

TDK DOLGOZAT

KORSZERŰ SZIMULÁCIÓS MÓDSZEREK, VIRTUÁLIS FÉKPAD- ÉS MOTORMODELLEK A KÖLTSÉGHATÉKONY FELSŐOKTATÁSÉRT

Készítette: Szabó Zsolt

Agócs Tamás

Belső konzulens: Dr. Hanula Barna

Absztrakt

Az elmúlt év során a mérnökképzés gyakorlati oldalának erősítése és a hallgatók versenyképességének növelése érdekében számos fejlesztést hajtottunk végre a BSc és MSc szakos mérnökhallgatók oktatásában egyaránt. Létrehoztunk egy virtuális fékpad laboratóriumot, ahol költséges berendezések nélkül szimulálható egy motor fékpad viselkedése, vezérlésére pedig az iparban egyik legelterjedtebb, modern applikációs feladatokra tervezett fékpad szoftverrel, az AVL CAMEO-val nyílik lehetőség. A motorok fékpad viselkedését jellegmezőkre épülő modellekkel szimuláltuk, melyeket korábbi mérések eredményeiből nyertünk, ezzel a virtuális motorok a valósághoz közeli eredményeket adtak. Az előző TDK sikereit követve sor került az új módszerekre épülő első laborgyakorlatra is, melyet a diákok nagy érdeklődéssel fogadtak.

A Széchenyi István Egyetem Belső Égésű Motorok Tanszékén több modern szimulációs eljárás oktatása folyik párhuzamosan, célunk ezen módszerek integrálása volt a virtuális fékpad továbbfejlesztéséhez. A tanszéken rendelkezésre áll az AVL BOOST szoftver is, mely egy fejlett, teljesen integrált motor-szimulációs eszköz, null- és egydimenziós megoldásokkal. Segítségével olyan motormodellek építhetők fel, melyekben a valóságos motorok összes paramétere definiálható. A szoftver nagy előnye, hogy összeköthető Matlab Simulink-el, melyben a korábbi, jellegmező alapú modelljeink is készültek. A BOOST-al felépített fizikai modellek lehetővé teszik a ciklikus folyamatok valós idejű szimulációját. Az eljárással tetszőleges motorok modelljei hangolhatók össze a virtuális fékpaddal.

Az összeállított virtuális motor- és fékpad labor Magyarországon egyedülálló lehetőséget biztosít a motorikus folyamatok oktatására. Ennek köszönhetően a hallgatók költséges berendezések használata nélkül szerezhetik a velük szemben támasztott egyre magasabb elvárásoknak megfelelő tudást.

Tartalomjegyzék

1.	Bevezetés.....	6
1.1.	Az ötlet	8
1.2.	Szoftverfejlesztés	8
1.3.	Új módszer	9
1.4.	V-modell	10
1.5.	Modell integráció	11
1.6.	A modellek felépítése.....	11
1.7.	Gyakorlati alkalmazások	12
1.8.	Kiválasztás	13
2.	Jellegmezőkre épülő fékpad- és motormodellek	14
2.1.	Feladat	14
2.2.	Célkitűzés.....	14
2.3.	Modellalkotás	14
2.3.1.	Az első modell	15
2.3.2.	Késleltetés szubmodell.....	16
2.3.3.	Rezonátor szubmodell.....	17
2.3.4.	Fogyasztásváltozás szubmodell	18
2.3.5.	NO _x változás szubmodell.....	20
2.3.6.	HC változás szubmodell	22
2.3.7.	CO változás szubmodell	23
2.4.	Az első modellből készült CAMEO mérés	24
2.5.	A második modell	25
2.5.1.	A Csatornaváltó elem.....	26
2.5.2.	Fordulatszám- és terhelésfüggő mérési mód.....	26
2.5.3.	Légviszony függő mérési mód	27
2.5.4.	Fajlagos fogyasztás szubmodell.....	29
2.5.5.	Effektív nyomaték szubmodell	29
2.5.6.	NO _x kibocsátás szubmodell.....	30
2.5.7.	HC kibocsátás szubmodell	30
2.5.8.	CO kibocsátás szubmodell	31

2.5.9.	A motor C-görbéje a légviszony függő mérési mód esetében	31
2.5.10.	A második modellből készült CAMEO mérés	32
3.	Fizikai motormodellek.....	33
3.1.	Feladat	33
3.2.	Célkitűzés.....	33
3.3.	Felhasznált modellek	33
3.3.1.	Egyhengeres kétütemű benzines motormodell	35
3.3.2.	Egyhengeres négyütemű benzines motormodell	36
3.3.3.	Egyhengeres négyütemű motorkerékpár motormodell	36
3.3.4.	Háromhengeres kétütemű diesel motormodell	37
3.3.5.	Négyhengeres ipari felhasználású diesel motormodell	37
3.3.6.	Négyhengeres négyütemű diesel motormodell.....	38
3.3.7.	Négyhengeres négyütemű benzin motormodell	38
3.3.8.	Hathengeres négyütemű diesel motormodell	39
3.3.9.	Hathengeres négyütemű tehergépjármű motormodell	39
3.3.10.	Hathengeres kétütemű diesel motormodell.....	40
3.3.11.	Hathengeres négyütemű benzin motormodell	40
3.4.	Matlab Simulink modellek.....	41
3.4.1.	A szelepemelési görbék megjelenítése.....	44
3.4.2.	A motor lefulladása	45
3.4.3.	A motor kopogása	46
3.5.	A szimuláció futtatása	47
3.6.	A modellek validálása	48
4.	Összefoglalás	50
5.	Kitekintés	51
	Irodalomjegyzék.....	52

Ábrajegyzék

1. ábra: Vezérlő egységek	8
2. ábra: Elektronikai egységek kapcsolata	9
3. ábra: A V-modell	10
4. ábra: A fejlesztés különböző fázisai	11
5. ábra: Modellek.....	13
6. ábra: A Design Space.....	15
7. ábra: Az első modell sematikus ábrája	16
8. ábra: A késleltetés szubmodell sematikus ábrája	17
9. ábra: A késleltetés szubmodell működés közben	17
10. ábra: A rezonátor szubmodell sematikus ábrája	18
11. ábra: A fajlagos fogyasztás szubmodell sematikus ábrája	18
12. ábra: A fajlagos fogyasztás 2D-s ábrája.....	19
13. ábra: A fajlagos fogyasztás 3D-s ábrája.....	20
14. ábra: A NO _x kibocsátás 2D-s ábrája	21
15. ábra: A NO _x kibocsátás 3D-s ábrája	21
16. ábra: A HC kibocsátás 2D-s ábrája	22
17. ábra: A HC kibocsátás 3D-s ábrája	22
18. ábra: A CO kibocsátás 2D-s ábrája	23
19. ábra: A CO kibocsátás 3D-s ábrája	23
20. ábra: A kézzel állítható mérés, működés közben.....	24
21. ábra: Az automatikus mérés pontjai	25
22. ábra: Az automatikus mérésnél kapott adatok.....	25
23. ábra: A fajlagos fogyasztás szubmodelljének sematikus ábrája.....	26
24. ábra: A második mérési mód fogyasztásváltozás szubmodellje.....	27
25. ábra: A második modell sematikus ábrája	28
26. ábra: A fajlagos fogyasztás görbéje.....	29
27. ábra: Az effektív nyomaték görbéje	29
28. ábra: A NO _x kibocsátás görbéje	30
29. ábra: A HC kibocsátás görbéje	30
30. ábra: A CO kibocsátás görbéje	31
31. ábra: A motor C-görbéje.....	31

32. ábra: A második mérés, működés közben	32
33. ábra: Az első mód, automatikus mérés adatai	32
34. ábra: Aktuátorok egy elkészült modell esetében	34
35. ábra: Szenzorok egy elkészült modell esetében	34
36. ábra: BST fájl létrehozása.....	35
37. ábra: Az első motormodell.....	35
38. ábra: A második motormodell	36
39. ábra: A harmadik motormodell.....	36
40. ábra: A negyedik motormodell	37
41. ábra: Az ötödik motormodell	37
42. ábra: A hatodik motormodell	38
43. ábra: A hetedik motormodell.....	38
44. ábra: A nyolcadik motormodell	39
45. ábra: A kilencedik motormodell	39
46. ábra: A tízedik motormodell	40
47. ábra: A tizenegyedik motormodell.....	40
48. ábra: AVL BOOST blokk Matlab Simulink-ben.....	41
49. ábra: Matlab Simulink főmodell	42
50. ábra: Matlab Simulink szubmodell	43
51. ábra: Szelepelemelési görbe BOOST-ban (fent) és Matlab Simulink-ben (lent).....	44
52. ábra: Eltolás szubmodell.....	45
53. ábra: Lambda kontroll szubmodell.....	45
54. ábra: Rezonátor szubmodell.....	46
55. ábra: AVL BOOST Script	47
56. ábra: Szimuláció kezelői felülete Matlab Simulink-ben	48
57. ábra: Eredmények BOOST-ban	49
58. ábra: Eredmények Matlab-ban	49
59. ábra: A hagyományos (balra) és az új oktatási módszer (jobbra).....	51

1. Bevezetés

Dolgozatunk témája egy hosszú távú projekt, melyet a Széchenyi István Egyetem Belső Égésű Motorok Tanszékével együttműködve fejlesztünk a tavalyi év elejétől kezdődően. Célunk, az utóbbi években átalakult új fejlesztési stratégiák és technológiák megismertetése a járműmérnök szakon tanuló hallgatókkal, valamint motoros ismereteik és gyakorlati készségeik továbbfejlesztése. Ennek megvalósításához számos korszerű eszközt használtunk fel és vezettünk be az oktatásba. Az alapvető ötlet az volt, hogy létrehozzunk egy virtuális motorvizsgáló labort, mellyel képesek vagyunk egy motor fékpadi viselkedését szimulálni, a hallgatók pedig egyenként, önállóan végezhetik el azokat a méréseket, amihez máskülönben egy nagy beruházási igényű fékpadra lenne szükség, magas üzemeltetési költségek mellett. Ez, a szak nagy létszámát figyelembe véve nem is lenne lehetséges, ugyanis ha minden hallgató önálló méréseket végezne a fékpadon, az olyan sok időt venne igénybe, ami nem fér bele a féléves tematikába.

További célunk, hogy a hallgatók találkozzanak a gyakorlatban használt fékpadi mérőszoftverekkel, ismerjék meg az elvégezhető vizsgálatokat és legyenek képesek a mért eredmények kritikus értékelésére. Ehhez a feladathoz először szükségünk volt egy olyan fejlett fékpadvezérlő szoftverre, amely képes a kimeneti, ill. bemeneti csatornáin egy virtuális környezetből érkező jeleket olvasni. Legalkalmasabb alternatívának az AVL List GmbH CAMEO nevű fékpadi szoftvere bizonyult, a motor és fékpad modelleket pedig Matlab Simulink segítségével készítettük el. A szimuláció során ezek helyettesítik a fizikai berendezéseket, így a környezet bármely számítógépen használható, ami rendelkezik az említett programokhoz szükséges licencekkel. A modellek elkészítését részletesen dokumentáltuk, a működőképes rendszert felépítettük és megtartottuk az első gyakorlatot is a hallgatóknak az elmúlt félév végén. Meggyőződésünk, hogy ezzel a mérnökhallgatók az eddigieknél lényegesen behatóbban ismerhették meg a fékpadi vizsgálatokat.

További tervünk az volt, hogy a motorok viselkedéséről is átfogó ismereteket adjunk át hallgatóinknak, melyhez a virtuális fékpadénál pontosabb modelleket szerettünk volna alkotni. A megoldást a matematikai modellek lecserélése jelentette fizikai modellekre, amelyhez új szoftveres környezetre volt szükség. A Belső Égésű Motorok Tanszéken már használatban volt az AVL GmbH BOOST nevű programja, ami speciális beállítások után képes Matlab Simulink-kel kommunikálni. A programot számos hallgató használja gyakorlattan bizonyos motorok elméleti felépítésére, most viszont szerettük volna bevonni ezt a felületet a virtuális fékpadlabor működésébe. Projektünk eredményeként a virtuális fékpad

jelenlegi változata tizenegy különböző típusú belső égésű motor modelljének mérését teszi lehetővé. A BOOST azonban elsősorban motormodellek felépítésére használható, valós idejű szimulációra és fékpadi méréshez hasonló feladat végrehajtására nem alkalmas. Ehhez újból fel kellett építenünk egy MATLAB Simulink modellt, amivel létrehozható a kapcsolat a BOOST-ból, C-kódként exportált modellel. A Simulink környezetet úgy alakítottuk ki, hogy a hallgatók képesek legyenek számos paraméter változtatására. Így megfigyelhetik a különböző motorok viselkedését több üzemállapoton keresztül és képet kaphatnak arról, hogy egy tényleges mérés során milyen nagyságrendű adatok fogadhatók el egy adott motortípustól. Ez utóbbi képesség különösen hasznos a mérnöki gyakorlatban, hiszen napjainkban a mérések nagy többsége számítógépen keresztül történik, ahol bármikor előfordulhatnak a vizsgálat végeredményét befolyásoló körülmények, hibák. A mérnöknek meg kell tudnia állapítani a kapott értékek helyességének valószínűségét, idegen szóval a „kredibilitást”, azaz lehetséges-e adott körülmények között a kapott eredmény, vagy mérési hibáról beszélhetünk.

A következő oldalakon részletesen bemutatjuk a projekt megvalósításához felhasznált szimulációs módszereket, melyek oktatási célra való felhasználása eddig példátlan. Olyan eljárásokat és kutatási módszereket honosítottunk meg a Belső Égésű Motorok Tanszék számítógépes laborjában, amelyekkel a mérnökhallgatók csak a végzés után, az iparban találkozhattak, hazánk területén belül pedig még ott sem.

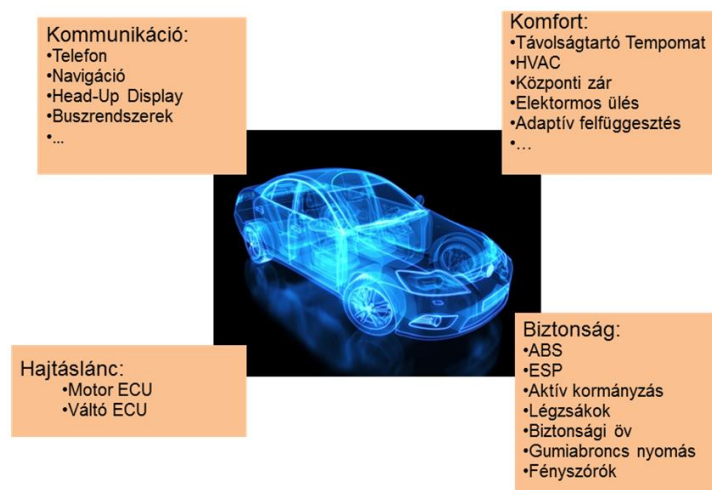
Bár kutatási területünk a belső égésű motorok és feladatunk is ennek, illetve a motorikus vizsgálatoknak a bemutatása és oktatásának korszerűsítése, fontosnak tartjuk a legújabb szimulációs technológiák és fejlesztési módszerek megismertetését is a leendő mérnökökkel. Az eddig megtartott laborgyakorlatok is arra világítottak rá, hogy a hallgatók fokozott érdeklődést mutatnak ezen területek iránt és hasznosnak tartják az ilyen irányú képzést.

1.1. Az ötlet

A virtuális motorfékpad labor megszületéséhez a járműipar utóbbi éveiben ugrásszerű fejlődésnek indult módszer adta az ötletet. Az úgynevezett modell alapú fejlesztést a jármű elektronikai rendszereit vezérlő szoftveres funkciók bővülése hívta életre. Az elmúlt két évtizedben a járművekben megjelenő szoftver alapú kényelmi és biztonsági szolgáltatások aránya 20-ról 80 százalékra emelkedett, ezzel növelve a komplexitást és a hibalehetőségeket is. Ezek minimalizálása hosszú és körültekintő tesztelést igényel. Mind az egyre bonyolultabb szoftverek, mind pedig az ezek tesztelésére irányuló eljárások igen körülményesek, ha C-kód alapú fejlesztőkörnyezettel dolgozunk. Tapasztalatok szerint a hagyományos kézi programozással végzett, és a modell alapú szoftverfejlesztés között 25-35% idő-és pénzbeli megtakarítás érhető el. Ennek az új módszernek az egyik eszközét, a Model-In-The-Loop szimulációt használhatjuk fel arra, hogy egy oktatási célra is alkalmas virtuális környezetet hozzunk létre.

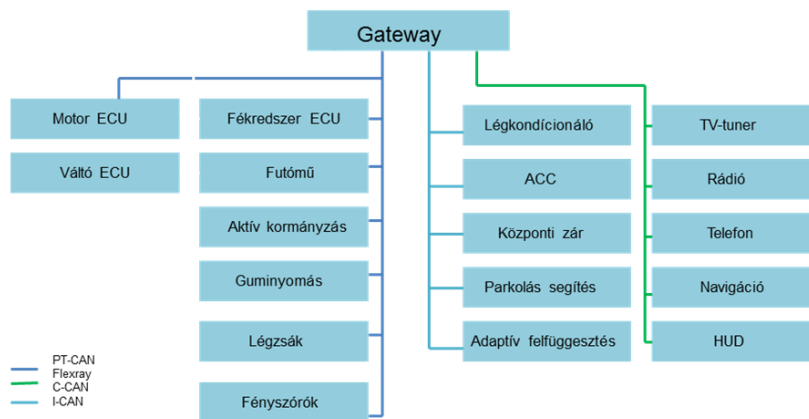
1.2. Szoftverfejlesztés

Napjainkban a járművek szolgáltatásai szinte kizárólag szoftveres irányítással, valamilyen mikrokontrolleren keresztül hozhatók működésbe. A számos funkciót alcsoportokba oszthatjuk, mindegyik saját elektronikai vezérlő egységgel rendelkezik, melyeknek kommunikálniuk kell egymással. Ezt szemlélteti az 1. ábra.



1. ábra: Vezérlő egységek

A különböző alegységeknek saját feladatuk ellátása mellett összehangolva kell működniük a teljes jármű elektronikai rendszerével, ez többféle buszrendszer szabvány hierarchikus kialakításával oldható meg.



2. ábra: Elektronikai egységek kapcsolata

A 2. ábrán a különféle protokollt használó elektronikai egységek kapcsolata látható. A fent bemutatott buszrendszer és sokféle elemének programozása már első ránézésre is bonyolult feladat, aminek megvalósítása az általában feszített időtervű fejlesztési folyamat során nagyban hátráltatja a munka előrehaladását. A leghosszadalmasabb és legtöbb hibalehetőséget rejtő lépés az elektronikai egységek programjának C-nyelven történő megírása, a programok validálása és tesztelése a szériaérett szint eléréséig. Erre kínál alternatív megoldást a következőkben bemutatott módszer.

1.3. Új módszer

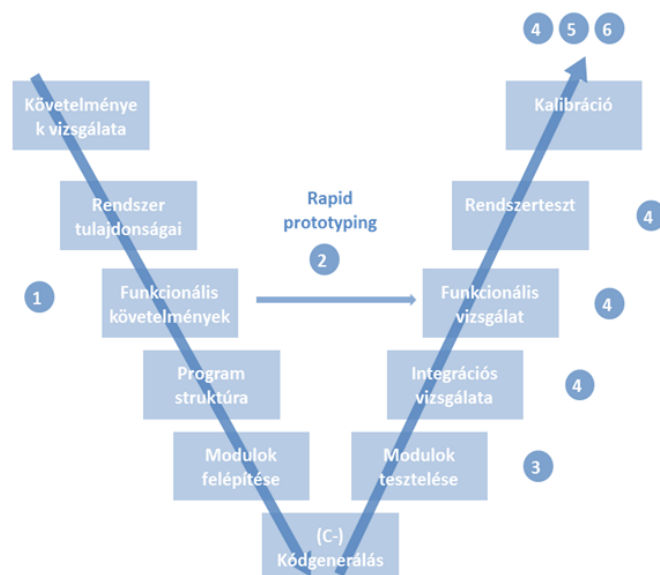
Az új megközelítés szerint az elektronikai vezérlőegységek (ECU) fejlesztése a megrendelő igényeinek részletezésével kezdődik. A rendszer működési vázlatát után az egyes funkciók leírása következik, ebben a fázisban van először lehetőség Model-In-The-Loop (MiL) szimuláció alkalmazására. Ez egy jellemzően grafikus programnyelven, például - esetünkben is – MATLAB Simulink segítségével készült modellt tartalmaz, mely a funkciók működését reprezentálja (3. ábra (1)). Ezen modellek segítségével kezdetét veszi a tesztelés, már a fejlesztés ilyen korai szakaszában, például egy külső számítógépről futtatva a járműben, vagy akár rapid prototyping módszerrel (2). Az ilyen elven működő fejlesztőkörnyezet legnagyobb előnye, hogy az igényekhez szabott modelleket – melyeken már tesztek hajthattunk végre – képes átfordítani C-kódra, töredékére csökkentve ezzel a szoftverfejlesztés idejét. Az így elkészült szoftver segítségével verifikálható a modell, amely mostmár C-nyelven íródott programként van jelen és már a végleges viselkedés is tesztelhető rajta, ezt a fázist Software-In-The-Loop-nak nevezi a szakma angol szakszóval (3). Az ezt követő integrációs teszt során a program más modulokkal való együttműködését vizsgálják, ekkor már gépi kódra fordítva és az alkalmazási területnek megfelelő hardveren (ECU), vagy

egy azt szimuláló számítógépes környezetben futtatva (4). Az ECU modul kalibrálása, mely a működési körülményekhez való jobb illesztést és a hatékonyabb működést segíti, ezután következik, immár Hardware-In-The-Loop folyamatként. A HiL szimulátorokat növekvő arányban használják az ideális jelleggörbék meghatározására, ezt offline-módnak nevezzük. Ilyenkor lehetnek segítségünkre az ún. DoE, azaz Design of Experiment módszerek, amiket például az AVL CAMEO szoftvere képes biztosítani (5). Segítségével a fékpadon, vagy más mérőrendszeren tényleges méréssel töltött idő nagymértékben csökkenthető a rendelkezésünkre álló modellek miatt. Ezeket képes a CAMEO az eredetileg szükséges mérések számának töredékéből felépíteni.

Szokás a HiL módszert tényleges fékpadon elvégzett mérés közben is használni, ez esetben a már elkészült motor mögé felépíthető a teljes jármű modellje, mely magában foglalja a teljes erőátviteli rendszert, vezetőt és a gördülési ellenállást is. Így akár egy teljes NEDC ciklusmérés is leszimulálható egy még el nem készült jármű és egy már meglévő motor kombinálásával (6).

1.4. V-modell

Az eddig megismert struktúrák és metódusok mentén körvonalazódik egy folyamat, mely során egy elképzelésből működő szoftver lesz. Ezt a folyamatot V-modellként, vagy V-diagramként szokták nevezni. Bár vannak más struktúrák is a metódus leírására, ez a legismertebb, leggyakrabban alkalmazott. A 3. ábrán mutatjuk be a V-struktúrát és annak lépéseit.



3. ábra: A V-modell

1.5. Modell integráció

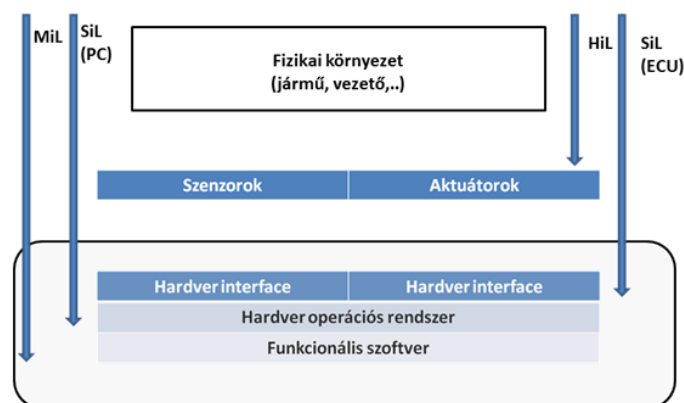
A fejlesztés korábban taglalt lépcsőfokain a fizikai rendszerről alkotott modell az integráció különböző szintjein megy át:

MiL: a teljes fizikai környezet, a hardver és a szoftveres funkciók is modellek formájában léteznek, melyek alkalmasak az célfeladat szimulálására. MATLAB Simulink formában lévő modellek.

SiL: fizikai környezet és hardvereszközök modell formában, szoftver C-nyelvű programkód formában, célhardver (ECU) feladattól függően lehet szimulált, és ebben az esetben a fejlesztő számítógépén fut a szimuláció, vagy futhat a célhardveren, esetleg prototípus eszközön.

HiL: csak a fizikai környezet van modellezve, gépi kód formában lévő szoftver fut a célhardveren, szenzorok és aktuátorok is bekötve.

A 4. ábra szemlélteti a modellezés mélységét a fejlesztés különböző fázisaiban egy motorvezérlő ECU példáján keresztül.



4. ábra: A fejlesztés különböző fázisai

1.6. A modellek felépítése

A jármű elektronikai rendszere több funkcióterületre különíthető el, ilyenek például a motorvezérlés, menetstabilizáló rendszer, szórakoztató-elektronika, vagy világítás. Modellalkotás szempontjából célszerű ezeket külön rendszerként kezelni és jellemző a további csoportosítás is. Ezek a „funkciócsaládok” egy adott berendezés különböző üzemállapotait hivatottak működésbe hozni. Például a világítás rendszeren belül definiáljuk az irányjelző modult, aminek kétféle feladata lehet: irányjelző, vagy vészvillogó.

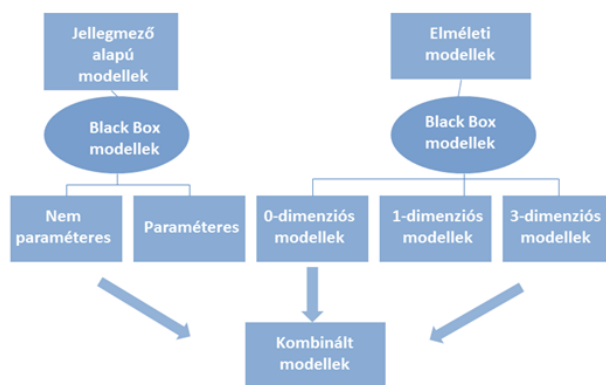
A fejlesztés különböző szakaszaiban, ahogyan azt már tárgyaltuk, más követelményeknek kell megfelelniük a modelleknek. Ezért léteznek funkció- és implementációs modellek.

- A funkciómodellek az adott rendszer tulajdonságait reprezentálják, nem tartalmazzák a jelfeldolgozó és diagnosztikai blokkokat és nem kötöttek hardverhez. Jellemzően a MiL fázisban használjuk őket.
- Az implementációs modellek ezzel szemben a szoftveres követelményekhez igazodnak, azaz úgy működnek, mint egy program, ami vezérli a kimeneteket, jeleket fogad és dolgoz fel a bemeneteken, majd ennek megfelelően állítja be a szükséges paramétereket és ad ki további jeleket. Ennek megfelelően tartalmaz jelfeldolgozó egységet és hibaszűrő algoritmusokat is. A fejlesztési folyamatban a SiL fázisban fordulnak elő.

Az autógyárak számos termékkel vannak jelen a piacon, melyek fejlesztése során a kompatibilitás sok problémát okozott korábban, ezért az autóiipari vállalatok létrehozták a saját, nyílt szoftverarchitektúrájukat, az AUTOSAR-t (Automotive Open System ARchitecture). Ez megkönnyíti az átjárást az autógyártók, beszállítók és más fejlesztők munkájában. Ez nemcsak a modellalkotás módját, a felhasznált blokkokat szabványosítja, hanem definiál egy struktúrát is, mely a fejlesztés során követendő.

1.7. Gyakorlati alkalmazások

A módszer alkalmazásának közelebbi megismerése céljából tekintsünk egy gyakorlati példát, esetünkben egy belső égésű motor modellezését. Egy ilyen összetett rendszer leírásakor jellemzően figyelembe kell vennünk fizikai törvényszerűségeket, melyek ismertek az adott motor esetén, illetve bizonyos változók elméleti összefüggéseit, feltételezéseket is. A végleges modell leginkább ebbe a két csoportba tartozó paraméterekből épül fel. A mérésen alapuló paramétereket az ún. „Black Box Model”, az elméleti összefüggésekre épülő fizikai modelleket a „White Box Model-t” tartalmazza. Előbbiek lehetnek paraméteres és nem paraméteres formában lévő adatok, utóbbiak pedig 0, 1, vagy 3 dimenziós modellek. A paraméteres formában ismert mérési adatokból DoE módszerekkel, vagy neurális hálók alkalmazásával a teljes rendszerre vonatkozó matematikai összefüggéseket lehet nyerni. HiL szimuláció esetén az 1 és 3 dimenziós modellek túl nagy számítási igényt jelentenek ezért legtöbbször 0 dimenziós modellek segítségével bontják fel a motort alkotóelemeire. Jellemzően ezeket a modelleket kombinálják a DoE módszerekkel nyert matematikai modellekkel. Ezt a felépítést szemlélteti az 5. ábra.



5. ábra: Modellek

1.8. Kiválasztás

Az ebben a fejezetben részletezett szoftverfejlesztési módszer több jellemzője miatt is felkeltette az érdeklődésünket. A virtuális motor-szimulációs labor számára először egy olyan fékpád vezérlésre alkalmas szoftvert kerestünk, amin keresztül lehetséges a virtuális környezet (motor, próbapád) szimulálása is és képes ezt a virtuális környezetet úgy kezelni, mint ahogyan egy valóságos berendezéssel tenné. Másik szempont az volt, hogy a hallgatók olyan kezelőfelülettel találkozzanak, amivel az iparban is jó eséllyel találják magukat szembe. Amikor megvizsgáltuk a piacot ilyen téren, ezen szempontoknak megfelelt az AVL CAMEO nevű programja, azonban mikor részletesebben is megvizsgáltuk, felfedeztük egy sokkal hasznosabb tulajdonságát is. A CAMEO alkalmas a fékpadi mérésekből DoE módszerek útján matematikai modelleket alkotni, melyek aztán felhasználhatóak a virtuális motor és fékpád felépítésére. Mivel a harmadik fő kritériumunk az volt, hogy találjunk egy modellalkotásra alkalmas eszközt a virtuális labor elkészítéséhez, a CAMEO erre a feladatra is kiválóan alkalmasnak bizonyult.

Ahogyan azt ez a fejezet is bemutatta, a modell alapú szimulációs módszerek sokfélék lehetnek, sok célra használhatók fel és még sok a kiaknázatlan lehetőség a technológiában. Ennek szellemében igyekeztünk dolgozatunkat többféle modellalkotási módszer kipróbálásával alátámasztani. A CAMEO-val készült matematikai modellek után szeretettük volna kipróbálni a fizikai modellek pontosságát is. Erre másik programot kellett találni, választásunk a szintén AVL által készített BOOST-ra esett. Elsősorban azért, mert mindenben alkalmas volt a célra, másodsorban pedig azért, mert azonos fejlesztő révén kompatibilitási problémák kisebb eséllyel merülhetnek fel, mintha más cég termékét próbáltuk volna meg az előző rendszerbe illeszteni. A Belső Égésű Motorok Tanszéken már régóta használatos a szoftver az oktatásban is, ezért a más tanórákon és foglalkozásokon elkészített BOOST modelleket is ki lehet próbálni a virtuális fékpád laboron keresztül.

2. Jellegmezőkre épülő fékpad- és motormodellek

2.1. Feladat

Ahhoz, hogy a hallgatók úgy használhassák a CAMEO szoftvert, ahogyan azt egy valós fékpadon is tennék, a motort és fékpadot egy modellel kell helyettesíteni. Olyan modellek megalkotására kell törekedni, melyek univerzális módon használhatóak benzines és dízel motortípushoz egyaránt. Könnyedén illeszthetők más-más motorhoz néhány adatsor kicserélésével, megkönnyítve ezzel a későbbiekben az oktatók munkáját, ha más motor adatait szeretnék a szimulációba beilleszteni.

2.2. Célkitűzés

Ahogyan az a bevezetőben és a kitekintésben is olvasható, projektünk már több köztes állapoton esett át, mindegyik önmagában is értékelhető és működőképes eredményeket hozott, és a jövőre nézve is vannak terveink arra, hogyan lehetne az eddigiektől eltérő területeken is hasznosítani a kidolgozott módszert. A szoftveres kezelőfelület és a mérési környezet megismertetése volt a célunk a hallgatókkal. Mindamellet, hogy költséges berendezések nélkül ismerhessék meg a fékpadon végzett vizsgálatok lefolyását és egyszerre több hallgató is végezhesen egyedi, önálló méréseket.

2.3. Modellalkotás

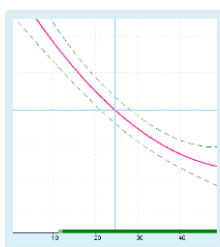
Mivel a CAMEO a modern HiL módszeres szimulációs feladatokra is fel van készítve, ezért lehetőséget ad rá, hogy a kommunikációs csatornák definiálásánál, azokat egy MATLAB Simulink fájlból generált C-kódból töltsük be. Ehhez először létre kellett hoznunk a virtuális környezet Simulink modelljét, mely a továbbiakban a motor és a fékpad helyettesítésére szolgál.

Három modellt készítettünk el, melyek közül kettő benzinmotor, a harmadik egy dízelmotor adatait tartalmazza. A harmadik nem kerül bemutatásra, mivel felépítése az első modellel megegyezik. Mindhárom esetben egy valódi motor lemért adatait értékeltük ki és használtuk fel. A modellek több alegységből épülnek fel. Erre azért van szükség, mert a CAMEO a bemenő és kimenő pontokat vizsgálja, ezért az általunk a CAMEO-ban később megjeleníteni és a mérésbe beilleszteni nem kívánt pontok szubmodellekbe kerültek. A modellek színekkel lettek ellátva, ezek az adott egységek egymáshoz tartozását, illetve az egyszerűbb átláthatóságot hivatottak szolgálni.

2.3.1. Az első modell

Az általunk elkészített első modell egy benzinmotor. Elsőként a CAMEO-t hívtuk segítségül. Az AVL munkatársai továbbképzésére épített több olyan modellt is, mely az adott motor teljes valódi mérési adatait tartalmazza, a szoftver oktatását továbbképzési célokra megkönnyíti. A tőlük kapott modellek segítségével a saját céljainknak megfelelő számítógépes mérést állítottunk össze, hogy az adatokat kinyerhessük és felhasználhassuk a saját modelljeink megalkotásához. Az első modellnél beállítottuk a CAMEO-t úgy, hogy a fordulatszám és az effektív nyomaték állításával, lambda értékét 1-en tartva megkapjuk a motor fajlagos fogyasztását, illetve károsanyag-kibocsátását, beleértve a CO, HC, NO_x értékeket. A valós méréseket mindig lambda 1-el végzik, mivel a motor is ott működik a hármas hatású katalizátor miatt.

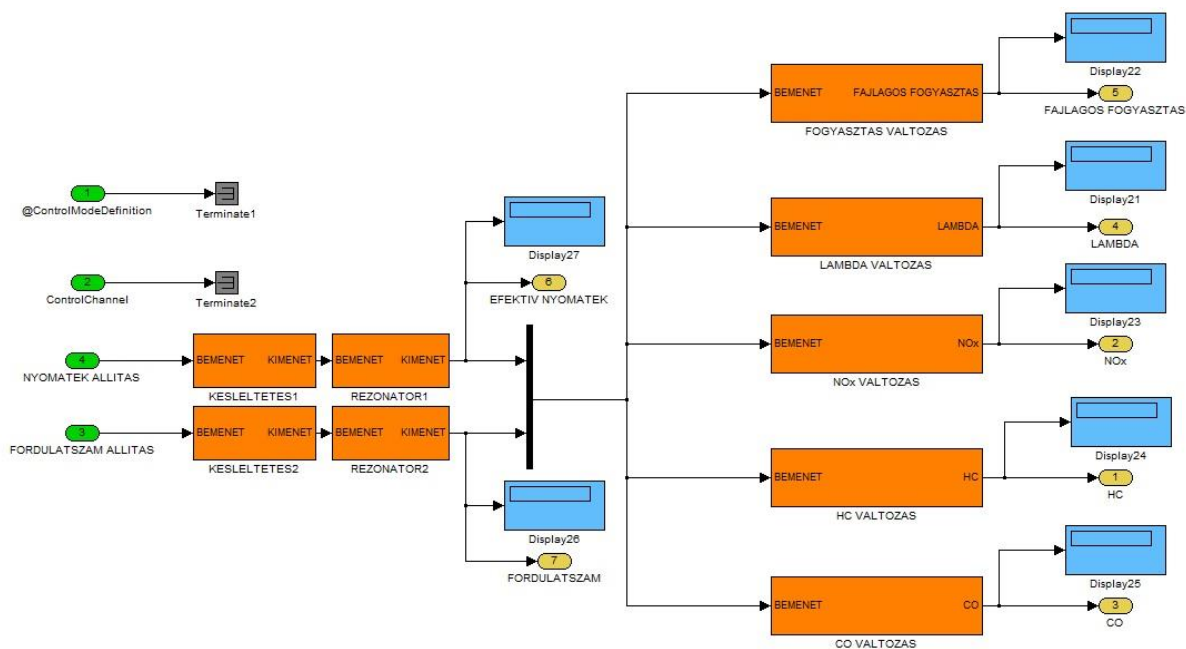
Fontos volt arra ügyelni, hogy az adatokat olyan tartományból nyerjük ki, ahol az úgynevezett Design Space-en belül vagyunk. A Design Space tartományára egy példa az 6. ábrán tekinthető meg. Az ábra alján látható zöld sáv mutatja, a tartomány határait. Ezen belül történt tényleges mérés, ha az adatokat a kívül eső tartományból vesszük, a modell már nem tudja teljes biztonsággal megadni a valóságnak megfelelő értékeket, csupán megbecsüli azokat. A bemenő adatoktól függően a modell rengeteg dimenziós is lehet, estünkben a modellek öt és hat dimenziósak voltak, ami az ember számára már felfoghatatlan, de egy program könnyedén tud vele számolni.



6. ábra: A Design Space

A Matlab Simulink-ben elkészített modell a 7. ábrán látható, melynek részletes felépítése a továbbiakban kerül bemutatásra. A modell elkészítésénél az egyik legnagyobb kihívás az volt, hogy miután a CAMEO-ba illesztés megtörtént, valóban úgy viselkedjen, mintha egy valódi motort mérnénk egy valós fékpadon. A mért értékek nem lehetnek konstansok a mérés alatt, hiszen azok egy valódi mérésen is folyamatos változást mutatnak egy bizonyos tartományon belül, a szenzorok mérési pontosságának függvényében. Ezen kívül az egyes bemenő paraméterek megváltoztatásánál nem kaphatunk rögtön visszacsatolást a kimenő adatoknál, mivel a valódi motornál is idő kell, hogy elérjünk egy adott fordulatszám,

vagy nyomaték értéket. A modell bemeneti pontokból, késleltetés szubmodellből, rezonátor szubmodellből, az egyes kimeneti paraméterek szubmodelljeiből (Fajlagos fogyasztás, Lambda, NO_x, HC, CO), kijelzőkből és kimeneti pontokból épül fel.



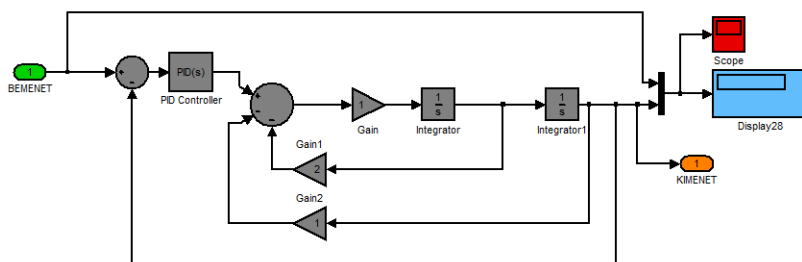
7. ábra: Az első modell sematikus ábrája

A programok közül egyik sem magyar nyelvű, de az egyszerű megértés kedvéért magyarul neveztünk el minden fontos paramétert, a többit a Matlab automatikusan felkínált nevén hagytuk. Mivel később pontosan ezen a néven kerülnek a CAMEO-ban megjelenítésre és ott már nincs lehetőség megváltoztatni őket, ezért fontos már a Matlab modellben ügyelni az elnevezésre. A következőekben az egyes szubmodellek felépítését, feladatát mutatjuk be részletesen.

2.3.2. Késleltetés szubmodell

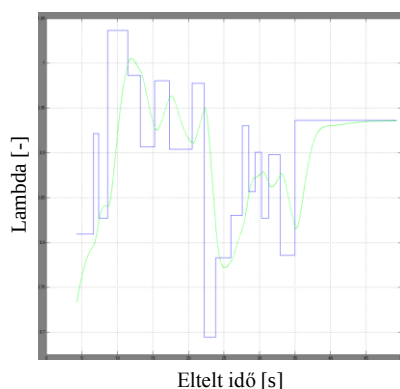
A késleltetés szubmodell a bemenetről érkező hirtelen változásokat megakadályozza. A kimeneti értékeknél a változás időben eltolva jelenik meg, a valóságban is lejátszódó folyamatoknak megfelelő módon. Ennek a szubmodellnek a beépítése a sikeres laborgyakorlat szempontjából elengedhetetlen volt, mert fontosnak tartottuk, hogy a diákok megértsék, hogy egy motoron az egyes paraméterek állandósulása időt vesz igénybe.

A modell lényegében egy párhuzamos kompenzációról alapuló, úgynevezett PID szabályozó. A sematikus vázlata a 8. ábrán látható. A piros színű Scope elemet használtuk fel a szabályozókör beállítására.



8. ábra: A késleltetés szubmodell sematikus ábrája

A működésére egy példa található a 9. ábrán. A képen a második modellben használt lambda változtatására adott reakció látható. Az ábrán a vízszintes tengely az időt mutatja másodpercben, a függőleges tengely pedig a lambda értékét. A kék vonal a bemenő jel, aminek változása rögtön jelentkezne a modell használata nélkül. A zöld vonal már a szabályozókör által adott választ jelenti a bemenő jelre. Látható, hogy csak némi idő eltelte után tud a kimenő jel a bemenő jellel megegyező értéket felvenni, teljesítve ezzel az általunk előírt követelményeket.

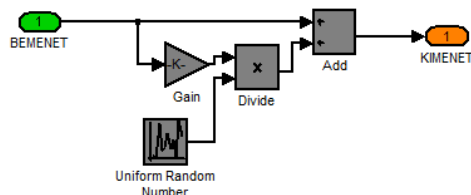


9. ábra: A késleltetés szubmodell működés közben

2.3.3. Rezonátor szubmodell

A rezonátor szubmodell hivatott a kimeneti konstans értékeket a valóságnak megfelelően folyamatos instabilitással ellátni szűk határokon belül. A motorikus folyamatok soha nem állandók, a motorra szerelt szenzoroknak is van hibahatárunk, ezért a modellben is változnia kell a kimeneti értékeknek egy szűk határon belül. Ezzel a szubmodellel az volt a célunk, hogy a diákok annak ellenére, hogy egy számítógépes szimuláció fut a háttérben, azt érezzék, hogy egy valódi motort mérnek pont úgy, mint egy valós fékpadon. A sematikus

vázlat az 10. ábrán látható. A felépítése egyszerű, a bemenő jelhez véletlenszerűen hozzáadja vagy kivonja a műszer pontossági határán még belül eső értéket. Ezt az értéket mi határozzuk meg, megszorozzuk véletlenszerűen +1 vagy -1 értékkel, ennek hatására időben folyamatosan változni fog a kimeneti jel értéke.

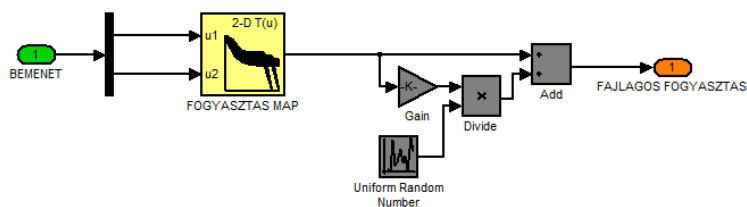


10. ábra: A rezonátor szubmodell sematikus ábrája

2.3.4. Fogyasztásváltozás szubmodell

A fogyasztásváltozás szubmodellben egy háromdimenziós felületként van megadva az egyes pontokban mért fogyasztásértékek, a nyomaték és a fordulatszám változás függvényében. A modell sematikus vázlata a 11. ábrán látható. A két bemeneti paraméterhez egy kimeneti érték a válasz, amit a rezonátor szubmodellnél bemutatott funkciót ellátó blokkok követnek. A szubmodellből kilépve végül ez az adat fog megjelenni a főmodell kimenetén, a CAMEO ezt fogja kijelezni.

A kapott eredmények szemléletes megjelenítésére keresnünk kellett egy megoldást. A fajlagos fogyasztás kagylódiagramját szerettük volna bemutatni a diákok számára. Ehhez felhasználtuk a CAMEO Map Generator felületét. A szoftver eredetileg a motor vezérlőegységében programozott térképek kalibrálására lett kialakítva, viszont az általunk megjeleníteni kívánt adatokhoz is tökéletes, némi átalakítással.



11. ábra: A fajlagos fogyasztás szubmodell sematikus ábrája

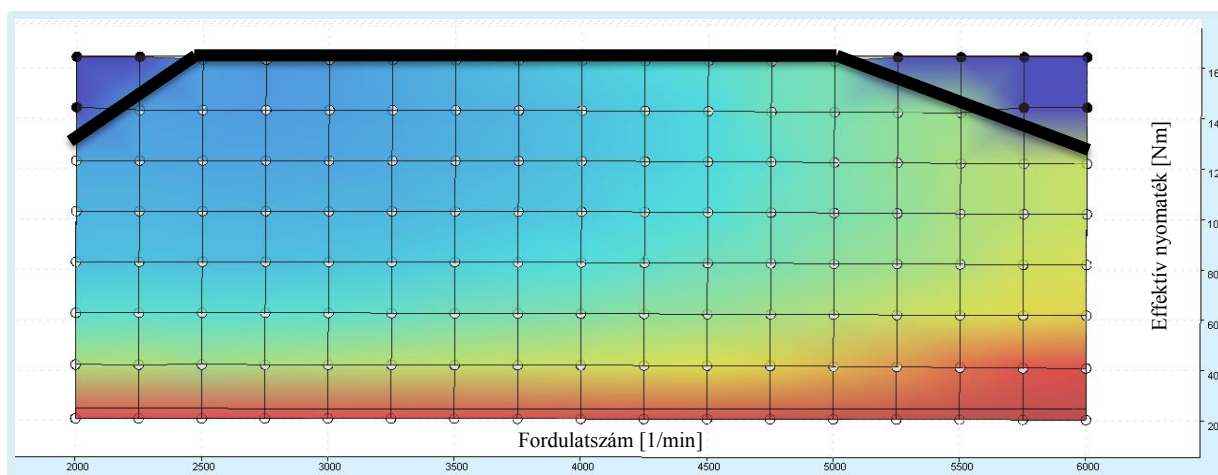
Az 1. táblázatban látható adatokat mértük le, ezek kerültek a Matlab-ba, illetve ezeket használtuk fel a kiértékelő diagramoknál is. A vízszintes tengelyen a percenkénti fordulatszám értékek találhatóak, a függőlegesen az effektív nyomatékok Nm-ben. Ezekhez páronként kapcsolódik az adott pontban mért fajlagos fogyasztás g/kWh-ban. A táblázatból

látható, hogy milyen tartományban végeztük a mérésünket. Az értékek $\lambda=1$ -nél kerültek mérésre, ezért a Lambda változás szubmodell nem kerül bemutatásra, abban a táblázatában minden érték 1. Egyedül egy rezonátor elem lett a modellbe beépítve, hogy az értékek ne pontosan 1-en álljanak minden időpillanatban.

1. táblázat: A fajlagos fogyasztás értékei

M/n	2000	2250	2500	2750	3000	3250	3500	3750	4000	4250	4500	4750	5000	5250	5500	5750	6000
20	500	500	500	500	500	503	506	509	512	515	518	521	525	531	537	543	550
40	355	360	350	355	360	365	370	375	380	385	390	400	420	440	470	490	500
60	298	300	297	298	300	305	315	320	325	335	340	350	360	375	385	395	400
80	265	268	265	265	268	272	277	285	290	300	312	320	335	345	360	380	385
100	257	258	256	255	257	258	263	268	273	284	291	310	320	335	355	365	375
120	249	249	247	247	248	251	255	260	268	275	285	300	320	337	355	365	375
140	0	244	242	240	243	248	254	258	267	275	285	310	320	335	345	0	0
160	0	0	243	238	239	250	257	265	275	285	296	310	317	0	0	0	0

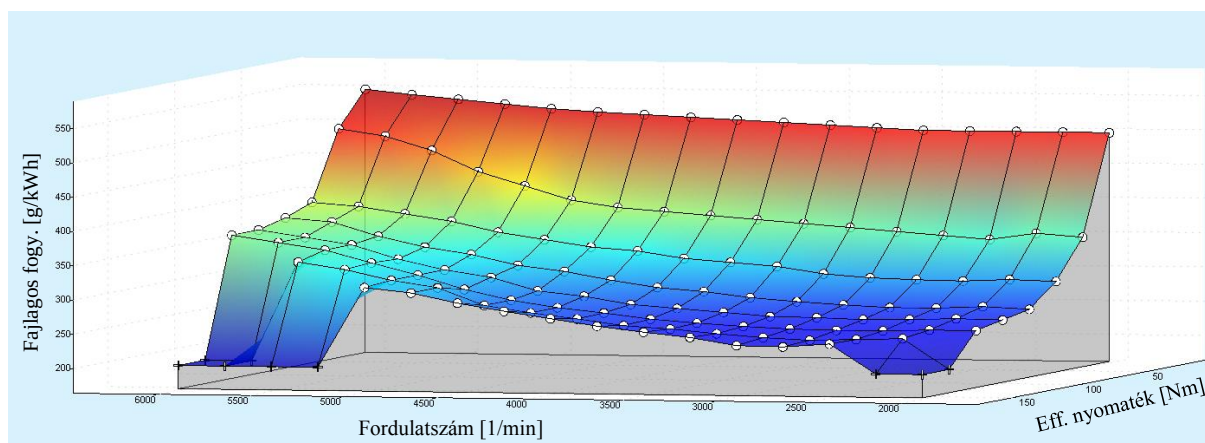
A 12. ábrán látható a fajlagos fogyasztás kagylódiagramja. A kék szín jelöli mindig a legalacsonyabb értéket, a piros a legmagasabbat. A köztes értékeket a program színátmenettel teszi szemléletessé. A vastag fekete vonal a motor maximális terhelését mutatja. A Matlab-nak és a CAMEO-nak a helyes működéshez minden értékre szüksége van, ezért a tartományon kívül eső értékeket 0-nak adtuk meg. Így a diákok is rögtön látni fogják, hogy a motor az általuk beállított értékekkel már nem képes működni.



12. ábra: A fajlagos fogyasztás 2D-s ábrája

A szoftver valójában ezeket az értékeket három dimenzióban jeleníti meg elősegítve a mérnökök munkáját. A szemléletesség kedvéért készítettünk mindegyik kimenő paraméter 3D-s felületéről is egy-egy képet. A képeken a működési tartományon kívül eső pontok nem 0-ák, mindig közel vannak a mellettük található ponthoz. Ennek oka, hogy a színek eloszlása így egyenletesebb, ezáltal könnyebb átlátni az ábrát.

A fajlagos fogyasztás 3D-s ábrája látható a 13. ábrán. A szoftverben a működésen kívül eső tartományban lévő pontokat feketére állítottuk át. A diákok a mérés során az általuk kapott értékeket szintén megtekinthetik ezen a felületen. Könnyen átlátható, bármilyen pozícióba elfordítható, ezért a megértést nagyban elősegíti.



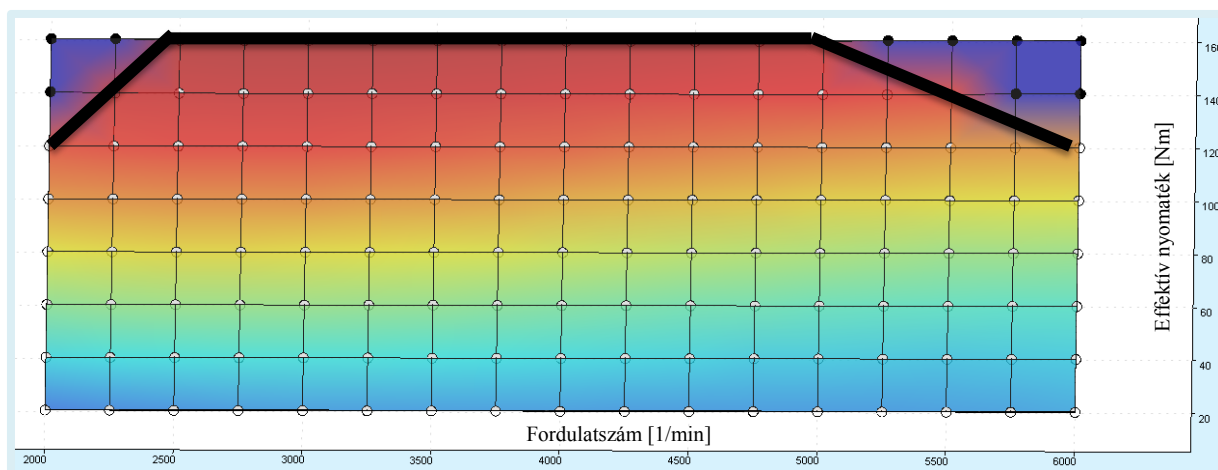
13. ábra: A fajlagos fogyasztás 3D-s ábrája

A többi kimeneti paraméter szubmodelljére csak kevésbé részletesen térünk ki. Ennek oka, hogy felépítésben megegyeznek ezzel a szubmodellel. A különbség csupán a beprogramozott háromdimenziós felületekben van. A modellek úgy lettek elkészítve, hogy ezeket a felületeket egy külső fájlból olvassák be. A felületek semmilyen változtatása nem befolyásolja a modellek működőképességét, ezáltal később bármelyik oktató megváltoztathatja őket tetszés szerint.

2.3.5. NO_x változás szubmodell

A szubmodell felépítését tekintve megegyezik a Fogyasztásváltozás szubmodellel, ezért csak a programozott adatok kerülnek bemutatásra. A 14. ábrán a NO_x kibocsátás látható a fordulatszám és a nyomaték változásának függvényében. A vastag fekete vonal itt is a motor maximális terhelését jelenti, a feketével jelölt pontokon a modell 0 értéket ad itt is.

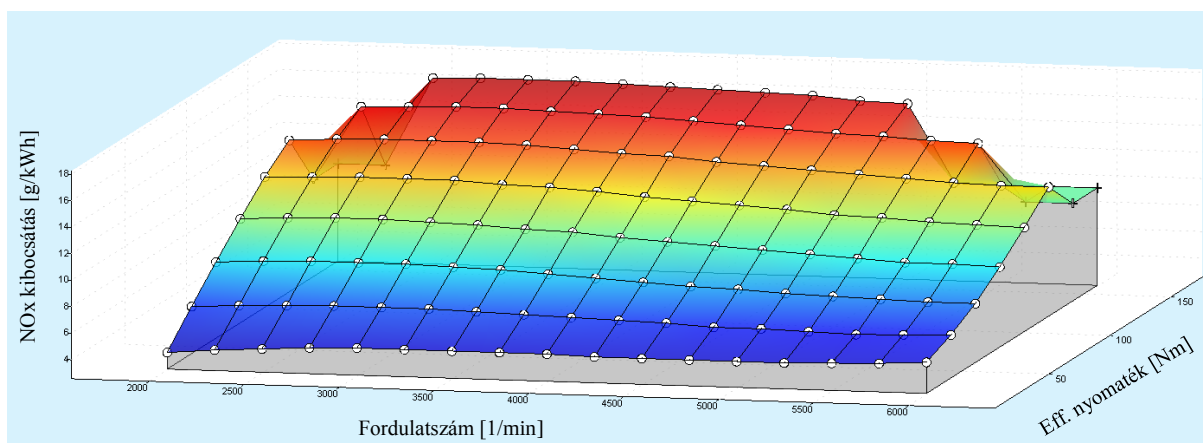
A programozott adatok a 2. táblázatban találhatóak, a táblázat felépítése a Fajlagos fogyasztás szubmodellnél ismertetett módon történt ebben az esetben is. Az adatok háromdimenziós képe a 15. ábrán látható.



14. ábra: A NOx kibocsátás 2D-s ábrája

2. táblázat: A NOx kibocsátás értékei

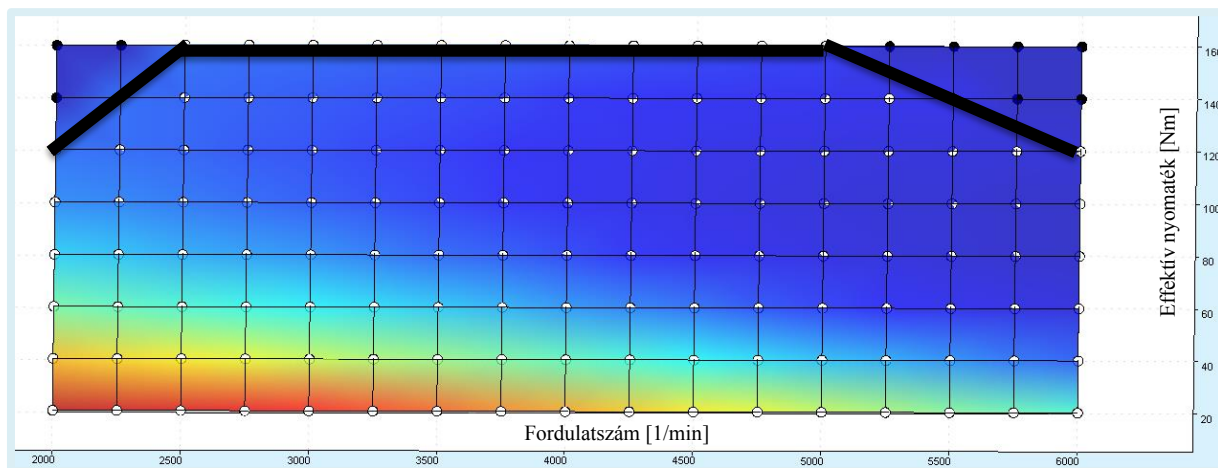
M\η	2000	2250	2500	2750	3000	3250	3500	3750	4000	4250	4500	4750	5000	5250	5500	5750	6000
20	3,9	4,2	4,4	4,6	4,7	4,7	4,7	4,7	4,7	4,6	4,6	4,6	4,6	4,6	4,7	4,8	5
40	6,2	6,4	6,5	6,6	6,6	6,6	6,5	6,4	6,3	6,2	6,1	5,9	5,9	5,8	5,7	5,8	5,9
60	8,4	8,6	8,7	8,7	8,7	8,6	8,5	8,3	8,1	7,9	7,7	7,6	7,4	7,3	7,2	7,1	7,1
80	10,5	10,7	10,8	10,8	10,7	10,6	10,4	10,3	10	9,8	9,6	9,4	9,2	9	8,8	8,8	8,7
100	12,5	12,6	12,7	12,7	12,7	12,5	12,4	12,2	11,9	11,7	11,5	11,3	11	10,9	10,7	10,6	10,5
120	14,1	14,3	14,4	14,4	14,4	14,3	14,1	13,9	13,8	13,5	13,3	13,1	12,9	12,7	12,5	12,4	12,4
140	0	15,6	15,7	15,8	15,8	15,7	15,6	15,5	15,3	15,1	14,9	14,7	14,6	14,4	14,3	0	0
160	0	0	16,7	16,8	16,8	16,8	16,7	16,6	16,5	16,4	16,3	16	15,9	0	0	0	0



15. ábra: A NOx kibocsátás 3D-s ábrája

2.3.6. HC változás szubmodell

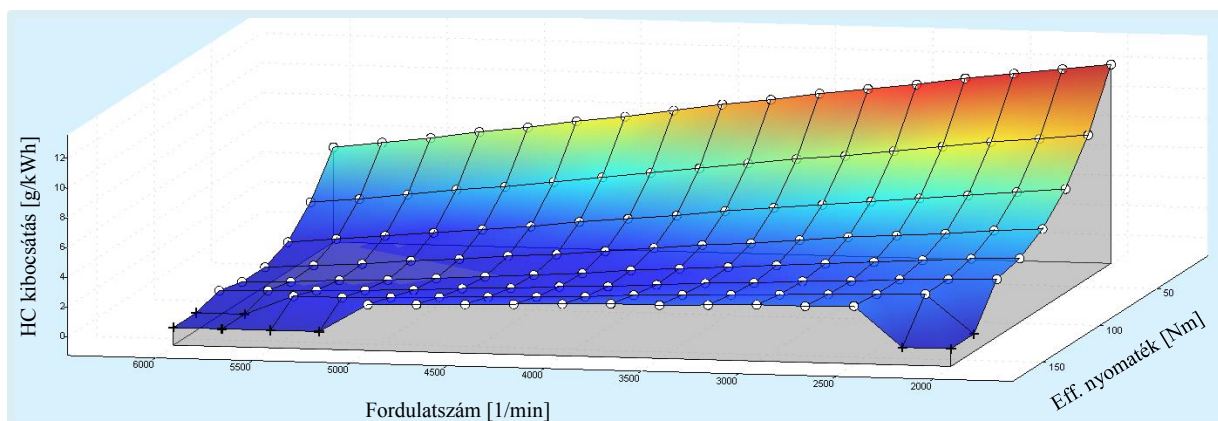
A szubmodell sematikus felépítése az előzőekkel megegyezik. A HC kibocsátás a 16. ábrán látható. A modellben található adatok a 3. táblázatban olvashatók. A táblázat a Fajlagos fogyasztás szubmodellnél ismertetett módon készült. Az értékek alapján kirajzolt háromdimenziós felület a 17. ábrán látható.



16. ábra: A HC kibocsátás 2D-s ábrája

3. táblázat: A HC kibocsátás értékei

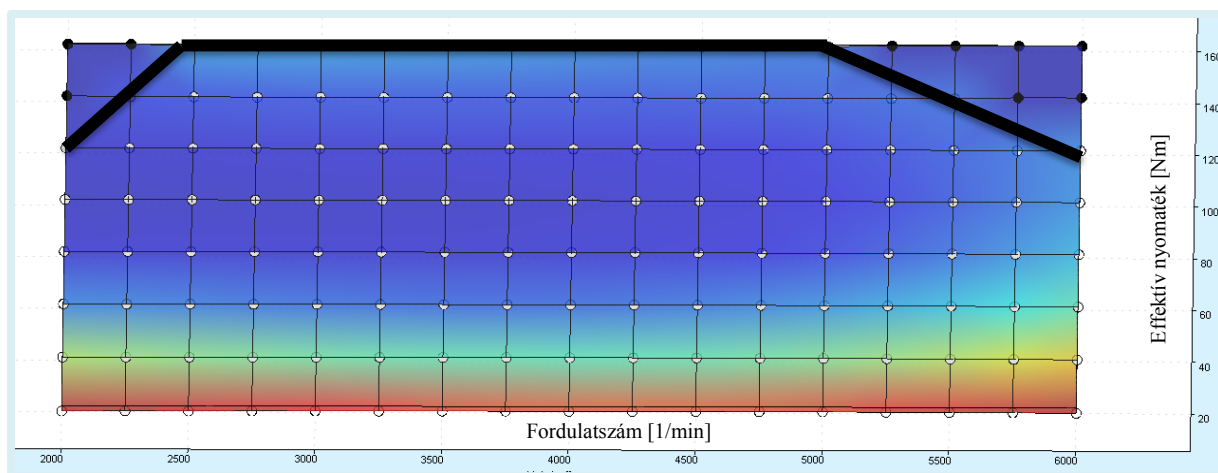
M\ $\dot{n}$	2000	2250	2500	2750	3000	3250	3500	3750	4000	4250	4500	4750	5000	5250	5500	5750	6000
20	12,2	11,8	11,4	11	10,5	10,1	9,7	9,2	8,8	8,3	7,8	7,4	6,9	6,5	6	5,6	5,2
40	8,45	8,1	7,8	7,4	7	6,6	6,3	5,9	5,5	5,1	4,7	4,3	3,9	3,6	3,2	2,8	2,5
60	5,8	5,5	5,2	4,9	4,6	4,3	4	3,7	3,3	3	2,7	2,3	2	1,7	1,4	1,1	0,8
80	4,1	3,9	3,6	3,4	3,2	2,9	2,6	2,4	2,1	1,8	1,5	1,3	1	0,8	0,5	0,3	0,1
100	3,1	3	2,8	2,6	2,4	2,2	1,99	1,8	1,6	1,3	1,1	0,9	0,7	0,5	0,4	0,2	0,1
120	2,7	2,6	2,5	2,3	2,2	2	1,9	1,7	1,5	1,4	1,2	1,1	0,9	0,8	0,7	0,6	0,5
140	0	2,6	2,5	2,4	2,3	2,2	2,1	2	1,8	1,7	1,6	1,5	1,4	1,3	1,3	0	0
160	0	0	2,7	2,6	2,6	2,5	2,4	2,4	2,3	2,2	2,1	2	1,9	0	0	0	0



17. ábra: A HC kibocsátás 3D-s ábrája

2.3.7. CO változás szubmodell

A sematikus ábra ennél a szubmodellnél is a többivel megegyező. A programozott adatok a 18. ábrán láthatók a fordulatszám és a nyomaték függvényében. A pontos értékeket a 4. táblázat tartalmazza.

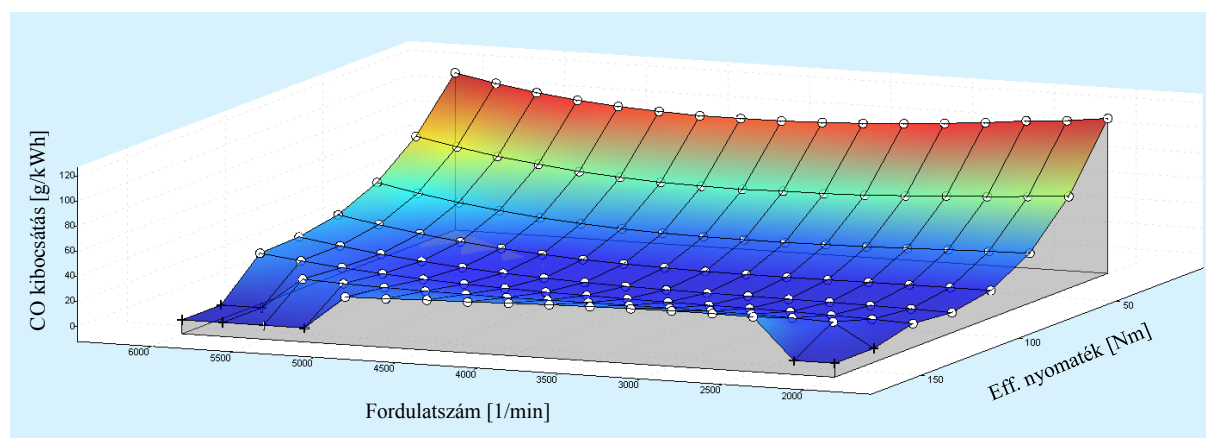


18. ábra: A CO kibocsátás 2D-s ábrája

4. táblázat: A CO kibocsátás értékei

M\n	2000	2250	2500	2750	3000	3250	3500	3750	4000	4250	4500	4750	5000	5250	5500	5750	6000
20	112,5	108,6	106,2	102,6	99,9	97,5	95,5	94,1	93,2	93,0	93,0	94,3	96,1	98,9	102,8	108,0	114,3
40	62,1	59,9	57,7	56,0	54,0	52,3	51,4	51,2	50,4	50,9	51,7	53,6	56,0	59,7	63,7	69,2	75,5
60	28,6	28,0	27,1	26,1	25,2	24,7	24,2	24,3	24,5	25,2	26,5	28,5	31,0	34,6	38,8	44,1	50,4
80	10,8	10,9	10,9	10,8	10,6	10,5	10,6	10,8	11,6	12,5	13,8	15,8	18,3	21,6	25,8	30,8	36,5
100	4,9	5,8	6,3	6,8	7,0	7,3	7,6	8,0	8,6	9,5	10,9	12,6	14,9	17,8	21,3	25,7	30,7
120	8,0	9,4	10,3	11,1	11,5	11,8	12,1	12,5	13,1	13,6	14,5	15,8	17,4	19,6	22,3	25,6	29,5
140	0	19,0	20,0	20,8	21,2	21,3	21,3	21,4	21,3	21,1	21,5	21,9	22,6	23,6	25,0	0	0
160	0	0	32,6	33,1	33,1	32,7	32,2	31,5	30,6	29,5	28,8	27,7	26,9	0	0	0	0

A kapott háromdimenziós felületet a 19. ábra szemlélteti. A modellből generált fájl segítségével a CAMEO úgy érzékeli, mintha közvetlenül egy fékpadra lenne rákötve. A továbbiakban ezt a CAMEO-ban felépített mérést mutatjuk be használat közben.



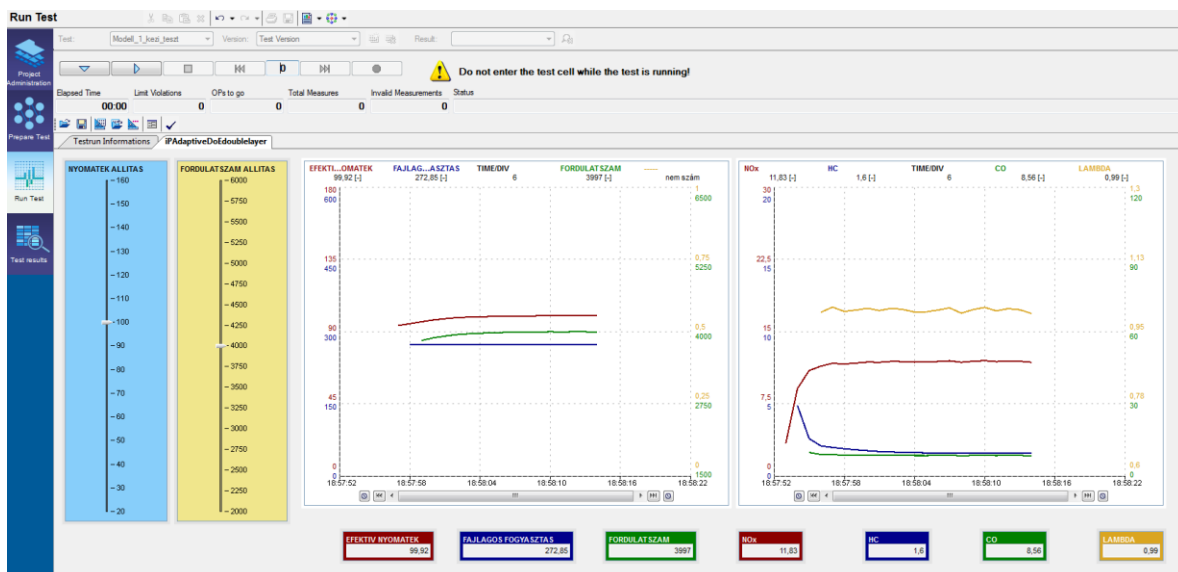
19. ábra: A CO kibocsátás 3D-s ábrája

2.4. Az első modellből készült CAMEO mérés

A mérést a diákok számára legkönnyebben megérthető módon építettük fel. Készítettünk kézzel állítható illetve automata üzemmódban végezhető mérést. Az előbbinél a diákok saját kezűleg változtathatják a bemenő paramétereket és megfigyelhetik a kimenő paraméterek változását. Az utóbbinál a CAMEO automatikusan végigmegy az összes mérési ponton, amit beprogramoztunk előre. Ezt a mérést úgy építettük fel, hogy minden egyes pontot lemérjen, amiből később a diákok megkapják az általunk Matlab-ban programozott értékeket.

A mérés elindítása után a szoftver gyakorlatilag semmilyen különbséget nem ismer fel az általunk definiált modell és egy valódi fékpad között, egy valós mérésnek megfelelő módon viselkedik. A megfelelő értékeket az általunk kívántra állítja, esetünkben ez a percenkénti 4000-s fordulatszámot és a 100 Nm-es nyomatékot jelenti. A diákoknak már csak annyi dolguk van, hogy a csuszkákat állítva vizsgálják a virtuális motor viselkedését és feljegyezzék az adatokat.

A kézzel állítható mérés a 20. ábrán látható működés közben. A színek itt is az egyes paraméterek egymáshoz tartozását, azok könnyebb beazonosítását segítik. A kép röviddel a rendszer indítása után készült, a paraméterek beálltak a megfelelő értékre, innen elkezdhető a mérés.

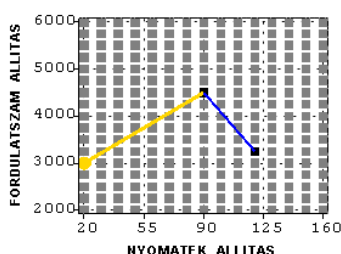


20. ábra: A kézzel állítható mérés, működés közben

A rendszer által az automatikus mérésnél beállított mérési pontok a 21. ábrán láthatók szürke pontokkal jelölve. A program véletlenszerűen választja ki a pontok kombinációját. Erre azért van szükség, mert egy valós motor mérésénél jobb adatokat kapunk, ha nem tartunk

egy paramétert sem konstans értéken, miközben a többi változtatjuk. A szoftver esetünkben sem tesz különbséget a mérési pontok sorrendjének megválasztásában.

A mérés adatait a program folyamatosan mutatja, majd elmenti a számunkra. A 22. ábrán egy részlet látható a mért adatokból. A második sorban található adatsorok az előző oldalakon ismertetett táblázatokkal megegyező értékeket mutatnak, ezzel az általunk készített mérés hitelesnek tekinthető.



21. ábra: Az automatikus mérés pontjai

Information Events Statistics Measurements															
Name	TestVersion	FORDULATSZÁ	NYOMATEK ALLI	ElapsedTime	MeasurementTir	Layer	SubTestNumber	ModelVerificatio	FORDULATSZÁ	LAMBDA	HC	FAJLAGOS FOI	NOx	EFEKTIV NYOI	
System		modell_1_syste	modell_1_syste	RunTest	RunTest	RunTest	RunTest	RunTest	modell_1_syste	modell_1_syste	modell_1_syste	modell_1_syste	modell_1_syste	modell_1_syste	
Unit		-	-	-	-	-	-	-	-	-	-	-	-	-	-
Type		Demand	Demand	Actual	Actual	Actual	Actual	Actual	Mean	Mean	Mean	Mean	Mean	Mean	
1	0	4500	90	46.322	131110193309.9	2	-1	0	4499.7	0.99942	1.2997	301.49	10.544	89.99	
2	0	3250	120	96.664	131110193400.2	2	-1	0	3249.9	0.99947	1.9999	251	14.291	119.9	
3	0	4500	90	146.79	131110193450.4	2	-1	0	4499.4	0.99867	1.2992	301.46	10.535	89.98	

22. ábra: Az automatikus mérésnél kapott adatok

2.5. A második modell

A második modell szintén egy benzinmotor virtuális számítógépes modellje. Az adatokat itt is az AVL-től kapott mérési adatokból nyertük ki, majd készítettünk magunknak az oktatási célra megfelelő modelleket. Itt egy teljesen más motortípus adataival dolgoztunk és a modellünk az előző továbbfejlesztett változatának tekinthető. Beépítettünk egy belső kapcsolót is, melynek segítségével két mérés között tudunk váltani a CAMEO-n belül. Erre azért volt szükség, hogy a motor működését az első modellhez hasonlóan $\lambda=1$ -nél megmutathassuk a diákoknak. Ezt követően megmutassuk egy adott fordulatszámon, állandó fojtószelepállás mellett a λ állításának a károsanyag-kibocsátásra, a fajlagos fogyasztásra és a motor által leadott effektív nyomatékra gyakorolt hatását.

A modell sematikus vázlata a 25. ábrán látható. Az első modelltől néhány elemben tér el. Itt is szubmodellek segítségével adtuk meg azokat a paramétereket, amiket később a CAMEO-ban nem kívánunk láthatóvá tenni. Bekerült egy λ állítás bemeneti paraméter, illetve egy, a váltás megvalósítására alkalmas kapcsolási mechanizmust építettünk fel. A kapcsoló a két mérés közti váltást problémamentesen megoldja. Az 1-es állásban az első

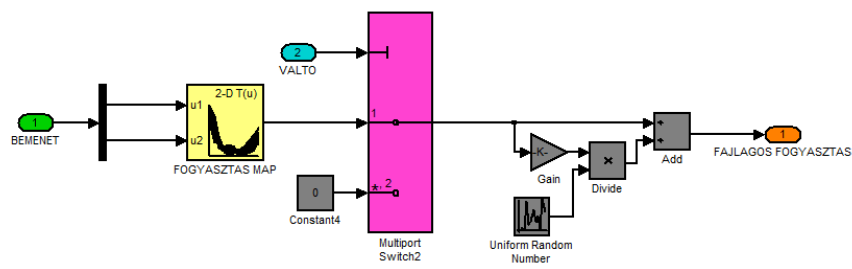
modellhez hasonlóan viselkedik a modell, itt a nyomaték és a fordulatszám az állítható paraméter. A 2-es állásban a lambda állítható, a kimenő paraméterek ennek függvényében változnak. A kimeneteknél és a szubmodelleknél szintén a kapcsoló segítségével váltunk a megjeleníteni kívánt értékek között. A továbbiakban bemutatásra kerülnek a modell elsőtől eltérő, illetve hozzáadott részei is.

2.5.1. A Csatornaváltó elem

A váltás mechanizmusát a Csatornaváltó elem látja el. A CAMEO-ban egy egyszerű tekerőgombot szimuláló elemet helyeztünk el, melynek két állása van. A Csatornaváltóval a kimenetre 1-est vagy 2-est adunk, ezzel a rendszer automatikusan átvált a kívánt mérési módra. Mivel a váltásnak több kapcsolót kell egy időben aktiválnia, ezért olyan bemeneti és kimeneti csatornákat alkalmaztunk, melyek több helyre is beköthetők és egyszerre több helyről is tudják az információt gyűjteni.

2.5.2. Fordulatszám- és terhelésfüggő mérési mód

Az első mérési mód teljesen megegyezik az első modellnél ismertetett módszerrel, egyedül a mérési tartományok különböznek, hiszen egy teljesen más mérésből készült a modell, továbbá kapcsoló elemek lettek elhelyezve minden egyes szubmodellben. A 23. ábra a fajlagos fogyasztás szubmodelljét mutatja. A 11. ábrával összehasonlítva látható az első modellhez képest az eltérés. A váltásnak köszönhetően az értékek a 2. mérési mód aktiválásával nullázódnak, így a teljes 1. mérési mód nulla értékeket fog a kimenetre adni. A kapcsoló után itt is minden szubmodellben megtalálható az értékeket folyamatos mozgásban tartó blokkok csoportja.



23. ábra: A fajlagos fogyasztás szubmodelljének sematikus ábrája

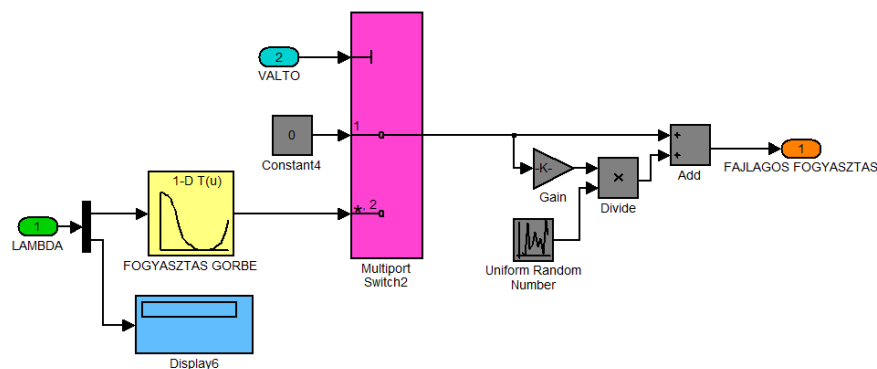
A továbbiakban nem mutatjuk be az 1-es mérési mód szubmodelljeinek táblázatait, illetve ábráit, mivel a részletes leírás az első modellnél megtalálható, azzal teljesen megegyező, csupán a programozott adatokban tér el attól. Ez a modell a teljes általunk mért tartományban működik, ezért nincsenek a táblázatokban 0 értékek.

2.5.3. Légviszony függő mérési mód

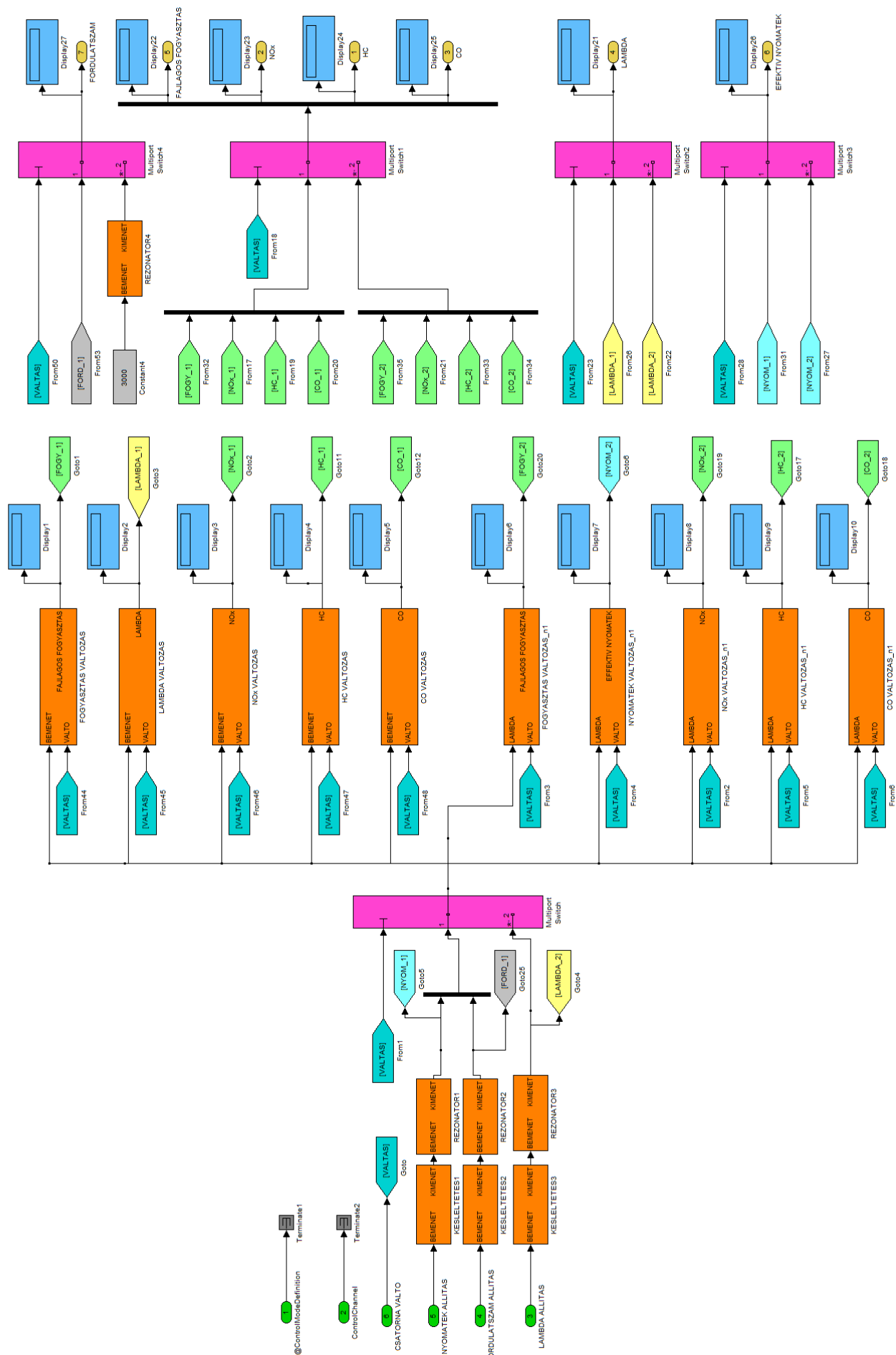
A második mérési módnál 3000-es percenkénti fordulatonál, állandó fojtószelepállás mellett történik a mérés. Ezzel a módszerrel a lambda széles tartományon belül változni fog, megvizsgálhatjuk, hogy a motor károsanyag-kibocsátására milyen hatással van az azonos mennyiségű levegőben elégetett különböző mennyiségű tüzelőanyag. A mérés célja, hogy a diákok az elméletben már tanult diagramokkal a valós mérés folyamán is találkozzanak, azokat könnyebben megértsék.

A kapcsolóra 2-es jelet adva a modellünk átvált erre az üzemmódra. Beállítja a kívánt sebességet, aktiválja a méréshez szükséges szubmodelleket, az első mérésnél alkalmazottakat kikapcsolja, kimenetükön 0 érték jelenik meg. A lambda 0,7 és 1,24 között változtatható, ennek hatására a motor összes paramétere drasztikusan megváltozik. A károsanyag-kibocsátás, a fogyasztás és a motor által leadott nyomaték is széles tartományban fog változni a kedvező, vagy kedvezőtlen égési körülményeknek köszönhetően.

Az szubmodellek felépítése megegyezik az első mérési mód modelljeivel, a különbség, hogy itt csak egy bemenő paraméterünk van, ennek hatására egy görbe mentén fog változni a kimenetünk és nem egy felületet rajzol ki a változók függvényében. Példaképpen a fogyasztásváltozás szubmodell a 24. ábrán látható. A továbbiakban bemutatjuk az egyes szubmodellekbe programozott pontos értékeket és azok görbéit.



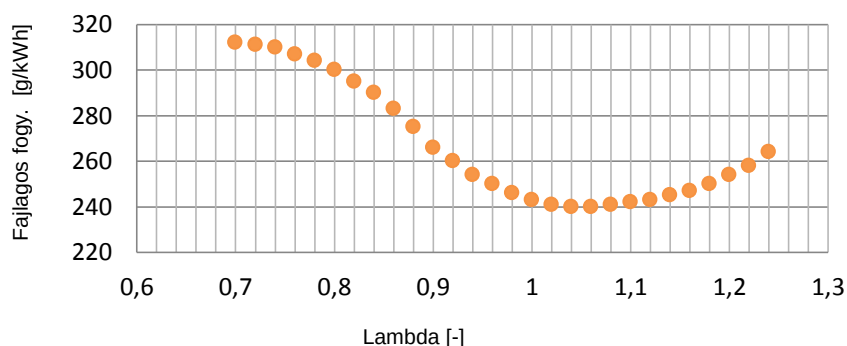
24. ábra: A második mérési mód fogyasztásváltozás szubmodellje



25. ábra: A második modell sematikus ábrája

2.5.4. Fajlagos fogyasztás szubmodell

A sematikus vázlat a 23. ábrán látható. A valódi fékpadon mért adatok egy két oszlopos mátrixban vannak megadva, ezeket olvassa ki a CAMEO a bemeneti lambda paraméter függvényében. Az adatsor diagramja a 26. ábrán látható, az egyes értékek az 5. táblázatban tekinthetők meg.



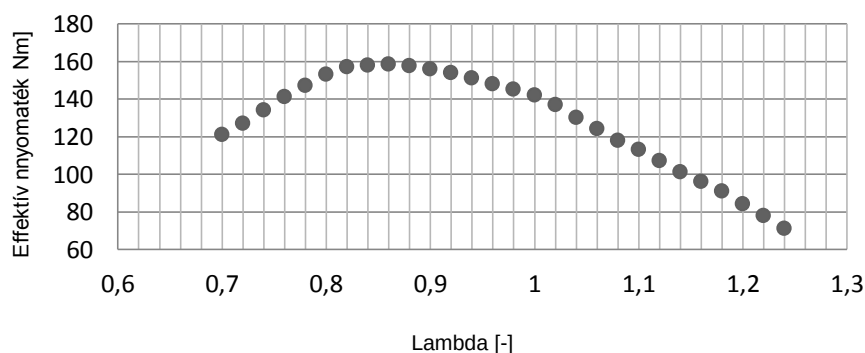
26. ábra: A fajlagos fogyasztás görbéje

5. táblázat: A fajlagos fogyasztás értékei

LAMBDA	0,7	0,72	0,74	0,76	0,78	0,8	0,82	0,84	0,86	0,88	0,9	0,92	0,94	0,96
Fogyasztás	312	311	310	307	304	300	295	290	283	275	266	260	254	250
LAMBDA	0,98	1	1,02	1,04	1,06	1,08	1,1	1,12	1,14	1,16	1,18	1,2	1,22	1,24
Fogyasztás	246	243	241	240	240	241	242	243	245	247	250	254	258	264

2.5.5. Effektív nyomaték szubmodell

Az effektív nyomaték változásának görbéje a 27. ábrán látható, a programozott pontos értékek a 6. táblázatban olvashatóak.



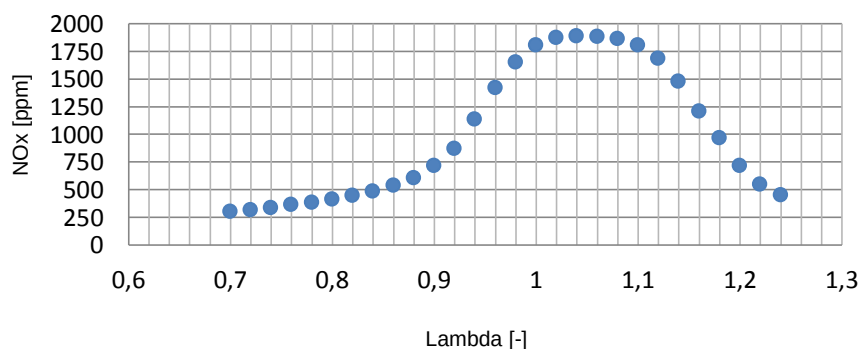
27. ábra: Az effektív nyomaték görbéje

6. táblázat: Az effektív nyomaték értékei

LAMBDA	0,7	0,72	0,74	0,76	0,78	0,8	0,82	0,84	0,86	0,88	0,9	0,92	0,94	0,96
Nyomaték	121	127	134	141	147	153	157	158	159	158	156	154	151	148
LAMBDA	0,98	1	1,02	1,04	1,06	1,08	1,1	1,12	1,14	1,16	1,18	1,2	1,22	1,24
Nyomaték	145	142	137	130	124	118	113	107	101	96	91	84	78	71

2.5.6. NO_x kibocsátás szubmodell

A NO_x szubmodell diagramja a 28. ábrán látható, a programozott értékeit a 7. táblázat tartalmazza.



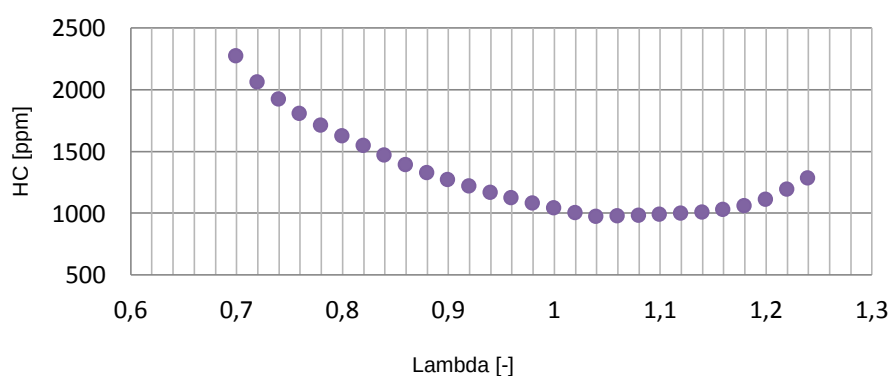
28. ábra: A NO_x kibocsátás görbéje

7. táblázat: A NO_x kibocsátás értékei

LAMBDA	0,7	0,72	0,74	0,76	0,78	0,8	0,82	0,84	0,86	0,88	0,9	0,92	0,94	0,96
NOx	302	319	337	363	385	415	448	486	540	605	717	874	1136	1422
LAMBDA	0,98	1	1,02	1,04	1,06	1,08	1,1	1,12	1,14	1,16	1,18	1,2	1,22	1,24
NOx	1652	1806	1874	1888	1886	1864	1806	1688	1479	1208	970	715	549	453

2.5.7. HC kibocsátás szubmodell

A HC kibocsátás szubmodelljébe programozott görbét a 29. ábra mutatja, a pontos értékek a 8. táblázatban olvashatóak.



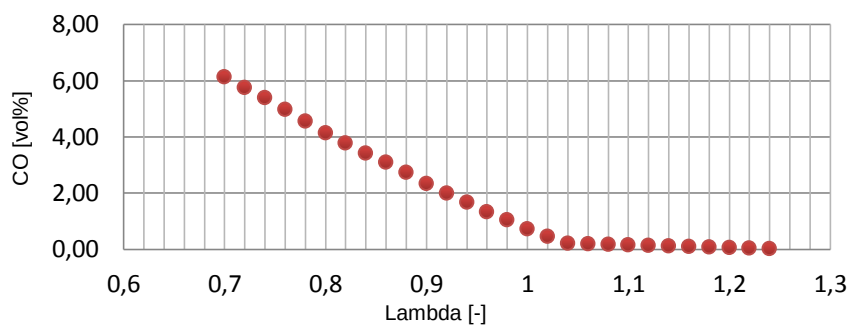
29. ábra: A HC kibocsátás görbéje

8. táblázat: A HC kibocsátás értékei

LAMBDA	0,7	0,72	0,74	0,76	0,78	0,8	0,82	0,84	0,86	0,88	0,9	0,92	0,94	0,96
HC	2271	2061	1921	1806	1708	1623	1547	1467	1390	1325	1268	1216	1164	1123
LAMBDA	0,98	1	1,02	1,04	1,06	1,08	1,1	1,12	1,14	1,16	1,18	1,2	1,22	1,24
HC	1081	1040	1001	972	974	980	988	998	1008	1029	1059	1108	1194	1281

2.5.8. CO kibocsátás szubmodell

A CO kibocsátás szubmodelljének görbét mutatja a 30. ábra, ennek pontos értékei a 9. táblázatban vannak.



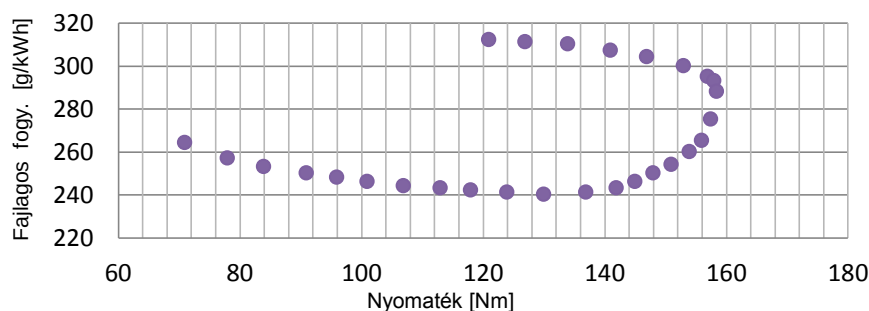
30. ábra: A CO kibocsátás görbéje

9. táblázat: A CO kibocsátás értékei

LAMBDA	0,7	0,72	0,74	0,76	0,78	0,8	0,82	0,84	0,86	0,88	0,9	0,92	0,94	0,96
CO	6,13	5,75	5,39	4,97	4,55	4,14	3,78	3,41	3,09	2,73	2,33	1,99	1,66	1,32
LAMBDA	0,98	1	1,02	1,04	1,06	1,08	1,1	1,12	1,14	1,16	1,18	1,2	1,22	1,24
CO	1,04	0,72	0,45	0,21	0,19	0,17	0,15	0,13	0,11	0,09	0,07	0,05	0,03	0,01

2.5.9. A motor C-görbéje a légviszony függő mérési mód esetében

A diákoknak szánt egyik mérési feladat a motor C-görbéjének megrajzolása. A lambda változtatásával feljegyzett effektív nyomaték és fajlagos fogyasztás értékek a 31. ábrán látható diagramot fogják adni. A 10. táblázat páronként tartalmazza az összetartozó értékeket.



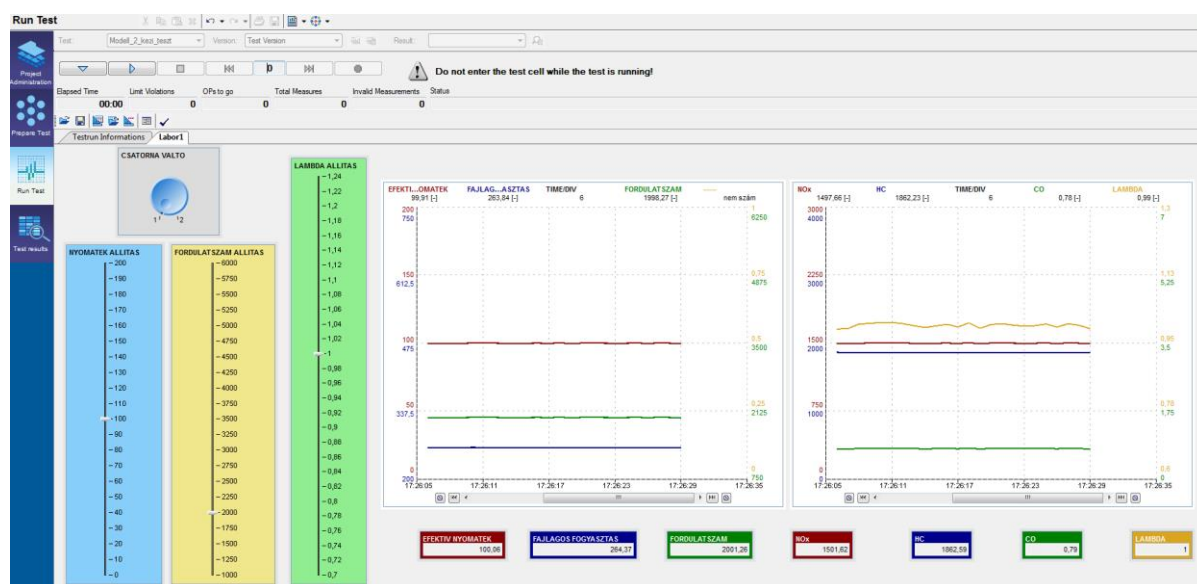
31. ábra: A motor C-görbéje

10. táblázat: A motor C-görbéjének értékei

LAMBDA	0,7	0,72	0,74	0,76	0,78	0,8	0,82	0,84	0,86	0,88	0,9	0,92	0,94	0,96
Nyomaték	121	127	134	141	147	153	157	158	159	158	156	154	151	148
Fogyasztás	312	311	310	307	304	300	295	290	283	275	266	260	254	250
LAMBDA	0,98	1	1,02	1,04	1,06	1,08	1,1	1,12	1,14	1,16	1,18	1,2	1,22	1,24
Nyomaték	145	142	137	130	124	118	113	107	101	96	91	84	78	71
Fogyasztás	246	243	241	240	240	241	242	243	245	247	250	254	258	264

2.5.10. A második modellből készült CAMEO mérés

A második modellhez szintén készítettünk automatikusan működő és kézzel állítható mérést. Működés közben a 32. ábrán látható módon néz ki a program felhasználói felülete. A két mérési mód közti bemeneti csatornaváltó forgatógombja a kézi mérés indításakor 1-es állásban van, miután a program a kívánt nyomaték, fordulatszám, lambda értékekre beáll, elkezdhető a mérés. A csatornák közti váltás bármikor megtörténhet, nem befolyásolja a modell működését. 1-es mérési módban a nyomaték és a fordulatszám csuszka állítható, 2-es módban a lambda csúszkát tudják a diákok használni az értékek megváltoztatására.



32. ábra: A második mérés, működés közben

Mindkét mérési módhoz külön készült automatikus mérés. Ezzel a diákok egy gombnyomás segítségével láthatják, hogyan zajlik a fékpadon egy valódi mérési folyamat, a CAMEO elkezdi lemérni az összes beállított pontban a kívánt értékeket és táblázatba rendezve egy kiértékelési fájlba menti el őket a későbbi felhasználáshoz. Leteszteltük mindkét mérési módnál a modellünket, a mért adatok a 33. ábrán láthatók. Három mért pont értékeit vizsgáltuk, ezek mindegyike megegyezett az általunk Matlab-ban programozottakkal, így a modellek hitelesek.

Information		Events		Statistics		Measurements																
Name	Test Version	FORDULATSZA		NYOMATEK ALLI		ElapsedTime	MeasurementTir	Layer	SubTestNumber	ModelVerificatio	FORDULATSZA		LAMBDA		FAJLAGOS FOI		NOx		EFEKTIV NYOM		CO	
System		modell_2_syste	modell_2_syste	RunTest	RunTest	RunTest	RunTest	RunTest	RunTest	RunTest	modell_2_syste	modell_2_syste	modell_2_syste	modell_2_syste	modell_2_syste	modell_2_syste	modell_2_syste	modell_2_syste	modell_2_syste	modell_2_syste	modell_2_syste	
Unit		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Type		Demand	Demand	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Mean	
1	0	3500	100	45.977	131111173056.8	2	-1	0	3499	0.9923	286.15	1487.3	99.973	0.7877								
2	0	1000	100	106.44	131111173157.3	2	-1	0	1000.8	0.99941	261.56	1477.1	100.04	0.7131								
3	0	5750	80	166.55	131111173257.4	2	-1	0	5746.9	0.98917	320.32	1055.1	79.964	1.1431								
4	0	5500	50	226.61	131111173357.5	2	-1	0	5496	0.98833	333.73	773.43	49.973	1.201								
5	0	1500	200	286.68	131111173457.5	2	-1	0	1499.9	0.9925	444.4	2506.7	199.94	0.7977								

33. ábra: Az első mód, automatikus mérés adatai

3. Fizikai motormodellek

3.1. Feladat

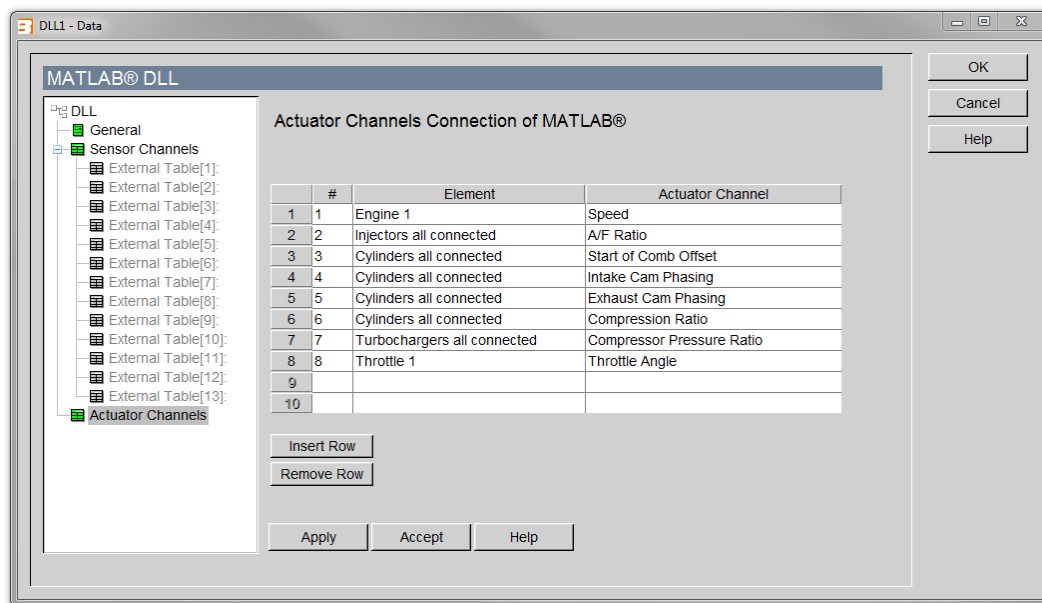
Fizikai motormodellek valós idejű szimulációjának megteremtése annak érdekében, hogy a hallgatók számítógépes környezet segítségével tudják a motor több paraméterét egyszerre vezérelni, vizsgálva a kimeneteken adott válaszokat. A szimuláció során mért adatoknak hűen tükröznie kell a valós motor viselkedését.

3.2. Célkitűzés

Az AVL BOOST-ban felépíthetők olyan motormodellek, melyek valós idejű szimulációjára lehetőség nyílik Matlab Simulink program segítségével. A két program összekötése lehetővé teszi, hogy a szoftverek egy összehangolt szimulációs háttérkörnyezetet biztosítsanak a hallgatók számára. A szoftvereket helyesen be kell állítani az ilyen jellegű felhasználásra, mivel eredetileg ilyen funkciót a szakemberek csak speciális esetekben alkalmaznak, olyan számítási feladatok elvégzésére, amire a BOOST nem, viszont a Simulink lehetőséget kínál. Cél ezt a funkciót kihasználni, segítségével létrehozni egy olyan szimulációt, ami alkalmas arra, hogy a hallgatók valós időben vezérelni tudják a számukra elkészített motormodelleket.

3.3. Felhasznált modellek

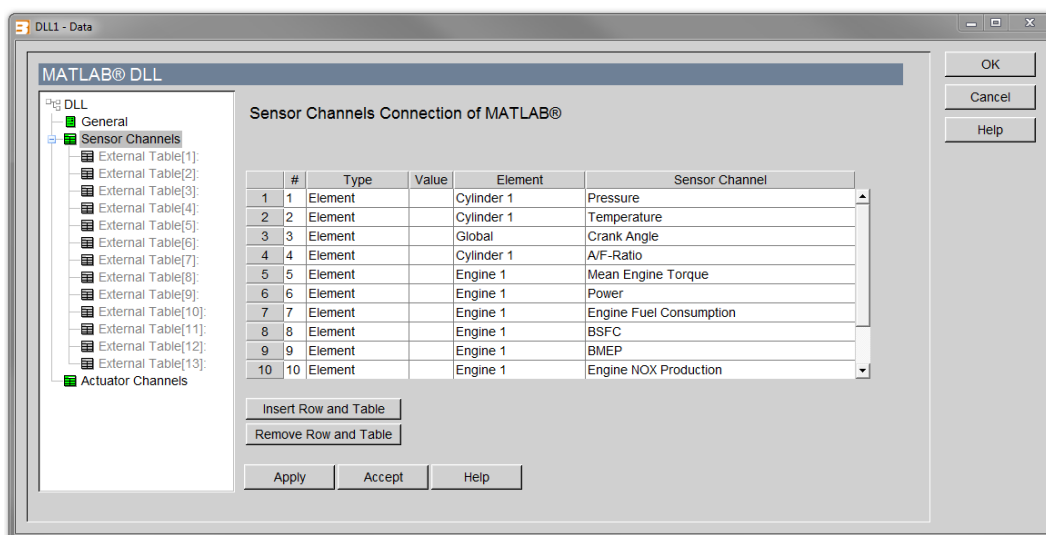
A virtuális környezet fizikai modellekkel, pontosabb eredményekhez vezet. Az ilyen modellekhez az AVL BOOST szoftver a legalkalmasabb. Rendelkezik a szükséges funkciókkal, kompatibilis a Matlab-bal, valamint a benne található validált példafájlok biztos alapot nyújtanak a virtuális környezetben történő szimulációhoz. A motormodellek paramétereinek vezérlésére az általunk elkészített MATLAB Simulink környezetet használjuk. A továbbiakban a felhasznált példafájlok funkcionalitásának bemutatása következik. A szoftverek helyes működéséhez létre kellett hozni minden BOOST modellen belül egy előre definiált Matlab DLL fájlt, melyben megadtuk az adott modellben mérni és vezérelni kívánt paramétereket. A bemeneti paramétereket aktuátorokként kell beállítani, ezen keresztül vezérelhetők a modellek Matlab-ban (34. ábra). Az ábrán látható nyolc aktuátor sorrendben a fordulatszámot, a lambdát, az égés kezdetének eltolását, a szívó- és kipufogó szelepek fáziseltolását, a kompresszió viszonyt, a turbófeltöltő kompresszió viszonyát, valamint a pillangószelep szöghelyzetét vezérlik.



34. ábra: Aktuátorok egy elkészült modell esetében

A vezérelhető paraméterek között vannak olyanok, melyek a motor valódi működtetését hivatottak bemutatni, mint például a benzinmotorok esetében a pillangószelep állása, de vannak konstrukciót változtató opciók is, mint például a kompresszió viszony. Erre azért van szükség, hogy a diákok megértsék, hogy milyen összetettek és milyen sok paramétertől függenek a motorban lezajló folyamatok.

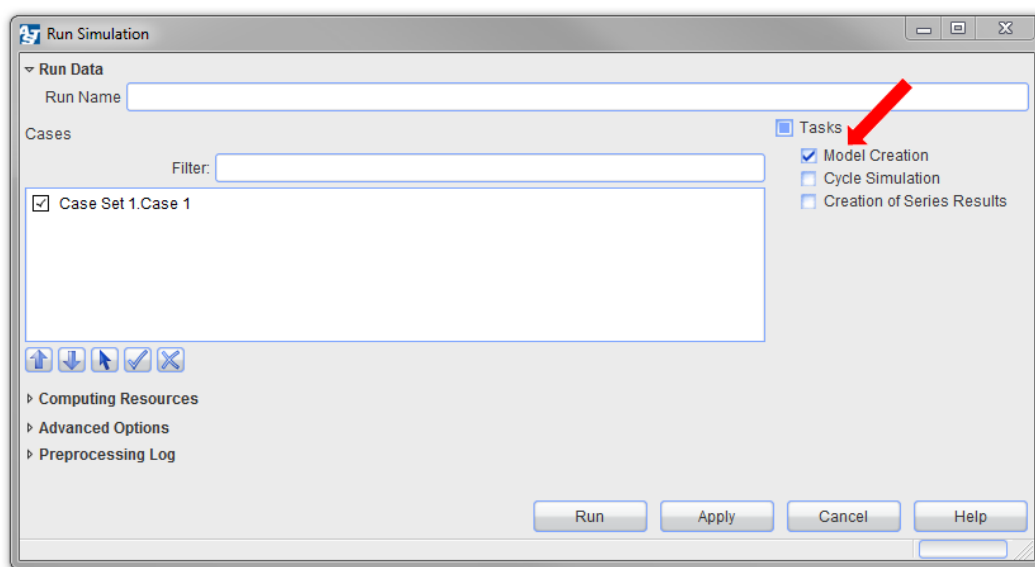
A megjeleníteni kívánt paramétereket szenzoroknak kell beállítani (35. ábra). Jelen esetben a tizenkettő beállított szenzor a nyomáslefutás, a hőmérséklet alakulása, a főtengely szöghelyzete, a lambda, a motor nyomatéka, teljesítménye, fogyasztása, fajlagos fogyasztása, effektív középnyomása, valamint a NO_x, CO és HC emisszió. A szenzorként és aktuátorként beállított paraméterek közül csak néhány eltérő, ezért ezek nem kerülnek bemutatásra.



35. ábra: Szenzorok egy elkészült modell esetében

A károsanyag-kibocsátás méréséhez az eredeti egyzónás égésfüggvényeket át kellett állítani kétzónásra. Ezzel a módszerrel lehetőség van az egyes modellek hengereiben keletkező káros anyagok megjelenítésére is.

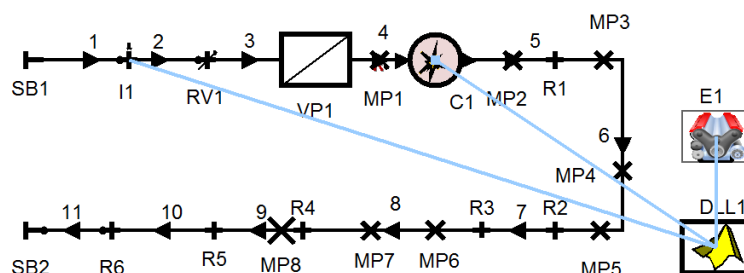
Az adott modellekhez tartozó megfelelő szenzorok és aktuátorok beállítása, valamint az adott paraméter vezérelhetőségének ellenőrzése után el kell indítani egy BOOST szimulációt, melynek során csak a modellt kell elkészíteni (36. ábra). Létrejön egy BST fájl, mely lényegében a BOOST motormodellt C-kód formájában tartalmazza. A Matlab-on belül található speciális blokk ezt a fájlt fogja a későbbiekben vezérelni.



36. ábra: BST fájl létrehozása

3.3.1. Egyhengeres kétütemű benzines motormodell

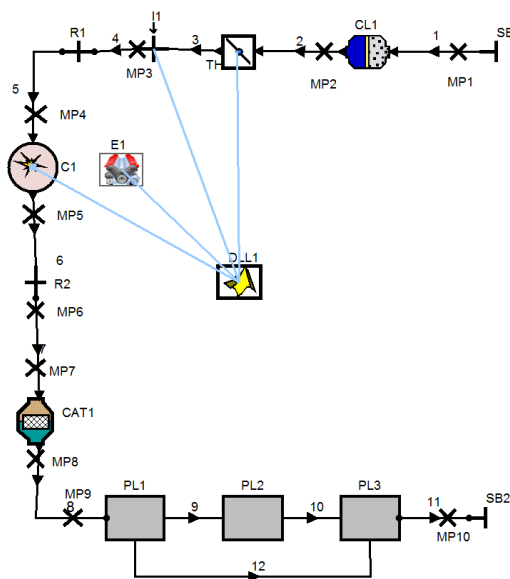
A legegyszerűbb motormodell egy egyhengeres kétütemű motornak a modellje (37. ábra). A motor hengerátmérője és lökete 54 mm. A vezérelni és monitorizálni kívánt paraméterek az injektor, a motor és a henger elemekben találhatóak, így ezeket össze kell kötni a Matlab DLL elemmel. A paraméterek megfelelő beállítása után a BOOST szimuláció lefuttatásával egy BST fájlt kell generálni belőle a további felhasználás érdekében.



37. ábra: Az első motormodell

3.3.2. Egyhengeres négyütemű benzines motormodell

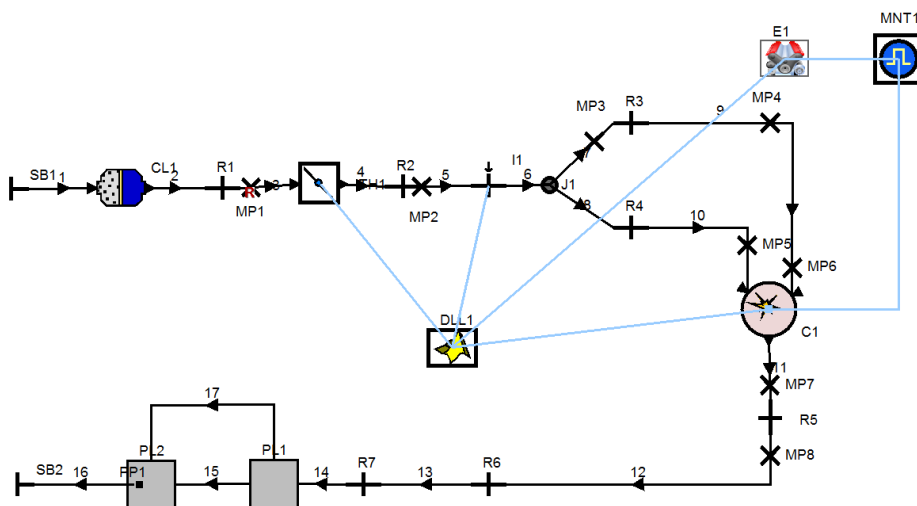
A második modell egy egyhengeres négyütemű benzines motorról készült (38. ábra). A hengerátmérője 84 mm, lökete 90 mm. Az általunk vezérelni és megjeleníteni kívánt egységek a pillangószelep, a henger, az injektor és a motor elemeken belül találhatók.



38. ábra: A második motormodell

3.3.3. Egyhengeres négyütemű motorkerékpár motormodell

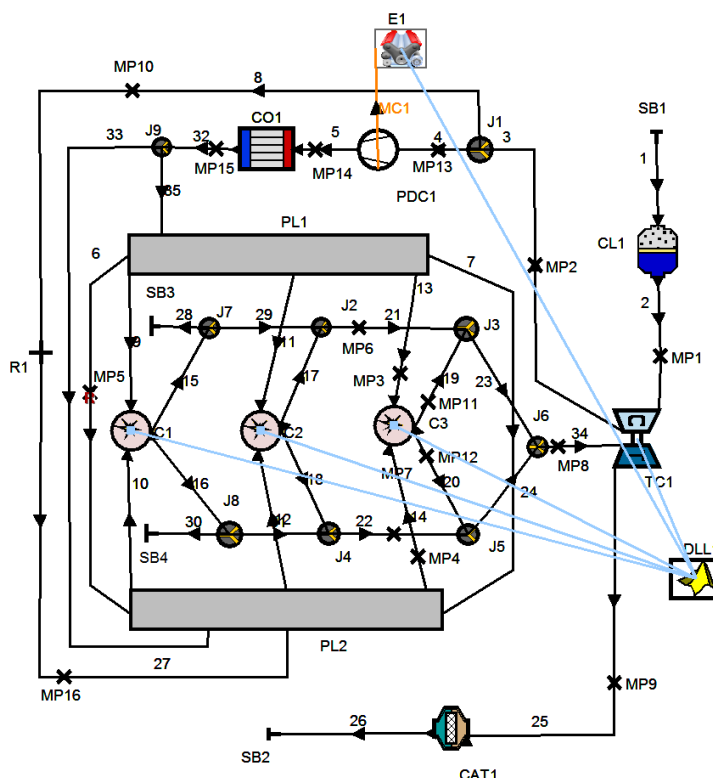
A harmadik modell egy motorkerékpárra szerelt egyhengeres négyütemű benzines motor modellje (39. ábra). A henger átmérője 46 mm, lökete 54 mm, két szívó és egy kipufogó szeleppel rendelkezik. Az általunk vezérelni és mérni kívánt egységek ebben az esetben is a pillangószelep, a henger, az injektor és a motor elemeken belül találhatók.



39. ábra: A harmadik motormodell

3.3.4. Háromhengeres kétütemű diesel motormodell

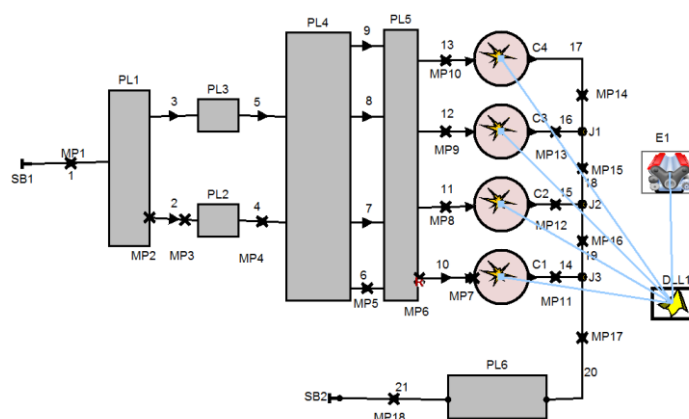
A negyedik modell egy háromhengeres kétütemű diesel motor modellje (40. ábra). A hengerek átmérője 71.5 mm, löketük 78.5 mm. A helyes működéshez mindhárom henger elemet, a motor és turbófeltöltő elemeket is össze kell kötni a Matlab DLL blokkal.



40. ábra: A negyedik motormodell

3.3.5. Négyhengeres ipari felhasználású diesel motormodell

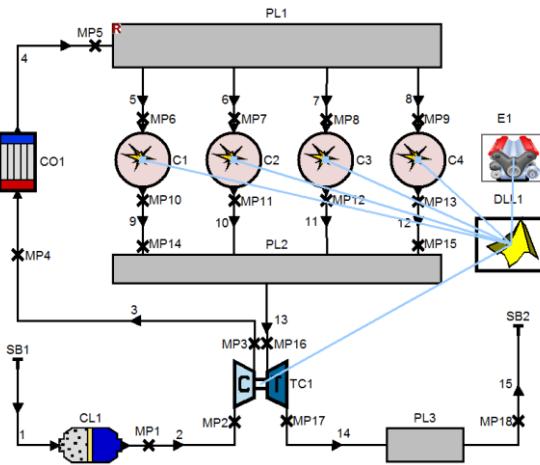
Az ötödik modell egy ipari felhasználású diesel motor modellje (41. ábra). A hengerek átmérője 75 mm, löketük 78 mm. Ebben az esetben a négy henger elemet és a motor elemet kell összekötni a Matlab DLL blokkal.



41. ábra: Az ötödik motormodell

3.3.6. Négyhengeres négyütemű diesel motormodell

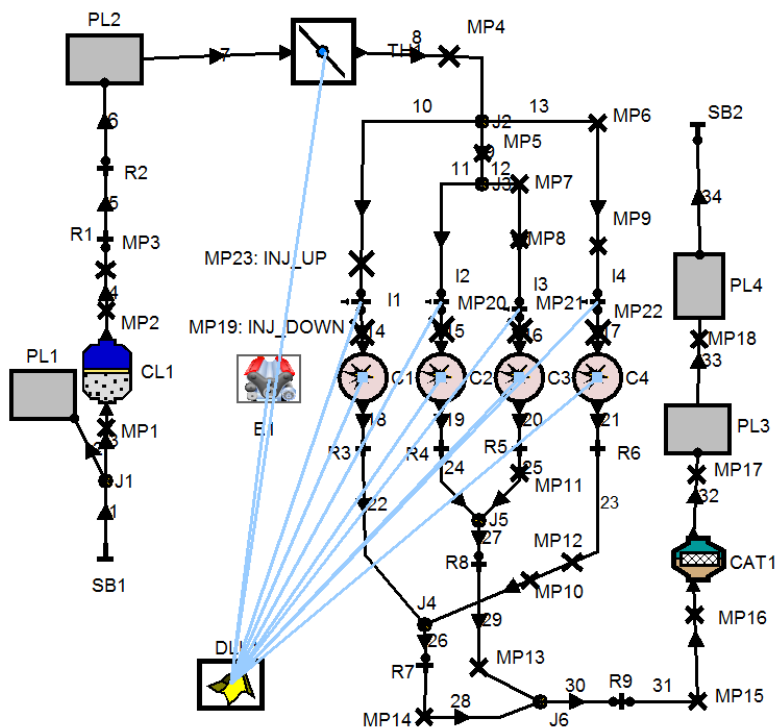
A hatodik modell egy négyhengeres négyütemű diesel motor modellje (42. ábra). A hengerek átmérője 90 mm, löketük 98 mm. A működéshez a négy henger elemet, a motor és turbófeltöltő elemeket kell összekötni a Matlab DLL blokkal.



42. ábra: A hatodik motormodell

3.3.7. Négyhengeres négyütemű benzín motormodell

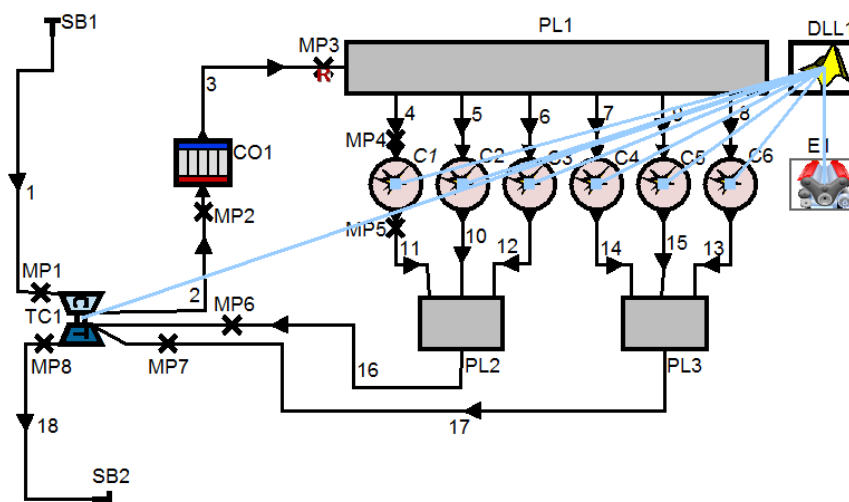
A hetedik modell egy négyhengeres négyütemű benzines motor modellje (43. ábra). A hengerek átmérője 86 mm, löketük 86 mm. Ebben az esetben a négy henger elem, a motor, az injektor és pillangószelep elemek vannak összekötve a Matlab DLL blokkal.



43. ábra: A hetedik motormodell

3.3.8. Hathengeres négyütemű diesel motormodell

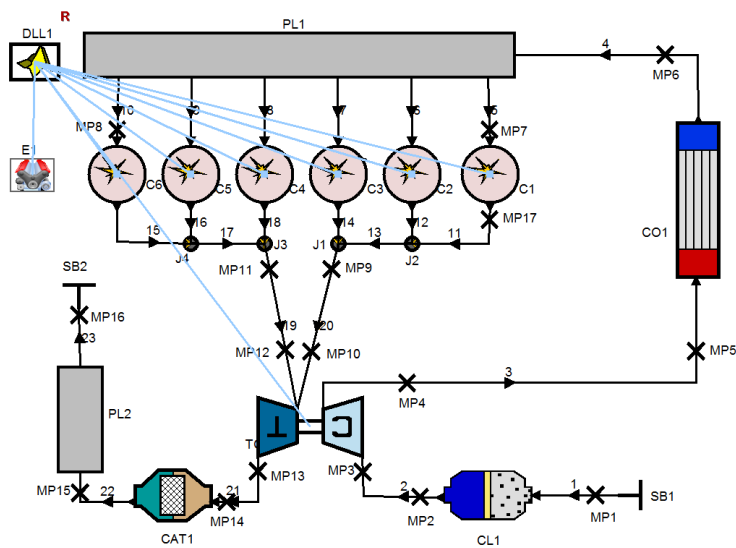
A nyolcadik motormodell egy hathengeres négyütemű diesel motornak a modellje (44. ábra). A motor hengerátmérője 100 mm, lökete 130 mm. A vezérelni és monitorizálni kívánt paraméterek a motor, a turbófeltöltő és a hat henger elemben találhatóak, így ezeket kell összekötni a Matlab DLL elemmel ebben az esetben.



44. ábra: A nyolcadik motormodell

3.3.9. Hathengeres négyütemű tehergépjármű motormodell

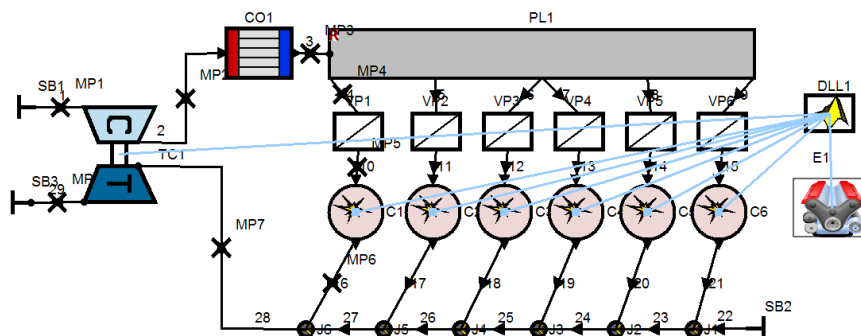
A kilencedik modell egy hathengeres négyütemű diesel haszongépjármű motor modellje (45. ábra). A hengerek átmérője 125 mm, löketük 148 mm. A Matlab-on keresztüli vezérlés és megjelenítés helyes működéshez a hat henger elemet, a motor és turbófeltöltő elemeket kell összekötni a Matlab DLL blokkal.



45. ábra: A kilencedik motormodell

3.3.10. Hathengeres kétütemű diesel motormodell

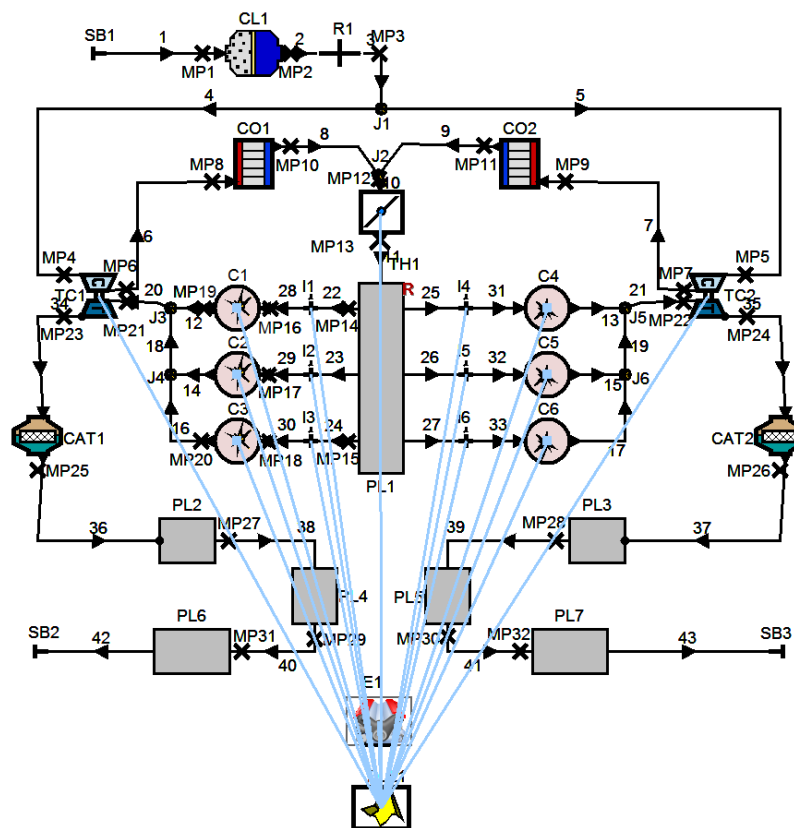
A tízedik modell egy hathengeres kétütemű diesel hajómotor modellje (46. ábra). A hengerek átmérője 900 mm, löketük 2100 mm. A helyes működéshez a hat henger, a motor és turbófeltöltő elemek vannak összekötve a Matlab DLL blokkal.



46. ábra: A tízedik motormodell

3.3.11. Hathengeres négyütemű benzín motormodell

A tizenegyedik modell egy hathengeres négyütemű benzines motor modellje (47. ábra). A hengerek átmérője 82 mm, löketük 84.6 mm. Ebben az esetben a hat henger, a motor, a pillangószelep, az injektor és turbófeltöltő elemeket kell összekötni a Matlab DLL blokkal.



47. ábra: A tizenegyedik motormodell

3.4. Matlab Simulink modellek

A BOOST-ban felépített motormodellek helyes működtetéséhez olyan Matlab Simulink modellekre van szükség, melyek teljesítik a felhasználói és a programkompatibilitási kritériumokat is. A kezelőfelületnek megegyezőnek kell lennie minden modell esetében, mivel a diákok a laborgyakorlaton véletlenszerűen kapnak egy-egy motortípust attól függően, hogy a rendelkezésre álló gépek közül melyikhez kerülnek. A laborgyakorlatok célja a motorikus folyamatok megértésének könnyebbé tétele, nem a Matlab szoftver kezelésének oktatása, ezért a modelleknek átláthatónak, egyszerűen kezelhetőnek kell lenniük, hogy a hallgatók a gyakorlat során akadálymentesen tudjanak méréseket, kiértékeléseket végezni rajtuk.

A BOOST segítségével generált BST fájlra az AVL mérnökei által programozott Simulink blokk segítségével hivatkozhatunk (48. ábra). Használatához hozzá kell adni a Matlab elérési útvonalához a BOOST megfelelő mappáját. A blokk segítségével megadható, hogy pontosan melyik BST fájl kerül felhasználásra az adott modellhez, illetve mennyi vezérelni kívánt szenzor, aktuátor van a modellen belül. A Simulink modellekben definiált szenzoroknak és aktuátoroknak a BOOST modellekkel megegyező sorrendben kell egymást követniük. Ennek elmulasztását a program nem érzékeli hibának, viszont az általunk állítani kívánt bemenő paraméter változtatása nem a megfelelő reakciót fogja kiváltani.

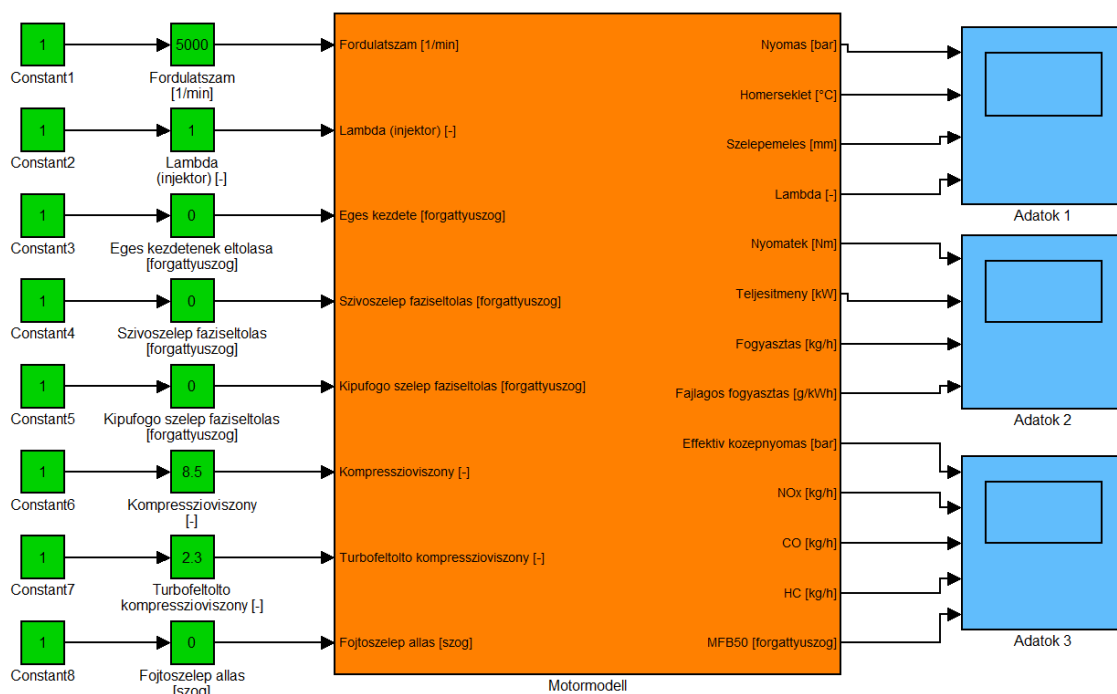


48. ábra: AVL BOOST blokk Matlab Simulink-ben

A számítógép laborban található tíz gép mindegyikéhez egy-egy motormodell tartozik a korábban már bemutatott, BOOST-ban felépített motormodellek közül. A benzines és a diesel motormodellek vezérléséhez és monitorizálásához készült Simulink modellek csupán néhány helyen térnek el egymástól. A benzineseknél a valóságnak megfelelő mennyiségi szabályozáshoz, a fojtószelep vezérlésére szolgáló bemenet található meg. A dieselknél a minőségi szabályozást biztosító befecskendezett üzemanyag mennyiség állítható.

A 49. ábra egy benzinmotor Simulink modelljét mutatja, a hallgatók a fájlok megnyitásakor ezzel találkoznak. Az ábra közepén látható, motormodell névvel ellátott blokk valójában egy szubmodell (subsystem), mely további szükséges blokkokat, valamint a 48. ábrán látható BST fájlra hivatkozó blokkot is tartalmazza. Ez a modellstruktúra lehetővé teszi,

hogy a bonyolult, a laborgyakorlat szempontjából nem releváns tartalmat a szubmodellen belül elrejtjük, ezáltal a diákok a tényleges feladatra, a mérés elvégzésére koncentrálhassanak. A főmodell felépítése megfelel a szimulációs szakemberek által megszokott balról jobbra irányuló struktúrával. A bemenő paraméterek, jelen esetben aktuátorok, az ábra bal oldalán láthatók zöld színnel jelölve. Középen a szubmodell, jobb oldalon a kiértékeléshez szükséges kék színnel jelölt grafikus megjelenítő elemek kaptak helyet.

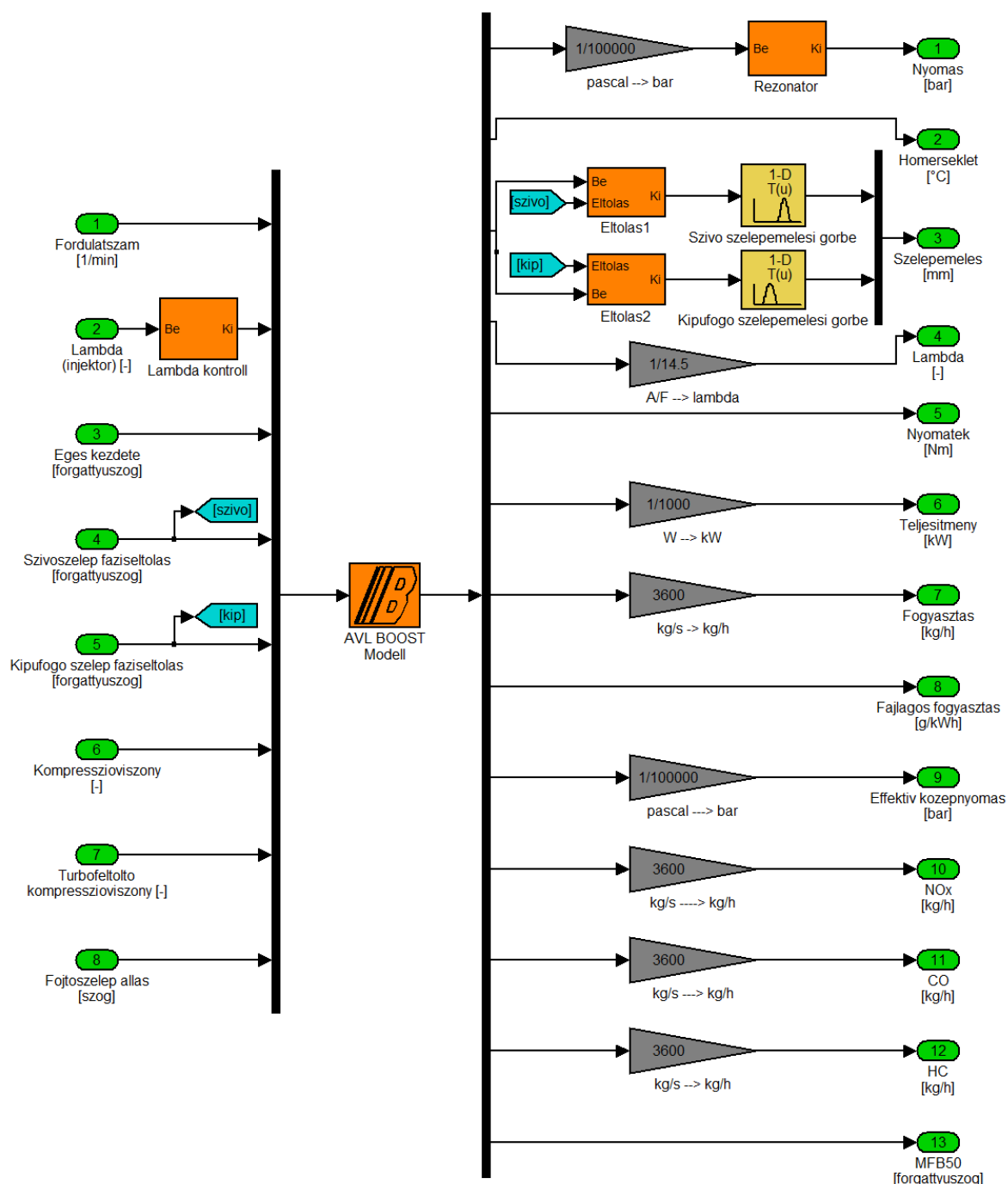


49. ábra: Matlab Simulink főmodell

A benzines modellekben található szubmodellek egyikét az 50. ábra mutatja be. A főmodellben beállított aktuátor adatok a baloldalon található bemeneti csatornákon érkeznek egy gyűjtő elem keresztül a középen található AVL BOOST Modell nevű blokkhoz. A gyűjtő elem szerepe, hogy az egyes bemeneti értékeket vektor formájában juttassa el a blokk bemenetéhez. A kimeneti paraméterek szintén vektor formájában jelennek meg, melyet egy elosztó elem segítségével juttatunk el a kimeneti pontokra.

A Simulink blokkok eredetileg angol nyelvűek, azonban a modellek működésének egyszerűbb megértéséhez minden blokk magyar elnevezéssel lett ellátva. A modellen belül található szürke színű blokkok a mértékegységek helyes megjelenítését teszik lehetővé, de helyet kaptak a modellen olyan plusz funkciókat ellátó blokkok is, melyek a motor kopogását, tönkremenetelét illetve lefulladását idézik elő. Ezek segítségével a diákok még közelebb juthatnak a motorok viselkedésének elsajátításához.

A szelepek nyitási görbéinek megjelenítésére is külön részekységek vannak a motormodellen belül. A görbék segítségével a változtatható szelepvezérlés bemutatására, a fáziseltolások szemléltetésére tökéletesen megfelelő az általunk alkalmazott módszer.

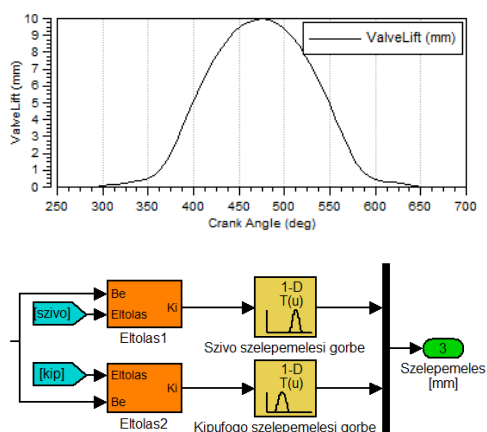


50. ábra: Matlab Simulink szubmodell

A következő fejezetekben a motormodellen belül található szubmodellek kialakításáról, működéséről lesz szó. Részletesen bemutatjuk az egyes blokkok felépítését, valamint szerepét a modellen belül.

3.4.1. A szelepemelési görbék megjelenítése

A szelepemelési görbék közvetlen megjelenítésére nincs lehetőség. Ennek oka, hogy a Matlab-ban beállított aktuátorok a BOOST-ban egy-egy paramétert változtatnak csak meg, a szelepemelési görbék viszont a főtengely szöghelyzetének függvényében definiált pontthalmazokból épülnek fel (51. ábra felül), a pontthalmazok közvetlen változtatása egyszerre nem lehetséges. A főtengely szöghelyzetének szenzoron keresztüli megjelenítése azonban lehetséges, így ezt kihasználva, néhány blokk Matlab-ba építésével megjeleníthetők a szelepemelési görbék is (51. ábra alul). Ehhez a BOOST-ban definiált görbesereget exportálnunk kell DAT fájl formájában, ami ezután könnyedén beimportálható a Matlab-ba változóként. A változónak két adatsora van, egyik a főtengely szöghelyzete $0^\circ - 720^\circ$ -ig, másik az adott szöghelyzethez tartozó szelepemelés mm-ben megadva. A két adatsor segítségével definiálható az ábra tetején található görbével azonos a Matlab-ban is, ezek a görbék a modellen belül a sárga színnel jelölt blokkokban vannak megadva „szívó-, kipufogó szelepemelési görbe” névvel. Ezek megadása azonban nem elegendő, mivel a szelepemelési görbék ebben az esetben csak konstans módon jelennének meg a főtengely mindenkori szöghelyzetének függvényében.

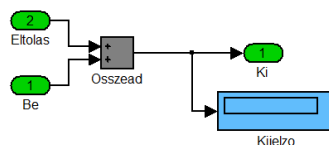


51. ábra: Szelepemelési görbe BOOST-ban (fent) és Matlab Simulink-ben (lent)

A változtatható szelepvezérlés helyes bemutatására szükség van egy további blokkra, melynek segítségével a szelepek fáziseltolása is helyesen megjelenik a Matlab-ban kirajzolt grafikonokon. Ez a narancssárga színnel ellátott blokk az „eltolás” nevet kapta, beépítése a szívó és kipufogó oldali szelepemelési görbékhez is szükséges. A szubmodellen belül található blokkok (52. ábra) a modell helyes működéséhez elengedhetetlenek.

A bejövő jelhez, mely esetünkben a főtengely szöghelyzete, hozzáadódik a fáziseltolás. Ezt a jelet közvetlenül az általunk szabályozott aktuátorról kapja a szubmodell, a két érték

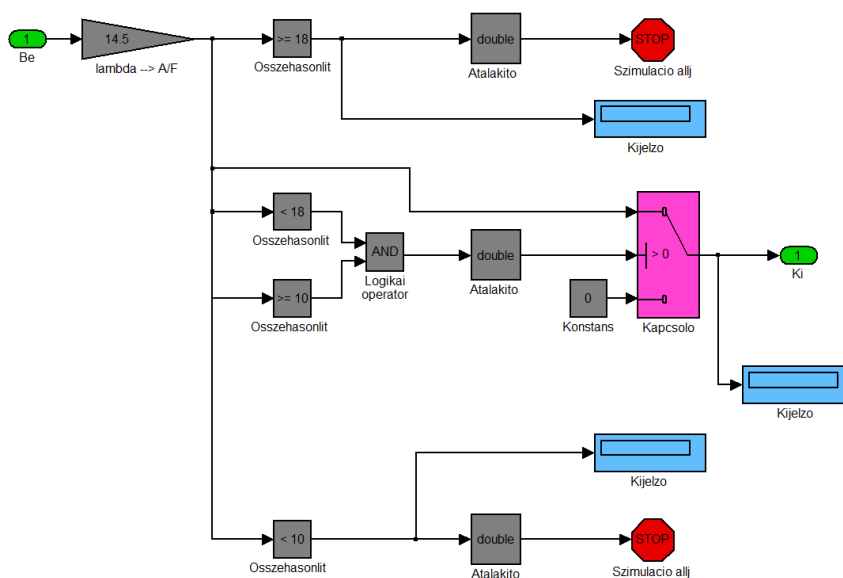
összege végül a kimeneten kerül a motormodellbe, majd azon át a főmodellbe. Ezzel a módszerrel már minden esetben helyesen jeleníthetők meg a szelepemelési görbék.



52. ábra: Eltolás szubmodell

3.4.2. A motor lefulladása

A motor lefulladásának szimulálására további szubmodell felépítésére volt szükség, mivel a BOOST modell reakciója nagyon lassú, több ciklus elteltével áll csak le a szimuláció. Ez a mi esetünkben nem engedhető meg, mivel a valós idejű ciklus-szimuláció még a BOOST-ban lelassításra került, gyakran az eredeti sebességének tizedére, hogy a hallgatók láthassák a görbék alakulását. A lassítási funkció nagy előnye magában rejti azt a hátrány, hogy irreálisan sokáig képes a motor az esetleges extrém lambda értékeken is üzemelni, ezért beépítésre került egy „lambda kontroll” nevű szubmodell, mely a benzines motorok szűk működési tartományát hivatott kontrollálni (53. ábra).



53. ábra: Lambda kontroll szubmodell

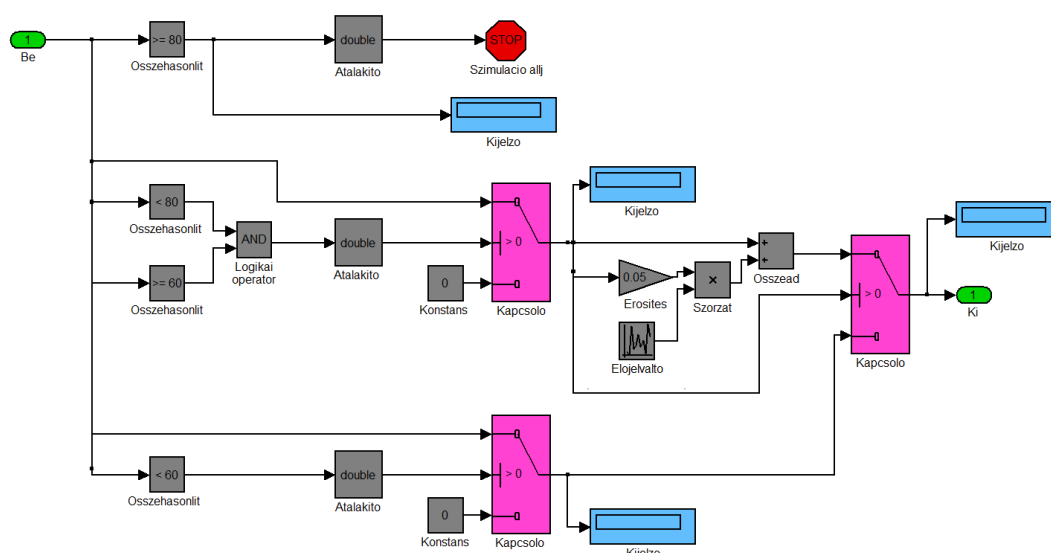
A szubmodellbe érkező lambda érték szorozódik az A/F aránnyal, mivel a BOOST-on belül ennek állítására van csak lehetőség. Az A/F arány nem más, mint a levegő-üzemanyag keverékének aránya. A tökéletes égéshez szükséges keverési arány, a sztöchiometrikus egyensúly esetén a $\lambda = 1$ értéket mutat, az ehhez szükséges levegő mennyiség üzemanyagonként eltérő lehet, esetünkben ez a 14,5/1 arány lesz.

Az 14,5-ös erősítés után a jel három további úton mehet tovább, attól függően, hogy az összehasonlító blokkok melyike engedi tovább. A túl nagy A/F értékek esetén, mikor a keverék túl szegény az égés beindulásához, a szimuláció leáll. Ugyanez történik abban az esetben is, ha a keverékben már túl sok az üzemanyag ahhoz, hogy az begyulladjasson. A motor helyes üzemi tartományán belüli A/F értéket a szubmodell továbbbengedi a kimenetre, ezáltal a szimuláció zavartalanul folytatódik. A helyes működéshez kellenek jelátalakítók, egy logikai operátor és egy kapcsoló elem is. A kijelzők segítségével látható, hogy mikor melyik úton halad a jel. Az aktív kijelzőn az A/F érték, az inaktív kijelzőkön 0 jelenik meg.

A szubmodell, ami csak a benzines motormodellekben található meg, az injektor szerepét betöltő aktuátor bemenő ágára lett bekötve (50. ábra), mert a hengerben mért A/F arány csak néhány ciklus után áll be az üzemi értékekre. Ha a kimenetre kötjük, a szimuláció nem indul el, hiszen az A/F értékek a szimuláció kezdetén még az üzemi tartományon kívül esnek. Megközelítőleg 5 ciklus után állnak be a megfelelő értékekre. A bemeneti ágra kötve a szubmodell tökéletes összhangban van a motormodellel.

3.4.3. A motor kopogása

A kopogás előidézésére is készült egy szubmodell, mely a „rezonátor” nevet kapta (54. ábra). Felépítését tekintve hasonló a lambda állítására szolgáló szubmodellhez, működése azonban eltérő. A motor üzemi tartományán belüli nyomásértékeket változtatás nélkül továbbbengedi az alsó ágon, az ettől eltérő értékekre viszont reakcióba lépnek a másik két ágba beépített blokkok.



54. ábra: Rezonátor szubmodell

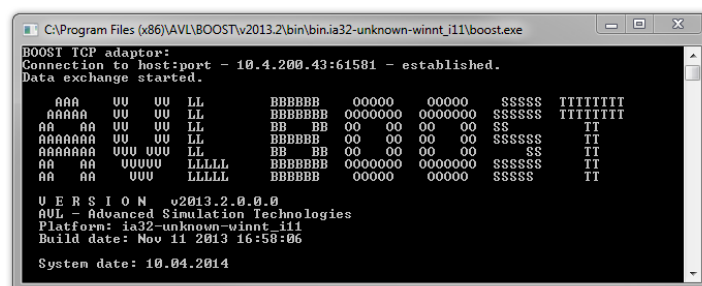
Egy bizonyos nyomásérték tartományban, ami modellenként eltérő, a szubmodell a valódi motoroknál megfigyelhető kopogást idézi elő. Egy előjelváltó és egy erősítő segítségével a nyomásérték a megadott tartományban a kopogásnál is megfigyelhető alakot vesz fel, a motorban hirtelen lezajló nyomáslengéseket tökéletesen reprodukálja. Amennyiben a nyomásérték meghaladja a motor által elviselhető értékhatárt, a szimuláció leáll, jelezve, hogy a motor tönkrement.

A szubmodell az első henger nyomását mutató szenzor elé került beépítésre (50. ábra). A lambda kontrollálására szolgáló szubmodellel ellentétben ennek a kimeneti ágon is zavartalan a működése, mivel a kiindulási paraméterek normál üzemi tartományon belül vannak a szimuláció kezdetekor.

3.5. A szimuláció futtatása

A modellek sikeres megalkotása után össze kell illeszteni az egyes BOOST modellekkel. A kiinduló paramétereket úgy kell beállítani, ahogy azok az eredeti modellben is be lettek állítva, elkerülve ezzel azt, hogy a modellek az eredeti motoroktól teljesen különböző üzemi tartományban legyenek. A motormodelleken belül található AVL BOOST modellben be kell állítani, hogy melyik BST fájlt fogjuk használni, ha ezt rosszul tesszük meg, a szimuláció hibaüzenettel indul, nem fut le.

A Matlab indítása előtt létre kell hoznunk a kapcsolatot a két szoftver között. A BOOST-al való kommunikáció az AVL mérnökei által programozott kód (script) segítségével jön létre. Parancssoron belül meghívjuk a scriptet, majd innen elindítva a Matlab-ot a szoftverek közti kapcsolat létrejön. A scriptre való hivatkozás átlag számítógép felhasználónak bonyolult folyamat, azonban ezt megkönnyítve egy egyszerű TXT fájlba elhelyeztük azt a három sort, amit egymás után a parancssorban megadva a script könnyedén meghívható. A sikeres beállítások után a szimuláció indításakor elindul a script (55. ábra), mely a szimuláció során folyamatosan tájékoztat bennünket a ciklusok lefutásáról, valamint a modellek esetleges korlátainak túllépéséről.



```

C:\Program Files (x86)\AVL\BOOST\v2013.2\bin\ia32-unknown-winnt_i11\boost.exe
BOOST TCP adaptor:
Connection to host:port - 10.4.200.43:61581 - established.
Data exchange started.

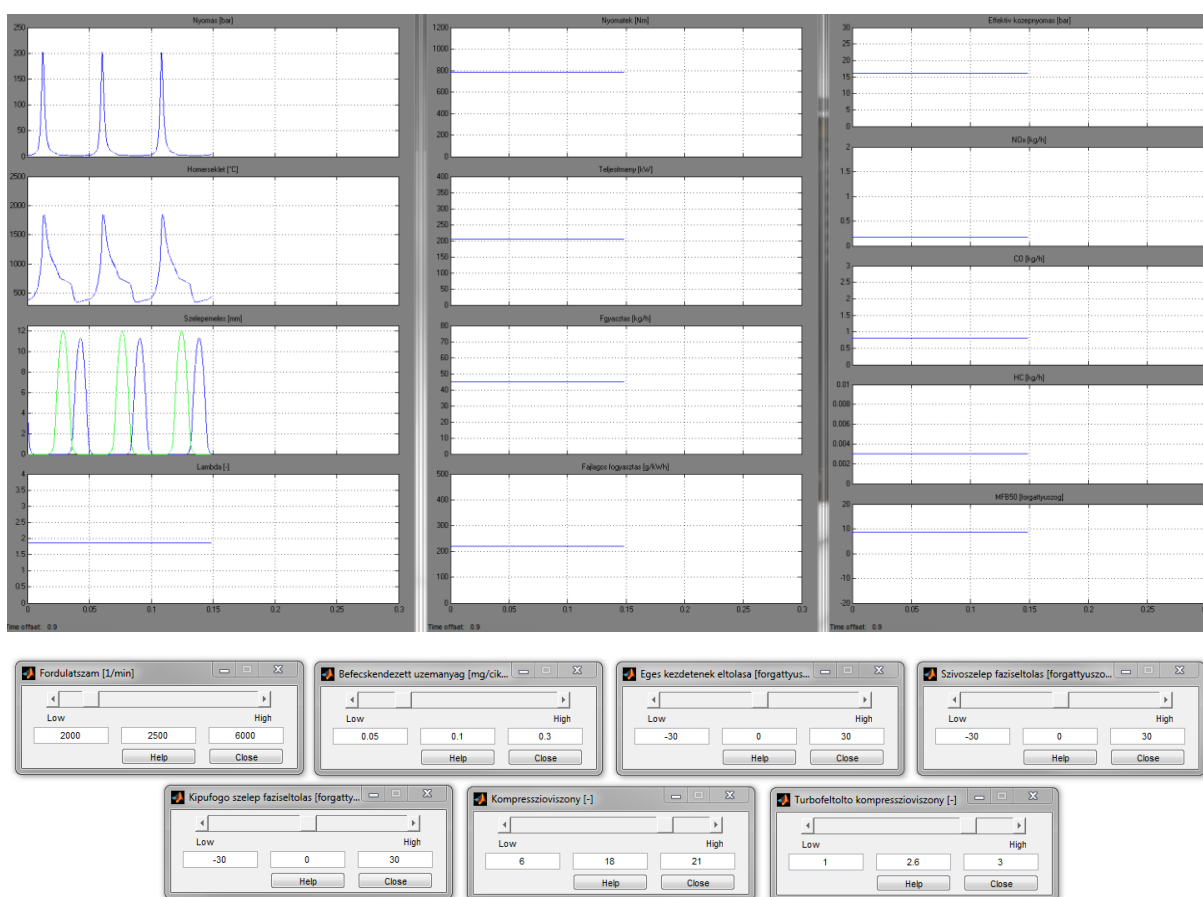
  AAA  UU  UU  LL  EEEEE  00000  00000  SSSSS  TTTTTT
  AAAAA UU  UU  LL  EEEEE  0000000 0000000 SSSSS  TTTTTT
  AA  AA UU  UU  LL  BB  BB  00  00  00  00  SS  TT
  AAAAAA UU  UU  LL  EEEEE  00  00  00  00 SSSSS  TT
  AAAAAA UUU UU  LL  BB  BB  00  00  00  00 SS  TT
  AA  AA UUUUU LLLLL  EEEEE  0000000 0000000 SSSSS  TT
  AA  AA UUUU  LLLLL  EEEEE  000000  00000  SSSSS  TT

V E R S I O N  v2013.2.0.0.0
AVL - Advanced Simulation Technologies
Platform: ia32-unknown-winnt_i11
Build date: Nov 11 2013 16:58:06
System date: 10.04.2014
  
```

55. ábra: AVL BOOST Script

A diákoknak lehetőségük nyílik arra, hogy az előre beállított programfelületen vezérelni tudják az összes bemenő paramétert és vizsgálják azok hatását a kimeneteken (56. ábra). A bemenetek állítására az ábra alján látható csúszkák segítségével van lehetőség. Ezek száma a motor bemeneti paramétereinek számával egyezik. A kimenetek az idő függvényében vannak megjelenítve a Matlab-ba épített kijelző (scope) elemek segítségével.

A görbék valós időben reagálnak a bemeneti paraméterek változtatására, létrehozva ezzel annak lehetőségét, hogy a motor komplex működését bemutathassuk. Az elkészült programfelület minden modellnél minimális eltéréssel megegyezik. A különbséget csak 1-1 paraméter jelenti, így ezek egyenként nem kerülnek bemutatásra, mivel a modellek működéséről plusz információkat nem tartalmaznak.



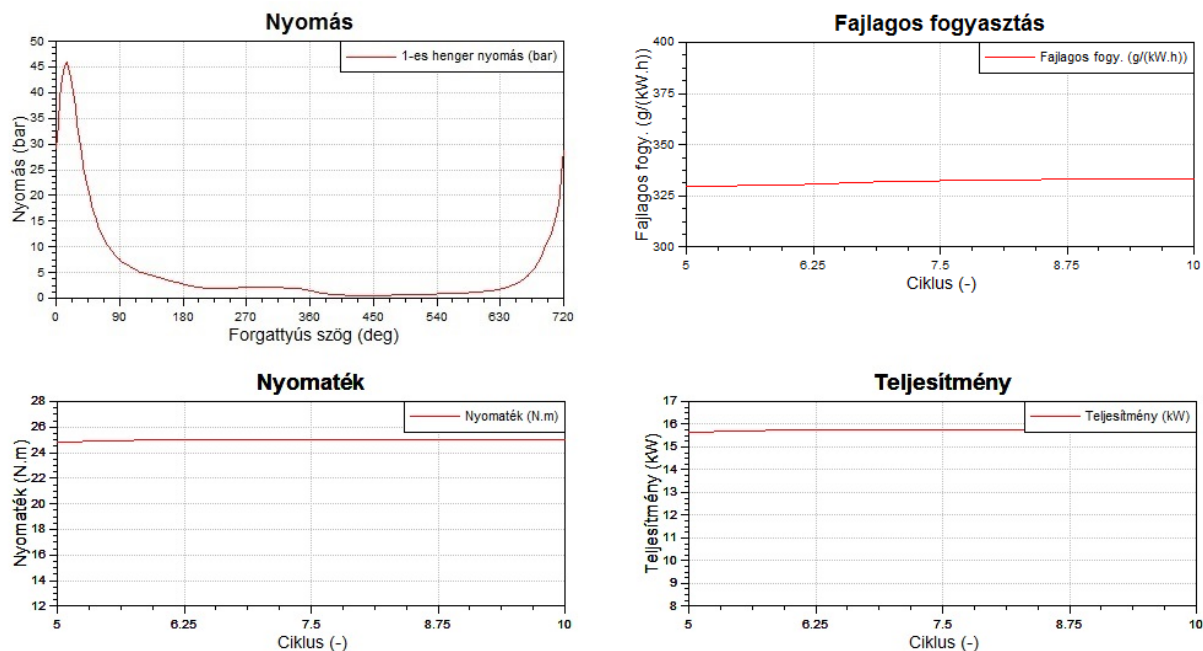
56. ábra: Szimuláció kezelői felülete Matlab Simulink-ben

3.6. A modellek validálása

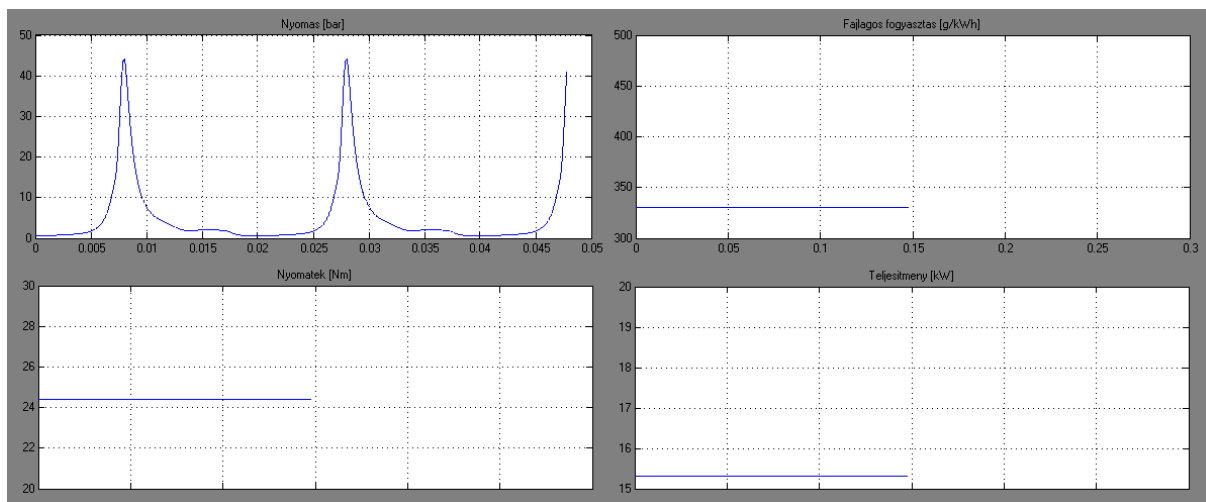
A modellek helyes működését ebben az esetben is validációval kell igazolni. A BOOST-ban lefuttatott és kiértékelt ciklus-szimulációk görbéinek megegyező értékeket kell adniuk az általunk Matlab-ban vezérelt modellel. A szimulációs értékek az első öt ciklusban még nem adnak kielégítő értéket, idő kell, míg ezek beállnak a megfelelő értékekre. A

szakemberek sem szokták ezeket figyelembe venni, így az ötödik ciklustól vannak az ábrák itt is megjelenítve (57. és 58. ábra).

Az ábrákon látható görbék közti minimális eltérést a két lefuttatott szimuláció között a hengerben definiált A/F arányok közti minimális különbség adja. Amennyiben ez a két érték is megegyezik, a két görbesereg tökéletesen fedi egymást.



57. ábra: Eredmények BOOST-ban



58. ábra: Eredmények Matlab-ban

Az általunk véletlenszerűen választott értékek, görbék, mértékegységek a tesztek során tökéletesen megegyeztek a BOOST-ban kiértékelt adatok és a Matlab-ban futtatott szimulációk között, ezért a modellek hitelesnek, oktatási célokra tökéletesnek tekinthetők.

4. Összefoglalás

Összefoglalásként elmondható, hogy a több állomást önmagába foglaló projektünk elérte célját. Sikerült bevezetni mind az alap, mind a mesterszintű járműmérnök képzésbe a virtuális motorvizsgáló laborgyakorlatot, ezáltal megismertette a hallgatókat olyan szoftverekkel, eljárásokkal és gyakorlati ismeretekkel, amelyekkel Magyarországon csak ritkán találkozhattak a mérnökök és az egyetemi oktatásban sem terjedt még el.

Dolgozatunknak két fő pillére a fékpadi mérési módszerek szélesebb körben való megismertetése a mérőkhallgatókkal, beleértve a szoftverek kezelését, az elvégzendő vizsgálatokat és az önálló mérési feladatokat is; valamint a motoros ismeretek bővítése, mely során a hallgatók fejlett szimulációs eljárások útján ismerhetik meg a motorok jellemző viselkedését, és képet kaphatnak arról, hogy különböző üzemállapotokban milyen értékek fogadhatók el egy mérés eredményeként.

Piackutatásunk sikerrel járt, hiszen megtaláltuk a céljainkhoz legalkalmasabb eszközöket, eljutottunk a grazi székhelyű AVL List GmbH-hoz, ahol egy négy hetes továbbképzésben is részesültünk, hogy megismerhessük a programok működését. Partnerkapcsolatot építettünk ki a vállalat CAMEO és fOX és csapatával, melynek jegyében licenceket kaptunk a Belső Égésű Motorok Tanszék számítógépes laborjában lévő tíz munkaállomásra. A CAMEO-t azért ítéltük különösen hasznosnak, mert egyszerre képes fékpaddvezérlő szoftverként funkcionálni, felhasználni virtuális környezetet bemenetként és az ehhez szükséges virtuális környezet előkészítésében is segítséget nyújt, matematikai modellek formájában. Projektünk másik részéhez, a fizikai modellekkel történő szimulációhoz az ugyancsak AVL által készített BOOST szoftvert hívtuk segítségül, mely a tanszéken már használatos különféle motortípusok viselkedésének szimulálására. A munka közben sikeres partnerkapcsolatot építettünk ki AVL BOOST grazi fejlesztő csapatával is.

Az elmúlt félév végén sikeresen megtartottuk az első „virtuális fékpadd” gyakorlatot, mely során a hallgatók a CAMEO szoftveren keresztül vezérelhették a motormodellek különböző paramétereit, tényleges fékpadd nélkül, a közeljövőben pedig megtartjuk második gyakorlatunkat a BSc szakos járműmérnök hallgatóknak, ahol fizikai modelleken vizsgálhatják a motorok jellemzőit. A következő félévtől a laborgyakorlat részét képezi a Belső Égésű Motorok Tanszék járműmérnök képzésének.

5. Kitekintés

Az eddig megtartott gyakorlat során, az általunk kidolgozott módszerek és eszközök a hallgatók körében rendkívül pozitív fogadtatásban részesültek. A járműmérnöki mellett a gépészmérnöki képzésben tanulók számára is beiktatjuk az ütemtervbe a számítógépes laborgyakorlatot. A jövőben, a végzett mérnökhallgatók tudása szélesebb spektrumú lesz, gyakorlati érzékük és mérnöki gondolkodásmódjuk továbbfejlődik. Jelen félév végén fogjuk megtartani a fizikai motormodelleket bemutató laborgyakorlatot, ahol a motorikus folyamatok közelebbről való megismertetése a célunk a hallgatókkal. Ettől a félévtől kezdve szeretnénk a „virtuális fékpad” gyakorlatot a Széchenyi István Egyetemen minden olyan mérnökképzés részévé tenni, ahol motoros ismereteket oktatnak. Úgy gondoljuk, hogy az általunk kidolgozott módszerrel a belső égésű motorok oktatása gyakorlatiasabbá, izgalmasabbá tehető és olyan helyeken, intézményekben is meghonosítható, ahol az korábban eszközhiány miatt nem volt lehetséges.

A jövőben a virtuális motorvizsgáló labor funkcionalitását bővítjük. A jelen dolgozat írásának pillanatában még nem elérhető AVL BOOST RT (Real-Time) 2014-es verziójú szoftverében már elérhető lesz a CAMEO-val való együttműködéshez szükséges interface. Az ott elkészült modelleket tehát közvetlenül a fékpad szoftver felületén lehet megfigyelni, működésükbe valós időben akár be is avatkozhatunk. Ezzel lehetőség nyílik arra, hogy a járműmérnök hallgatók más tanórák keretein belül elsajátított BOOST ismereteikkel felépített motormodelleket azonnal ki is próbálhassák fékpad környezetben. Ez hasznos lehet a BOOST-ot használó SZENGINE Formula Student versenycsapatának is, mivel ők is régóta használják már a programot motorjuk továbbfejlesztésére.

Szeretnénk minden, Széchenyi István Egyetemen tanuló mérnökhallgató számára elérhetővé tenni a szimulátort és ezzel hozzájárulni az intézményünkben kikerülő mérnökök tudásszintjének növeléséhez a virtuális laborgyakorlatok adta lehetőségekkel (59. ábra).



59. ábra: A hagyományos (balra) és az új oktatási módszer (jobbra)

Irodalomjegyzék

- [1] AVL List GmbH
Basics & Beyond the Basics AVL CAMEO 2013
AVL List GmbH, Graz, 2013
- [2] AVL List GmbH
BOOST dokumentáció
AVL List GmbH, Graz, 2013
- [3] AVL List GmbH
CAMEO tanfolyam oktatási anyag
AVL List GmbH, Graz, 2013
- [4] Basshuysen, R. van
Ottomotoren mit Direkteinspritzung 2. Auflage
Vieweg+Teubner GWV Fachverlage GmbH, Wiesbaden, 2008
3. Gemischbildung und Verbrennungsverfahren
- [5] Dipl. –Math. Beine, M.
The Successes and Challenges of Autocoding
ATZelektronik 2008/02, (p. 38-42.)
- [6] Bosch Műszaki útmutató füzetek
Motorelektronika
Maróti kiadó
- [7] Bosch Műszaki útmutató füzetek
Otto-motorok kipufogógáz technikája
Maróti kiadó
- [8] Dobes, T. – Kokalj G. – Rotschädl, R. – Lick, P.
Trendsetting Management of Control Unit Data
ATZelektronik 2007/03, (p. 24-26.)
- [9] Heywood, J. B.
Internal Combustion Engine Fundamentals
McGraw –Hill, Inc, 1988
2. Engine Design and Operating Parameters
11. Pollutant Formation and Control
15. Engine Operating Characteristics

- [10] Kalmár I. – Dr. Stukovszky Zs.
Belsőégésű motorok folyamatai
Műegyetemi kiadó, Budapest, 1998
- [11] Dr. Klauda, M. – Dr. Lauff, U.
Function and Software Development for ECUs
ATZelektronik 2008/06, (p. 10-15.)
- [12] Dr. Lakatos I. - Dr. Nagyszakolyai I.
Elektronikus dízelszabályozás
Novadat Bt, Győr, 1996
- [13] Markus, E. – Randoll, H. – Knauer, M.
Development and Future Challenges to Engine Control
ATZelektronik 2006/01, (p. 2-5.)
- [14] Dr. Mauss, J. – Simons, M.
Chip simulation of automotive ECUs on PC
ATZelektronik 2012/06, (p. 44-48.)
- [15] Mollenhauer, K.
Handbuch Dieselmotoren
Springer-Verlag Berlin, Heidelberg, 1997
5.2. Elektronische Regelung der Dieseleinspritzung
- [16] Pischinger, R. - Klell M. - Sams T.
Thermodynamik der Verbrennungskraftmaschinen 3. Auflage
Springer-Verlag, Wien, 2009
- [17] Reif, K.
Bosch Autoelektrik und Autoelektronik 6. Auflage
Vieweg+Teubner Verlag, Springer Fachmedien Wiesbaden, 2011
- [18] Vogels, M. S. – Heindl, A. – Krenn, M. – Leithgöb R. – Heppner, B.
High Performance ECU Calibration Using Realtime ECU Interface
ATZelektronik 2008/02, (p. 32-37.)
- [19] Wiley-ISTE
Formal Methods: Industrial Use from Model to the Code
ISTE Ltd, London, 2012