



**SZÉCHENYI  
EGYETEM**  
UNIVERSITY OF GYŐR

**EKÖP**



Egyetemi Kutatói Ösztöndíj Program

# **EKÖP-KDP Tanulmánykötet**

Kutatási téma:

**Követelményalapú modellvalidációs eljárások vizsgálata  
elektronikus vezérlőegységek funkció fejlesztéséhez**

**LENNERT JÓZSEF RICHÁRD<sup>1</sup>**

**JVP4KX**

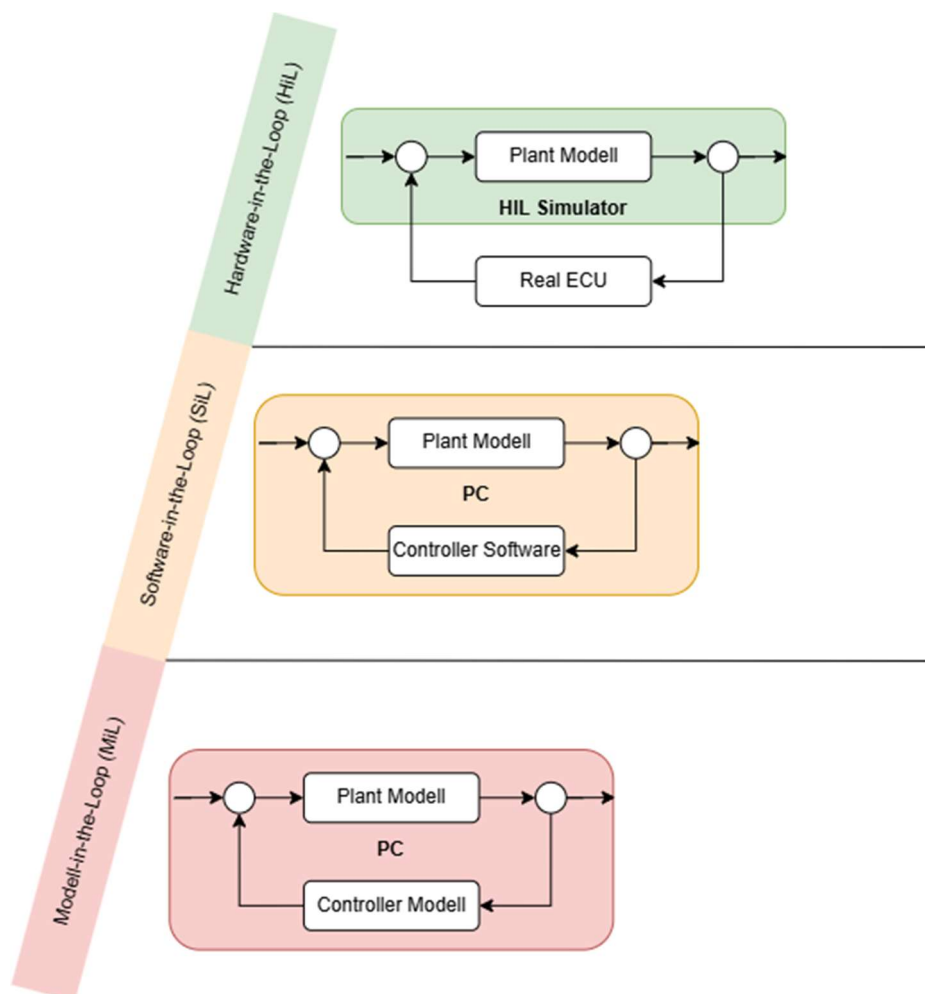
**Győr, 2026.04.25**

---

<sup>1</sup> Széchenyi István Egyetem Multidiszciplináris Műszaki Tudományi Doktori Iskola,  
Témavezető: Dr. Fodor Dénes, egyetemi docens, Vállalati Szakértő: Dr. Goricsán István,  
Munkáltató: Audi Hungaria Zrt.

## 1. Bevezetés

Napjainkban a különböző villamos és elektronikus (E/E) rendszerek rendkívül fontos szerepet játszanak számos iparág területén, mint például az autópárhban [1]. Ezek a rendszerek az esetek jelentős részében ráadásul biztonságkritikus feladatot látnak el, így kulcsfontosságú, hogy fejlesztésük idő- és költséghatékonyan, valamint a meghibásodás jelentette kockázat minimalizálásával történjen. Ezen tényezők mind hozzájárulnak ahhoz, hogy a szimulációkon és így a különféle digitális ikreken alapuló digitális fejlesztési folyamatok egyre nagyobb szerephez jussanak, hiszen ezeknek a folyamatoknak a korábban említetteken felül számos további előnye is van [2]. Egyrészt nem igénylik a tesztelni kívánt rendszer fizikai megvalósítását, így rendkívül gyorsan és rugalmasan módosíthatók. Másrészt az elvégzett tesztek bármennyiszer megismételhetők, és megvalósíthatók olyan tesztesetek is, amelyek egy fizikai prototípus esetében csak rendkívül nehezen és prototípusonként csak egyszer lennének elvégezhetők (például roncsolásos vizsgálatok).



1. ábra Különböző típusú X-in-the-Loop (XiL) szimulációk

Az E/E rendszerek fejlesztése során különösen fontosok az 1. ábrán bemutatott X-in-the-Loop (XiL) szimulációk, valamint a modellalapú tervezés, amelyek kulcsszerepet játszanak az elektronikus vezérlőegységek (ECU) és a rajtuk futó szoftverek fejlesztése során. A modellalapú tervezés kiindulási pontja az úgynevezett Model-in-the-Loop (MiL) eljárás, amely során az irányítani kívánt (plant) és az irányítórendszer matematikai modelljét is egy megfelelő környezetben, például MATLAB/Simulink-ben leképezik, és tesztelik, hogy a tervezett szoftver képes-e ellátni a neki szánt feladatokat. Amennyiben a MiL szimuláció sikerrel jár, úgy a Software-in-the-Loop (SiL) eljárás következik, amely során a irányítórendszer matematikai modelljéből kódot generálnak és ellenőrzik egyrészt, hogy a kódgenerálás sikeresen végrehajtható volt-e, másrészt, hogy a generált kód képes maradt-e a plant modell megfelelő irányítására [3]. A sikeres SiL szimulációkat Hardware-in-the-Loop (HiL) szimulációk követik, amelyeket bizonyos esetekben Processor-in-the-Loop (PiL) szimulációk is megelőzhetnek. A Hardware-in-the-Loop szimulációk esetében már a tényleges ECU-ra töltik fel a szoftvert és ezt kapcsolják össze egy valós idejű szimulátorral, amelyet a plant modell futtatására használnak fel. A valós idejű szimulátor a rajta futtatott plant modell segítségével elhitei a vezérlőegységgel, hogy a tényleges rendszerbe került integrálásra, így számos funkcionális tesztelhető és a vezérlőn található szoftver egészen addig finomhangolható, amíg annak a tényleges rendszerbe történő illesztése már nem hordoz magában kockázatot [4, 5].

Ahogy az az előzőekben ismertetésre került, a XiL szimulációk során az irányítórendszer gyakorlatilag minden szimulációs lépésnél változik, hiszen a MiL szimuláció esetében matematikai modell, a SiL esetében már C forráskód, a HiL esetében pedig valódi vezérlőegység kerül felhasználásra. Ezzel szemben a plant modell vagy annak bizonyos elemei akár a teljes XiL szimulációs folyamat során változtatás nélkül kerülnek felhasználásra. Éppen ezért kulcsfontosságú, hogy ezeket a digitális ikreket a fejlesztési fázis bármely szakaszában hatékonyan lehessen optimalizálni, illetve a szimulációk során fellépő hibák analízise is hatékony legyen. Mivel az E/E rendszerek plant modelljében jelentkező hibák a fejlesztési folyamat bizonyos lépéseit teljesen megakaszthatják, így kulcsfontosságú olyan hibaanalízis megoldások kutatása és fejlesztése, amelyek optimalizálhatják ezt a folyamatot [3, 6].

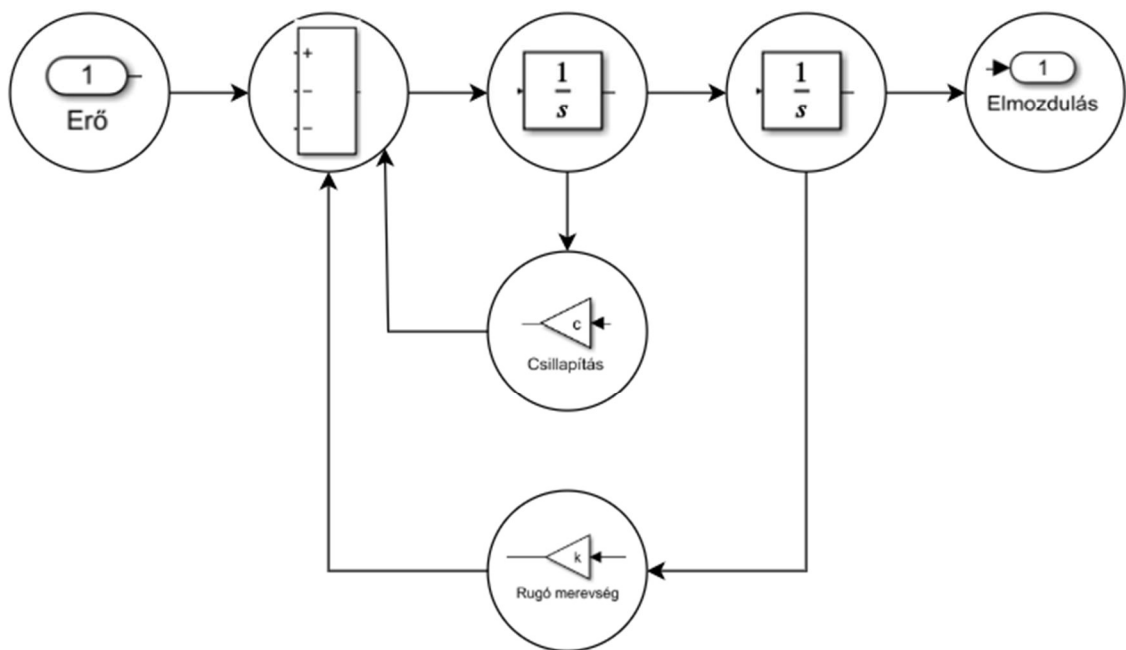
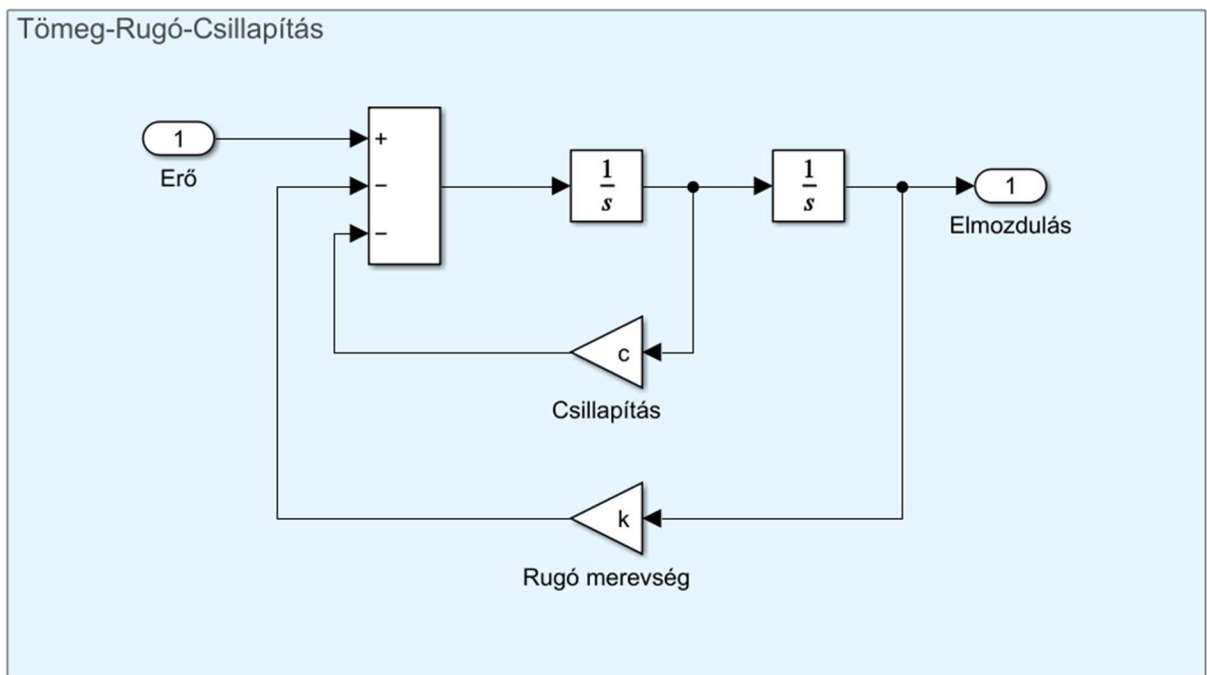
## **2. Az alkalmazott módszerek**

A MATLAB/Simulink-alapú digitális ikrek hibaanalízisét számos különböző módszerrel lehet elvégezni. Ezek közül a legkézenfekvőbb a manuális hibaanalízis, amelyet a fejlesztési környezet a „Highlight to source” és a „Highlight to destination” beépített funkciókkal támogat. Ez a módszer azonban komplex matematikai modellek esetében rendkívül lassú és számos

speciális blokk esetében a visszakövetett jel elvesztésének kockázatát is magában hordozza. Éppen ezért ez a megoldás hosszútávon nem gazdaságos a hibaanalízis elvégzésére.

Egy másik lehetséges megközelítés lehetne a mesterséges intelligencia alkalmazása is, amely rendkívül gyors és pontos hibaanalízist tenne lehetővé, azonban ebben az esetben is számolni kell néhány korlátozó tényezővel. Egyrészt a mesterséges intelligencia alapú megoldások esetében mindig szükség van megfelelő mennyiségű adatra a tanításhoz, illetve a tanítás eredményének érvényesítéséhez. A tanítás az esetek többségében kifejezetten időigényes folyamat, és ennek a folyamatnak a megismétlése a digitális ikrek szerkezetét és felépítését érintő minden egyes nagyobb változtatás után szükséges lehet. Másrészt ezen digitális ikrek a vállalatok számára fontos és értékes szellemi tulajdonok, amelyeknek védelme a vállalatok számára kulcskérdés, így mesterséges intelligenciával történő analízisük komoly aggályokat vethet fel a szellemi tulajdon védelmének szempontjából is.

Ezen okokból kifolyólag a kutatás során egy saját megoldást dolgoztunk ki a XiL szimulációk során alkalmazott digitális ikrek hibaanalízisére. Ezen megoldás lényege, hogy a digitális ikreket, mint gráfok komplex rendszerét kezeljük. Ennek a megoldásnak köszönhetően jelentősen javítható a hibaanalízis gyorsasága és pontossága még a speciális blokkok jelenléte esetén is, miközben a megoldáson alapuló szoftver semmilyen későbbi tanítást nem igényel, és a szoftvernek köszönhetően az egész folyamat rendkívül jól automatizálható is. Emellett fontos megemlíteni, hogy ez a megoldás nem csak magát a hibaanalízist könnyítheti meg, hanem elősegítheti magának a modellnek a feltérképezését és összefüggéseinek átlátását is, amelynek köszönhetően a hibaanalízishez szükséges modellspecifikus tudás jelentősen csökkenthető [7].



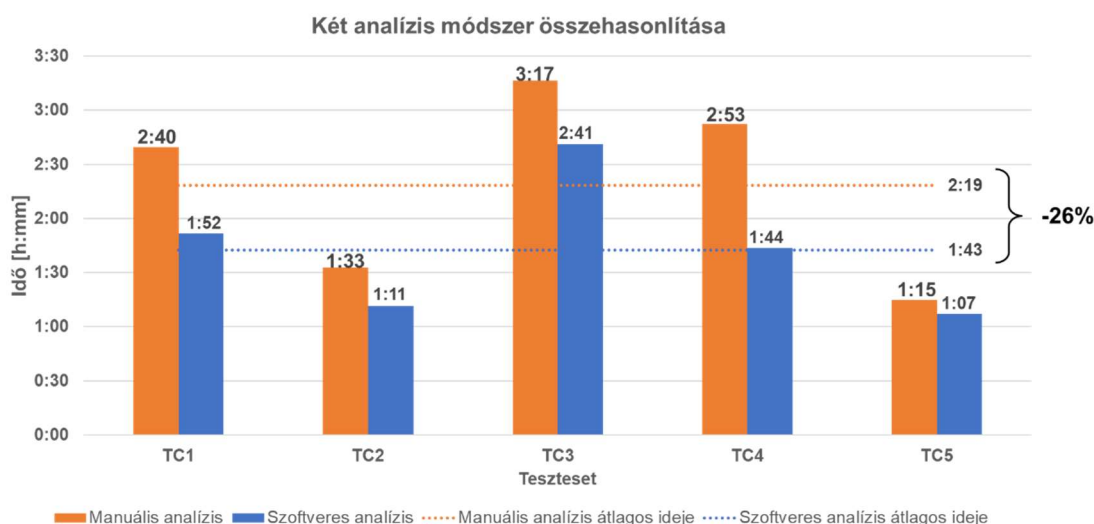
2. ábra Tömeg-rugó-csillapítás megvalósítása MATLAB/Simulink modellként és ezen modell értelmezése irányított gráfként

Ahogy a 2. ábra is szemlélteti, bármely MATLAB/Simulink modell értelmezhető irányított gráfként, mivel a grafikus modellen belül a különböző blokkok közötti jellánc minden esetben rendelkezésre áll, így ezzel a megközelítéssel mind a jelek visszakövetése, mind pedig egy adott jel teljes rendszerre gyakorolt hatása jól feltérképezhetővé és értelmezhetővé válik. Természetesen fontos kiemelni, hogy ez a megoldás alapvetően komplex szimulációs modellek

és buszrendszerek modelljeinek esetében tekinthető igazán hatékonyak, mivel ilyen esetekben a jellánc nem látható át egyértelműen.

### 3. A kutatás jelenlegi státusza

A korábban bemutatott módszert alapul véve kifejlesztettem egy MATLAB-alapú szoftvereszközt, amely képes a Simulink modellként megvalósított digitális ikrek matematikai modelljét az előző ábrán látható módon irányított gráfként kezelni, és lehetővé teszi egy adott kimeneti jel értékét befolyásoló összes korábbi jel visszakövetését. Ez a szoftvereszköz először egyszerű tesztmodellek esetében került kipróbálásra, amelyek szerkezete ugyan megegyezett az E/E rendszerek digitális ikreivel, azonban komplexitásuk jóval kisebb volt. Mivel ezek a tesztek nagymértékű hatékonyságjavulást mutattak, így kezdetét vehette a szoftvereszköz valós digitális ikreken történő tesztelése is. Ezen tesztek eredményei is rendkívül bizalomkeltőek, hiszen a szoftvereszköz első élesben tesztelt változata 25% körüli analízisidő csökkenést eredményezett, amelyet a 3. ábra szemléltet [7].



3. ábra A manuális és a szoftveres analízis időigényének összehasonlítása

A tesztelésbe bevont munkatársaim a visszajelzései alapján elvégzett módosításoknak és finomhangolásoknak köszönhetően az analízisidő további csökkenése várható, amely jelentősen csökkentheti ezen szűk keresztmetszet kritikusságát. Ezen felül a szoftvereszköz tesztelése során egy másik mérőszám is bevezetésre került, amely a szükséges műveletek száma. Ez lehetővé teszi a hatékonyság növelésének pontosabb és egyéb tényezőktől teljesen független mérését.

#### **4. További tervek**

A kutatás jelenlegi fókusza elsődlegesen az offline modellanalízishez kapcsolódó új megoldás kidolgozása és az ezen alapuló szoftvereszköz megvalósítása volt, azonban jövőbeni terveim között szerepel a módszer továbbfejlesztése és használhatóságának kiterjesztése. Egyrészt, ahogy az korábban említésre került, a módszer és az azon alapuló szoftvereszköz a jelenlegi formájában csak a jelek visszakövetésére használható, azonban már most megjelent az igény arra, hogy egy adott bemeneti érték teljes digitális ikerre gyakorolt hatása is vizsgálható legyen. Másrészt a kutatás rendkívül fontos célja az offline modell és a XiL szimulációs környezetek közötti kapcsolat megvalósítása, amely a jelenlegi szoftvereszköz nagymértékű kibővítését igényelheti. Ennek fő oka, hogy a SiL és HiL szimulációk esetében az eredetileg MATLAB/Simulink-ben létrehozott plant modellek már nem feltétlenül grafikus modellként jelennek meg, hanem a szimuláció futtatási környezete által igényelt átalakításon, például kódgeneráláson és fordításon átesve, amely jelentősen nehezítheti a visszakövetést. Ezen felül rendkívül fontos a kimenetre hatást gyakorló jelek visszakövetésének és a teljes jellánc feltérképezésének elkülönítése, ugyanis ezzel a szoftvereszköz sokkal általános célúbbá válhat, hiszen ezzel a módosítással már nem csak hibaanalízisre, hanem a modell működésének és a benne jelenlévő komplex összefüggéseknek a megértésére is használhatóvá válik.

## Felhasznált irodalmak:

1. Askaripoor, H.; Hashemi Farzaneh, M.; Knoll, A. E/E Architecture Synthesis: Challenges and Technologies. *Electronics* 2022, 11, 518.  
<https://doi.org/10.3390/electronics11040518>
2. Mihai, S.; Yaqoob, M.; Hung, D.V.; Davis, W.; Towakel, P.; Raza, M.; Karamanoglu, M.; Barn, B.; Shetve, D.; Prasad, R.V.; et al. Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects. *IEEE Commun. Surv. Tutor.* 2022, 24, 2255–2291. <https://doi.org/10.1109/COMST.2022.3208773>
3. Shylla, D.; Jain, A.; Shah, P.; Sekhar, R. Model in Loop (MIL), Software in Loop (SIL) and Hardware in Loop (HIL) Testing in MBD. In Proceedings of the 2023 4th IEEE Global Conference for Advancement in Technology (GCAT), Bangalore, India, 6–8 October 2023; pp. 1–6. <https://doi.org/10.1109/GCAT59970.2023.10353323>
4. Farooq, U. Model Based Verification of Electronic Control Unit (ECU) in High Lift Systems: Verification and Validation. In Proceedings of the 2017 IEEE International Systems Engineering Symposium (ISSE), Vienna, Austria, 11–13 October 2017; pp. 1–5. <https://doi.org/10.1109/SysEng.2017.8088318>
5. Bouscayrol, A. Different Types of Hardware-In-the-Loop Simulation for Electric Drives. In Proceedings of the 2008 IEEE International Symposium on Industrial Electronics, Cambridge, UK, 30 June–2 July 2008; pp. 2146–2151.  
<https://doi.org/10.1109/ISIE.2008.4677304>
6. Szalay, Z. Next Generation X-in-the-Loop Validation Methodology for Automated Vehicle Systems. *IEEE Access* 2021, 9, 35616–35632.  
<https://doi.org/10.1109/ACCESS.2021.3061732>
7. Lennert, J.R.; Fodor, D.; Szalay, I. Development of an Automated Solution for the Error Analysis of MATLAB/Simulink-Based Digital Twins. *Eng. Proc.* 2025, 113, 49.  
<https://doi.org/10.3390/engproc2025113049>