

Rózsa Máté, Dr. Csátár János

Interfész algoritmusok vizsgálata Power-Hardware-in-the-Loop szimulációs környezetben

A cikk a Power-Hardware-in-the-Loop (PHIL) szimulációs környezetben használt interfész algoritmusokat tárgyalja. A kutatásunk célja olyan interfész módszerek kidolgozása, amelyekkel a PHIL szimuláció széles paramétertartományban elvégezhető. Ennek első lépéseként a nemzetközi irodalomban megtalálható interfész algoritmusokat implementáltuk Matlab Simulink környezetben. A BME-MVM Smart Power Laboratórium eszközkészletét használva valós PHIL környezetben teszteltük az implementált interfész algoritmusokat. A szimulációs eredményekből megállapíthattuk, hogy az irodalmak szerinti interfész algoritmusok limitációkkal rendelkeznek, amelyek befolyásolják a PHIL szimuláció sikerességét.

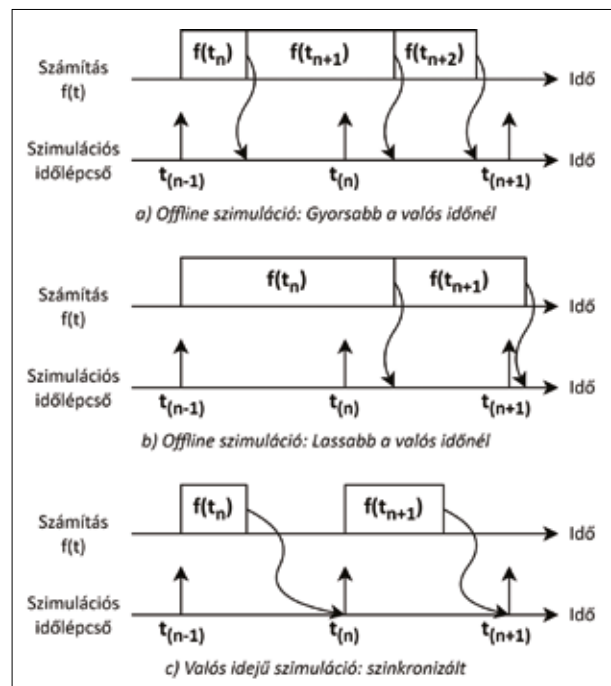
This article describes the interface algorithms used in the Power-Hardware-in-the-Loop (PHIL) simulation environment. Our research aims to develop interface methods to perform PHIL simulation over a wide range of parameters. As a first step, we implemented interface algorithms available in the international literature in Matlab Simulink environment. Using the BME-MVM Smart Power Laboratory toolkit, the implemented interface algorithms were tested in a real PHIL environment. From the simulation results, we found that the interface algorithms reported in the literature have limitations that affect the success of PHIL simulation.

1. BEVEZETÉS

Az elmúlt évtizedek során a szignifikánsan növekvő elektifikáció hatására folyamatos igény mutatkozik a villamosenergia-rendszer jövőbemutató fejlesztési lehetőségeinek kutatására. A megújuló termelők, az elektromobilitás és a fogyasztói igények növekedési tendenciájával a villamosenergia-rendszer fejlődése nehezen képes lépést tartani. A kutatás-fejlesztési folyamatok során elvégzendő vizsgálatokhoz a modellezés és a szimulációk használata elengedhetetlen egyrészt a gyors, iteratív és mindazonáltal költséghatékony feladatvégzéshez, másrészt a villamosenergia-rendszer kiterjedtsége és költségessége végett nem lehetséges egy valós tesztrendszer kialakítása. A szimulációk használata a számítástechnika fejlődésével és a számítási kapacitás növekedésével egyre szélesebb felhasználói réteg számára vált elérhetővé. Ennek eredményeképp a szimulációs keretrendszerek elterjedten használtak a villamos energetika területén belül. A modellben leképezett hálózatrészen végezhetünk például energiaáramlási (loadflow) vagy transziens vizsgálatokat, ezen túlmenően a hálózati modelltől eltérő, de villamos energetika szempontjából fontos vizsgálatok is elvégezhetőek a rendelkezésre álló rugalmas eszközkészlet segítségével (pl.: szabályozók, vezérlő berendezések modellezése).

2. VALÓS IDEJŰ SZIMULÁCIÓ

Az elkészített modelleket a szimulációs keretrendszerben diszkrét idejű megoldó algoritmus (solver) segítségével futtatjuk, amelyek közül létezik állandó és változó időlépcsőjű verzió is. A szimulációk fontos tulajdonsága a futtatási idő és az időlépcső valós időhöz mért elhelyezkedése, a különböző eseteket az 1. ábra szemlélteti. Offline esetben a futtatás gyorsaságát a számítási kapacitás határozza meg, általános cél a leggyorsabb sebesség elérése. Egy időlépcső számításához szükséges idő lehet kevesebb és több is, mint a szimulációs időlépcső nagysága. Valós idejű szimuláció esetén az időlépcső a valós idővel szinkronozott, így a modell állapotváltozásai azonos időben történnek egy annak megfelelő fizikai rendszerrel. Fontos megemlíteni, hogy csak fix időlépcsővel dolgozó solver használható és maximálisan akkora számítási igényű modell futtatható, amelynek számításához szükséges idő nem több, mint az időlépcső nagysága. Ha a számítási kapacitás nem elegendő, úgynevezett túlfutás (overrun) lép fel és nem lesz számítási eredmény az adott időlépcsőben. A számítási kapacitás kritikussága és a valós idővel történő szinkron működés biztosítása érdekében valós idejű szimulációra célszámítógépeket, ún. Real-Time-Simulatort (RTS) alkalmazunk. A valós idejű szimulációról és felhasználási területeiről bővebben olvashatunk az OPAL-RT Technologies által írt cikkben [1].



1. ábra Offline és valós idejű szimuláció viszonya a valós idővel [1]

3. A HARDWARE-IN-THE-LOOP SZIMULÁCIÓ

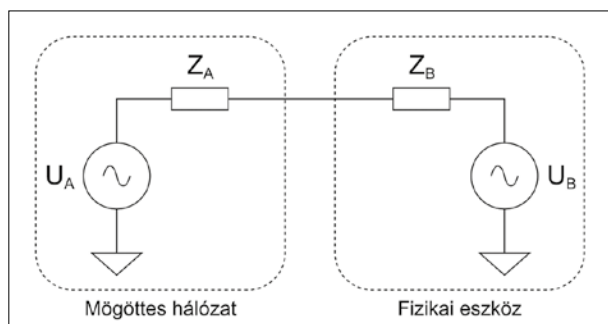
A valós idejű futás kihasználásával a modellhez interfész egységeken keresztül külső rendszerekből tudunk jeleket, adatokat ki- és visszavezetni. Például A/D és D/A átalakítókkal analóg és digitális jeleket vagy Ethernet interfészen keresztül bármilyen Ethernet alapú kommunikációs protokoll használható (IEC 61850, Modbus TCP stb.). A modellből kivezetett jelle egy fizikai eszközt kapcsolva, majd ennek válaszjelét visszacsatolva a modellbe úgynevezett hardware-in-the-loop (HIL) szimulációs környezetet alakíthatunk ki. Ennek segítségével a tesztelt eszköz számára üzemszerű környezetet emulálha-

tunk, amelyben a teszt scenáriók szabadon definiálhatóak és automatizálva elvégezhetőek. A HIL szimulációk halmazában az alábbi funkcionális típusokat különböztetjük meg:

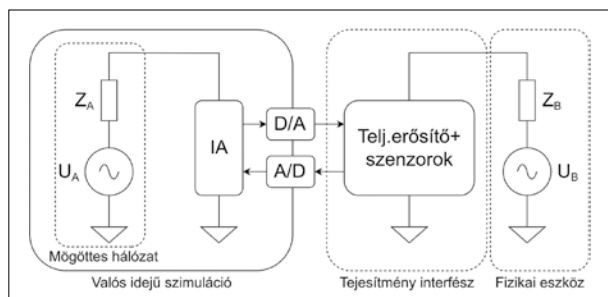
- Rapid-Control-Prototyping (RCP): A szabályozót a szimulációban valósítjuk meg, a szakasz (szabályozott berendezés) egy fizikai eszköz, így a szabályozó paraméterei gyorsan, akár futás közben is változtathatóak.
- Controller-Hardware-in-the-Loop (CHIL): A kialakítás inverze az RCP-nek, a szabályozó fizikai eszközként van jelen és a szakasz modelljét szimuláljuk (pl.: védelmi és automatizálási készülékek tesztelése).
- Software-in-the-Loop (SIL): A szabályozó és a szakasz is a szimulációban helyezkedik el, a két részegység közti kapcsolat szoftveresen van megvalósítva, így elkerülhetőek a jelfeldolgozás során keletkező hibák.
- Power-Hardware-in-the-Loop (PHIL): Az alapvető HIL struktúrát teljesítményerősítővel és szenzorokkal egészítjük ki, így létrehozható egy erősáramú csatlakozási pont, amelyhez a tesztelni kívánt eszközt csatlakoztatjuk (pl.: PV inverter tesztelése).

4. INTERFÉSZ ALGORITMUSOK PHIL SZIMULÁCIÓS KÖRNYEZETBEN

A PHIL szimulációnál a modell és a fizikai eszköz közti kapcsolódás megvalósítása a többi HIL struktúrával szemben nem egyértelmű, hiszen a modellben lévő virtuális erősáramú pontra szeretnénk csatlakozni egy teljesítményerősítő közbeiktatásával, majd ugyanerre a pontra csatoljuk vissza a fizikai eszköz hatását. Ezért a modellben szükséges kidolgozni egy olyan részt (interfész algoritmust), amely képes a teljesítményerősítő vezérlésére és a mért villamos mennyiségek alapján valósághűen emulálja a fizikai eszköz hatását.



2. ábra Szimulálni kívánt elrendezés (referencia)



3. ábra A PHIL blokkvázlata [2]

A 2. ábra szemlélteti a referencia elrendezést: a mögöttes hálózat és a fizikai eszköz egy-egy Thevenin-ekvivalenssel szerepel és a köztük lévő összeköttetés ideális. A részletes PHIL szimulációs blokkvázlat a 3. ábra szerinti. A mögöttes hálózat és a fizikai eszköz közti kapcsolatot az alábbi egység alakítja ki:

- IA: interfész algoritmus
- A/D, D/A konverterek
- Teljesítmény interfész: teljesítményerősítő és szenzorok.

Megfigyelhető, hogy a részletes blokkvázlat elemei az IA-n kívül definiált paraméterekkel fognak rendelkezni, így a virtuális és a valós oldal közti kapcsolat kialakításánál csak az IA-t választhatjuk meg szabadon.

4.1 Interfész algoritmus jelentősége: stabilitás és pontosság

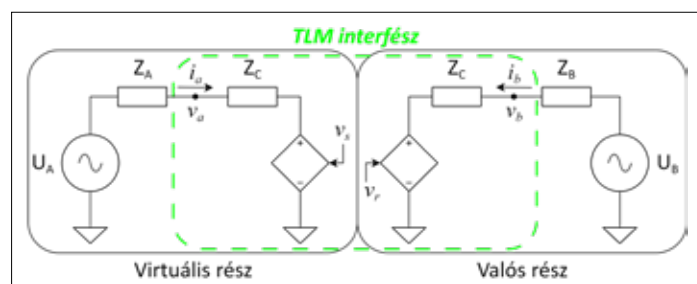
A nagy teljesítménnyel (átlagosan 10-100 kW) végzett PHIL szimulációk megkövetelik, hogy a rendszer állandóan stabil tartományban üzemeljen. Ellenkező esetben az instabilitás következtében megnövekvő feszültségek és áramok a tesztelt berendezés tönkremeneteléhez vezethetnek. Ugyanakkor fontos követelmény a szimulációval szemben, hogy a lehető legpontosabb legyen, közelítse az ideális esetet (2. ábra), amikor a modellezett hálózat fizikai megvalósításához csatlakozik a tesztelt eszköz.

Az instabilitás fő okozója a szimulációs időlépcső generálta késleltetés, amely miatt a szimulációban a tesztelt eszköz hatása mindig késleltetve jelenik meg. Ez a következő időlépcső számításában hibát eredményez, amely újbóli ki- és visszacsatolása növeli a helyes értéktől való eltérést, ami egy bizonyos szinten felül instabilitáshoz fog vezetni. A pontosságot a késleltetésből fakadó hibákon túl az A/D, D/A kvantálási hibák, a teljesítményerősítő erősítési hibája és a szenzorok mérési bizonytalansága is befolyásolja.

Az interfész algoritmus megválasztásával mindkét követelményt közvetlenül befolyásolhatjuk, így helyes megválasztása kulcsfontosságú.

4.2 Ideal Transformer Model – ITM

Az ideális transzformátor modell a legegyszerűbb, ennek köszönhetően a leggyakrabban alkalmazott IA, felépítését a 4. ábra szemlélteti. A virtuális oldalon mért feszültséget erősítjük egy teljesítményerősítővel, majd a valós rész áramát vezérelt áramforráson keresztül csatoljuk vissza a virtuális részre [3].



4. ábra ITM algoritmus blokkvázlata

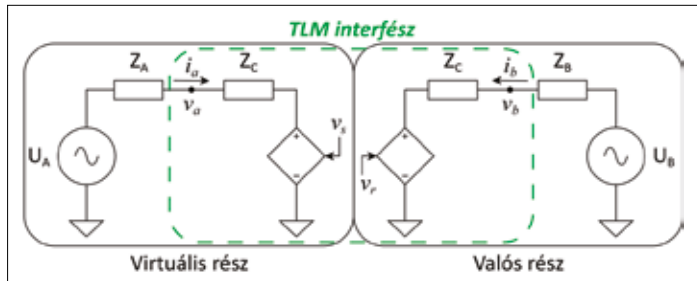
A módszer előnye az egyszerűség és a könnyű implementáció. Hátránya a stabilitási tartomány impedanciafüggése és a transziens jelenségekre való érzékenysége [4]. A stabilitási tartomány növelésére használhatóak kiegészítő módszerek, amelyek közül a legjelentősebbek:

- soros induktivitás elhelyezése a valós oldalon (áram simítása) [5]
- aluláteresztő szűrő alkalmazása az áram-visszacsatolásban [2], [6]
- koordináta-transzformáció alkalmazása (dq- vagy αβ-transzformáció).

4.3 Transmission Line Model – TLM

A TLM modell blokkvázlata az 5. ábra szerinti. A módszer a Bergeron távvezeték modellen alapul, a távvezeték két

csatolt Thevenin- vagy Norton-ekvivalenssel helyettesíthetjük, amelyben az impedancia (Z_c) értéke megegyezik a helyettesített távvezeték karakterisztikus impedanciájával [3]. A valós és a virtuális oldalon mért feszültségeket (v_a , v_b) és áramokat (i_a , i_b) behelyettesítve a hullámegyenletekbe kiszámítható a küldő és fogadó oldali feszültségek értékei (v_s , v_r). Az IA által leképezett távvezeték befutási idejének emulálására a szimulációs időlépcső okozta késleltetést használjuk fel.

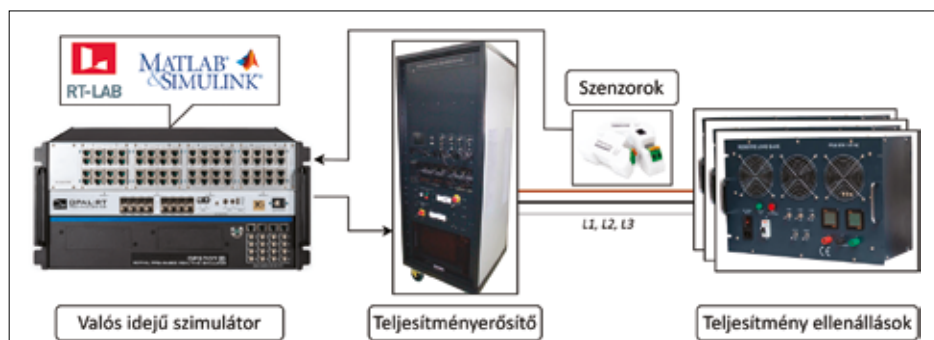


5. ábra TLM algoritmus blokkvázlata

A TLM algoritmus előnye, hogy a virtuális és a valós állapotváltozóit szétcsatolja, ezért a módszer nagy stabilitási határokkal rendelkezik. Hátránya, hogy a valós oldali karakterisztikus impedanciát egy ellenállással kell kialakítsuk, ami egyrészt csökkenteni fogja a teljesítményerősítő feszültség kivezérési tartományát, másrészt az ellenálláson nagymértékű veszteség is disszipálódik. Továbbá hátránya még a frekvenciafüggő viselkedés és a távvezeték jellegből adódó természetes teljesítményfüggő meddő teljesítmény előállítás [7].

5. INTERFÉSZ ALGORITMUSOK VIZSGÁLATA VALÓS TESZTKÖRNYEZETBEN

Az előző fejezetekben bemutatott interfész algoritmusokat Matlab Simulink környezetben valósítottuk meg, majd egy OPAL-RT OP5707XG típusú valós idejű szimulátort használva végeztük a laborteszteket. A mérésekhez használt eszközök elrendezése a 6. ábra blokkvázlatán látható. Az RTS-hez az analóg kimenetein keresztül egy Puissance Plus 3x7 kVA-es lineáris teljesítményerősítőt csatlakoztattunk, ez valósította meg az erősáramú csatlakozási pontot, amelyhez a tesztlini kívánt eszközt csatlakoztattuk. A valós oldali villamos mennyiségek mérésére Imperix gyártmányú szenzorokat használtunk. A tesztelt eszköz fázisonként egy-egy, fokozatonként kapcsolható teljesítmény ellenállás volt az egyszerűség és jó összehasonlíthatóság végett. A következőkben az ITM és TLM algoritmus összehasonlító méréseinek eredményeit ismertetjük. A vizsgálatokat 50 Hz-es szinuszzel végeztük. A virtuális és valós oldali teljesítményekből számítható az interfész algoritmus amplitúdó- és fázishibája. A valós oldalhoz



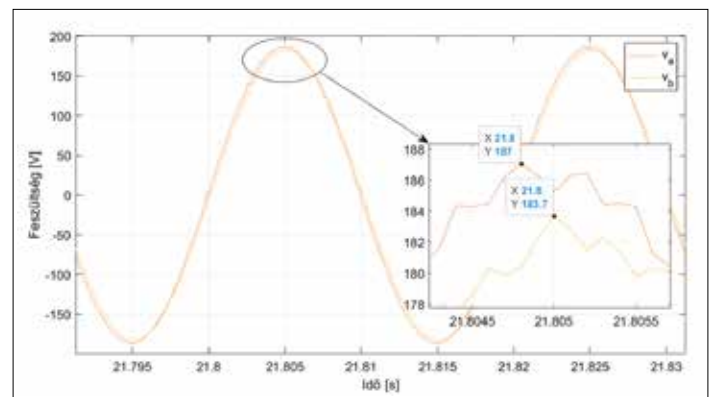
6. ábra Mérési elrendezés

viszonyított hibákat az 1. táblázat foglalja össze. A normál és fizikai induktivitással kiegészített ITM algoritmus közel azonos eredményeket mutat, a fázishiba a szimulációs időlépcső késleltetéséből fakad, míg az amplitúdóhiba a szenzorok áttételhibájából származik. Az aluláteresztő szűrővel kiegészített ITM algoritmusnál az amplitúdó- és fázishibában is látszik az aluláteresztő szűrő csillapító és fázisrontó hatása. A dq-transzformációs ITM módszernél nincs fázishiba, amely a módszer legnagyobb előnye. A TLM algoritmusnál mindkét hiba mértéke nagyra mondható. Ez egyrészt a távvezeték jellegből adódó átvitt teljesítménytől függő fázishibának, másrészt az algoritmus nagy zajérzékenységének és késleltetési inkonzisztencia-érzékenységének köszönhető.

1. táblázat Az interfész algoritmusok pontossága

Interfész algoritmus	Amplitúdóhiba	Fázishiba
ITM	1,228%	2,815°
ITM + aluláteresztő szűrő	-0,491%	13,536°
ITM + fizikai induktivitás	1,242%	2,879°
ITM + dq-transzformáció	1,242%	0,000°
TLM	8,300%	10,067°

A 7. ábra a normál ITM algoritmus feszültség jelalakját szemlélteti. Megfigyelhető mind az amplitúdó, mind a fázishelyzet kismértékű eltérése a virtuális és valós oldali jelek közt.



7. ábra Normál ITM módszer feszültség jelalakja

A stabilitás szempontú vizsgálatoknál a TLM algoritmus az elvárt eredményeknek megfelelően nem vált instabillá. Az ITM algoritmus változatainak vizsgálatokor a fizikai ellenállás értékének változtatása helyett a valós oldalról visszacsatolt áram értékét szoftveresen egy skálátényezővel változtattuk, így keresve a stabilitási határt. A 2. táblázat eredményeiben a normál ITM-et referenciaként használtuk, ez egységnyi stabilitással rendelkezik. Az aluláteresztő szűrővel 5,5-szeres stabilitásnövelő hatást értünk el, a fizikai induktivitással ennél csekélyebb 1,12-szeres stabilitás-

tásjavulást tapasztaltunk. A dq-transzformációs esetben volt a legnagyobb a stabilitás, itt a tényleges határt nem sikerült elérni, így csak egy alsó határt tudtunk meghatározni.

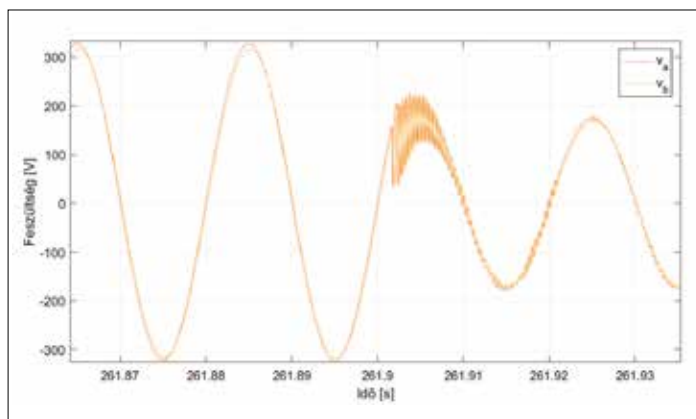
2. táblázat A stabilitásvizsgálat eredményei

Interfész algoritmus	Stabilitási határ – skálátényező
ITM	1
ITM + aluláteresztő szűrő	5,5
ITM + fizikai induktivitás	1,12
ITM + dq-transzformáció	>5,5

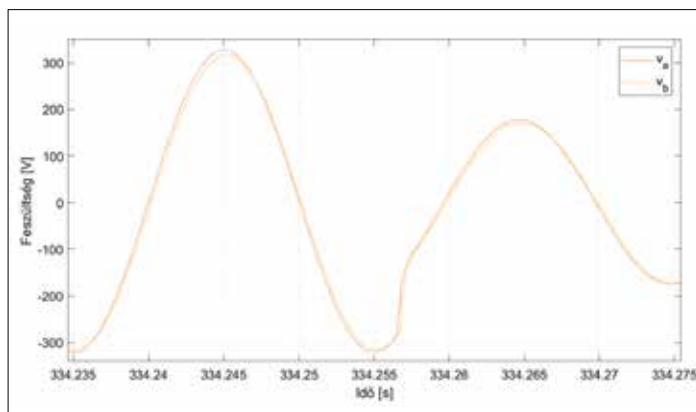
A szimulációnak transziens jelenségek esetén is stabilnak kell lenni. A teljesítményellenállások bekapcsolási transziensét a 8. ábra szemlélteti normál ITM algoritmust használó esetben, míg a 9. ábra az aluláteresztő szűrővel kiegészített módszer eredményét mutatja. Látható, hogy stabilitásnövelő módszer nélkül a kapcsolás hatására a feszültségjelben kialakul egy lecsengő transziens komponens, ami a második esetben a növelt stabilitás miatt nem jelentkezik.

6. ÖSSZEFOGLALÁS

A cikk első részében a valós idejű szimulációt és a PHIL funkcionális kategóriát ismertettük. Ezt követően két interfész algoritmus elvi konstrukcióját, előnyeiket és hátrányaikat mutattuk be, majd laboratóriumi méréseink eredményeit prezentáltuk. Az eredmények alapján elmondható, hogy az ITM algoritmus kiegészítő módszer nélküli esetben jó pon-



8. ábra Kapcsolási transziens hatása normál ITM algoritmus



9. ábra Kapcsolási transziens hatása aluláteresztő szűrővel kiegészített ITM esetén

tosságú, azonban a stabilitási tartomány szűkössége erősen limitálja az elvégezhető szimulációk körét. Az ITM stabilitása növelhető kiegészítő módszerekkel, de a pontosságot rontják. Az aluláteresztő szűrő alkalmazása csillapítást és jelentős fázishibát okozott. A fizikai induktivitás kismértékben csökkentette a pontosságot, de a stabilitási tartományt nem növelte meg jelentősen. A dq-transzformációs módszer a késleltetésből adódó fázishibát maradéktalanul megszüntette, de jelentős limitáció, hogy csak az implementált frekvenciákon képes átvitelt biztosítani. A TLM algoritmus az ITM algoritmussal szemben nagy stabilitási tartománnyal rendelkezik, de pontossága jóval elmarad. Az átvitt teljesítménytől függő fázishiba és a fizikai ellenállás jelentős hátránya a módszernek.

Az elméleti feltevéseinket a laboratóriumi vizsgálatok alátámasztották, egyben láthatóvá vált, hogy az interfész algoritmusok csak bizonyos körülmények közt használhatóak sikeresen. Ezért a későbbiekben célunk olyan interfész módszerek kidolgozása, amelyek szélesebb paramétertartományban képesek stabil és egyben pontos működésre.

Az interfész algoritmusok vizsgálatával kapcsolatosan bővebben az alábbi TDK dolgozatban olvashatunk. [8]

IRODALOMJEGYZÉK

- [1] J. Bélanger, P. Venne, and J.-N. Paquin, "The What, Where and Why of Real-Time Simulation," *Planet RT*, pp. 39–47, 2010.
- [2] T. Hatakeyama, A. Riccobono, and A. Monti, "Stability and accuracy analysis of power hardware in the loop system with different interface algorithms," *Apr. 2016*, pp. 1–8. doi: 10.1109/COMPEL.2016.7556671.
- [3] W. Ren, M. Steurer, and T. L. Baldwin, "Improve the stability and the accuracy of power hardware-in-the-loop simulation by selecting appropriate interface algorithms," *IEEE Trans Ind Appl*, vol. 44, no. 4, pp. 1286–1294, 2008, doi: 10.1109/TIA.2008.926240.
- [4] J. Dolenc, A. Božiček, and B. Blažič, "Stability analysis of an Ideal-Transformer-Model interface algorithm," in *2019 7th International Youth Conference on Energy (IYCE)*, 2019, pp. 1–6. doi: 10.1109/IYCE45807.2019.8991575.
- [5] X. Zha, C. Yin, J. Sun, M. Huang, and Q. Li, "Improving the stability and accuracy of power hardware-in-the-loop simulation using virtual impedance method," *Energies (Basel)*, vol. 9, no. 11, 2016, doi: 10.3390/en9110974.
- [6] A. Markou, V. Kleftakis, P. Kotsampopoulos, and N. Hatziaargyriou, "Improving existing methods for stable and more accurate Power Hardware-in-the-Loop experiments," in *IEEE International Symposium on Industrial Electronics*, Institute of Electrical and Electronics Engineers Inc., Aug. 2017, pp. 496–502. doi: 10.1109/ISIE.2017.8001296.
- [7] O. Tremblay, D. Rimorov, R. Gagnon, and H. Fortin-Blanchette, "A Multi-Time-Step Transmission Line Interface for Power Hardware-in-the-Loop Simulators," *IEEE Transactions on Energy Conversion*, vol. 35, no. 1, pp. 539–548, Mar. 2020, doi: 10.1109/TEC.2019.2941567.
- [8] Rózsa Máté and Dr. Csátár János, "Interfész algoritmusok vizsgálata Power-Hardware-in-the-Loop szimulációs környezetben", 2022, <https://tdk.bme.hu/VIK/DownloadPaper/Interfeszing-algoritmusok-vizsgalata-Power> (Megtekintve: 2023.04.07.)



Rózsa Máté

BME-MVM Smart Power Laboratórium
szakértő
BME VET villamosmérnök MSc hallgató
MEE-tag
rozsa.mate@edu.bme.hu



Dr. Csátár János

BME-MVM Smart Power Laboratórium
területvezető,
BME VET adjunktus
MEE-tag
csatar.janos@vik.bme.hu