 2 fokos szórást engedélyez. Ha a termosztát be van kapcsolva, a terem hőmérsékletének 2 fokkal a beállított hőmérséklet fölé kell emelkednie, mielőtt kikapcsol. Ha a termosztát ki van kapcsolva a terem hőmérsékletének 2 fokkal a beállított hőmérséklet alá kell csökkennie, hogy bekapcsoljon. Ez a paraméter érték természetesen módosítható.

A szabályozó részrendszer ismertetésével a Simulink modellem minden elemét bemutattam, a továbbiakban a modellel végzett különböző szimulációk eredményét ismertetem.

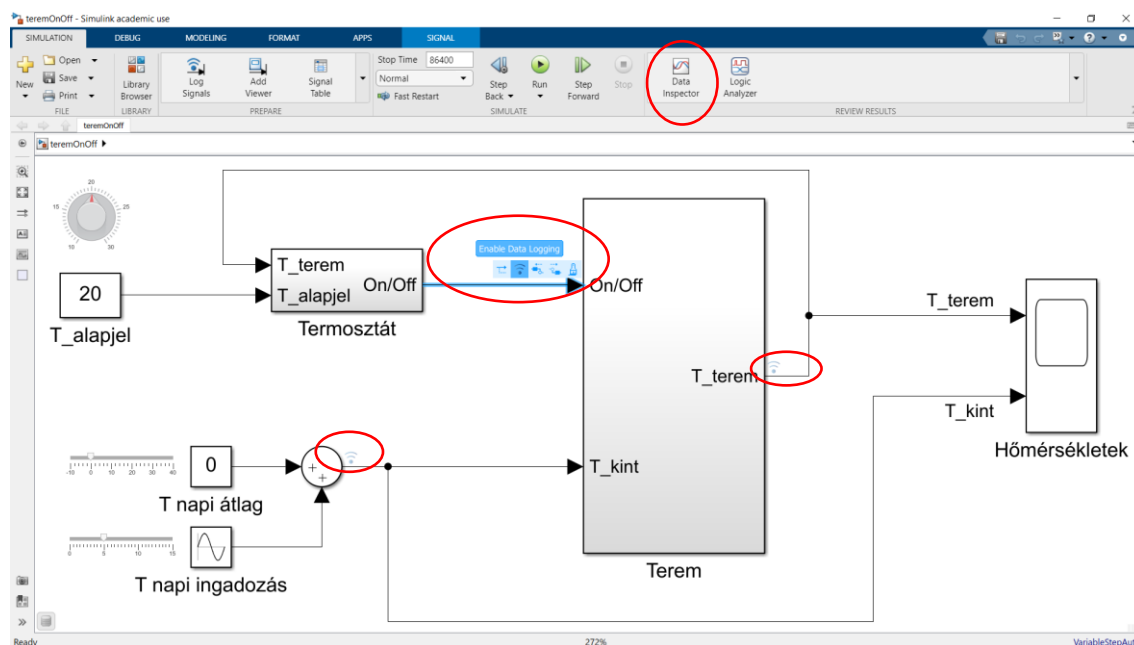
Az ON/OFF szabályozó működését mutatja be a 4.15. ábra.



4.15. ábra Az ON/OFF szabályozó működése

Az ábrán látható a modellemhez tartozó „Scope” blokk jeleinek, a terem hőmérsékletének és a kinti hőmérsékletnek az időbeli viselkedése. Az idő melyet vizsgálok egy nap, 24 óra (86400 s). A terem kezdeti hőmérséklete 15 °C, az alapjel értéke 20 °C, a kinti hőmérséklet átlagos értéke 0 °C.

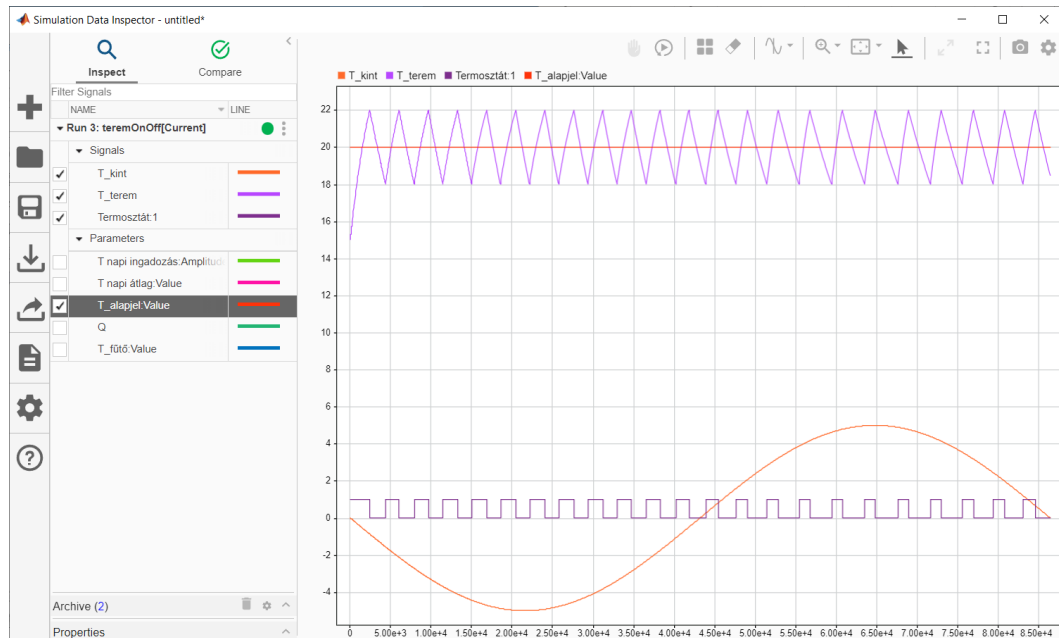
A szimuláció eredménye nemcsak a „Scope” blokkban jeleníthető meg, hanem a jelekre kattintva további beállítási lehetőségek nyílnak meg, ahogy azt a 4.16. ábra mutatja.



4.16. ábra Jelek megjelenítése

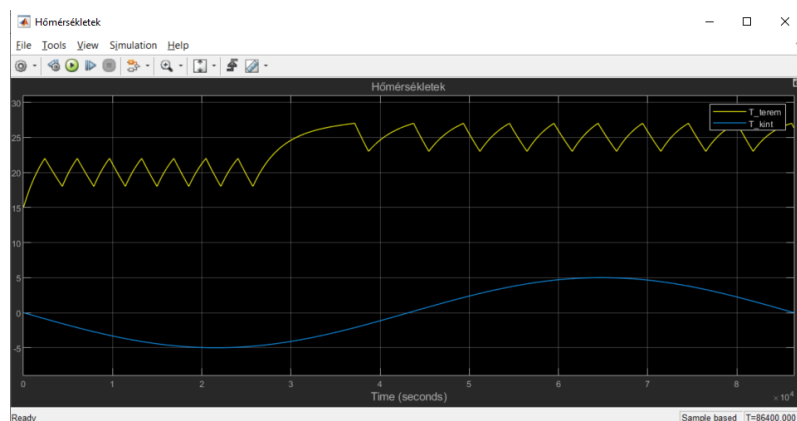
Itt kiválasztom a vizsgálandó jelekhez az „Enable Data Logging” funkciót, majd a szimuláció során a „Data Inspector” funkciót, amivel megjelenik egy újabb grafikus ablak, melyben vizsgálni tudom az általam kiválasztott jeleket (4.17. ábra). Több adatsort tudok egyszerre megjeleníteni, mely lehetővé teszi az összehasonlítást és a változások lekövetését. Az adatsorok beállítására az ablak bal oldali részén van lehetőségem. Mappákban találjuk meg az általunk definiált jeleket. A „Signals” részben található a „T_kint”, „T_terem” és a „Termosztát1”. A „Parameters” mappában a „T_napi ingadozás amplitude”, „T_napi átlag”, „T_alapjel” a „Q” és a „T_fűtő” blokk paraméterek. Ezek közül a kinti és benti hőmérsékletet, termosztát változását és az alapjel változását fogom vizsgálni. A 4.17. ábra mutatja, hogy a beállított terem hőmérsékletem (alapjel) 20 °C, a kinti

hőmérséklet -5 és $+5$ °C között ingadozik, a termosztát pedig a be- és kikapcsolást mutatja (1-nél bekapcsol, 0-nál kikapcsol). Megfigyelhető, hogy ha nagyobb a kinti hőmérséklet, rövidebb ideig van bekapcsolva a termosztát, mint alacsonyabb külső hőmérséklet esetén.



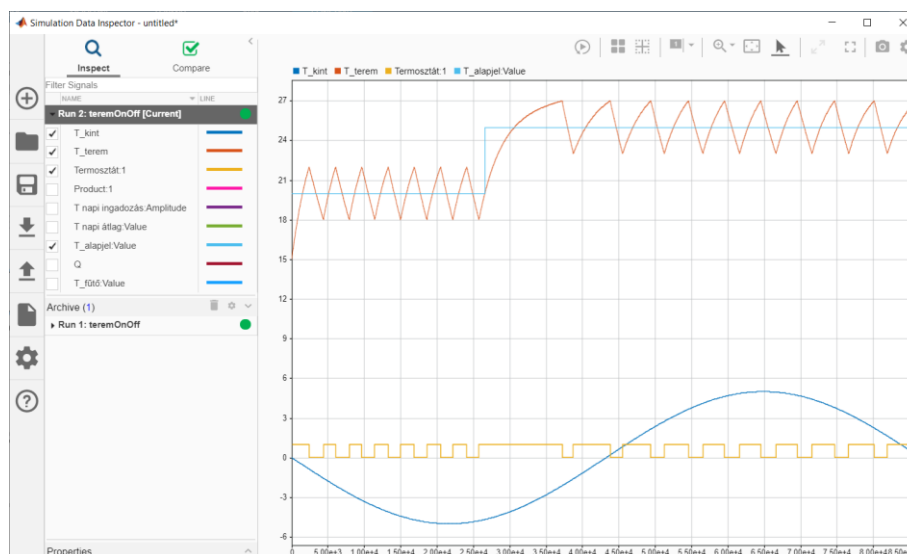
4.17. ábra ON/OFF szabályozó - megjelenítés Data Inspector-ban megjelenítve

A következőkben azt fogom vizsgálni, ha megváltoztatom az alapjelet, hogyan fog változni a termosztát viselkedése. Első lépésként elindítom a korábbi példában használt paraméterekkel a szimulációt, majd egy idő után átállítom az alapjel hőmérsékletét 25 °C-ra. A 4.18. ábra jól szemlélteti, hogy hol történik az alapjel beállításának megváltoztatása.



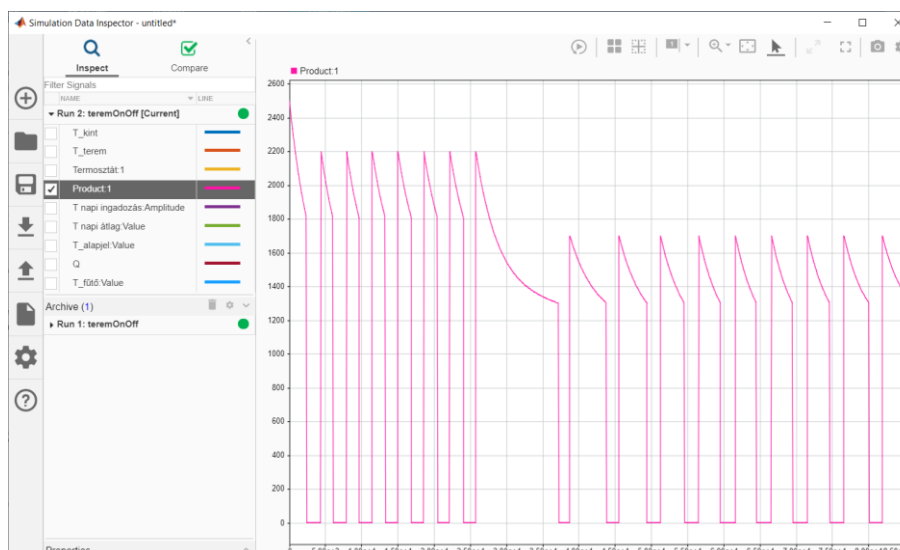
4.18. ábra ON/OFF szabályozó - alapjel változtatása

Megfigyelhetjük, hogy az alapjel megnövelésével a terem hőmérsékletének beállítása több időt vesz igénybe. A 4.19. ábra a kinti és a terem hőmérséklete mellett az alapjel értékét és a termosztát állapotát is mutatja. Ha megnézzük a termosztát ki/be kapcsolásának diagramját, látszik, hogy amikor megnöveltük az alapjelet, sokkal hosszabb ideig volt bekapcsolva, hogy elérje a beállított hőmérsékletet, mint az alapjelváltás előtt vagy után.



4.19. ábra ON/OFF szabályozó - alapjel változtatása Data Inspector-ban megjelenítve

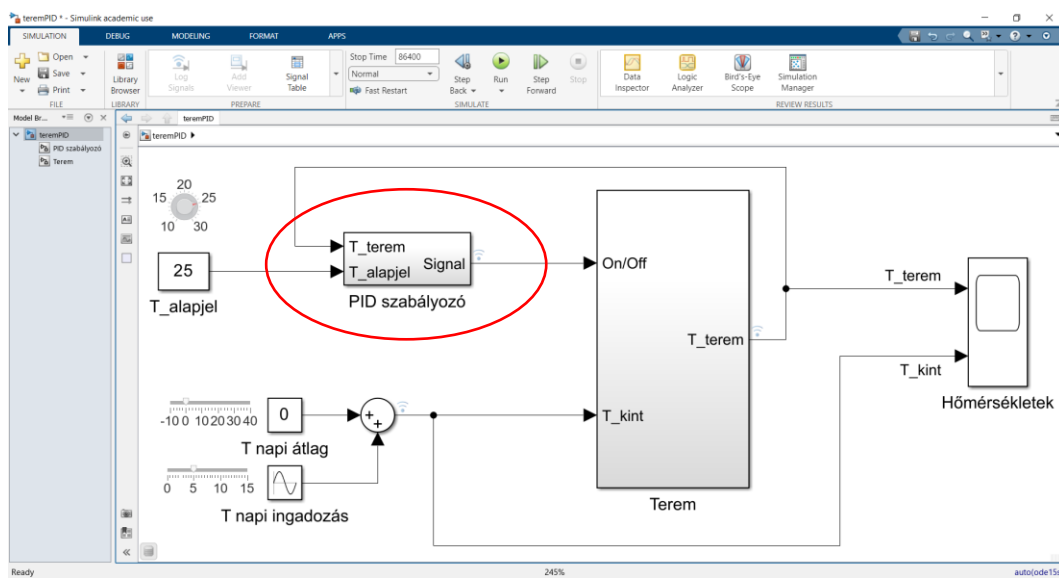
A 4.20. ábra mutatja a fűtőberendezés által átadott hőenergiát be- és kikapcsoláskor.



4.20. ábra ON/OFF szabályozó - energiaátadás

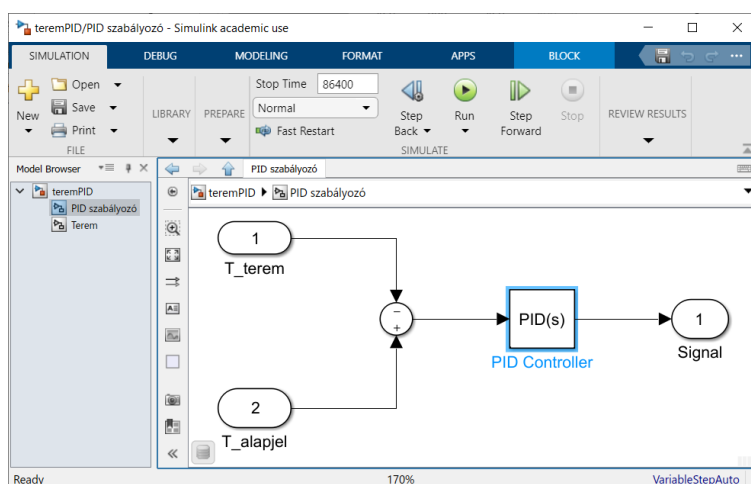
4.6.2 PID szabályozó

A következőkben részletesen bemutatom az általam készített fűtés szabályzó rendszert PID szabályozóval. Alapnak a korábban bemutatott rendszert használok, ami változik benne, az a szabályozó részrendszer lesz. Ezt a 4.21. ábra mutatja be.



4.21. ábra Simulink modell – Szabályozó részrendszer PID szabályozóval

A PID szabályozó részrendszer (4.22. ábra) tartalmaz két bemeneti elemet, a terem hőmérsékletének és az alapjelnek az értékét. Ennek különbsége adja a „PID controller” blokk inputját.



4.22. ábra A PID szabályozó részrendszer

A PID szabályozó paraméterei a következők (4.23. ábra):

- P – Proportional gain, a bemenő jel és a kimenő jel közötti erősítés mértéke
- I – Integral gain: az alapjeltől való eltérés múltbeli értékeinek időbeli integrálásából származó érték. A cél, hogy a görbe alatti terület minél kisebb legyen.
- D – Derivative gain: alapjeltől való eltérés jövőbeli becsléséből meghatározott érték, amely mutatja, hogy ilyen gyorsan reagál a hibára.

Block Parameters: PID Controller

PID 1dof (mask) (link)

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **PID** Form: **Ideal**

Time domain:

☒ Continuous-time

☐ Discrete-time

Discrete-time settings

Sample time (-1 for inherited): **-1**

Compensator formula

$$P \left(1 + I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}} \right)$$

Main Initialization Output Saturation Data Types State Attributes

Controller parameters

Source: **internal**

Proportional (P): **1**

Integral (I): **0.01**

Derivative (D): **0.01**

☒ Use filtered derivative

Filter coefficient (N): **100**

Automated tuning

Select tuning method: **Transfer Function Based (PID Tuner App)** **Tune...**

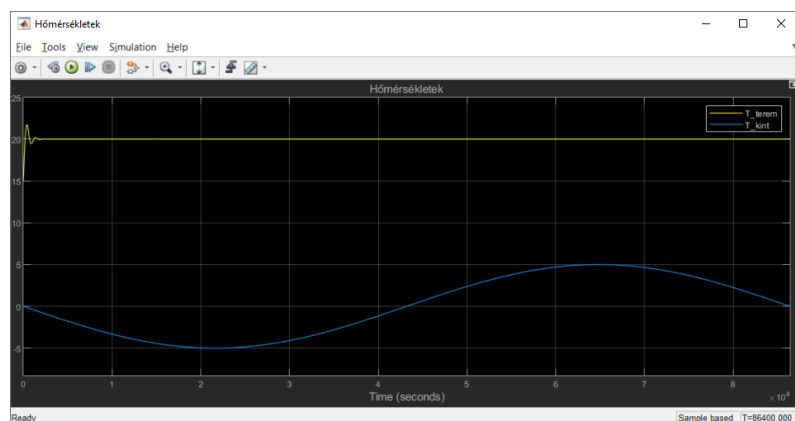
☒ Enable zero-crossing detection

OK **Cancel** **Help** **Apply**

4.23. ábra „PID Controller” blokk paraméterei

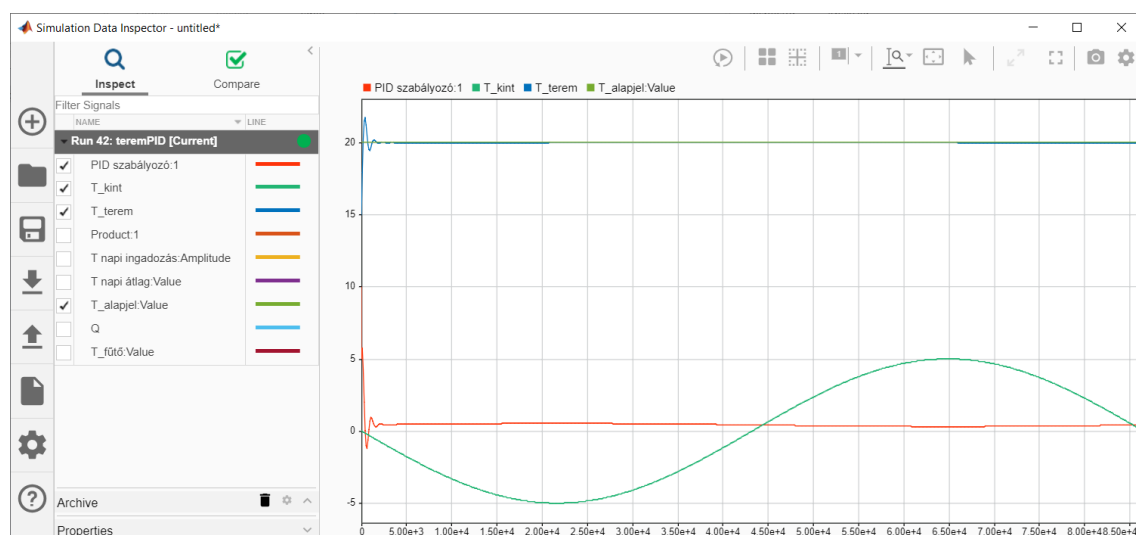
A PID szabályozó kimenetén kapott érték a „Signal” outport-ra kerül.

A PID szabályozóval kiegészített Simulink modellel is elvégeztem a 4.6.1 fejezetben bemutatott esettanulmányokat. Ezek közül elsőként a PID szabályozó működését ismertetem. Itt az alapjel értéke 20 °C, a terem kezdeti hőmérséklete 15 °C, a külső átlaghőmérséklet 0 °C, napi ingadozása 5 °C. A terem és a külső hőmérséklet értékek idődiagramját a 4.24. ábra szemlélteti. Látható, hogy a terem hőmérséklete a kezdeti ingadozás után szépen követi az alapjelet.



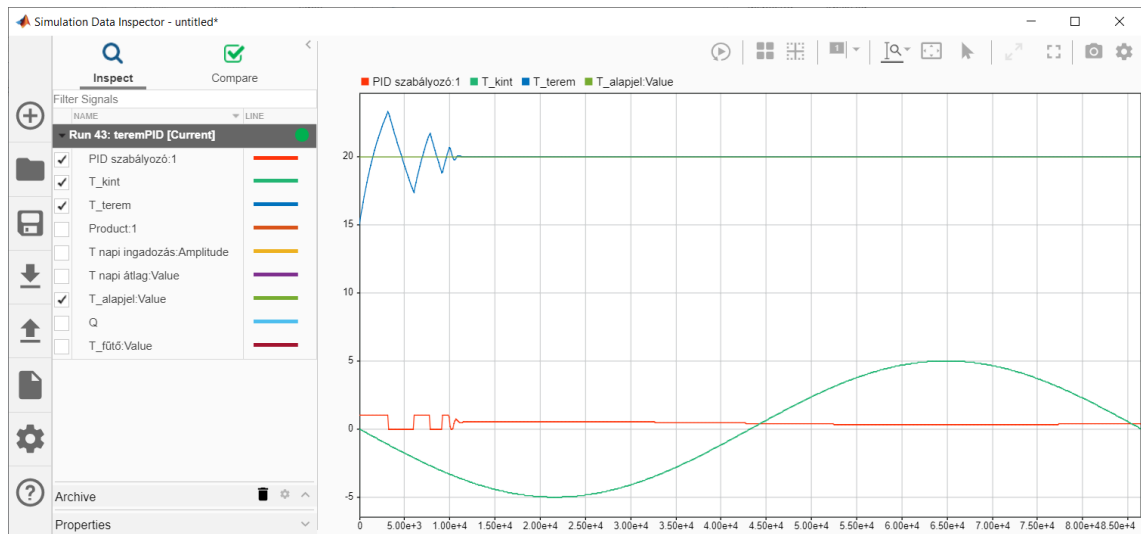
4.24. ábra PID szabályozó működése

A terem hőmérsékletét, a külső hőmérsékletet, az alapjelet, valamint a PID szabályozó kimeneti jelét Data Inspector-ban megjelenítve a 4.25. ábra tartalmazza.



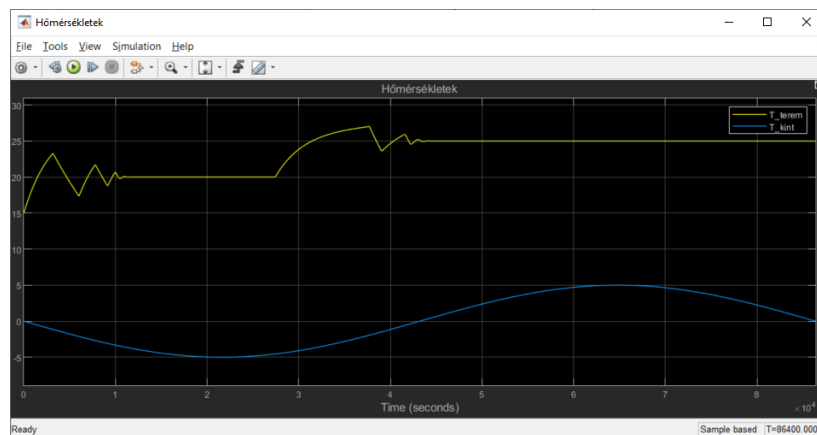
4.25. ábra PID szabályozó - megjelenítés Data Inspector-ban megjelenítve

Az ábrán látható, hogy a szabályozó kimeneti jele (piros színű jel) nem korlátos, emiatt kezdetben „lökésszerű” melegítés és hűtés is megjelenik. Ezek kiküszöbölésére az output jel értékét $[0,1]$ intervallumra korlátoztam (ami a korábban vizsgált ON/OFF szabályozó korlátaival egyezett). Az így kapott szimuláció eredményét a 4.26. ábra mutatja.

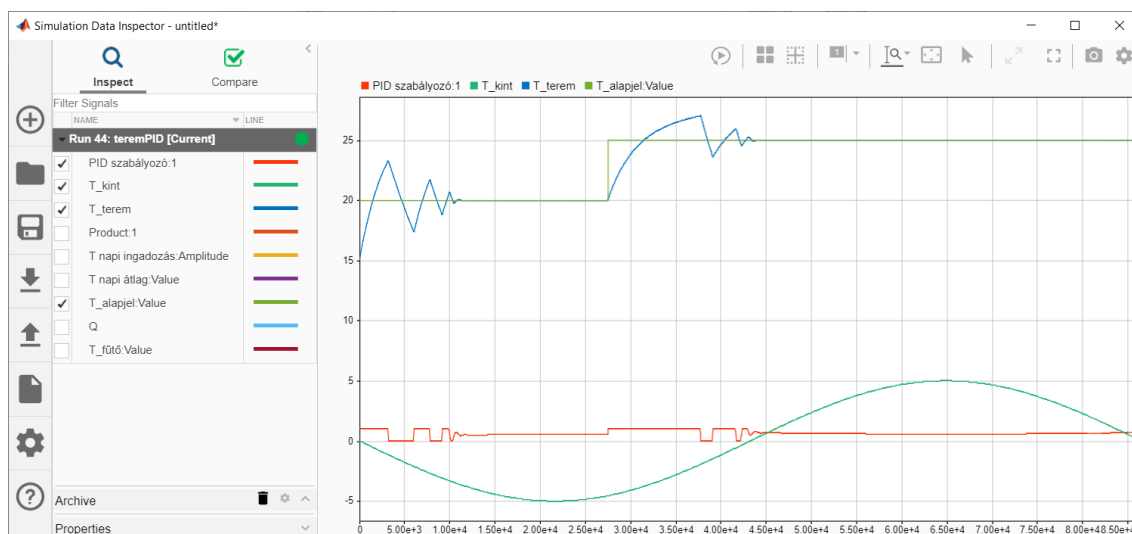


4.26. ábra PID szabályozó output korlátokkal - megjelenítés Data Inspector-ban megjelenítve

A következő vizsgálatom az alapjel módosítása 20 °C-ról 25 °C-ra, amelynek szimulációját a 4.27. ábra, valamint a 4.28. ábra mutatja.

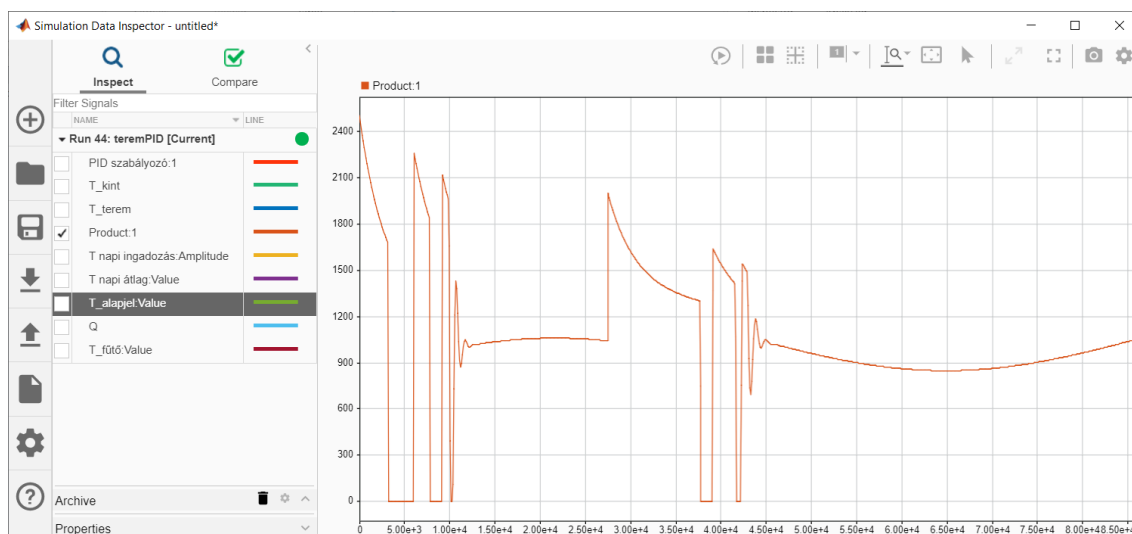


4.27. ábra PID szabályozó - alapjel változtatása



4.28. ábra PID szabályozó - alapjel változtatása Data Inspector-ban megjelenítve

A 4.29. ábra mutatja a fűtőberendezés által átadott hőenergiát a vizsgált alapjel módosítás folyamata során.



4.29. ábra PID szabályozó - energiaátadás

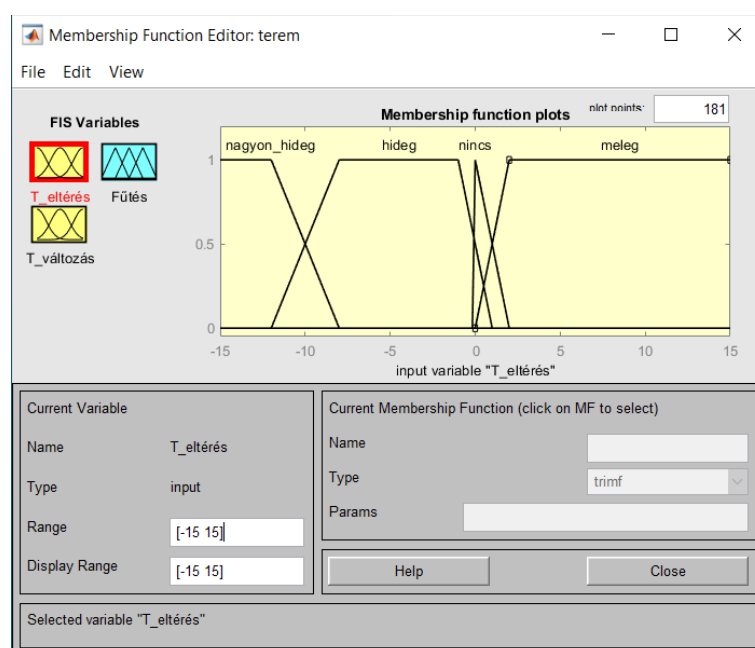
4.6.3 Fuzzy szabályozó

A fuzzy szabályozó megvalósítása során első lépésként megterveztem a fuzzy rendszert, az ehhez tartozó bemeneteket és a kimenetet. Fontos, hogy milyen tagsági függvényeket alakítok ki, hogy az a későbbiekben jól működjön a Simulink modellemmel is. Ezért olyan input változókat választottam, amelynek értéke a modellből meghatározható.

Két bemenetet készítettem. Az egyik neve „T_eltérés”. Ez azt mutatja meg, hogy mennyi az eltérés a teremben a beállított hőmérséklethez képest. Ehhez négy tagsági függvényt definiáltam. Ennek tulajdonságait az 1. táblázat, a Matlab Fuzzy Toolbox használatával készített fuzzy halmazokat pedig a 4.30. ábra tartalmazza.

1. táblázat: A „T_eltérés”.bemenet tagsági függvényei

	Tagsági függvény neve	Típusa	Kitüntetett pontjai
A1	nagyon_hideg	trapéz	(-20 -15 -12 -8)
A2	hideg	trapéz	(-12 -8 -1 1)
A3	nincs	háromszög	(0 0 2)
A4	meleg	trapéz	(0 2 15 17)



4.30. ábra: „T_eltérés” fuzzy halmazai

Ügyeltem arra, hogy a tagsági függvények között legyen átfedés. A trapéz alakú tagsági függvények mellett a „nincs” tagsági függvény azért lett „keskeny” háromszög alakú, mert egy szűk intervallum az, ahol az eltérés minimális, ezért nem kell majd változtatni a fűtést.

A következőkben matematikai formában írom le a tagsági függvényeket:

$$A1 = \begin{cases} 1, & \text{ha } x \leq -12^\circ\text{C} \\ (-8^\circ\text{C} - x)/4^\circ\text{C}, & \text{ha } -12^\circ\text{C} < x < -8^\circ\text{C} \\ 0, & \text{ha } x \geq -8^\circ\text{C} \end{cases}$$

$$A2 = \begin{cases} 0, & \text{ha } x \leq -12^\circ\text{C} \text{ vagy } x \geq 1^\circ\text{C} \\ (x - (-12^\circ\text{C}))/4^\circ\text{C}, & \text{ha } -12^\circ\text{C} < x < -8^\circ\text{C} \\ 1, & \text{ha } -8^\circ\text{C} \leq x \leq -1^\circ\text{C} \\ (1^\circ\text{C} - x)/2^\circ\text{C}, & \text{ha } -1^\circ\text{C} < x < 1^\circ\text{C} \end{cases}$$

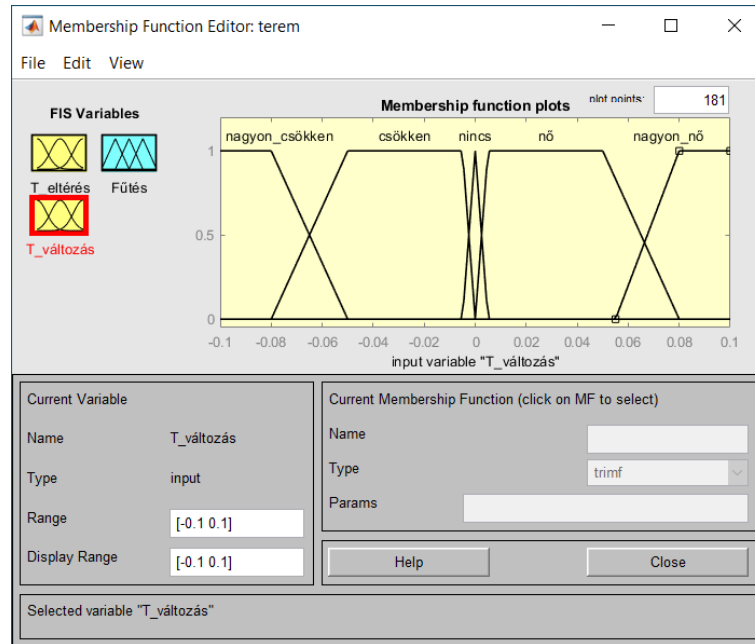
$$A3 = \begin{cases} 0, & \text{ha } x < 0^\circ\text{C} \text{ vagy } x \geq 2^\circ\text{C} \\ 1, & \text{ha } x = 1^\circ\text{C} \\ (2^\circ\text{C} - x)/2^\circ\text{C}, & \text{ha } 0^\circ\text{C} < x < 2^\circ\text{C} \end{cases}$$

$$A4 = \begin{cases} 0, & \text{ha } x \leq 0^\circ\text{C} \\ (x - 0^\circ\text{C})/2^\circ\text{C}, & \text{ha } 0^\circ\text{C} < x < 2^\circ\text{C} \\ 1, & \text{ha } x \geq 2^\circ\text{C} \end{cases}$$

A második bemenetnek „T_változás” nevet adtam. Ez azt mutatja meg hogyan változik a szoba hőmérséklete egységnyi idő alatt. 5 tagsági függvényt készítettem, négy trapéz és egy háromszög alakú görbét. A 2. táblázatban láthatóak a tagsági függvények megnevezései, azok típusa és a hozzá tartozó kitüntetett pontok. A 4.31. ábra a Fuzzy Toolbox-szal elkészített fuzzy halmazokat jeleníti meg.

2. táblázat: A „T_változás” bemenet tagsági függvényei

	Tagsági függvény neve	Típusa	Kitüntetett pontjai
A1	nagyon_csökken	trapéz	(-0.145 -0.1 -0.08 -0.05)
A2	csökken	trapéz	(-0.08 -0.05 -0.005 0)
A3	nincs	háromszög	(-0.005 0 0.005)
A4	nő	trapéz	(0 0.005 0.05 0.08)
A5	nagyon_nő	trapéz	(0.055 0.08 0.1 0.15)

4.31. ábra $T_változás$ fuzzy halmazai

A tagsági függvények matematikai formában a következőképpen írhatók le:

$$A1 = \begin{cases} 1, & \text{ha } x \leq -0.08 \\ (-0.05 - x)/0.03, & \text{ha } -0.08 < x < -0.05 \\ 0, & \text{ha } x \geq -0.05 \end{cases}$$

$$A2 = \begin{cases} 0, & \text{ha } x \leq -0.08 \text{ vagy } x \geq 0 \\ (x - 0.08)/0.03, & \text{ha } -0.08 < x < -0.05 \\ 1, & \text{ha } -0.05 \leq x \leq -0.005 \\ (0 - x)/0.005, & \text{ha } -0.005 < x < 0 \end{cases}$$

$$A3 = \begin{cases} 0, & \text{ha } x \leq -0.005 \text{ vagy } x \geq 0.005 \\ (x - (0.005))/0.005, & \text{ha } -0.05 < x < 0 \\ (0 - x)/0.005, & \text{ha } 0 < x < 0.005 \end{cases}$$

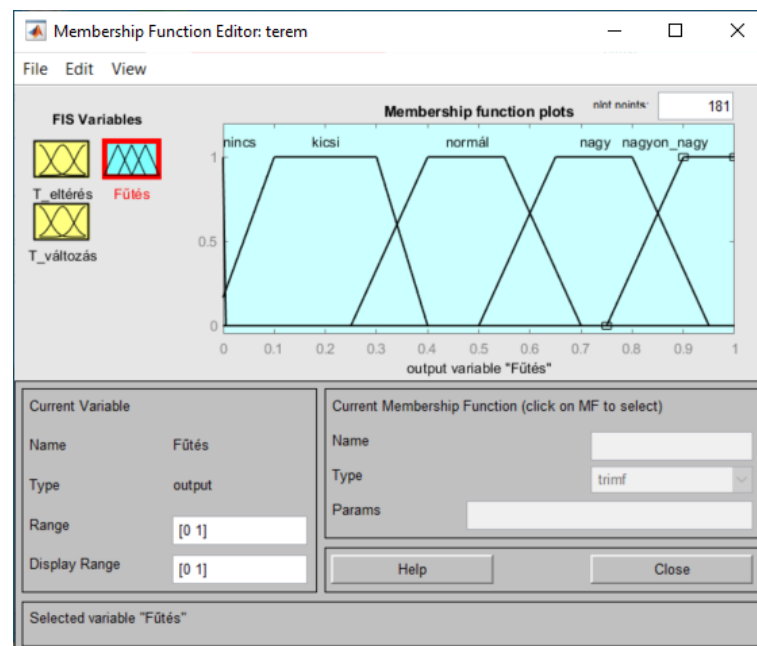
$$A4 = \begin{cases} 0, & \text{ha } x \leq 0 \text{ vagy } x \geq 0.08 \\ (x - 0)/0.005, & \text{ha } 0 < x < 0.005 \\ 1, & \text{ha } 0.005 \leq x \leq 0.05 \\ (0.08 - x)/0.03, & \text{ha } 0.05 < x < 0.08 \end{cases}$$

$$A5 = \begin{cases} 0, & \text{ha } x \leq 0.055 \\ (x - 0.055)/0.025, & \text{ha } 0.055 < x < 0.08 \\ 1, & \text{ha } x \geq 0.08 \end{cases}$$

Ezt követően a kimenetet készítettem el. Egy kimenet van, amely a „Fűtés” nevet kapta. Öt tagsági függvényt készítettem, ezek szintén trapéz és háromszög alakú görbék. A tagsági függvények tulajdonságait a 3. táblázat, a Fuzzy Toolbox-szal megjelenített tagsági függvényeket a 4.32. ábra mutatja be.

3. táblázat: „Fűtés” kimenet tagsági függvényei

	Tagsági függvény neve	Típusa	Kitüntetett pontjai
A1	nincs	háromszög	(-0.25 0 0.005)
A2	kicsi	trapéz	(-0.02 0.1 0.3 0.4)
A3	normál	trapéz	(0.25 0.4 0.55 0.7)
A4	nagy	trapéz	(0.5 0.65 0.8 0.95)
A5	nagyon_nagy	trapéz	(0.75 0.9 1 1.2)



4.32. ábra „Fűtés” fuzzy halmazai

Egyenletek formában a tagsági függvények a következőképpen jelennek meg:

$$A1 = \begin{cases} 1, & \text{ha } x \leq 0 \\ (0.005 - x)/0.005, & \text{ha } 0 < x < 0.005 \\ 0, & \text{ha } x \geq 0.005 \end{cases}$$

$$A2 = \begin{cases} 0, & \text{ha } x \geq 0.4 \\ (x - (-0.02))/0.08, & \text{ha } -0.02 < x < 0.1 \\ 1, & \text{ha } 0.1 \leq x \leq 0.3 \\ (0.4 - x)/0.1, & \text{ha } 0.3 < x < 0.4 \end{cases}$$

$$A3 = \begin{cases} 0, & \text{ha } x \leq 0.25 \text{ vagy } x \geq 0.7 \\ (x - 0.25)/0.15, & \text{ha } 0.25 < x < 0.4 \\ 1, & \text{ha } 0.4 \leq x \leq 0.55 \\ (0.7 - x)/0.15, & \text{ha } 0.55 < x < 0.7 \end{cases}$$

$$A4 = \begin{cases} 0, & \text{ha } x \leq 0.5 \text{ vagy } x \geq 0.95 \\ (x - 0.5)/0.15, & \text{ha } 0.5 < x < 0.65 \\ 1, & \text{ha } 0.65 \leq x \leq 0.8 \\ (0.95 - x)/0.15, & \text{ha } 0.8 < x < 0.95 \end{cases}$$

$$A5 = \begin{cases} 0, & \text{ha } x \leq 0.75 \\ (x - 0.75)/0.15, & \text{ha } 0.75 < x < 0.9 \\ 1, & \text{ha } x \geq 0.9 \end{cases}$$

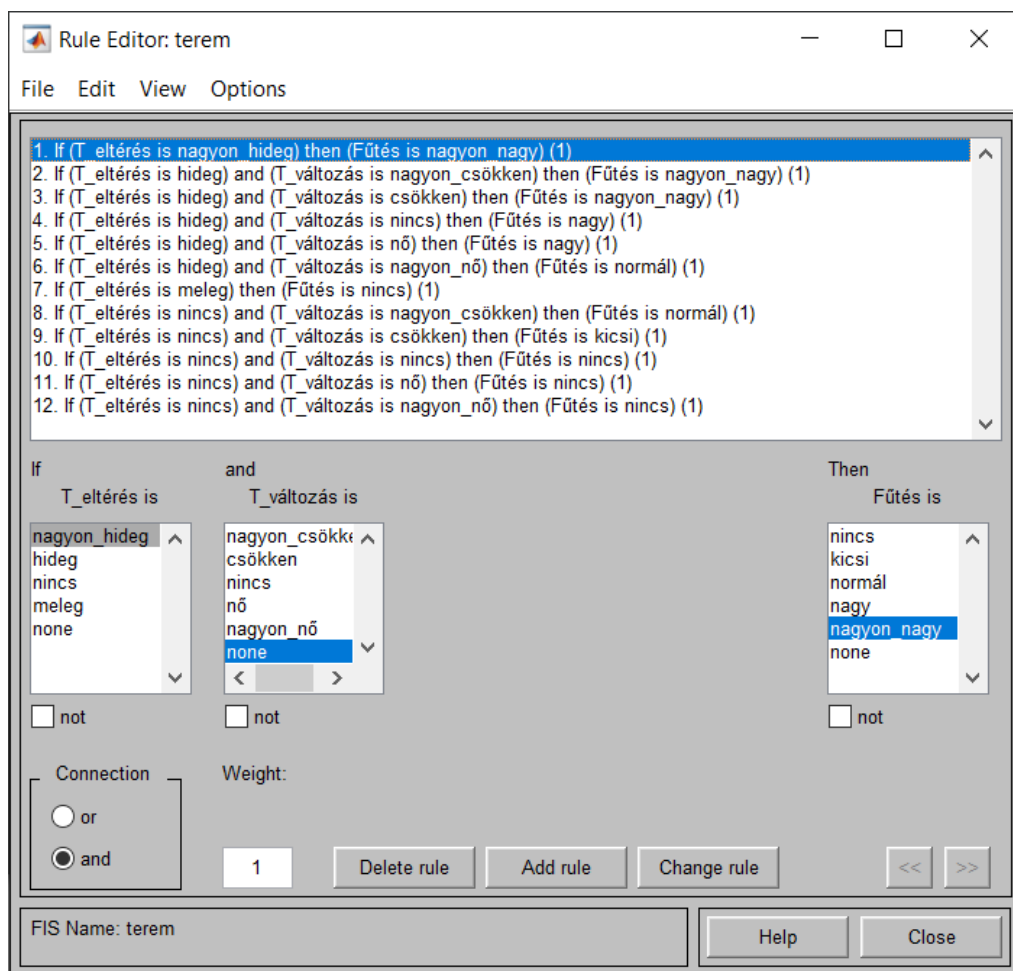
A bemeneti és a kimeneti tagsági függvények megtervezése és elkészítése után következett a szabályok elkészítése. 12 szabályt állítottam fel, ügyelve, hogy minden olyan lehetőséget figyelembe vegyek, ami előfordulhat a hőmérsékletszabályozás során.

Ezek a szabályok a következők:

1. Ha a T_eltérés nagyon hideg akkor a Fűtés legyen nagyon nagy.
2. Ha a T_eltérés hideg és a T_változás nagyon csökken akkor a Fűtés legyen nagyon nagy.
3. Ha a T_eltérés hideg és T_változás csökken akkor a Fűtés legyen nagyon nagy.
4. Ha a T_eltérés hideg és T_változás nincs akkor a Fűtés legyen nagy.
5. Ha a T_eltérés hideg és T_változás nő akkor a Fűtés legyen nagy.
6. Ha a T_eltérés hideg és T_változás nagyon nő akkor a Fűtés legyen normál.
7. Ha a T_eltérés meleg akkor nincs Fűtés.
8. Ha a T_eltérés nincs és a T_változás nagyon csökken akkor a Fűtés legyen normál.
9. Ha a T_eltérés nincs és a T_változás csökken akkor a Fűtés kicsi legyen.
10. Ha a T_eltérés nincs és a T_változás nincs akkor nincs Fűtés.
11. Ha a T_eltérés nincs és a T_változás nő akkor nincs Fűtés.

12. Ha a T_eltérés nincs és a T_változás nagyon nő akkor nincs Fűtés.

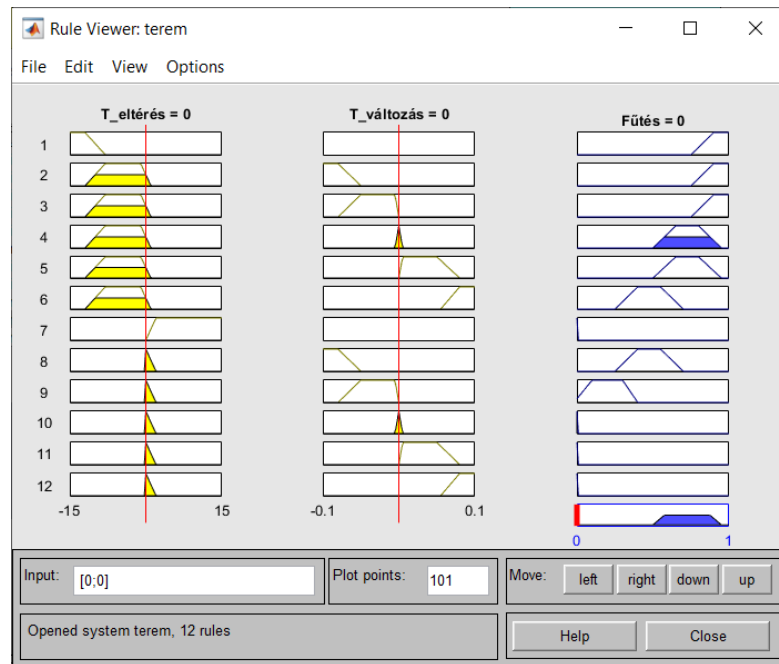
A megtervezett szabályok Fuzzy Toolbox-szal definiált formáját a 4.33. ábra mutatja.



4.33. ábra Fuzzy szabályok

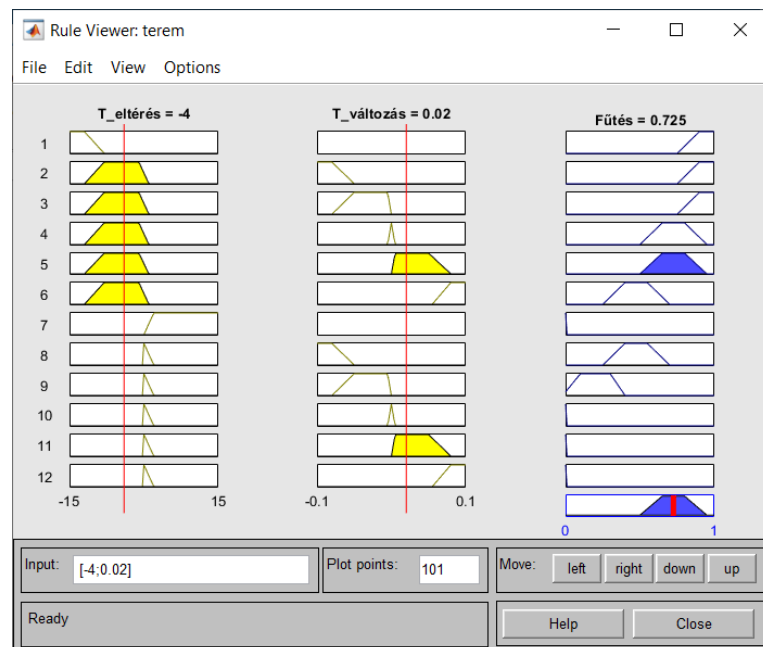
A szabályok elkészítését követően megnéztem a View/Rules nézetet, melyben kirajzóldtak az általam definiált szabályok. Ezt az 4.34. ábra szemlélteti.

Minden sor 1-1 szabályt képvisel, így 12 sor látható. Az első két oszlop a szabály „if” része, az utolsó oszlop pedig a „then” részét, a következtetést mutatja. Az utolsó oszlop legalsó táblájában történik a fuzzy kombináció, ahol megjelenik a defuzzifikáció eredménye is, azaz a fuzzy szabályozó jel valós értéke (piros függőleges vonallal). Látható, hogy amennyiben a „T_eltérés” =0 és a „T_változás” =0, akkor a „Fűtés” értéke 0.



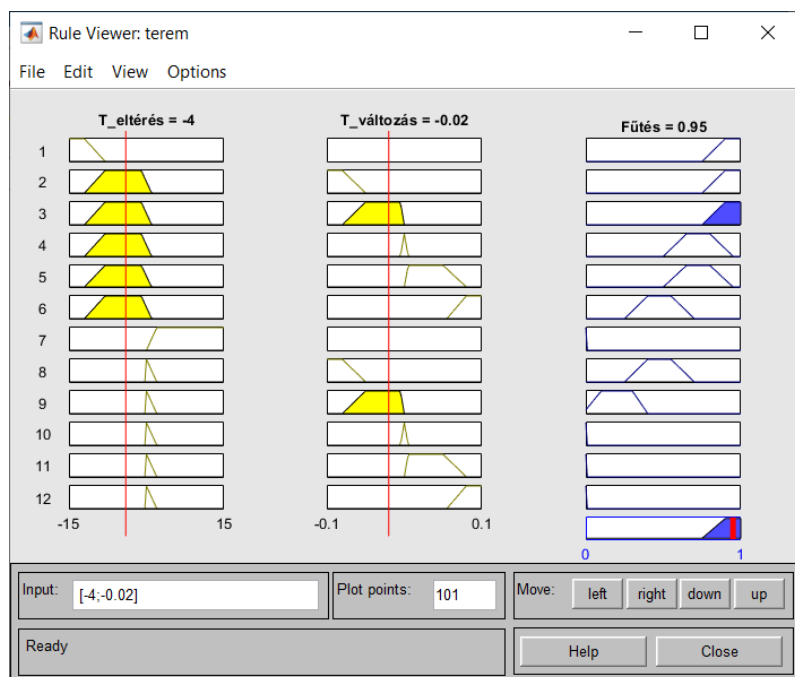
4.34. ábra Következtetés fuzzy szabályokkal – 1. eset

A 4.35. ábra azt az esetet mutatja, amikor a $T_{\text{eltérés}}$ értéke -4 (a teremben 4 °C-kal hidegebb van, mint a beállított érték) és a $T_{\text{változás}}$ értéke 0.02 (azaz melegszik a terem), akkor a $Fűtés$ értéke 0.725.



4.35. ábra Következtetés fuzzy szabályokkal – 2. eset

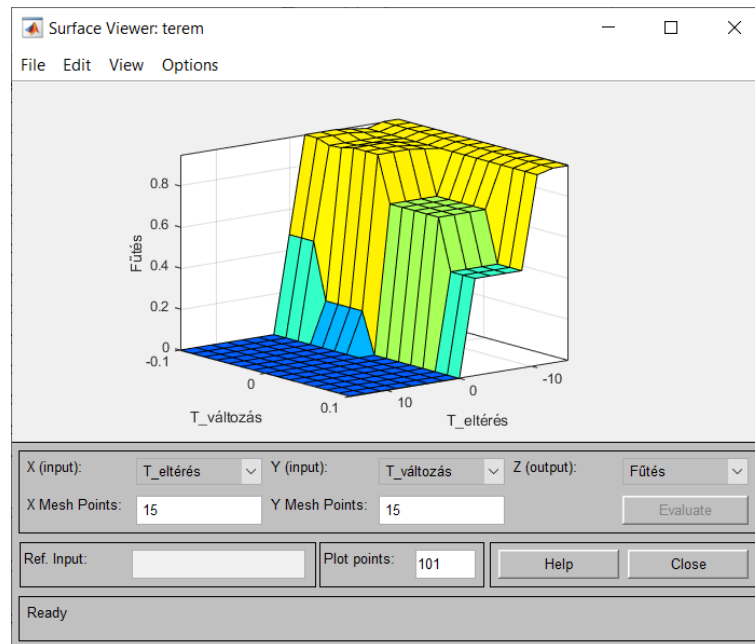
A 4.36. ábra esetén a $T_eltérés$ értékét nem változtattam (a teremben 4 °C-kal hidegebb van, mint a beállított érték), azonban a $T_változás$ értékét -0.02-re állítottam (azaz hűl a terem), akkor a $Fűtés$ értéke 0.95-re növekedett.



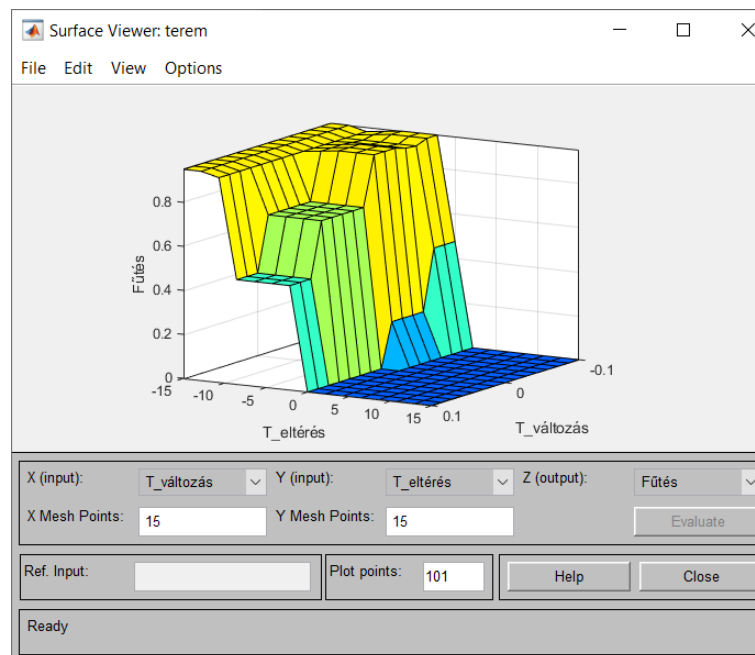
4.36. ábra Következtetés fuzzy szabályokkal – 3. eset

Ezzel a szabálynézővel egyszerre csak egy input kombináció hatását tudjuk megnézni. A következőben a szabály felületnézőt (Surface Viewer) fogom használni, melyben grafikusán láthatjuk a rendszer teljes kimenetét.

A felületeket két nézőpontban jelenítettem meg. Először X tengelynek a „ $T_eltérés$ ”-t választottam (4.37. ábra), az Y tengely a „ $T_változás$ ” lett, második esetben pedig felcseréltem ezt a két tengelyt (4.38. ábra).

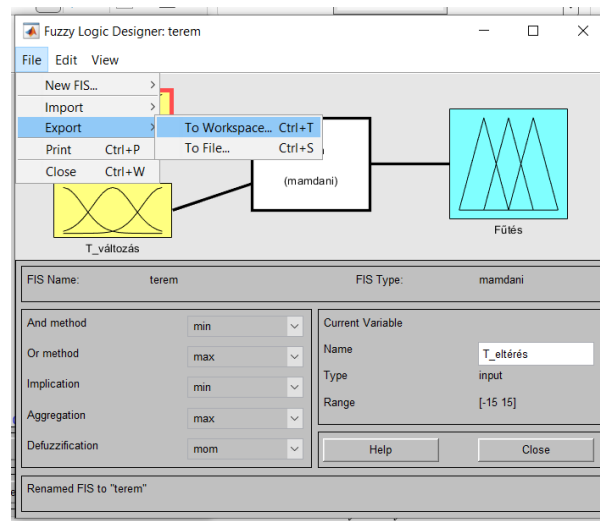


4.37. ábra Felület 1.



4.38. ábra Felület 2.

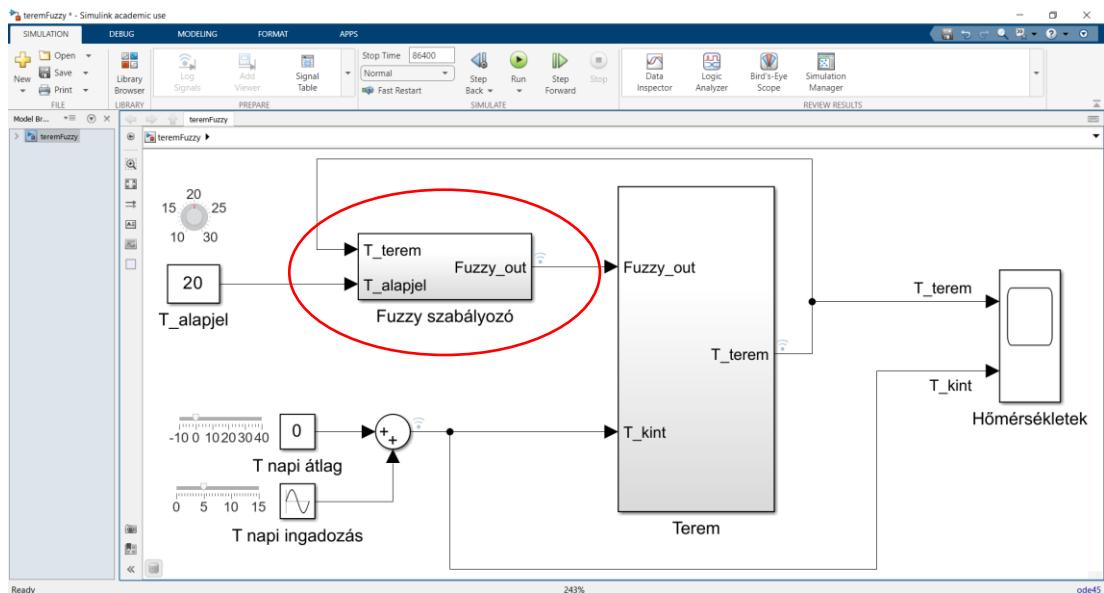
Utolsó lépésként kimentettem az állományomat To workspace formátumba, hogy a Simulink programmal készített modellemben be tudjam illeszteni. Jelen esetben „terem” néven mentettem el. Ezt a 4.39. ábra szemlélteti.



4.39. ábra Fuzzy rendszer mentése Matlab Workspace-ként

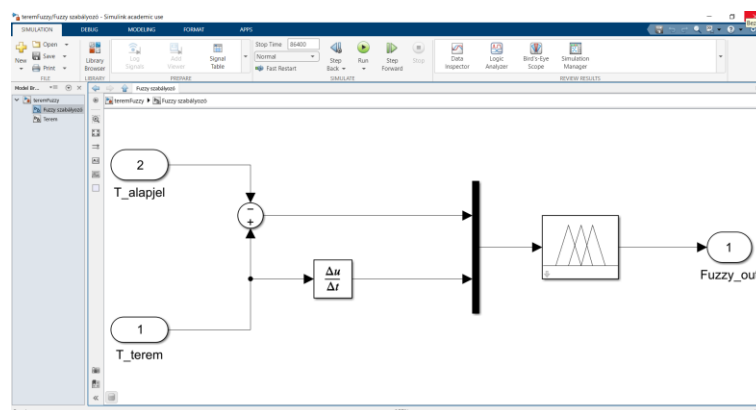
A következő lépésben szeretném bemutatni, a Simulink modellel keresztül, hogy működik az elkészített fuzzy szabályozó.

A 4.40. ábra mutatja be a Fuzzy szabályozóval kiegészített Simulink modellt. Hasonlóan a korábban bemutatott modellekhez, itt is maradt a blokkok struktúrája, azok szerepe is ugyanaz, ami változott, az a Fuzzy szabályozó.



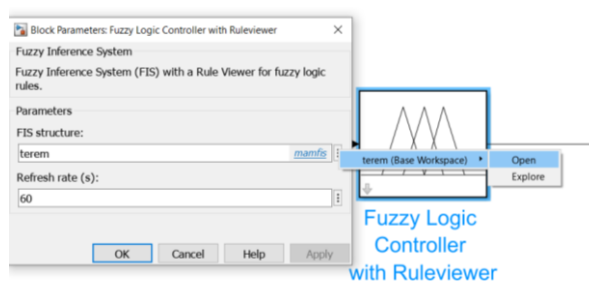
4.40. ábra Simulink modell – Szabályozó részrendszer Fuzzy szabályozóval

A 4.41. ábra mutatja a Fuzzy szabályozó részrendszert, ahol látszik, hogy két bemenete van a bloknak, egyik a terem aktuális hőmérséklete, a másik pedig a beállított alapjel. A $\frac{\Delta u}{\Delta t}$ jelölésű „Derivative” blokk adja meg, hogy egységnyi idő alatt mennyit változik a hőmérséklet, ami lehet növekedés vagy csökkenés.



4.41. ábra Fuzzy szabályozó részrendszer

A Fuzzy Logic Controller with Ruleviewer blokkba kell megadni és betölteni a Fuzzy rendszerből kimentett állományt (4.42. ábra). A bloknak egy kimenete van, ami a „T_alapjel” -et és a „T_változás”-t tartalmazó kételemű vektor, a kimeneti értéke pedig a „Fuzzy_out” nevű „Outport” blokk.

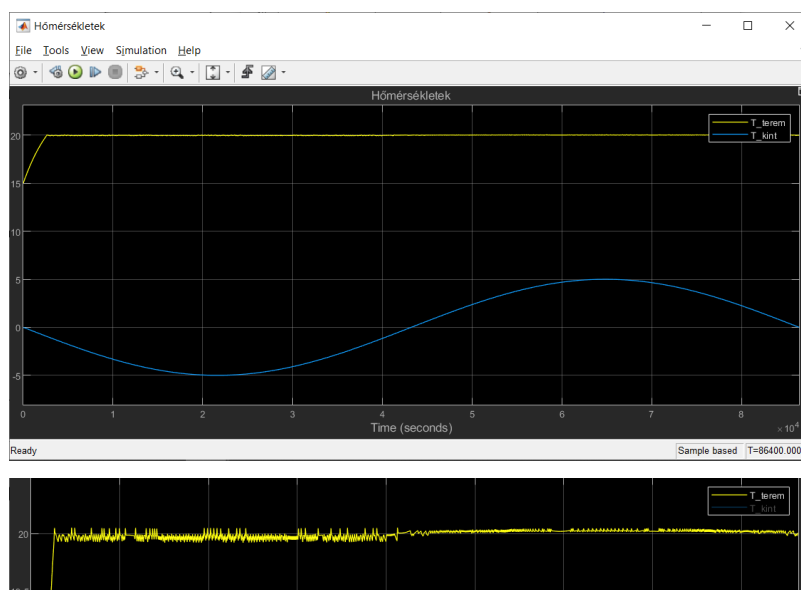


4.42. ábra Fuzzy Logic Controller

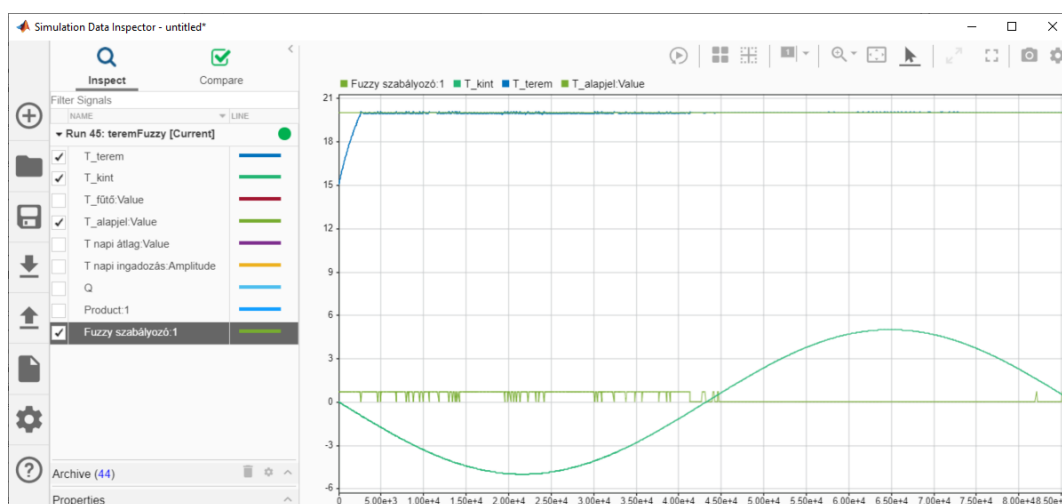
A Fuzzy szabályozó működését a Simulink modellemben az előző két szabályozóval elvégzett szimulációkhoz hasonlóan illusztrálom.

Az első esettanulmányban, alapjelnek 20 °C-ot állítottam be, napi átlag 0 °C az ingadozás +/- 5 °C. 24 óra időtartamban vizsgáltam ebben az esetben is a szabályozót. Amikor a

futtatást elindítom, automatikusan megjelenik a Fuzzy rendszer Rule Viewer nézetében a terem szabályaival történő következtetés, ami folyamatosan mutatja a szabályok „működését” a szimuláció időtartama alatt (ahogy erre a 4.34. ábra-4.36. ábra adott példát). A szimuláció eredményét a 4.43. ábra szemlélteti (ahol nagyítva is látható a „T_terem” hőmérsékletváltozása), a Data Inspector-ban megjelenített értékeket pedig a 4.44. ábra mutatja. Az ábrákról látható, hogy minimális ingadozással tartja az alapjel értékét a terem hőmérséklete.

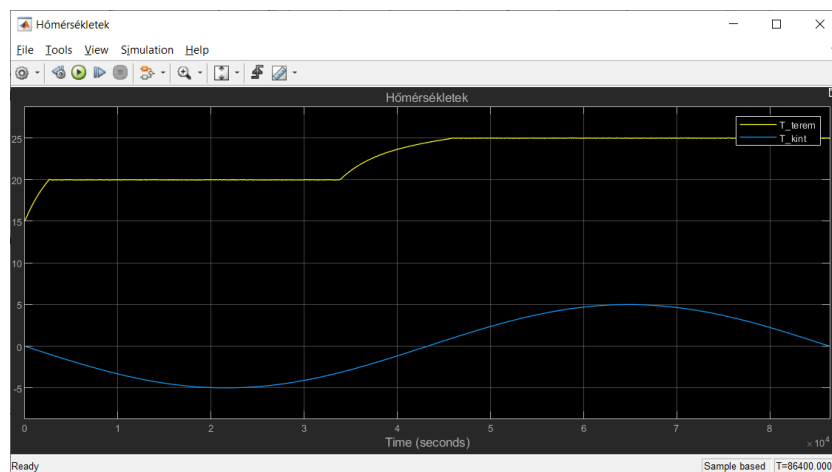


4.43. ábra Fuzzy szabályozó működése



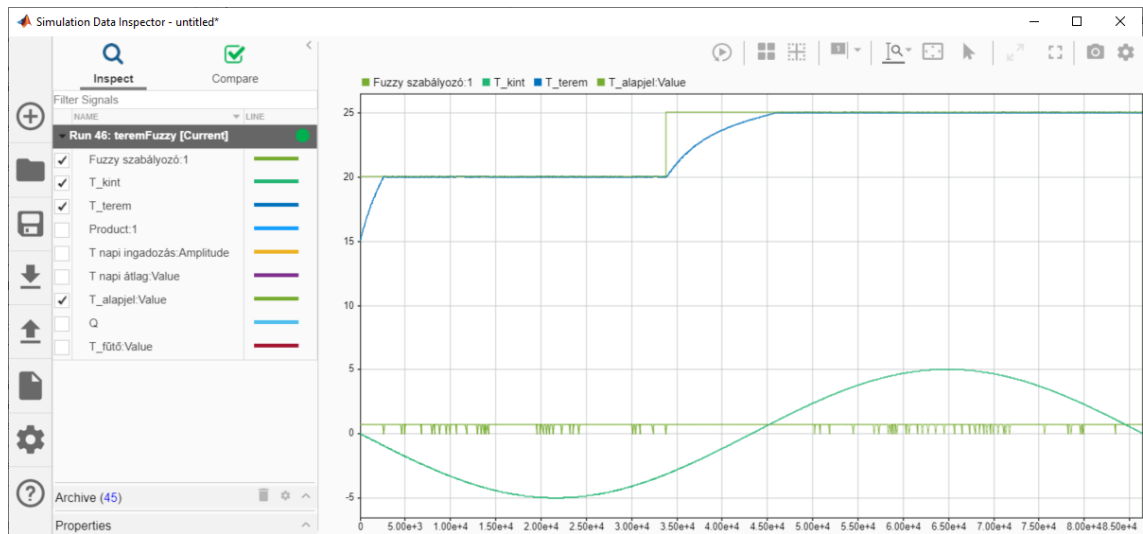
4.44. ábra Fuzzy szabályozó - megjelenítés Data Inspector-ban megjelenítve

A következő lépésben vizsgáltam az alapjel-váltást 20 °C-ról 25 °C-ra a többi paraméter változatlanul hagyása mellett. Itt azt figyeltem meg, hogy ha változtatom az alapjelet, hogyan változik a terem hőmérséklete. Mennyi időre van szükség, hogy beálljon az új alapjelre. A 4.45. ábra mutatja, hogy a 20 °C terem hőmérsékletet a kezdeti állapotból aránylag hamar el lehet érni, viszont mikor az alapjel 25 fokra lett állítva, akkor több időre volt szüksége a rendszernek, hogy elérje a kívánt hőmérsékletet.



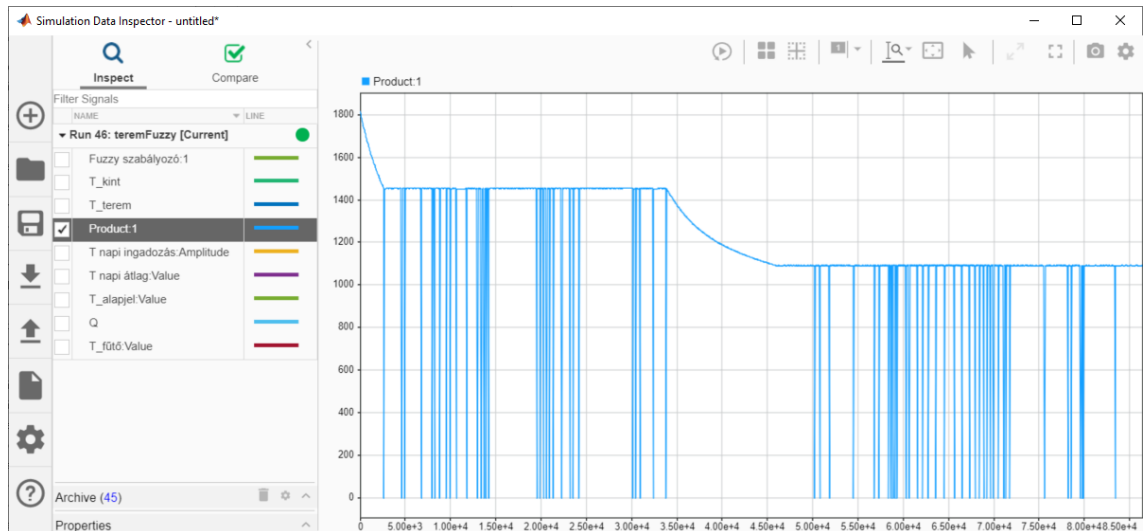
4.45. ábra Fuzzy szabályozó - alapjel változtatása

A következő ábrán (4.46. ábra) lehetőségünk van további változók megjelenítésére. Amit fontosnak tartok kiemelni egy megjelenítésben, hogy a külső hőmérsékletváltozás és az alapjel-változás hatásaként megfigyelhető, hogy milyen sűrűn, milyen értékekre vált a szabályozó output jele (Fuzzy szabályozó 1).



4.46. ábra Fuzzy szabályozó - alapjel változtatása Data Inspector-ban megjelenítve

A 4.47. ábra mutatja a fűtőberendezés által átadott hőenergiát a vizsgált alapjel módosítás folyamata során.



4.47. ábra Fuzzy szabályozó - energiaátadás

5. ÉRTÉKELÉS, TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

Mindhárom vizsgált szabályozóról megállapítható, hogy jól megoldható vele egy helyiség hőmérsékletszabályozása.

Az ON/OFF szabályozó a működési elvének megfelelően az alapjel közelében tartja a hőmérsékletet. Az alapjel környezetében minél kisebb a megadott hibaintervallum, annál gyakrabban kapcsol be. Az alapjelváltást és a külső hőmérséklet változását jól követi. A szabályozó nagy előnye az egyszerűség és az ütemezhető kapcsolási funkció. Hátránya, hogy csak két állapotot ismer, amely meghatározza a teljes teljesítményt és a kikapcsolást.

A PID szabályozónál az alapjel módosításánál először nagy a kilengése a jelnek, túllendülés van, de mindig közelít az alapjelhez, ameddig el nem éri. A paraméterek jó beállításával nagyon hatékonyan működtethető. Ezek beállításához számos módszer használható [11], azonban ezek irányításelméleti ismereteket feltételeznek.

Az elkészített Fuzzy szabályozó hasonlított a PID szabályozóhoz. Mivel a terem hőmérséklet és az alapjel eltérését, valamint a terem hőmérsékletének változását használtam a szabályozó bemeneteként, egy P és D taggal rendelkező szabályozót valósítottam meg a fuzzy logika eszközeivel. A megvalósított szabályozó az alapjel és a külső hőmérséklet változását egyaránt jól követte. Nagy előnye, hogy az emberi gondolkodáshoz hasonló szabályok formájában lehetett leírni a rendszer működését, ami minimális irányításelméleti tudással is megvalósítható volt.

Természetesen, mint minden munka, ez is továbbfejleszthető, folytatható. További simulációk elvégzésével meghatározható például, hogy milyen hőmérséklet értéket érdemes beállítani a teremben ahhoz, hogy egyrészt energiahatékonyan, másrészt jó közérzetet biztosítva működtethessük a rendszert. Ehhez figyelembe vehető például Fuzzy szabályozó esetén a teremben levő hallgatók száma, a napszakok, az árnyékolás és sok egyéb tényező.

6. ÖSSZEFOGLALÓ

A diplomamunkám során Matlab Toolbox, azon belül a Simulink és a Fuzzy rendszer segítségével három szabályozó rendszer kialakítását végeztem el, amellyel egy terem hőmérsékletszabályozása megvalósítható. Ezek a következők: ON/OFF szabályozó, PID szabályozó és a Fuzzy szabályozó. A rendszereket egy oktatási teremre terveztem meg, az elkészített Simulink modellben a terem megfelelő paramétereit használtam. A terem kiválasztásánál fontos szempont volt számomra az, hogy ne gépterem legyen, hogy ne keljen többletenergiaival számolnom a modellemben a radiátorok által átadott hőenergián kívül és hogy ne földszinti terem legyen. A rendszer elemeinek elkészítése során fontos volt a tervezés szakasza, a cél meghatározása, felhasználva az általam kiválasztott paramétereket.

A terem Matlab Simulink modelljének elkészítése után a szabályozó blokkban először egy ON/OFF szabályozót valósítottam meg és vizsgáltam a működését, megnézve a rendszer előnyeit és hátrányait.

Második módszerként egy PID szabályozót készítettem. Itt is különböző beállítások mellett szimuláltam a rendszeremet.

Harmadik lépésként egy Fuzzy szabályozót készítettem el. Ehhez először a Matlab Fuzzy Toolbox-ban végeztem el a bemeneti és kimeneti tagsági függvények megtervezését, mi legyen az, amit figyelembe veszek. Két bemeneti változót készítettem, ezek közül egyik a terem hőmérsékletének az alapjeltől való eltérését mutatja, a másik pedig a terem aktuális hőmérsékletváltozását. Egy kimeneti változóm volt, ami a fűtés mértékét adta meg. Ezekhez fuzzy tagsági függvényeket definiáltam, majd elkészítettem a fuzzy szabályokat a rendszer működtetéséhez.

Mindhárom szabályozónál megnéztem a terem fűtési rendszerének működését, amelyet különböző esettanulmányokkal illusztráltam. A kapott eredmények megmutatták, hogy mind a három szabályozó jól működik és használható.

7. SUMMARY

In my thesis I developed three control systems for the temperature control of a room using Matlab Toolbox, including Simulink and Fuzzy. These are ON/OFF controller, PID controller and Fuzzy controller. The systems were designed for a classroom and the Simulink model was created using the appropriate parameters of the classroom. When selecting the room, it was important for me that it should not be a computer lab, that I should not have to account for any additional energy in my model other than the heat energy transferred by the radiators and that it should not be a ground floor room. When designing the elements of the system, it was important to go through the design phase and define the objective, using the parameters I had chosen.

After creating a Matlab Simulink model of the classroom, I first implemented an ON/OFF controller in the control block and tested its operation, looking at the advantages and disadvantages of the system.

As a second method, I created a PID controller. Here again I simulated my system under different settings.

As a third step, I created a fuzzy controller. To do this, I first used the Matlab Fuzzy Toolbox to design the input and output membership functions, what should be considered. I created two input variables, one of them representing the deviation of the room temperature from the setpoint, the other the actual temperature change of the room. I had one output variable, which was the heating rate. I defined fuzzy membership functions for these and then created the fuzzy rules to operate the system.

For each of the three controllers, I examined the operation of the room heating system, which I illustrated with various case studies. The results obtained showed that all three controllers work well and can be used.

8. IRODALOMJEGYZÉK

- [1] T. Vámos, R. Bars és D. Sik, „SYSBOOK – Rendszerekről és szabályozásokról több szinten,” in *Új módszerek a szakképzésben – Kollaboratív online tartalomfejlesztés*, Esztergom, Spori Print Vincze Nyomda, 2020, pp. 133-151.
- [2] S. Zoltán, *A digitális PID szabályozó*, Budapesti Műszaki és Gazdaságtudományi Egyetem: Elektronikus Eszközök Tanszéke, 2008.
- [3] L. A. Zadeh, „Fuzzy sets,” *Information and Control*, 8, 3, 1965.
- [4] L. T. Kóczy és D. Tikk, *Fuzzy rendszerek*, Budapest: TypoTEX, 2000.
- [5] L. R. Piglerné és W. Á. Starkné, *Ágens-Technológia*, TypoTEX, 2011.
- [6] J.-S. R. Jang és N. Gulley, *The Fuzzy Logic Toolbox for use with MATLAB*, Natick, Massachusetts: The MathWorks, Inc., 1995.
- [7] E. Peasley, *An Introduction to Using Simulink*, Department of Engineering Science, University of Oxford, 2018.
- [8] „Fuzzy Logic Toolbox,” MathWorks, [Online]. Available: <https://www.mathworks.com/products/fuzzy-logic.html>. [Hozzáférés dátuma: 3 május 2024].
- [9] N. Fernando, *Design of a Home Temperature Controller*, California State University, Northridge, 2016.
- [10] *Fuzzy Logic Toolbox™ User's Guide*, https://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy_ug.pdf: MathWorks, 2024.
- [11] M. Gerzson és S. Pletl, *Irányítástechnika*, Typotex, 2011.

- [12] I. S. Kobersi, V. I. Finaev, S. A. Almasani és K. W. A. Abdo, „Control of the Heating System with Fuzzy Logic,” *World Applied Sciences Journal*, 23, 11, pp. 1441-1447, 2013.
- [13] A. Taneva, D. Atanasova és A. Daskalov, „Fuzzy Logic Control in Building Automation Application,” *Engineering Proceedings*, 41, 14, p. 10, 2023.

9. ÁBRAJEGYZÉK

2.1. ábra Rendszer és elemei.....	3
2.2. ábra Egy rendszer modellje.....	3
2.3. ábra Rendszer és szabályozó.....	5
2.4. ábra Éles halmaz és fuzzy halmaz	9
2.5. ábra Háromszög alakú tagsági függvények	10
2.6. ábra Trapéz alakú tagsági függvények.....	10
2.7. ábra Gauss görbe tagsági függvények	10
2.8. ábra Hőmérséklet univerzumon értelmezett fuzzy halmazok.....	11
2.9. ábra Fuzzy halmazok uniója	12
2.10. ábra Fuzzy halmazok metszete	13
2.11. ábra Fuzzy halmaz komplementere	13
2.12. ábra Fuzzy szabályt definiáló fuzzy halmazok	14
2.13. ábra Fuzzy szabály végrehajtása.....	15
2.14. ábra Következtetés fuzzy szabályhalmazon	15
2.15. ábra Fuzzifikáció.....	17
2.16. ábra Defuzzifikációs módszerek	19
2.17. ábra Fuzzy szakértői rendszer.....	20
3.1. ábra Simulink indítása	23
3.2. ábra A Simulink nyitó oldala	23
3.3. ábra Egyszerű Simulink modell.....	24
3.4. ábra Blokkok paramétereinek megadása	24
3.5. ábra Szimuláció eredménye	25
3.6. ábra Library Browser megnyitása.....	25
3.7. ábra Simulink Library Browser	26
3.8. ábra Szimuláció indítása	27
3.9. ábra Matlab - Fuzzy Logic Toolbox interakciós képességei [6].....	29
3.10. ábra A Fuzzy Logic Toolbox grafikus felülete [6]	30
3.11. ábra Fuzzy Logix Toolbox általános szerkesztő felülete.....	31

3.12. ábra Fuzzy műveletek módosítása	32
3.13. ábra Tagsági függvény szerkesztő – input változók	32
3.14. ábra Tagsági függvények hozzáadása	34
3.15. ábra Tagsági függvény szerkesztő - kimeneti változó	34
3.16. ábra Szabály szerkesztő	35
3.17. ábra Szabály szerkesztővel létrehozott szabályhalmaz.....	36
3.18. ábra Szabály megjelenítő	37
3.19. ábra Felület megjelenítő.....	39
4.1. ábra A 103. előadóterem fényképe (saját fotó).....	41
4.2. ábra A 103. előadóterem metszete (saját ábra)	42
4.3. ábra Modell elkészítésének folyamata	43
4.4. ábra A vizsgált rendszer Simulink modellje	44
4.5. ábra „Scope” blokk konfigurációs paramétereinek beállítása	45
4.6. ábra Simulink modell - Terem részrendszer	47
4.7. ábra Terem részrendszer és annak blokkjai	48
4.8. ábra Simulink modell - Külső hőmérséklet szimulálása.....	49
4.9. ábra Külső hőmérséklet	50
4.10. ábra Simulink modell – Hőmérsékletek megjelenítése.....	50
4.11. ábra Simulink modell – Szabályozó részrendszer az alapjellel	51
4.12. ábra T_alapjel beállítási paramétereit	52
4.13. ábra Az ON/OFF szabályozó részrendszer	53
4.14. ábra Az „On-Off kapcsoló” paramétereit	54
4.15. ábra Az ON/OFF szabályozó működése.....	54
4.16. ábra Jelek megjelenítése	55
4.17. ábra ON/OFF szabályozó - megjelenítés Data Inspector-ban megjelenítve.....	56
4.18. ábra ON/OFF szabályozó - alapjel változtatása.....	56
4.19. ábra ON/OFF szabályozó - alapjel változtatása Data Inspector-ban megjelenítve	57
4.20. ábra ON/OFF szabályozó - energiaátadás	57
4.21. ábra Simulink modell – Szabályozó részrendszer PID szabályozóval	58
4.22. ábra A PID szabályozó részrendszer	58

4.23. ábra „PID Controller” blokk paramétere	59
4.24. ábra PID szabályozó működése	60
4.25. ábra PID szabályozó - megjelenítés Data Inspector-ban megjelenítve	60
4.26. ábra PID szabályozó output korlátokkal - megjelenítés Data Inspector-ban megjelenítve	61
4.27. ábra PID szabályozó - alapjel változtatása	61
4.28. ábra PID szabályozó - alapjel változtatása Data Inspector-ban megjelenítve	62
4.29. ábra PID szabályozó - energiaátadás	62
4.30. ábra: „T_eltérés” fuzzy halmazai	63
4.31. ábra T_ változás fuzzy halmazai	65
4.32. ábra „Fűtés” fuzzy halmazai	66
4.33. ábra Fuzzy szabályok	68
4.34. ábra Következtetés fuzzy szabályokkal – 1. eset	69
4.35. ábra Következtetés fuzzy szabályokkal – 2. eset	69
4.36. ábra Következtetés fuzzy szabályokkal – 3. eset	70
4.37. ábra Felület 1.	71
4.38. ábra Felület 2.	71
4.39. ábra Fuzzy rendszer mentése Matlab Workspace-ként	72
4.40. ábra Simulink modell – Szabályozó részrendszer Fuzzy szabályozóval	72
4.41. ábra Fuzzy szabályozó részrendszer	73
4.42. ábra Fuzzy Logic Controller	73
4.43. ábra Fuzzy szabályozó működése	74
4.44. ábra Fuzzy szabályozó - megjelenítés Data Inspector-ban megjelenítve	74
4.45. ábra Fuzzy szabályozó - alapjel változtatása	75
4.46. ábra Fuzzy szabályozó - alapjel változtatása Data Inspector-ban megjelenítve	76
4.47. ábra Fuzzy szabályozó - energiaátadás	76