

A power hardware-in-the-loop (PHIL) szimulációban az interfész algoritmus közvetlenül befolyásolja a pontosságot és a stabilitást, ezért helyes megválasztása elengedhetetlen. A cikkben két interfész algoritmusról: az ideal-transformer-model (ITM) és a transmission-line-model (TLM) algoritmusokról olvashatunk. Kutatásunk során a módszereket valós szimulációs környezetben vizsgáltuk. A szimulációs keretrendszer nemideális, ezért számos hardveres és szoftveres nehézség merült fel a tesztek folyamán. Ezen jelenségek többsége általánosan kiterjeszthető a PHIL szimulációkra. Pontos ismeretük esetén az interfész módszer jobban megválasztható és bizonyos hibák hatásának eliminálására további megoldásokat alkalmazhatunk. Az interfész algoritmus tulajdonságai tovább javíthatóak többszintű interfész kialakításával, ahol a módszerek előnyei egyesíthetők. A cikk második felében CPU és FPGA alapú hardvert egyaránt alkalmazó többszintű megoldást ismertetünk, amely stabilitási feltétele független a szimulációban lévő mögöttes impedancia nagyságától. Végül a két interfész módszer PHIL szimulációs eredményeivel alátámasztjuk az ITM és TLM algoritmusok előnyeit és hátrányait.

Choosing the right interface algorithm in power hardware-in-the-loop simulation is essential, because it directly affects the accuracy and stability of the simulation. This paper describes two interface algorithms used in PHIL environments: the ideal-transformer-model (ITM) and the transmission-line-model (TLM) algorithms. In our research, the methods were tested in a real simulation environment. The simulation framework is non-ideal and therefore several hardware and software difficulties were encountered during the tests. Most of these phenomena can be extended to all PHIL simulations. With a precise knowledge of non-idealities, the interface method can be better chosen and additional solutions can be applied to eliminate the effects of errors. The properties of the interface algorithm can be further improved by designing a multi-level interface where the advantages of the methods can be combined. In the second half of the paper, a multi-level solution using both CPU and FPGA based hardware is presented, where the stability is independent of the virtual side impedance. Lastly, the real-time PHIL simulation results are used to demonstrate the advantages and disadvantages of the ITM and TLM algorithms.

Címszavak: valós idejű szimuláció (RTS); power hardware-in-the-loop (PHIL); interfész algoritmus (IA); ideal transformer model (ITM); transmission line model (TLM); többszintű interfész algoritmus

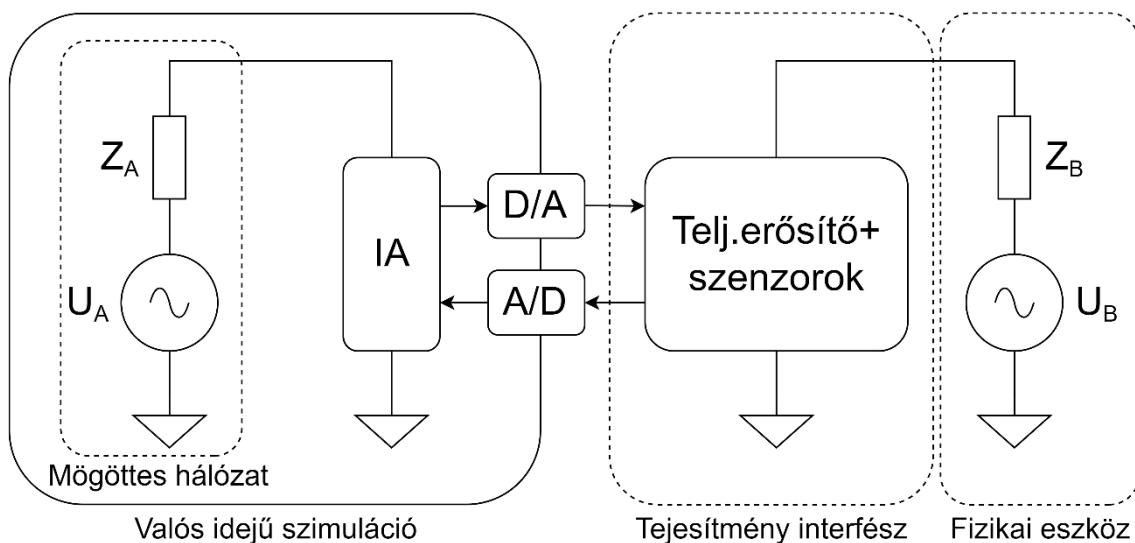
1. Bevezetés

Napjainkban a villamosenergia-rendszer eddig nem látott mértékű átalakuláson megy keresztül, amelyet elsősorban a fogyasztói igények megváltozása indukál. A megújuló-elosztott termelés, az elektromobilitás és a növekvő számú villamos energiával üzemeltetett berendezés új kihívások elé állítja a szakembereket. A szakterület specialitása, hogy a villamosenergia-rendszer kiterjedtsége és költségessége egy valós tesztrendszer kialakítását nem teszi lehetővé. Ezért a kutatás-fejlesztési folyamatokban a modellezés és a szimulációk használata elengedhetetlen a rugalmas, gyors és hatékony feladatvégzéshez.

A power hardware-in-the-loop (PHIL) a valós idejű szimuláció egy változata, ahol egy fizikai eszközt erősáramú kapcsain keresztül csatlakoztatunk a modellhez. Ezen módszer alkalmazásával a villamos készülékeket (pl.: inverterek, hajtási rendszerek) hatékonyan vizsgálhatjuk tetszőleges körülmények közt, akár automatizáltan az előre elkészített szcenáriók alapján. A PHIL szimulációhoz szükséges egy interfész a modell és a fizikai eszköz között, amely a valós idejű szimulátor kisjelű ki- és bemeneteit erősáramú csatlakozási ponttá alakítja. Az interfész megvalósítása közvetlenül befolyásolja a mérési eredményeket, ezért cél azon ideális, de fiktív eset közelítése, amikor a modell és a fizikai eszköz csatlakozási pontjai galvanikus kapcsolatban vannak.

A PHIL szimuláció részletes blokkvázlatát a(z) 1. ábra szemlélteti. A részegységek az alábbiak:

- Valós idejű szimulátor: a modell valós időben fut és rendelkezik gyengeáramú ki- és bemenetekkel a külső egységek csatlakoztatásához.
- Mögöttes hálózat: a modellben leképezett hálózat Thevenin-ekvivalense.
- Fizikai eszköz: a tesztelt erősáramú eszköz Thevenin-ekvivalense.
- Teljesítmény interfész: a gyengeáramú – erősáramú átalakítást végzi, teljesítményerősítőből és szenzorokból felépülő egység.
- IA – Interfész algoritmus: a visszamért villamos mennyiségek felhasználásával a modellben valósághűen emulálja a fizikai eszköz hatását.



1. ábra: A PHIL szimuláció blokkvázlata [1]

A virtuális és valós oldal között kialakított interfész részegységeinek nemidealitásai a szimuláció stabilitását és pontosságát egyaránt befolyásolják. A PHIL blokkvázlatban megfigyelhető, hogy egy adott szimulációs környezetben legegyszerűbben az interfész algoritmus megválasztásával tudjuk a minőségi paramétereket alakítani. Az IA tervezésekor jelentős előny, ha a szimulációs környezetből adódó hibák teljeskörűen ismertek. Az egyes interfész algoritmusok a különböző eredetű hibákra eltérő mértékben érzékenyek, így a szimulációban vizsgálni kívánt paraméterek is nagyban befolyásolják az IA kiválasztását.

A hibákat két nagy csoportba sorolhatjuk: lehet determinisztikus vagy sztochasztikus. Amennyiben a hiba determinisztikus, úgy jó eséllyel alkalmazhatunk korrekciót, például

szenzor offset kompenzálást. Ha a hiba sztochasztikus, akkor teljes mértékben még állandósult állapotban sem lehet korrigálni a hibát, például a mintavételi időközök egymástól való eltérése.

A nemzetközi szakirodalomban az interfész algoritmusok 5 alapvető változata taglalt:

- ITM – Ideal Transformer Model
- TLM – Transmission Line Model
- DIM – Damping Impedance Method
- PCD – Partial Circuit Duplication
- TFA – Time-variant First-order Approximation [2].

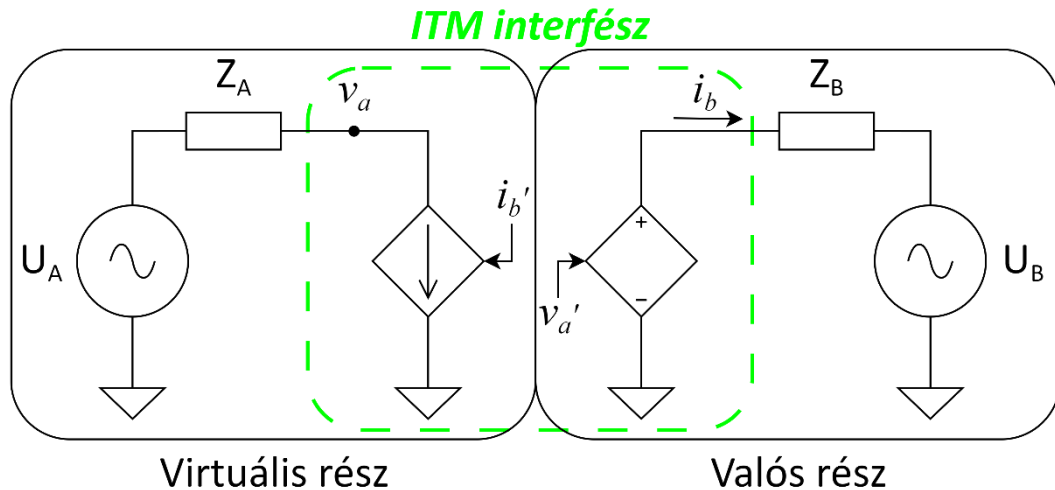
Kutatásunk során két módszert választottunk ki az előzetes irodalomkutatást követően. Előfeltételként fogalmaztuk meg, hogy a rendelkezésre álló szimulációs keretrendszerben megvalósítható interfész szükséges, ezért a valós oldalon feszültséggenerátoros kialakítású módszereket kerestünk. Feltételként szabtuk továbbá azt, hogy a választott megoldás ne legyen idegen a villamosenergia-hálózatok/rendszerek modellezésétől és kezdetben egyszerűen, szubiterációktól mentesen kivitelezhető legyen. Ezért egyik interfész algoritmusként az ITM-et választottuk. A villamosenergia-rendszer pontjait távvezetékek kötik össze. A TLM algoritmus egy távvezeték képez le, így természetes interfészként szolgálhat a megcélzott felhasználási területen. Ezért második interfész algoritmusként a TLM-et választottuk.

Az interfész algoritmusok alkalmazása során offline és valódi PHIL környezetben számos szimulációs keretrendszer okozta probléma merült fel, amelyek nagy része általánosságban is kiterjeszthető a PHIL környezetekre. A hibákra megoldási lehetőségek keresése folyamatosan munkánk részét képezi, amely aktuális állását jelen cikk keretében kívánjuk ismertetni.

A következőkben először a két vizsgált interfész algoritmust ismertetjük. Majd előnyeik és hátrányaik, a szimulációs keretrendszer okozta nehézségek bemutatása olvasható. Ezt követően a többszintű interfész algoritmus koncepciót és a PHIL szimuláció statikus és dinamikus tulajdonságainak ezen kialakítással elérhető javulását taglaljuk. Végül a két alap interfész módszer laborszimulációinak eredményeit prezentáljuk.

2. ITM – Ideal Transformer Model

Az ITM algoritmus kialakítása lehet feszültség- vagy áramgenerátoros. A PHIL szimulációs keretrendszerek teljesítményerősítői általában feszültségforrások, ezért a cikkben a feszültséggenerátoros felépítést ismertetjük. Az ideális transzformátor modell $a(z)$ 2. ábra szerint épül fel. A virtuális oldalon mért feszültséget a valós oldalon egy teljesítményerősítővel alakítjuk erősáramú jellé, majd a valós oldalon mért áram nagyságát visszacsatoljuk a virtuális oldalra. Az ITM algoritmus egyszerű, robosztus, ezért alkalmazása széles körben elterjedt, de kis stabilitási tartománya miatt gyakran kiegészítő módszerek szükségesek [2], [3].



2. ábra: ITM algoritmus felépítése

Az ITM algoritmus stabilitási tartományát két tényező befolyásolja: a virtuális és valós oldali impedanciák aránya és a késleltetések. A nyílt hurkú átviteli függvénye az (1) egyenlet szerinti. Ha az átviteli függvény késleltetés nélküli tagját nézzük, akkor a (2) kritérium teljesülése esetén stabil lesz a rendszer. Ha az interfész nemideális, akkor a késleltető tag rontja a stabilitási tartományt, így a (2) kritérium teljesülése nem elegendő a stabilitáshoz [1], [4].

$$G_{OL} = \frac{Z_A}{Z_B} \cdot e^{-s\Delta t} \quad (1)$$

$$\frac{Z_A}{Z_B} \leq 1 \quad (2)$$

A stabilitás növelésére a szakirodalom kiegészítő módszereket definiál:

- A visszacsatolt áramjel szűrésével a stabilitási tartomány kiterjeszthető: aluláteresztő vagy polinomiális szűrő megfelelő hangolásával. A nagyobb frekvenciák kiszűrésével a vizsgálható frekvenciatartomány csökken, a szűrő amplitúdó- és fázishibája pontatlanságot okoz [1], [3], [5]–[8].
- A valós oldalon soros fizikai induktivitás alkalmazása szintén növeli a stabilitást, azonban fizikai elem lévén drágítja a szimulációt és szintén csökkenti a pontosságot [3].
- A virtuális oldalon sönt ágban elhelyezett kondenzátor hasonló eredménnyel rendelkezik, mint a valós oldali induktivitás, de a módszer gyakorlati alkalmazása nem elterjedt [9].

A kiegészítő módszerek használatával nem növelhető a stabilitási tartomány minden határon túl, mert a pontosság már nem lesz megfelelő. Amennyiben a kiegészített tartomány sem elegendő a szimuláció elvégzéséhez, úgy más interfész módszert kell kidolgozni.

A késleltetés jelentős részét legtöbb esetben a szimulációs időlépcső adja, ehhez adódik hozzá az A/D, D/A konverziós idő, a teljesítményerősítő és szenzorok késleltetése és a jelterjedési idő. A késleltetés fázishibát okoz, amely csökkenti a stabilitási tartományt és a valós és virtuális oldalon mért hatásos és meddő teljesítmény arányokat eltolja. A késleltetések mélyebb

ismertetését két részre kell felosztani az implementációs környezet és egyben a szimulációs időlépcső nagysága alapján.

Az univerzális CPU-t használó solverek időlépcsője hardveres korlátok miatt $T_s \geq 20\mu s$. A szimulációs időlépcsőhöz képest a PHIL környezet többi eleme okozta késleltetés elhanyagolható nagyságú, ezért elegendő csak a szimulációs időlépcsőt figyelembe venni. A késleltetés pontos értékének tudatában a fázishiba kompenzálására számos módszer alkalmazható.

- Az áramvisszacsatolásban dq- vagy Fourier-transzformációval eliminálható a fáziskülönbség. Az interfész algoritmus így mindkét esetben csak azon frekvenciákon működik, amelyekre implementálva lett a transzformáció, ezért ettől eltérő frekvenciák nem vizsgálhatóak [10], [11].
- A virtuális oldalon mért feszültség fázishelyzetének befolyásolása (siettetése) után csatoljuk ki a teljesítményerősítőt a jelet. Ez egyrészt Fourier-transzformációval, másrészt Smith-prediktor alkalmazásával is megvalósítható. Mindkét esetben jó szimulációs eredmény érhető el [12]–[15].
- A virtuális oldalon áraminjektálással kompenzálható a teljesítményeltolás, ennek előnye, hogy az interfész dinamikai paramétereit nem befolyásolja a módszer, viszont a kompenzálás csak az implementált frekvenciákon történik.

A kis időállandós rendszerek vizsgálatához előnyös az FPGA alapú szimuláció, amely szimulációs időlépcsője $T_s \leq 1\mu s$. Az FPGA saját programozási nyelvén (általában VHDL) leírt kódok futtatására alkalmas, de ennek alkalmazása szakértelmet igényel. Az Opal-RT Technologies által fejlesztett eFPGASim eHS solver Matlab Simulink környezetben elkészített áramköri modellek szimulálását teszi lehetővé néhány száz nanoszekundumos időlépcsővel. Így nincs szükség speciális tudásra az FPGA adta lehetőségek kiaknázásához. Az eHS solver használatának hátulütője, hogy a be- és kimenetek nem szinkronizáltak a szimulációs időlépcsőhöz, a mintavételi időközök eltérhetnek és a kimenetre sem egyforma időközönként érkezik az elvárt jel („jitter” jelenség). A kis időlépcső miatt a szimulációs keretrendszer többi elemének késleltetése nem hanyagolható el, amelyek kumulált értékére egy intervallumot tudunk meghatározni. Az összegzett késleltetés nagysága a fentebb említett nehézségek ellenére jóval kisebb, mint a CPU solver-es implementációnál, így a stabilitás növelésére kisebb időállandójú szűrő vagy kisebb fizikai induktivitás is elegendő. A fázishiba a kis késleltetés miatt kevésbé jelentkezik, ezért kompenzálásra általában nincs szükség (és a pontos kompenzálás csak állandó értékű késleltetés esetén lenne lehetséges).

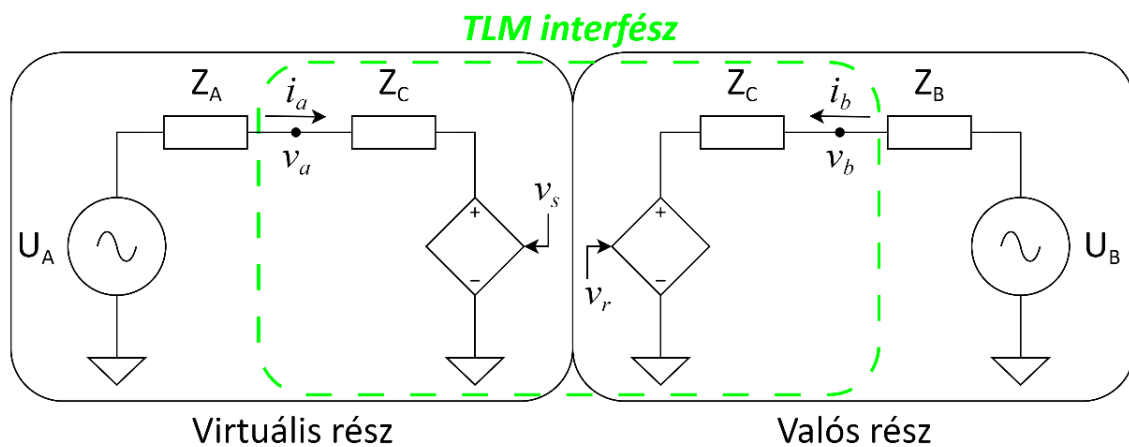
Az interfész pontosságát mindkét implementációs környezetben befolyásolják a virtuális és valós rész közt elhelyezett eszközök tulajdonságai. Az ITM algoritmus a pontatlanságok miatt nem megfelelően adja vissza a fizikai eszköz hatását a modellben. A szimulációs környezet építőelemeit tekintve a következő konzekvenciák vonhatóak le:

- A/D, D/A konverzió: A többi építőelemhez képest laboratóriumi környezetben elhanyagolható mértékű hibát okoz, nagy sávzélességű és mintavételi frekvenciájú kártyákat alkalmazunk, melyek felbontása is rendkívül jó.
- Lineáris teljesítményerősítő: A teljes kivezérlési tartományban kis mértékű hibával követi a bemeneti jelet. Amennyiben áttétel vagy ofszet hibával rendelkezik a készülék, úgy a hibák ismeretével kompenzálható.

- Áram- és feszültségmérő szenzorok: Általánosan a szenzorok a szimulációs környezet gyengepontjai. Az ofszet- és áttételhiba kompenzálásra szorul, amely további nehézsége ezen értékek hőmérsékletfüggése. A szenzor sávszélessége és egyben az áttétel frekvenciafüggése a gyors jelváltozások esetén csökkenti a pontosságot. A mérési zajok eliminálására hangsúlyt kell fektetni lehetőleg hardver szinten, de szoftveresen megvalósított szűréssel (akár átlagolással) is csökkenthető.
- Solver: A solver algoritmus okozta számítási pontatlanságok elhanyagolható mértékűek, azonban a mintavételi időközök nem egyformák, amely fázishibát okoz. A szimulációs időlépcsőben a mintavételezés pillanatszerű, így a mérési zajok hatása közvetlenül jelentkezik a szimulációban.

3. TLM – Transmission Line Model

A távvezeték modell (TLM) a Bergeron távvezeték modellen alapuló módszer, amely a hullámegyenletekből származtatott sorozatos hullámindításokkal képezi le a távvezeték két vége közti kapcsolatot. A következőkben gyakorlati megfontolásból a TLM feszültséggenerátoros változatát ismertetjük, amelyből analóg módon származik az áramgenerátoros felépítés. A TLM blokkvázlata a(z) 3. ábra szerinti. A virtuális és valós oldalon egy-egy Thevenin-modell kap helyet, amely Z_C helyettesítő impedanciájának értéke azonos a vezeték karakterisztikus impedanciájával. Veszteségmentes esetben a befutási idő egy szimulációs időlépcső nagyságú. A két feszültségforrás közti csatolást a(z) (3)(4) egyenletek írják le [2].



3. ábra: TLM algoritmus felépítése

$$v_r(k) = 2 \cdot v_a(k-1) - v_s(k-1) \quad (3)$$

$$v_s(k) = 2 \cdot v_b(k-1) - v_r(k-1) \quad (4)$$

A TLM nagy stabilitással rendelkezik még tranziensek esetén is, de elvéből kiindulva egy fizikai ellenállás beiktatása szükséges, amely csökkenti a teljesítményerősítő kivezérlési tartományát és jelentős veszteséget okoz [16].

A TLM algoritmusnak két megközelítési módja lehetséges:

- Első esetben az interfész algoritmus által leképezett távvezeték felhasználatuk a modellben a fizikai eszköz és a mögöttes hálózat közti kapcsolat kialakításában. Az interfész paraméterezésekor próbáljuk közelíteni a távvezeték jellemzőit. CPU-t

használó implementációs környezetben a szimulációs időlépcső korlátossága miatt rövid távvezetékek megvalósítása nem lehetséges.

- Második esetben csak interfész algoritmusként használjuk a TLM-et. A távvezeték jelleg miatt az átvitt teljesítménytől függő meddő teljesítmény termelés/fogyasztás hibát okoz. A távvezeték koncentrált paraméterű helyettesítése egy szűrőként jellemezhető, ezért a nagyobb frekvenciájú jeleket csillapítja.

A módszer PHIL alkalmazáson kívüli vonatkozása az ún. „stubline” modell, amely elve azonos a TLM-vel, de minden része szoftveres környezetben megvalósított. A stubline állapotterek szétcsatolására alkalmazható, így számítási erőforrás spórolható meg a mátrixműveleteken. A [17], [18] cikkek bizonyítják a TLM algoritmus elvének megalapozottságát és jó alkalmazhatóságát teljesen szoftveres környezetben.

A PHIL környezetben fellépő hibák a TLM algoritmus pontosságát csökkentik. A mintavételi időközök eltérése és a hardveres komponensek további késleltetései miatt a távvezeték befutási ideje nem lesz állandó értékű. A modellből indított hullám nem megfelelő időpillanatban érkezik a valós oldalra és a valós oldalon mért értékekből számolt hullám szintén nem lesz ideális. A késleltetés inkonzisztenciája az elvárttól eltérő állandósult állapotot alakít ki. A solver pillanatszerű mintavétele a zajérzékenységet növeli. A szenzorok hibái, a fizikai ellenállás bizonytalansága és hőmérsékletfüggése fokozza a pontatlanságot.

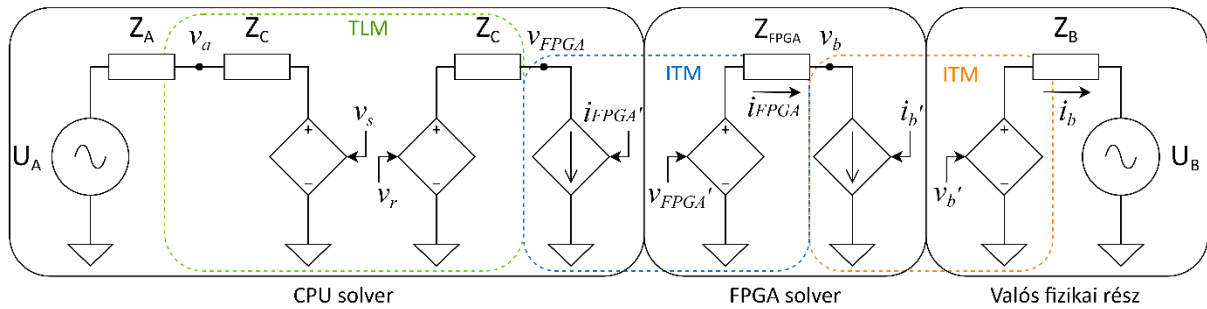
4. Többszintű interfész algoritmus

Az előző fejezetekben ismertettük az ITM és TLM algoritmusokat. Összegezve az ITM a szimulációs keretrendszer hibáinak ismeretében a kiegészítő és kompenzáló módszerekkel jól alkalmazható. Az FPGA solver-es megvalósítás további javulást biztosít a kis időlépcső által. Hátránya a virtuális és valós oldali impedanciák arányától függő stabilitás, amely még kiegészítő módszerekkel is limitációkat okoz. A TLM algoritmus ezzel szemben bármilyen impedancia viszony mellett stabil, de a késleltetésre, mérési hibára való érzékenység és a fizikai ellenállás miatt alkalmazása nehézkes.

Az interfész algoritmus tulajdonságai tovább javíthatóak több szinten kialakított interfész algoritmusok kombinálásával. Ez lehetővé teszi a módszerek előnyeinek egyesítését, amellyel bizonyítottan jobb eredmény érhető el, mint önálló esetben [19], [20]. Egy adott kialakításhoz használható azonos vagy több típusú interfész algoritmus, amelyek implementációs környezete rugalmasan változtatható. Például kétszintű ITM esetén két különböző szimulációs időlépcsővel futó részre bontjuk az interfész algoritmust. A külső kis időlépcsős általában FPGA-n futó ITM alakítja ki a tesztelt eszközzel a kapcsolatot. A nagyobb időlépcsővel CPU-n futtatott modell és az FPGA modell közé egy további ITM algoritmust helyezünk. Így egyesíteni tudjuk a két implementációs környezet előnyeit: a kis időlépcsős résszel a fizikai eszköz hatását megbízhatóan le tudjuk követni, míg a nagy időlépcsős szimulációban egy kiterjedtebb hálózatrész is leképezhető. A két külön időlépcsővel futtatott modell közt az interfész algoritmus szoftveres kialakítás által jobban kézben tartható, könnyebben és pontosabban lehet korrekciót vagy kompenzálást alkalmazni.

A(z) 4. ábra blokkvázlata több típusú interfész algoritmust kombináló megoldást szemléltet. A valós fizikai rész és az FPGA modell közt egy ITM algoritmust valósítunk meg, amely kis késleltetéssel rendelkezik, ezért a fázishiba hatása kevésbé jelentkezik és a tranzienst folyamatok jobban lekövethetőek. A gyors interfész precízebbé teszi a fizikai rész hatásának

megjelenítését a szoftveres világban. Az FPGA modellben a mögöttes hálózat impedanciájának egy részét elhelyezhetjük (Z_{FPGA}), de érdemes kis értéken tartani a nagy stabilitási tartomány érdekében. A CPU és FPGA solver határán egy második ITM interfész kap helyet. A CPU modellben a mögöttes hálózathoz egy TLM algoritmussal csatlakoztatjuk az FPGA-részt. A TLM teljesen szoftveres megvalósítása miatt a késleltetések és a mérések ideálisak, ezért a távvezeték modell mindig stabil lesz. A létrehozott három szintű interfésszel gyenge mögöttes hálózati szimulációk is végezhetőek, amelyek az ITM módszerrel biztosan stabilitási tartományon kívül esnének.



4. ábra: Többszintű interfész algoritmus

Az impedencialapú stabilitási feltétel két részre bontható. A külső ITM algoritmusra a(z) (5) egyenletnek minimálisan teljesülnie kell, továbbá a megjelenő késleltetést számításba kell vegyük, ami rontja a stabilitási tartományt. A belső, CPU-FPGA közti ITM algoritmusra a(z) (6) szerinti feltételnek kell teljesülnie.

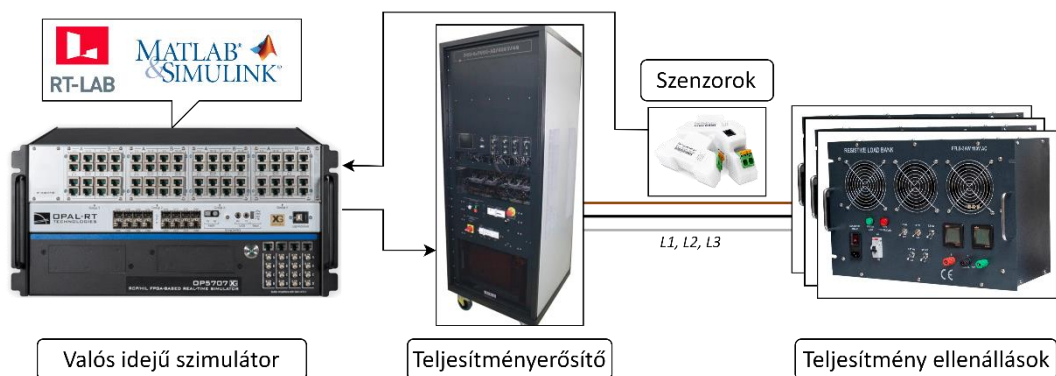
$$Z_{FPGA} \leq Z_B \quad (5)$$

$$Z_C + Z_{FPGA} \leq Z_B \quad (6)$$

A TLM algoritmus minden esetben stabil lesz, így a Z_A mögöttes hálózat impedanciája elhanyagolható a stabilitásvizsgálat során. A Z_C karakterisztikus impedancia közvetlenül befolyásolja a belső ITM algoritmus stabilitását. Ez limitálja a Z_C nagyságának szabad megválasztását, így a TLM algoritmus csak korlátozottan alkalmazható távvezeték reprezentálására.

5. Laborvizsgálatok

A laborszimulációk során az ITM és TLM algoritmusok stabilitását és pontosságát vizsgáltuk. A szimulációs keretrendszer blokkvázlatát az 5. ábra szemlélteti. A valós idejű szimulációt egy Opal-RT OP5707XG célhardveren futtattuk, jelen bemutatott eredmények esetében csak CPU alapú solvert használva. A teljesítményinterfészt egy háromfázisú lineáris teljesítményerősítő és Imperix gyártmányú szenzorok alkották. A fizikai eszköz fázisonként egy-egy fokozatonként kapcsolható teljesítményellenállás volt.



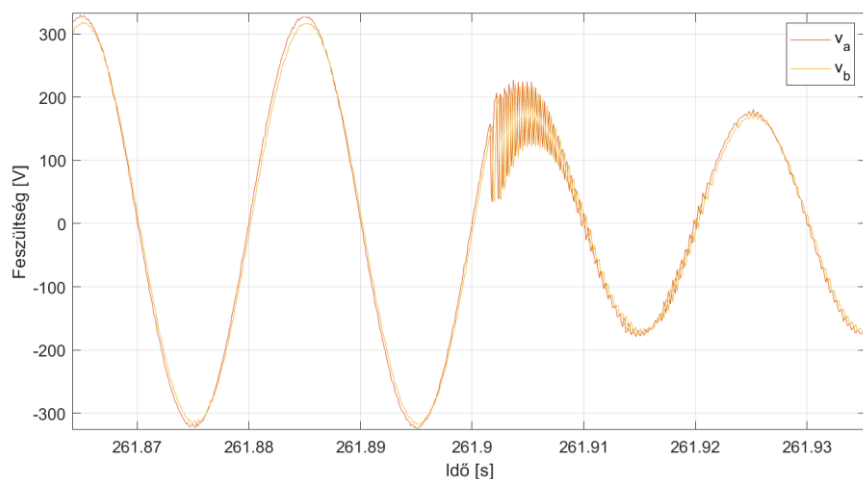
5. ábra: Laborszimulációs elrendezés

A mérési eredményeket az 1. táblázat foglalja össze. A stabilitásvizsgálat során a normál ITM algoritmust referenciaként vettük, így egységnyi stabilitással rendelkezik, amelyhez viszonyítva látható a többi módszer stabilitása. Az ITM kiegészítő módszerek minden esetben növelték a stabilitást, a fizikai induktivitás kisebb, míg az aluláteresztő szűrő és a dq-transzformáció nagyobb mértékben. A TLM algoritmus nem vált instabillá, ezzel igazolva előnyét. Az ITM algoritmusok amplitúdóhibái kis mértékűek, amelyekhez képest a TLM nagyobb pontatlanságot mutatott. A fázishiba az ITM módszereknél a késleltetésből adódik, amelyet az aluláteresztő szűrővel kiegészített esetben az aluláteresztő szűrő időállandója (fázishibája) növel. A TLM algoritmus fázishibája az interfészen átvitt teljesítmény arányában változott, a táblázatban látható eredmény a többi módszerrel azonos üzemállapot mellett adódott.

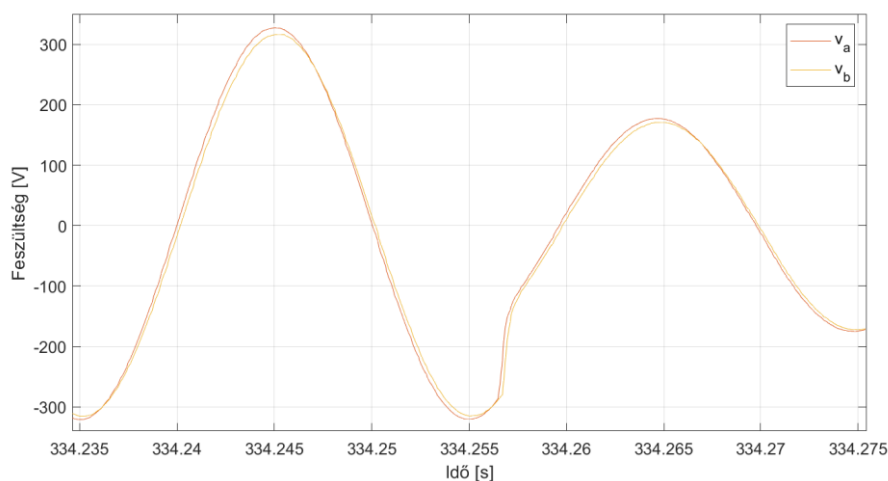
1. táblázat: Interfész algoritmusok stabilitása és pontossága

Interfész algoritmus	Stabilitás	Amplitúdóhiba	Fázishiba
ITM	1	1,228%	2,815°
ITM + aluláteresztő szűrő	5,5	0,491%	13,536°
ITM + fizikai induktivitás	1,12	1,242%	2,879°
ITM + dq-transzformáció	>5,5	1,244%	0,000°
TLM	nem vált instabillá	8,300%	10,067°

A stabilitást tranziens esetben is vizsgáltuk, a teljesítményellenállások bekapcsolásával tranziens jelenséget előidézve. A mögöttes hálózatot úgy hangoltuk, hogy stacioner esetben még épp stabil legyen a rendszer. A normál ITM módszert a 6. ábra, az aluláteresztő szűrővel kiegészített ITM algoritmust a 7. ábra szemlélteti. A stabilitást növelő módszer nélküli esetben egy lecsengő tranziens komponens látható, a növelt stabilitású esetben ez nem jelentkezik. Látható az aluláteresztő szűrő stabilitást növelő hatása, de a tranziens folyamatok vizsgálhatósága romlik.



6. ábra: Bekapcsolási tranziens normál ITM esetén



7. ábra: Bekapcsolási tranziens aluláteresztő szűrővel kiegészített ITM esetén

6. Összefoglalás, konklúzió

A PHIL szimulációs környezet számos alkotóelemmel rendelkezik, amelyek hatása együttesen befolyásolja a virtuális és valós rész közt kialakított kapcsolat minőségi mutatóit. Egy adott keretrendszerben az interfész algoritmust úgy kell megválasszuk, hogy a szimulációs követelmények teljesüljenek.

Az ITM egyszerű és robosztus megoldás, amely felhasználhatósága tovább javítható a kiegészítő és kompenzáló módszerekkel. Két implementációs környezetben: a CPU-t vagy FPGA-t használó solverok esetében bemutattuk az interfész kialakítását nehezítő hardveres és szoftveres problémákat. CPU solver esetén a késleltetés és a pillanatszerű mintavétel gondot okoz, amely hatások bizonyos szintig javíthatóak szoftveresen, de tranziens körülmények közt nem feltétlenül adnak jó eredményt. Az FPGA solverben leképezhető modell kiterjedtsége limitált, de a kisebb szimulációs időlépcsővel a tranziens folyamatok jobban lekövethetőek.

A TLM algoritmus numerikus módon reprezentálja a távvezeték működését. A PHIL környezetben való alkalmazásakor több hátránya van. A valós oldalon a fizikai ellenállás csökkenti a teljesítményerősítő kivezérlési tartományát és energiát diszipál. A késleltetések

inkonzisztenciája miatt a befutási idő változik, amely csökkenti a pontosságot. Végül a mérési zajokra mutatott érzékenysége jelentős. Azonban a távvezeték modell ideális megvalósítása mindig stabil, így szoftveres környezetben jól alkalmazható.

A többszintű interfész koncepció célja a módszerek előnyeinek egyesítése. A bemutatott kialakításban háromszintű interfész algoritmust ismertetünk, amellyel gyenge mögöttes hálózatu szimulációk is végezhetőek. A külső interfész kis időlépcsővel jól leképezi a valós részt a szimulációban. Ennek illesztése a nagyobb időlépcsővel futtatott részhez egy ITM és egy TLM algoritmussal történik. A TLM minden része szoftveresen implementált, ezért a stabilitási tartományt nem befolyásolja a mögöttes hálózati impedancia. A cikkben elméleti oldalról ismertettük a potenciális előnyökkel rendelkező többszintű interfész módszert. Ezzel kapcsolatos előzetes laborvizsgálatok stabilitás és pontosság szempontból is ígéretes eredményekkel szolgáltak, azonban további validáció szükséges. A jövőbeli tesztek során a többszintű struktúrával elért statikus és dinamikus tulajdonságok javulását teljesítményelektronikai berendezések interfész algoritmus szempontú vizsgálatával is szeretnénk alátámasztani.

Végül a valós PHIL szimulációs keretrendszerben elvégzett laborszimulációkkal szemléltettük az ITM és TLM algoritmusok előnyeit és hátrányait. Az ITM önálló és kiegészítő módszerekkel ellátott verziói is vizsgálatunk tárgyát képezte. Látható, hogy az ITM stabilitása limitált, amely kiegészítő módszerekkel bizonyos szintig javítható, de mindegyik kiegészítő módszer további hátrányokkal jár. Az aluláteresztő szűrővel kiegészített ITM fázishibát, a fizikai induktivitás minimális pontosságbeli romlást okoz, a dq-transzformáció pedig az átvitelt egy frekvenciára korlátozza. A TLM nagy fokú stabilitást biztosít, de pontossága elmarad az ITM-hez képest.

Irodalomjegyzék

- [1] T. Hatakeyama, A. Riccobono, and A. Monti, “Stability and accuracy analysis of power hardware in the loop system with different interface algorithms,” in *2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2016, pp. 1–8. doi: 10.1109/COMPEL.2016.7556671.
- [2] 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,” in *IEEE Transactions on Industry Applications*, 2008, pp. 1286–1294. doi: 10.1109/TIA.2008.926240.
- [3] M. G. Kashani, H. Pulakhandam, S. Bhattacharya, F. Katiraei, and D. Kaiser, “Design considerations and test setup assessment for power hardware in the loop testing,” in *2017 IEEE Industry Applications Society Annual Meeting*, 2017, pp. 1–8. doi: 10.1109/IAS.2017.8101770.
- [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)*, IEEE, 2019, pp. 1–6. doi: 10.1109/IYCE45807.2019.8991575.
- [5] G. Lauss, F. Lehfuß, A. Viehweider, and T. Strasser, “Power hardware in the loop simulation with feedback current filtering for electric systems,” in *IECON Proceedings*

- (*Industrial Electronics Conference*), 2011, pp. 3725–3730. doi: 10.1109/IECON.2011.6119915.
- [6] A. Markou, V. Kleftakis, P. Kotsampopoulos, and N. Hatziargyriou, “Improving existing methods for stable and more accurate Power Hardware-in-the-Loop experiments,” in *IEEE International Symposium on Industrial Electronics (ISIE)*, Institute of Electrical and Electronics Engineers Inc., Aug. 2017, pp. 496–502. doi: 10.1109/ISIE.2017.8001296.
 - [7] A. Summers, J. Hernandez-Alvidrez, R. Darbali-Zamora, M. J. Reno, J. Johnson, and N. S. Gurule, “Comparison of Ideal Transformer Method and Damping Impedance Method for PV Power-Hardware-In-The-Loop Experiments,” in *IEEE 46th Photovoltaic Specialists Conference (PVSC)*, IEEE, 2019, pp. 2989–2996. doi: 10.1109/PVSC40753.2019.8980690.
 - [8] H.-J. Heo, C.-H. Park, and J.-M. Kim, “Modified Interface Algorithm of PHIL Simulator to Improve Harmonic Current Accuracy,” in *IEEE PELS Workshop on Emerging Technologies: Wireless Power Transfer (WoW)*, 2020, pp. 304–308. doi: 10.1109/WoW47795.2020.9291318.
 - [9] 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.
 - [10] A. Castro, P. Zuniga, F. A. Uribe, and E. Barocio, “Phase compensation scheme to improve the accuracy of a power hardware-in-the-loop experiment based on a synchronous generator,” in *2015 IEEE Eindhoven PowerTech*, 2015, pp. 1–6. doi: 10.1109/PTC.2015.7232517.
 - [11] W. Ren, M. Steurer, and S. Woodruff, “Applying Controller and Power Hardware-in-the-Loop Simulation in Designing and Prototyping Apparatuses for Future All Electric Ship,” in *2007 IEEE Electric Ship Technologies Symposium*, 2007, pp. 443–448. doi: 10.1109/ESTS.2007.372124.
 - [12] B. Lundstrom and M. V. Salapaka, “Optimal Power Hardware-in-the-Loop Interfacing: Applying Modern Control for Design and Verification of High-Accuracy Interfaces,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 11, pp. 10388–10399, Nov. 2021, doi: 10.1109/TIE.2020.3032918.
 - [13] Z. Feng, R. Pena-Alzola, M. H. Syed, P. J. Norman, and G. M. Burt, “Adaptive Smith Predictor for Enhanced Stability of Power Hardware-in-the-Loop Setups,” *IEEE Transactions on Industrial Electronics*, vol. 70, no. 10, pp. 10204–10214, Oct. 2023, doi: 10.1109/TIE.2022.3224196.
 - [14] E. Guillo-Sansano, A. J. Roscoe, C. E. Jones, and G. M. Burt, “A new control method for the power interface in power hardware-in-the-loop simulation to compensate for the time delay,” in *Proceedings of the Universities Power Engineering Conference (UPEC)*, IEEE Computer Society, Oct. 2014, pp. 1–5. doi: 10.1109/UPEC.2014.6934618.

- [15] E. Guillo-Sansano, A. J. Roscoe, and G. M. Burt, "Harmonic-by-harmonic time delay compensation method for PHIL simulation of low impedance power systems," in *2015 International Symposium on Smart Electric Distribution Systems and Technologies (EDST)*, 2015, pp. 560–565. doi: 10.1109/SEDST.2015.7315271.
- [16] 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.
- [17] B. Ahmed, A. Abdelgadir, N. A. Saied, and A. A. Karrar, "A Compensated Distributed-Parameter Line Decoupling Approach for Real Time Applications," *IEEE Trans Smart Grid*, vol. 12, no. 2, pp. 1761–1771, Mar. 2021, doi: 10.1109/TSG.2020.3033145.
- [18] X. Zhai *et al.*, "Multi-rate real-time simulation of modular multilevel converter for HVDC grids application," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 1325–1330. doi: 10.1109/IECON.2017.8216225.
- [19] G. F. Lauss, "Interfacing challenges in PHIL simulations for investigations on P-Q controls of grid connected generation units in electric power systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10964–10970, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.2468.
- [20] F. Ashrafidehkordi and G. De Carne, "Improved Accuracy of the Power Hardware-in-the-Loop Modeling using Multirate Discrete Domain," in *IEEE 13th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, 2022, pp. 1–5. doi: 10.1109/PEDG54999.2022.9923128.