



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Elektronikus Eszközök Tanszéke

Bevezető az LTspice XXIV szimulációs program használatába

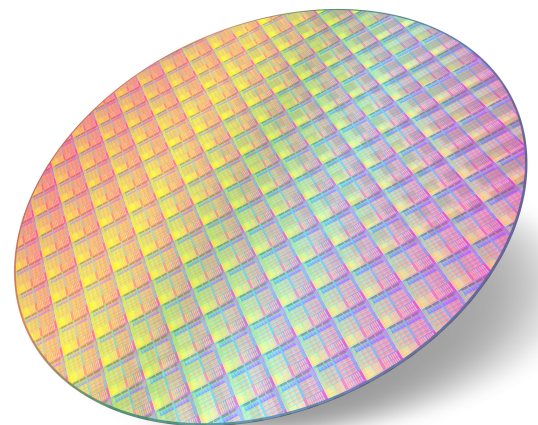
Készítette

Deák E. Dávid
Dr. Bognár György

Lektorálta

Dr. Plesz Balázs

2024 Augusztus
v1.0



Tartalomjegyzék

1. Bevezető	3
2. LTspice telepítése	3
2.1. Windows és OSX	4
2.2. Linux	4
3. Áramkörök összeállítása	6
3.1. Sémakerkesztő használata	6
3.2. Független feszültség- és áramforrás (V és I források)	7
3.3. Vezérelt feszültség- és áramforrások (E,F,G és H források)	9
3.4. Tetszőleges viselkedésű források (BV és BI források)	11
3.5. Hierarchikus áramköri blokkok	12
4. SPICE szimulációs direktívák	14
4.1. Munkapont meghatározása és a .op direktíva	14
4.2. Forrásparaméterek változtatása és a .dc direktíva	15
4.3. Kisjelű analízis és a .ac direktíva	16
4.4. Nagyjelű analízis és a .tran direktíva	18
4.5. Zajszimulációk és a .noise direktíva	22
5. Összetett szimulációk futtatása	23
5.1. Parametrikus szimulációk	23
5.2. Monte Carlo szimulációk	25
5.3. Áramkör vizsgálata diszkrét frekvenciákon	26
5.4. Kétpólus és kétkapu paraméterek számítása	27
6. Szimulációs eredmények vizualizációja és utófeldolgozása	29
6.1. Hullámformák ábrázolása	29
6.2. Fourier-transzformációs analízis	32
6.3. Mérések elvégzése és a .measure direktíva	34
7. Modellfájlok felhasználása	36
7.1. Modellfájlok importálása	37
7.2. Eszközkönyvtárak importálása	38
7.3. Eszközkönyvtárak készítése	38
8. Összefoglalás	41
A. Használt gyorsgombok listája	42
B. PDF-he csatolt fájlok listája	43

1. Bevezető

Az LTspice az elektronikai iparágban az egyik legelterjedtebb¹ SPICE (Simulation Program with Integrated Circuit Emphasis) alapú, analóg áramkörök szimulálására szolgáló szoftver. A SPICE programcsalád fejlesztése az 1970-es években kezdődött a Berkeley egyetemen a SPICE1 programmal, de igazi sikereket a SPICE2 1975-s debütálása hozta. Ennek következtében a SPICE betűszó elválaszthatatlanul összeforrt az áramkör-szimulátorok világával. Jelenleg a szabványosított, ipari áramkörtervezési módszertanok szerves részét képezi a tervezett áramkör viselkedésének szimulálása SPICE-alapú szoftverek segítségével.

A SPICE programok elterjedését további változatok is segítették, melyek mind a mai napig megtalálhatók a piacon. Ilyen programok pl. a PSPICE (jelenleg a CADENCE birtokolja), a HSPICE, a TINA (Toolkit for Interactive Network Analysis), amit a budapesti DesignSoft cég készített el 1990-ben, valamint az **LTspice** is. Ezen szimulátoroknak közös tulajdonsága, hogy az áramköri leírást egy ún. **NETLIST**, vagy magyarul netlista fájl tartalmazza. Ebben a fájlban szövegesen és szabványosított formátumban található meg az áramköri komponensek, azok paramétereit, az áramköri összeköttetések leírása (az alkatrész melyik lába melyik csomóponthoz kapcsolódik), valamint az esetleges egyéb kulcsszavak és szimulációs direktívák. Ezeket a netlista állományokat beolvasva végzik el a szimulációs motorok a szimulációkat. A könnyebb kezelhetőség érdekében a modernebb szoftverek egy grafikus kezelőfelülettel fedik el a szöveges fájlokat és a háttérben futó szimulációs motorokat.

Az LTspice nevét a Linear Technology cégről kapta, mely eredetileg hozta a piacra 1999-ben. A cég a szoftvert ingyenesen (freeware-ként) és korlátozások nélkül² dobta piacra, mely nagyban elősegítette az LTspice elterjedését. 2017-ben a céget felvásárolta az Analog Devices közel 15 milliárd amerikai dollárért, azonban az akvizíciót követően sem változott a szoftver freeware jellege! A szoftver hosszabb fejlesztési ciklussal rendelkezik, így nem ritka, hogy a kiadott verziók között akár 5-10 év is eltelhet. Ennek következtében valószínűsíthető, hogy a jelenlegi XXIV változat és a hozzá íródott segédlet is aktuális marad.

A legfrissebb elérhető változat az alapszoftveren felül számos beépített modellfájljal érkezik az Analog Devices, a Linear Technology és a Maxim Integrated cégek áramköri kínálatából. Ezen felül számos edukatív áramköri és szimulációs példa is található az alap installáció mellett, melyek remekül demonstrálják a szoftverben rejlő lehetőségeket. Ennek megfelelően buzdítjuk az olvasót az egyes funkciók mélyeb megértésére és a hozzájuk kapcsolódó tesztáramkörök kipróbálására!

Manapság az LTspice-t előszeretettel alkalmazzák a nagyfrekvenciás elektronika, a kapcsolóüzemű tápegységek, az audiótechnika és a teljesítményelektronika területein. A széleskörű elterjedésének és a szimulátorban rejlő lehetőségek miatt életszerűnek tartjuk a szoftver használatát és ennek megfelelően íródott ez a segédlet, mellyel reméljük a szoftver használata könnyebbé és érthetőbbé válik majd! Fontos megjegyezni, hogy ezen segédlet, ugyan törekszik a részletességre, de nem kíván az LTspice teljeskörű magyar nyelvű dokumentációjaként szolgálni. Ebben a dokumentumban részletezett információkon felül javasoljuk az olvasónak az LTspice angol nyelvű súgójának használatát (**F1 gyorsgomb**), valamint a hivatalos súgóban nem dokumentált funkciókhoz a szintén angol nyelvű [wikipedia](#) oldalt, mint referencia. Meg kell ugyanakkor jegyezni, hogy ezen dokumentációk kissé szűkszavúan fogalmaznak, valamint a komponensek és funkciók használatát elsősorban a netlistában szereplő formátumban közlik, így a kezdő felhasználók számára nehezebben értelmezhetőek! Ezen hiányosságok miatt is ajánljuk jelen segédlet használatát!

2. LTspice telepítése

Az LTspice telepítéséhez látogassunk el az [Analog Devices](#) honlapjára és töltsük le az operációs rendszerünknek megfelelő változatot! A jelenleg támogatott operációs rendszerek a Windows 10-es 64bit-es és ennél frissebb operációs rendszerek, valamint az OSX 10.15-ös és frissebb változatai.

¹Mike Engelhardt gondolatai

²Számos szoftver esetében az ingyenes változat csak limitált módon működik. Például korlátozzák a szimulálható csomópontok számát, vagy korlátozzák a felhasználható erőforrások mennyiségét. Nincs ez így az LTspice esetében, ahol a szoftvernek csakis korlátozások nélküli és ingyenes változata létezik!

Linux rendszerre történő telepítéshez szükséges egy Windows emulátor. Ennek a folyamatát részletesen taglaljuk a későbbiekben.

2.1. Windows és OSX

Windowsra történő telepítéshez csak a letöltött **msi** fájlt kell futtatni és beállítani a telepítési állományok elérési útját, ami alapértelmezetten a **C:\Program Files\ADI\LTspice** mappa, ha a szoftvert a számítógépünk összes felhasználójának telepítjük. Abban az esetben, ha csak az aktuális felhasználónk számára szeretnénk telepíteni, úgy az alapértelmezett telepítési út a **C:\users\USERNAME\AppData\Local\Programs\ADI\LTspice** mappa lesz. Figyelem az elérési útban szereplő **USERNAME** almappa lecserélendő az aktuális felhasználó felhasználónevére! Ebben a segédletben a továbbiakban az **előbbi** módon telepített változatra közöljük az elérési utakat. Ettől eltérő esetben az elérési utakat módosítani kell!

A telepítés utáni első futtatáskor az LTspice jelezheti, hogy az alkatrészkönyvtár régóta nem volt frissítve és lehetőséget biztosít ennek megoldására. Ha a frissítés mellett döntünk egy rövidebb-hosszabb telepítési folyamat indul meg, mely során a hivatalos online forrásokból a program letölti a szükséges modellfájlokat és integrálja azokat a telepített példány könyvtáraiba.

OSX operációs rendszer esetén a folyamat hasonlóan követhető le, azonban az alapértelmezett elérési út más lesz.

2.2. Linux

Linux rendszer esetében is lehetőségünk van az LTspice futtatására egy Windows kompatibilitási réteg telepítésével. Ebben a dokumentumban az egyik legelterjedtebb programot a **wine**-t fogjuk használni. Linux esetében a csomagkezelő disztribúciónként változik, így itt csak **Ubuntu** operációs rendszerhez és az **apt-get** csomagkezelőhöz tartozó esetet részletezzük. A telepítéshez ajánlott a Wine legfrissebb változatát telepíteni, azonban az nem található meg az Ubuntu linux repozitóriumában, így azt nekünk kell hozzáadni. Ehhez futtassuk le a következő parancsokat a terminálból, ahogy az a **Wine** hivatalos oldalán szerepel:

```
user@computer:~$ sudo dpkg --add-architecture i386
user@computer:~$ sudo mkdir -pm755 /etc/apt/keyrings
user@computer:~$ sudo wget -O /etc/apt/keyrings/winehq-archive.key https...
```

A parancsok segítségével engedélyezzük a csomagkezelőnk számára az i386-os, azaz a 32-bites alkalmazások telepítését. Windows-os applikációk esetén sok program még mindig használ 32-bites könyvtárakat, így szükséges ezek engedélyezése, különben az LTspice nem fog futni. Az utolsó két parancs hozzáadja a szükséges digitális aláírási kulcsokat a kulcskarikánkhoz.

Ezután adjuk hozzá a disztribúciónknak megfelelő forrásfájl állományt a forrásfájlok listájához. Esetünkben az Ubuntu 22.04.4-es LTS változatához megfelelően a Jammy Jellyfish forrásállományt kell kiválasztanunk.

```
user@computer:~$ sudo wget -NP /etc/apt/sources.list.d/ https...
```

Frissítsük ezután a lokális repozitóriumaink listáját, és telepítsük a legfrissebb Wine stabil verzióját:

```
user@computer:~$ sudo apt-get update
user@computer:~$ sudo apt-get upgrade
user@computer:~$ sudo apt-get install --install-recommends winehq-stable
```

A wine telepítése után le kell töltenünk a Windows-os telepítő **msi** fájlt az [Analog Devices](#) honlapjáról. Ezután a terminál ablakban el kell navigálnunk a letöltött fájlt tartalmazó mappába és el kell indítsuk a telepítést wine-t használva.

```
user@computer:~$ cd Downloads
user@computer:~/Downloads$ wine msiexec /i LTspice64.msi
```

Ezután a Windows-os esethez hasonlóan kell folytatni a telepítést. Mivel egy virtualizációs rétegen keresztül telepítettük az LTspice-t, így a programindító **exe** fájl a következő mappában található:

```
~/.wine/drive_c/Program\ Files/ADI/LTspice
```

Linux alatt az exe fájlok nem futtathatóak, azonban a **wine**-t felhasználva elindíthatjuk őket egy kompatibilitási rétegen keresztül. Ehhez a következő parancsot kell futtassuk:

```
wine ~/.wine/drive_c/Program\ Files/ADI/LTspice/LTspice.exe
```

A parancs futtatását megkönnyítendő létrehozhatunk egy alias-t, mellyel egy könnyebben megjegyezhető kulcsszóval tudjuk elindítani az LTspice-t. Ehhez nyissuk meg a **.bash_aliases** fájlt kedvenc szövegszerkesztőnkkel.

```
user@computer:~/Downloads$ nano ~/.bash_aliases
```

Ebbe a file-ba illesszük be a következő sort, hogy hozzáadjunk egy alias-t a shell környezetünkhöz:

```
alias ltspice='wine ~/.wine/drive_c/Program\ Files/ADI/LTspice/LTspice.exe&'
```

A fájl mentését követően már csak be kell töltenünk a shell környezetbe annak tartalmát:

```
user@computer:~/Downloads$ source ~/.bash_aliases
```

Ezután az LTspice az **ltspice** parancssal futtathatóvá válik a terminál felületünkből. Az első futtatás alkalmával itt is érdemes frissíteni a modellfájlokat, ahogy a program azt fel is ajánlhatja nekünk.

A fent leírt telepítési folyamatot helyettesíthetjük a PDF-hez csatolt³ **UbuntuInstall.sh** script fájl futtatásával majd a fenti source parancs hívásával.

Sajnálatos módon a Wine alapértelmezett témája megrekedt a Windows 95 világában, így a szemnek kényelmesebb használathoz javasoljuk annak átállítását. Ehhez keressünk egy szimpatikus témát [ezen az oldalon](#). Az új téma alkalmazásához futtatnunk kell a wine konfigurációs parancsát:

```
user@computer:~/Downloads$ winecfg
```

Ezután a Desktop Integration fül alatt kattintsunk az Install theme... parancsra és válasszuk ki a letöltött témánkhöz tartozó **.msstyle** fájlt (A szerzők a **gelxp2.msstyle** használatát javasolják). Telepítés után a Theme legördülő menüből ki is tudjuk választani a frissen telepített témánkat, és témától függően még további beállításokat is tehetünk.

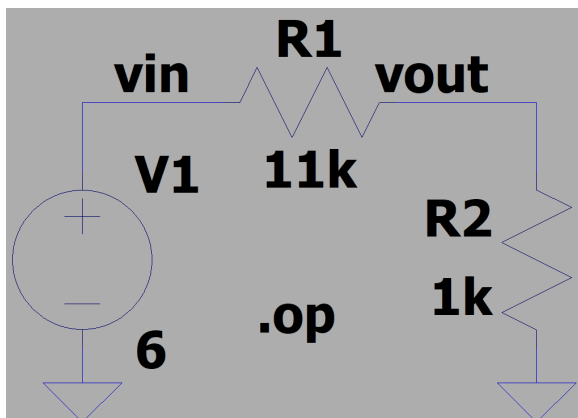
³A PDF 1.3-as verziójától kezdve lehetőség nyílik tetszőleges fájl PDF dokumentumhoz csatolására. Ehhez a PDF dokumentumot egy PDF 1.3-as verziót beolvasni képes programmal kell megnyissuk. Ilyen program pl. Adobe Acrobat vagy az Okular.

3. Áramkörök összeállítása

Az LTspice-t elindítva egy egyszerű és modern kezelőfelület tárul elénk. Ez a modern kezelőfelület az LTspice legutóbbi (XXIV) változatával frissült meg. Előző verziók esetében a kezelőfelület kinézete a Windows 95 kinézetét tükrözte, valamint az alapértelmezett gyorsgombok is ezt tükrözték. A tradicionális LTspice gyorsgombok nagyrészt az F1-9 billentyűket használják. Az új változatban a program futása előtt felkínálja nekünk a választási lehetőséget a modern és a tradicionális gyorsgombok használatához. Természetesen van lehetőségünk a gyorsgombok megváltoztatására a **(Tools > Settings > Schematic > Keyboard shortcuts)** felületen keresztül. Az egyszerűbb beállítás érdekében ehhez a pdf-hez csatolva megtalálható egy **shortcuts.txt** fájl, amely egyszerűen importálható az LTspice-ba.

3.1. Sémakerkesztő használata

Áramkörök szimulálásához először meg kell rajzolnunk őket, ehhez pedig egy új sémát kell nyitnunk **(File > New schematic)** vagy a **Ctrl+N** gyorsgomb megnyomásával. A megnyíló szürke felületen tudunk áramköri komponenseket lehelyezni és áramköri összeköttetéseket készíteni, valamint a komponensek paramétereit beállítani. A fontosabb gyorsgombok listáját az A. függelék tartalmazza. Első példaként nézzük az 1. ábrán látható feszültségosztó kapcsolást.



1. ábra. Egy példakapcsolás áramköri rajza

--- Operating Point ---		
V(vin):	6	voltage
V(vout):	0.5	voltage
I(R1):	-0.0005	device_current
I(R2):	-0.0005	device_current
I(V1):	-0.0005	device_current

2. ábra. Szimulációs eredmények

Áramkörök megrajzolásához célszerű először a szükséges komponenseket lehelyezni. Ehhez a **(Edit > Component)** menüpontot vagy a **P** gyorsgombot használhatjuk. A kiválasztott alkatrészt a **CTRL+R** gyorsgomb segítségével forgathatjuk lehelyezés előtt, valamint a **CTRL+E** gyorsgombbal tükrözhetjük. Az alkatrészek elhelyezése után célszerű beállítani az alkatrészparamétereiket. Ehhez az áramköri komponensekre jobb egérgombbal kattintva egy felugró ablakban meg tudjuk adni a szükséges paramétereiket (pl. feszültségforrás feszültségét és az ellenállások értékét stb.). Az LTspice felismeri az SI prefixumokat valamint az egyszerűsített jelölést is. Ennek megfelelően egy 3300 Ω -s ellenállásértéket megadhatunk **3.3k** és **3k3** módon is. Egyetlen fontos megjegyzés, hogy a prefixumokat kis- és nagybetűre érzéketlenül kezeli a szoftver, így a mega és a milli prefixumok megkülönböztetése végett a mega prefixumot **meg** rövidítéssel kell megadni!

Ezt a lépést követően az összeköttetéseket célszerű kialakítani. Ehhez egyszerű és könnyen átlátható séma esetén a **wire** eszközt célszerű használni **(Edit > Draw Wire)** avagy **W** gyorsgomb. A vezetékvezést alapértelmezetten 90°-s kanyarok alkalmazásával hozhatjuk létre. Ha ettől szeretnénk eltérni, úgy a vezetékszakasz lerakása közben a **CTRL** gombot lenyomva tetszőleges meredekségű vezeték húzható. Vezetékszakaszok között jobb egérgombot nyomva megszakíthatjuk a vezetékvezés folyamatát, de még mindig a vezeték szerkesztő eszköz marad aktív.

Összetettebb vezetékvezés esetén célszerű lehet az összeköttetéseket csupán a vezetékek címkézésével kialakítani. Ehhez a **(Edit > Label Net)** opciót vagy a **N** gyorsgombot kell kiválasszuk. Ekkor a felugró szövegdobozba beírva a vezeték nevét és azt hozzáillesztve a sémán lévő egyik vezetékdarabhoz vagy áramköri

pólushoz létrehozhatunk elektromos kapcsolatokat. Ezzel a módszerrel átláthatóbb és könnyebben olvasható lesz a séma. A módszer másik előnye, hogy az így elnevezett vezetékek és csomópontok a hullámformák megjelenítése során is az általunk megadott nevekké lesznek beazonosíthatók.

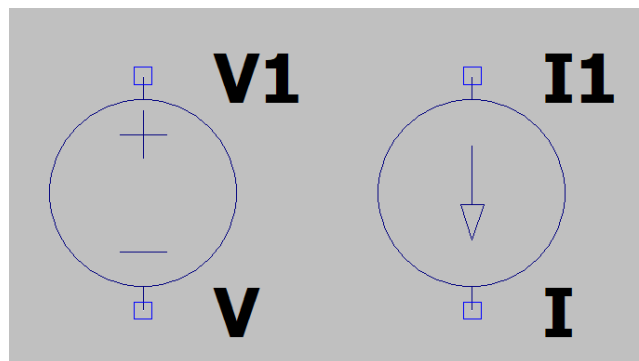
A sémán történő navigáláshoz ki kell lépni az aktuális szerkesztőeszközből (pl. vezetékvezése esetén a vezetékvező eszköz), ehhez az **ESC** gombot kell megnyomni. Ekkor lehetőségünk nyílik a séma fókuszban lévő részének mozgatására a bal egérgomb lenyomásával történő egérmozdulatok segítségével. Ha szeretnénk a séma egy távolabbi részére navigálni, akkor célszerű lehet a teljes sémát a fókuszba hozni a **SPACE** billentyű lenyomásával majd az ezt követően a fókusz nagyításával az **egérgörgő** segítségével.

Az áramkör megrajzolása után el kell döntenünk, hogy milyen szimulációval szeretnénk vizsgálni a kapcsolást. Ennek megfelelően el kell helyezni egy szimulációs direktívát a kapcsolási rajzon (**Simulate > Configure Analysis**), vagy az **.** billentyű lenyomásával direkt is be tudjuk gépelni a szükséges parancsokat. A szimulációs parancsokról bővebben is értekezünk a 4. fejezetben. Ha megrajzoljuk az 1. ábra kapcsolását és meghatározzuk a kapcsolat munkapontját, vagyis azokat a csomóponti feszültségeket és áramokat melyek mellett a hálózati egyenletek teljesülnek, úgy a 2. ábra eredményeit kapjuk. Látható, hogy a kimeneti csomópont feszültsége 0,5 V és az ellenállásokon, valamint a forráson 5 mA áram folyik, ahogy azt el is várjuk egy $11k\Omega$ -s és $1k\Omega$ -s ellenállásból álló feszültségosztó esetén.

A következőkben az egyes áramköri komponensek részletesebb bemutatásával és a paramétereinek értelmezésével fogunk foglalkozni.

3.2. Független feszültség- és áramforrás (V és I források)

Az egyik leggyakrabban használt áramköri elemek a független feszültség- és áramforrások, melyek LTspice-beli áramköri jele a 3. ábrán látható. Feszültségforrást "voltage", míg áramforrást "current" nevek alatt fogjuk megtalálni az (**Edit > Component**) vagy **P** gyorsgomb lenyomásával.

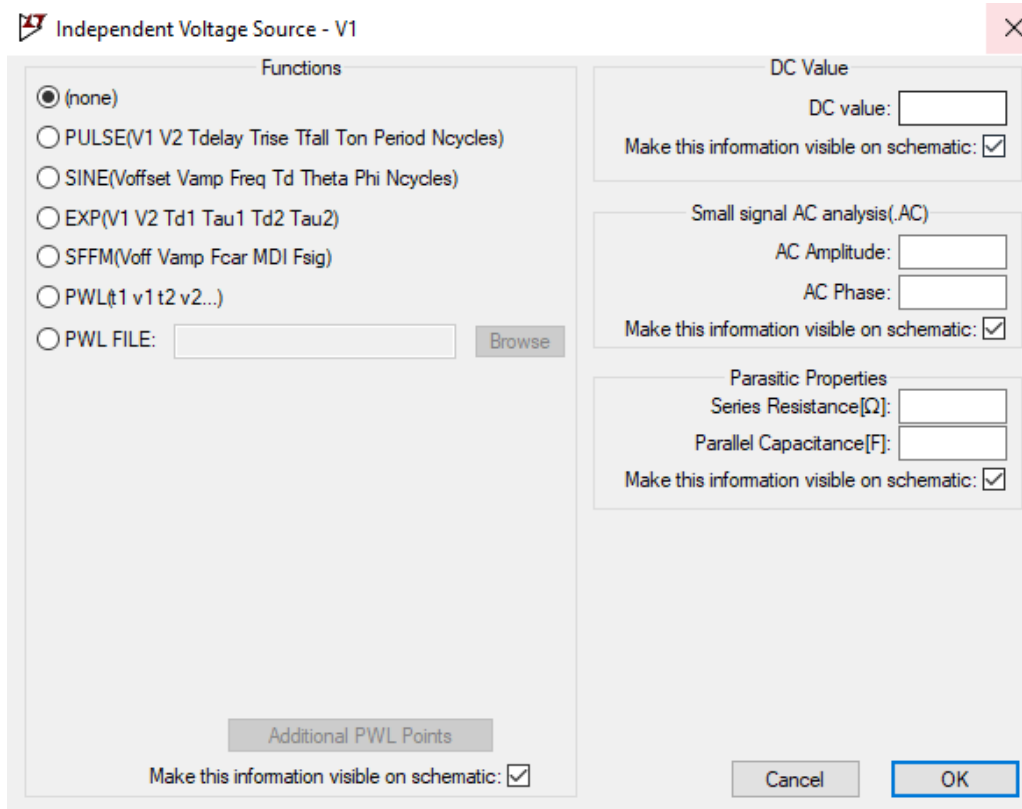


3. ábra. Független feszültség- és áramforrás áramköri jele

A forrásokra jobb egérgombbal kattintva be tudjuk állítani paramétereiket. Feszültségforrás esetében egy állandó feszültséget és egy soros ellenállást tudunk megadni, ezáltal egy Thevenin forrást tudunk modellezni. Áramforrások esetében alapértelmezetten csak az állandó áramértéket tudjuk beállítani. Fontos megjegyezni, hogy az LTspice használata során **soha sem kell mértékegységet megadnunk**, a paraméterek a megfelelő mértékegységben vannak értelmezve. Azonban azt is fontos megjegyezni, hogy az LTspice ismeri az SI prefixumokat⁴, így ha szeretnénk egy 100 mV-os és 1 k Ω -s belső ellenállással rendelkező feszültségforrást létrehozni, úgy a 100m-es és 1k-s értékeket kell begépeljünk a megfelelő helyekre (DC value és Series resistance mezőkbe).

Ezeket a forrásokat felhasználva elő tudunk állítani összetettebb jeleket is, ha a források beállítási lehetőségeinél az **Advanced** opcióra kattintunk. Mivel a két forrás nagyon hasonló, így csak a feszültségforrást mutatjuk be részletesebben, az áramforrás esetén analóg módon értelmezettek a paraméterek.

⁴Ez alól csak a **M** kivétel, hiszen a program a netlistákat kis- és nagybetűre érzéketlenül olvassa be. Annak érdekében, hogy a **Mega** prefixum is értelmezve legyen, az LTspice a **meg** rövidítést használja.



4. ábra. Független feszültségforrás kibővített paraméterlistája

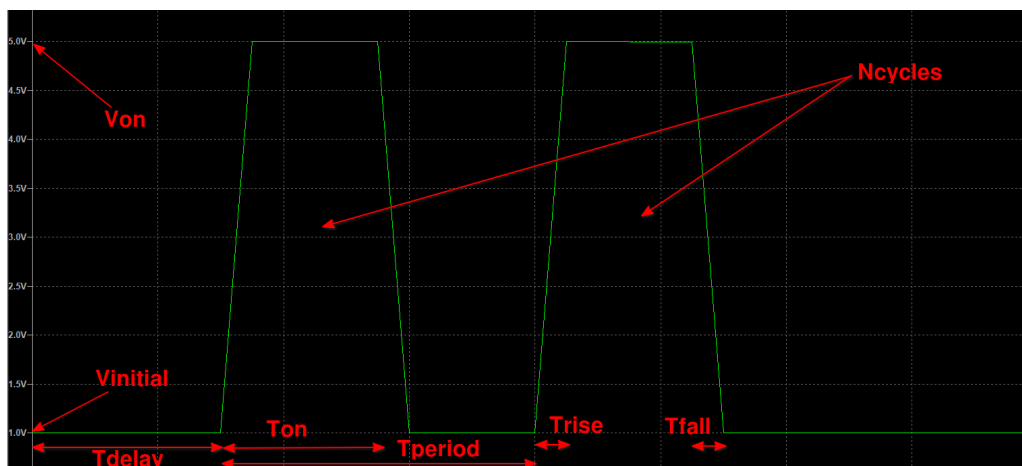
A forrás paramétereinek között a jobb felső sarokban az állandósult értéket tudjuk beállítani (DC value), alatta a kisjelű paramétereket tudjuk megadni, melyek az **AC** szimuláció során lesznek felhasználva. Lásd később a 4. fejezetben. A kisjelű paraméterek alatt a forrás parazitáit tudjuk beállítani. Feszültségforrás esetében soros ellenállást és párhuzamos kapacitást tudunk modellezni.

A paraméterek között bal oldalon lévő opciókkal időben változó forrásokat tudunk definiálni. Ezekből az opciókból leggyakrabban a PULSE, SINE és PWL opciókat használjuk, így ebben a dokumentumban csak ezeket mutatjuk be.

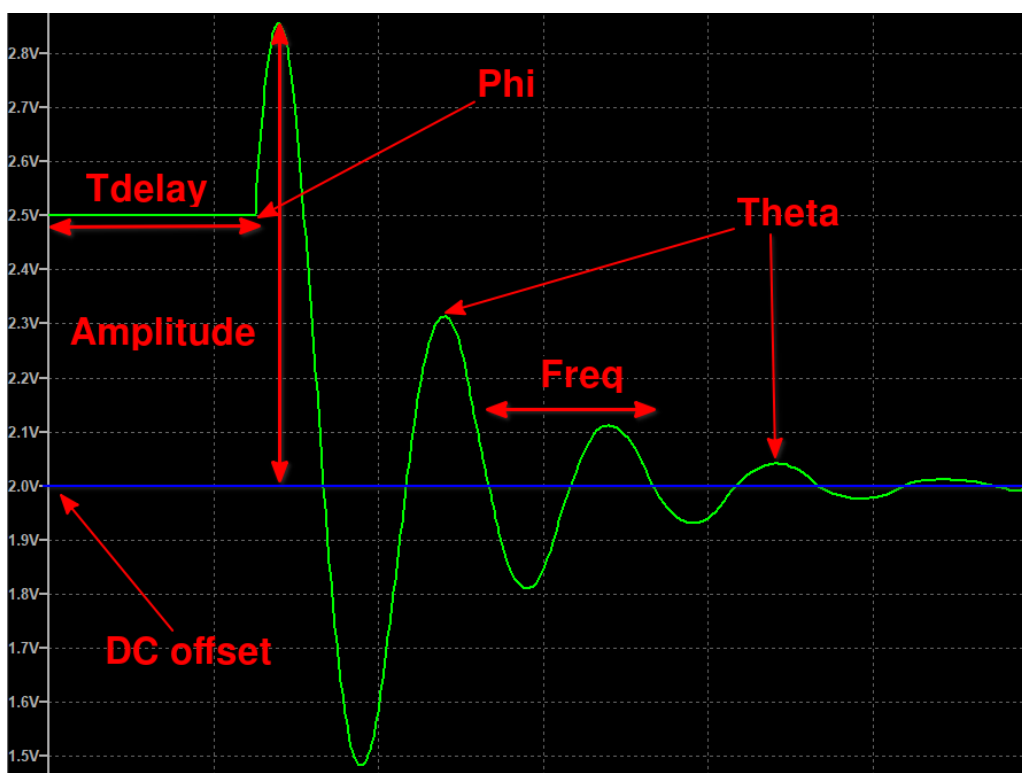
Pulse opciót választva egy általánosított négyszögjelet tudunk létrehozni. Az általánosítás a véges fel- és lefutási időkből ered, melynek következtében a jelalak trapéz jellegű lesz. Az egyes paraméterek értelmezése az 5. ábrán látható. Az egyes paraméterek magyarázatai: **Vinitial** → kezdeti feszültség szint, **Von** → bekapcsolt állapot feszültség szintje, **Tdelay** → időkéscsúszás a bekapcsolásig, **Trise** → felfutási idő, **Tfall** → lefutási idő, **Ton** → bekapcsolt állapotban lévő idő, **Tperiod** → periódusidő és **Ncycles** → impulzusok száma. Ha a paramétereknek nem adunk értéket akkor azok figyelmen kívül lesznek hagyva a szimulációkból.

Sine forrás esetében egy általános szinuszhullámot tudunk megadni. a 6. ábrán egy példa hullámot láthatunk a paraméterek egy adott értékeire. Az egyes paraméterek magyarázatai: **DC offset** → a jel DC középpértéke, **Amplitude** → a jel amplitúdója, **Freq** → a jel frekvenciája, **Theta** → a jel csillapítási tényezője, **Phi** → a jel kezdőfázisa. A Tdelay és Ncycles paraméterek a Pulse forrásnál tárgyaltakkal azonosak. a 6. ábrán egy 2 V középpértékű és 1 V amplitúdóval rendelkező csillapodó szinuszjelet jelenítettünk meg. Mivel a jel késleltetve van, így a jel belépése előtt a kezdeti értékét veszi fel. Az ábrán jól megfigyelhető, hogy a jel már az első periódusában sem veszi fel a 3 V-os értékét (2 V-os középpérték plusz 1 V-os amplitúdó), ez a jelenség a jelentős csillapítási faktornak tudható be.

PWL forrás esetében lehetőségünk van a feszültség időfüggvényét tetszőleges időpontokban megadni. A nem definiált időpillanatokban a feszültségforrás szomszédos időpillanatai közötti lineáris interpolációval közelített értékeket kapjuk. Innen ered a forrás neve is, hiszen a PWL az angol piecewise linear, azaz szakaszonként lineáris,



5. ábra. PULSE forrás jelalakja

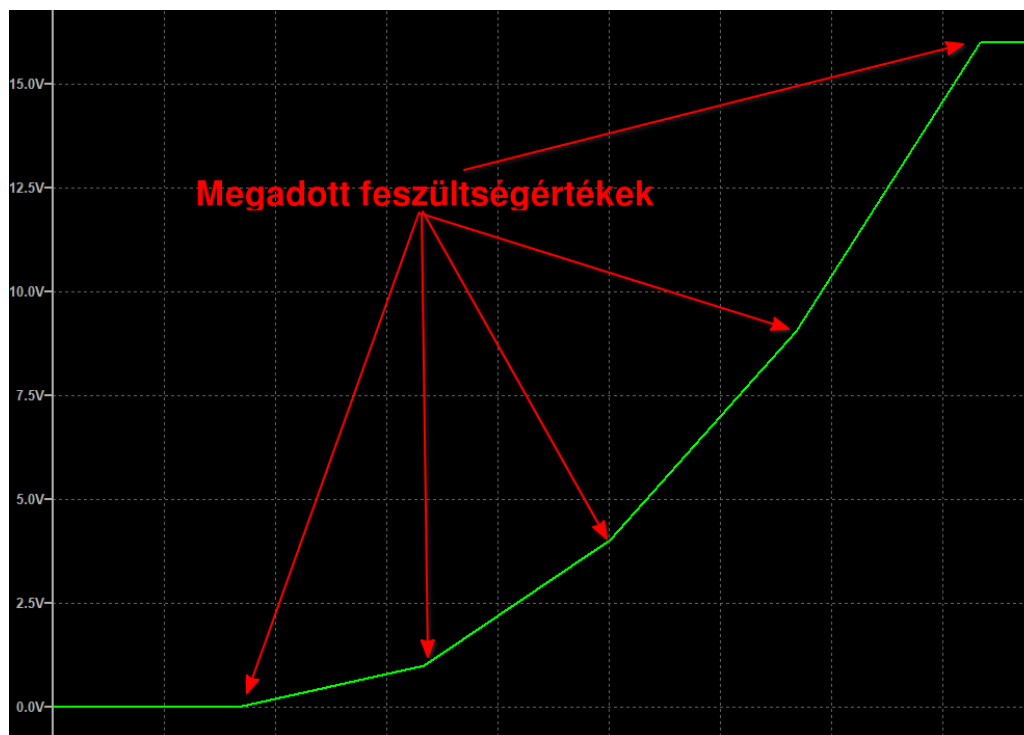


6. ábra. SINE forrás jelalakja

szóösszetétel rövidítéséből származik. Ezzel a módszerrel tetszőleges számú időpillanatban leírható tetszőleges bonyolultságú bemeneti jel, csak meg kell adnunk az egyes időpillanatokban felvett feszültségértékeket. A 7. ábrán egy ilyen jelforrásból kijövő példát láthatunk. Komplex és sok mintából álló jel esetében lehetőségünk van egy **.wav** kiterjesztésű fájlt is megadni, mint jelforrás. Ebben az esetben a PWL FILE opciót kell kiválasztanunk.

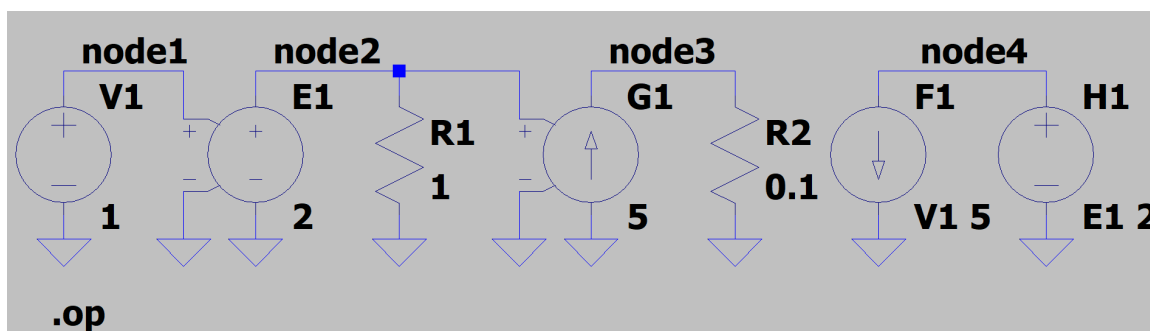
3.3. Vezérelt feszültség- és áramforrások (E,F,G és H források)

Elektronikai modellezés során sokszor előfordul, hogy egy áram- vagy feszültségjel konstansszorosát szeretnénk előállítani vagy feszültségek különbségét szeretnénk képezni. Ezeket a funkciókat ún. vezérelt forrásokkal tudjuk megvalósítani. Az LTspice-on belül 4 db ilyen forrás található meg, plusz 2 db tetszőleges viselkedésű forrás, melyekről a következő szakaszban foglalkozunk bővebben. Ezeket az áramköri komponenseket (ahogy



7. ábra. PWL forrás jelalakja

a többi komponens esetén is) a prefixük különbözteti meg. Ennek megfelelően az LTspice-ban megtalálható komponenseket az angol ÁBC betűivel szokás jelölni. Ez a feszültségforrások esetén V, ellenállás esetén R és vezérelt források esetén a kimaradó betűket felhasználva az E, F, G és H prefix jutott. A források működésének megértése érdekében figyeljük meg a 8. ábra kapcsolását és vegyük végig a különböző források által elvégzett műveleteket.



8. ábra. Vezérelt forrásokat felhasználó kapcsolat

A kapcsolat bemenetén a V1 forrás beállítja a **node1** csomópont potenciálját 1 V-ra, ezt követően az E1 forrás képezi a **node1** és **gnd** csomópontok közötti potenciálkülönbség kétszeresét és ezt a **node2** és **gnd** csomópontok közé kényszeríti. Ezen feszültség hatására az 1 Ω -s R1 ellenálláson 2 A áram kezd el folyni, amit az E1 forrás biztosít. G1 forrás a **node2** és **gnd** közötti feszültség ötszörösét (5 Siemens-szeresét), azaz 10 A-t átkényszerít a 0.1 Ω -s R2 ellenálláson, ezáltal 1 V feszültségesést produkálva rajta.

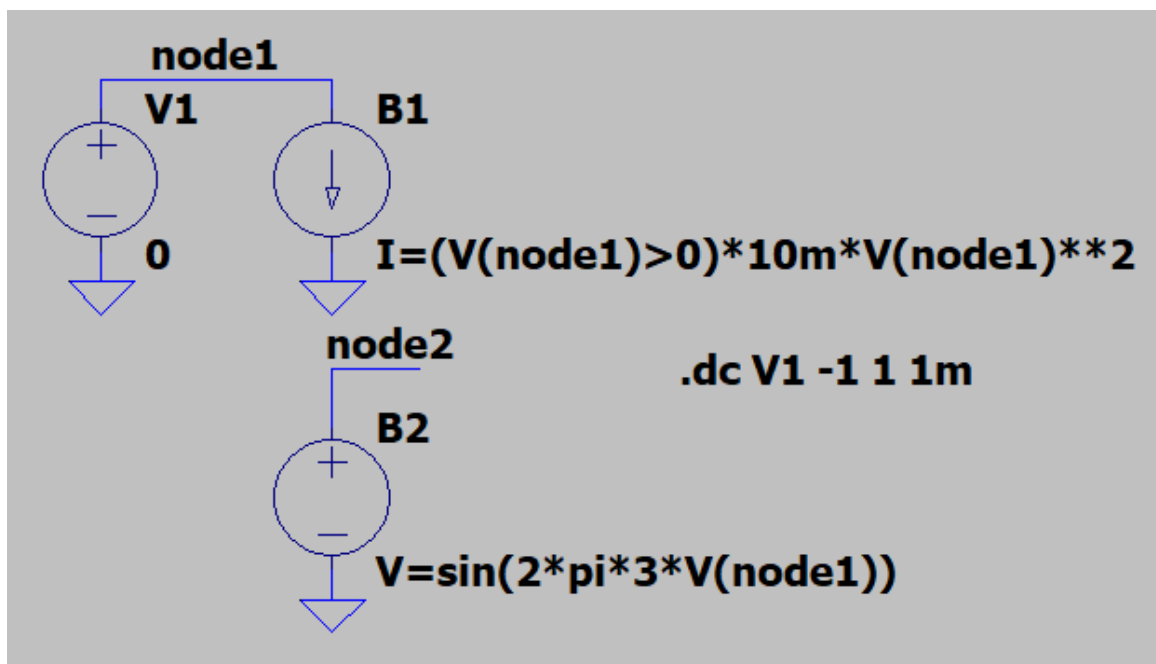
F és H források vezérlőjele egy megadott feszültségforrás árama tud csak lenni, így F1 forrás a V1 forráson átfolyó áram, kapcsolásunkban 0 A, 5 Siemens-szeresét folytatja át H1 forráson. H1 forrás pedig az E1 forrás áramának, -2 A kapcsolásunkban, a 2 Ohm-szorosát állítja elő **node4** csomóponton tehát -4 V-t.

A feszültségforrások áramai a villamosságtani szimmetrikus referenciáirányok szerint vannak meghatározva, tehát a forrás (és minden kétpólus) árama a feszültségének referenciáirányával azonos (+ ből mutat a - ba). Ennek köszönhető, hogy E1 forrás áram -2 A.

A feszültséggel vezérelt források (E és G) esetében E2 és G2 néven rendelkezésre áll a vezérlőjel fordított polaritású szimbóluma is. Egyes esetekben kényelmesebb lehet ennek a használata, így érdemes említést tenni róla. Szimulációs szempontból nincsen különbség a források között, hiszen csak a csomópontok sorrendje lesz felcserélve a netlista fájlban.

3.4. Tetszőleges viselkedésű források (BV és BI források)

Az előző szakaszban szereplő források közös jellemzője, hogy a kimeneti mennyiségük (feszültség- vagy áramjel) csak a vezérlőmennyiség konstansszorosával tud arányos lenni. Egyes esetekben szükségünk lehet összetettebb viselkedésű komponens szimulációjára. Ebben az esetben tudjuk használni az ún. tetszőleges viselkedésű, vagy másnéven B forrásokat (az angol arbitrary Behavioral sources rövidítésből). A tetszőleges viselkedés jelző a források által szolgáltatott viselkedési leírások széles tárházára utal. A források kimeneti jelét különböző matematikai és logikai függvényekkel és azok kompozícióival tudjuk leírni. Ezt a leírási módot a továbbiakban vezérlési egyenletnek fogjuk hívni. A teljesség igénye nélkül⁵ lehetőségünk van a hatvány, trigonometriai, hiperbolikus, idő szerinti integrálás és deriválás függvényeket, lineáris interpoláció és logikai alapműveletek kombinációjával az általunk kívánt viselkedést leírni. Természetesen a vezérlési egyenletben használhatjuk a hálózat villamos jeleit is a V(A,B) és Ix() kifejezésekkel a vezérlési egyenletekben.



9. ábra. Vezérelt források használata

A források működésének illusztrációja végett figyeljük meg a 9. ábrát. Az ábrán a B1 forrás árama és a B2 forrás feszültsége az 1. egyenletek alapján függenek a node1 csomópont feszültségétől.

⁵A források által biztosított matematikai formulák listájához látogassunk el a sűgő `F1\LTspice\Circuit elements\B. Arbitrary Behavioral Voltage or Current Sources` fűlre.

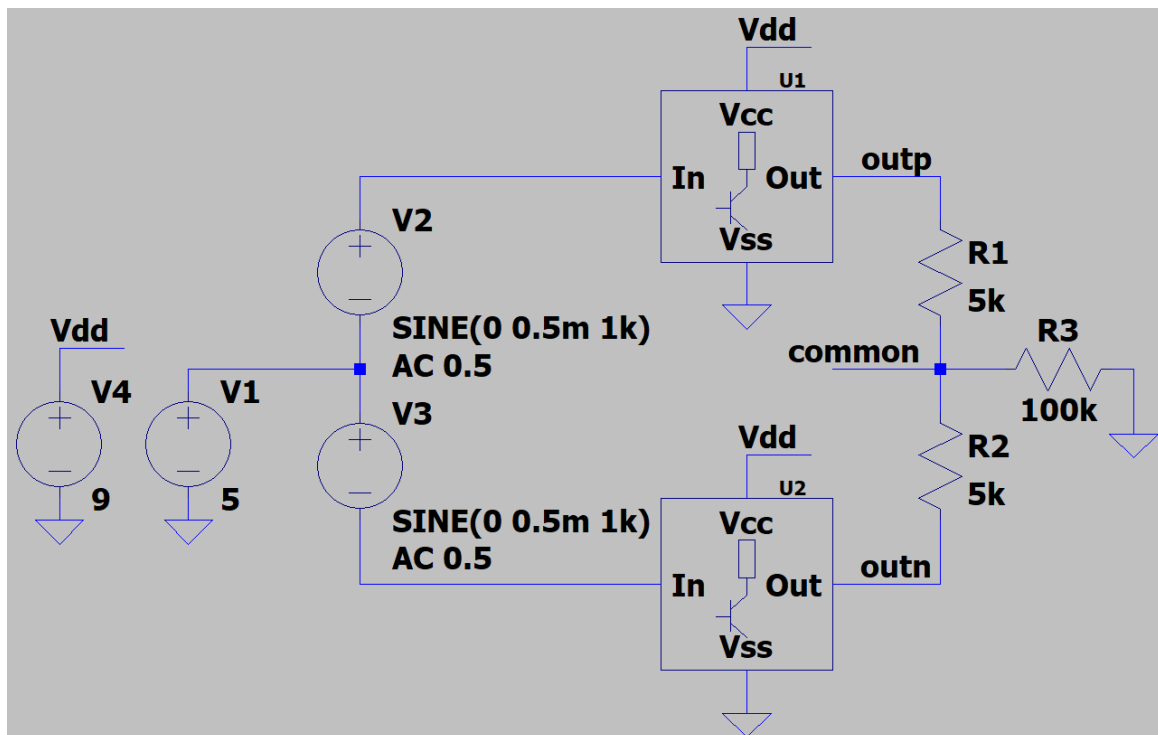
$$I_{B1}(U) = \begin{cases} 10 \left[\frac{mA}{V^2} \right] U^2, & \text{ha } U \geq 0 \\ 0, & \text{máskülönben} \end{cases} \quad (1)$$

$$V_{B2}(U) = \sin \left(6\pi \left[\frac{1}{V} \right] U \right)$$

A forrásokat felhasználva megvalósíthatunk aktív és passzív kétpólusokat is, valamint tetszőlegesen bonyolult vezérlőjelek is előállíthatók, azonban ügyelni kell a források jeleinek folytonosságára is. Nem folytonos és nem folytonos deriváltakkal rendelkező jelek esetén nehézkes lehet a megoldó motornak konvergencia megoldást találnia, így tranziens szimuláció (lásd 4. fejezetet.) esetén pl. nagyon kicsi időlépést kényszerül használni, ezzel megnövelve a szimulációhoz szükséges számítási időt!

3.5. Hierarchikus áramköri blokkok

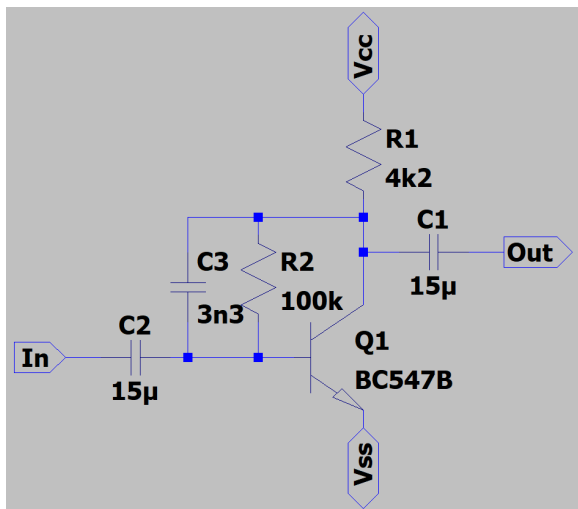
Az eddig említett áramköri elemek rengeteg lehetőséget biztosítanak számunkra áramköri komponensek és kapcsolások modellezésére, azonban előfordul, hogy a kapcsolat mérete vagy összetettsége miatt kedvezőbb lenne az áramköri alkomponenseket hierarchikusan rendszerezni. Hierarchikus rendezés alatt az áramköri komponensek funkció szerinti csoportosítását értjük, pl. egy audió erősítőben megtalálható egy előerősítő kapcsolás, egy teljesítmény erősítő és egy tápegység, csak hogy néhányat említsünk az áramköri blokkokból. Ezeket a komponenseket célszerű külön sémákon megrajzolni és az elsődleges sémán (**top level schematic**) csak egy szimbólummal hivatkozni rájuk. Ez a metodika elősegíti az áramköri komponensek rendszerezését, funkcionális dekompozícióját, áramköri komponensek újrafelhasználhatóságát és a séma könnyebb átláthatóságát. Egy ilyen hierarchikus sémára mutatunk példát a 10. ábrán.



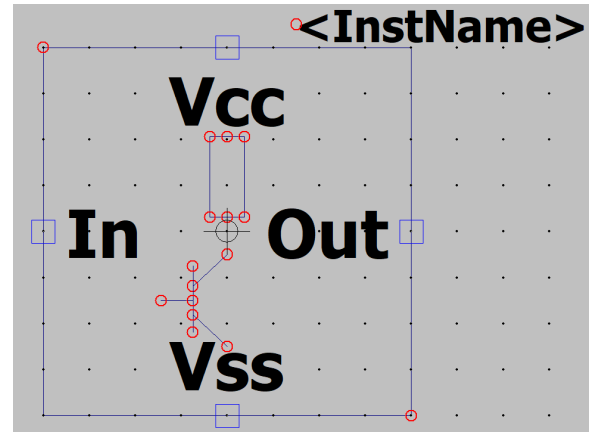
10. ábra. Előerősítő hierarchikus tesztkapcsolása

Hierarchikus blokkok készítése során meg kell rajzolnunk a blokk sémáját és ki- és bemeneti kapcsolódási pontokat (ún. **pineket**) kell elhelyeznünk rajta. Ezek a kapcsolódási pontok gyakorlatilag a csomópontok és vezetékek címkéit jelentik, amelyek a blokk elektromos összeköttetéséért felelősek. A címkék esetében ha

kiválasztjuk a pólusok (portok) típusát (input, output vagy bidirectional), akkor az egyszerű szöveges megjelenítés helyett egy keretezett szövegdobozt fogunk látni, a pólus típusának megfelelően (bemeneti, kimeneti és kétirányú). Erre láthatunk példát a 11. ábrán.



11. ábra. Előerősítő kapcsolási rajza



12. ábra. Előerősítő szimbóluma

A kapcsolási rajz megrajzolása után egy jelképes szimbólum ábrát kell készíteni, amely majd a hierarchikus blokkot fogja reprezentálni a magasabb rendű kapcsolási rajzokon. Célszerű a blokk szimbólumának megrajzolásánál a blokk funkcióját is feltüntetni, legalább jelképesen és egyszerűsítve, ezzel megkönnyítve a magasabb rendű séma olvashatóságát, valamint elkerülve a semmitmondó téglalapokkal telirakott sémákat. Új szimbólum készítéséhez (**File > New symbol**) menüponot kell válasszuk, ekkor megnyílik az szimbólumszerkesztő, mely nagyon hasonlít a sémakeresztőhöz, azonban itt leginkább grafikai primitíveket (egyenes, téglalap, ellipszis és körív) tudunk felhasználni a szimbólum elkészítéséhez. Ezeket a grafikai primitíveket a (**Jobb egérgomb > Draw**) menüből választhatjuk ki. A grafikai primitívek ún. horgonypontokkal vannak definiálva (anchor points), melyek a primitívek utólagos szerkesztését teszik lehetővé. Ezeket a horgonypontokat piros körök jelölik és a megjelenítésüket a (**View > Anchor points**) vagy az **O** gyorsgomb megnyomásával tudjuk ki- és bekapcsolni. A primitívek szerkesztéséhez a **Move (M)** vagy **Stretch (S)** eszközöket tudjuk használni melyekkel a horgonypontokat megfogva vagy kijelölve tudjuk módosítani az adott primitívet. A 12. ábrán az illusztratív előerősítő szimbólumát láthatjuk.

A szimbólum megrajzolása után el kell helyeznünk rajta a séma pólusait, amit a (**Jobb egérgomb > Add pin**) vagy **P** gyorsgombbal tudunk megtenni. Fontos megjegyezni, hogy a szimbólum megrajzolása során egy sűrűbb ráccsal dolgozunk, mely lehetővé teszi a finomabb ábrák megrajzolását, azonban a pólusok csatlakozásait csak egy ritkább rács pontjaira tudjuk elhelyezni (fekete pontok a szimbólumszerkesztő felületén). Célszerű a szimbólum megrajzolása során figyelni ezekre, hogy a portok a szimbólum ábrájának szerves részét képezzék! A portok neveinek megválasztása során, ha a szimbólum egy LTspice primitívet (pl. dióda) vagy beépített áramköri könyvtári komponenst tartalmaz (pl. netlista fájlal megadva), úgy a port nevének nincs elektromos szempontból jelentősége. Ha a szimbólum egy alacsonyabb szintű sémát reprezentál (ahogy ebben a példában szerepel), úgy a portok neveinek egyezniük kell a reprezentált séma vezetékcímkeivel, hiszen ezen a név egyezésen keresztül valósulnak meg az elektromos összeköttetések. Mind a két esetben célszerű a portokat abban a sorrendben hozzáadni a szimbólumhoz ahogy azt a netlistában szeretnénk, vagy elvárt a netlistával megadott áramköri komponensek esetén, lásd később a 7. fejezetben.

A szimbólum elkészítése után hozzá kell adnunk az esetleges attribútumokat az elkészített szimbólumunkhoz, ehhez a (**Edit > Attributes > Edit Attributes**) menüponot kell megnyissuk. Az egyetlen attribútum amit szerkesztenünk kell az a szimbólum típusa (symbol type). Ezt a beállítást át kell kapcsoljuk **Block** állapotra, ezzel jelezve az LTspice-nak, hogy egy alacsonyabb rendű és sémával definiált áramköri blokk van

reprezentálva ezzel a szimbólummal. Fontos megjegyezni, hogy az LTspice a szimbólum és a mögöttes séma között az azonos fájlneven keresztül teremt kapcsolatot (leszámítva a kiterjesztést).

A szimbólum elkészítéséhez egy alternatív mód az automatikus szimbólumgenerátor használata. Ehhez az áramkör sémáján a **(Jobb egérgomb > Hierarchy > Open this sheet's symbol)** menüt kell kiválasztanunk, ekkor az LTspice ha nem találja a sémához tartozó szimbólumot, úgy felajánlja az automata generálást. Ebben az esetben egy generikus téglalapot kapunk szimbólumként, ahol a pólusok a hozzájuk tartozó címkék típusa alapján kerülnek elhelyezésre. Egyszerűbb blokkok esetében célszerű lehet ezt a módszert használni a szimbólum generálására és ezen a szimbólumon csak kisebb mértékű módosításokat végezni!

4. SPICE szimulációs direktívák

Az áramköri kapcsolási rajzok összeállítása, valamint a hozzájuk rendelt szimbólumok segítségével összetett szimulációs modelleket tudunk létrehozni, azonban ezekben a szimulációs modellekhez még nem adtuk hozzá az elvégzendő szimulációkat és azok paramétereit. Ebben a fejezetben az LTspice szimulációs képességeit ismertetjük röviden egyszerű szimulációk esetére. Összetett, több iterációt és egyéb utófeldolgozást kívánó szimulációk futtatását az 5. fejezetben ismertetjük majd bővebben.

A futtatandó szimulációs analízis parancsát ún. SPICE direktívával tudjuk megadni, ezeket a direktívákat a kapcsolási rajzra elhelyezett . utasításokkal tehetjük meg. Fontos megjegyezni, hogy szimulációs analízisből egyidejűleg csak egyetlen példány lehet a szimulálandó sémán! Több különböző szimuláció futtatásához létrehozhatunk több szimulációs direktívát, melyek közül egy kivételével mindet kommentként vagy szöveggként hozzuk létre. Ezt követően a különböző szimulációk futtatásához manuálisan kell kiválasztanunk a futtatandó szimulációs direktívát és a többit kommenté vagy szöveggé alakítani.

Szimulációs direktívák hozzáadásához és szerkesztéséhez a **(Simulate > Configure Analysis)** , (A gyorsgomb) menüt kell kiválasztanunk, ekkor egy grafikus felületen beállíthatjuk a szimulációhoz szükséges paramétereiket. Gyakorlottabb felhasználók számára a szimulációs direktívák szintaktikája ismert lehet, emiatt kényelmesebb lehet a grafikus felületet kihagyva a direktívákat egyenesen a sémára elhelyezni. Ehhez a **(Jobb egérgomb > Draft > SPICE directive)** parancsot, vagy a . gyorsgombot kell használnunk. Ekkor egy szöveges mezőbe begépelve a szimulációs direktívát az lehelyezhetővé válik a séma felületén.

4.1. Munkapont meghatározása és a .op direktíva

Az egyik alapvető funkcionális az áramköri szimulációs programoknak a nemlineáris munkapont meghatározása. Ezen szimuláció során a sémán lévő összes komponens nemlineáris kisfrekvenciás karakterisztikáját és a komponensek közötti összekapcsolási kényszereket is figyelembe véve egy - általában nemlineáris - egyenletrendszert állít össze és old meg a SPICE szimulációs motor. A nemlineáris algebrai egyenletrendszer megoldására a megoldó motor a **Newton-Raphson** módszert használja. Az egyenletrendszer sikeres megoldásához szükséges a szimulálandó áramköri komponensek karakterisztikájának és azok deriváltjainak folytonossága (C^1 folytonossági függvényosztály). Ez a folytonosság az LTspice összes beépített komponense esetén fennáll, azonban az általunk készített kapcsolások esetében ez nem mindig garantált! Konvergenciaproblémák esetében célszerű a problémásnak vélt áramköri komponens karakterisztikáit módosítani, hogy simábbak legyenek (C^1 folytonosság!).

Munkaponti szimuláció futtatásához a sémán a következő direktívát kell elhelyeznünk, majd elindítani egy szimulációt a **(Simulate > Run)** parancssal, vagy a **SHIFT+R** gyorsgombbal.

.op

A szimuláció lefutása után a csomóponti potenciálok és az ágáramok egy felugró ablakban lesznek listázva. Célszerű a szimuláció futtatása előtt a számunkra releváns csomópontokat címkékkel ellátni (**N**), hogy a szimulációs eredmények közül könnyebben beazonosíthatóak legyenek az eredmények. Alapértelmezetten csak az elsődleges sémán lévő csomóponti potenciálok, valamint a sémán lévő eszközök áramai lesznek megjelenítve. Ha

szeretnék a hierarchikus blokkok potenciáljait is meghatározni, úgy a **(Tools > Settings > SaveDefaults)** menüpont alatt be kell kapcsoljuk a **Save subcircuits nodal voltages** opciót. Az áramkörökben található komponensek portáramainak megjelenítéséhez pedig a **Save subcircuits device currents** opciót kell engedélyeznünk. Munkaponti szimulációra és annak eredményére egy példát láthatunk az 1. és 2. ábrákon.

4.2. Forrásparaméterek változtatása és a .dc direktíva

A gyakorlatban többször előfordul, hogy az áramkörünk munkapontját több forrásfeszültség esetén is szeretnénk meghatározni és optimalizálni valamilyen metrika alapján. Ebben az esetben lesz hasznos számunkra a DC sweep szimuláció. A szimuláció során akár három független forrás állandósult (DC value) értékét változtathatjuk többféle módon. Alapértelmezés szerint a források értékeit **lineárisan változtatjuk**, vagyis az egyes szimulációs értékek egy számtani sort képeznek. Lineáris lépésközon kívül választhatunk oktáv, dekád vagy lista alapú lépésközöket.

Oktáv és dekád választása esetén a forrás értékei logaritmikus skálán lesznek egyenletesen elosztva, vagyis az értékek egy mértani sort fognak alkotni. Mind a két esetben meg kell adni a szimulációk kezdeti és végső értékét, valamint hogy oktávonként vagy dekádonként hány szimulációt szeretnénk elvégezni. Két érték távolsága 1 oktáv, ha az értékek hányadosa kettő⁶, míg 1 dekád távolság esetében a hányados tíz lesz. Matematikailag a 2. formulák adják meg az egyes szimulációs pontokat. A formulákban v_0 a mértani sor kezdő értéke és n az egy oktávban vagy dekádban lévő osztópontok száma.

$$\begin{aligned} o_i &= v_0 \sqrt[n]{2}^i \\ d_i &= v_0 \sqrt[n]{10}^i \end{aligned} \quad (2)$$

A szimulációs értékek megadásának utolsó lehetősége a lista szerinti felsorolás. Ekkor egymás után kell felsorolnunk a szimulálandó értékeket tetszés szerinti sorrendben. A szimuláció során az értékek növekvő sorrendben lesznek szimulálva és kiértékelve.

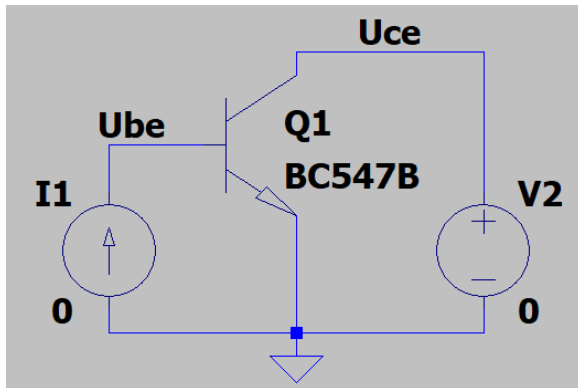
A DC parancs használatára egy jelképes példa egy bipoláris tranzisztor kimeneti karakterisztikáit meghatározó szimulációs összeállítás. A kapcsolást a 13. ábrán láthatjuk, míg a szimulált kollektorköri áramokat a 14. ábrán figyelhetjük meg. A kapcsolás egy bázisra kötött független áramgenerátorból és a kollektorra kötött független feszültségforrásból áll. A karakterisztikák meghatározásához a következő SPICE direktívát kell elhelyezni a sémán (. gyorsgomb):

```
.dc V2 0 1 1m I1 list 0 25u 75u 150u 200u
```

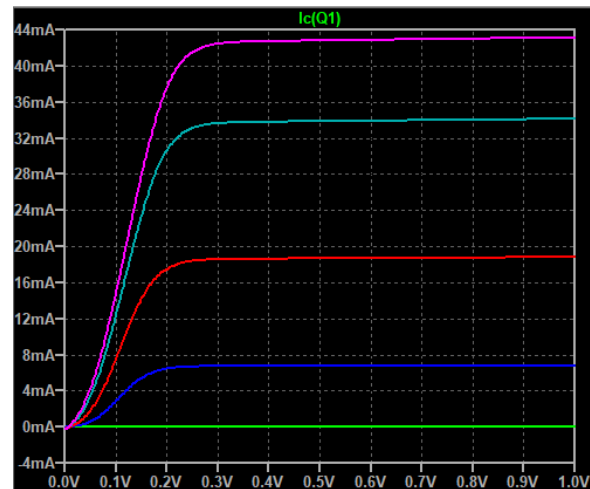
Kezdő felhasználók számára ebben az esetben is ajánlott a grafikus szerkesztőfelület használata, mely a **(Simulate > Configure Analysis > DC sweep)** menüből érhető el. A fenti direktíva a V2-s forrás feszültségét 0 V és 2 V között 1 mV-os lineáris lépésekkel változtatja, valamint az I1-es forrás értékét 0 μA , 25 μA , 75 μA , 150 μA és 200 μA értékekre állítja be a lista opciót használva. A direktíva használata során az elsődlegesen változtatott forrás értékei lesznek felhasználva a jelek ábrázolása során az X tengelyként, míg a másodlagos és harmadlagosan változtatott forrásértékek független görbéket fognak alkotni, melyeket különböző színekkel annotál majd a program.

A DC sweep szimuláció alapértelmezetten csak 3 független forrás együttes változtatását teszi lehetővé, azonban ahogy később az 5. fejezetben látni fogjuk lehetőségünk van tetszőleges számú forrás értékeit változtatni a **.step** direktíva felhasználásával. Meg kell azonban jegyezni, hogy sok forrás változtatása esetén a szükséges szimulációs idő igen gyorsan fog növekedni, hiszen minden újonnan felvett forrás esetében az összes korábbi szimulációs lépést meg kell ismételni!

⁶Az oktáv alapján gondolhatnánk, hogy a nyolcas számhoz köthető az elnevezés, azonban a nyolcas szám az oktáv eredeti zenei gyökereihez nyúlik vissza. Egy adott zenei hang és az 1 oktávval távolabbi szomszédja eléréséhez a 8. zenei hangra kell ugorjunk, azonban az egy oktávnyira lévő hangok frekvenciában kifejezve egymás dupláit. Az elnevezés gyökere az emberi hallásra vezethető vissza. Két hangot harmonikusnak érzékelünk, ha a frekvenciáik hányadosa kicsi értékű számlálással és nevezővel rendelkezik pl. $\frac{f_1}{f_2} = \frac{3}{2}$.



13. ábra. Karakterisztika meghatározó kapcsolás



14. ábra. BJT kimeneti karakterisztikái

4.3. Kisjelű analízis és a .ac direktíva

Kisjelű analízis során az alapfeltevésünk, hogy a kapcsolás áramaiban és feszültségeiben történő változás eleendően kis mértékű ahhoz, hogy az áramköri nemlinearitások hatása elhanyagolható legyen. Lényegében a kapcsolást a munkapontban közelítjük annak linearizált változatával, analóg módon ahogy egy függvényt az elsőrendű Taylor-polinomjával, azaz véges tagszámig számolt **Taylor-sorával** közelítjük.

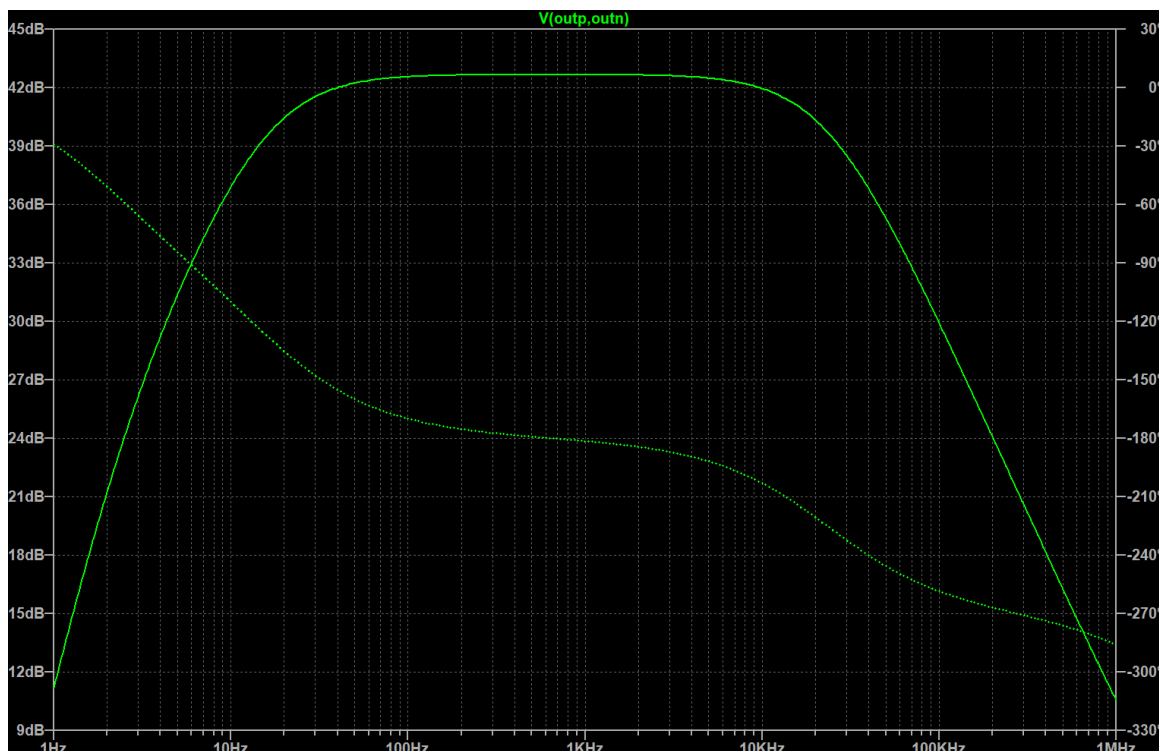
A linearizált hálózati modell megengedi nekünk, hogy a kapcsolásunkat az elméleti villamosságtan lineáris hálózatszámítási eszközeivel kezdjük el vizsgálni. Ennek megfelelően értelmezhetünk egy átviteli karakterisztikáját (vagy átviteli függvényét, megfelelő általánosításokkal élve), mely a frekvencia függvényében határozza meg a kapcsolás bemenetére érkező szinuszos jel komponenseinek amplitúdóerősítését (amplitúdó karakterisztika) és fázistolását (fáziskarakterisztika).

Kisjelű szimuláció futtatásakor a hálózat gerjesztését a független források fogják biztosítani. Ennek megfelelően a független források paraméterezésénél be kell állítani az **AC amplitude** értéket a szimulációnak megfelelő értékűre. Fontos megjegyezni, hogy a kisjelű szimuláció futtatásakor csakis a linearizált hálózatot veszi figyelembe a szimulátor, így nincs szükség a gerjesztő mennyiségeket kicsiny értékűre választani. Ennek megfelelően a gyakorlatban az AC amplitúdót rendszerint 1 V-ra választjuk, hiszen így a logaritmikus skálán ábrázolva az egységnyi erősítéshez a 0 dB-es érték fog tartozni! Kisjelű szimulációra láthatunk egy példát a 10. ábra kapcsolását felhasználva, hozzáadva a következő AC direktívát:

```
.ac dec 100 1 1meg
```

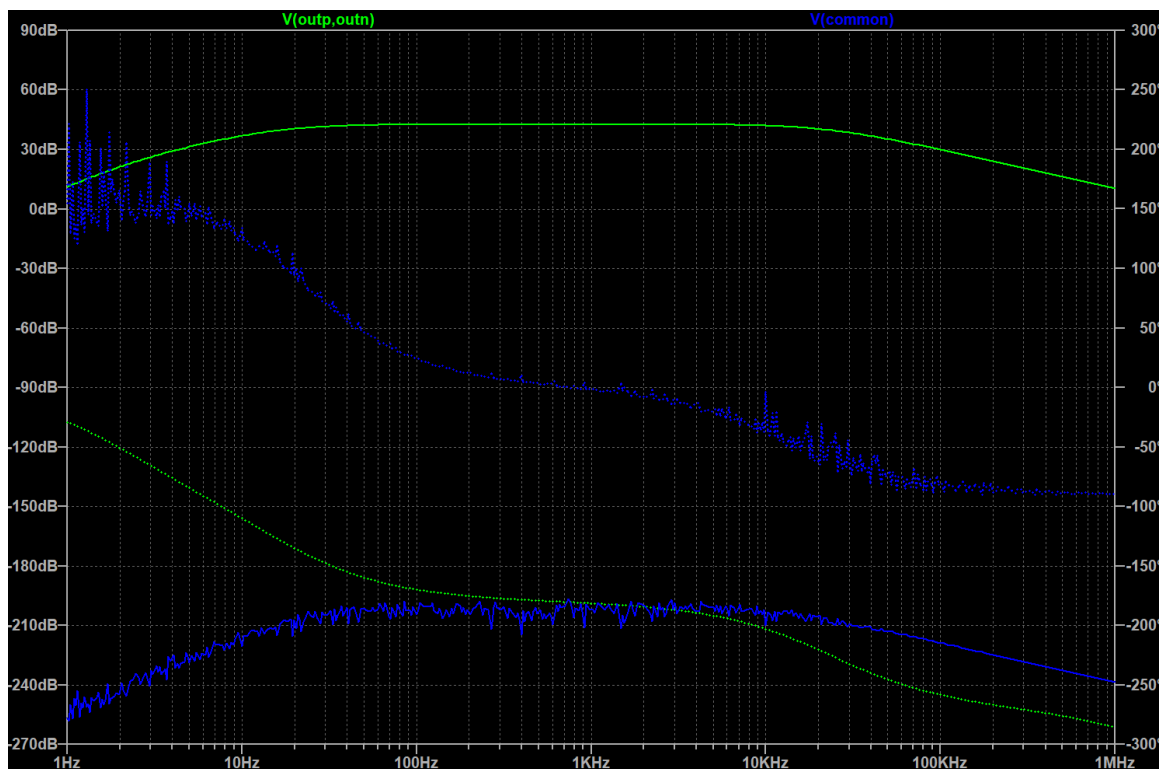
A direktíva dekádonként 100 pontban 1 Hz-től 1 MHz-ig meghatározza a kapcsolást átvitelét. Természetesen itt is több lehetőségünk van a szimulálandó frekvenciák megadására. A DC sweep szimulációhoz hasonlóan itt is tudjuk használni a **lineáris**, az **oktáv** és a **lista** opciókat a **dekádok** helyett. A direktívának köszönhetően a szimulátor **meghatározza a kapcsolás munkapontját**, elvégzi a linearizálást és a lineáris hálózatnak numerikusan meghatározza az átviteli karakterisztikáját. Az eredmények ábrázolásánál a vizsgált jeleket alapértelmezetten egy Bode-diagrammon ábrázolva láthatjuk, ahogy az a 15. ábrán láthatjuk is. Bode diagrammok ábrázolásánál az alapértelmezett beállítás, hogy az amplitúdó- és fáziskarakterisztikákat egy ábrán ábrázolja az LTspice, folytonos vonallal ábrázolva az amplitúdóátmenetet és szaggatottal a fázist. Az amplitúdóátmenetet a bal oldali, míg a fázisátmenetet a jobb oldali skála alapján van értelmezve és alapértelmezetten az amplitúdóátmenetet dB-ben van kifejezve. Természetesen lehetőségünk van ezeket a beállításokat módosítani, ahogy azt a 6. fejezetben be is mutatjuk.

Az ábráról leolvasható, hogy a 10. ábrán lévő differenciális jelekkel meghajtott előerősítő kimenetén kb. 42 dB-es erősítés figyelhető meg egy 15 Hz és 23 kHz közötti áteresztősávban, ezen a sávon kívül az erősítés monoton módon csökken! Fontos megjegyezni, hogy a 15. ábra nem az erősítés értékét adja meg, hanem a differenciális



15. ábra. Előerősítő átviteli karakterisztikája

jelek $V(\text{outp}, \text{outn})$ dB-ben kifejezett értékét. Ezeket a jeleket azonban tudjuk erősítésként értelmezni, hiszen az eredeti kapcsolásban a bemeneti differenciális jelünk 1 V amplitúdójú volt, így a kimeneti jeleink egy az egyben az erősítésnek felelnek meg!



16. ábra. Átvitel a "Common" csomópontra

Fontos megjegyezni, hogy a munkaponti linearizálás teljes mértékben figyelmen kívül hagyja a kapcsolás nemlinearitásait! Ennek köszönhetően, ha ábrázoljuk az átvitel mellett a 10. ábra *Common* csomópontjának feszültségét úgy az áteresztősávban -210 dB-es értéket láthatunk. A következő szakaszban a nagyjelű szimulációk során látni fogjuk, hogy ennél az értéknél nagyságrendekkel nagyobb jeleket fogunk látni a *Common* csomóponton.

4.4. Nagyjelű analízis és a .tran direktíva

A kisjelű szimuláció előnye az alacsony erőforrásigény, így relatíve könnyű a nagyszámú szimuláció futtatása. Kisjelű szimuláció során a megoldó motornak a hálózat linearizációját csak egyszer kell végig számolnia, majd az így adódó lineáris hálózatra minden szimulációs pontban egy algebrai egyenletrendszert kell megoldani, mely egyszerű feladat a számítógépek számára. A módszer hátránya, hogy az így számolt megoldás csak közelítőleg pontos és fontos nemlineáris jelenségeket elfedhet!

A pontosabb szimulációk érdekében a számítások során figyelembe kell vegyünk a hálózat nemlinearitásait is! Ennek következtében a megoldandó probléma egy nemlineáris differenciál-egyenletrendszerre vezet. Általános esetben az ismeretlenekből álló vektorra érvényes egyenletrendszer a 3. egyenlettel írható le, ahol f tartalmazza a hálózatra érvényes nemlineáris egyenleteket.

$$\frac{d}{dt}\vec{y}(t) = \vec{f}(t, \vec{y}) \quad (3)$$

Ennek az egyenletnek az általános megoldásához integráljuk az egyenletrendszer mind a két oldalát és használjuk fel a hálózatra érvényes kezdeti feltételt: $\vec{y}(t_0) = \vec{y}_0$. Ennek megfelelően az ismeretlen változók időfüggvénye a 4. egyenlet szerint alakul.

$$\vec{y}(t) = \vec{y}_0 + \int_{t_0}^t \vec{f}(\tau, \vec{y}(\tau))d\tau \quad (4)$$

A 4. egyenletből látható, hogy az ismeretlen időfüggvények meghatározásához numerikus integrálást kell alkalmaznunk. A pontos numerikus integrálás érdekében kis időlépéseket kell alkalmaznunk, így az elvégzendő számítások száma gyorsan nő.

Nagyjelű szimuláció futtatása során a hálózat gerjesztését a független források időfüggő viselkedése fogja meghatározni, így a szimuláció futtatása előtt be kell állítani a szükséges források viselkedését a szimulációnak megfelelően.

Nagyjelű szimuláció futtatásához a **.tran**⁷ direktívát kell hozzáadjuk a sémához. A direktíva paramétereinek megadásához ismét használhatjuk a grafikus szerkesztőfelületet (**Simulate > Configure Analysis > Transient**). A direktíva minimalista paraméterezéséhez elegendő a végső szimulációs időt megadnunk, ekkor a szimulátor az áramköri komponensek értékéből számít egy becslést a szükséges időlépés meghatározásához.

.tran 100m

Abban az esetben, ha mi szeretnénk meghatározni a maximális időlépés nagyságát úgy 4 paraméter megadása szükséges. Ezen paraméterek sorrendje a következő:

.tran Tprint Tstop Tstart Tmaxstep

A paraméterek magyarázata a következő: **Tstop** a szimulálandó időtartam hossza, **Tstart** a szimulált változók mentésének kezdeti pillanata, **Tprint** a szimulált jelek ábrázolásakor használt időskála kezdeti értéke és végül **Tmaxstep** a szimulátor által használt maximális időlépés nagysága.

A kis- és nagyjelű szimulációk összehasonlításához végezzük el az előző szakaszban ismertetett audió előerősítő nagyjelű analízisét. Ehhez a 10. ábra kapcsolását egészítsük ki a következő direktívával:

⁷A direktíva neve a *Transient* angol szó rövidítéséből származik, mely az időben változó jelek leírását takarja.

```
.tran 0 40m 30m 1u
```

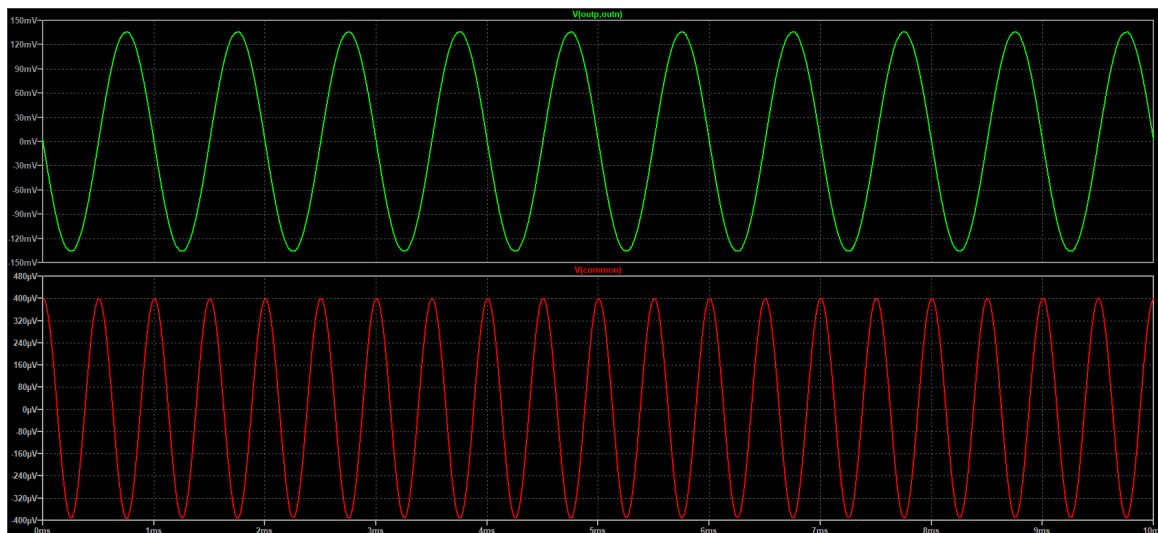
Vagyis a szimulátor egy 40 ms-ig tartó tranziens szimulációt fog végezni, maximum $1\mu\text{s}$ -es időlépést használva, a szimulált jeleket csak 30 ms után kezdi el elmenteni és a megjelenítés során az idő tengelyt 0-tól kezdve rajzolja ki. Mivel csak 30 ms-től mentjük az adatokat, így az ábrázolt hullámformák 10 ms hosszú időtartamot ölelnek fel.

Ha ábrázoljuk a kimeneti differenciális feszültséget és a *common* csomópont feszültségét akkor szembevetően láthatóvá válik a hálózat nemlinearitásának hatása! Az ábráról leolvashatjuk a kimeneti jel amplitúdóját, mely **136 mV**-ra adódott, ennek megfelelően az erősítés dB-ben kifejezett értékére **42.6 dB**-t kapunk, mely igen közel van a korábbi AC szimuláció során kapott maximális értékhez! Azonban a *common* csomópont feszültsége esetében nem ez a helyzet. A tranziens szimuláció eredményeként **396 μV** -ot kapunk, ez **-8 dB**-es erősítésnek felel meg, mely nagyságrendekkel több, mint az AC szimulációnál számolt **-210 dB**-es érték!

További érdekességként figyeljük meg, hogy a *common* csomóponton előálló szinuszos jel frekvenciája **kétszerese** a fölötte lévő kimeneti jel frekvenciájának! Ez a viselkedés alapvetően idegen a lineáris hálózatoktól, hiszen azok csak a bemeneti jel amplitúdóját és fázisát változtathatják meg! A jelenség az előerősítőben lévő tranzisztor nemlineáris karakterisztikája miatt lép fel! A tranzisztor bemeneti karakterisztikáját egy exponenciális függvény írja le, és mint exponenciális függvény a Taylor-sora tartalmazni fog egy négyzetes tagot. Ez a négyzetes tag felelős a frekvencia kétszeresedéséért, a szinuszos függvények hatványazonossága következtében.

$$\sin^2(x) = \frac{1 - \cos(2x)}{2} \quad (5)$$

Tehát, ha egy szinuszos jelet négyzetre emelünk, úgy az eredmény tartalmazni fog egy 0 Hz-es, azaz időben állandó, és egy kétszeres frekvenciájú komponenset. A nemlinearitásból származó 0 Hz-es tag nem látható az ábrán, mivel a kapcsolásban a kimenet egy leválasztó kondenzátoron keresztül van kivezetve, így az egyenáramú részek tökéletesen le vannak választva! Ez a frekvencia többszöröződés általánosan igaz a nemlineáris hálózatokra!



17. ábra. Nagyjelű szimulációs eredmények

Audiótechnikában és jelátvitel esetében a jelfeldolgozó hálózatok nemlineáris torzítását a Teljes Harmonikus Torzítással (THD-vel röviden⁸) jellemzik. Ez a mérőszám megmutatja, hogy a jel effektív értékének RMS⁹

⁸Az angol Total Harmonic Distortion szakkifejezésből.

⁹Az angol Root Mean Square rövidítéséből. A szakkifejezés az effektív érték számítmódjára utal, ahol a jel négyzetéből számolt integrálközepérték gyökét képezzük.

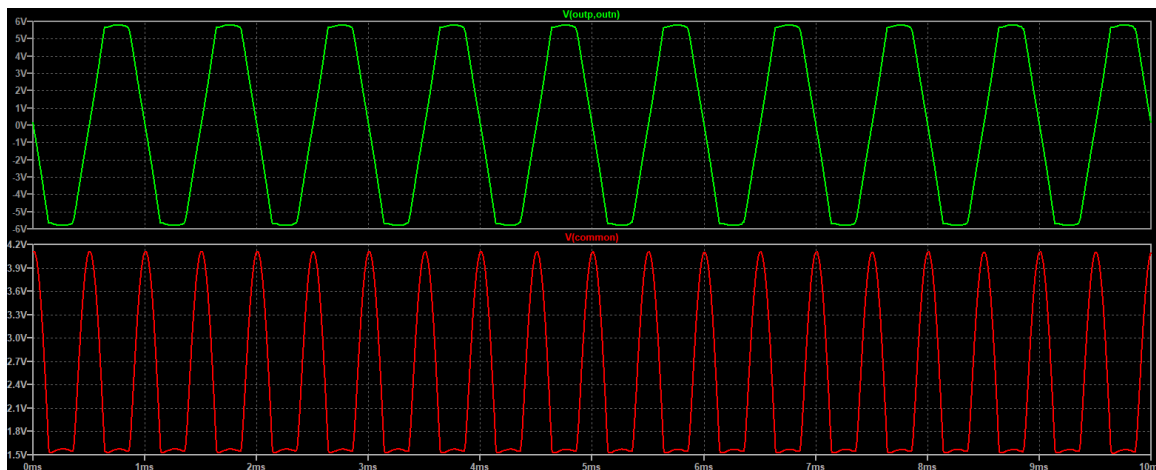
amplitúdó hány %-a adódik a felharmonikus tartalomból. Mivel szinuszos jelek esetében az RMS amplitúdó a különböző frekvenciájú komponensek négyzetes közepével arányos így a THD a 6. formula alapján számolható. A formulában A_i az i -edik harmonikus amplitúdóját jelöli, ahol az 1-es index az alapharmonikus.

$$THD = \frac{\sqrt{\sum_{i=2}^{\infty} A_i^2}}{\sqrt{\sum_{i=1}^{\infty} A_i^2}} \quad (6)$$

A harmonikus torzítás erősen jelszintfüggő, hiszen a torzítások a munkaponttól távoli nemlinearitásokból erednek. Ennek megfelelően a gerjesztés amplitúdóját változtatva megfigyelhetjük, hogyan változik a torzítás mértéke. A szimulációs eredményeket az 1. táblázatban foglaltuk össze. Mint látható 20 mV-os jelamplitúdó felett a torzítás jelentősen növekszik! Ez a torzítás a jelalak torzulásával jár együtt, ahogy azt meg is figyelhetjük a 18. ábrán a szinuszos jel trapézodásából! A THD számításához szükséges szimulációs beállításokat az 5. fejezetben ismertetjük bővebben a **.four** direktíva tárgyalásakor!

A [mV]	THD [%]
1	0
2.154 43	0.001 416
4.641 59	0.009 175
10	0.043 222
21.5443	2.749 631
46.4159	6.929 302
100	7.685 063

1. táblázat. Teljes harmonikus torzítás változása

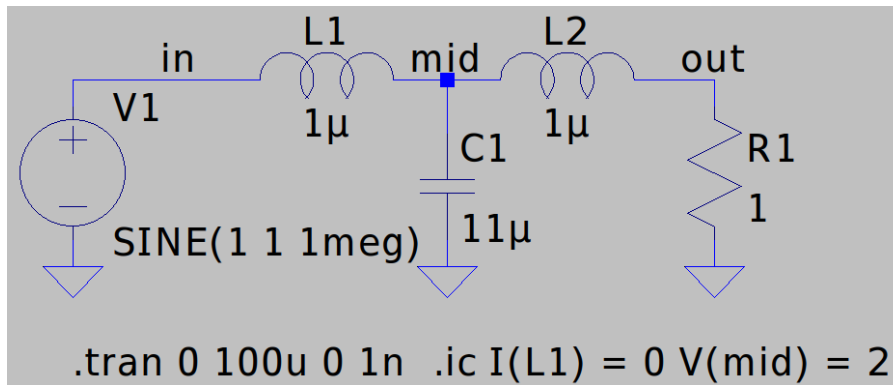


18. ábra. Nagyjelű torzítás megjelenése 100 mV-os bemeneti jel esetén

Bonyolult hálózatok és szimulációk esetén szükséges lehet, hogy a hálózatot egy adott kezdeti állapotból kiindulva szimuláljuk. Ekkor a triviális megoldás, hogy olyan gerjesztő jeleket állítunk be, mellyel a hálózat véges időn belül beáll a szükséges kiindulási állapotba, ezután pedig elkezdhetjük menteni a szimulációs adatokat. Ennek a módszernek a használatakor igen gyorsan abban a helyzetben találhatjuk magunkat, hogy a szimulációs idő nagyon nagy részét csupán a kezdeti állapot beállítására használjuk fel, így a processzorunk nagyrészt feleslegesen számol. Ennek a problémának egy jobb megoldási módja a beépített **.ic** direktíva használata!

A direktíva használatával beállíthatjuk csomópontok kezdeti potenciálját, ezzel beállítva a csomóponton lévő kapacitások kezdeti töltésmennyiségét is, valamint beállíthatjuk a kapcsolásunk tekercseinek kezdeti áramait. A direktívának köszönhetően a numerikus megoldó a hálózat kezdeti állapotának számolásánál figyelembe fogja venni az általunk megadott paramétereket, és azokkal konzisztens kezdeti megoldásból kezdi el numerikus

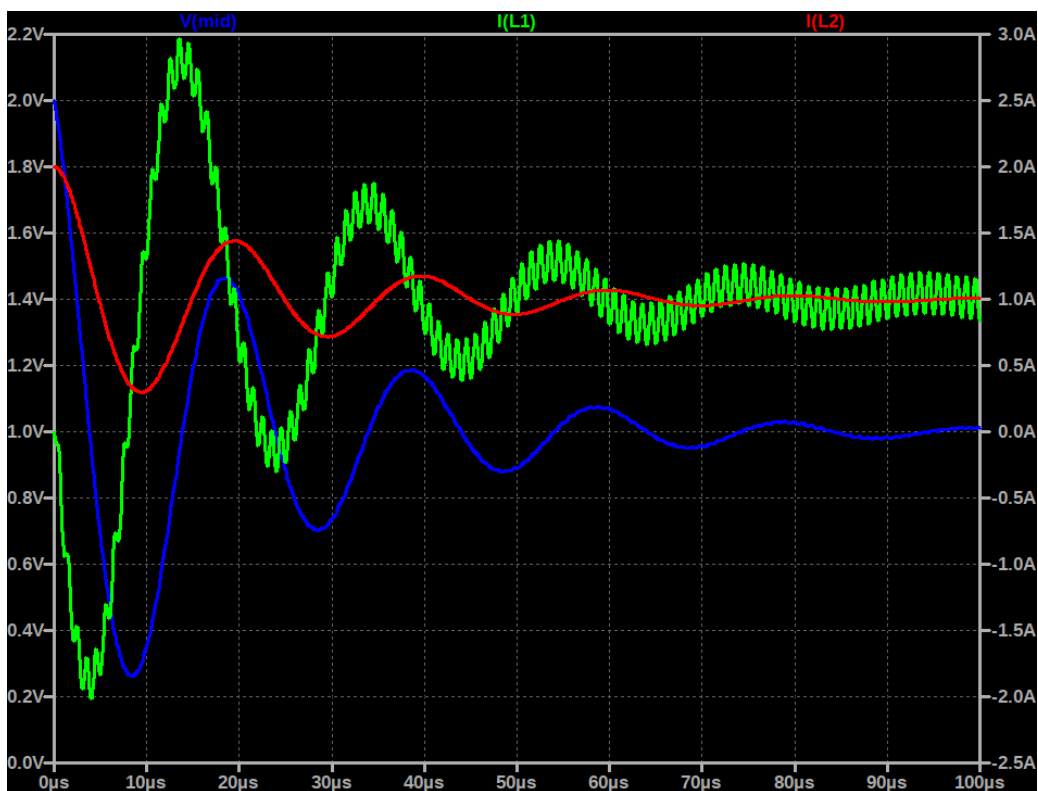
integrálással számolni a későbbi időpillanatokot. Ezzel a módszerrel kezdeti értékek megadásánál nem kell a hálózat összes csomópontjára és tekercsére kezdeti értéket megadjunk, csupán a számunkra fontosaknak!



19. ábra. Kezdeti értékek beállítása

A 19. ábrán láthatunk egy példakapcsolást, amin szemléltettük a kezdeti értékek beállítását egy T topológiájú LC szűrőn keresztül. Az ábrán látható direktívával az L1-es tekercset árammentes kezdőállapottal számoljuk, míg a C1-es kondenzátor feszültségét 2 V nagyságúra állítottuk be. Mivel az L2-es tekercsre nem állítottunk be kezdeti feltételt, így ott a szimulátor a tekercset egy rövidzárral helyettesítette, feltéve, hogy a hálózat egy állandósult állapotból indult ki. Ennek megfelelően az L2 tekercs árama 2 A-es értékkel indul, ahogy az a C1-es kondenzátor feszültségéből és az R1-es ellenállás értékéből következik. A kiadódó hullámformákat a 20. ábrán láthatjuk.

```
.ic I(L1) = 0 V(mid) = 2
```



20. ábra. Hullámformák kezdeti értékekkel

4.5. Zajszimulációk és a .noise direktíva

Az elektronika egyes alkalmazási területein fontos szempont a kapcsolások vizsgálata zajos környezetben. Tipikusan ilyen alkalmazási terület a nagyfrekvenciás környezetben alkalmazott kis zajú erősítők tervezése. Egy erősítő zaján azt az additív zajteljesítményt értjük, amit az áramkör a jelátviteli láncba történő beiktatásával hozzáad a jelekhez.

Az elektronikai kapcsolásokban előforduló zajokat több alkatrész és fizikai folyamat termeli, azonban ezekkel a folyamatokkal ebben a dokumentumban nem foglalkozunk. Szimulációs szempontból számunkra az alkatrészek saját zaja és annak szimulációja fontos, így a továbbiakban csak ezekre fókuszálunk.

A különféle zajok karakterizációjára a teljesítménysűrűség-spektrumot szokás használni. Ez a spektrum megmutatja, hogy az adott frekvencián és annak kicsiny környezetében a zajnak mekkora teljesítménye található meg.

Legelső körben az ellenállások termikus zaját vizsgáljuk. A környezeti hőmérséklet függvényében minden ellenállással bíró áramköri komponens zajt termel. Ezt a zajfajtát **Johnson és Nyquist** neveihez kötjük, azonban fehér zajnak is szokás nevezni, hiszen a zaj teljesítménysűrűség-spektruma független a frekvenciától.¹⁰ Ennek megfelelően egy Δf sávszélességű tartományon belül megtermelődő termikus zaj teljesítményét és az ebből kifejezhető ekvivalens feszültség- és áramértékeket a 7. formulákkal határozhatjuk meg. A formulákban szereplő k_B a Boltzmann állandó, míg T az abszolút hőmérséklet.

$$P = k_B T \Delta f$$

$$P = \frac{\overline{U_n^2}}{R} = \overline{I_n^2} R \quad (7)$$

Felhasználva ezeket az összefüggéseket kifejezhetjük a termikus zajjal ekvivalens RMS feszültség- és áramértékeket a 8. formulákkal. Ennek megfelelően a zajos ellenállásokat modellezhetjük egy zajmentes ellenállással és egy soros zaj-feszültségforrással, vagy egy párhuzamos zaj-áramforrással.

$$U_{n,RMS} = \sqrt{k_B T \Delta f R}$$

$$I_{n,RMS} = \sqrt{\frac{k_B T \Delta f}{R}} \quad (8)$$

Félvezető komponensekben gyakran előforduló zaj az ún. rózsaszín, villódzó vagy $1/f$ zaj. Ennek a zajnak a teljesítménysűrűség-spektruma egy frekvenciával fordítottan arányos résszel rendelkezik. Az $1/f$ zaj modellezésére az aktív áramköri komponensek (tranzistorok, IC-k és egyéb félvezetők) gyári modelljei extra forrásokat tartalmaznak. Ezeknek a forrásoknak az értékei erősen függenek az adott eszköz munkapontjától, így a zajszimuláció során az áramköri munkapont beállítása kiemelten fontos.

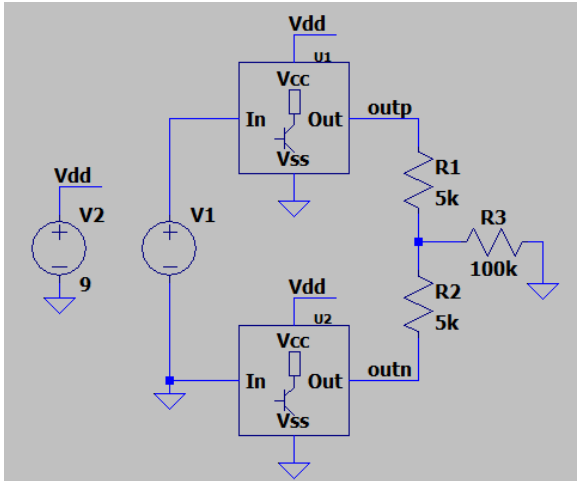
A zajok hatására az egyes csomópontokon megjelenő zajkomponensek frekvenciafüggő vizsgálatához a zajfeszültség spektrumát szokás felhasználni. Ezt a zajfeszültséget azonban nem mint egy konkrét jel feszültségének kell értelmezni! A zajfeszültség adott frekvenciájú komponense a zajteljesítmény azonos frekvenciájú és annak egy adott sávszélességén (általában 1 Hz) belül lévő komponenseiből számolt teljesítményből adódik. Ennek megfelelően a zajfeszültség szokásos mértékegysége a nV/\sqrt{Hz} .

A zajszimuláció illusztrálására alakítsuk át az előző szakaszban ismertetett differenciális audió előerősítő kapcsolást a 21. ábrának megfelelően. Zaj szimulációk futtatásához egészítsük ki a sémát a következő **.noise** direktívával:

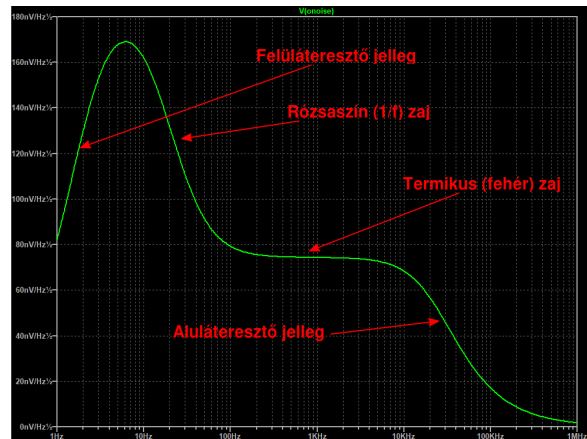
```
.noise V(outp, outn) V1 dec 100 1 1meg
```

¹⁰A különböző színű zajokat a teljesítménysűrűség-spektrumuk alapján különböztetjük meg. A kis frekvenciákat a vörös, míg nagy frekvenciákat a kék fénynek feleltetjük meg, így a kiadódó spektrumot, mint a látható fény spektrumot, tudjuk értelmezni és színekhez rendelni.

A direktíva paramétereivel beállítjuk, hogy a szimuláció során a kimenetünk az outp és outn csomópontok közötti feszültség, a kapcsolás **zajmentes** bemenete a V1 forrás, a kimeneti spektrumot pedig dekádonként 100 ponttal szeretnénk szimulálni 1 Hz és 1 MHz között. Az így előálló zajteljesítmény-sűrűség spektrumból számolt feszültségspektrumot a 22. ábrán tekinthetjük meg. Az ábrán jelöltük a zajspektrum nevezetes szakaszait. Az alul- illetve felüláteresztő jelleg az erősítő kapacitásai miatt alakul ki, melyek nemcsak a bemeneti jelet, de a zajt is szűrik. A középső lankás terület az ellenállások termikus zajából adódik és a frekvencia csökkenése miatti emelkedés pedig a tranzisztorok 1/f zajából adódik.



21. ábra. Zajszimulációs példakapcsolás



22. ábra. Zajszimuláció eredménye

Ha kíváncsiak vagyunk a szimulált frekvenciatartományon belüli zaj RMS feszültségére, úgy a szimulált jel nevére *Vnoise* a **CTRL + bal egérgomb** lenyomásával kattintva egy felugró ablakban megkapjuk az integrált zajfeszültség RMS értékét. Esetünkben ez az érték **14.243 μV** -ra adódott, vagyis az előerősítők beiktatásával **14.243 μV_{RMS}** -nyi zajt is beiktattunk a jelátviteli láncunkba!

5. Összetett szimulációk futtatása

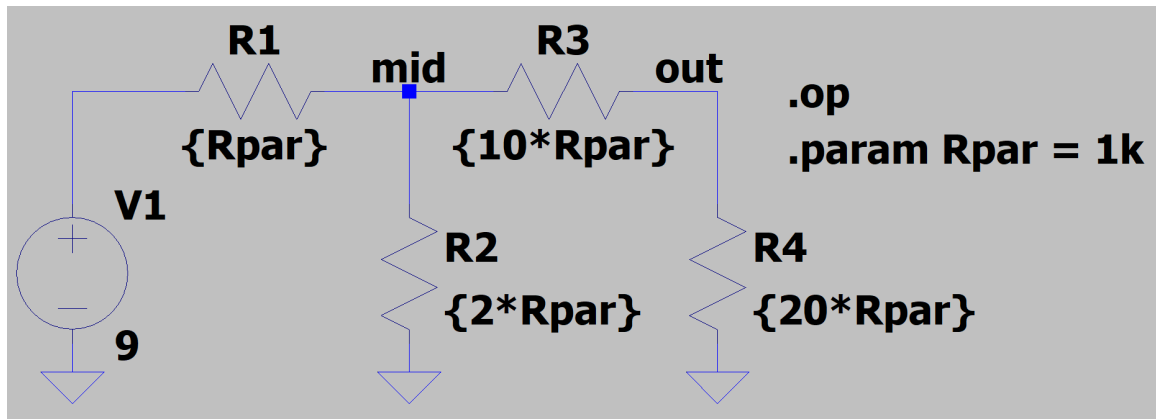
Az eddigiek során bemutattuk és megismertettük a kezelőfelület használatát, valamint bemutattuk az LTspice egyszerűbb szimulációs lehetőségeit. Ebben a fejezetben az szoftver által szolgáltatott összetettebb funkciók megismerésére fogunk fókuszálni, melyek lehetővé teszik a szoftver alkalmazását a mindennapi mérnöki életben előforduló problémák modellezésére és számítására.

5.1. Parametrikus szimulációk

Az LTspice lehetőséget biztosít az alkatrészértékek parametrizálására ún. séma-változók bevezetésével. Ezeket a változókat felhasználhatjuk komponensek értékeinek megadására, vagy mint matematikai változók melyek formulákba illeszthetők be. Az LTspice-ban paraméterek megadása a `{ }` zárójelekkel történik. A kapcsolószárójel között szereplő kifejezések a szimulációk futtatása előtt kiértékelődnek és a szimuláció már a kiértékelt valós számokkal fog végig futni. Paraméterezés használatának illusztrálására nézzük meg a 23. ábra kapcsolását.

Az ábrán egy kaszkádosított feszültségosztót valósítottunk meg, melyben az első fokozat az R1 és R2 ellenállásokból, míg a második fokozat az R3 és R4 ellenállásokból áll. Az ellenállásosztó minden ellenállása egy parametrikus ellenállásértéktől függ, ez a paraméter az **R_{par}** változó. A paraméter függvényében az R1-R4 ellenállások a séma-változó 1x, 2x, 10x és 20x szorzását veszik fel, ezzel skálázva az ellenállásosztó be- és kimeneti impedanciáját. A sémán elhelyezett parametrikus séma-változók értékeinek megadásához a **.param** direktívát kell használnunk. Ez a sémán a következő szintaktikával szerepel.

```
.param Rpar = 1k
```



23. ábra. Változó paraméterek használata

A `param` direktíva után a paraméterek és értékeiknek megadása történik ún. kulcs-érték párok alkalmazásával. Ebben az esetben az `Rpar` változónknak $1\text{ k}\Omega$ -s értéket adunk. Ezt az értéket a szimulátor behelyettesíti a kapcsolásképletek közötti kifejezésekbe (R1-R4 ellenállások értékei), majd elvégzi a szükséges szimulációt, amely a munkapont meghatározás ebben az esetben. Ezzel a módszerrel könnyen tudjuk állítani a séma több alkatelemének az értékét egyidejűleg.

A kapcsolásképletek közötti kifejezésekben akár bonyolultabb matematikai formulákat is alkalmazhatunk az adott komponens értékének megadásánál. Ebben az esetben az alkalmazható matematikai és logikai függvények nagyrészt megegyeznek a tetszőleges viselkedésű források esetén elérhetőekkel, így a bővebb lista felelevenítése érdekében kérjük az olvasót a 3. fejezet B forrásokról szóló szakaszának átismétlésére.

Valóságos szimulációs modellek esetében előfordul, hogy a séma kapcsolását nem csupán paraméterekkel szeretnénk ellátni, hanem ezeket a paramétereket szeretnénk változtatni egy előre definiált minta alapján. Ebben az esetben tudjuk használni a `.step` direktívát. Ehhez módosítsuk a 23. ábra `.param` direktíváját a következőképpen:

```
.step param Rpar 1k 10k 500
```

Ezzel a direktívával megadhatjuk az LTspice számára, hogy a megadott szimulációt végezze el az `Rpar` változó értékeinek változtatásával. A paraméter értékeinek sora megadható a 3. fejezet `.dc` szimulációjánál látottakkal (lin, dec, oct és list) módon. Ennek értelmében a fenti direktíva az `Rpar` értékét $1\text{ k}\Omega$ és $10\text{ k}\Omega$ között fogja változtatni $500\text{ }\Omega$ -s léptékekkel, ennek eredményeként összesen 19 szimulációt fog lefuttatni a szoftver.

A `.step` direktívával nem csak paraméterek értékeit tudjuk változtatni, hanem független források értékeit is (hasonlóan a `.dc` direktíva felhasználásával), hőmérsékletet is a `.step TEMP` direktívával, valamint áramköri modell paramétereit is. Ha csak pár hőmérsékleti értéket szeretnénk változtatni, úgy használhatjuk a `.temp` direktívát is, mely után fel kell sorolnunk a szimulálandó hőmérsékleti értékeket $^{\circ}\text{C}$ -ban értelmezve. Ezen kívül lehetőségünk van több paramétert is változtatni, ebben az esetben a megadott paraméterek minden lehetséges kombinációját elő fogja állítani a szoftver, így a teljes paraméterter le lesz fedve, azonban ennek következtében gyorsan megnőhet a szükséges szimulációk száma. Gondoljunk csak bele, ha 3 paramétert szeretnénk 20 különböző értékkel szimulálni, úgy a lehetséges kombinációk száma $20^3 = 8000$, azaz ennyi szimuláció futtatása lesz szükséges!

Abban az esetben, ha nem szeretnénk a séma változóik által kifizített paraméterter összes lehetséges kombinációját szimulálni, csak bizonyos kombinációkat, lehetőségünk van a `table()` függvény használatára. A függvény a sémaparaméterek összetartozó kombinációinak megadását teszi lehetővé. Az összetartozó kombinációkat egy indexváltozóval tudjuk megkülönböztetni, és a szimulációk során ezt az indexváltozót tudjuk léptetni a `.step` direktíva felhasználásával. A módszer illusztrálásához változtassuk meg a 23. ábra kapcsolásának R3 és R4 ellenállás értékét `{Rpar2}` és `{2*Rpar2}` kifejezésekre és módosítsuk a sémán szereplő direktívákat a következők szerint:

```
.op
.param Rpar=table(x,0,1k,1,2k,2,5k,3,10k)
.param Rpar2=table(x,0,100,1,200,2,500,3,1k)
.step param x list 0 1 2 3
```

A fenti direktívák használatával a kaszkádosított feszültségosztónkon a 2. táblázat szerinti paraméterkombinációk fognak lefutni.

x	0	1	2	3
Rpar	1k	2k	5k	10k
Rpar	100	200	500	1k

2. táblázat. Meghatározott sémaváltozókombinációk megadása

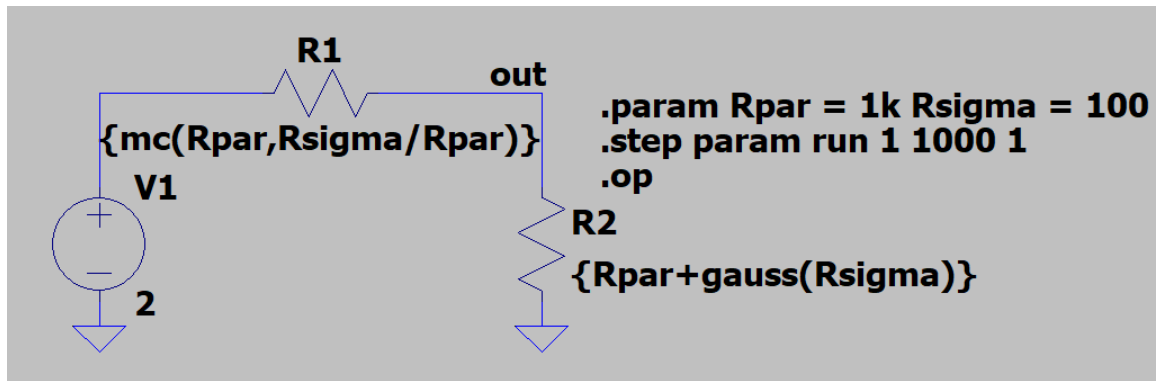
Több paraméter változtatása esetén az ábrázolandó mennyiségek és jelek ábrázolása nem magától értendő feladat, így erre a 6. fejezetben még vissza fogunk térni és részletesen be fogjuk mutatni, hogyan lehet kiválasztani a paraméterhalmaz futó változóját.

5.2. Monte Carlo szimulációk

Az eddigiek során ismertetett szimulációs eljárások és kapcsolások során a sémán szereplő alkatrészek fix és névleges értéket képviseltek, és a szimulátorunk ezen értékek alapján végezte el a szükséges számításokat. A valóságos alkatrészeink ezzel ellentétben ugyan egy fix értékkel rendelkeznek (környezeti változásoktól eltekintve), azonban ezek az értékek általában nem esnek egybe az alkatrész névleges értékével. Ezzel a ténnyel az alkatrészgyártók is tisztában vannak, így az alkatrészekhez hozzárendelnek egy toleranciaosztályt is. Ennek megfelelően az általunk megvásárolt $1\text{ k}\Omega$ -s 5%-s ellenállás tényleges ellenállása a névérték körül 5%-t tud szórni, vagyis a névleges $1\text{ k}\Omega$ -s ellenállás értéke $950\ \Omega$ és $1050\ \Omega$ között bármi lehet. A névértéktől vett eltérés következtében kapcsolásunk viselkedése megváltozhat pl. a szűrőnk törésponti frekvenciája eltolódik, vagy a feszültségosztó kimeneti feszültsége megváltozik. Ezen hatások vizsgálata érdekében szükséges lehet változó paraméterekkel többször elvégezni a szimulációt és a szimulációs eredményekből statisztikát készíteni. A metodika ennek köszönhetően lett elnevezve Monte Carlo módszernek, hiszen a Monacóban található város a kaszinóiról, így a szerencsejátékról és a valószínűségekről híres.

Az alkatrészparaméterek véletlenszerű változtatásához véletlenszámokra és parametrikus alkatrészértékekre van szükségünk. Ehhez kihasználhatjuk, hogy az LTspice a $\{\}$ -k között megadott kifejezéseket a szimuláció lefuttatása előtt kiértékeli és behelyettesíti, valamint a kapcsolósárójelek közötti kifejezések tartalmazhatnak különböző matematikai függvényeket is. Monte Carlo szimuláció során különböző valószínűségi eloszlásokkal tudjuk befolyásolni paramétereink értékeit. A két leggyakrabban használt függvény a **mc** és a **gauss**. Előbbi függvény egyenletes, míg utóbbi Gauss-eloszlású véletlen számokkal tér vissza minden egyes futtatáskor. A függvények használatához meg kell adnunk azok argumentumait is. Az mc függvény két argumentumot vár **mc(x, y)** formában és visszaad egy $x(1+y)$ és $x(1-y)$ közötti értéket, egyenletes eloszlással. A gauss függvény ezzel ellentétben csupán egyetlen paramétert vár **gauss(x)** formájában, majd visszaad egy 0 várható értékű és x szórással rendelkező Gauss-eloszlású véletlen értéket. A függvények használatához figyeljük meg a 24. ábra kapcsolását.

A kapcsolat egy egyszerű két darab ellenállásból álló feszültségosztót valósít meg. R1 és R2 ellenállások értékeit az Rpar és Rsigma paraméterek segítségével határozzuk meg a 9. egyenletek szerint.



24. ábra. Monte Carlo szimulációs példakapcsolás

$$\begin{aligned}
 R_{1,low} &= R_{par} \left(1 - \frac{R_{sigma}}{R_{par}}\right) = R_{par} - R_{sigma} \\
 R_{1,high} &= R_{par} \left(1 + \frac{R_{sigma}}{R_{par}}\right) = R_{par} + R_{sigma} \\
 R_1 &\sim U(R_{1,low}, R_{1,high}) \\
 R_2 &\sim \mathcal{N}(\mu = R_{par}, \sigma^2 = R_{sigma}^2)
 \end{aligned} \tag{9}$$

A fentiekén kívül a sémán megtalálható még a **.param** direktíva, mellyel megadjuk a felhasznált paraméterek értékeit (R_{par} és R_{sigma}) formájában, valamint megadjuk, hogy a munkaponti **.op** szimulációt 1000x futtassuk le. A futások számának megadása a **.step param run** direktívával történt.

A szimulációt elvégezve a kérdéses jelek pl. az **out** csomópont feszültsége vizsgálhatóvá válik. Ezzel a módszerrel ugyan meg tudjuk vizsgálni a véletlenszerű alkatrészértékek hatását a számunkra fontos jelekre és paraméterekre, azonban az eredményekről számszerű statisztika készítésére nincs lehetőségünk az LTspice-on belül. Ehhez a sémán automatizált mérési utasításokat kell elhelyeznünk, és a kiadódó logfájlból kell kinyerjünk a szimulációs eredményeket, melyek utána kiértékelhetők az általunk választott adatkezelő szoftverrel pl. Excel, Matlab vagy Python segítségével.

5.3. Áramkör vizsgálata diszkrét frekvenciákon

Az előző fejezetben ismertetett nagyjelű tranziens szimuláció kapcsán bevezettük a Teljes Harmonikus Torzítás fogalmát. Ezzel a mérőszámmal számszerűsíthető egy nemlineáris hálózat torzítása, azaz az egyfrekvenciájú szinuszos bemenet hatására létrejövő felharmonikus tartalom. Ebben a szakaszban bemutatjuk, hogy hogyan automatizálható ennek a mérőszámnak a meghatározása LTspice segítségével. A szimuláció során használjuk fel a 10. ábra kapcsolását és egészítsük ki azt a következő direktívákkal:

```
.tran 0 40m 30m 1u
.four 1k 10 V(outp,outn)
```

A **.tran** direktíva hatására egy nagyjelű szimulációt végzünk 40 ms-ig, és annak csak az utolsó 10 ms-t rögzítjük, míg a **.four**¹¹ direktíva segítségével megadjuk, hogy a bementi frekvenciánk 1 kHz, kimeneti jelünk az outp és outn csomópontok közötti feszültség és ennek a jelnek szeretnénk 10 db harmonikusát meghatározni, beleértve az alapharmonikust is. A szimuláció eredményeként előáll a 25. ábra szerinti kimenet, melyet a logfájlból találunk. A logfájl előhívásához használjuk a **(View > SPICE Output Log)** menüpontot vagy a **CTRL+L** gyorsgombot.

¹¹A direktíva a Fourier név rövidítése.

```

N-Period=1
Fourier components of V(outp,outn)
DC component:2.05559e-05

Harmonic      Frequency      Fourier      Normalized      Phase      Normalized
Number        [Hz]          Component    Component       [degree]    Phase [deg]
1             1.000e+3      1.360e-1    1.000e+0        -88.61°     0.00°
2             2.000e+3      6.867e-7    5.050e-6         89.55°     178.16°
3             3.000e+3      1.040e-6    7.650e-6         84.56°     173.17°
4             4.000e+3      3.431e-7    2.523e-6         89.73°     178.34°
5             5.000e+3      2.745e-7    2.019e-6         89.81°     178.42°
6             6.000e+3      2.287e-7    1.682e-6         89.85°     178.45°
7             7.000e+3      1.960e-7    1.441e-6         89.86°     178.47°
8             8.000e+3      1.715e-7    1.261e-6         89.87°     178.48°
9             9.000e+3      1.524e-7    1.121e-6         89.89°     178.50°
10            1.000e+4      1.370e-7    1.007e-6         89.92°     178.53°
Partial Harmonic Distortion: 0.001016%
Total Harmonic Distortion: 0.000000%

```

25. ábra. Diszkrét frekvenciájú Fourier-komponens analízis eredménye

Az eredményeket megnézve látható, hogy az analízis a jelek 1 periódusából számolt értékeit tartalmazza (N-Period=1), ez azonban beállítható a `.four` direktíva módosításával. Pl. ha szeretnénk a jel 5 periódusát felhasználni a számítások elvégzéséhez, úgy a direktívánk a következőképpen módosul:

```
.four 1k 10 5 V(outp,outn)
```

Az eredmények között megtalálhatjuk az egyes harmonikusokhoz tartozó jelamplitúdók és fázisok értékeit, valamint ezen értékek alapharmonikusra normalizált értékeit. A logfájl végén pedig számszerűsítve megtalálhatjuk a számított részleges (Partial) és teljes (Total) harmonikus torzítás értékeit.

5.4. Kétpólus és kétkapú paraméterek számítása

Nagyfrekvenciás technikában alkalmazott áramkörü elemek esetében gyakran találkozhatunk az illesztett be- és kimenetek, illetve impedanciák fogalmával. Ezen áramkörü elemek legtöbbször, mint lineáris kétpólusok és kétkapuk vannak kezelve, ennek megfelelően alkalmazhatjuk rájuk a lineáris hálózatelmélet fogalom- és eszköztárát. Az LTspice-ban a hálózati portokkal kapcsolatos szimulációk elvégzéséhez a `.net` direktívát kell használnunk, mely a `network` angol szó rövidítéséből származik. A direktíva mellett szükséges még egy `.ac` direktíva futtatása is, mellyel a szimulálandó frekvenciatartományt tudjuk megadni és ezen a tartományon fognak lefutni a `.net` direktíva számításai.

Kétpólusok vizsgálatához meg kell adjuk a kétpólus kapcsaira csatlakozó feszültségforrás azonosítóját, valamint opcionálisan megadhatjuk a kétpólust meghajtó forrás belső ellenállását. Ez a belső ellenállás a nagyfrekvenciás technikában leggyakrabban 50 vagy 75 Ω . Kétpólusvizsgálathoz figyeljük meg a 26. ábra kapcsolását ahol egy reflektorral ellátott dipól antenna illesztőhálózatát szimuláltuk. Az antenna frekvenciafüggő komplex impedanciáját egy egyedi szimbólummal és egy aláramkörü leírást tartalmazó fájlal adtuk meg.¹²

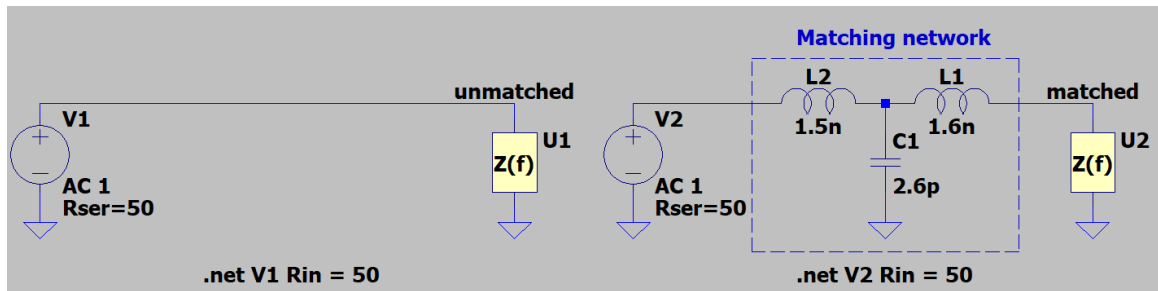
```

.ac lin 1000 2G 2.9G
.include antennaImpedance.sub
.net V1 Rin = 50
.net V2 Rin = 50

```

A felhasznált direktívákkal a 2.4 GHz-es WiFi sáv és annak 400 MHz-es környezetében (2 GHz-től 2.9 GHz-ig) végzünk összesen 1000 különböző frekvencián szimulációkat, majd pedig ezen tartományon meghatározzuk a kétpólusok viselkedését leíró paramétereket a frekvencia függvényében. Az `include` direktíva húzza be az antenna viselkedését leíró `.sub` fájl, mely az antenna bemeneti impedanciáját modellező aláramkört tartalmazza.

¹² Az antenna bemeneti impedanciája az antennageometria erős függvénye. A bemeneti impedancia frekvenciafüggésének meghatározásához az elektromágneses tér numerikus számítása szükséges. Ez a feladat túlmutat ezen dokumentum hatáskörén, így nem részletezzük. Az érdeklődő olvasó figyelmébe ajánljuk a Matlab Antenna Toolbox programját.

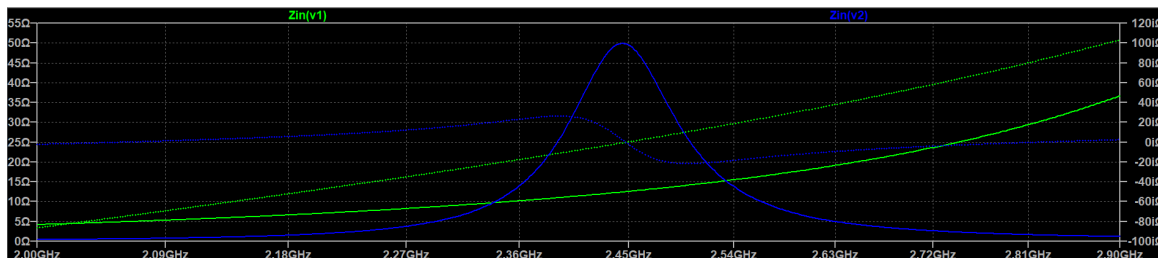


26. ábra. Kétpólusok vizsgálata .net direktívával

Kétpólusok esetében az LTspice meghatározza a kétpólus bemeneti impedanciáját, bemeneti admintanciáját és a reflexiótényezőjét¹³. Ezen paraméterek a frekvencia függvényében ábrázolhatóak a $Z_{in}(V1)$, $Y_{in}(V1)$ és $S_{11}(V1)$ ¹⁴ mennyiségek ábrázolásával.

Fontos megjegyezni, hogy a .net direktívák esetében a bemeneti és a későbbiekben a kimeneti ellenállás megadása opcionális és bizonyos esetekben el is hagyható. Alapértelmezés szerint minden kapu 1 Ω -s ellenállással van lezárva. A bemeneti feszültségforrás soros ellenállását beállítva, valamint a direktívát kiegészítve az Rin kulcsszóval ez az érték lecserélhető. A két módszer a szimuláció során eltérő eredményeket tud adni, így ügyelni kell rájuk! A forrás soros ellenállásának beállításával az .ac szimuláció során minden jel ezen ellenállás figyelembevételével fog szimulálódni, míg ha csak a .net direktívában adjuk meg, úgy csak a hálózati paraméterek számításánál lesz figyelembe véve!

A 27. ábrán megfigyelhetjük, hogy az antenna és a jelforrás közé kapcsolt illesztőhálózatnak köszönhetően a jelforrás az antennát egy tisztán 50 Ω -s impedanciaként látja a WiFi sáv környezetében. Az ábrán a folytonos vonalak az impedanciák valós részét ábrázolják és a bal oldali skála alapján értelmezendők, míg a szaggatott vonalak a képzetes részt mutatják és a jobb oldali skála alapján értelmezendők.



27. ábra. Antenna bemeneti impedanciája illesztőhálózat nélkül (zöld) és illesztőhálózattal (kék)

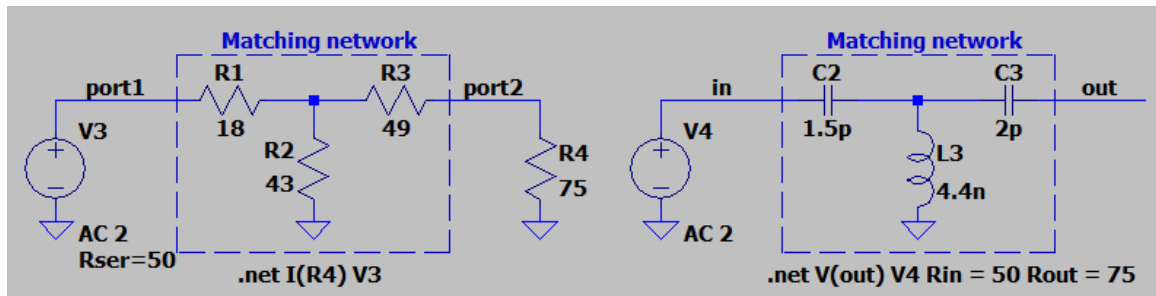
Az illesztőhálózatnak köszönhetően az antennára jutó hatásos teljesítményt 45 - 55%-kall tudtuk megnövelni, valamint ezzel egyidőben illesztettük is az antennát a meghajtó hálózatához, ezáltal minimalizálva a gerjesztő hálózat felé reflektálódó teljesítményt.

Kétkapuk szimulálásához a .net direktívát még ki kell egészíteni a kimeneti kaput képező jellel. Ezt kétféleképpen tudjuk megadni a szimulátor számára. Egyrészt megadhatjuk a kimeneti kapu kapcsait egy feszültségjellel, avagy megadhatjuk a kimeneti kaput lezáró ellenállást.

A 28. ábra kapcsolásain a bal oldalon egy rezisztív impedanciaillesztő hálózatot láthatunk, melynek bemenetét a V3-as feszültségforrás képezi és kimenetét pedig az R4-es ellenállás zárja le. Az ábra jobb oldalán

¹³Nagyfrekvenciás technikán feszültség és áramjelek mérése igencsak nehézkes tud lenni, hiszen a mérendő mennyiségek erősen függenek a mérővezetékek hosszától és azok egymáshoz képesti helyzetétől. Ezen a területen ezért általában teljesítmények áramlását mérik és meghatározzák a kétpólusra érkező és az onnan reflektálódó teljesítményeket.

¹⁴Nagyfrekvenciás technikában előszeretettel használják a kétpólusok és sokkapuk leírására az ún. szórési paramétereket. Ezzel a leírásmóddal megadhatók a sokkapuk portjaira érkező és onnan visszaverődő teljesítményhullámok. Kétpólusok esetében az S11 paraméter gyakorlatilag a komponens reflexiótényezőjével egyezik meg.



28. ábra. Impedanciaillesztő kétkapú hálózatok

egy reaktív illesztőhálózat figyelhető meg. Mind a két hálózat a kimeneti valós 75Ω -s ellenállást illeszti a bemeneti valós 50Ω -ra, azonban a rezisztív hálózat frekvenciafüggetlenül teszi ezt miközben a teljesítményből 10 dB -t disszipál. Ezzel ellentétben a reaktív változat frekvenciafüggően végzi az impedanciaillesztést, azonban teljesítményt nem disszipál!

```
.net I(R4) V3
.net V(out) V4 Rin = 50 Rout = 75
```

Az első direktíva felhasználásával megadhatjuk, hogy a 28. ábra rezisztív illesztőhálózatának az R4-s ellenállás a lezárása és a V3-as forrás a gerjesztése. Figyeljük meg, hogy ebben az esetben nem adtuk meg a bemeneti és kimeneti impedanciákat, hiszen a V3-as forrásnál beállítottuk annak soros ellenállását, valamint a kimeneti portot egy ellenállás áramával határoztuk meg, így automatikusan ezen ellenállás értéke tekintendő illesztett lezárásnak. A direktíva második sorával a reaktív hálózat kimenetét és bemenetét állítottuk be, valamint megadtuk a kapuk referencia ellenállásait. Fontos megjegyezni, hogy a kétkapuparaméterek számításakor a kimeneti lezárás figyelembe lesz véve, így azt külön a kapcsoláson beiktatni nem kell!

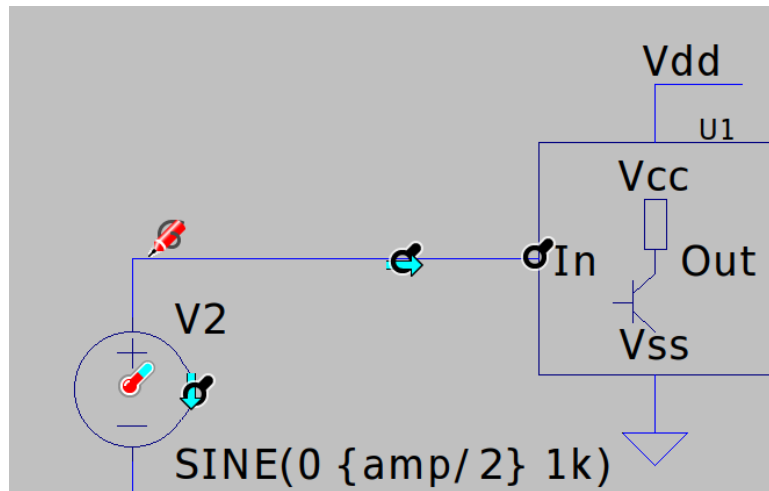
Kétkapuk esetében az LTspice meghatározza a kétkapuk viselkedését leíró impedancia-, admittancia-, hibrid- és szórás mátrixokat, ezen kívül a kapuk felé néző be- és kimeneti impedanciákat is.

6. Szimulációs eredmények vizualizációja és utófeldolgozása

Az előző fejezetekben megismertettük az olvasót az LTspice alapvető használatával, beleértve a különböző komponensek beállítási lehetőségeit valamint a futtatható szimulációkat. Ebben a fejezetben bemutatjuk az LTspice utófeldolgozást segítő és az elvégzett szimulációk eredményeinek vizualizációját támogató funkcióit.

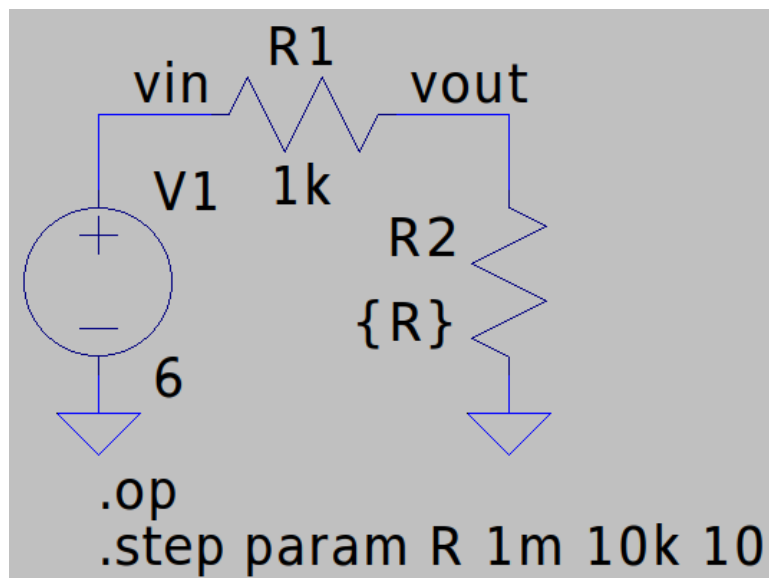
6.1. Hullámformák ábrázolása

Alapértelmezés szerint a szimulációk lefutását követően a sémakeresztő mellett megjelenik a hullámforma megjelenítő ablak is. Ebben az ablakban tudjuk ábrázolni a különböző jeleinket a szimulációs lépések függvényében. A szimulációk végeztével a hullámforma megjelenítő ablak lesz aktív, így a séma különböző vezetékerei fölé navigálva az egérrel több új szimbólum jelenik meg a szokásos egérkurzor helyett. Az egyik leggyakrabban használt eszköz a vezetékek potenciálját leolvadó piros színű feszültségmérő tű. Ebben a pozícióban maradva, ha az **ALT** billentyűt is lenyomva tartjuk, úgy az adott vezetéken átfolyó áramot tudjuk megmérni a lakatfogó ikonnal. Ebben az esetben a vezeték áramát a kék színű referenciairány mutatja majd. Standard kétpólusokra navigálva hasonlóan megmérhetjük a kétpóluson átfolyó áramot. Ha közben az **ALT** billentyűt is lenyomva tartjuk, úgy megjelenik a hőmérő ikon, mellyel a kétpólus teljesítményét tudjuk leolvasni. Végül pedig ha egy több kivezetéssel rendelkező sokpólussal van dolgunk, úgy az adott pólus kapcsai felé navigálva megjelenik az áramszimbólum nélküli lakatfogó, mellyel az adott kivezetés áramát jeleníthetjük meg. Az ikonokat és a hozzájuk tartozó pozíciókat a 29. ábrán szemléltetjük.



29. ábra. Különböző mérőeszközök villamos jelek megjelenítéséhez

A további lehetőségek illusztrációja végett módosítsuk az 1. ábra kapcsolását a 30. ábra szerint. A szimulációk során az R2 ellenállás értékét változtatjuk a `.step` direktívával $1\text{ m}\Omega$ és $10\text{ k}\Omega$ között $10\text{ }\Omega$ -s lépésként. A szimuláció végeztével ábrázoljuk a kimeneti feszültséget (vout csomópont potenciálja) és a kimeneti áramot (R2 ellenállás árama), így a 31. ábrán látott hullámformákat kapjuk.

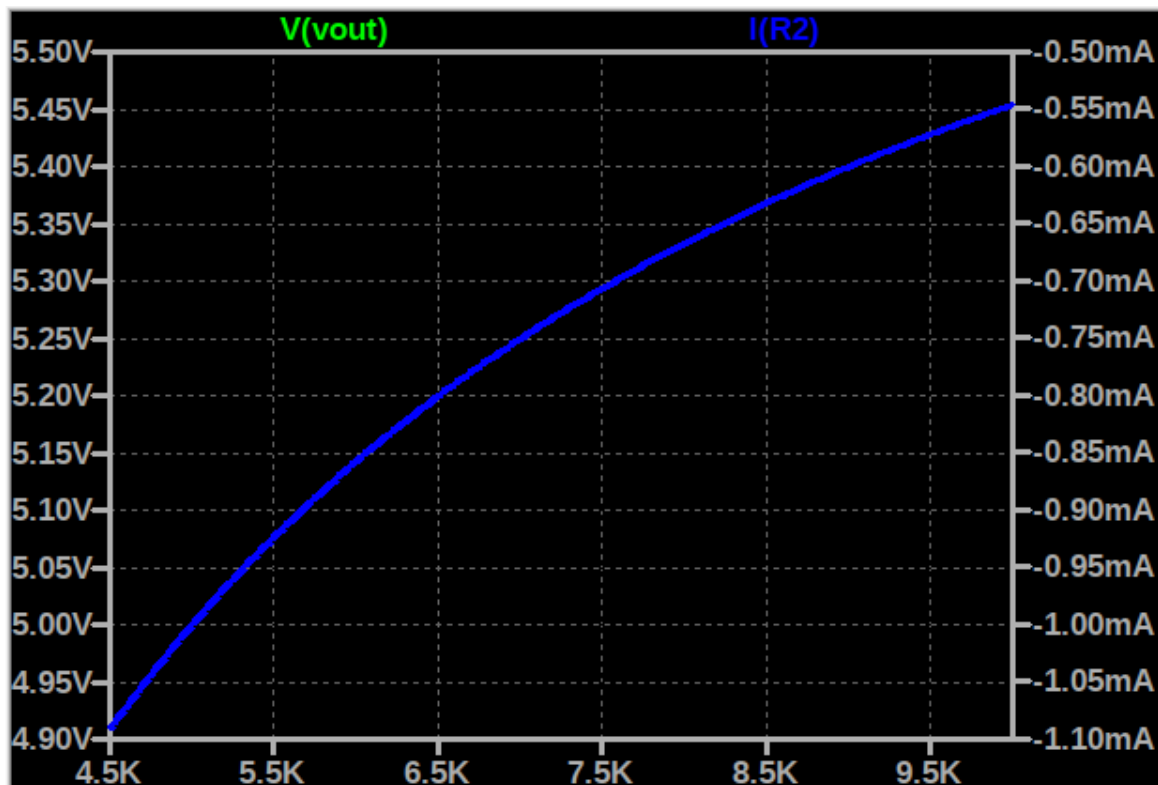


30. ábra. Paraméterezett feszültségosztó kapcsolás

Mivel az R2 ellenállás sémára történő elhelyezésekor az 1-es csomópont került alulra és a 2-es felülre, így az alapértelmezett áramirány az ellenálláson alulról felfele mutat, így a rajta átfolyó áram előjele negatív értékű, ahogy az a 31. ábra jobb oldali tengelyén látszik is. A negatív áramértéket elkerülhetjük, ha a sémakeresztő felületen megfordítjuk az R2 ellenállást, vagy a hullámforma megjelenítőn az I(R2) jelre jobb egérgombbal kattintunk és a felugró **Expression Editor** felületen átírjuk az ábrázolandó kifejezést $-I(R2)$ -re!

Az R2 ellenálláson disszipálódó teljesítményt az **ALT** billentyűt lenyomva tartva és az ellenállásra kattintva tudjuk ábrázolni a legegyszerűbben. Ekkor megjelenik egy új görbe az ábrán, valamint a bal oldalt megjelenik még egy tengely, mely az újonnan hozzáadott jelünkhöz tartozik.

Az ábrázolandó jeleken matematikai műveleteket is végezhetünk. Az alkalmazható függvények és operátorok nagyrészt átfednek a 3. fejezetben a B forrásoknál ismertekkel, kiegészülve néhány ábrázolásra vonatkozó

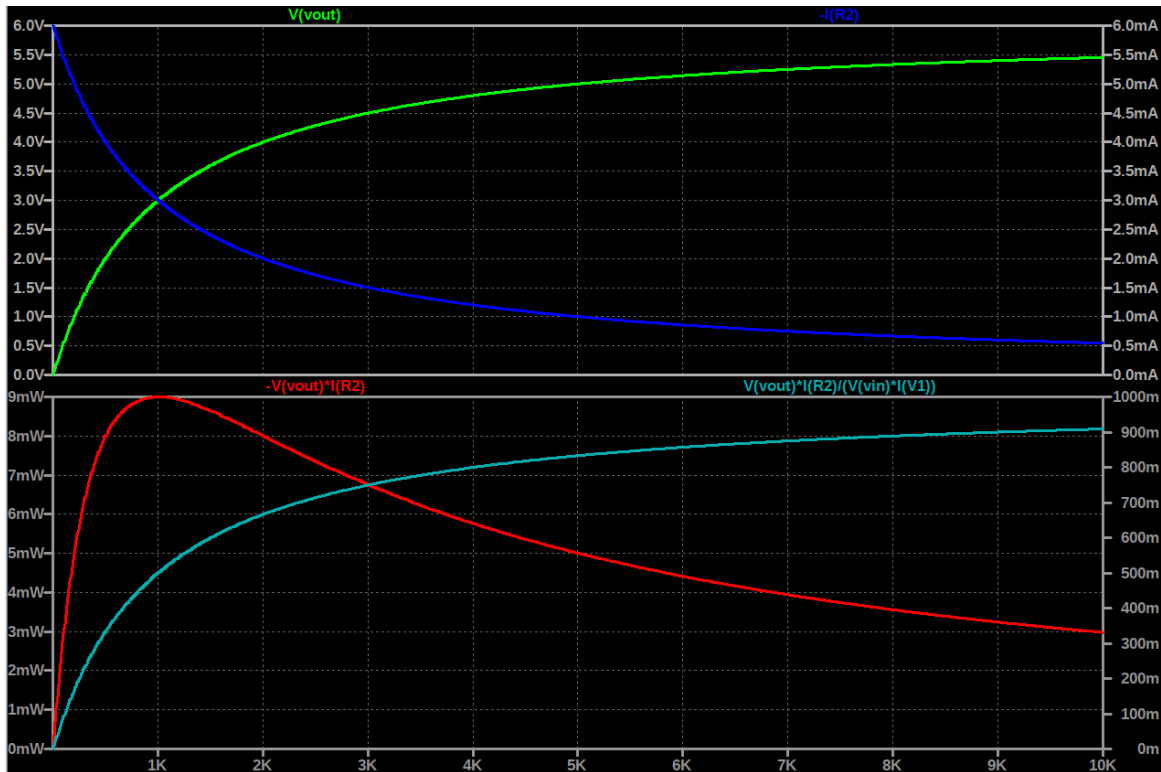


31. ábra. Feszültségosztó kimeneti feszültsége és árama

függvénnyel. Ilyen függvény például a numerikus derivált operátora, a $D()$ függvény. Ezeket a matematikai műveleteket felhasználva ábrázolni tudjuk a kapcsolás teljesítményátadási hatásfokát. A hatásfokot úgy határozzuk meg, hogy az $R2$ ellenállás teljesítményét elosztjuk a $V1$ forrás által leadott teljesítménnyel. A kétpólusokon eső feszültség és a rajtuk átfolyó áram szorzata mind az $R2$ mind a $V1$ kétpólusok esetében negatív, így a hatásfok számításakor a negatív előjelek kiesnek. Ennek megfelelően a teljesítményátviteli hatásfokot a következő formulával tudjuk számolni: $V(\text{vout}) \cdot I(R2) / (V(\text{vin}) \cdot I(V1))$. Összetettebb jelek ábrázolásához azokat manuálisan kell az ábrához hozzáadjuk (**Jobb klikk > Add Traces**), vagy az **A** gyorsgomb lenyomásával. Ekkor felugrik egy ablak, melyben kiválaszthatjuk az ábrázolandó jeleket, valamint alul az **Expression(s) to add:** szövegdobozba be is írhatjuk az ábrázolni kívánt mennyiségünk formuláját.

Ezzel összességében az ábránkon 4 mennyiséget ábrázoltunk, melyek mind különböző mértékegységgel rendelkeznek, így 4 külön skála tartozik hozzájuk (kettő bal oldalt és kettő jobb oldalt). Annak érdekében, hogy az ábráink jobban áttekinthetőek legyenek, adjunk hozzá a hullámforma megjelenítőhöz ábrázolási síkokat (**Jobb klikk > Add Plot Pane Above/Below**) opciók, vagy a **P**, **B** gyorsgombok felhasználásával. Az így megjelenő ábrázolási síkra egyszerűen áthúzhatjuk az ott ábrázolni kívánt jeleinket, ha a jelek nevénél kezdjük el vonszolni őket.

Jeleink ábrázolásakor a mértéket adó skálák automatikusan generálódnak a jelek terjedelméhez illeszkedve. Abban az esetben, ha ezen skálák nem illeszkednek az elvártak megfelelően a jelszintekhez, lehetőségünk van módosítani azokon. Ehhez kattintsunk jobb egérgombbal a módosítani kívánt függőleges skálára és a felugró ablakban állítsuk be a skála minimális és maximális értékét, a skála léptékét (lineáris vagy logaritmikus), valamint az ábrán lévő rács osztásközét. Ha nem szeretnénk, hogy az ábránkon látszódjon a leolvasást segítő rács, úgy azt ki tudjuk kapcsolni a (**Jobb klikk > View > Grid**) opció kikapcsolásával. Ezekkel a beállítási lehetőségekkel elő tudunk állítani a 32. ábrához hasonló ábrát. Az ábrán jól látható, hogy az $R2$ ellenállás növelésével a teljesítményátviteli hatásfokunk ugyan monoton nő, de az átvihető teljesítmény $1 \text{ k}\Omega$ -nál veszi fel a maximumát és onnantól kezdve monoton csökken. Maximális teljesítményátvitel esetében a hatásfok éppen 50%!



32. ábra. Több jel ábrázolása külön ábrázolási síkokon

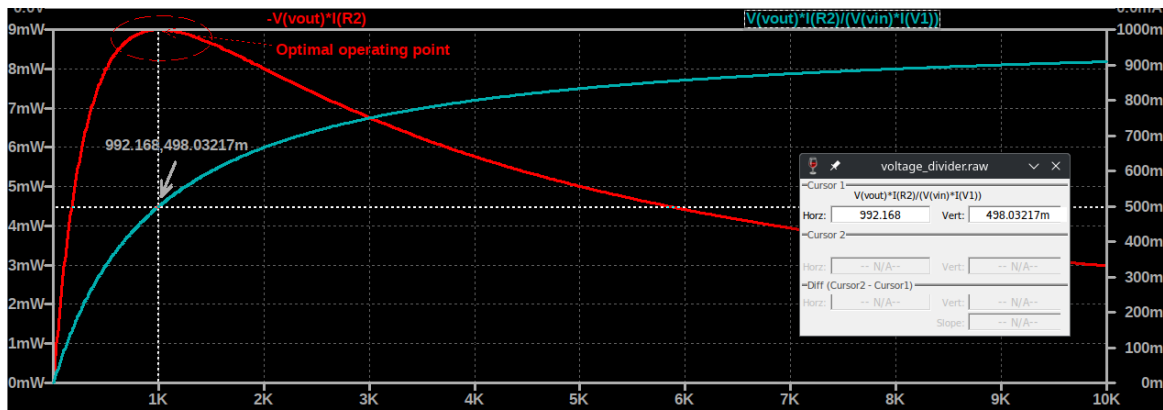
Az áramköri paraméterek változtatása esetén (.step vagy a .dc direktívák használata) előfordulhat hogy az ábránkon a vízszintes tengelyen nem a kívánt paraméter szerinti függését látjuk a jelalakjainknak. Ekkor lehetőségünk van a vízszintes tengelyre jobb egérgombbal kattintva beállítani, hogy melyik paramétert szeretnénk az x-tengelyen ábrázolni valamint, hogy ezt a paramétert milyen határok között és milyen lépésközzel szeretnénk ábrázolni, hasonlóan a függőleges tengelyek esetén.

Az ábrák pontosabb leolvashatósága érdekében lehetőségünk van kurzorok segítségével több tizedesre pontosan meghatározni a szimulált értéket egy adott paraméterállás esetén. Ehhez kattintsunk egyszer a mérni kívánt jel nevére! Ekkor a jelünkön megjelenik egy fehér célkereszt, kétszeri kattintás hatására pedig két célkereszt formájában kettő darab kurzor adódik hozzá a jelünkhöz. A kurzorok által jelölt pontos értéket a kis méretű felugró ablakban láthatjuk. Ebben szerepelnek az 1-s illetve 2-es kurzor horizontális és vertikális koordinátái, valamint a kurzorok vertikális és horizontális különbségeiből származtatott értékek. A kurzorok pozicionálásához horizontális irányban tudjuk őket vonszolni, míg vertikális irányban mindig a jelünkhöz fognak illeszkedni!

Az ábrázolt jelalakokon elhelyezhetünk különböző jelmagyarázó jelölőket is a **(Jobb klikk > Notes & Annotations)** menüpont alpontjaiból válogatva. Ezekkel a nevezetes és fontos pontokat vagy tendenciákat szoktok jelölni, ezzel is fókuszálva az ábrát néző mérnök figyelmét a lényeges részletekre, ahogy az a 33. ábrán is látható. Annak érdekében, hogy az ábrázolt hullámformáink jobban láthatóak legyenek lehetőségünk van az ábrázolási felület beállításait is módosítani a **(Tools > Settings > Waveforms)** menüpont alatt. Itt át tudjuk állítani a hullámformák és kurzorok vastagságát valamint a feliratok betűtípusát és betűméretét is.

6.2. Fourier-transzformációs analízis

Villamosmérnöki munkák kapcsán nagyon gyakran előfordul, hogy a mért vagy szimulált jeleinket szeretnénk időtartomány helyett frekvenciatartományban szeretnénk megvizsgálni. Ehhez a numerikus analízisből ismeretes Diszkrét Fourier Transzformáció (DFT) nyújthat nekünk segítséget. Ezt az algoritmust az esetek túlnyomó többségében az ún. gyors Fourier-transzformációval valósítják meg. Az algoritmust az angol Fast Fourier



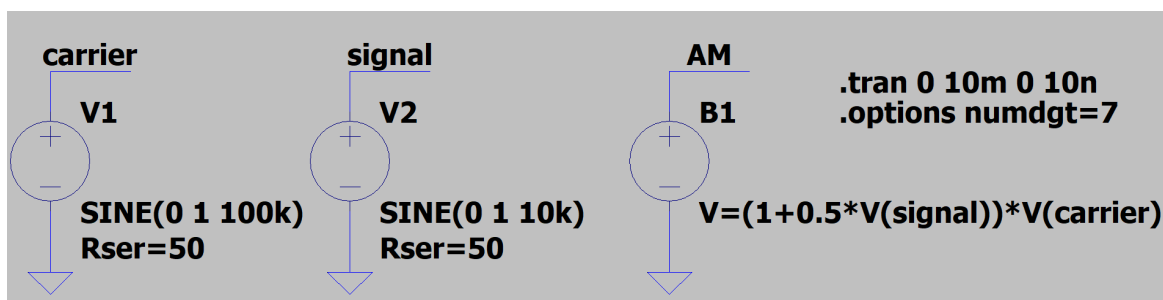
33. ábra. Kurzorok és jelmagyarázatok elhelyezése az ábrákon

Transform-nak, vagy röviden FFT-nek rövidítve használja, így ennek köszönhető, hogy a szaknyelvben a DFT és FFT fogalma összeforrt, és a mérnökök gyakorlatilag ugyanazt értik alattuk.

Az LTspiceban FFT analízis végzésére időtartománybeli szimulációt kell végezzünk a **.tran** direktíva felhasználásával. Ennek eredményeként előállnak a vizsgálható jeleink, pontosabban azok mintái, a szimuláció diszkrét időlépésenként számítja ki a jeleinket. Az FFT analízis elvégzéséhez a hullámforma megjelenítő ablakban kell a **(Jobb egérgomb > View > FFT)** opciót kiválasztanunk. Ekkor egy felugró ablakban meg kell adnunk az FFT-vel transzformálódó jeleinket. Ezen kívül ebben az ablakban tudjuk beállítani az FFT során felhasznált minták számát, mely alapértelmezetten a tranziens szimuláció összes időlépését tartalmazza. Ezután, ha több jelet transzformáltunk, úgy ki kell választanunk, hogy ezen jelek közül melyiket szeretnénk megjeleníteni. Ezután egy újabb felugró ablakban már láthatóvá is válik az ábrázolt jelek spektrális tartalma.

A folyamat illusztrálása érdekében figyeljük meg a 34. ábra kapcsolását. A kapcsolat a bemeneti 10 kHz-es bemeneti jelet egy 100 kHz-es vivővel szorozza össze, felhasználva a 3. fejezetben ismertetett viselkedési forrásokat. A szorzás eredményeként előáll egy két oldalsávú, nem elnyomott vivővel rendelkező, analóg amplitúdómodulált jel (AM-DSB/NSC). A sémán szereplő analóg szorzásból következő moduláció megértéséhez végig tudjuk követni a 10. egyenleteket, melyek az általános, m modulációs mélységgel rendelkező AM-DSB jel előállítását tükrözik. A trigonometrikus azonosságokat felhasználva szét tudjuk bontani a cos függvények szorzatát egy összeg és egy különbségi frekvenciakomponensre.

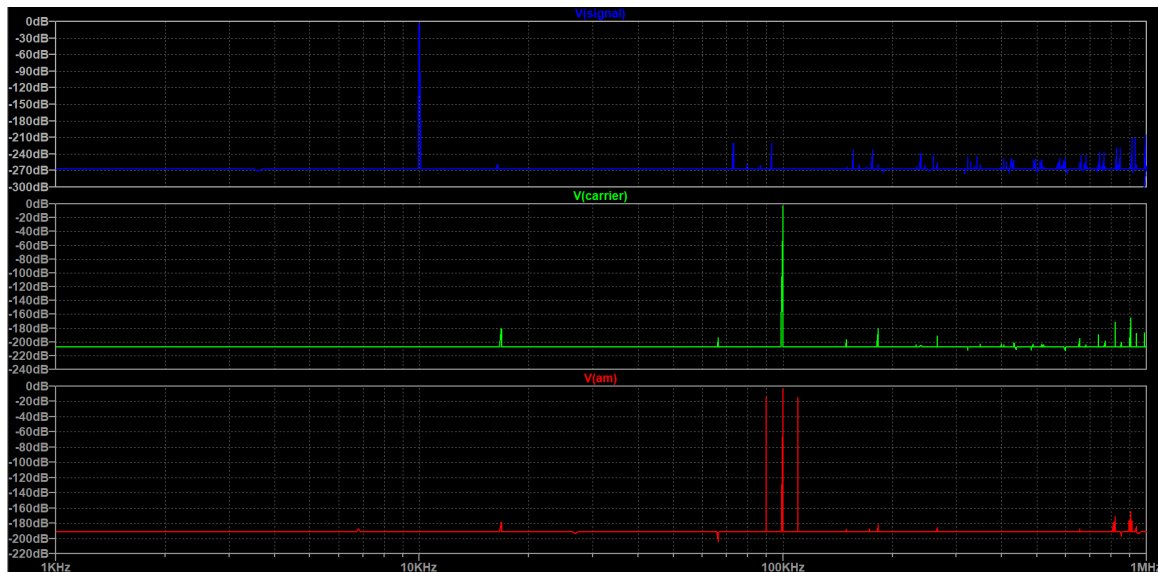
$$\begin{aligned}
 (1 + m \cos(\omega_s t)) \cos(\omega_c t) &= \cos(\omega_c t) + m \cos(\omega_s t) \cos(\omega_c t) \\
 &= \cos(\omega_c t) + \frac{m}{2} \cos((\omega_s + \omega_c) t) + \frac{m}{2} \cos((\omega_s - \omega_c) t)
 \end{aligned}
 \tag{10}$$



34. ábra. Amplitúdómodulációt megvalósító kapcsolás

Az modulációs eljárásra jellemző spektrumot megjelenítve az előbb említett FFT algoritmussal látható lesz a jellemző spektrális kép. A 35. ábrán láthatjuk szétszedve különböző ábrázolási síkokra a bemeneti 10 kHz-

es jelünket, a 100 kHz-es vivőjelünket és a kettő szorzatából előálló amplitúdómodulált jelünk kétoldalsávú spektrumát.



35. ábra. Amplitúdómodulált jel spektruma

Annak érdekében, hogy csökkentsük a numerikus számítás során a jeleinkre szuperponálódó nagyfrekvenciás zajokat, úgy érdemes a sémán elhelyezni egy **.options** direktívát melyben a szimuláció során alkalmazott pontosságot emeljük meg. Alapértelmezés szerint az LTspice a számítások során egyszeres pontossággal számol (single precision) vagyis a számok ábrázolására a C programozási nyelvből ismert **float** típust használja. A numdgt érték 6-nál nagyobbra állítása esetén a program áttér **double** típusú számábrázolásra.

```
.options numdgt=7
```

Ahhoz, hogy az ábrázolt spektrumban a spektrális csúcsok jól kivehetőek legyenek, és ne kenődjenek el a DFT algoritmus ablakozása miatt¹⁵, célszerű a szimulációs időt úgy megválasztani, hogy több periódust öleljen fel, ezzel csökkentve az ún. spektrális szivárgást az ábrázolt spektrumokban.

Utolsó optimalizációs lehetőségként még megemlítjük, hogy alapértelmezés szerint az LTspice az időtartománybeli szimuláció eredményeit csak tömörítve menti el, ennek következtében az FFT analízis során különböző torzítások léphetnek fel. Ez a tömörítés kikapcsolható a **(Tools > Settings > Compression > Enable 1st (2nd) Order Compression)** opciók kikapcsolásával. A 35. ábrát az előző bekezdésekben említett beállítások együttes használatával értük el. Ezen beállítások nélkül az olvasó a miénktől eltérő eredményt fog kapni.

6.3. Mérések elvégzése és a **.measure** direktíva

A fejezet elején megismerkedtünk a hullámformák vizualizációjával valamint a kurzorok használatával. Kurzorok segítségével a hullámformákról könnyedén tudtunk számértékeket leolvasni a független változó (vízszintes tengely) különböző értékei mellett. A gyakorlatban azonban előfordul, hogy a szimulációk végeztével csupán csak néhány specifikus pontban szeretnénk leolvasni a hullámformákban tárolt információt. Ilyen esetekben lehet hasznos számunkra a **.measure** vagy **.meas**¹⁶ direktíva. Ebben a szakaszban megismertetjük az olvasót az automatizált mérések néhány gyakrabban használt esetével.

¹⁵A DFT algoritmus használatát felfoghatjuk úgy, mint egy folytonos idejű jel és egy ablakfüggvény szorzatának Fourier-transzformáltját. Ebben az esetben az ablakfüggvényünk egy négyszögfüggvény, melynek szélessége megegyezik a szimulációs időnk hosszával. Mivel az időtartománybeli szorzás azonos a frekvenciatartománybeli konvolúcióval, így a spektrumunkban a négyszögablakra jellemző $\sin(x)/x$ -es spektrumot fogjuk látni a bementi jelünk frekvenciáival eltolva. Mivel a kezdeti jelünk mintavételezett volt, így a végleges DFT spektrumot a folytonos és ablakozott spektrumok mintáiból is elő tudjuk állítani.

¹⁶Az LTspice mind a két változatot elfogadja

A `measure` direktívával elvégzett méréseknek alapvetően két változata létezik, az egyik változat esetében a mérés egy adott pontra vonatkozik, míg a másik változat esetén pedig a mérés egy intervallum fölött értelmezett. Először a pontok felett értelmezett változatot tárgyaljuk. A direktíva általános szintaktikájának tárgyalása helyett, inkább néhány példán keresztül illusztráljuk az elvégzett méréseket.

```
.measure TRAN val1 FIND V(out) AT 5m
.measure TRAN val2 FIND V(out)*I(Vout) WHEN V(x)=3*V(y)
.measure TRAN val3 DERIV I(out) AT 2m
.measure TRAN val4 PARAM 3*val1/val2
```

A direktívákban szereplő **TRAN** kulcsszó megadja, hogy az adott mérés mely szimulációs fajta esetén értékelődjön ki. Lehetséges értékei a **AC**, **DC**, **OP**, **TRAN**, **TF** és **NOISE**. A **val1-val4** a mérési eredmények azonosítására szolgáló címkék lesznek, vagyis az első mérés eredményét a **val1** név alatt fogjuk megtalálni a későbbiekben. Ezt követően meg kell adni, hogy milyen mérést szeretnénk elvégezni. A lehetséges értékek a **FIND**, **DERIV** és **PARAM**. A **FIND** paranccsal egy meghatározott mérési pontban tudjuk leolvasni a vizsgált jel értékét. A mérési pontot meghatározhatjuk a független változó egy adott értékével, ekkor a **AT** parancsot használjuk, vagypedig egy adott feltétel teljesülése esetén, mely esetben a **WHEN** paranccsal adjuk meg a feltételt. Ennek megfelelően az első mérési eredmény az out csomópont feszültségét olvassa le a tranziens szimuláció 5. ms-ban. Val2 értékben pedig az out csomópont feszültsége lesz eltárolva, amikor az x csomópont feszültsége az y csomópont feszültségének háromszorosával egyezik meg.

A **DERIV** paranccsal a jel megváltozásának pillanatértékét olvashatjuk le. Tranziens szimuláció esetén ez egybeesik az idő szerinti deriválttal. A parancs használata mellett itt is kétféleképpen adhatjuk meg a mérési pont feltételét az **AT** és a **WHEN** kulcsszavakkal.

Az utolsó lehetőség a pontok felett értelmezett mérésekre a **PARAM** kulcsszó használata, mellyel a korábbi méréseink értékeit tudjuk továbbszámolni. Ez a fajta mérés lehetőséget ad, hogy összetettebb mennyiségeket is meghatározzunk. Ennek a mérésnek egy lehetséges felhasználási módja a kapcsolóüzemű tápegységek hatásfokának számítása, hiszen ekkor meg kell mérjünk a be- és kimeneti teljesítményeket, valamely mérési pontokban és ezek hányadosából tudjuk meghatározni az energiaátalakítás hatásfokát.

Az intervallumok felett értelmezett mérésekkel alapvetően más típusú mennyiségeket tudunk kinyerni a szimulációinkból, így a továbbiakban ezekkel foglalkozunk.

```
.measure TRAN Uavg AVG V(out)
```

A fenti direktívával meg tudjuk határozni az out csomópont feszültségének átlagértékét az **AVG** kulcsszóval, valamint a mérés eredményét eltároljuk az Uavg változóba. Az átlagon kívül lehetőségünk van még maximum, minimum, csúcstól-csúcsig érték, RMS középérték és határozott integrál számítására is. Ezekhez fel kell használjuk a **MAX**, **MIN**, **PP**, **RMS** és **INTEG** kulcsszavakat. Hasonlóan a pontok felett értelmezett mérésekhez, itt is lehetőségünk van extra feltételek megadására, melyekkel a méréseinket tovább specifikálhatjuk. Egy ilyen specifikus mérésre láthatunk egy példát a következő direktívánál. A direktíva az out csomópont feszültség idő szerinti integrálját számolja ki. Az integrálás alsó határaként az az időpont van meghatározva, ahol az in csomópont a 2 V-os értéket másodjára éri el egy felfutó éllel. Ha lefutó élre szeretnénk indítani az integrálást, úgy a **RISE** helyett a **FALL** parancsot kell használjuk. Az integrálás felső határát a **TARG** paranccsal tudjuk megadni, ahol példánkban az in csomópont 6 V-os küszöbérték 5. kereszteződéséig számolunk. Ekkor nem specifikáltuk, hogy a küszöbértéket felfutó vagy lefutó ágon érzük el.

```
.measure TRAN Uint INTEG V(out) TRIG V(in)=2 RISE=2
TARG V(in)=6 CROSS=5
```

A mérési direktívák szerkesztésének és a szintaktikák tanulásának könnyítése végett lehetőségünk van a parancsok szerkesztésére egy grafikus felületen keresztül is. Ehhez el kell készítenünk egy kezdetleges direktívát, pl. a `.meas` direktíva, amit ezután elhelyezünk a sémán. A már lehelyezett direktívára jobb egérgombbal

kattintva megnyílik a grafikus szerkesztőfelület, ahol könnyebben tudjuk megadni méréseink specifikációját. A grafikus szerkesztőfelület a 36. ábrán látható. A grafikus szerkesztő nagy előnye emellett, hogy lehetőségünk van a szimulációnk lefuttatása után letesztelni a mérési direktívánk eredményét a bal alsó sarokban található **TEST** gomb segítségével!

Előfordulhat, hogy az automatizált méréseink nem adják vissza a kívánt eredményt. Ebben az esetben egy lehetséges megoldás lehet a szimulált pontok számának növelése pl. tranziens szimuláció esetén a maximális időlépés csökkentésével. Ezzel megnövelve az esélyt, hogy a mérések kiértékelésekor a szimulált adatpontok beleesnek a direktíva toleranciasávjába.

36. ábra. Mérési direktívák grafikus szerkesztőfelülete

Áramkörü paraméterek változtatása esetén szükséges lehet a paraméterek hatását vizsgálni a mérési eredményeken. A korábbiakban csak a hullámformákból egyszerűen kinyerhető információk ábrázolásával foglalkoztunk, azonban az LTspice-ban lehetőségünk van a mérési eredmények grafikus megjelenítésére is. Ehhez a szimulációk futtatása után meg kell nyitnunk a kimeneti log fájlt (**View > SPICE Output Log**) vagy a **CTRL + L** gyorsgomb használatával. A log fájlban jobb egérgombbal klikkelve és kiválasztva a **Plot .step'ed .meas data** opciót egy új hullámformamegjelenítő ablak ugrik fel. Ebben az ablakban jobb egérgomb **Add Traces** vagy a **A** gyorsgombot lenyomva kiválaszthatjuk, hogy a mérési eredményeink közül melyiket szeretnénk megjeleníteni az ábrán.

Az így előálló másodlagos hullámforma megjelenítő ablakban a fejezet elején leírtakkal azonosan tudjuk módosítani az ábrázolt jeleinket. Kiterjed ez mind a tengelyek módosítására, a független változó megváltoztatására és a jelek közötti matematikai műveletekre is. Az egyetlen kitétel, hogy a másodlagos hullámforma megjelenítőben csak és kizárólag a mérési eredményeink jelennek meg mint jelek, az elsődleges jeleinket (áramok és feszültségek) nem tudjuk felhasználni.

7. Modellfájlok felhasználása

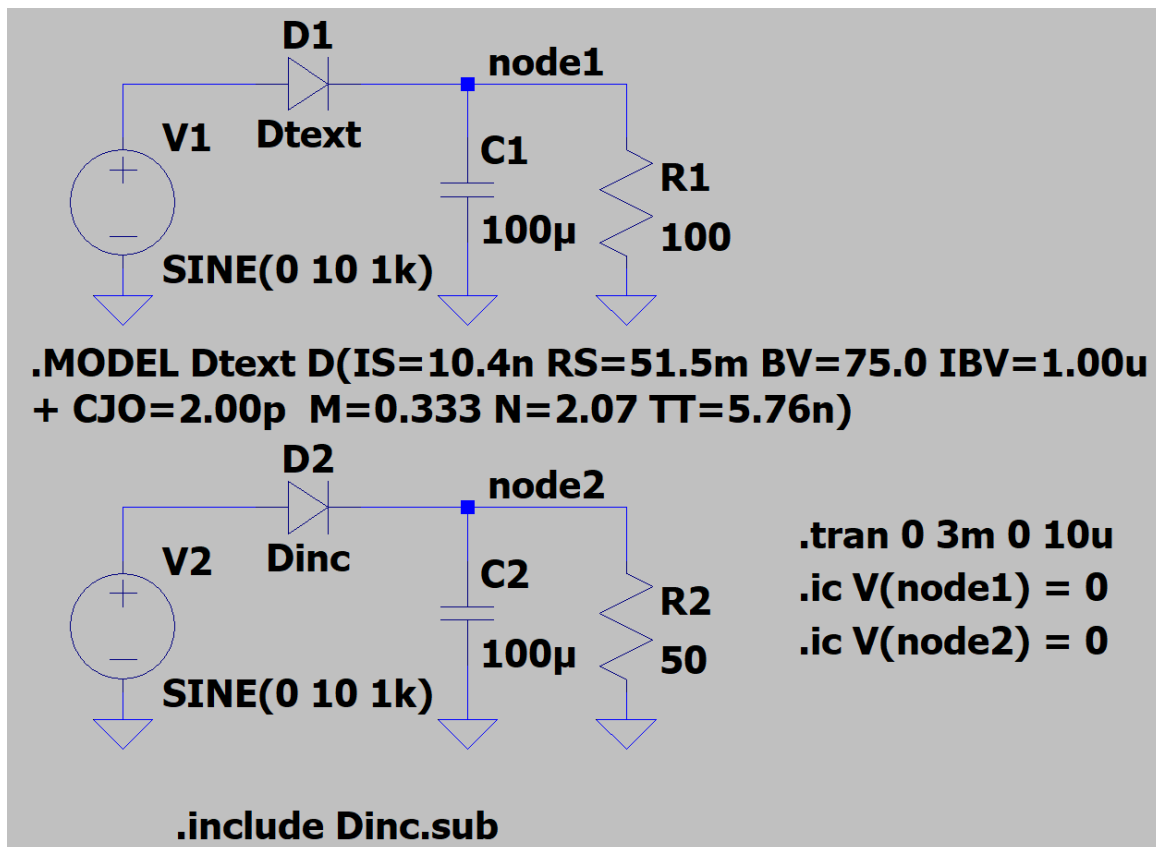
Szimulációs modellezés során előfordul, hogy a modellezni kívánt alkatrész modellparamétereit nem tudjuk saját magunk megadni (esetleges mérések alapján), vagy nem tudunk az általunk használt szoftver könyvtárban alkalmas modellfájlt találni. Ekkor rá vagyunk kényszerülve, hogy ún. harmadik féltől származó modelleket

használjunk. Ezeket a modelleket egy külső forrásból kell beszerezzük, vagyis meg kell őket vásárolnunk, vagy ingyenes és szabad hozzáférésű modelleket kell használunk.

Ebben a fejezetben bemutatjuk, hogyan adhatunk hozzá modellfájlokat az általunk készített modellekhez és hogyan tudjuk felhasználni az általunk készített kapcsolásokat alkatrészek és modulok modellezésére.

7.1. Modellfájlok importálása

Az LTspice lehetőséget biztosít harmadik féltől származó modellek felhasználására és szimulálására. Ehhez leggyorsabb esetben csak a szükséges modellfájlt kell hozzáadunk a kapcsolási sémához. Ezt kétféleképpen tudjuk megtenni. Egyrészt bemásolhatjuk a sémára modellfájl teljes tartalmát, vagy egy direktíva hozzáadásával. Az első esetben a sémán elérhetővé válik a modellezet komponens leírása (általában egy netlista formájában), így a sémát olvasó mérnök mélyebb betekintést nyerhet a kapcsolat működésébe. Abban az esetben, ha a modellfájl számunkra nem képvisel értékes hozzáadott információt, úgy célszerű a modellfájlt egy külső fájlból hozzáadni a sémánkhoz. Erre szolgál az **.include** direktíva, mellyel a megadott szöveges fájl tartalma teljes egészében bemásolódik a szimulált séma netlistájához a szimuláció megkezdése előtt. Ezekre a lehetőségekre látunk példákat a 37. ábrán.



37. ábra. Külső modellek megadásának két lehetséges módja

a 37. ábrán egy egyutas egyenirányító kapcsolás két példányra figyelhető meg. A felső kapcsolásban az egyenirányító dióda típusa (modellje) **Dttext**. Ennek a diódatípusnak a működését egy **SPICE Diódamodel** írja le. A modell pontos működésének megadásához rengeteg paraméter numerikus értékét meg kell adnunk. Ezen paraméterek a kapcsolat alatt látható **.model** direktíva meghívásával vannak specifikálva. A direktíva után megadjuk az általunk modellezett komponens nevét (ebben az esetben a Dttext), mely egy spice diódamodel és specifikáljuk az alapértelmezett dióda paramétereitől eltérő értékeket egy kulcs-érték pár szintaktikával.

A diódamodel alternatív megadási módja az **.include** direktíva használata. Ebben az esetben meg kell adnunk a szöveges fájl nevét (ebben az esetben Dinc.sub) és el kell helyeznünk a szükséges fájl a sémával

azonos könyvtárba (vagy az LTspice alapértelmezetten keresett könyvtárainak valamelyikébe). A `.sub` fájl lényegi tartalma ezen szimuláció során teljesen azonos a `.model` direktívánál használtakkal.

Mind a két esetben a szimulációk gond nélkül lefutnak és szépen mutatják a terhelt egyutas egyenirányító jellegzetes hullámformáit, azonban a sémán gyökeresen eltérő módon jelennek meg. A fenti kapcsolat mélyebb betekintést enged a sémát néző mérnök számára a szimulációs modell beállításaiába és az azokból fakadó hibalehetőségekbe, míg az utóbbi egy hierarchikusabb módon kiemeli a lényegi részt és elfedi a kevésbé lényegeseket.

Fontos megjegyezni, hogy bizonyos alkatrészgyártó cégek az általuk gyártott alkatrészek SPICE modelljeit csak titkosított formában teszik publikusan elérhetővé. Ez a leírás mód többnyire gépi kódból álló szöveges fájl formájában jelenik meg. Ennek következtében a titkosított alkatrészmodellek esetében az `.include` direktíva használata javasolt, hiszen semmilyen többletinformációt nem ad hozzá a modellfájl a kapcsolási sémánkhoz!

7.2. Eszközkönyvtárak importálása

Egyes alkatrészgyártó cégek a szimulációs modellfájlokat csoportosítják a modellezett komponens szerint ún. alkatrészkönyvtárakba. Ezekben az alkatrészkönyvtárakban többnyire azonos célú vagy viselkedésű alkatrészek SPICE modelljeit adják meg egy aláramkör (`.subckt` direktívát felhasználva) formájában, majd ennek az aláramkörnek a különböző paraméterezéssel ellátott példányait sorolják fel, szintén aláramköri modellek felhasználásával. A módszer előnye, hogy az alkatrészek viselkedését modellező áramköri kapcsolást csak egy példányban kell összeállítani és a különböző alkatrészek modelljei ebből leszármaztathatók a modell paramétereinek módosításával.

Alkatrészkönyvtárak importálásához lényegében az `.include` direktívával azonos módon kell eljárunk. A sémán el kell helyeznünk egy `.lib` direktíva utasítást mely után megadjuk az importálni kívánt alkatrészkönyvtár fájlnevét, kiterjesztéssel együtt! Az importált fájl elérési útjához fűződő gondolatok azonosak az `include` direktívánál ismertekkel.

7.3. Eszközkönyvtárak készítése

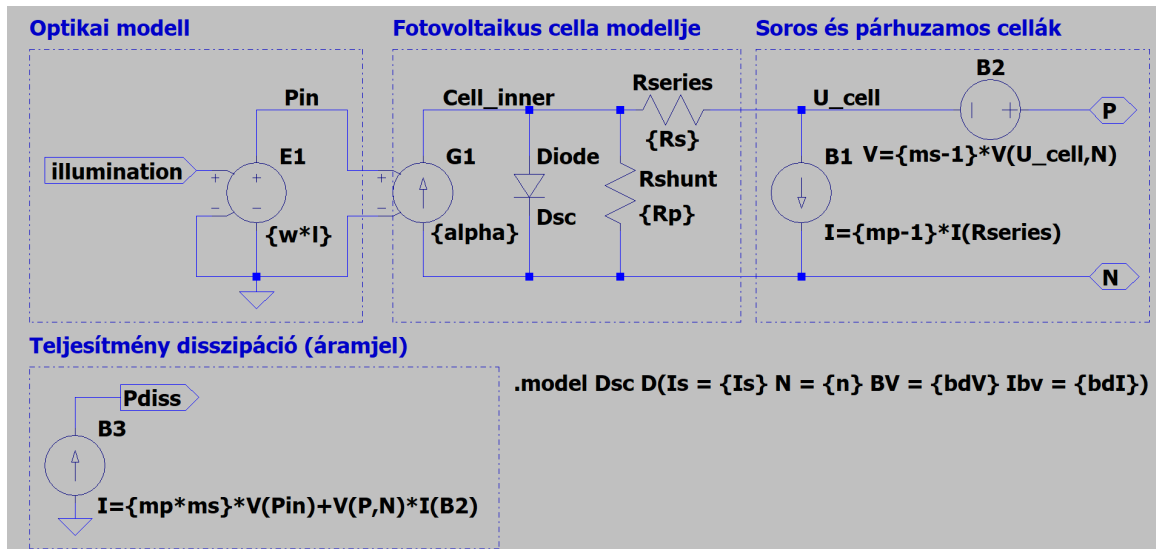
Ebben a szakaszban bemutatjuk, hogyan készíthetünk saját alkatrészkönyvtárat egy áramköri komponens modellezésére. A folyamatot egy összetettebb kapcsolat példáján vezetjük végig, azonban a kapcsolat működésére és a modellezett jelenségek leírására nem térünk ki¹⁷.

A modellezés első lépéseként a paraméterekkel ellátott áramköri modellt kell megalkossuk melyből az alkatrészkönyvtárat a későbbiekben származtatjuk majd. Ennek az áramköri leírásnak a megadása történhet rögtön a netlistafájl megadásával, azonban annak átláthatósága és az esetleges hibák elkövetésének veszélye miatt, csak indokolt esetben javasoljuk az olvasók számára a használatát! Az áramköri leírás megadásának egyszerűbb és nyomonkövethetőbb formája a grafikus sémakeresztő használata. A sémakeresztőn a 3. fejezetben ismertetett módon meg tudjuk rajzolni az áramköri modellünket. A példánk során a 38. ábra kapcsolását fogjuk felhasználni.

A séma megrajzolása után ki kell nyernünk a netlista szerinti leírást. Ehhez a **(View > SPICE Netlist)** opciót kell kiválasztanunk. Ekkor egy felugró ablakban megjelenik a sémával ekvivalens netlista alapú áramköri leírásunk, ahogy azt a 39. ábrán láthatjuk is. Ebben a leírásban az egyes áramköri részek tematikusan rendezetlen formában lettek rögzítve, így célszerű lehet azokat átcsoportosítani és kommentekkel ellátni, ahogy az a 40. ábrán is látható. Mint már említettük a modellezett komponenseket ún. aláramkörök formájában tudjuk hozzáadni az LTspice-hoz. Ehhez nyissunk meg egy új szöveges fájlt és adjuk hozzá a 40. ábra tartalmát.

A `.subckt` direktíva után az aláramkörünk nevét kell megadjuk (SolarCell) majd a külső áramköri összeköttetések neveit. Fontos megjegyezni, hogy netlisták alkalmazásánál az aláramkör elektromos összeköttetések nem a vezetékek nevei alapján történnek (ahogy az a hierarchikus kapcsolások esetében történt), hanem a netlistán szereplő csomópontok sorrendje alapján. Így példánkban az aláramkör példányosításánál a első vezeték címke lesz összekötve a 38. ábra kapcsolásának P kivezetésével és így tovább!

¹⁷A téma iránt érdeklődő hallgató számára javasoljuk a **Zöld villamosenergia-termelés** című tárgyat a további részletek megismerése érdekében!



38. ábra. Összetett fotovoltaikus cella áramköri modellje

```

Z:\Complex_pv_model.asc
Diode Cell_inner N Dsc
Rshunt Cell_inner N {Rp}
Rseries U_cell Cell_inner {Rs}
E1 Pin 0 illumination 0 {w*1}
G1 N Cell_inner Pin 0 {alpha}
B1 U_cell N I={mp-1}*I(Rseries)
B2 P U_cell V={ms-1}*V(U_cell,N)
B3 0 Pdis I={mp*ms}*V(Pin)+V(P,N)*I(B2)
.model D D
.lib C:\Users\David\AppData\Local\LTspice\lib\cmp\standard.dio
.model Dsc D(Is = {Is} N = {n} BV = {bdV} Ibv = {bdI})
* Optikai modell
* Fotovoltaikus cella modellje
* Soros és párhuzamos cellák
* Teljesítmény disszipáció (áramjel)
.backanno
.end
    
```

```

* Solar cell model for SPICE simulations
* Developed by David Deak for educational use only, commercial use is not authorized
* Developed for practice course in Green Electricity Production (BMEVIEEM11)
* January 2024

.subckt cell1 1 2 3 4
X1 1 2 3 4 SolarCell w={w} l={l} ms={ms} mp={mp} alpha=0.3 Is=1e-9 n=1 bdV=10 bdI=1 Rs=1m Rp=1k
.ends

.subckt SolarCell P N Illumination Pdis
* optical model
E1 Pin 0 illumination 0 {w*1}
* solar cell one diode model
G1 N Cell_inner Pin 0 {alpha}
D1 Cell_inner N Dsc
.model Dsc D(Is = {Is} N = {n} BV = {bdV} Ibv = {bdI})
Rshunt Cell_inner N {Rp}
Rseries U_cell Cell_inner {Rs}
* series and parallel multipliers
B1 U_cell N I={mp-1}*I(Rseries)
B2 P U_cell V={ms-1}*V(U_cell,N)
* power dissipation
B3 0 Pdis I={mp*ms}*V(Pin)+V(P,N)*I(B2)
.ends
    
```

39. ábra. Aláramkör netlista reprezentációja

40. ábra. Alkatrészkönyvtár netlistája

.subckt SolarCell P N Illumination Pdis

Ezután megadjuk az aláramköri részben szereplő áramköri komponenseket és az azokat összekötő vezetéseket. Ezt célszerű a grafikus felületen megszerkesztett séma netlistájából egyszerűen kimásolni egy tematikus sorrendben. A tematizálást segíthetjük kommentek házzáadásával is! A netlista fájlban minden sor ami a * karakterrel kezdődik kommentként lesz értelmezve. Az áramköri leírás végét az **ends** direktíva jelzi.

Ezek után hozzá tudjuk adni az alkatrészkönyvtárunkhoz a példányosított és paramétereikkel ellátott konkrét szimulációs modelljeinket. Ehhez egy új sorban létre kell hoznunk egy egyedi névvel rendelkező aláramkört (.subckt direktíva), példányosítanunk kell benne a generikus áramköri leírásunkat és meg kell adjuk a paramétereit. A + jelet felhasználva az előző sor tartalmát egészíthetjük ki, mintha csak folytattuk volna azt. Ennek használata lehetővé teszi a hosszú sorok elkerülését, valamint lehetővé teszi a netlista fájlok tartalmának több sorba rendezését. Netlista fájlok esetében minden áramköri komponens egy-egy sorban foglalhat csak helyet + jel használata nélkül.

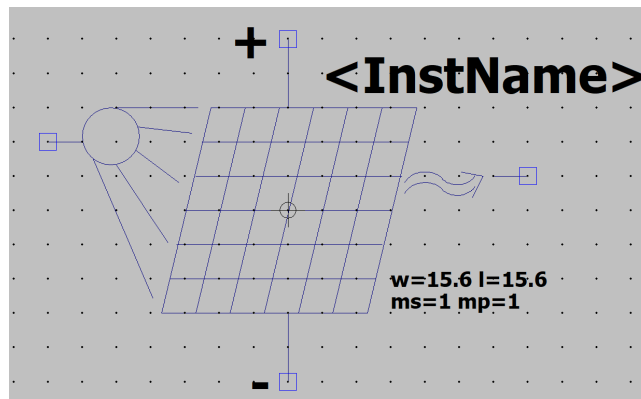
```

.subckt cell1 1 2 3 4
X1 1 2 3 4 SolarCell w={w} l={l} ms={ms} mp={mp} alpha=0.3
+ Is=1e-9 n=1 bdV=10 bdI=1 Rs=1m Rp=1k
.ends
    
```

Ebben az esetben a specifikus komponensünk a **cell1** nevet kapta és a kivezetései az 1, 2, 3, 4 címkeket kapták. Ezen áramköri részen belül létrehozzuk az aláramkörünk egy példányát. A komponens egyedi azo-

nosítója az X1-es, melyben a **X** prefixum jelzi, hogy netlistával modellezett aláraköréről van szó és nem egy hierarchikus blokkról. Ezután meg kell adjuk az X1-es komponens leírását (ez a generikus SolarCell modell) és a paramétereket egymás mellé felsorolva, kulcs-érték párok szintaktika megadásával. Alternatív módon ezeket megadhatnánk a cell1 netlistájában a **.param** direktívát használva is. A w, l, ms és mp paraméterek esetében alkalmaznunk kell a -t, hiszen ezeket a paramétereket a szimbólum fogja szolgáltatni nekünk egy hierarchikus réteggel feljebb!

Ezzel el is készítettük az alkatrészkönyvtárunkat és hozzá is adtunk egy paraméterekkel specifikált komponenst is, azonban jelenlegi állapotában az alkatrészkönyvtár csak netlista formájában adható hozzá sémákhoz. A könnyebb kezelhetőség érdekében készíthetünk hozzá egy szimbólumot, amit sémákon lehelyezve elérjük az alkatrészkönyvtárat. Új szimbólum hozzáadásához nyissunk meg a szimbólumszerkesztőt (**File > New Symbol**) és rajzoljuk meg alárakörünk szimbólumát a 41. ábra szerint.



41. ábra. Fotovoltaikus cellát reprezentáló szimbólum

Az áramköri **pin**-ek megadásánál ügyeljünk azok sorrendjére, hiszen az áramköri összeköttetések ez a sorrend alapján kerül megvalósításra! Ennek megfelelően a kivezetéseket állítsuk be a **.subckt** direktívának megfelelő sorrendbe! Ezek után hozzá tudunk adni a szimbólumunkhoz ún. attribútumokat, melyek a szimbólum paraméterezését segítik és a sémán is láthatóvá tehetők. Ezzel lehetőségünk van a fontosabb paramétereket a séma szerkesztője számára is elérhető tegyük. Az attribútumok beállításához a (**Jobb egérgomb > Attributes > Edit Attributes**) vagy a **CTRL + A** gyorsgombot kell megnyomjunk.

Az attribútumok közül mindenképpen át kell állítsuk a szimbólum típusát **Cell**-re, ezzel jelezve az LTspicenak, hogy ne keressen a szimbólumhoz tartozó hierarchikus blokkot, csak egy netlistát. A következőkben beállíthatjuk a sémáról elérhető paramétereinket. A paramétereket melyeket a **Value**, **Value2**, **SpiceLine** vagy **SpiceLine2** mezőkbe írunk mind meg tudjuk jeleníteni a sémán, valamint módosítani is tudjuk őket onnan! A paramétereket tetszőlegesen elhelyezhetjük ezekben a mezőkben, hiszen a sorrendjük csak azt befolyásolja, hogy a szimulációs netlistába milyen sorrendbe fognak bekerülni. Ennek megfelelően a Value és Value2 mezők tartalmát feltölthetjük a 41. ábrának megfelelően. A **Description** mezőben megadhatjuk a szimbólum által reprezentált kapcsolás rövid leírását. A legutolsó fontos paraméter a **ModelFile** sor. Ezen soron tudjuk megadni a szimbólum által reprezentált alárakört leíró könyvtárat. Ennek megfelelően állítsuk be ezt a sort **solar_cell.lib**-re. Ennek hatására a szimbólum sémára való lehelyezésével meghatározzuk az alkatrészkönyvtárat tartalmazó fájlnevet is, így nem lesz szükséges további **.include** direktívák elhelyezése a séma felületén! A beállított attribútumok megjelenítéséhez a (**Jobb egérgomb > Attributes > Attribute window**) vagy a **CTRL + W** gyorsgombot kell megnyomjunk. Ekkor egy felugró ablakban ki tudjuk választani a sémán elhelyezendő attribútumunkat és azt leokézva le tudjuk helyezni azt a sémára.

A szimbólumon elhelyezett attribútumok között megtalálható a w, l, ms és mp változók. Ezeket a változókat megjelenítjük a sémán, így az alárakör példányosítása után is könnyen szerkeszthetők lesznek. Azonban a végső netlista összeállításakor ezeket a változókat tovább kell küldeni a cell1 példányon belül példányosított SolarCell áramkörnek. Ezt a célt szolgálják a $w = w$ és hasonló sorok. Vagyis a SolarCell alárakör w paramé-

terének adjuk a szimbólum w paraméterének értékét! Ebben az esetben a változók azonos neve megnehezíti az értelmezést, de a dolog lényegi része csak és kizárólag numerikus értékek továbbküldése az alacsonyabb szintű alárámköröknek!

Az így elkészített alkatrészkönyvtárat és szimbólumot már csak hozzá kell adnunk az LTspice megfelelő mappáihoz, hogy az LTspice-t elindítva hozzájuk is férjünk. Ehhez másoljuk át a **solar_cell.lib** fájlnkat a **C:\Users\USERNAME\AppData\Local\LTspice\lib\sub** mappába. Fontos, hogy ne a **C:\Program Files\ADI\LTspice** mappába rakjuk! Az elkészített szimbólum **.asy** fájlját pedig másoljuk át a **C:\Users\USERNAME\AppData\Local\LTspice\lib\sym** mappába, vagy ebben a mappában létre is hozhatunk egy új mappát a tematikus rendezés érdekében. Ha almappába másoljuk a szimbólum fájlját úgy a sémához való hozzáadáskor is az azonos nevű almappában kell majd keressük a szimbólumunkat.

Az LTspice újraindítása után az így elkészített alkatrészkönyvtárunkat már el fogjuk tudni érni a hozzáadott szimbólumunkon keresztül és példányosítani is fogjuk tudni a sémakeresztőben. Egyetlen fontos paramétert azonban nem állítottunk eddig még be. Eddig a pontig csak az alkatrészkönyvtárat specifikáltuk (a ModelFile sor segítségével) azonban nem mondtuk meg az LTspice-nak, hogy ebből a könyvtárból melyik alkatrészt szeretnénk használni. A pontos alkatrész kiválasztásához a példányosított szimbólum **SPICE Model** sorába be kell írunk a használni kívánt alkatrészünk nevét, mely esetünkben **cell1!** Ezáltal szimulálhatóvá vált az alkatrészünk számos absztrakciós rétegen keresztül!

8. Összefoglalás

Ezzel végére értünk az LTspice alapvető szintű működésének és az általa nyújtott szimulációs és modellezési lehetőségeinek bemutatásának. Ugyan a segédlet nem tért ki a szoftver minden részletére, de jelen segédlet elkészítésekor ügyeltünk a lehetőségek széleskörű bemutatására és az egyes szimulációk illusztrálására példakapcsolásokon keresztül. Reméljük jelen segédlet hiánypótló jelleggel elősegíti a magyarországi egyetemi hallgatók és szimulációs mérnökök eligazodását az LTspice mélyebb világában és teszi ezt magyar nyelven! A segédletben illusztrált példakapcsolások és egyéb segédfájlok megtalálhatók jelen PDF dokumentumhoz csatolva, ezzel is megkönnyítve azok elérését és a szükséges fájlok csomagként történő kezelését! A program mélyebb megértéséhez javasoljuk az Olvasónak a súgófájl használatát, melyet az **F1** gyorsgombbal tud előhívni!

Jelen segédlet csak egy előzetes és nem végleges változat, emiatt kérjük az olvasókat az esetleges elírások és hibák jelzésére a [szerzőknél](#).

A. Használt gyorsgombok listája

Az LTspice könnyebb és gyorsabb használhatósága érdekében célszerű röviden áttekinteni az LTspice alapértelmezett és általunk használt gyorsgombjait. Az egyes gyorsgombok az LTspice-on belül a különböző szerkesztőkörnyezetekben eltérőek lehetnek, hiszen a különböző szerkesztőkörnyezetekben más és más funkcionalitást használunk gyakrabban. A program 4 szerkesztőkörnyezetet különböztet meg, ezek a **séma**, **szimbólum**, **hullámforma** és **netlista**. Ezekben a környezetekben definiált gyorsgombokat a programon belül szerkeszteni nem lehet, azonban le lehet tölteni őket egy szöveges fájlba, azon belül módosítani, majd ezt a szöveges fájlt vissza lehet tölteni az LTspice-ba. A meglévő gyorsgombok a **(Tools > Settings > Drafting options > Keyboard shortcuts)** menüből érhetők el illetve itt van lehetőségünk új gyorsgombok hozzáadására egy szöveges fájlból. A következő táblázatban összefoglaljuk a fontosabb gyorsgombokat, fókuszálva a séma szerkesztőre és a hullámforma megjelenítőre.

Séma szerkesztő		Hullámforma megjelenítő	
Új komponens hozzáadása	P	Függőleges igazítás	CTRL+Y
Vezeték rajzolása	W	Rács megjelenítése	CTRL+G
Vezetékcímke hozzáadása	N	Hullámforma hozzáadása	A
Törlés	DEL	Törlés	DEL
Másolás ¹⁸	CTRL+C	Ábrázolási sík hozzáadása felül	P
Mozgatás ¹⁹	M	Ábrázolási sík hozzáadása alul	B
Húzás ²⁰	S	Kurzorcímke hozzáadása	L
SPICE direktíva	.	Szöveg hozzáadása	T
Föld szimbólum	G	Vonal rajzolása	I
Ellenállás	R	Ábrajelölők másolása	CTRL+C
Kapacitás	C	Ábrajelölők mozgatása	M
Induktivitás	L	Ábrajelölők húzása	S
Dióda	D		
Feszültségforrás	V		
Forgatás	CTRL+R		
Tükrözés	CTRL+E		
Szimuláció indítása	SHIFT+R		
Szimuláció leállítása	SHIFT+H		
SPECE log fájl megnyitás	CTRL+L		
Egész séma látótérbe helyezése	SPACE		

3. táblázat. Gyakran használt gyorsgombok listája

¹⁸A kijelölt áramkört rész megkettőzi

¹⁹A kijelölt áramkört rész mozgatja, függetlenül az összeköttetésektől

²⁰A kijelölt áramkört rész húzza, megtartva az összeköttetéseket

B. PDF-he csatolt fájlok listája

Fájlnév	Fájl leírása
UbuntuInstall.sh	Wine telepítő script Ubuntu 22.04-hez
shortcuts.txt	Importálható gyorsgombok beállítási fájlja
audio_preamp_example.asc	Audió előerősítő tesztkapcsolása
audio_preamp.asc	Audió előerősítő belső kapcsolási rajza
audio_preamp.asy	Audió előerősítő szimbóluma
bjt.asc	BJT karakterisztika meghatározásához használt kapcsolás
paramExample.asc	Parametrikus kaszkád feszültségosztó
mcExample.asc	Monte Carlo szimulációs példakapcsolás
noise.asc	Zajszimuláció kapcsolási rajza
voltage_divider.asc	Feszültségosztó kapcsolási rajza
efgh_sources.asc	Vezérelt források
Bsources.asc	Tetszőleges viselkedésű források
antennaImpedance.sub	Antenna impedanciamenetét leíró modell
ArbitraryImpedance.asy	Tetszőleges impedanciamenetet modellező szimbólum
netExample.asc	Tesztkapcsolás kétpólus és kétkapu paraméterek számításához.
Dinc.sub	Diódamodel include direktíva használatához
include.asc	Modellfájlok importálását bemutató séma
Complex_pv_model.asc	Összetett fotovoltaikus modell
solar_cell.lib	Fotovoltaikus cellák könyvtára
Solar_cell.asy	Fotovoltaikus cella szimbóluma

4. táblázat. Csatolt fájlok listája és leírása