

Egyedi szimulációs környezet kialakítása modellalapú szabályozások vizsgálatához

Érchegyi Attila , Dr. Rác Ervin

attila.erchegyi@gmail.com , racz.ervin@kvk.uni-obuda.hu

Abstract: Kutatásom azt a célt szolgálja, hogy létrehozzak egy, a valóságot megközelítő szimulációs környezetet a modell alapú szabályozások teszteléséhez. Az előadás során a készülő dolgozatomból kiindulva, meghatározom, hogy a teljesítmény-hőmérséklet transzformáción felül, milyen járulékos fizikai jelenségek befolyásolják a hőmérséklet szabályozás működését. Ezeket matematikai úton közelítem meg, majd az így kapott függvényeket implementálom C# nyelven. Töreksem arra, hogy az eredmény moduláris és paraméterezhető legyen. Ezután, a modulokat egyesével vizsgálat alá veszem, és értékelem a működésüket. Az előadásom végén bemutatok egy komplex környezetet, amelyben megvizsgálom egy modellalapú szabályozás működését.

Keywords: szimuláció; modellalapú szabályozás; hőmérséklet szabályozás; C#

1. Bevezető

Az egyik munkám során az volt a feladatom, hogy egy több különálló helyiségből álló épület légtechnikájának vezérlését alakítsam át. Az első szempont az volt, hogy az aktuális rendszer két különálló részből tevődik össze, amelyeket egy rendszerbe kell összevonnom. A PLC vezérli a légbefűvást, elszívást, és a szűrők eltömődésének figyelését, mellette egy analóg szabályozó pedig a helyiség temperálásáért és légnyomásuk differenciálásáért felel. A másik szempont, hogy az alkalmazott technológiák miatt, nincs lehetőség a helyi kísérletezgetésre, hanem egy előre jól beállított, és bizonyított működésű rendszert kell hadrendbe állítanom. Ezért ezt a feladatot a fizikai jelenségek leírásával közelítettem meg, és arra építettem föl a szabályozásokat. A teszteléshez valamilyen modellre van szükségem. Az arányosan kicsinyített modell megépítését elvettettem, mert nem biztos, hogy a működése megegyezne a valósággal. Ezért maradt az a megoldás, hogy egy számítógépes modellt alkotok meg, amely a lehető legnagyobb pontossággal tükrözi a valóságot. A következőkben erről fogok beszélni.

A fejezetekben említett programkódok hosszára tekintettel, nem a cikkbe illeszttem be azokat, hanem külön, a githubon teszem elérhetővé őket. Az elérési lehetőségre minden esetben hivatkozom.

2. Fűtőtéljesítmények időbeli felfutása

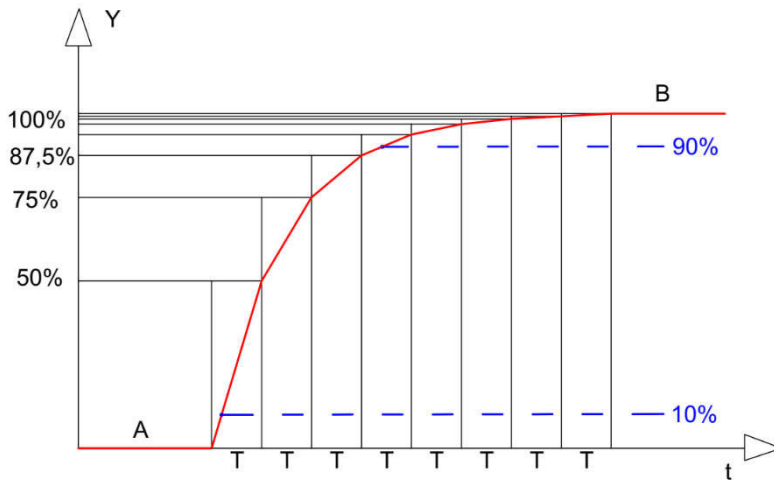
Bekapcsoláskor a fűtőtéljesítmények által leadott teljesítmény csak fokozatosan éri el a névlegest. Ennek okát az atom-, és rácsszerkezetben kell keresni. Vegyünk egy pár atomból álló rácsszerkezetet, amitegységnyi energiával gerjesztünk. Az első atom ezt mind elnyeli, és a többi atomhoz képest jobban elkezd rezegni. Ezzel instabil állapotba, gerjesztetté válik. Mivel ez a gerjesztett állapot nem tartható fenn, ezért az atom a többlet energiát, illetve annak egy részét kibocsátja, átadja a szomszédos atomoknak. Így ha nem is lesznek azonos állapotban, de csökken a közöttük lévő energia különbség. De azok is tovább, tovább és tovább fogják adni, amíg a rácsszerkezet javarészt minden atomja egy stabil, de magasabb energiaszintű állapotba nem kerül. Belátható, hogy egy ilyen energialöket eloszlása nem egy pillanat alatt zajlik le.

Ha tovább gondolkodjuk a jelenséget, amely teljesítmény felfutásnál fellép, akkor az is elmondható, hogy a felfutás meredeksége időben változó, mert a belső energia növekedésével, nő a fűtőművet alkotó fűtőtéljesítmények belső ellenállása is. Az ellenállás növekedésével a töltések áramlása így lassul. Vagyis csökken az a gerjesztés, ami a rácsszerkezet energiájának növekedését okozza. Ez a folyamat mindaddig lassul, amíg a stabil állapotot el nem érjük. Ebben az állapotban a betáplált energia már csak arra elegendő, hogy pótoljuk azt a veszteséget, amelyet az áramló levegő vesz el.

Ezek alapján nézzük meg, hogy a fenti jelenséget miként tudjuk a szimulációban létrehozni. Ehhez térjünk vissza az ógörögökhöz, akiknek a matematikában is nagyon sokat köszönhetünk. Élt közöttük egy Zénón nevű filozófus, akihez 8 paradoxont köthetünk. Az egyikben Achilleusz egy teknőssel versenyez. Mivel Achilleusz feltételezte, hogy gyorsabb lesz a teknősnél, ezért adott neki száz láb előnyt. Mire utána indult, a teknős már tovább ment. Így a száz láb után, kevesebb, de még mindig valamennyi hátrányban maradt. Minden ilyen „ugrásával”, lefaragott ebből, de a paradoxon szerint sose érte utol a teknőst. Ha feltételezzük, hogy a közöttük lévő távolság mindig feleződik, akkor kapunk egy matematikailag jól leírható, és programkódként jól implementálható függvényt, amely alkalmas a felfutás szimulálására.

Most kapcsoljuk ki a fűtőművet, és nézzük meg, hogy ez esetben mi történik. A rendszer stabil állapotában az áramló közeg közel annyi energiát vont el a fűtéstől, mint amennyit villamosan betápláltunk. Amikor kikapcsoljuk a betáplálást, akkor ez az utánpótlás megszűnik, és elkezd csökkenni a fűtőmű remanens energiája. Ahogy a névleges teljesítményt, úgy a bekapcsolás előtti állapotot se tudjuk elérni pillanatszerűen. Ennek vizsgálatához ismét vissza kell térnünk a korábban vázolt elmülethez. Az áramló közeg a fűtőtéljesítmények felületéről, azaz a rác szélső atomjairól vesz el energiát. Ahogy a felfűtésnél, itt is a rácspontokban lévő atomok egymásnak fogják tovább és tovább adni az energiájuk egy részét, amivel ismét stabil állapotba kerülnek. Ez a transzfer nem történhet meg pillanatszerűen, és ez

okozza majd a pillanatnyi leadott teljesítmény időben elnyújtott lefutását. Ahogy csökken a teljes fűtőmű energiája, azzal együtt csökken a közte, és az áramló közeg közötti energiakülönbség is. Így elmondható, hogy egyre kisebb és kisebb energiadagokat kell az atomoknak egymásközt átadniuk. Ezáltal lassul a fűtőmű visszahűlése, a lefutáskor a felfutás tükörképét kapjuk. Azaz a fűtőmű kezdetben meredeken veszít az energiájából, majd egyre lassabban éri el a kezdeti, vagy nevezhetjük úgyis, hogy nyugalmi állapotát. A szimulációban nem csak a fűtési, hanem a visszahűlési folyamatot is meg kell valósítanunk. Mellette azt is figyelembe kell venni, hogy a felfűtéskor nemcsak betáplálásunk volt, hanem az áramló közeg miatt veszteségünk is, amíg hűtéskor, csak az utóbbi. Ezért a két folyamat lefutása időben különbözik, a visszahűlés gyorsabban történik. Ez a felfutásnál nagyobb, a lefutásnál pedig kisebb érték lesz.



1. ábra: Felfutási meredekségek az egyes T szakaszokon

A felfutás szimulálásához először határozzunk meg egy egységnyi időt, amely legyen T! Az első T idő alatt fut fel a függvényünk a végérték feléig, és utána minden egységnyi idő alatt a fennmaradó különbség felével nő, vagyis a végértéket folyamatosan közelítjük, de sohase érjük el. Ez felírható úgy, hogy:

$$B-A \approx \frac{B-A}{2} + \frac{B-A}{4} + \frac{B-A}{8} + \frac{B-A}{16} + \dots + \frac{B-A}{2^n} \quad (1)$$

$$B-A \approx \sum_{i=1}^n \frac{B-A}{2^i} \text{ ha } i \in \mathbb{N}^+ \quad (2)$$

Az A és B különbségének felosztását a végtelenségig folytathatnánk, ezért szabjunk egy ésszerű határt annak, hogy meddig végezzük el a valós matematikai sor összegzését. Ezt úgy tegyük, hogy eressünk egy olyan pontot, amely a T egész számú többszöröse, és a kiinduló (A) és végérték (B) különbségének 1%-át éppen nem haladja meg. Ez a feltétel a szimulációnak már egy elfogadható pontosságot ad.

1. Táblázat

A felfutás meredeksége az egyes T szakaszokon a végértékhez viszonyítva

Időegység	1T	2T	3T	4T	5T	6T	7T	8T
ΔY	50%	25%	12,5%	6,25%	3,125%	1,5625%	0,78125%	0,78125%
Y	50%	75%	87,5%	93,75%	96,875%	98,4375%	99,21875%	100%

A fenti táblázat alapján kimondható, hogy a sor összegzésének 7. lépésben elérünk egy olyan pontot, ahol a pontosságunk 1%-nál már éppen jobb. Ezen felül, a táblázatban felvettem egy 8T időpillanatot is. Ennek oka, hogy a szimulációban szeretnék lehetőséget adni arra is, hogy a valósággal ellentétben a fűtőteljesítmény elérhesse a névleges értéket. A későbbiekben úgy fogom kialakítani a rutint, hogy bemeneti paraméterként meghatározható legyen, hogy a felfutás elérheti-e vagy sem a végértéket. Vagyis, hogy a felfutás 7T vagy 8T ideig tartson.

Ahhoz, hogy a felfutást implementálni tudjam a programba, kell egy bemenő paraméter, amely meghatározza a felfutás hosszát. Erre két lehetőség van. Az első, hogy megadom a 8T illetve 7T idő pontos értékét, azaz a beállási időt. A másik, hogy a felfutási időt [1] adom meg, amelyet gyakrabban használnak az egyes szakirodalmakban. Ezt az időt a következőképpen definiálják: A felfutási idő, a függvény felfutási szakaszán felvett két pont között eltelt idő. Általában ezt a két pontot a végérték 10% és 90%-nál határozzák meg. A továbbiakban én is ennél a meghatározásnál maradok.

A következő lépésben összefüggést kell teremtenünk a felfutási idő, és a korábban bevezetett egységnyi idő között. Amennyiben visszatérünk az 1. táblázathoz, akkor megállapíthatjuk, hogy a névleges 10%-ához tartozó pillanatérték a függvényen 0T és 1T, míg a 90%-ához tartozó 3T és 4T között található. Ezek különböző, de ismert meredekségű szakaszok, ezért a bekapcsolástól (0T) eltelt időt a függvény mindkét keresett pontjára könnyedén meghatározhatjuk:

$$t_{10\%} = \frac{T}{Y_1 - Y_0} * Y_{10\%} = \frac{T}{50 - 0} * 10 = \frac{1}{5} * T \quad (3)$$

$$t_{90\%} = 3 * T + \frac{T}{Y_4 - Y_3} * (Y_{90\%} - Y_3) = 3 * T + \frac{T}{93,75 - 87,5} * (90 - 87,5) = \frac{17}{5} * T \quad (4)$$

Ezek különbségeként megkapjuk a felfutási időt:

$$t_{fel} = t_{90\%} - t_{10\%} = \frac{17}{5} * T - \frac{1}{5} * T = \frac{16}{5} * T = 3,2 * T \quad (5)$$

Mivel az egységnyi időt keressük, és a felfutási időt határozzuk meg paraméterként, ezért az egyenletet T-re átrendezve megkapjuk, hogy:

$$T = 0,3125 * t_{fel} \quad (6)$$

Most térjünk át az elméletre a program implementálására. Először is létrehoztam egy „Felfutas” nevű class-t [2]. Felvettem lokális változókat, amelyekben az

inicializálási paramétereket, és átmeneti értékeket fogom tárolni. Létrehoztam a példányosításkor lefutó függvényt (public Felfutas(...)), amelynek a bemenő paraméterei definiálják a fel-, és lefutást. A függvénytörzsben található a paraméterek eltárolása, és szükség esetén, az egységnyi idők kiszámítása.

A Felfutas class következő része, a „work” függvény. Itt bemenetként meg kell adni, hogy be van-e kapcsolva a fűtőmű, vagy sem. A függvény lefutásakor első lépésben megvizsgálja, hogy a bemeneti adat értéke megváltozott-e az előző állapothoz képest. Mivel a fűtőt teljesítmények vagy be-, vagy kikapcsolt állapotban vannak, ezért a bemeneti bool változó kellőképpen definiálja azok állapotát. Több fűtőmű esetén többször kell példányosítani majd a Felfutas classst.

A „work” függvény további részében megvizsgálja T_tar értékét, vagyis, hogy a fel- vagy lefutás melyik meredekségű szakaszán tart a program. Ez alapján kiszámolja az aktuális kimeneti értéket a következő egyenlet alapján:

$$kimenet = korábbi\ érték \pm maximális\ érték * meredekség * (főciklusidő / T\ időalap) \quad (7)$$

Vagyis a kimeneti érték megkapható, ha a korábbi kimeneti értékhez hozzáadjuk az adott szakaszra számított legkisebb változást. Ez a változás a névleges teljesítmény, az adott T egységidejére vonatkozó meredekség, és a főciklusidő és időalap hányadosának szorzatából számítható ki. A kimeneti értéket mind két irányba korlátozzuk, hogy ne lépje túl az adott szakaszra vonatkozó határértékeket.

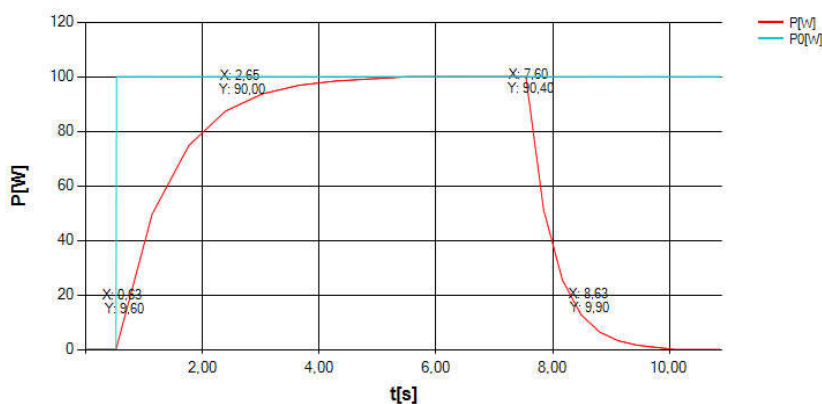
Ezután a függvény eltárolja a jelenlegi időt és ez alapján, ha szükséges, átugrik a következő szakaszra, a végén pedig visszatér a kiszámított teljesítmény értékkel.

Térjünk át a program tesztelésére. Inicializáláskor a következő paraméterekkel példányosítottam a classst:

$$fel = new Felfutas(100, 8, -1.0f, -1.0f, 2, 1, 10); \quad (8)$$

Vagyis a fűtés névleges teljesítményét 100W-ban határoztam meg. A szakaszok száma 8, azaz a névleges teljesítményt el fogja érni a kimeneti függvény. T időegységeket nem, de a fel- és lefutásokat 2s-ban és 1s-ban határoztam meg, illetve megadtam a főciklusidőt milliszekundumban.

A főciklust egy 10ms-os timer megszakításaként létrehoztam. Ebbe helyeztem el a „fel” -ként példányosított Felfutas class „work” függvényét. Majd a kimeneti értéket egy chart-ban ábrázoltam a névleges teljesítménnyel párhuzamosan. Eredményként a következőt kaptam:



2. ábra: A Felfutas class work függvény kimeneteként kapott le és felfutási függvény

Az látható, hogy a fűtőteljesítmény bekapcsolásakor először meredeken nő a pillanatnyi teljesítmény értéke. Ahogy telik az idő telítésbe megy, rásimul a névleges teljesítményt szimbolizáló vízszintes egyenesre. A felfutási szakaszon jelöltem a felfutási idő definíciójához tartozó két pontot. Az egyiket a 10%-os, másikat az 90%-os érték körül. A két pont között eltelt idő, vagyis ΔX megegyezik az inicializáláskor megadott 2 másodperccel.

Nézzük a lefutást! Itt a teljesítmény kikapcsolása után meredeken el kezd csökkenni a pillanatnyi teljesítmény. Majd az eredeti állapot megközelítve a meredekség csökken, és lassan rásimul az X tengelyre. Ahogy a felfutási szakaszon, itt is jelöltem az ominózus két pontot. Ez esetben is elmondható, hogy a két pont X koordinátájának különbsége megegyezik a megadott lefutási idővel.

A kis eltérést mindkét esetben az okozza, hogy a főciklusidő, és az egységnyi idők között nincs teljesen 1 nagyságrendnyi különbség. Ám ennek ellenére elmondható, hogy a várt működést elértem valamint az, hogy a létrehozott classt a továbbiakban tudom használni a fel- és lefutás szimulálására.

3. Légcsatorna késleltetés

A fűtőműveken áthaladva a levegő, légcsatornákon keresztül jut el a fűteni kívánt helyiségig. Ennek hatására a rendszerben egy késleltetés jelenik meg. Maga a késleltetés az az idő, amely alatt a fűtőműtől eljut a levegő a helyiségig. Egy szakasz okozta késleltetés meghatározható a légcsatorna hosszának és az áramló levegő közepes sebességének hányadosából. Ez akkor, és csak akkor igaz, ha ideális gázt feltételezünk, amely állandó keresztmetszetű légcsatornában halad, és a vizsgált szakasz forrás- és nyelőmentes. Mi történik akkor, ha ebből egy vagy több állítás nem igaz? Induljunk ki a kontinuitási egyenletből, ami kimondja, hogy: „... az

állandó sűrűségű áramló közeg sebességének arányát a csatorna keresztmetszetek aránya szabja meg, [3].

$$A_1 \cdot v_1 = A_2 \cdot v_2 \quad (9)$$

ahol:

- A_1 és A_2 , a cső tetszőleges két pontjának keresztmetszete [m^2]

- v_1 és v_2 , a kiválasztott két keresztmetszethez tartozó közeg áramlási sebesség [$\frac{m}{s}$]

Ez felírható az A_1 és A_2 keresztmetszeteken egységnyi idő alatt áthaladó légmennyiségekkel is:

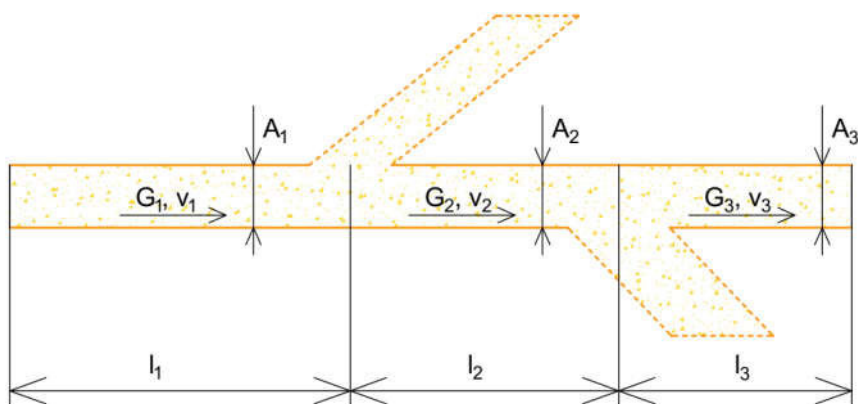
$$G = A \cdot v \quad (10)$$

$$G_1 = G_2 \quad (11)$$

ahol:

- G_1 és G_2 , a cső két tetszőleges keresztmetszetén áthaladó légmennyiség [$\frac{m^3}{s}$] vagy [$\frac{m^3}{h}$]

Egy vizsgált rendszer esetén akkor nem beszélhetünk nyelőmentességről, ha több különálló helyiség temperálását egyazon fűtő- és hűtőmű látja el. Ekkor első lépésben redukálnunk kell az egységet csak azokra a légcsatornákra, amelyek a beavatkozó és a hőmérsékletérzékelő között találhatók. Az így kapott redukált rendszert fel kell osztanunk kisebb szakaszokra, amelyeknek a kezdetét és a végét az elágazások, vagy a rendszer két végpontja határozza. A felosztás után kapott szakaszokra a szállított légmennyiségek és közepes sebességek leolvashatók a gépészeti tervekről. A gépészeti tervezés folyamán első lépésként meghatározzák a helyiségek térfogatából és a szükséges légcsereszámából a befűvott levegő mennyiségét. A helyiségektől visszafele haladva az elágazásokban szállított légmennyiségeket összeadják, ebből megkapják az előző szakaszban szállított levegő mennyiségét. Az összevonásokat mindaddig elvégzik, amíg már nem marad több elágazás. Ez a mennyiség fogja meghatározni a befűvőventillátor teljesítményét.



3. ábra: A légesatorna felosztása és paraméterei

Térjünk vissza a felvetett problémánkhoz. A fenti ábrán található egy légesatorna, amelyet három részre osztottam a korábban leírt kívánalmak alapján. Tegyük fel, hogy a gépészek mindegyik szakaszra megadták a szükséges paramétereket. Ezek alapján a következő képletekkel határozhatjuk meg a szakaszok késleltetési idejét:

A szállított mennyiségből:

$$G = A * v \quad (12)$$

$$G = A * \frac{l}{t} \quad (13)$$

$$t = \frac{A * l}{G} \quad (14)$$

ahol:

- G, a csőszakasz által egységnyi idő alatt szállított légmennyiség $\left[\frac{m^3}{s}\right]$
- A, a csőszakasz keresztmetszete $[m^2]$
- v, a csőszakaszban áramló levegő közepes sebessége $\left[\frac{m}{s}\right]$
- l, a csőszakasz hossza $[m]$
- t, a csőszakasz késleltetése $[s]$

Az áramlás közepes sebességéből:

$$v = \frac{l}{t} \quad (15)$$

$$t = \frac{l}{v} \quad (16)$$

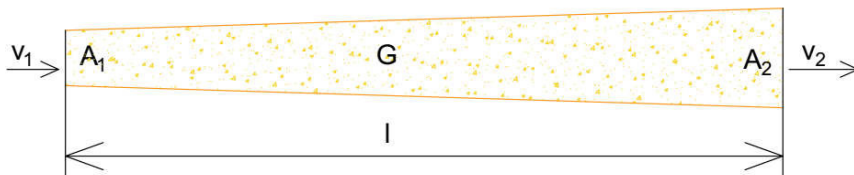
ahol:

- v , a csőszakaszban áramló levegő közepes sebessége $\left[\frac{m}{s}\right]$
- l , a csőszakasz hossza $[m]$
- t , a csőszakasz késletetése $[s]$

A nem forrásmentes esetet, pontosabban azt, ha egynél több forrásunk van, nem érdemes vizsgálni. Elméletben lehet ilyen, de a valóságban nem alkalmaznak több egymástól független fűtőművet, amelyek valahol a szállítási hely felé félúton egymásba csatlakoznának. Ehhez hasonló helyzet lehet, amikor:

- az elszívott levegő egy részét visszakeverik a befűvotthoz. Ez a fűtőművek előtt történik meg. Komplex modell esetén érdemes figyelembe venni. Főleg, ha nincs a keverékre vonatkozó mért adatunk.
- tartalék ventillátorok esetén, amikor is az épület légtechnikájára vonatkozó előírások megkövetelik, hogy egyszeres hiba esetén is meglegyen a szükséges légbefűvás. Itt főleg az átkapcsolási tranziensek okozhatnak problémát.

Amit érdemes megvizsgálni, az a változó keresztmetszet. Az elágazások mellett, lehetnek olyan csőszakaszok, amelyeknek a keresztmetszete egy adott hosszban megváltozik. Ez esetben, a korábbi képlet már nem alkalmazható. Nézzük, miként kell eljárni ekkor:



4. ábra: Változó keresztmetszetű légesatorna

A fenti ábrán egy olyan csőszakasz látható, ahol a levegő egy kisebb, A_1 keresztmetszeten lép be, és egy nagyobb, A_2 keresztmetszeten lép ki. A kontinuitási törvény alapján elmondható, hogy kilépéskor a levegő áramlása lassabb lesz, mint belépéskor. Azt az időintervallumot keressük, ami alatt a levegő áthalad a csőszakaszon. Ehhez induljunk ki először az áramlási sebességből. Első lépésként felírható az átlagsebesség képlete:

$$v = \frac{l}{t} \quad (17)$$

Ebből egy dolog ismert, a cső hossza. A t időt keressük, és az átlagsebességet kellene megadnunk. Mivel ismerjük a két keresztmetszetet, valamint a belépő levegő áramlási sebességét, meg tudjuk határozni kilépéskor a levegő sebességét:

$$A_1 \cdot v_1 = A_2 \cdot v_2 \quad (18)$$

ahol:

- A_1 és A_2 , a csőszakasz két végének keresztmetszete
- v_1 és v_2 , a csőszakasz végében mérhető pillanatnyi sebesség

A két pillanatnyi sebességből egyetlen egy esetben lehet az átlagsebességet meghatározni, ha a cső keresztmetszete egyenletesen változik. Ekkor a két pillanatnyi sebesség számtani közepéből kaphatjuk meg.

Általános esetet vizsgálva, már nagyon apró darabokra kell felosztanunk a csőszakaszt. Ekkor a ki és belépő oldalon a keresztmetszetet azonosnak lehet tekinteni. Minden ilyen csőszakaszban átáramló levegő pillanatnyi sebessége felírható a következő módon:

$$v_{pill} = \frac{\delta l}{t_{\delta l}} \quad (19)$$

ahol:

- v_{pill} , a csőszakaszban áramló levegő pillanatnyi sebessége $\left[\frac{m}{s}\right]$
- δl , az egységnyi csőszakasz hossza $[m]$
- $t_{\delta l}$, az egységnyi csőszakasz késleltetése $[s]$

Az egységnyi kis csőszakaszainkra felírhatjuk a pillanatnyi sebességet a szállított mennyiség, és a keresztmetszet alapján, ahogy korábban az állandó keresztmetszetű cső esetén is megtettük:

$$v_{pill} = \frac{G}{A_{\delta l}} \quad (20)$$

ahol:

- $A_{\delta l}$, az egységnyi csőszakasz keresztmetszete $[m^2]$

A két képletet vonjuk össze, és rendezzük át az egyik oldalát a késleltetési időre:

$$\frac{G}{A_{\delta l}} = \frac{\delta l}{t_{\delta l}} \quad (21)$$

$$t_{\delta l} = \frac{\delta l * A_{\delta l}}{G} \quad (22)$$

Az alapján, amit kaptunk, két dolgot állapíthatunk meg. Az egyik a nevezőben található G , azaz a szállítási teljesítmény. A kontinuitási törvényből kiindulva ennek az értéke a csővezeték keresztmetszetétől független. Vagyis ez állandó. A másik, hogy a számlálóban az egységnyi csőhossz, és annak a keresztmetszetének szorzata egy térfogatot ad. Ez a térfogat az egységnyi csőhossz térfogata. Ezért kifejezhetjük a késleltetést a térfogattal is:

$$t_{\delta l} = \frac{V_{\delta l}}{G} \quad (23)$$

Most, hogy előbb felbontottuk a csövet állandó keresztmetszetű kis szakaszokra, majd meghatároztuk mindegyiknek a késleltetését, ideje újra összerakni, és megnézni mit kapunk:

$$t = \sum t_{\delta l} = \sum \frac{V_{\delta l}}{G} = \frac{\sum V_{\delta l}}{G} = \frac{V}{G} \quad (24)$$

Azt állítjuk, hogy véges, de nagyszámú állandó keresztmetszetű csőszakaszt hoztunk létre. Elméletben minden egyes darabnak tudjuk a térfogatát. Ha ezeket összerakjuk, akkor ismét az eredeti csőszakaszhoz jutunk. Így lényegében azt kaptuk, hogy a késleltetés kiszámítható a csőszakasz térfogata és a szállítási teljesítmény hányadosából, ahol a szállítási teljesítmény adott, valamint a térfogat ismert vagy számítható. Elmondható, hogy ez a képlet igaz lesz nem csak a változó keresztmetszetre, hanem az állandóra is, vagy akár a kanyaródó csőszakaszokra is egyaránt. Felmerül a kérdés, hogy lehetne-e az egész csövet egyben, a fűtőműtől a befűvásig egyetlen képlettel leírni? A válasz: nem! Pontosabban, nagyon kivételes esetben. Ugyanis, általában nem egy helyiség fűtését szolgálja ki egy-egy fűtőmű, hanem többét. Ezért, a teljes légcsatorna hosszát tekintve, az elején nagyobb szállítási mennyiségről beszélhetünk, mint a kilépő oldalon, ahol a hőmérséklet-mérésünk van.

Térjünk át a megvalósításra. Létrehoztam egy Csatorna nevű class-t [4], és benne két lokális változót. Az egyiket az idő, a másikat a bemeneti adatok tárolására. A class első függvénye, amely példányosításkor fut le, a bemeneti paraméterek alapján kiszámolja a szükséges késleltetést. Három féle módon teheti meg az alapján, hogy az első paraméternek milyen értéket adunk. Az első az állandó keresztmetszetű légcsatorna a szállított mennyiség ismeretében. A második az állandó keresztmetszetű légcsatorna a légáramlás sebessége ismeretében. A harmadik az általános, amivel a változó keresztmetszetű légcsatorna írható fel a térfogata és a szállított mennyiség ismeretében. A kiválasztott meghatározási módból kapott időtag alapján, ezután inicializálja az adat_tar változótömböt.

A ciklikusan meghívható shift függvény felelős az adat_tar tömb kezeléséért. Első lépésben eltolja a tárolt értékeket egyel, a tömb utolsó helyére pedig eltárolja a bemenő adatot. Majd, a műveletei befejeztével, visszatér a tömb első elemének értékével.

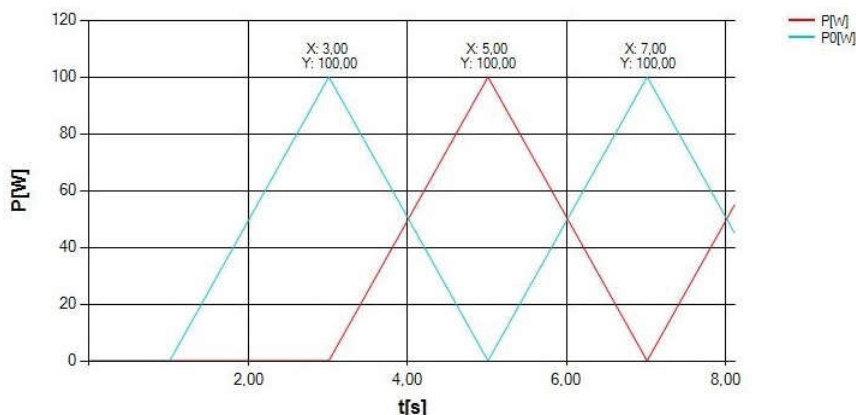
A működés teszteléséhez létrehoztam egy keretprogramot. Ennek az inicializáló szakaszában példányosítok egy Csatorna class-t, lcs néven, a következő paraméterekkel:

$$\text{lcs} = \text{new Csatorna}(2, -1, 4, -1, 2, -1, 10); \quad (25)$$

Itt megadom a 2-es módot, azaz egy állandó keresztmetszetű csatornát írok le a csatorna hosszával (4 m), és légáramlás sebességével (2 m/s). Ezáltal létrehoztam egy 201 elemű adattároló tömböt.

A keretprogram ciklikus része egy 10ms-os időzítő megszakítására fut le. Ebben hívom meg a lcs.shift() függvényt az aktuális bemenő értékkel. Az eredeti

teljesítmény értéket (P_0), valamint az lcs classunk kimenetét (P) egy chart-ba rögzíti minden ciklusban. Valamint a végén, létrehozza a P_0 változását, hogy a kapott ábra szemléletes legyen. Így a következő ábrát kaptam:



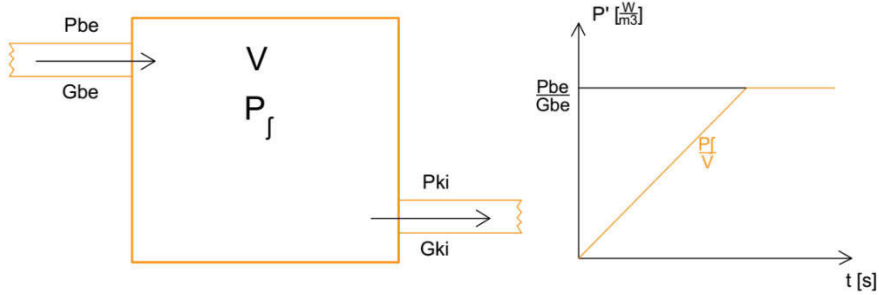
5. ábra: A Csatorna class work függvény kimenete a bemenő jel függvényében

Ezen kék színnel láthatjuk az eredeti teljesítmény értéket, ami a shift függvény bemenő értéke. Valamint piros színnel, a shift függvény kimenő értékét, ami már a cső késleltetésével töltünk el az időben. Mindegyik csúcs jelölve van, amit 100W-on érnek el. A két csúcs között pontosan 2 másodperc van. Ez megfelel annak a számított késleltetési időnek, ami a csatorna hosszának és a légáramlás sebességének hányadosa ad. Ezáltal kimondható, hogy az elméletben felvázolt, majd megvalósított függvény megfelel a várt működésnek, a továbbiakban használható a légszűrő késleltetésének szimulálására.

4. A helyiség integráló hatása

A modellezés utolsó lépéseként, nézzük meg, hogy mi tapasztalható akkor, amikor a légszűrőből kilép a fűtött levegő a helyiségbe. Első lépésben határozzuk meg milyen paramétereink vannak. Adott a helyiség térfogata (V). Erre határozzák meg a befűtött levegő mennyiségét (G_{be}), a szükséges légcsereszám alapján. A befűtött levegőnek emellett többlet energiája van, ez a fűtőteljesítmény (P_{be}), amellyel a külső levegő hőmérsékletéhez képest rendelkezik. A helyiségből minden esetben távozik a terem levegőjének egy része. Ez történhet aktív módon, elszívó légszűrővel, vagy passzív módon, szivárgással a nyílászárókon keresztül. A kiáramló levegő térfogatát az határozza meg, hogy a helyiségre milyen nyomás értéket állapítanak meg. Lehet légköri, ekkor a kiáramló levegő térfogata megegyezik a beáramlóéval. Ilyen például egy átlagos irodahelyiség. Pormentes helyiség esetén túlnyomásos, ahol az elszívott levegő mennyisége kisebb, mint a

befűvotté. Illetve alulnyomásos, ahonnan a pormentes helyiségbe lépünk be. Ez utóbbinál nagyobb térfogatú levegőt kell kiszállítani a térből, mint amennyit bejuttattunk. A távozó levegő, ahogy a befűvott levegő, két paraméterrel rendelkezik: a szállított légmennyiséggel (G_{ki}), és a fűtőtéljesítményével (P_{ki}).



6. ábra: A helyiséget szimuláló paraméterek és pillanatnyi értékük alakulása

Paraméterek meghatározása után nézzük meg, mi történik. Egységnyi idő alatt belép a helyiségbe G_{be}/t mennyiségű levegő, ami P_{be} fűtőtéljesítményt hordoz magában. Ez a helyiség szempontjából azt jelenti, hogy a levegőjének belső energiája megnő. Azért, hogy a korábbiakhoz hasonlóan majd a helyiséget szimuláló programblokk kimenete teljesítmény legyen, a belső energia helyett, többlet hőteljesítményt fogok kifejezni. Így azt írhatjuk fel, hogy:

$$P_{fn} = P_{fn-1} + P_{be} \quad (26)$$

$$P_f = \int P_{be} \quad (27)$$

A helyiség hőteljesítményét megkapjuk a beáramló levegő fűtőtéljesítményének integráljaként. Hozzá kell tenni azt, hogy az eredményként kapott függvény korlátos kell, hogy legyen. Az könnyen belátható, ha 28°C -os levegőt fűjünk be egy helyiségbe, akkor annak a hőmérséklete nem haladhatja meg a 28°C -ot, vagyis a befűvott levegő hőmérsékletét egyéb forrás nélkül. Befűváskor két különböző hőmérsékletű közeg találkozik. Az energiájuk igyekszik kiegyenlítődni. A befűvott levegő mennyisége végtelennek tekinthető a helyiség levegőjéhez képest, a folyamatos fűtés és befűvás miatt, ezért a két közeg energiájának kiegyenlítésénél az tapasztalható, hogy a helyiség levegőjének energiája konvergál a befűvottéhoz.

A befűvás mellett, viszont elszívás is történik. Erről elmondható, hogy a meglévő hőteljesítmény egy részét elveszítjük. Ezt a mennyiségét a helyiség, és az elszívott levegő térfogatának hányadosa fogja meghatározni.

$$P_{ki} = P_{fn} \frac{G_{ki} \cdot t}{V} \quad (28)$$

Ha a fenti két képletet összevonom, akkor megkapom egy adott időpillanatban a helyiség többlet hőteljesítményének leírását:

$$P_{jn} = P_{jn-1} + P_{be}(t) - P_{jn-1} \frac{G_{ki} * t}{V} \quad (29)$$

$$P_{jn} = P_{jn-1} \left(1 - \frac{G_{ki} * t}{V}\right) + P_{be}(t) \quad (30)$$

Mivel a P_{be} értékét nem tudjuk, csak a fűtőművek pillanatnyi teljesítményét (P), ezért a képletet az ismert adatok ismeretében kell módosítanunk:

$$P_{be} = P * \frac{G_{be}}{G} \quad (31)$$

$$P_{jn} = P_{jn-1} + P * \frac{G_{be}}{G} - P_{jn-1} \frac{G_{ki} * t}{V} \quad (32)$$

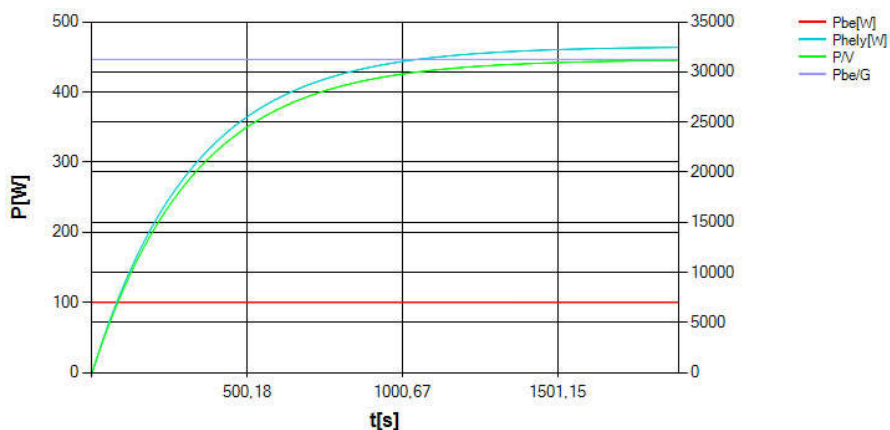
$$P_{jn} = P_{jn-1} \left(1 - \frac{G_{ki} * t}{V}\right) + P * \frac{G_{be}}{G} \quad (33)$$

A képletek ismeretében, áttérhetünk a megvalósításra. Létrehoztam egy *Helyiseg* nevű classot [5]. Ebben első lépésként, felvettem változókat a paraméterek és az aktuális állapot tárolására. Az inicializáló függvénye bemenő paraméterként várja a következőket: *init_V* – helyiség térfogata (m³), *init_G* – a teljes fűtött légmennyiség (m³/s), *init_Gbe* – helyiségbe befűvott levegő mennyisége (m³/s), *init_Gki* – helyiségből elszívott levegő mennyisége (m³/s), *init_alap* – főciklusidő (ms). Példányosításkor menti a létrehozott class a paramétereket. A ciklikusan meghívható függvény bemenete a pillanatnyi fűtőteliesség értéke a teljes fűtött légmennyiségre vonatkozóan.

Előzőkhöz hasonlóan létrehoztam egy keretprogramot, amelynek inicializáló részében példányosítottam egy *Helyiseg* classot a következő módon:

$$\text{hely} = \text{new Helyiseg}(72.93f, 0.224f, 0.224f, 0.224f, 10); \quad (34)$$

A bemenő paramétereket egy valós eset alapján választottam ki. A vizsgált helyiség térfogata 72.93m³. A befűvott, és elszívott levegő egyaránt 0.224m³/s. Úgy választottam ki a teljes fűtött légmennyiséget, hogy az a beáramló levegő mennyiségével azonos legyen, így a szimulációból kapott eredmény átláthatóbb. A főprogram ciklus ideje 10ms. Ezek alapján a következő eredményre jutottam:



7. ábra: A Helyiség class work függvény kimeneteként kapott pillanatnyi teljesítmény a bemenő teljesítmény függvényében

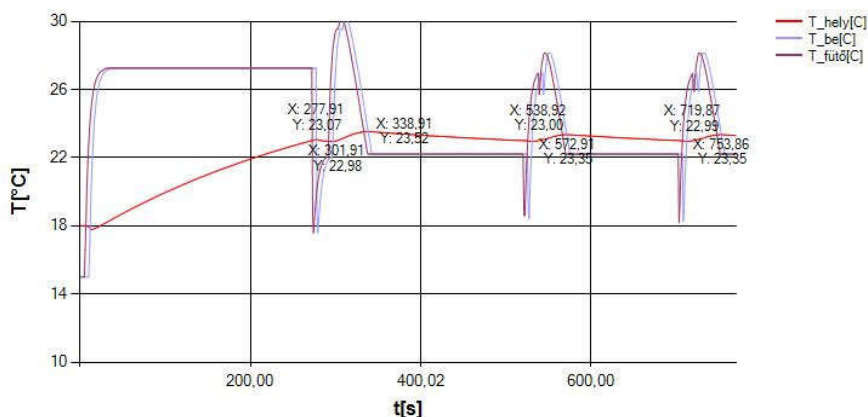
A futás során a beáramló levegő többlet hőteljesítménye 100W. Négy dolgot jelenítettem meg a grafikonon: a beáramló levegő hőteljesítményét (P_{be}), a helyiség hőteljesítményét (P_{hely}), a helyiség egységnyi térfogatára vett hőteljesítményt (P/V) és a befűvott levegő egységnyi térfogatára jutó hőteljesítményt. A kapott grafikonon azt láthatjuk, hogy az állandó fűtés mellett megindul a helyiség hőteljesítménye, vagy hétköznapi kifejezéssel akár azt is mondhatjuk, hogy a hőmérsékletének növekedése. Ez mindaddig folytatódik, amíg el nem éri a P/V görbe a P_{be}/G egyenest. Vagyis, amíg a helyiség egységnyi térfogatának hőteljesítménye el nem éri az befűvott levegő térfogatának egységére jutót. A tapasztalt jelenség, megfelel az elvártnak, vagyis, hogy folyamatosan melegedik addig a helyiség, amíg annak a hőmérséklete el nem éri a befűvott levegő hőmérsékletét.

5. Befejezés

Az elemek megalkotása után, összeilleszttem őket egy valós eset modellezésére. Az épületben, amelyet vizsgállok, olyan fűtőmű található, amely 6 külön kapcsolható fűtőtéljesítményből áll. Ez, és a ventilátor együtt több helyiséget lát el. A fűtött levegő különböző elágazások után éri el mindet. Három különböző hőmérséklet-érzékelőnk van. Az egyik közülük a vizsgált helyiségben található meg. Egy a külső levegő hőmérsékletét, egy pedig a befűvott levegő hőmérsékletét méri. A modellhez példányosítottam 6 Felfutas class-t, egyesével mindegyik fűtőtéljesítményre. A légcsatornákat 9 Csatorna class-val hoztam létre. Majd a végére illesztettem egy Helyiség class-t, amit temperálok. A fűtőműveket egymással párhuzamosan, a többi elemet pedig sorosan kötöttem össze egy átmeneti változóval. A vizsgált

szabályozásom ezektől függetlenül külön függvényben futtattam, és bemeneti paraméternek az aktuális hőmérséklet értékeit adom, kimenetként a Felfutas class-ok kétállapotú bemeneteit kapom meg.

Induljunk ki egy üzemszüneti állapotból. A szabadban 15 °C van. A fűtés nincs üzemben, de a felügyelt szobában 18 °C van. Azt a megkötést kell tennünk az ott tartózkodó személyzet érdekében, hogy a befűvott levegő hőmérséklete nem haladhatja meg a 28 °C-ot. Elindítva a modellt, a következő eredményt kaptam:



8. ábra: Az összeállított szimuláció eredményei a technológiai szempontból kiemelt helyeken

Az ábrán három dolgot ábrázoltam. Pirossal a helyiség hőmérsékletét, késsel a befűvott levegő hőmérsékletét, lilával pedig egy fiktív hőmérsékletet, amely közvetlenül a fűtőműből kilépő levegőé. Ez utóbbira azért volt szükség, hogy bemutassam mekkora az a késleltetés, amíg a forrástól eljut a levegő helyiségig. A grafikon alapján azt láthatjuk, hogy a helyiség integrálásával összehasonlítva ez a holtidő ebben a példában nem számottevő, de jelen van.

Nézzük, hogy alakul a helyiség hőmérséklete! Induláskor még hideg levegőt fúj be a rendszer egy rövid ideig. Ezért a szobában egy rövid ideig csökken a hőmérséklet, de nem számottevően. Fokozatosan fűtjük a helyiséget a megengedett maximális hőmérsékletű levegővel, hogy minél előbb elérje a kívánt hőmérsékletet. Amint a hőmérséklet eléri a 23 °C-ot, átvált a program a finom szabályozásra. Azt a kilengést, amely látható a kék és lila görbén, a fűtőtéljesítmények átváltása okozza. A szoftver ekkor két nagy fűtőtéljesítményt kapcsol át. Amíg az egyik teljesítménye felfut, addig a másiké csökken, de valamennyi hőt továbbra is leadnak a korábban működő fűtőtéljesítmények.

Látható, hogy a későbbiekben szépen beáll a helyiség hőmérséklete. A szabályozás kicsivel több mint 0,3°C-os eltéréssel képes ezt tartani. A kék görbe kilengései is kisebbek, mivel kisebb teljesítményeket kell csak átkapcsolni hőmérséklet tartásához, viszont, mind a három megjelenített hőmérsékleti csúcsérték átlépi a 28 °C-os határt. Ez azt jelenti, hogy egy rövid ideig a megengedettől magasabb

hőmérsékletű levegőt juttatunk be a helyiségbe. Ezért, a szabályozásban szükséges majd az átkapcsolások figyelembevétele. Időben el kell tolni a ki, és bekapcsolásokat egymástól.

Összeségében elmondható, hogy a modell megfelel a cikk elején támasztott elvárásoknak, vagyis moduláris és kellőképpen beállítható a rendelkezésre álló gépészeti adatokkal. Ugyanakkor, tovább is kell fejlesztenem, hogy komplexebb vizsgálatokat is végre tudjak hajtani vele. Az első, hogy a teljesítmény felfutásoknál megoldást kell keresnem, hogy paraméterezhető legyen pusztán a fűtőmű fizikai jellemzőivel is, mivel nem minden esetben áll rendelkezésre adat a felfutási időkről. A másik, hogy olyan, most elhanyagolt jelenségekkel is foglalkoznom kell, mint a hőveszteségek a falakon, hőátadás a csatorna falán, valamint áramlási anomáliák a helyiségekben vagy légcsatornáknakban. A célom, hogy a végén eljussak egy olyan felhasználóbarát eszközig, amiben az építészeti és gépészeti adatok és rajzok alapján egy komplett épületet össze tudok állítani, és a nem kontrollált helyiségek esetén is megtudhatók a hőmérsékletük alakulása.

6. Hivatkozások

[1] Neszveda József, "A minőségi jellemzők az időtartományban", Automatika I., Klasszikus szabályozásméletek, Budapest, 2010, 95.-96. oldal

[2] Felfutás class (2020), Érchehyi Attila, Elérve: 2020.11.20 [Online], Elérés: https://github.com/Mad31nCh3rn0b1l/modell_szimu/blob/main/felfutas

[3] Dr. Fekete Iván, Dr. Menyhárt József, "Ideális gázok áramlástörvényei", A légtechnikai elméleti alapjai, Budapest, 1975, 66. oldal

[4] Csatorna class (2020), Érchehyi Attila, Elérve: 2020.11.20 [Online], Elérés: - https://github.com/Mad31nCh3rn0b1l/modell_szimu/blob/main/csatorna

[5] Helyiség class (2020), Érchehyi Attila, Elérve: 2020.11.20 [Online], Elérés: - https://github.com/Mad31nCh3rn0b1l/modell_szimu/blob/main/helyiseg