



Budapesti Műszaki és Gazdaságtudományi Egyetem
Közlekedés- és Járműmérnöki Kar
Közlekedés- és Járműirányítási Tanszék

Forgalomfüggő zöldidő-összehangolási algoritmus fejlesztése

Gráf Tamás

2013

Tartalomjegyzék

1. A feladat bemutatása.....	3
2. A jelenleg alkalmazott zöldidő-összehangolási algoritmusok bemutatása.....	5
2.1. Osztóponti módszer.....	5
2.2. MAXBAND, MULTIBAND módszerek.....	8
2.3. TRANSYT, PASSER és ALLONS-D módszerek.....	10
3. A vizsgált útszakasz és a jelenlegi forgalmi állapotok bemutatása.....	13
3.1. A vizsgált útszakasz bemutatása.....	13
3.2. Az elvégzett forgalomszámlálás részletezése.....	17
3.3. Jelenlegi forgalmi állapotok bemutatása.....	21
4. Zöldidő-összehangolás tervezése különböző periódusidőkre.....	26
4.1. LISA+ szoftver bemutatása.....	26
4.2. Jelzéstervek elkészítése.....	34
4.3. Azonos ciklusidejű jelzéstervek összehangolása.....	40
5. Forgalomfüggő irányítás elkészítése.....	45
5.1. Kétszintű forgalomfüggő modell bemutatása.....	45
5.2. Forgalomfüggő irányítás 1. szint - igények detektoros vizsgálata.....	47
5.3. Forgalomfüggő irányítás 2. szint - dinamikus periódusidő változtatás.....	51
6. Az elkészített irányítás értékelése, összehasonlítása a jelenlegi állapottal.....	57
6.1. VISSIM szoftver bemutatása.....	57
6.2. Kapcsolat a LISA+ és a VISSIM szoftverek között.....	59
6.3. A tervezett irányítás értékelése.....	61
7. Az algoritmus továbbfejlesztésének lehetőségei.....	68
8. Összefoglalás.....	70
Köszönetnyilvánítás.....	71
Irodalomjegyzék.....	72
Ábrajegyzék.....	73
Táblázatjegyzék.....	75
Mellékletek.....	76

1. A feladat bemutatása

Szakdolgozatom választott témája egy forgalomfüggő zöldidő-összehangolási algoritmus fejlesztése és tesztelése. A témaválasztás során tekintettel voltam arra, hogy a témakör gyakorlati jellegű legyen, a tesztelésre használt szakasz kiválasztásánál pedig személyes indokok is vezéreltek. Utazásaim során - a budapesti városközpont és a lakóhelyem (Budafok) között - sokszor tapasztaltam, hogy a használt útvonal körülményei nem a legmegfelelőbbek. Különösen problémás azonban a helyzet a Budafoki úti szakaszon, ahol személygépjárművel haladva egyes esetekben rendkívül sűrűn meg kell állni. Ezzel szemben a párhuzamos Szerémi úton megállás nélkül lehet végighaladni, hasonló körülmények között. Innen jött az ötlet, hogy miért ne lehetne akkor ezt az állapotot megvalósítani a Budafoki úton is.

A dolgozat alapvető témája tehát már adott volt, mint a zöldidő-összehangolás tervezése. Céлом volt továbbá, hogy a tervezésemet valós adatokkal támasszam alá, ezért egy forgalomszámlálás elvégzését is beterveztem a diplomakészítés programjába. Kapacitásbeli hiányosságok miatt erre csak egy hatvan perces intervallumban került sor, de ez is jó alapot adott a tervezéshez.

Szükség volt még valamire, amitől a szakdolgozatom valamivel több lesz, mint egy átlagos összehangolást magában foglaló tervezés. Ehhez onnan jött az ötlet, hogy az érintett szakaszon bizonyos csomópontokban már telepítve voltak mágneses hurokdetektorok. Ezt továbbgondolva jutottam el ahhoz a feltevéshez, hogy akár a zöldidő-összehangolást is lehetne forgalomfüggővé tenni. Ez első ötletként annyit jelentett volna, hogy amennyiben nincs igény a mellékirányban vagy a főirányból balra kanyarodó sávokban elhelyezett detektorokon, akkor a hangolási sáv kiszélesedhetne, azaz több autó tudna átjutni a szakaszon megállás nélkül. Ehhez a hangoláson kívül szükség van új detektorok elhelyezésére is, valamint el kell készíteni azt a logikát is, ami alapján a forgalomfüggő program működik. Ennek a lényegét az adja, hogy gyakorlatilag a főirány folyamatos zöldet kap, ez csak akkor szakad meg, ha bármely más irányból igény jelentkezik.

Ezt a kombinációt még mindig nem találtam megfelelő szakdolgozati témának, hiszen igazából két független dolgot kívántam megvalósítani egy adott szakaszon. Ennek a helyzetnek a megoldását jelentette a forgalomfüggőség kiterjesztése kétszintűre. Ez a kétszintűség azt jelenti, hogy a programok nem csak az előbb említett igényektől függenek, hanem a főirány forgalomnagyságától is. Attól függően, hogy mekkora a forgalom magán a Budafoki úton, egy központi rendszer választja ki az adott forgalomnagysághoz optimális ciklusidejű programokat az előre kialakított változatok közül. Ennek a módszernek a segítségével elmondható, hogy létrejött egy olyan forgalomfüggő zöldidő-összehangolási módszer, amely hatékony irányítást képes megvalósítani arteriális hálózatokon, mégis rendkívül egyszerű, hiszen a főirány forgalomnagyságának mérésére sávonként csak 1-1 hurokdetektort kell telepíteni a szakasz elejére.

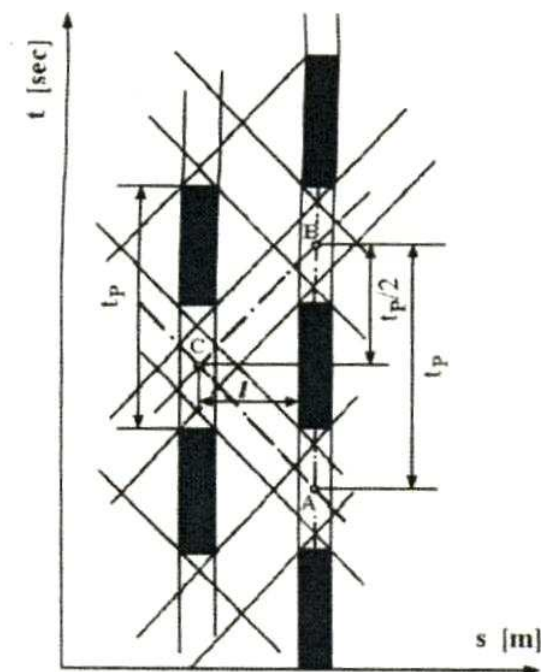
Az algoritmus kidolgozásához két szoftvert használtam, a Schlothauer & Wauer által kifejlesztett LISA+ szoftvert és a PTV AG által forgalmazott VISSIM szoftvert. Előbbit alkalmaztam a feladat megvalósításához, különös tekintettel a jelzéstervek, az összehangolás, és a forgalomfüggő logikák elkészítésére. Utóbbi szoftverrel tudtam modellezni a teljes vonali hálózatot, értékeltem az elkészült terveket, valamint össze tudtam hasonlítani a jelenlegi és az általam tervezett állapotot.

2. A jelenleg alkalmazott zöldidő-összehangolási algoritmusok bemutatása

A következő három alfejezetben néhány olyan módszert szeretnék bemutatni, melyeket zöldidő-összehangolás tervezésekor alkalmazni szoktak. Ezek között vannak hagyományosabbak, mint az első fejezetben részletezett osztóponti módszer, de vannak komoly matematikai programozási háttérrel ellátott eljárások is, úgy mint a MAXBAND, MULTIBAND módszerek. Az első két alfejezetben részletezett eljárások arteriális hálózatok irányítására lettek kifejlesztve, ugyanúgy, ahogy a PASSER módszer is, ám bemutatok olyan metódusokat is, melyek alapvetően hálózati irányításra lettek tervezve, de megvalósítható velük egy vonali hálózat koordinálása is.

2.1. *Osztóponti módszer*

Zöldidő-összehangolás tervezése során gyakran alkalmazott metódus az osztóponti módszer [1], leginkább egyszerűsége miatt. Elsőként meg kell határozni a legfontosabb paramétereket az összehangolandó hálózatra. Célunk ebben az esetben egy hangolási ábra felrajzolása, melyen a vízszintes tengelyen az út, a függőleges tengelyen az idő szerepel. (Meg kell jegyezni, hogy egyes források a két tengelyen lévő értéket felcserélik, így az idő kerül a vízszintes tengelyre, az út pedig a függőlegesre. A magyar szakirodalom ugyanakkor az előbb leírt változatot alkalmazza.) A hangolási ábrához adottak a szakaszon elhelyezkedő csomópontok távolságai, így ezeket az értékeket tudjuk megjeleníteni a vízszintes tengelyen. A csomópontok távolsága alatt a két stop vonal távolsága értendő.



1. ábra: A módszer nevét adó osztópont megjelenítése az út-idő diagramban
(forrás: [1])

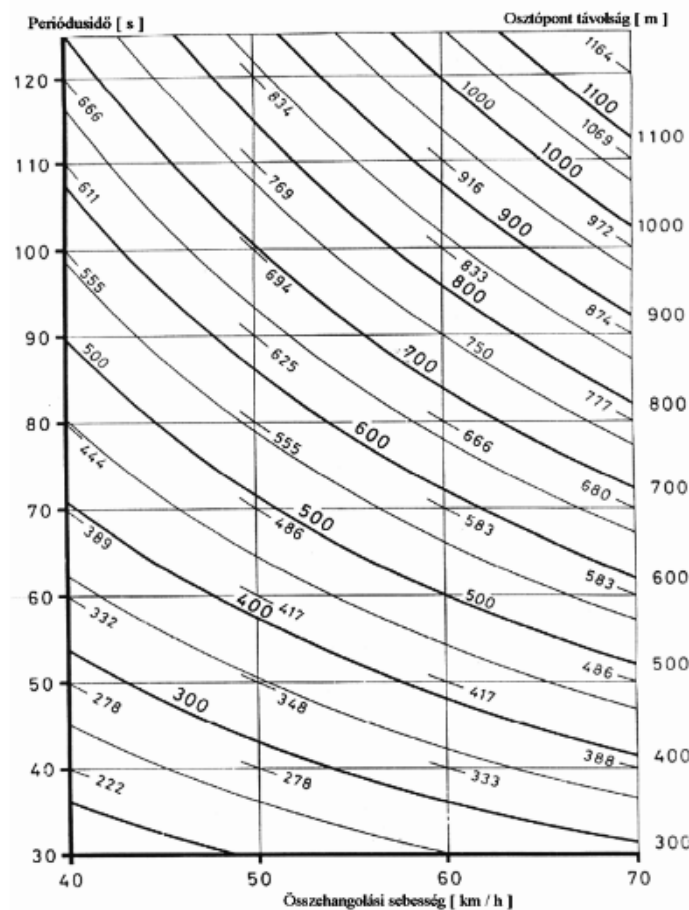
A tervezés kezdetén tudjuk, hogy hány csomópontot szeretnénk összehangolni, de ki kell választanunk azt is, melyek lesznek ezek közül a mértékadó csomópontok. Mértékadónak tekinthető az a csomópont, ahol a keresztirányú forgalom nagyobb mértékű, így a főirány zöldideje rövidebb. Fontos, hogy jól válasszuk meg a mértékadó csomópontokat a tervezés további sikeressége szempontjából.

Kifejezetten fontos érték a hangolás tervezésekor a ciklusidő megfelelő megválasztása. Ahhoz, hogy a hangolás akadálymentes legyen, minden csomópont ciklusidejének meg kell egyeznie, vagy egymás egész számú többszörösének kell lennie. Ez mindenféleképpen alapkövetelménye egy összehangolt *arteriális* (vonali) hálózatnak. A rendszer ciklusidejének kiválasztása során a csomópontok közül a legnagyobb ciklusidővel rendelkezőt kell alapul venni. Amennyiben az egyes csomópontok periódusidői nagy mértékben térnek el egymástól a szakasz két részén, a hangolást szét kell bontani és két külön feladatként megoldani.

Szintén fontos meghatározandó paraméter az összehangolt hálózatban közlekedő járművek sebessége, azaz az összehangolási sebesség. Ebben az esetben a hálózaton normál körülmények között haladó járművek tényleges sebességéből kell kiindulni. A hangolási ábrában az összehangolási sebességet a mértékadó csomópontokban a zöld jelzések végét összekötő egyenes meredeksége mutatja meg. Másik fontos érték a szomszédos csomópontok között értelmezhető zöldidő eltolások (*offsetek*) definiálása. Ezeket az értékeket nagyon pontosan kell meghatározni, mivel a szakaszon áthaladó járműveket sem fölöslegesen megállítani nem szeretnénk, sem a megengedett

sebességnél gyorsabb haladásra ösztönözni. Egyébiránt a zöldidő eltolások a hangolási sebesség és a csomópontok távolsága alapján számíthatóak az ismert összefüggésekkel.

A módszer nevében is megtalálható osztópontnak azokat a pontokat nevezzük, ahol az ellentétes irányban közlekedő járműoszlopok találkoznak. A periódusidő és az összehangolási sebesség ismeretében meghatározhatjuk az osztópontok ideális távolságát.



2. ábra: Összefüggés a periódusidő, az összehangolási sebesség és az osztópont távolság között
(forrás:[1])

A megoldandó feladat ezután adódik, ugyanis van egy ideális hangolási állapotunk, ahol az osztópontok adott távolságonként követik egymást, és van a valóságos hálózatunk, ahol az egyes csomópontok távolságai adottak, és csak a legritkább esetben egyeznek meg egymással. A két rendszert ezért össze kell hangolni egy olyan állapotra, amely a leghatékonyabb irányítást biztosítja a szakaszra. Itt leginkább azt az elvet kell szem előtt tartani, hogy azok a mértékadó csomópontok, ahol jelentős a keresztirányú forgalom, osztópont közelébe essenek a hangolási ábrában. Ez értelemszerűen azért fontos, mert az osztópontban találkoznak az ellentétes irányban haladó járművek, így a mellékiránynak hosszabb zöldidő adható.

Maga a hangolási ábra megszerkesztése olyan mérnöki feladat, melynek jelentős részét képezi a különböző lehetőségek kipróbálása, majd az optimális változat kiválasztása. A tervezéshez szükséges paraméterek a következők:

- az adott útszakasz, a csomópontok távolsága.
- az összehangolási sebesség.
- a periódusidő.
- a főirány és a mellékirányok forgalmi terhelése, valamint az ezekből számított zöldidők hossza.
- a zöldidő eltolások értékei.
- a kiadódó sáv szükséges szélessége, azaz a megállás nélkül a szakaszon átbocsátandó járművek száma.

A tervezés során a mértékadó csomópontok közé felrajzolt út-idő sávokból kell kiindulni, melynek meredeksége megegyezik a hangolási sebességgel. Amennyiben egy adott csomópontban szükség van a fázistervek módosítására, meg kell vizsgálni, hogy a fázisok sorrendjének felcserélésével nem küszöbölhető-e ki a probléma. Ha erre lehetőség van, mindenképpen élni kell vele, hiszen nagyobb hasznot hoz a hangolás javítása, mint amekkora a fáziscseréből adódó veszteség az adott csomópontban.

Ez a módszer a lehető legkorábbi zöldidő-összehangolási metódus, melyet máig előszeretettel alkalmaznak, de a megfelelő használatához ugyanakkor szükséges bizonyos tervezői tapasztalat is.

2.2. MAXBAND, MULTIBAND módszerek

A MAXBAND módszer [2] az Amerikai Egyesült Államokban alkalmazott legnépszerűbb metódus vonali hálózat összehangolása esetén. Ez a módszer is kétirányú forgalom összehangolására lett kifejlesztve, de jelentősen eltér az előzőleg bemutatottaktól. Amíg az osztóponti módszer esetén (2.1 fejezet) a lényegi kérdések a tervezőre voltak bízva, a MAXBAND matematikai programozás segítségével oldja meg a szükséges feladatokat. Ezek többnyire megegyeznek az előzőekkel, a program kiszámolja a ciklusidőt, a zöldidő eltolásokat, a hangolási sebességet, valamint a csomópontokban a legoptimálisabb fázissorrendet. A programozás célja a két irányban haladó út-idő sávok maximalizálása.

A módszer nagy előnye, hogy képes olyan összehangolásokat is létrehozni, melyek figyelembe veszik a két irány forgalmi terhelése közötti különbségeket. Jól alkalmazható olyan szakaszokon, ahol a csúcsidőszakban az egyik irány jóval nagyobb forgalmi igénybevételnek van kitéve, mint az ellenkező. A módszer vegyes egész értékű lineáris programozást használ, ennek segítségével tudja a mellékirány fázisok ideális sorrendjét is meghatározni. A programozás során a következő értékeket és korlátozásokat veszi figyelembe a módszer:

- a főirány, valamint a kanyarodó irányok forgalmi terhelése.

- a szükséges sávszélesség, tehát az a járműmennyiség, amit egy „zöldhullám” alatt át kell engedni.
- periódusidő korlátozása.
- a vizsgált szakasz megengedett sebességének korlátozása.
- a sebességek a szakaszon belüli változtatásának korlátozása.

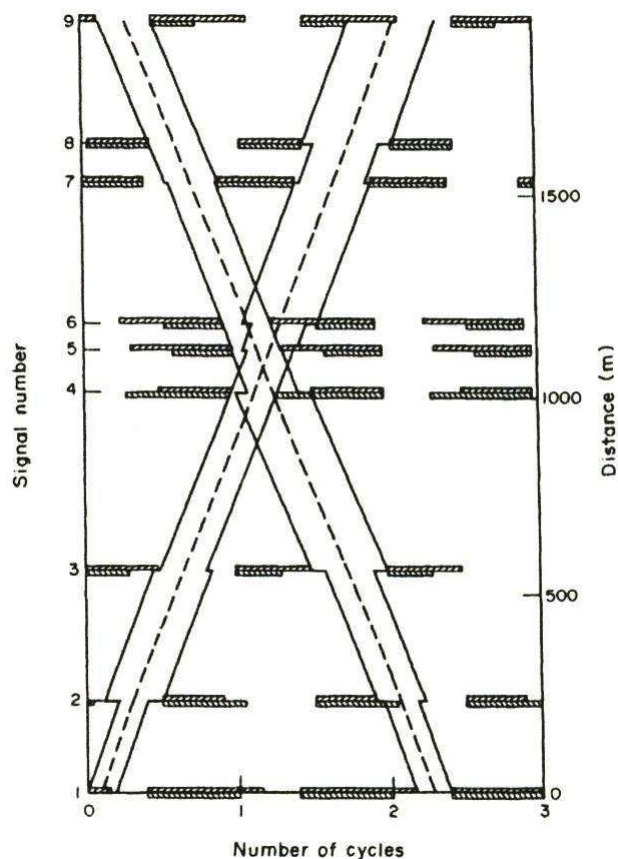
A program célja a következő értékek megtalálása:

- ciklusidő.
- zöldidő eltolások.
- sávszélesség mindkét irányban (nem feltétlenül egyezik meg).
- összehangolási sebesség.
- balra kanyarodó fázisok helye a jelzéstervben.

Mindezeket a célokat a program úgy próbálja megvalósítani, hogy közben a két irányban megállás nélkül átbocsátható járművek számának összegét igyekszik maximalizálni.

A MAXBAND módszer problémája, hogy a kívánt sávszélesség nem függ az aktuális forgalmi állapotoktól, emiatt nem képes minden helyzetben a leghatékonyabb irányításra. Ennek a hiányosságnak a kiküszöbölésére alkották meg a MULTIBAND módszert, a MAXBAND továbbfejlesztését.

A MULTIBAND módszer [2] az összehangolni kívánt szakaszt felosztja több rövidebb intervallumra, majd ezeket a későbbiekben külön kezeli. Ennek hatására a rendszer rugalmasabbá válik, az egyes szakaszokat pedig egyénileg súlyozhatjuk, a hálózat betöltött szerepük fontossága szerint. A MAXBAND-hoz képest itt nem kell állandó sávszélességet használnunk, a rész szakaszok sávszélessége egymástól eltérő is lehet.



3. ábra: Út-idő diagram a MULTIBAND módszer segítségével
(forrás:[2])

Hasonlóan az előző módszerhez, a MULTIBAND is vegyes egész értékű lineáris programozást használ. Azonban mivel ebben az esetben a szakasz részekre van bontva, a cél az egyes részek súlyozott sávszélességeinek a maximalizálása:

$$\max B = \frac{1}{n-1} \sum_{i=1}^{n-1} (a_i b_i + \bar{a}_i \bar{b}_i) \quad (1)$$

ahol,

$a_i, \bar{a}_i$: az egyes szakaszok súlyozó tényezői, mindkét irányban

$b_i, \bar{b}_i$: az egyes szakaszok sávszélességei, mindkét irányban

Ezen változtatások segítségével a MULTIBAND módszer a MAXBAND-hoz képest jóval hatékonyabb irányítást tesz lehetővé a vonali hálózatok összehangolása terén.

2.3. TRANSYT, PASSER és ALLONS-D módszerek

A következőkben tekintsünk át további három módszert, melyekkel lehetséges a jelzőlámpák összehangolása. Amíg a PASSER módszer az előzőekhez hasonlóan arteriális hálózatok irányítására lett kifejlesztve, a TRANSYT és ALLONS-D módszerekkel ettől eltérő elrendezésű hálózatokat is lehet optimalizálni.

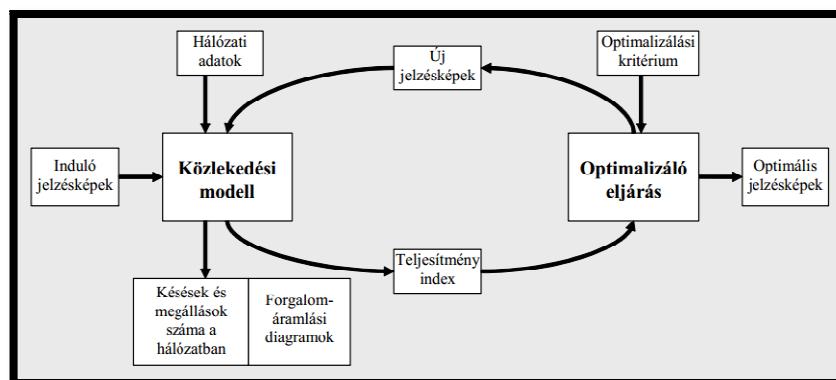
A PASSER [4] (Progression Analysis and Signal System Evaluation Routine) nevű program a MAXBAND-hez és a MULTIBAND-hez hasonlóan vonali hálózatok optimalizálására szolgál. A módszer célja tehát a sáv szélesség maximalizálása. Az eljárás bemenő adatai a következők:

- forgalom nagyság értékek, sáv számok.
- kanyarodó irányok forgalom nagysága.
- minimális fázisidők.
- tiltási mátrix.
- zöldidő eltolások.

A kimenő adatokat három féleképpen kaphatjuk meg a programból. Elsőként választhatjuk az összes adat megjelenítését táblázatos formában, melyek értékelését később magunk végezhetjük el. Választhatjuk a program által nyújtott ajánlatot is, amely vonatkozik a fázis sorrendekre, a ciklus időre és a zöldidő eltolások értékeire. Az általa ajánlott változatot a program olyan értékekkel támasztja alá, mint az átbocsátott járművek száma, a késési idők csökkenése, vagy a megspórolt üzemanyag mennyisége. A harmadik megjelenítési formaként választhatjuk az út-idő diagramot is, melyről grafikus úton kapjuk az információkat.

Mindenképpen meg kell említenünk a TRANSYT [3] módszert is ebben a témában, noha ennek a módszernek a lényegi része nem a vonali hálózatokra koncentrál. A TRANSYT egy integrált közlekedéstervezési módszer, amely magában foglalja a hálózat modellezését, valamint annak optimalizálását. Alapja, hogy a szimulációt követő értékelés során a program a vezetők számára „kényelmetlen” helyzetek számát vizsgálja, és a megfelelő *offset* alkalmazásával próbálja ezt minimalizálni.

A TRANSYT program felépítése a következő ábrán látható:

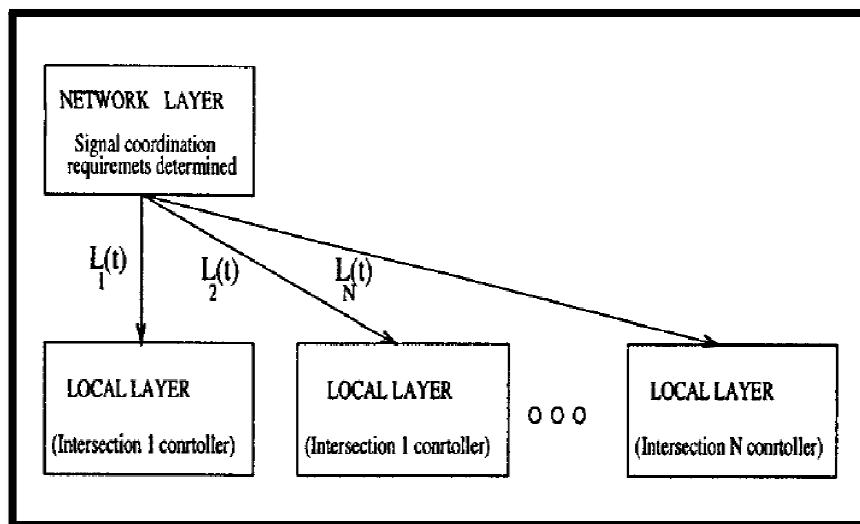


4. ábra: A TRANSYT program felépítése
(forrás: [10])

Az optimalizálás során a program az előzőekhez hasonlóan a periódusidőt, a fázis sorrendet és a zöldidő eltolások értékeit keresi, a járművezetőt érő negatív hatások minimalizálása mellett.

A harmadik módszer egy valós idejű, decentralizált hálózati irányítást mutat be, ez az ALLONS-D [5] (Adaptive Limited Lookahead Optimization of Network Signals - Decentralized) metódus. A rendszer a dinamikus optimalizálást a más rendszereknél is alkalmazott gördülő időhorizontú módszer segítségével hajtja végre. Az ALLONS-D célja a késési idők összegének minimalizálása. Késési idő alatt azt az időt kell érteni, amennyivel lassabban jut át az adott jármű az adott szakaszon (hálózaton), mint ideális esetben.

Az ALLONS-D módszer esetén az eljárás kidolgozói felismerték azt az összefüggést, hogy egy hálózat irányítása nem feltétlenül akkor lesz a leghatékonyabb, ha a rajta elhelyezkedő csomópontok irányítását egyesével optimalizáljuk. Ezért létrehoztak egy két rétegű irányítási formát, ahol a csomóponti rétegekhez egy $L_i(t)$ súlyozó tényező tartozik, mely függ az időtől. A hálózati réteg ezeket veszi figyelembe a meghatározott követelményeknek megfelelően.



5. ábra: Két rétegű irányítás az ALLONS-D módszer esetében
(forrás: [5])

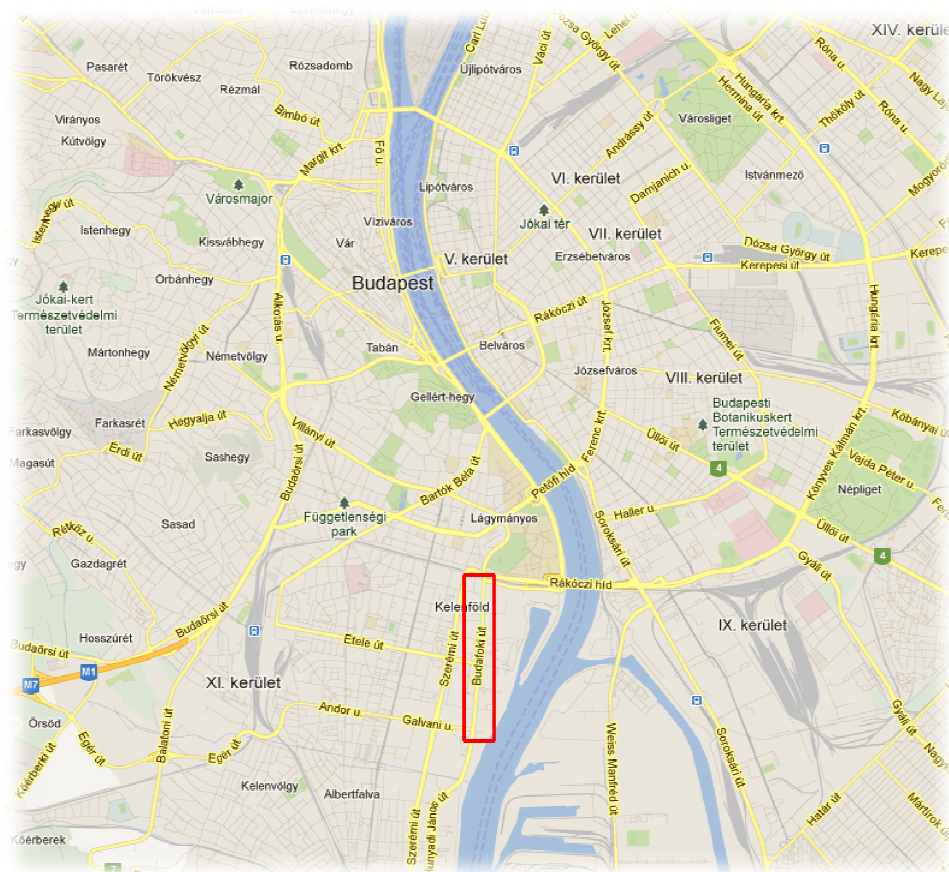
Előszeretettel használják ezt a módszert tranzit utak átbocsátóképességének javításához, de szimulációk igazolják, hogy a tömegközlekedés előnyben részesítése során is alkalmazható.

3. A vizsgált útszakasz és a jelenlegi forgalmi állapotok bemutatása

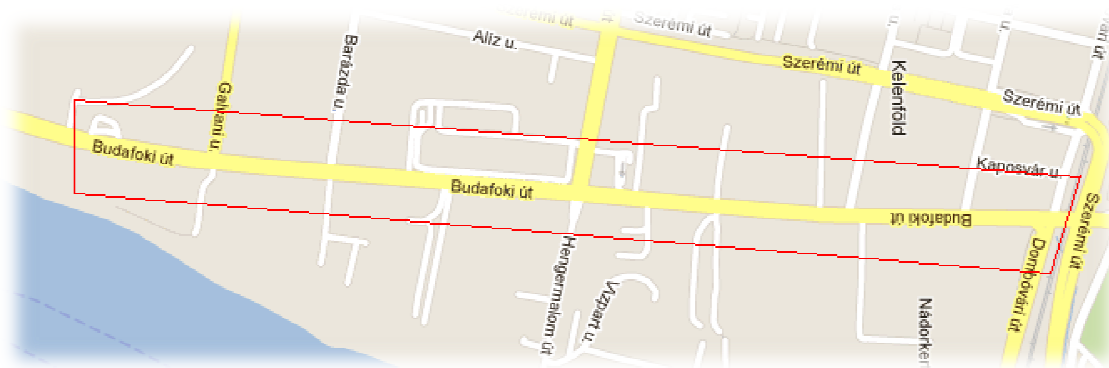
A harmadik fejezetben az általam választott és vizsgált szakasz, a Budafoki út kerül bemutatásra. Elsőként a budapesti hálózatban betöltött szerepét vizsgáltam meg, majd az elvégzett forgalomszámlálást mutatom be, egy adott módszertan alapján. Végül a forgalomszámlálás eredményeinek elemzése következik, hiszen a későbbi tervezési lépések is erre épülnek.

3.1. A vizsgált útszakasz bemutatása

A Budafoki út fontos szerepet játszik Budapest közlekedésében, leginkább a dél-budai régió forgalomlebonnyolódása szempontjából. Dolgozatomban a Budafoki út külső szakaszával foglalkozom, amely a Dombóvári úttól mindaddig tart, amíg az út Hunyadi János út néven folytatódik tovább.



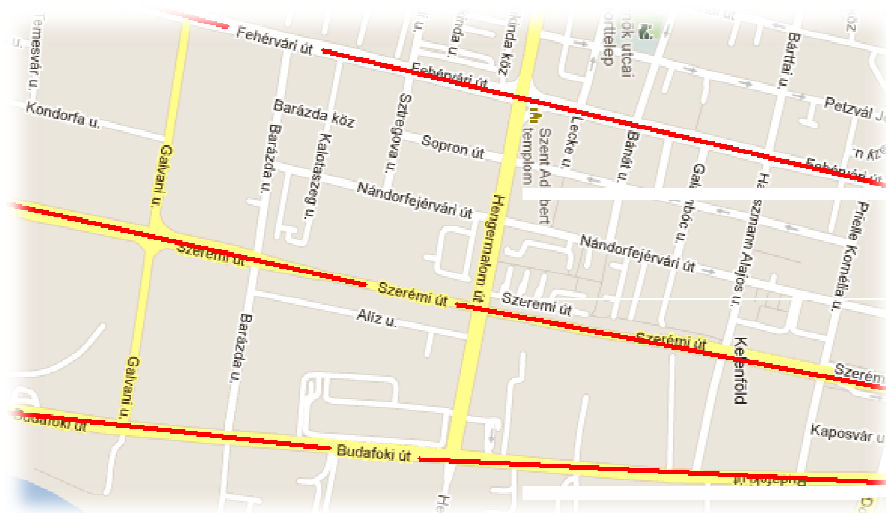
6. ábra: A Budafoki út elhelyezkedése Budapesten belül



7. ábra: A vizsgált útszakasz

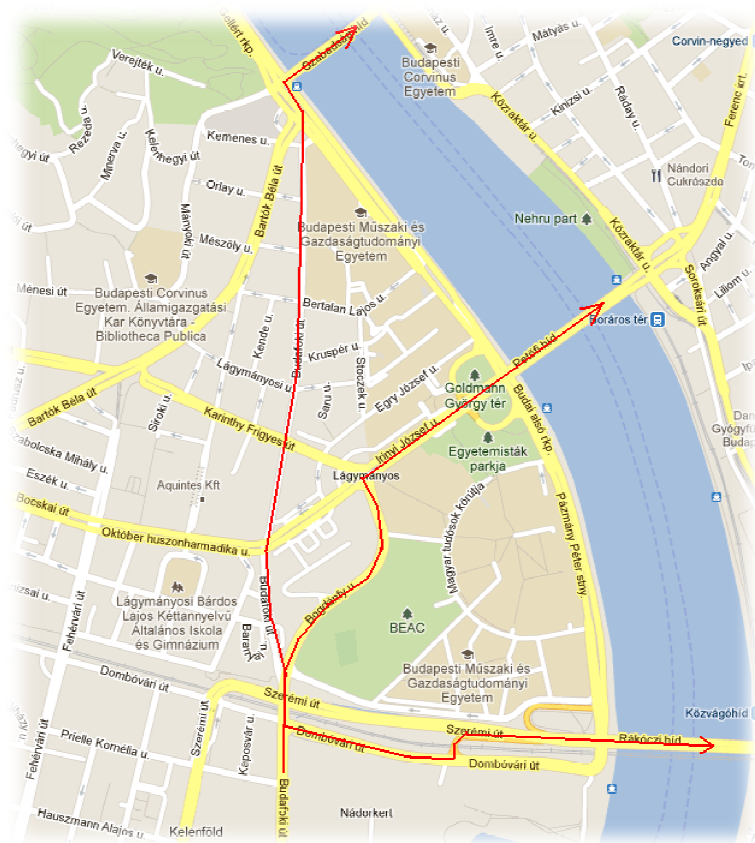
A Budafoki út belső szakasza, mely a Dombóvári úttól a Szent Gellért térig terjed, nem képezi a vizsgálat részét, itt a forgalom 2-szer 1 sávon bonyolódik le. A vizsgált szakasz, ahogy már említettem, fontos részét képezi a dél-budai hálózatnak, elsősorban speciális helyzete miatt. Az útszakaszon 2-szer 2 sávon bonyolódik le a forgalom, amely elsősorban a város szélére eső XXII. kerület lakosaiból, másrészt a vidékről beérkező közlekedőkből tevődik össze. A XXII. kerületből a szakasz a Leányka utcai felüljárón keresztül közelíthető meg, míg a vidékről érkezők a 6-os számú főútról csatlakozhatnak rá a vizsgált szakaszra. A 6-os út mintegy közvetlen folytatásaként is értelmezhető a Budafoki út, amely ennek is köszönhetően a nap minden részében nagymértékű forgalmat bonyolít le.

Mivel főútvonalról érkező forgalomról van szó, értelemszerűen a beömlő járműoszlop sebessége nagyobb, mint a belterületen közlekedőké, emiatt fontos szerepe van a sebességkorlátozó tábláknak is. A vizsgált útszakaszon a megengedett sebesség 50 km/h óra, de az azt megelőző részen 60-70 km/h-val is közlekedhetnek a járművek. A déli irányból érkező autósoknak azonban nem a Budafoki út jelenti az egyetlen alternatívát. A 6-os út, és a város csatlakozásánál a járművezetők választhatják a Szerémi utat is a Budafoki mellett, amelyen összehangolt jelzőlámparendszer található, a megengedett legnagyobb sebesség pedig 60 km/h. Ideális esetben a két útszakaszra jutó járműmennyiség megoszlása azok kapacitásainak arányával egyezik meg. Ez azonban a gyakorlatban ritkán fordul elő, a járművezetők nem feltétlenül az ideális feltételek alapján választanak útirányt. Harmadik párhuzamos alternatíva lehet a Fehérvári út, amely a Leányka utcai felüljáróról nyílik.



8. ábra: Térkép a három párhuzamos útról

Ez a szakasz térben távolabb helyezkedik el az előző kettőtől, valamint lakóövezet veszi körbe, ám a nagy számú keresztirányú kapcsolódási útvonal miatt muszáj említést tennünk róla. A Fehérvári út ellentétben a Budafoki és a Szerémi úttal szinte végig 2-szer 1 sávos, a csomópontok irányítása itt is főként jelzőlámpákkal van megoldva. Ezért amikor jelzőlámpa-összehangolást végzünk az adott szakaszon, figyelembe kell venni az előbb felsorolt tényeket. Aki az agglomerációból szeretné megközelíteni a város központját, ezen három alternatíva közül választhat. Céлом a Budafoki úti lámpák összehangolásával az, hogy a szakasz versenyképes partnere legyen a két másik párhuzamos útvonalnak, az ideális forgalomlebonnyódás érdekében, azaz ne alakuljanak ki torlódások a szakaszon, a járművek a megengedett legnagyobb sebességgel tudjanak haladni, valamint a megállások száma minimalizálva legyen. Tekintsünk most egy kicsit a szakasz másik felére, amely Budapest centrumának kapujában található, fontos Duna hidak közelében.



9. ábra: A pesti oldal megközelíthetősége a Budafoki út irányából

A Budafoki útról több kiemelkedő közlekedési ág is elérhető. Könnyen megközelíthető innen a pesti oldal a Rákóczi vagy a Petőfi hídon keresztül, de a közép- és észak budai rész felé is jók a közlekedési kapcsolatok. A Rákóczi hídra vezet fel közvetlenül ugyanakkor a Szerémi út is, így ennek a szakasznak a forgalma nem csak a Budafoki út járműveiből adódik. Ez a „találkozás” gyakran vezet torlódásokhoz, melyek szinte mindig hatással vannak az utak korábbi szakaszaira is. A Petőfi híd esetében a Budafoki út forgalma a Bocskai út forgalmával találkozik, itt a torlódás esélye kisebb, a forgalomlefolys ideálisabb. Fontos megjegyezni még, hogy a Dombóvári úton keresztül érhető el egy újabb jelentős közlekedési folyosó, a budai alsó rakpart. A Dél- és Észak-Buda között közlekedő autósok nagy része ezt az útvonalat használja, a magasabb megengedett sebesség miatt. Ezeket a megállapításokat igazolja a forgalomszámlálás eredménye is.(lásd: 3.3. Jelenlegi forgalmi állapotok bemutatása)

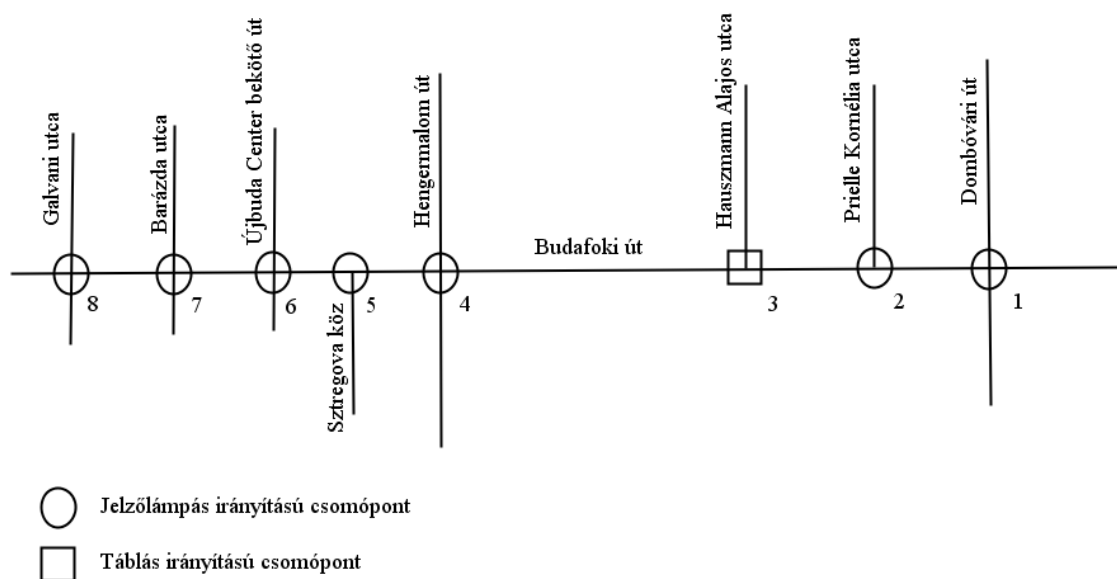
Amennyiben közvetlenül a Budafoki út környezetét vizsgáljuk, látható, hogy a vizsgált szakasz 8 helyen van keresztezve mellékutcák által, ezek közül csak a Hauszmann Alajos utca az, amely nem jelzőlámpával irányított. Ennek a forgalma amúgy sem túlságosan jelentős, mivel hozzá közel található a párhuzamos Prielle Kornélia utca, amellyel együtt képes lebonyolítani a nem túl nagy méretű keresztirányú forgalmat. Alapvetően az egész környékre jellemző, hogy a lakók száma meglehetősen kevés, főként irodaházak és telepek találhatóak a környéken, így a forgalom mennyisége is eszerint alakul. A Budafoki úttól keletre nem találhatóak mellékutcák a szakasz mellett

(kivétel a Sztregova köz), a Duna közelsége miatt nem épültek ki igazi útvonalak, az ebbe az irányba kanyarodó utak többnyire különböző telephelyek belső útjaira vezetnek. Jó példát láthatunk erre például a Barázda utcai vagy az Újbuda Center melletti csomópontban. Ez a féloldalasság mindenképpen kedvező zöldidő-összehangolás tervezésekor, mivel a kevésbé forgalmas keresztirányú szakaszoknak nem szükséges minden ciklusban kiadni a zöldidőt, elegendő csak abban az esetben, ha az adott irányból forgalmi igényt érzékelünk. Ezzel a témával részletesebben az 5. fejezet foglalkozik.

3.2. Az elvégzett forgalomszámlálás részletezése

Egy új tervezés kezdetekor elengedhetetlen a megfelelő és pontos adatgyűjtés. Mivel tervezésem alapja egy zöldidő-összehangolás elkészítése az adott – Budafoki úti - szakaszra, feltétlenül szükséges, hogy pontos, mért adatokkal tudjak dolgozni. Ennek megfelelően mielőtt megkezdődne a tervezési folyamat, forgalomszámlálást kellett végezni a kiválasztott területen, melyből kiderült az egyes forgalmi igények nagysága a szakasz egyes részein. Az ilyen jellegű adatgyűjtésnek akkor van értelme, ha az adatok megbízhatóak, frissek és tükrözik a valóságot, azaz alapjai lehetnek a szakasz és az adott csomópontok modellezésének.

A forgalomszámlálás előkészítésének első lépéseként meg kell határozni a számlálendő csomópontokat, valamint az ehhez szükséges létszámot. A Budafoki út vizsgált szakaszán hét darab lámpás csomópont található, ezek környezetében mindenképpen szükséges az adatgyűjtés.



10. ábra: Jelzőlámpás irányítású csomópontok elhelyezkedése

Minden esetben törekedni kell arra, hogy a forgalomszámlálást a lehető legkevesebb emberrel végezzük, leginkább takarékosági szempontokból. Ugyanakkor az is fontos, hogy ne számoljunk minden szakaszt duplán, hiszen az itt fellépő redundancia következtében a hibalehetőségek száma is növekszik, és egy idő után abba a hibába eshetünk, hogy nem tudjuk mi a valósághű eredmény, és mi alakult ki a hibáknak köszönhetően. Ezért tehát a számlálók létszámát úgy alakítottam ki, hogy lehetőleg minden egyes számláló külön szakaszt számoljon, maximum páran számoljanak olyan ágakat, amik más számolásokból is kiadódhatnak, ellenőrzés céljából. Ezt az elvet a konkrét szakaszra alkalmazva tekintsük a hét jelzőlámpás csomópontot, valamint vegyük hozzá az egyetlen fontosabb, csak KRESZ táblákkal szabályozott csomópontot, a Hauszmann Alajos utca becsatlakozását. Így összesen nyolc csomópont vesz részt a számlálásunkban. Meg kell vizsgálnunk ezek egymáshoz képest való elhelyezkedését, valamint a csomópontok ágait, valamint az esetleges tiltásokat.

1. táblázat: A forgalomszámlálás során számlálendő irányok száma

Csomópont száma	Számlálendő irányok száma	Forgalomszámlálók [fő]
1	9	3
2	3	1
3	4	1
4	10	3
5	4	1
6	4	1
7	4	1
8	6	2
Összesen:	44	13

Tapasztalatok alapján egy számlálóra négynél több csomóponti ágat nem célszerű rábízni, mivel könnyen követhetlenné válhat a forgalom, különösen a nagyobb volumenű ágak esetén. Azonban a négy, vagy annál kevesebb ág esetén is figyelni kell arra, hogy egy időben maximum 2 csomóponti ág jusson 1 számlálóra. Mivel jelzőlámpás kereszteződésekről van szó, főleg arra kell figyelni, hogy a három vagy négy számolt ág ne egy fázisba essen. Ugyanakkor azt is szem előtt kell tartani, hogy a nagy forgalmú ágak egyenletesen legyenek elosztva a számlálók között. Fontos kijelölnünk azokat a pontos pozíciókat, ahol a számlálók az adott csomóponton belül elhelyezkednek.

Itt praktikussági, hatékonysági, és ergonómiai szempontokat ajánlott figyelembe vennünk. Ideális esetben a számláló forgolódás nélkül képes végigkövetni az általa számlálendő járműveket, minden általa számolt irányból. A munkát leegyszerűsíti, ha a járműveket szemből látja érkezni, így több ideje marad egy-egy jármű regisztrálására. Miután kiosztottam a számláló embereket az egyes pozíciókba, valamint elláttam őket

az erre a célra gyártott számlálólapokkal (11. ábra), megkezdődhetett a forgalomszámlálás folyamata.

Időszak	BALRA KANYARODÓ "A" irány							EGYENESEN HALADÓ "B" irány							JOBBRA KANYARODÓ "C" irány						
	Szgk.	tehergépkocsi			busz			Szgk.	tehergépkocsi			busz			Szgk.	tehergépkocsi			busz		
		könnyű	nehéz	szerelvénymotor	szóló	csuklós	kerékpár		könnyű	nehéz	szerelvénymotor	szóló	csuklós	kerékpár		könnyű	nehéz	szerelvénymotor	szóló	csuklós	kerékpár
15:00-15:15																					
15:15-15:30																					
15:30-15:45																					
15:45-16:00																					

11. ábra: Forgalomszámláló lap minta

Az időpont kiválasztásában szerepet játszott a tervezés célja, azaz hogy a forgalomszámlálás a zöldidő-összehangolás megalapozására készül. Emiatt igyekeztem olyan időpontot választani, amikor a szakasz nem válik túltelítetté, azaz nem alakul ki rajta torlódás. Meg kell jegyezni azonban, hogy a megfelelő hangolás megtervezésével a torlódások is csökkenthetőek. Olyan időpontot kellett találni, amikor a járműfolyamok szabadon tudnak haladni, lehetőleg mindkét irányban, értelemszerűen kerülni kellett a reggeli, illetve a kora esti csúcsórát. Ugyanakkor a modell szempontjából érdekes lehet a hangolás alkalmazása nagyobb forgalomnagyságok esetében is. A dolgozat későbbi szakaszában (6.3. fejezet) természetesen megvizsgáltam azt az esetet is, amikor a modellezett forgalom jóval meghaladja a mért forgalom nagyságát.

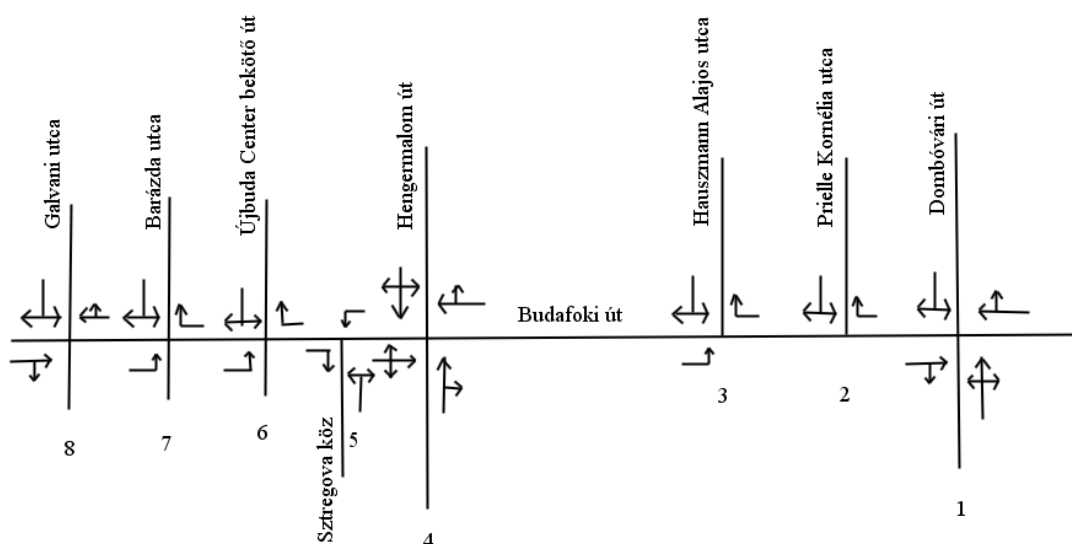
A választás végül a konkrét esetben egy délutáni időpontra esett, 15 órától 16 óráig tartó időszakban. Mivel a jelzőlámpákat órai forgalmakra tervezik, ezért a forgalomszámlálást egy 60 perces intervallumban végeztem el. Ez negyedórás bontásban történt, 15 percenként összesítve a számolt járműveket, különböző járműkategóriák szerint. Az egyes járműkategóriák megkülönböztetésére azért van szükség, mert az útterhelési jellemzőik olyan mértékben térnek el egymástól, például egy kerékpáros és egy szerelvénymotor között, hogy nem tehetjük meg, hogy ezeket a különbségeket elhanyagoljuk. Ennek érdekében felhasználtam az általános gyakorlatban használt egységjármű szorzókat, amelyek az előbb részletezett különbségeket kívánják

érzékelteni. Az egységjármű szorzók általánosan használt értékek, melyek a következő táblázatból olvashatóak ki (2. táblázat).

2. táblázat: Egységjármű szorzók kül- és belterületen
(forrás:[6])

Járműtípus	Külterületi szorzó (E_k)	Belterületi szorzó (E_b)
személy- és kistehergépkocsi	1,0	1,0
szülő autóbusz	2,5	1,8
csuklós autóbusz	2,5	2,5
könnyű tehergépkocsi	2,5	1,4
szülő nehéz tehergépkocsi	2,5	1,8
tehergépkocsi-szerelvény	2,5	2,5
motorkerékpár, segédmotoros kerékpár	0,8	0,7
kerékpár	0,3	0,3
lassú járművek	2,5	2,5

A számlálók pozícióit úgy alakítottam ki a korábban részletezett elveknek megfelelően, hogy a lehető legkevesebb járművet kelljen a számlálásban résztvevőknek leregisztrálniuk. Ezzel kapcsolatban általánosságban elmondható, hogy a Budafoki út forgalmát csak a szakasz elején, a végén, és egy köztes csomópontban – a legforgalmasabb helyen, a Hengermalom úti kereszteződésben – számoltuk, mindkét irányban. A maradék 5 számlálandó csomópontban csak a főútról lekanyarodó, és a mellékutcaíráról a főútra kanyarodó járműveket számoltuk.



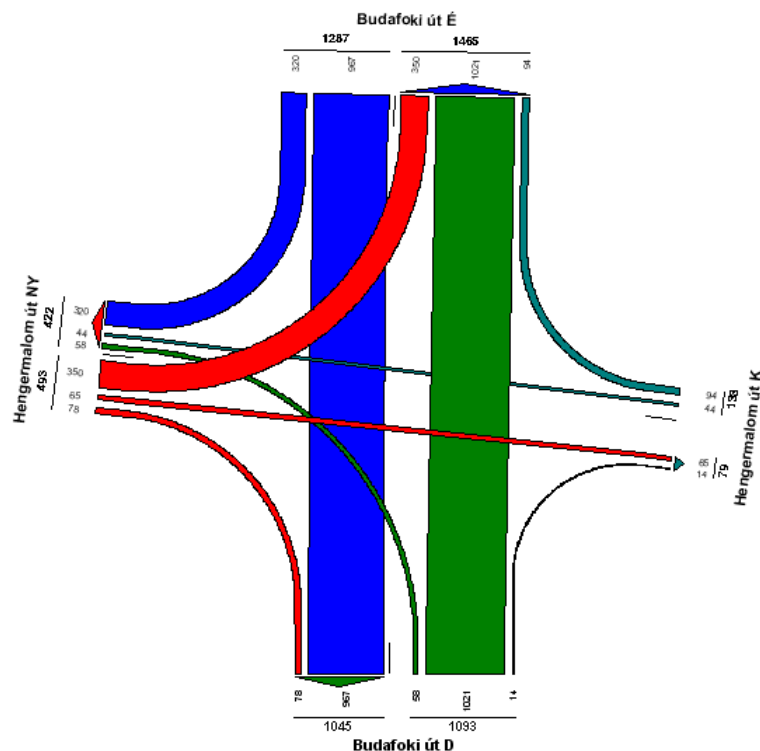
12. ábra: Számolt irányok csomópontonként

A szakasz korábban részletezett aszimmetriája miatt minden csomópontba csak az egyik irányból csatlakozott mellékutca, emiatt ezeken a helyeken elegendő volt egy ember a számláláshoz. A Hengermalom úti csomópont teljes számolása mellett azért döntöttem, mert az itt kiadódott értékek vizsgálata mintegy ellenőrző információként jól jellemzi a számlálás minőségét. A kelet felőli belső iparterületekről érkező járműveket az Újbuda Center melletti csomópontban számoltattam meg, majd ezen adat alapján végeztem becslést a Barázda utcai és Galvani utcai csomópontokra

3.3. Jelenlegi forgalmi állapotok bemutatása

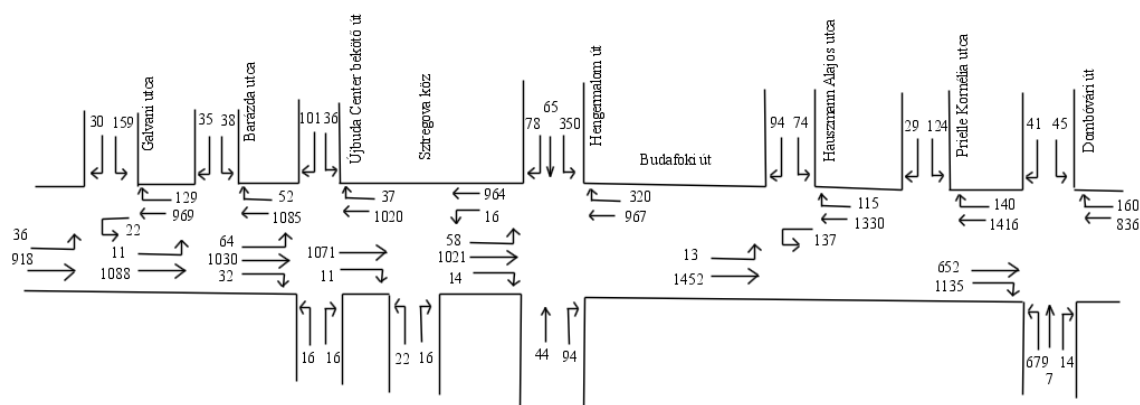
Miután elvégeztük a számlálást, elkezdődhetett az adatok összerendezése, belőlük a főirány forgalomnagyságának kiszámítása, ugyanis a hangolásban ezek az értékek játsszák a legnagyobb szerepet.

Összehasonlítva a szakasz végpontján, közbenső pontján, és kezdőpontján mért értékeket, valamint a mellékutcába távozó, és onnan érkező járműveket elmondhatjuk, hogy a számlálás minősége megfelelő volt, az eltérések 10%-os hibahatáron belülre estek. Ezekre a hibákra magyarázat lehet az útközben leparkoló, vagy visszaforduló járművek „eltűnése”, de természetesen az emberi hiba sem kizárható (például a járműkategória megítélésében). Miután megvizsgáltam a számlálás minőségi megfelelőségét, az adatokat egységjármű (EJ) mértékegységben rendeztem össze, majd ezekből készítettem forgalomáramlási ábrákat a könnyebb áttekinthetőség érdekében (lásd Mellékletek 4).



13. ábra: Forgalomáramlási ábra minta

Ezek az ábrák az egyes csomópontokra a kanyarodási irányok volumenei vannak feltüntetve, a szemléltetésben segítségképpen az egy irányból érkező irányok egyező színnel. A forgalom számszerű mennyiségét az adott kanyarodási irányban lévő forgalomáramlási sáv vastagsága szemlélteti. E mellett készítettem egy olyan áttekintő ábrát is, ahol a teljes szakasz forgalmi terhelése fel van tüntetve, itt tehát végigkövethető a hangolás szempontjából fontos főirányban a járművek mennyiségének változása.



14. ábra: A vizsgált szakasz forgalmi terhelése [jármű/óra]

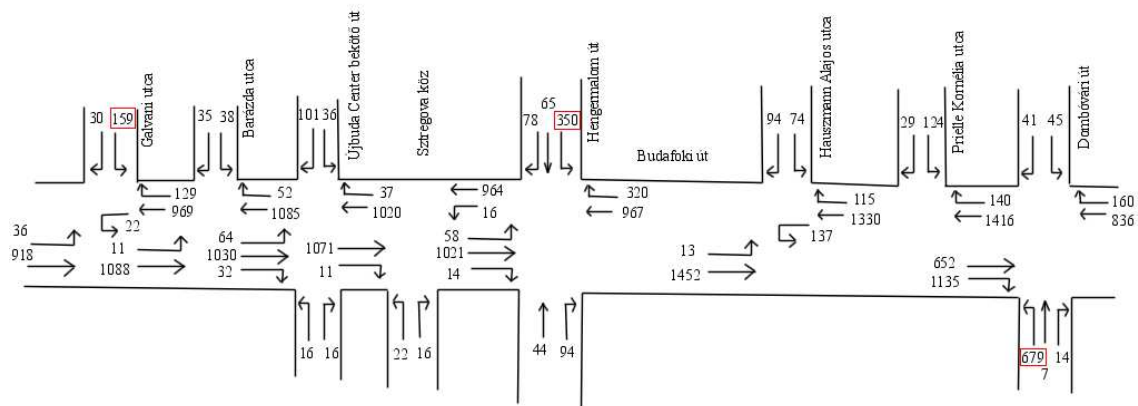
A forgalomszámlálási eredmények kiértékelése során a következő információkat tudtam meg a vizsgált szakasról. A főirány forgalma szinte végig ezer egységjármű felett

alakult a szakaszon mindkét irányban, egy-két helyen ugyanakkor megközelítette az 1500 egységjármű/óra forgalmat is. Mivel a szakaszon a főirányban mindenhol 2 sáv áll rendelkezésre a haladásra, ez nem jelentett kritikus telítődést a szakaszon. Általánosságban elmondhatjuk, hogy a Dombóvári út és a Hengermalom út közötti részen a forgalom nagyobb volumenű volt, mint a Hengermalom út és a Galvani utca között. Ez a trend mindkét irányban érvényesült.

3. táblázat: A Budafoki út forgalma

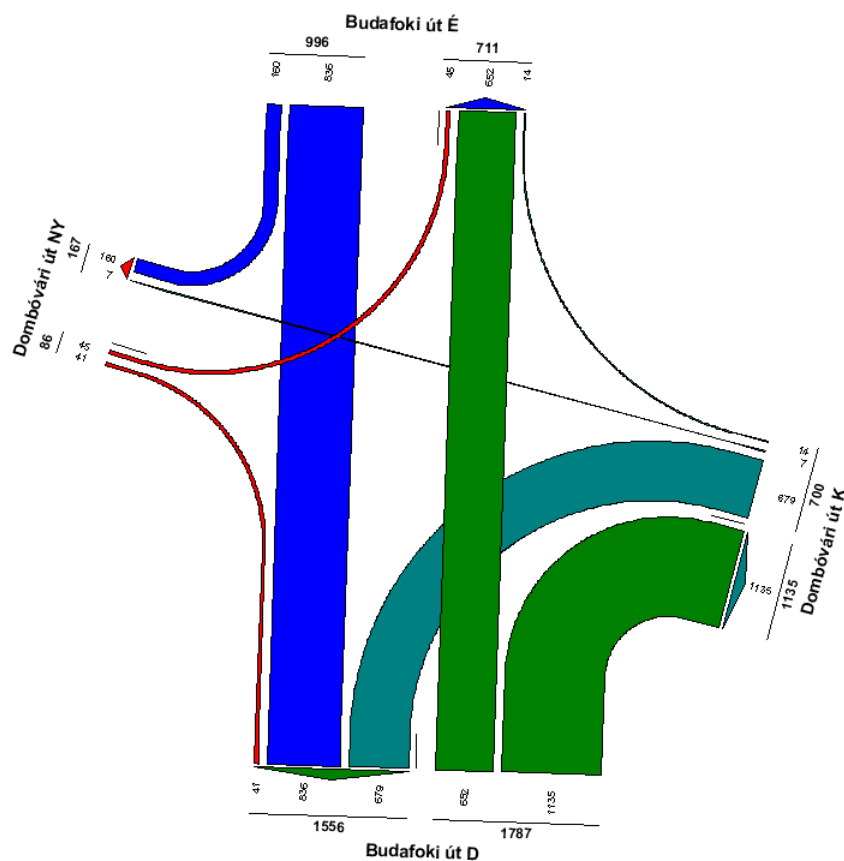
Mellékutca	É -> D irány forgalom [jármű/óra]	D -> É irány forgalom [jármű/óra]
	996	
Dombóvári út		
	1556	1663
Prielle K. u.		
	1445	1526
Hauszmann A. u.		
	1287	1465
Hengermalom út		
	1057	1093
Sztregova köz		
	1057	1082
Újbuda Center		
	1137	1126
Barázda u.		
	1098	1099
Galvani u.		
		954

A főútvonalról az észak felé tartó irányban a mellékutcákba balra kanyarodó sávok forgalma sehol nem érte el a száz egységjárművet sem, ami mindenképpen pozitív a zöldidő-összehangolás tervezésével kapcsolatban. A dél felé haladó irányban a balra kanyarodás csak a Sztregova köz esetében van megengedve, de itt is csak 16 egységjármű volt a forgalom egy óra alatt. A mellékirányokból érkező kanyarodó közlekedők mennyisége szinte sehol sem haladta meg a 100 egységjárműt óránként, éppen ezért ezeken a helyeken szinte mindenhol mágneses hurokdetektorokat alkalmaznak. Ez alól kivételt képez a Prielle Kornélia utcából balra kanyarodó 124 EJ, ezek feltehetően azok az autósok, akik a Szerémi úton közlekedve nem szeretnének felhajtani a Rákóczi hídra. A Galvani utcából balra kanyarodó irány átlagosnál nagyobb forgalmát azok az Egérút felől érkező autósok adják, akik a budai alsó rakparton szeretnének tovább közlekedni észak felé. Jelentősen kiemelkedik még továbbá a Hengermalom utcából nyugat felől balra kanyarodó 350 egységjármű, de ehhez az igényhez igazodva a csomópontban két balra kanyarodó sáv is található.



15. ábra: Nagyobb forgalmú mellékirányok

Meg kell vizsgálni még mindenképpen a Dombóvári út – Budafoki út csomópontot, mivel ez a kereszteződés eltér a választott szakasz többi csomópontjától.



16. ábra: Budafoki út – Dombóvári út csomópont forgalomáramlási ábrája [jármű/óra]

Amennyiben a dél felé haladó irányt tekintjük, ez a csomópont vizsgálatom kezdete, éppen ezért kiemelt figyelemmel kell kezelni. Az észak felől a Budafoki útról érkező forgalom nagyság nagyjából megegyezik a Dombóvári útról kelet felől balra kanyarodó

forgalom nagyságával, így mindenképpen érdemes megvizsgálni, hogy a hangolásunkat melyik forgalmi hullámhoz igazítsuk. A dél felől érkező irányban a Dombóvári úti csomópont az utolsó a vizsgált szakaszon, ugyanakkor ez a Budafoki út egyik legkritikusabb része is. A rakpart és a Rákóczi híd felé közlekedő nagymértékű (1135 EJ/óra) jobbra kanyarodó forgalom számára csak egy viszonylag rövid kanyarodó sáv van biztosítva, amelyről a forgalom könnyen visszatorlódhat az egyenesen haladó sávba, ezzel torlódást okozva és lassítva az áthaladást. Ugyanakkor itt az egyenesen haladó irány, az előbbieket követve alacsonyabb, mint a Budafoki út korábbi szakaszain, még a 700 egységjármű/óra értéket sem éri el.

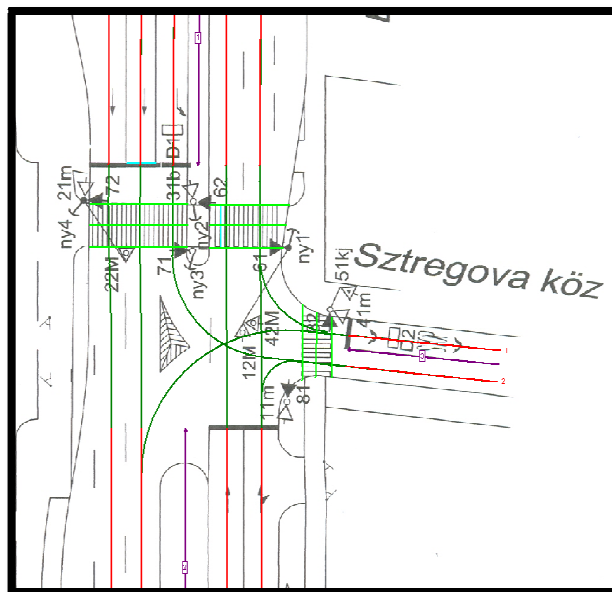
4. Zöldidő-összehangolás tervezése különböző periódusidőkre

Az összegyűjtött adatok alapján el kellett végezni a szakaszon elhelyezett jelzőlámpák összehangolását. Ezt a LISA+ szoftverrel készítettem el, ezért az első alfejezetben a program felépítését és használatát mutatom be. Az összehangolás elkészítése során törekedtem a dinamikus irányításra, ezért több jelzéstervet is készítettem az egyes csomópontokhoz, különböző periódusidővel. Ezt a folyamatot mutatja be a második alfejezet. A harmadik alfejezetben végül az azonos periódusidőkkel rendelkező programok összehangolására került sor, így a tervezett irányítással a szakaszon közlekedő járművek rövidebb idő alatt érhetnek el céljukhoz.

4.1. LISA+ szoftver bemutatása

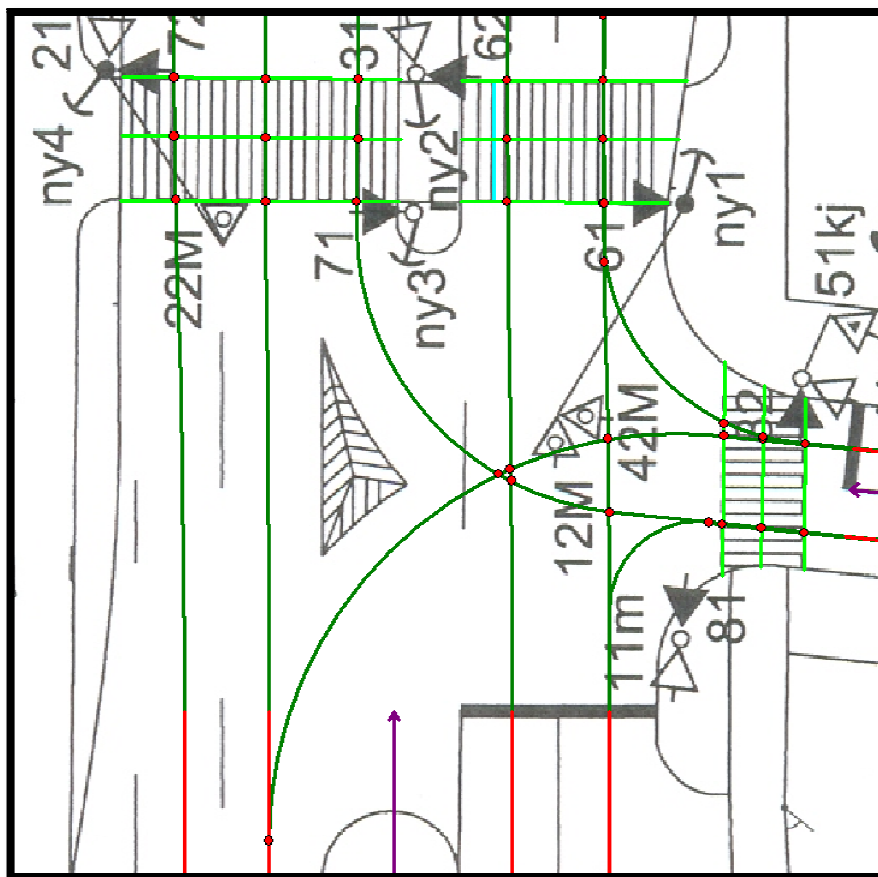
A forgalomszámlálásban részt vevő hét darab jelzőlámpás csomópont modellezését a LISA+ szoftverrel [7] végeztem el. A szoftver alapvetően csomópontok modellezésére, geometriai adatok tárolására, jelzőlámpás forgalomirányítás tervezésére, és jelzőlámpás csomópontok összehangolási programjainak elkészítésére használható. Ezeket a lépéseket a megfelelő sorrendben követve építhető fel a tervezési folyamat a legalapvetőbb részekről a szerteágazó, bonyolult logikák megvalósításáig.

Első lépésként mindegyik csomópontnál létre kell hozni a tervezés alapjául szolgáló csomópont statikus adatait. Ennek alapja a BKK által szolgáltatott helyszínrajz (lásd Mellékletek 1), melyet digitalizálva használok a tervezés háttereként.



17. ábra: A LISA+ grafikus tervezőfelülete

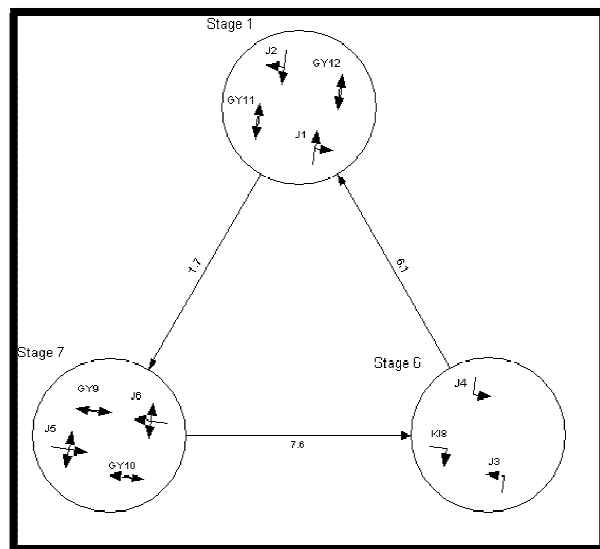
Erre a háttérre viszem fel a forgalmi sávok elhelyezkedését olyan formában, ahogy az a valóságban is megtalálható. A forgalmi sávok után definiáltam a csomópontban megengedhető mozgásokat, majd ehhez megadtam az egyes irányok kanyarodási íveit, ívsugarukkal együtt, melyet a közbensőidő számításnál kell figyelembe venni, ugyanis a kanyarodó irányoknál a tervezési sebesség az ívsugár függvénye. Amennyiben a csomópontban irányt és sávot váltó járművek viselkedése nem csak egy egyszerű körív mentén történik, létrehozhatunk segédvonalakat, melyekkel bonyolultabb mozgások is definiálhatók. A járműforgalmi sávokhoz hasonlóan a helyszínrajzról át kell venni az egyes gyalogátkelők és kerékpáros keresztező irányok geometriáját is. Emellett, hogy a tervezésnél reális értékeket kapjunk, szükséges beállítani a megfelelő méretarányt is. Amennyiben létrehoztuk a megfelelő közlekedési irányokat, a programmal megjeleníthetők az áthaladó járművek egymással, illetve gyalogosokkal vagy kerékpárosokkal létrejövő konfliktuspontjai.



18. ábra: Konfliktuspontok megjelenítése a LISA+ programban

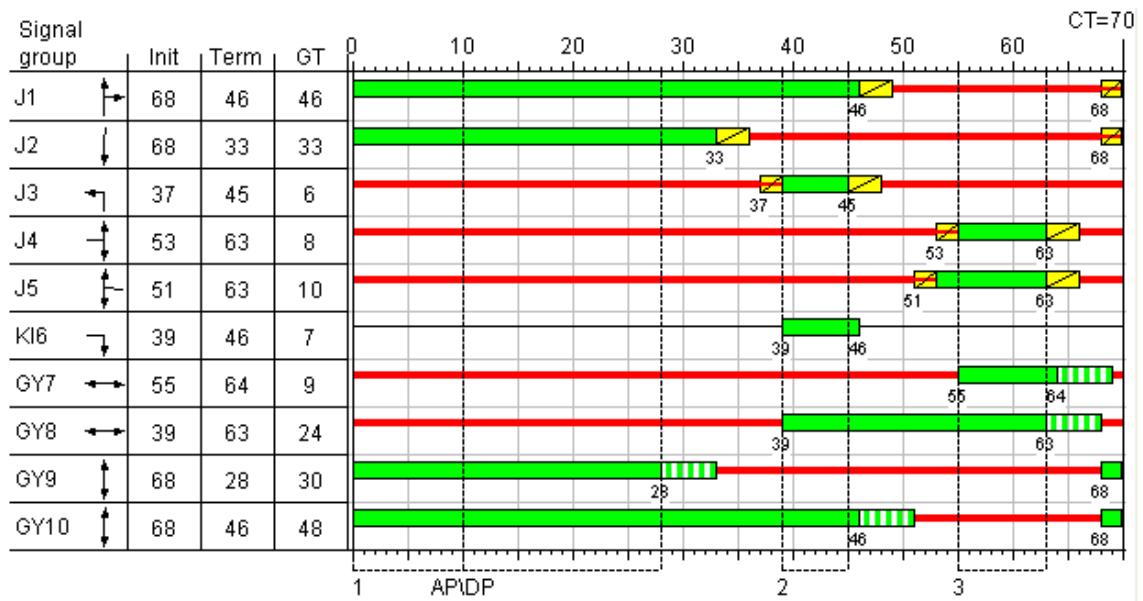
Ezek a konfliktusok jelennek meg később a tervezésnél a tiltási mátrixban, illetve innen kapjuk majd a közbenső idő számításánál a kihaladási illetve behaladási távolságokat. Ha ezzel is végeztünk, jelzőlámpás csomópont esetén következik az egyes jelzőcsoportok hozzárendelése az adott irányokhoz. Megadhatjuk a jelzőfej típusát,

beállíthatjuk azok statikus értékeit (pl. piros-sárga 2 másodperc), és definiálhatjuk, hogy maszkos avagy telizöldes irányításról van szó. A jelzőcsoportok elnevezését célszerű úgy megadni, ahogyan az a hivatalos helyszínrajzon szerepel. Amennyiben kész vagyunk ezekkel a teendőkkel, amiket egy grafikus felületen végezhetünk el, exportáljuk az adatokat a tervezés további lépéseihez. Az így létrehozott jelzőcsoportokat ezután fázisokba rendezzük, egy halmazba integrálva az egy időben megengedett mozgásokat.



19. ábra: A Barázda utcai csomópont fázisai

Ezeket a halmazokat a fázissorrend tervezésénél a program a minden lehetséges sorrend szerint összerendezi, majd matematikai módszerek (26. ábra, 27. ábra, 28. ábra) segítségével kiválasztja a számunkra ideálisat. Ennél a lépésnél figyelembe kell venni az egyes fázisok közötti közbenső időket, melyek kiszámításakor szükség van a megelőző fázisban érdekelt irány kihaladási hosszára, az azt követő fázishoz tartozó irány behaladási hosszára a konfliktuspontig, valamint az egyes közlekedő csoportok kihaladási és behaladási sebességére. Ezeket a sebességeket a Magyarországon érvényes szabványból határozhatjuk meg, külön gyalogosokra, kerékpárosokra, valamint a gépjárművekre is. Utóbbiak esetében kanyarodásnál nem statikus értéket szolgáltat a szabvány, hanem a sebességet a kanyarodási sugártól teszi függővé, mint ahogy azt már korábban említettem. Az így kiadódott közbenső időket használjuk fel később a jelzéstervek ellenőrzésére. A jelzéstervek első lépéseként meg kell határoznunk az adott program ciklusidejét, majd ezt követően egy grafikus felületen tudjuk felvenni az egyes jelzőcsoportokhoz tartozó jelzésekép-váltásokat.

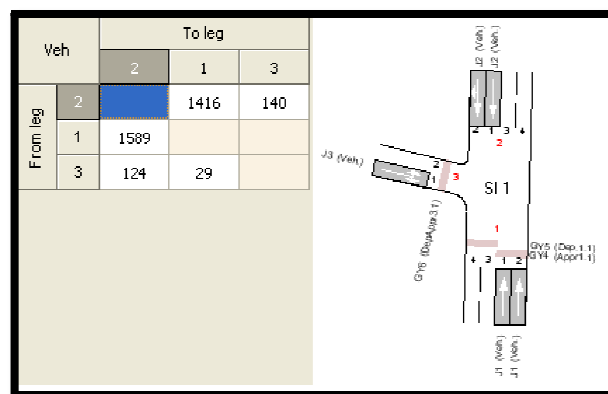


20. ábra: Jelzésterv minta a LISA-ban (Újbuda Center bekötő út, $T_C = 70$ mp)

Meg kell adnunk a zöldidő kezdő és befejező másodpercét, a program pedig a jelzőcsoport típusától függően alakítja ki a fennmaradó időtartamot. Előnyös tulajdonságnak tekinthető, hogy a program minden esetben azonnal ellenőrzi a közbensőidők meglétét, és jelez, amennyiben a korábban meghatározott feltételeknek nem tesz eleget az aktuális program. Egy csomóponthoz értelemszerűen több program is létrehozható, melyekkel a szimuláció során meghatározott időpontban programváltást hozhatunk létre. Ebben az esetben meg kell határoznunk a jelzésterven belül egy programváltási pontot is. Így nem csak fix jelzésterveket hozhatunk létre, hanem forgalomfüggő programok elkészítésére is lehetőség van. Ez jelen esetben azt jelenti, hogy létrehozhatunk különböző programváltási utasításokat, melyek függhetnek az időtől vagy az egyes detektorok és nyomógombok aktuális állapotától. Ezeket a detektorokat és nyomógombokat szintén a programban hozhatjuk létre, hozzárendelve őket az egyes jelzőcsoportokhoz. Az elnevezésnél célszerű itt is a helyszínrajzon feltüntetetteket felhasználni, ez a későbbi adatexportálásnál is megkönnyítheti a dolgunkat. A különböző forgalomtól függő elemek viselkedését is meg kell határozni, például hogy a detektor milyen jellegű legyen, amennyiben egyszer áthaladtak rajta, onnantól kezdve az értéke maradjon változatlan, vagy csak az áthaladást érzékelje, mintegy forgalomszámláló detektorként. Amennyiben létrehoztuk ezeket a nyomógombokat és detektorokat, el kell készítenünk az adott program(ok)hoz tartozó logikát. Ezt a LISA+ programban blokkdiagramos formában tehetjük meg, számítógépes algoritmusok mintájára.

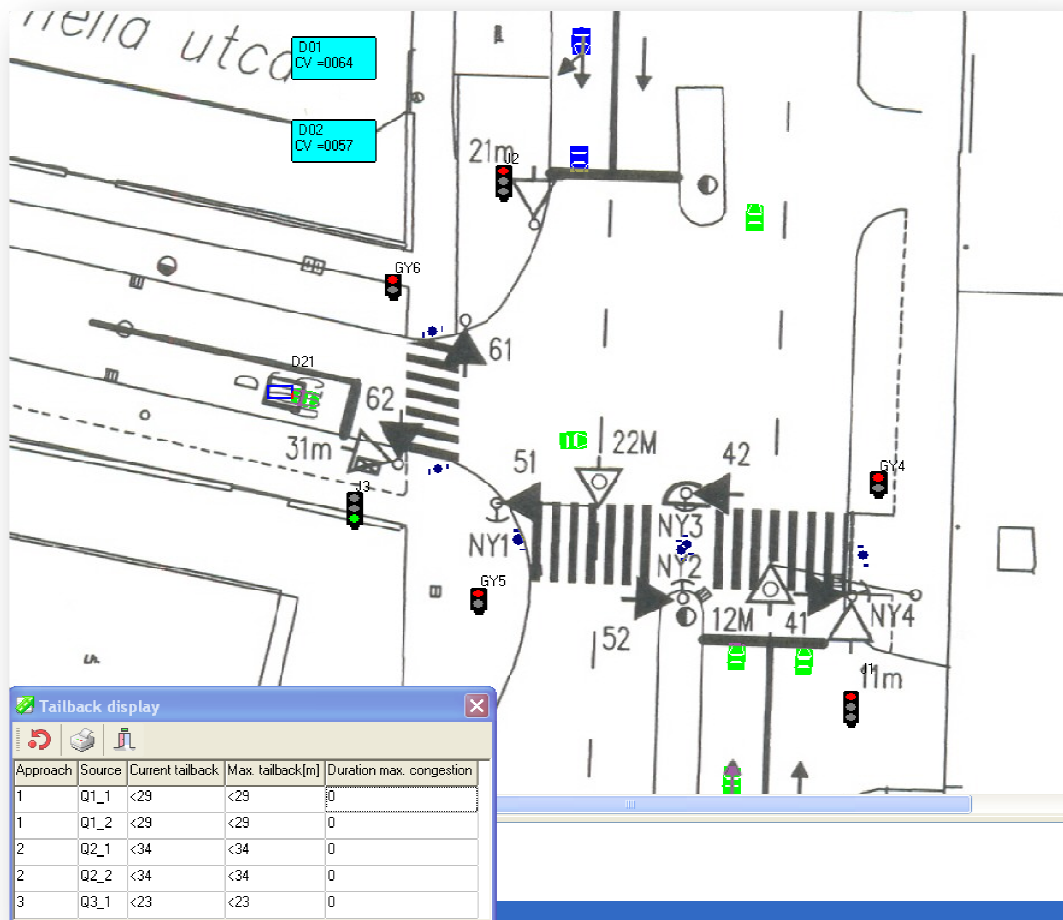
Ezt a logikát később meghívva a szimuláció során ellenőrizhetjük a működés megfelelőségét. A jelzőlámpás tervezés adatait a jelzőcsoportoktól kezdve a fázisterveken keresztül egészen a logikáig exportálhatjuk a programból, akár egy másik szimulációs program (VISSIM), akár egy konkrét forgalomirányító berendezés (pl. Actros VTC 3000 [8]) számára.

Ahhoz, hogy valóság-hű szimulációt tudjunk létrehozni a programban, szükség van az előzetesen előkészített forgalmi adatok felvitelére is. Amennyiben létrehoztuk a csomóponton belül megengedett közlekedési irányokat, ezekhez mátrixos formában hozzárendelhetjük az adott forgalomnagyságokat.



21. ábra: Forgalmi adatok beviteli felülete a LISA+ szoftverben

Ezeket megtehetjük egységjármű/óra mértékegységben is, megadhatjuk a konkrét forgalomszámlálás eredményeit, de százalékosan is bejelölhetjük az egyes járműtípusok előfordulási arányát a szakaszon. Természetesen itt nem csak a gépjárműforgalom adatairól van szó, a gyalogosok és keresztező kerékpárosok számát is megadhatjuk egy órára lebontva. Ezeknek az értékeknek a prezentálására a program lehetőséget ad forgalomáramlási ábrák készítésére, melyek segítségével könnyen és átláthatóan jellemezhetjük az adott csomópontot. Ha megvagyunk a forgalmi adatok felvételével, áttérhetünk a szimulációs környezetbe. A szimuláció során beállíthatjuk, hogy milyen logika alapján szeretnénk a programokat lefuttatni, vagy fix program esetén melyiket kívánjuk használni. Meghatározhatjuk a közlekedő járművek alapvető adatait is, mint például azok sebessége, de olyan adatokat is megadhatunk, mint a követési távolság vagy egyéb különböző vezetéstechnikai jellemzők. A szimuláció időtartamát beállítva el is indíthatjuk azt, melyet grafikus megjelenítés segítségével követhetünk. A futtatás közben valós időben követhetjük a fázisterv aktuális alakulását, kiírathatjuk az egyes forgalmi sávokhoz tartozó aktuális sorhosszakat, vagy vizsgálhatjuk az áthaladó járművek számát egy adott keresztmetszeten.



22. ábra: Szimulációs környezet a LISA+ szoftverben

Ugyanakkor hozzá kell tennünk, hogy ez a szimulációs felület nem a LISA+ erőssége, ezért ebben a projektben ehhez a munkarészhez a VISSIM szoftver szimulációs környezetét használjuk, amely képes arteriális forgalom modellezésére is.

A LISA+ szoftver másik rendkívül hasznos felülete, amelyről eddig még nem tettünk említést, a jelzőlámpás kereszteződések összehangolásával foglalkozik. Ebben az esetben a korábbiakban részletezett módon lemodellezett csomópontok közül kell kiválasztanunk azokat, amelyeket szeretnénk a hangolásba bevenni. Ezek után, amennyiben a megfelelő sorrendben kiválasztottuk őket, meg kell adni minden csomópontnál azokat a jelzőcsoportokat – mindkét irányban – amelyekre a hangolást tervezni szeretnénk. Ezek a legtöbb helyen értelemszerűen az egyenes irányok lesznek, de előfordulhat olyan nagyobb forgalmú kanyarodó irány is, ahol az előző állítás nem érvényesül. Amennyiben ezzel is megvagyunk, nagyon fontos beállítani az egyes csomópontok távolságát, amelyekre a hangolást tervezzük.

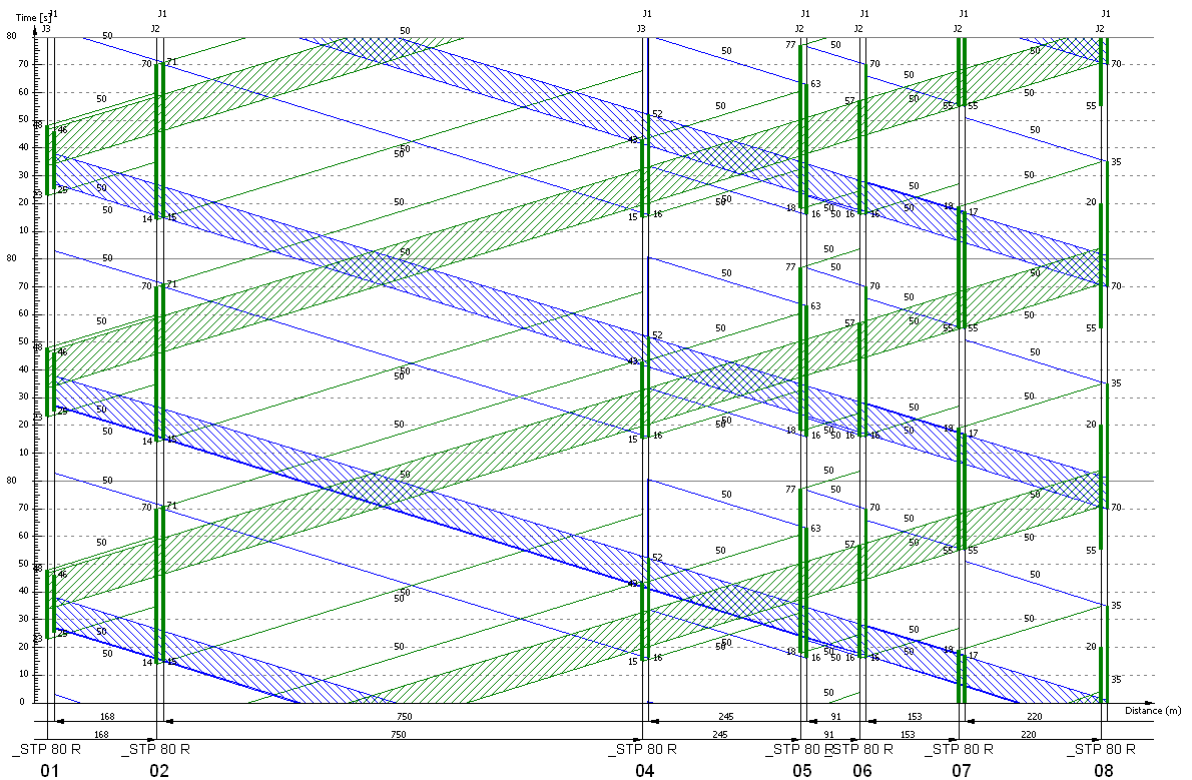
Intersection	Signal groups	
	DOT	OD
01	J3	J1
02	J2	J1
04	J3	J1
05	J2	J1
06	J2	J1
07	J2	J1
08	J2	J1

Intersections	Distances	
	DOT	OD
01 - 02	168	168
02 - 04	750	750
04 - 05	245	245
05 - 06	91	91
06 - 07	153	153
07 - 08	220	220

23. ábra: Az összehangolt irányok jelzőcsoportjai, valamint az egyes csomópontok közötti távolság értékek [m]

Ebben az esetben törekednünk kell a minél pontosabb értékekre, ezzel is csökkentve a hiba lehetőségét. Fontos, hogy a távolságokat a két stop vonal között mérjük le, éppen emiatt az is előfordulhat, hogy két csomópont közötti távolság a két irányban eltér egymástól, a geometriai különbségek miatt. Az én feladatom esetében jelentős eltérést nem tapasztaltam az ellenkező irányban lévő csomópont távolságok között, ezért átlagos távolság értékekkel dolgoztam.

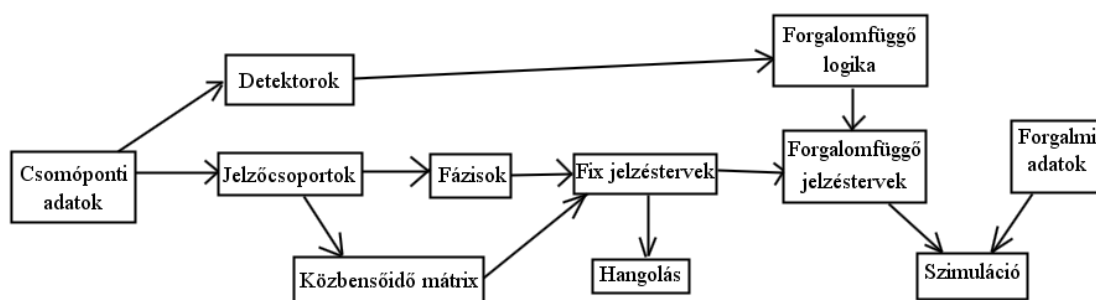
Ezzel meghatároztuk a hangoláshoz szükséges statikus adatokat, és kiválaszthatjuk a hangolásban résztvevő jelzőlámpa programokat minden egyes csomópontban. Természetesen olyan jelzésterveket kell választanunk egy adott hangoláshoz, melyek periódusideje megegyezik, vagy a periódusidők egymás egész számú többszörösei. Amennyiben ez is megtörtént, megjeleníthetjük a zöldidő-összehangolási ábrát, ahol a vízszintes tengelyen a távolságok, a függőleges tengelyen az idő szerepel.



24. ábra: Hangolási ábra (80 másodperces periódusidejű programok)

A csomópontokban található függőleges vonalak jelzik az adott helyen működő jelzéstervet, a ferde vonalak pedig a két irányban közlekedő járműblokkokat. Ezeknek a vonalaknak a meredeksége a hangolási sebesség függvénye, értelemszerűen minél laposabb az egyenesünk, a hangolási sebesség annál magasabb. Amennyiben kívánjuk, a program képes megjeleníteni azt a sáv szélességet, azaz gyakorlatilag azt a járműszámot, ahányan képesek az adott szakaszon megállás nélkül végighaladni. Zöldidő-összehangolás tervezésénél célunk ennek a járműszámának a maximalizálása, lehetőleg mindkét forgalmi irányban. Lehetőség van arra a programban, hogy a hangolási ábrán megjelenő jelzésterveket időben eltoljuk, mivel gyakorlatilag ezzel a programon nem változtatunk, csak az egyes csomópontokban futó fázistervek közötti időtartamot, az úgynevezett offsetet változtatjuk.

Az alábbi blokkdiagram szemlélteti a LISA+ szoftver teljes munkafolyamatát:

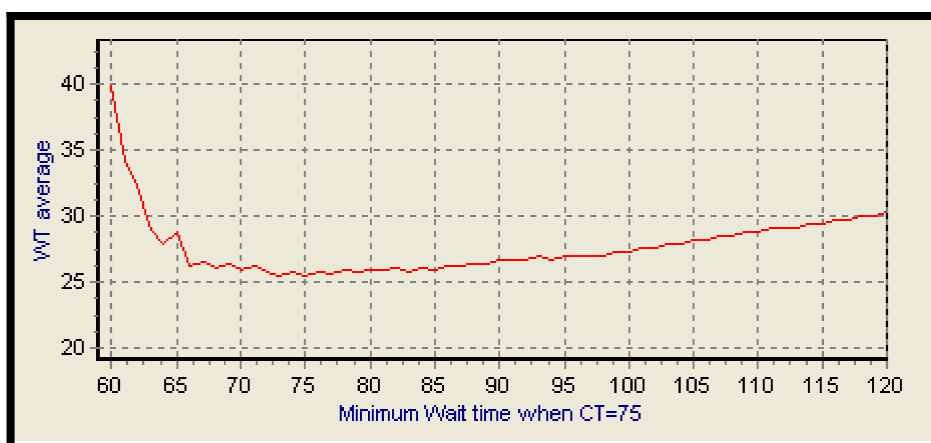


25. ábra: Blokkdiagram a LISA+ programban történő munkafolyamatról

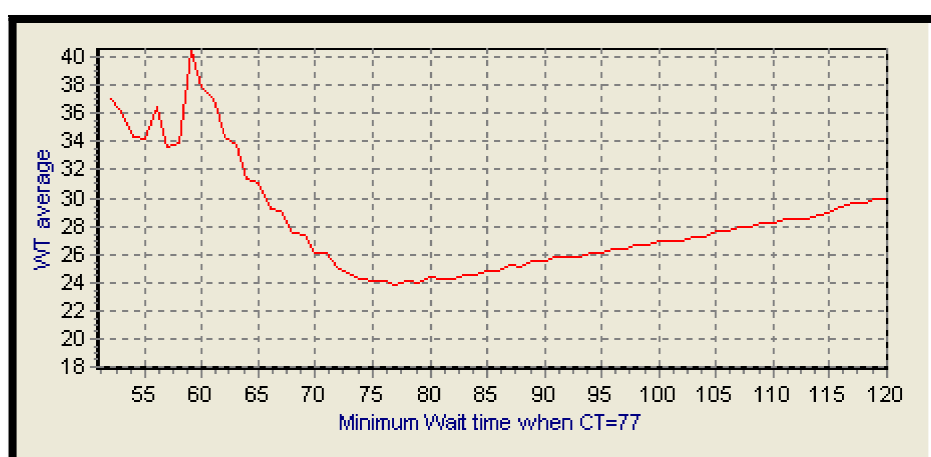
4.2. Jelzéstervek elkészítése

Jelzőlámpás forgalomirányítás során nagyon fontos a megfelelő jelzéstervek elkészítése. A jelzéstervek, más néven fázistervek vagy fázisidőtervek megadják, hogy az egyes jelzőcsoportok mikor milyen jelzéseképet mutatnak, vagy egy másik értelmezés szerint, hogy az egyes fázisok mikor kapnak engedélyt a csomóponton történő áthaladásra. Egy adott jelzésterv elkészítésénél több dologra is tekintettel kell lenni. Elsőként meg kell határozni a program ciklusidejét (periódusidejét), amely azt határozza meg, hogy hány másodpercenként ismétlődik az adott program. Miután ez megtörtént, a közbensőidő-mátrix ismeretében, a forgalmi igények alapján kell kiosztani a zöldidőket, figyelembe véve a periódusidő korlátait.

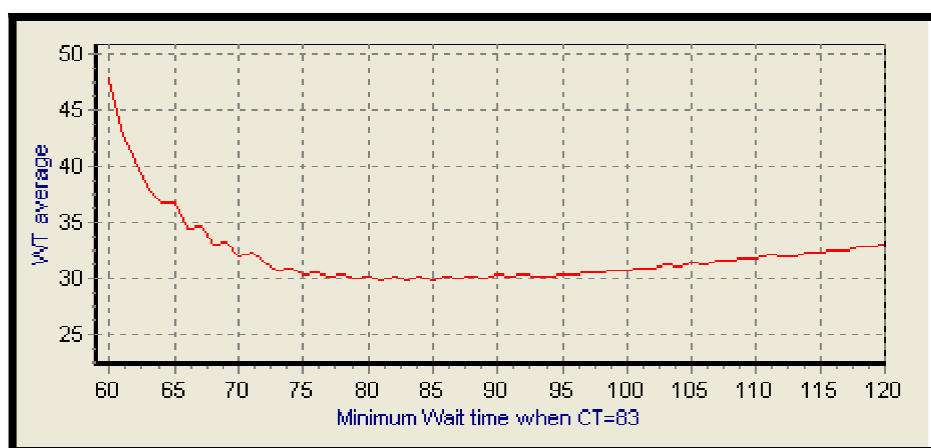
Vizsgáljuk most ezt meg a konkrét feladattal kapcsolatban. A programok elkészítését a LISA+ szoftver segítségével végeztem, mivel a program képes a bevitt adatok alapján kiszámítani a kívánt fázisterveket. Bemenő adatként már a korábbi fejezetben említettek szerint felhasználtam a közbensőidő-mátrix értékeit (BKK adatai alapján, lásd Mellékletek), és a számolt forgalmakat az egyes irányokra. Az optimalizálás során ezeket az értékeket az algoritmus nem változtatja meg. Lehetőség van továbbá a ciklusidő optimalizálására is, ez esetben azt a periódusidőt választja ki az algoritmus, amelyik esetén a csomópontbeli várakozási idők átlaga a legkisebb. Elsőként ezzel a módszerrel szerettem volna meghatározni a csomópontokra az optimális ciklusidő értékeket. Ehhez a LISA+ három féle optimalizálási algoritmust kínál fel, melyek a Schnabel/Lohse, a HBS és az RVS optimalizálási módszerek. A módszerek közötti eltérést a következő grafikonok szemléltetik.



26. ábra: Schnabel/Lohse-féle optimalizálás függvénye (Budafoki út – Dombóvári út)



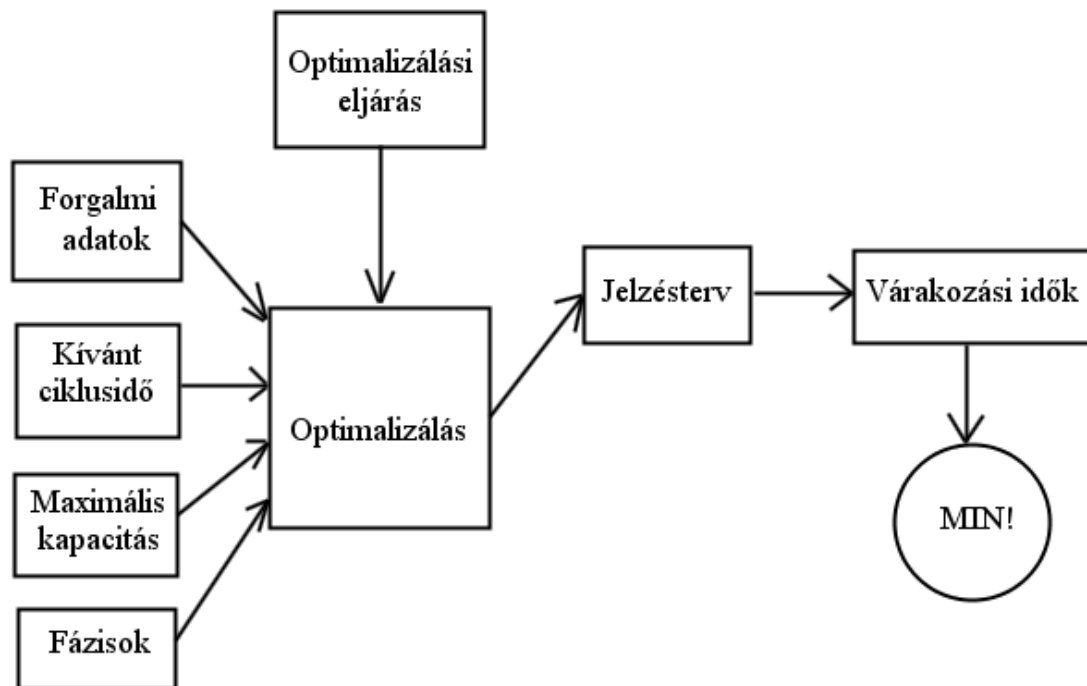
27. ábra: HBS típusú optimalizálás függvénye (Budafoki út – Dombóvári út)



28. ábra: RVS típusú optimalizálás függvénye (Budafoki út – Dombóvári út)

Az ábrákon a várakozási (késési) idők átlagát láthatjuk megjelenítve a periódusidő függvényében.

A módszer kiválasztása mellett be tudjuk még állítani az általunk engedélyezett maximális kapacitás-kihasználás értékét, valamint hogy a fázisterv legfeljebb hány fázisú legyen. A feladatban a kapacitás-kihasználás értékét a hangolási feladatoknál megszokottak szerint 80%-ra állítottam, és négynél több fázis nem engedélyeztem egy programon belül.



29. ábra: Az optimalizálás folyamata a LISA+ programban

Mivel a hangolásomat szeretném forgalomfüggővé tenni a periódusidők változtatásával is, az egyes csomópontokhoz több fázistervet is el kell készítenem. Itt élek egy kis egyszerűsítéssel, ami a feladatot illeti, ugyanis a program nem lesz képes akármilyen ciklusidőt beállítani, csak az általam előre beállított értékeket, azaz 60, 70, 75, 80, 90, 100, 110, valamint 120 másodpercet. Éppen ezért minden csomóponthoz létre kellett hoznom 8 darab programot, minden ciklusidőhöz egyet. Mivel azonban három különböző optimalizálási módszer áll rendelkezésemre a programban, ezért ezt a nyolc programot elkészítettem mind a három algoritmus segítségével. Így egy csomópontoz tartozóan lett 24 darab fázistervem, azaz 7 csomópont esetében összesen 168 darab. Ezekre a programokra vizsgáltam a várakozási időket.

A LISA+ program rögzített, vagy szabadon választható ciklusidejű optimalizálás esetén is a legkisebb átlagos várakozási idővel rendelkező programot választja ki. Ez természetesen meghatározza az egyes fázisok (és ezzel együtt a jelzőcsoportok) zöldidejét, valamint a fázisok sorrendjét is. Előfordulhat az is, hogy egyes esetekben a ciklusidő változtatásával a fázissorrend változtatása csökkentheti a várakozási időket.

A célom ebben az esetben az volt, hogy adott periódusidőnél meghatározzam, hogy melyik algoritmus adja ki számomra a legkedvezőbb eredményt. Ezért megvizsgáltam az egyes csomópontokhoz és ciklusidőkhöz tartozó késési idők átlagát, ami a következőképpen alakult:

4. táblázat: A késési idők átlaga csomópontonként a Schnabel/Lohse optimalizálási módszerrel

Schnabel/Lohse	60	70	75	80	90	100	110	120
1	39,902	25,888	25,503	25,808	26,609	27,292	28,724	29,985
2	7,067	7,216	6,943	7,139	7,546	7,967	8,398	8,466
4	67,329	24,003	23,818	23,81	24,643	25,051	26,913	28,527
5	9,357	8,07	7,643	7,728	6,69	6,245	5,904	5,93
6	11,292	11,261	11,396	11,553	11,769	12,425	12,742	13,438
7	18,249	14,398	13,4	12,726	11,633	10,796	10,144	10,02
8	12,887	12,416	12,51	12,63	12,932	12,907	13,335	13,452

5. táblázat: A késési idők átlaga csomópontonként a HBS optimalizálási módszerrel

HBS	60	70	75	80	90	100	110	120
1	30,061	26,04	24,054	24,336	25,399	26,625	28,232	29,989
2	5,538	5,986	6,636	7,303	7,742	8,194	8,657	9,125
4	29,134	23,853	24,508	23,088	22,928	23,823	24,927	26,286
5	9,161	8,063	7,636	7,27	6,682	6,236	5,894	5,63
6	11,081	10,38	10,153	9,986	10,459	11,008	12,208	12,612
7	20,81	14,164	13,381	12,707	11,614	10,776	10,123	9,61
8	12,933	11,998	11,677	11,427	11,464	11,943	12,817	13,323

6. táblázat: A késési idők átlaga csomópontonként az RVS optimalizálási módszerrel

RVS	60	70	75	80	90	100	110	120
1	47,78	32,064	30,405	30,219	30,113	30,706	31,818	32,907
2	9,439	9,481	9,098	9,211	9,491	9,821	10,184	10,197
4	351613948,6	31,571	29,665	28,69	29,096	29,17	30,232	31,628
5	11,358	9,515	8,918	8,438	7,715	7,204	6,838	6,803
6	13,873	13,369	13,351	13,418	13,457	13,976	14,209	14,826
7	22,995	17,193	15,808	14,77	13,28	12,254	11,519	11,007
8	16,128	14,943	14,737	14,648	14,376	14,57	14,887	14,993

Vizsgáltam emellett a nem rögzített ciklusidő esetében az optimális periódusidőhöz tartozó várakozási idő értékét, ami értelemszerűen kisebb vagy egyenlő volt az egyes rögzített periódusidős értékekhez képest.

7. táblázat: A minimális várakozási időhöz tartozó optimális periódusidő értékek (Schnabel/Lohse módszerrel)

Csomópont száma	Periódusidő	Várakozási idő
1	75	25,503
2	47	6,617
4	74	23,316
5	110	5,904
6	63	10,83
7	119	9,688
8	71	12,335

8. táblázat: A minimális várakozási időhöz tartozó optimális periódusidő értékek (HBS módszerrel)

Csomópont száma	Periódusidő	Várakozási idő
1	77	23,855
2	60	5,538
4	85	22,409
5	120	5,63
6	81	9,959
7	120	9,61
8	86	11,207

9. táblázat: A minimális várakozási időhöz tartozó optimális periódusidő értékek (RVS módszerrel)

Csomópont száma	Periódusidő	Várakozási idő
1	83	29,884
2	64	9,042
4	80	28,69
5	112	6,779
6	74	13,035
7	120	11,007
8	90	14,376

Ezek után, ahhoz, hogy megállapítsam az egyes ciklusidejű programok minőségét, szükségem volt valamilyen statisztikai módszerre, amivel sorrendet állítok fel az egyes algoritmusok között. A módszerem a következő volt, minden periódusidő esetén vizsgáltam a várakozási idők eltéréseinek négyzetét az optimális ciklusidőhöz tartozó várakozási időhöz képest. Majd az adott módszerhez és periódusidőhöz tartozó négyzetes eltérés értékeket átlagoltam. Így redukáltam a program értékelését egy számszerű értékre. Ezeket egymáshoz képest vizsgálva következő eredményeket kaptam:

- Alacsony periódusidők (60 másodperc) esetén a legjobb értéket a HBS algoritmus adta. Az RVS módszerrel ilyen alacsony periódusidők esetén a kapacitás-kihasználás értéke 100% fölé emelkedett, tehát ez semmiképpen sem használható.
- Közepes periódusidő értékeknél (70, 75 másodperc) szükséges használni a Schnabel/Lohse algoritmust.
- Hosszabb ciklusok esetében (80, 90, 100, 110, 120 másodperc) a legkedvezőbb az RVS módszer használata, ez biztosítja legkisebb eltérést az optimális programoktól
- Összességében a legkisebb átlagos eltérést a 100 másodperces ciklusidejű program biztosította.

10. táblázat: Schnabel/Lohse módszerrel készült programok értékelése

	60	70	75	80	90	100	110	120
1	244,80%	103,04%	100,00%	102,41%	108,86%	114,52%	126,85%	138,24%
2	114,06%	118,92%	110,10%	116,40%	130,05%	144,97%	161,08%	163,69%
4	833,87%	105,98%	104,35%	104,28%	111,71%	115,44%	133,23%	149,69%
5	251,18%	186,83%	167,58%	171,33%	128,40%	111,89%	100,00%	100,88%
6	108,71%	108,12%	110,73%	113,80%	118,09%	131,62%	138,43%	153,96%
7	354,82%	220,87%	191,31%	172,55%	144,18%	124,18%	109,64%	106,97%
8	109,15%	101,32%	102,86%	104,84%	109,91%	109,49%	116,87%	118,93%
Átlag	288,08%	135,01%	126,70%	126,52%	121,60%	121,73%	126,59%	133,20%

11. táblázat: HBS módszerrel készült programok értékelése

	60	70	75	80	90	100	110	120
1	158,80%	119,16%	101,68%	104,07%	113,36%	124,57%	140,06%	158,04%
2	100,00%	116,83%	143,58%	173,90%	195,43%	218,92%	244,36%	271,49%
4	169,03%	113,30%	119,61%	106,15%	104,69%	113,02%	123,74%	137,60%
5	264,77%	205,11%	183,96%	166,74%	140,86%	122,69%	109,60%	100,00%
6	123,80%	108,63%	103,93%	100,54%	110,29%	122,18%	150,26%	160,37%
7	468,92%	217,23%	193,88%	174,84%	146,06%	125,74%	110,96%	100,00%
8	133,17%	114,61%	108,56%	103,96%	104,64%	113,57%	130,80%	141,33%
Átlag	202,64%	142,13%	136,46%	132,89%	130,76%	134,38%	144,25%	152,69%

12. táblázat: RVS módszerrel készült programok értékelése

	60	70	75	80	90	100	110	120
1	255,63%	115,12%	103,52%	102,25%	101,54%	105,58%	113,36%	121,25%
2	108,97%	109,95%	101,24%	103,77%	110,18%	117,97%	126,86%	127,18%
4	>500%	121,09%	106,91%	100,00%	102,85%	103,37%	111,04%	121,53%
5	280,72%	197,01%	173,06%	154,93%	129,52%	112,93%	101,75%	100,71%
6	113,27%	105,19%	104,91%	105,96%	106,58%	114,96%	118,82%	129,37%
7	436,44%	243,99%	206,26%	180,06%	145,57%	123,94%	109,52%	100,00%
8	125,86%	108,04%	105,09%	103,82%	100,00%	102,72%	107,24%	108,77%
Átlag	>500%	142,91%	128,71%	121,54%	113,75%	111,64%	112,65%	115,54%

Az egyes táblázatokban szereplő értékeket a könnyebb áttekintés érdekében színskálával láttam el, ahol a kék színtől a sárgán át a pirosig terjedően romlik az egyes programok minősége. Az átlagolásnál minden periódusidőre zölddel jelöltem a három módszer közül a legmegfelelőbbet, azaz a továbbiakban használtat, míg pirossal a másik két, elvetendő változatot.

Így tehát minden csomóponthoz elkészült nyolc darab fázisterv különböző ciklusidőkkel, különböző optimalizálási algoritmusok segítségével (lásd Mellékletek 5).

Mivel a programok megfelelőek, következő lépés a hangolási ábrák elkészítése, és a zöldidő-eltolások, az úgynevezett offsetek meghatározása. Az offset egy rendkívül fontos fogalom a zöldidő-összehangolások területén is. Ez adja meg a zöld jelzések kezdete közötti időkülönbséget, azaz ezzel lehet szabályozni a szomszédos csomópontok programjainak időbeli elhelyezkedését. Ezeknek az értékeknek a pontos meghatározásával alakíthatóak ki olyan hullámok, ahol a járművek megállás nélkül tudnak végighaladni az adott szakaszon.

4.3. Azonos ciklusidejű jelzéstervek összehangolása

Az előző fejezet eredményeiből kiindulva szerettem volna az egyes csomópontokban létrehozott jelzésterveket összehangolni annak érdekében, hogy minél több jármű legyen képes a szakaszon megállás nélkül áthaladni. Ahhoz, hogy ez megvalósulhasson, úgy kellett rendeznem a programok által meghatározott zöldidőket, hogy azok igazodjanak az egyes csomópontok távolságaihoz és a haladási sebességhez. Ehhez volt segítségemre az úgynevezett hangolási ábra, melynek felhasználásával grafikus módon voltak megoldhatóak az előbbieken felsorolt feladatok.

Mivel a fázistervek létrehozásánál minden csomópontban nyolc különböző periódusidejű programot készítettem, értelemszerűen ez nyolc hangolási ábra előállítását jelentette (lásd Mellékletek 6). Először azonban meghatároztam a

csomópontok távolságát a következő táblázat szerint, a hangolási sebességet pedig 50 km/óra-ra állítottam, mivel a szakaszon ennyi a megengedett legnagyobb sebesség.

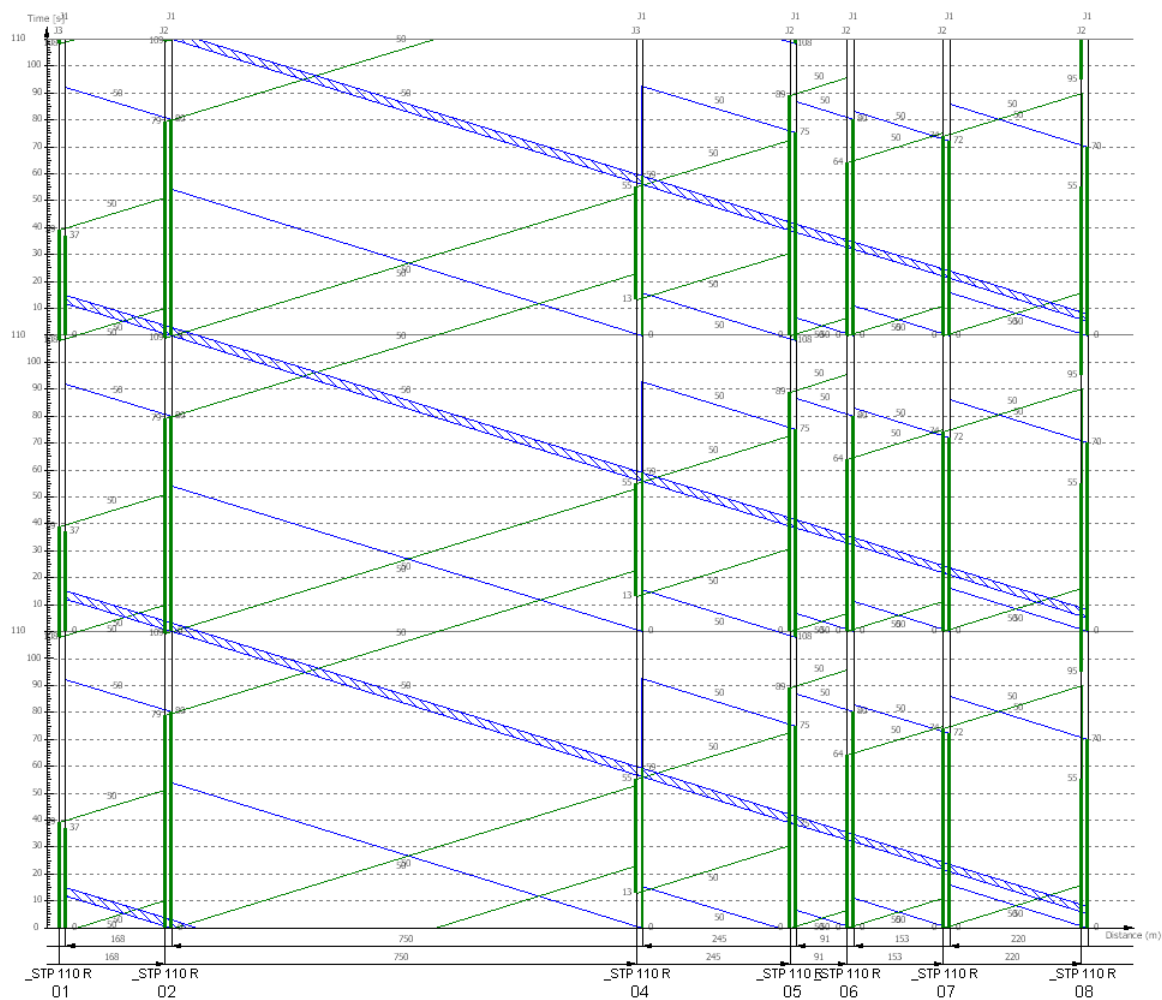
13. táblázat: Jelzőlámpás csomópontok távolsága

Mellékutca	Csomópontok távolsága [m]
Dombóvári út	168
Prielle K. u.	750
Hengermalom út	245
Sztregova köz	91
Újbuda Center	153
Barázda u.	220
Galvani u.	

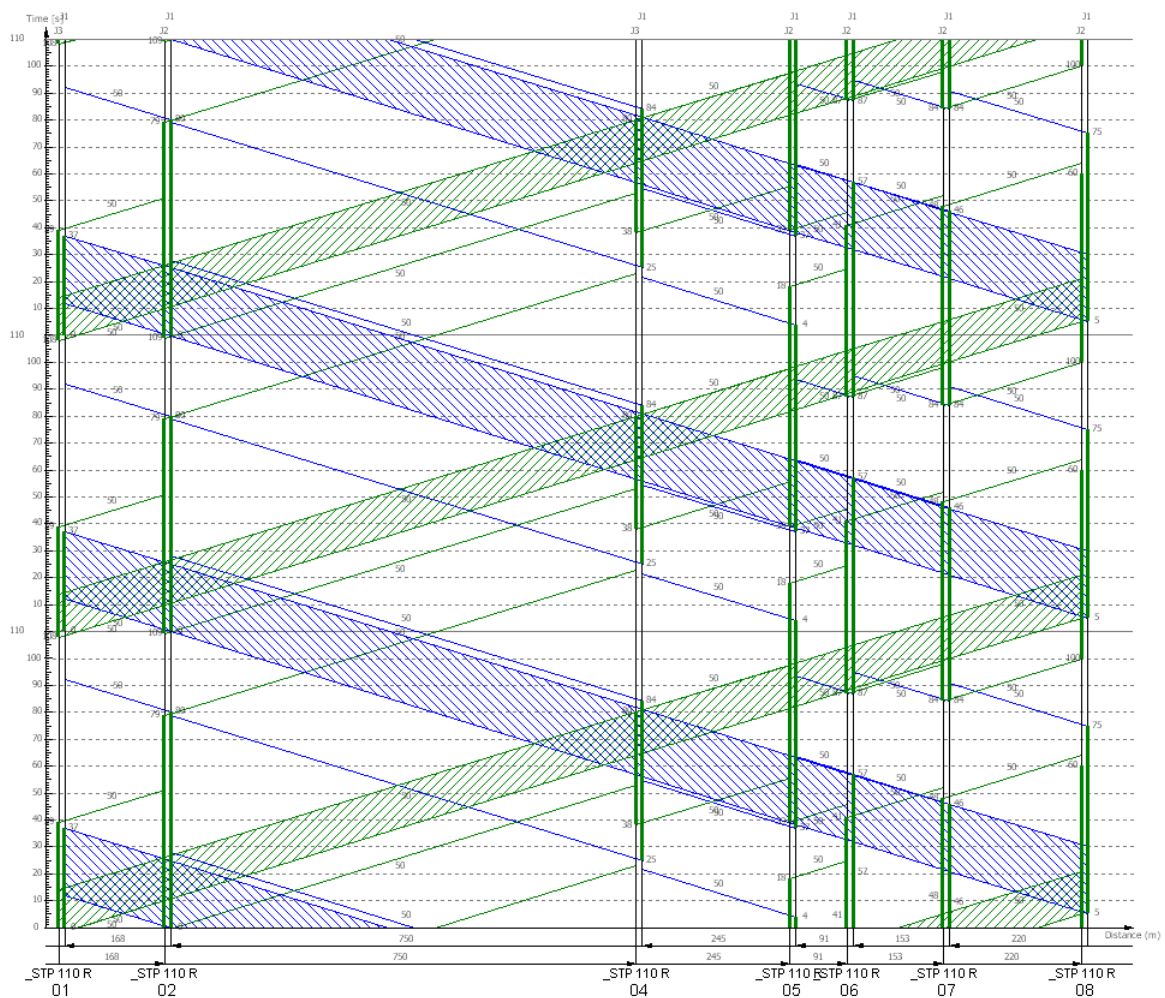
Ezek után az adott ciklusidőt kiválasztva a programmal megjelenítettem a fázisterveket az egyes csomópontokban, egy út-idő diagram segítségével. A fázistervek időbeli (függőleges) eltolásának segítségével létrehozhatunk úgynevezett zöld sávokat, ahol az adott irányban megállás nélkül tudnak haladni a járművek. Ezeknél az eltolásoknál ugyanakkor nagyon fontos figyelni arra, hogy az egy csomópontban lévő két főirány zöldidejét csak ugyanannyival tolhatjuk el, és csak ugyanabba az irányba, mert maga a fázisterv csak így nem fog módosulni. Az adja a feladat szépségét, hogy az eltolások (*offsetek*) segítségével úgy kell kialakítani a fázisterveket, hogy mindkét irányban a lehető legtöbb legyen a megállás nélkül átbocsátható járművek száma.

Első lépésként azoknak a csomópontoknak kellett beállítani az eltolását, ahol a legrövidebbek a zöldidők, azaz ahol a mellékirányok a leghangsúlyosabbak. Ez ebben az esetben a Dombóvári úti és a Hengermalom úti csomópontokat jelentette. Úgy kellett tehát ebben a két csomópontban a zöldidő helyzetét megválasztanom, hogy mindkét irányban létre tudjak hozni egy „zöld sávot”. Mivel a sávok meredeksége (azaz a haladási sebesség) és a csomópontok távolsága is adott, az egyes ciklusidejű hangolási ábrákon elvárható a különböző sáv szélesség kiadódása. Miután a kritikus csomópontokat (Dombóvári út, Hengermalom út, Galvani utca) beállítottam, a köztes helyzetű, nagyobb zöldidővel rendelkező csomópontok (Prielle K. utca, Sztregova köz, Újbuda Center, Barázda utca) fázisterveit az így kialakult sávokhoz rendeztem. Ennek a munkamenetnek a befejeztével létrejött egy olyan sáv, amely jellemzi a megállás nélkül átbocsátható járműveket, mindkét irányban.

Miután elkészítettem egy ilyen hangolási ábrát, leolvastam az eltolásoknak az értékét, és ezeket rögzítettem az egyes programok esetén. Ez az információ innentől kezdve ugyanúgy a fázisterv része, mint minden egyéb a programhoz kapcsolódó adat.



30. ábra: Hangolási ábra a zöldidő eltolások meghatározása előtt (110 másodperces periódusidő esetén)



31. ábra: Hangolási ábra a zöldidő eltolások meghatározása után (110 másodperces periódusidő esetén)

Előfordulhat olyan eset is, amikor az előre eltervezett hangolási feladat nem megoldható egy adott ciklusidő, adott fázisstervek és adott hangolási sebesség esetén. Ilyenkor több lehetőség is rendelkezésre áll, amivel „megoldható” az ilyen jellegű probléma. Az egyszerűbb módszer, hogy az egyik irányban a csomópontot kihagyjuk a hangolásból, így a „zöldhullám” rövidül, ami mindenképpen negatív az átbocsátóképesség szempontjából, ugyanakkor a maradék szakaszon akár jóval szélesebb sáv is létrejöhet, ami kompenzálhatja egy csomópont kiesését. Általánosságban törekedni kell arra, hogy ha valamelyik csomópontot ki kell hagynunk a hangolásból, az lehetőleg a két szélső csomópont egyike legyen.

A másik megoldás egy speciálisabb eset, amely nem alkalmazható akármelyik csomópontban. Ehhez figyelembe kell venni azt a megállapítást, hogy nem csak az egyenes irányok hangolhatók össze, kanyarodó irányok is bekapcsolódhatnak a hangolásba az egyenesek helyett, amennyiben rendelkeznek akkora forgalomnagyság értékkel, hogy értelme legyen a hangolás elkészítésének. Ez a feltétel teljesül a Dombóvári út - Budafoki út csomópont esetében, ugyanis az eredetileg a hangolásba bevont dél felé egyenesen haladó irány (836 EJ/óra), és a kelet felől dél felé balra

kanyarodó irány (679 EJ/óra) forgalma között nem tapasztalható nagy mértékű eltérés. (lásd: 16. ábra: Budafoki út – Dombóvári út csomópont forgalomáramlási ábrája [jármű/óra])

Így megvizsgálhatom annak a lehetőségét, hogy a hangolási ábrában ezt az új irányt tüntessem fel, és amennyiben a zöldidő elhelyezkedése illeszkedik a dél felé haladó forgalmi irány hangolási sávjához, úgy abban az esetben ez az irány bevonható a hangolásba. Még egyszer fontos hangsúlyozni, hogy ehhez a lépéshez olyan mellékirány szükséges, melynek forgalma nem tér el nagy mértékben az eredeti főirány forgalmától.

Amennyiben rendelkezésre áll hangolási ábra, a LISA+ program képes adott forgalmi adatok mellett értékelni az elkészült zöldidő-összehangolást. Természetesen forgalmi adatként a forgalomszámlálás során kapott értékeket használtam, ezen vizsgáltam a hangolás megfelelőségét. A program nem csak az egyes irányokban átengedett járművek mennyiségét adja meg külön-külön, valamint összesen is, de olyan értékeket is ki tud számítani, mint a különböző kibocsátott károsanyagok koncentrációja, vagy az üzemanyag fogyasztása. Ezeket az értékeket összehasonlítva meghatározható az egyes ciklusidejű hangolások között az az irányítás, amely esetében erre az adott forgalomra az átbocsátás a legkedvezőbb, valamint a környezetszennyezés a legkisebb mértékű.

Az egyes hangolásokat az alapján értékeljük, hogy egy ciklus alatt hány járművet tudnak áteresztetni az adott szakaszon megállás nélkül. A táblázatban szereplő értékek a két irány értékeinek átlagai.

14. táblázat: A szakaszon megállás nélkül átbocsátott járművek száma

Periódusidő	Járművek száma/ciklus	Járművek száma/óra
60	7,11	426,60
70	9,07	466,46
75	9,49	455,52
80	9,83	442,35
90	8,77	350,60
100	7,78	280,08
110	10,60	346,75
120	10,07	302,10

Mivel az átengedett járművek száma a ciklusidőtől is függ, ezért egy órára kiterjesztve vizsgáltam a hangolásokat. Ez alapján kijelenthető, hogy a zöldidők összehangolása a 70-80 másodperces intervallumban valósítható meg a leghatékonyabban.

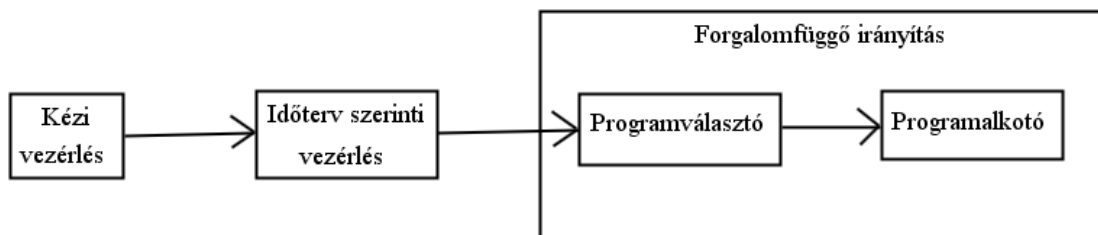
5. Forgalomfüggő irányítás elkészítése

A most következő fejezetben a kétszintű forgalomfüggő irányítás módszertanának leírására kerül sor. A dinamikus irányítás magában foglalja nem csak az összehangolt főirány szakaszát, de a mellékirányban keresztező utcákat is. Ezzel a típusú kétszintű irányítással hatékonyabb forgalomlefolyszt érhetünk el, mint a hagyományos „igény detektoros” módszerrel, ugyanakkor alig merülnek fel plusz költségek ennél az eljárásnál, mivel csak pár extra detektort kell a szakaszra telepíteni. A leginkább költségkímélő, hogy nem kell új gépeket beszerezni az irányításhoz, a kihelyezett forgalomirányító berendezések is képesek elvégezni a szükséges számításokat.

Az első alfejezetben a módszertan általánosságban kerül bemutatásra, majd a második és harmadik alfejezet foglalkozik az irányítás két szintjével. Az első szinten történik meg a mellékirányokban jelentkező igények detektoros vizsgálata, míg a második szinten a főirány forgalomnagyságától függően változik a szakaszon üzemelő fázisidő ciklusideje.

5.1. Kétszintű forgalomfüggő modell bemutatása

A modern közlekedésirányítási koncepciók egyik legfontosabb alapelve, hogy a hálózat koordinálása valós időben legyen megvalósítva, ezt nevezzük dinamikus irányításnak. Ebbe a kategóriába tartozik a forgalomfüggő irányítás is, melynek alapja, hogy a csomópontokban futó programok kialakításánál valós idejű adatokat használunk fel.



32. ábra: Irányítási stratégiák fejlődése

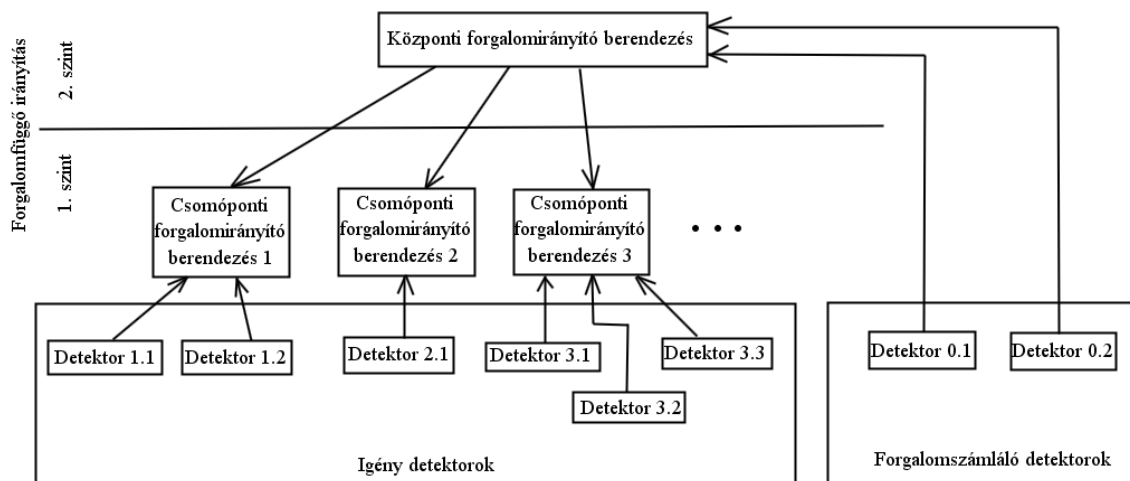
Korábban egy csomópont irányítását úgy oldották meg a szakemberek, hogy az előzőleg mért különböző adatok alapján hozták létre az úgynevezett fix programot, majd ezt küldték ki a csomópontban található forgalomirányító berendezésnek. Természetesen ezt a módszert a mai napig használják a forgalomirányításban, e mellett jelentek meg egyéb, továbbfejlesztett módszerek is. Ebben a fix programos esetben a program ideálisan viselkedik amennyiben a forgalom a mérések szerint alakul, vagy kis mértékben tér el attól, de tudjuk, hogy a valóságban ritkán fordul elő, hogy egy adott

szakasz a nap 24 órájában azonos forgalmi terhelésnek legyen kitéve. Ennek köszönhetően hozták létre azt az irányítási módot, amikor a forgalomirányító berendezésben több programot is lehet tárolni. Ez az irányítás továbbra sem valós idejű, mivel a programok itt is előre lement adatok alapján készülnek, de fejlődést jelent az előzőekhez képest. Ennél a módszernél megvizsgálják a csomópontban jelentkező különböző terhelési állapotokat (reggeli és délutáni csúcsidőszak, napközbeni és esti forgalom, stb.), és ezekhez illeszkedő programokat telepítenek a berendezésbe. Szükség van továbbá egy programváltó órára is, amely az előzőleg meghatározott időpontokban vált az egyes programok között. Látható, hogy ez a módszer egy sokkal rugalmasabb irányítást biztosít mint az előző, ezáltal hatékonyabban is tud működni.

Az igazi újítást azonban az jelentette, hogy a forgalomirányító berendezések képessé váltak adatokat fogadni a hálózatról, és ez alapján optimalizálni a programokat. Az első ilyen módszer a programválasztó logika volt, amely az előző esethez hasonlóan előre betáplált, fix forgalmi adatok alapján elkészített programokat tartalmaz. Ám a váltásokat a korábbiaktól eltérően nem egy időmérő szerkezet biztosítja, hanem az éppen aktuális forgalmi jellemzők. Ezeknek a mérésére különböző módszereket dolgoztak ki (lásd 5.2. fejezet), de Magyarországon leginkább a mágneses hurokdetektorok terjedtek el. Ezek segítségével lehet meghatározni a forgalomnagyságokat és a forgalmi igényeket, majd az elérhető programtárból az ehhez legoptimálisabbat választja ki a számítógép. Ez a módszer ötvözi a statikus és dinamikus irányítási rendszereket, valamint mindenképpen előremutat a dinamikus forgalomirányítás további lehetőségei felé.

Így születhetett meg a legújabb, valós idejű irányítás, a programalkotó forgalomirányító berendezés. Nagyon fontos különbség, hogy ebben az esetben az előre legyártott programok már csak egy alapot biztosítanak az irányításhoz, valamint biztonsági funkciókat látnak el. Fontos eleme itt is a rendszernek a járművek detektálása, legjellemzőbb példa erre egy kis forgalmú sávban elhelyezett úgynevezett „igény detektor”, amely érzékeli, hogy az adott irányból érkezik-e forgalmi igény. Ez azt eredményezi, hogy előfordulhat, hogy bizonyos fázisok egy cikluson belül nem kerülnek kiadásra. Tehát eljutottunk odáig, hogy a berendezés valós időben vizsgálja a csomópont környezetét, és dönt arról, hogy egy meghatározott fázis után melyik lesz a következő. Így akár az is előfordulhat, hogy több egymást követő ciklusban eltérő programok keletkeznek. Természetesen nem csak a fázisok sorrendjének variálásával lehet hatékonyabb irányítást elérni, lehetőség nyílik egyes fázisok nyújtására vagy lerövidítésére is. Nagyon fontos ugyanakkor megjegyezni, hogy ez a programalkotó logika veszélyeket is rejt magában. Mivel „szabadon” variálja az egyes jelzőcsoportok zöldidejét, minden esetben ellenőrizni kell azt, hogy ez nem vezet-e balesetveszélyes szituációhoz. Máshogyan megfogalmazva, a gép csak olyan utasításokat hajthat végre, amelyek nem kockáztatják a közlekedés biztonságosságát. Hogy egy konkrét példát említsek, a berendezésnek minden alkalommal ellenőriznie kell a közbenső idők meglétét az éppen aktuális programban. Hiba észlelése esetén a készülékeket sárga villogó üzemmódra váltásra célszerű programozni.

Az általam tervezett rendszer ennek a fejlődési folyamatnak [9] megfelelően forgalomfüggő, valós idejű irányítást biztosít a meghatározott szakaszra. A címben már szereplő kétszintű irányítási mód arra utal, hogy az előbb ismertetett módszerek közül utóbbi kettőt együttesen használja fel. Tehát egyrészt forgalmi változók, jelen esetben forgalomnagyság értékek alapján választ előzőleg elkészített programok közül, továbbá a csomóponti detektor adatokat is figyelembe veszi.



33. ábra: Kétszintű forgalomfüggő irányítás

Ahhoz, hogy ez a fajta irányítás megvalósítható legyen, szükség van a megfelelő forgalomirányító berendezésekre. A csomóponti forgalomirányító gépek mellett szükség van egy központi gépre is, amely a kommunikációért felelős az egyes kihelyezett berendezések között. A központi gép kapja az adatokat a szakasz elején elhelyezett detektorokból, ezekből számítja ki az éppen aktuális forgalomnagyság értékeket, majd ezen adatok alapján ad utasítást a csomópontokban lévő berendezéseknek a programváltásra. A csomópontokban elhelyezett gépek kezelik az ott kihelyezett detektorok igényadatait, azaz ők döntenek el, hogy az adott mellékirány fázisok kiadásra kerüljenek-e az adott ciklusban. Következzen most az irányítás két szintje, külön-külön részletesebben megvizsgálva.

5.2. *Forgalomfüggő irányítás 1. szint - igények detektoros vizsgálata*

A forgalomfüggő irányítási módszerek lényege, hogy a rendszer valós idejű adatokkal dolgozik. Emiatt szükség van arra, hogy ezeket a bejövő adatokat valamilyen eszközzel detektálni tudjuk. Erre a mérésre többféle módszert is kifejlesztettek az idők során, különböző országokban különböző mérési módszerek terjedtek el. Magyarországon a legnépszerűbb az úgynevezett induktív hurokdetektoros mérőeszköz. A rendszer úgy

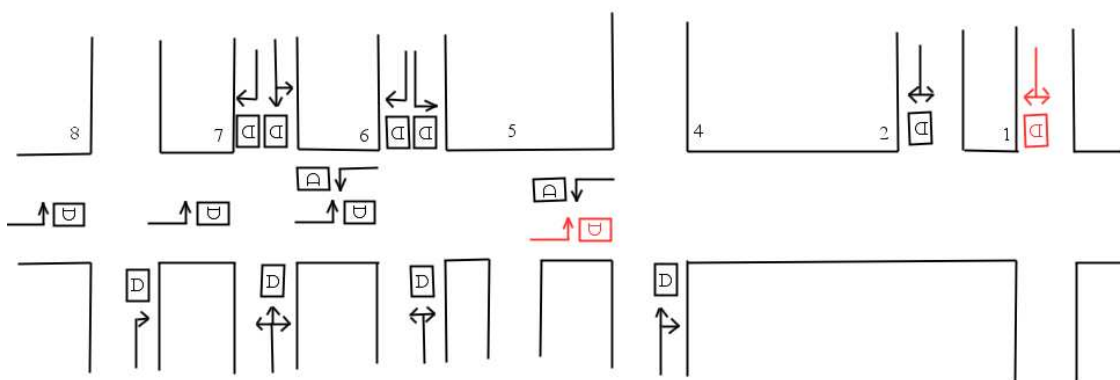
működik, hogy a kívánt helyre téglalap alakban vezetékot helyeznek a burkolatba. A jármű, mint mágnesezhető anyag, ha a detektor közelébe érkezik, elhangolja annak mágneses terét, a berendezés pedig ennek az elhangolódásnak a mértékét vizsgálja. A forgalomirányító berendezés képes érzékelni, a detektor aktuális foglaltsági állapotát, de vizsgálhatjuk, hogy egy időszakban hányszor kapott feszültséget a detektor, vagy akár azt is, hogy mióta van rajta folyamatosan feszültség, azaz mióta áll azon a helyen jármű. A technika előnye, hogy kifejezetten olcsó, valamint nem igényel speciális berendezést. Hátránya, hogy telepítéséhez fel kell bontani a burkolatot, továbbá egyéb útburkolati munkák során könnyen sérülhet.

Léteznek egyéb detektálási eljárások is, melyek nem mágneses elven alapulnak. Ezek közül egy az ultrahangos járműérzékelés, melynek alapja, hogy az úttest felé helyezett készülék ultrahang jeleket bocsát ki magából. A kibocsátott jelek visszaverődését figyelik, melyek akkor torzultak, amennyiben az adott keresztmetszetben jármű tartózkodik. Előnye, hogy viszonylag olcsó, hátránya viszont, hogy meg kell oldani az úttest fölé telepítést, valamint érzékeny a környezeti változásokra is. Egy másik modernebb technika a videós járműérzékelő berendezés, amely egy adott képalkotó egységet foglal magába. Ennek a módszernek mindenképpen előnye, hogy több információt biztosít az előző módszerekhez képest. Hátránya, hogy csak megfelelő látási viszonyok között alkalmazható, valamint a kihelyezett berendezés nagyobb értéket képvisel, mint a korábbiak. További hátránya, hogy telítettséghez közeli forgalom vizsgálatára nem a legalkalmasabb. A következő alkalmazott járműérzékelési módszer a passzív infravörös detektor. Ez az eszköz az út mentén telepítve érzékeli az elhaladó járművek által kibocsátott infravörös sugárzást. Előnye, hogy gyorsan és egyszerűen telepíthető, ezért olyan helyekre, ahol ideiglenesen van szükség mérésre ideális lehet. Hátránya, hogy a különböző zavaró tényezők nagy hatással vannak a berendezés működésére. Végül tekintsük az elsősorban a rendőrség által alkalmazott radar-, vagy lézeres detektorokat. Ezek legnagyobb előnye rendkívül nagy pontosságuk, ám a berendezés költségigénye miatt nem igazán elterjedt a használata.

A feladathoz készített tervezés során induktív hurokdetektorokat alkalmaztam, a hazai mintának megfelelően. A cél ebben az esetben az volt, hogy a kis forgalmú mellékirányú ágakból érzékeljük, van-e igény az adott ciklusban zöld jelzésre. Ennek érdekében igényfelmérő detektorokat helyeztem el az adott sávokban, közvetlenül a stop vonal elé, hogy az első felálló járművet érzékelje a detektor. Egyes csomópontokban a forgalom irányának megoszlása miatt több detektort is elhelyeztem a modellben. Az, hogy melyik mellékirányok kapjanak minden ciklusban zöldet, és melyek függjenek a kihelyezett detektoroktól, a forgalomszámlálás értékei alapján lett meghatározva (lásd 15. ábra). Végül mindössze 3 olyan mellékirány maradt, amely nem lett detektortól függő a nagyobb mértékű forgalma miatt. Ezek a következők: Dombóvári úti ág kelet felől, valamint a Hengermalom úti és a Galvani utcai ág nyugat felől. Az összes többi nem a főirányhoz tartozó ágakban detektorok kerültek elhelyezésre a modellben.

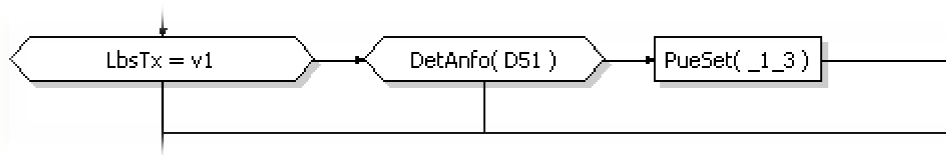
15. táblázat: Hurokdetektorok száma a vizsgált szakaszon

Csomópont száma	Detektorok száma (jelenlegi áll.)	Detektorok száma (tervezett áll.)
1	0	1
2	1	1
4	1	2
5	2	2
6	4	4
7	5	5
8	2	2



34. ábra: Detektorok elhelyezkedése a szakaszon (pirossal a tervezettek)

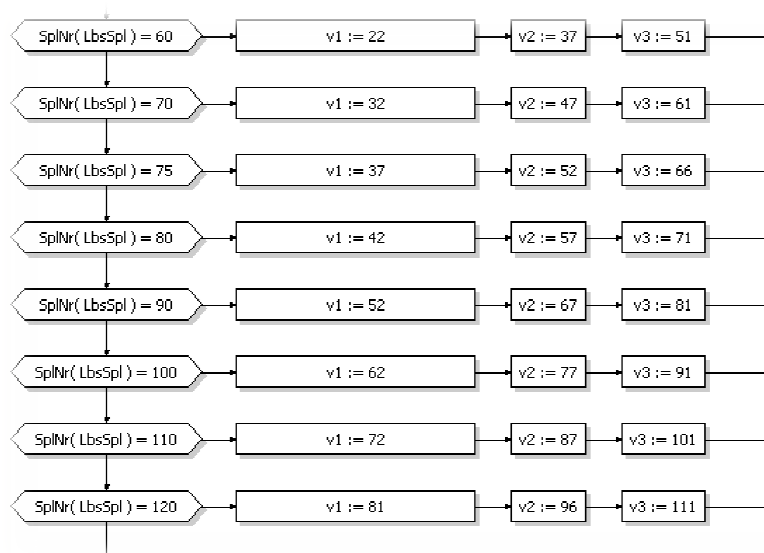
Miután a detektorok üzembe lettek helyezve, a forgalomirányító berendezés minden pillanatban meg tudja ítélni, hogy az adott irányból van-e igény éppen a zöld jelzésre. Ugyanakkor a program nem biztosítja egyből a továbbhaladást az oda érkező járműveknek, a mellékirány zöldjét csak akkor adja ki, amikor az az eredeti fázistervben is megtörténne. Így tehát a detektor szerepe „csupán” annyi, hogy eldönti, hogy az adott fázis szerepeljen-e az adott ciklusban. Ennek az elvnek megfelelően kell felépítenünk azt a forgalomfüggő logikát az egyes csomópontokhoz, amely biztosítja a mellékirány fázisok függését a detektoroktól. Nagyon fontos, hogy mielőtt elérkezünk a kérdéses fázishoz, kérdezzük le, hogy van-e bejelentkezés az adott detektoron. Amennyiben igen, kiadható a berendezésnek a fázisváltás utasítás. Amennyiben nincs igény a detektoron nem kell fázist váltani a főirány továbbra is zöld maradhat. Ha olyan csomópontról van szó, ahol több detektoros mellékirány is van, minden fázisváltás előtt vizsgálni kell az adott detektor állapotát. Fontos megjegyezni, hogy amennyiben nincs igény a detektoron, nem az előző fázis marad változatlanul, hanem a főirány fázisa kapja meg a zöld jelzést.



35. ábra: A detektoron jelentkező igény hatására indul a fázisátmenet

Az ábrán látható módon az első váltási pontban vizsgálja a rendszer az 5-ös csomópont 1-es detektorának foglaltságát. Amennyiben van igény a detektoron, a program indítja a fázisátmenetet az 1-es fázisból a 3-asba.

Azokat a pontokat, ahol a gépnek vizsgálnia kell a detektorok állapotát, az előzőleg elkészített program alapján kell meghatározni. Ezek másodpercben kerülnek rögzítésre egy adott konstans segítségével. A programban, amikor a ciklusidő változója egyenlő a megfelelő konstanssal, ellenőrizni kell a detektor foglaltságát. A konstansok értékének rögzítése egy úgynevezett elő-logika lefutása során történik.



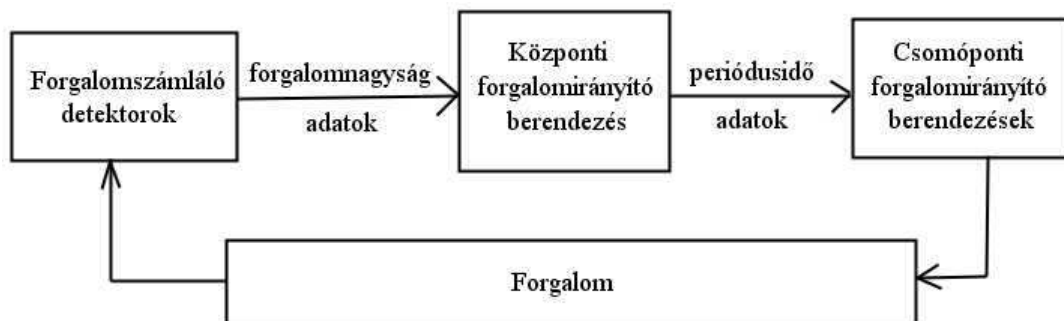
36. ábra: A konstansok értékének megadása az éppen futó programtól függően

Korábban már szóba kerültek a fázisváltások, melyek meghívása a programban a logika alapját képezi. Ezek funkciója, hogy úgy töltsék ki a fázisok közötti időtartamot, hogy a közbensőidő mátrix értékeivel ne legyen ütközés. A LISA+ programban amennyiben definiáltuk a fázisokat, megadhatjuk, hogy melyik fázisból melyikbe szeretnénk váltani, a szoftver pedig létrehoz nekünk egy rövid fázisváltó programot. Ezeknek a megfelelő időben történő meghívásával az eredetileg tervezett fázistervet kaphatjuk vissza.

5.3. *Forgalomfüggő irányítás 2. szint - dinamikus periódusidő változtatás*

A munkamenet korábbi részében a fázisterveket több ciklusidő felhasználásával is elkészítettem (4.2. fejezet). Majd ezek után a hangolási ábrák létrehozásánál (4.3. fejezet) is mind a nyolc felhasznált periódusidő értékre elkészült a tervezett összehangolás. Megfigyelhető, hogy a programok hatékonysága nagyban függ a vizsgált szakasz forgalomnagyságától. Éppen ezért a tervezett rendszernek a leghatékonyabb forgalomlefolysis biztosítása érdekében rugalmasnak kell lennie ilyen tekintetben is. Ezzel eljutottunk a forgalomfüggőség második szintjére, a periódusidők dinamikus változtatásához.

Miután létrehozzuk a hangolási terveket az egyes ciklusidőkhöz, a következő megoldandó feladat az, hogy biztosítsuk ennek a megvalósulását a modellben, valamint létrehozzuk azt az elvet, ami alapján a váltás megtörténik. Az első feladat megoldását egy központi gép létrehozása jelenti, amelyet olyan kapcsolattal kell ellátni, ami biztosítja a kommunikációt a csomópontokba kihelyezett gépekkel. Továbbá a bemenő adatok biztosításához szükség van még sávonként két detektorra a főirány végein elhelyezve, melyek értékei jellemzik a főirány aktuális forgalmi állapotát. Ezen bemenő adatok alapján a központi gép kiküldi az információkat a csomópontba telepített gépeknek az esetleges programváltásokról.

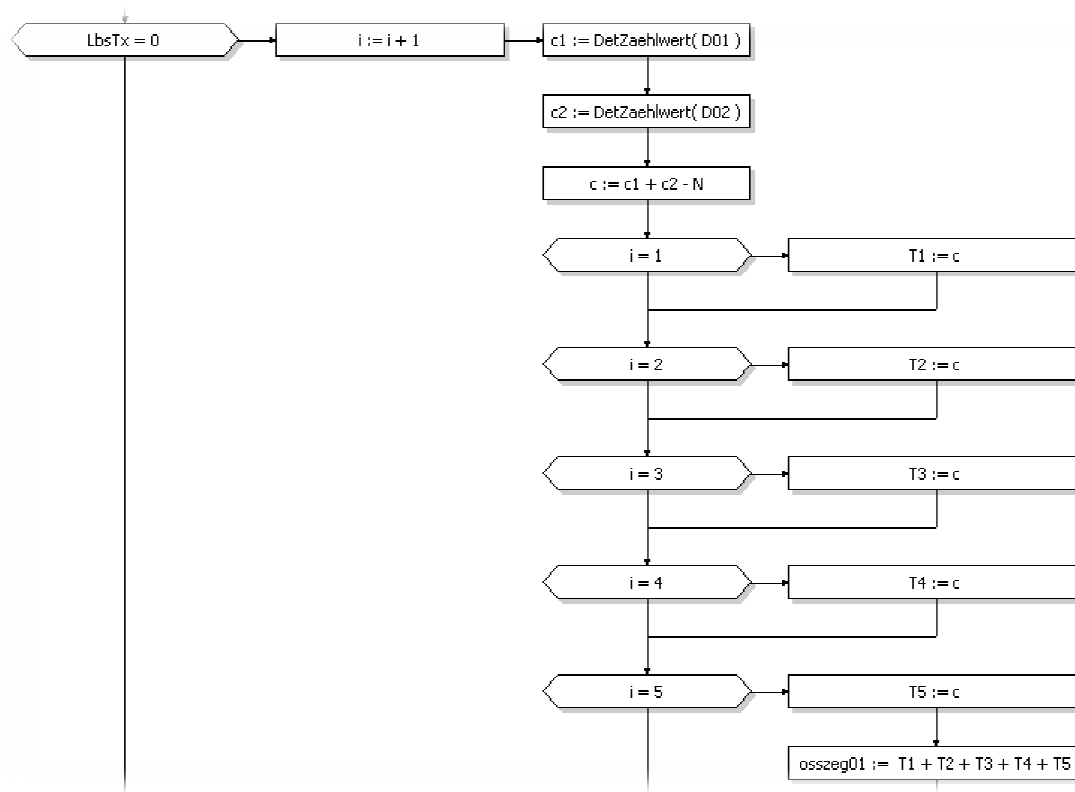


37. ábra: A központi forgalomirányító berendezés működése

A központi gépnek tehát képesnek kell lennie adatokat fogadni, továbbá utasításokat is tudnia kell továbbítani a többi gépnek. Ehhez rendkívül fontos a megfelelő kommunikáció megtervezése és kiépítése. Ezzel bővebben a 6.2. fejezet foglalkozik. A módszer működése szempontjából a következő lépések a meghatározóak:

Irányonként két detektort helyeztem el, mivel a főirány mindenhol két sávval rendelkezik, ezek a detektorok mérik a rajtuk áthaladó járművek számát. Ezt a két értéket összeadva kaptam meg az egy irányból érkező járművek mennyiségét. Ezeket az értékeket ciklusonként egy külön változóba mentettem ki, további felhasználás céljából. Azon elv alapján, hogy a túlságosan gyakori váltások összezavarnák a járművezetőket, úgy határoztam, hogy a rendszer öt ciklusonként avatkozzon be a működésbe. Ez

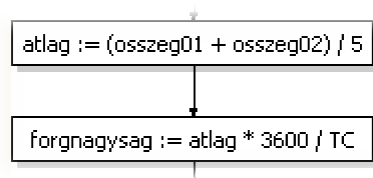
alapján, ha letelt az öt ciklusnyi idő, az eltárolt öt értéket összeadja a program, és elküldi a központi gépnek.



38. ábra: A központi gépnek továbbított detektor adatok előállítása

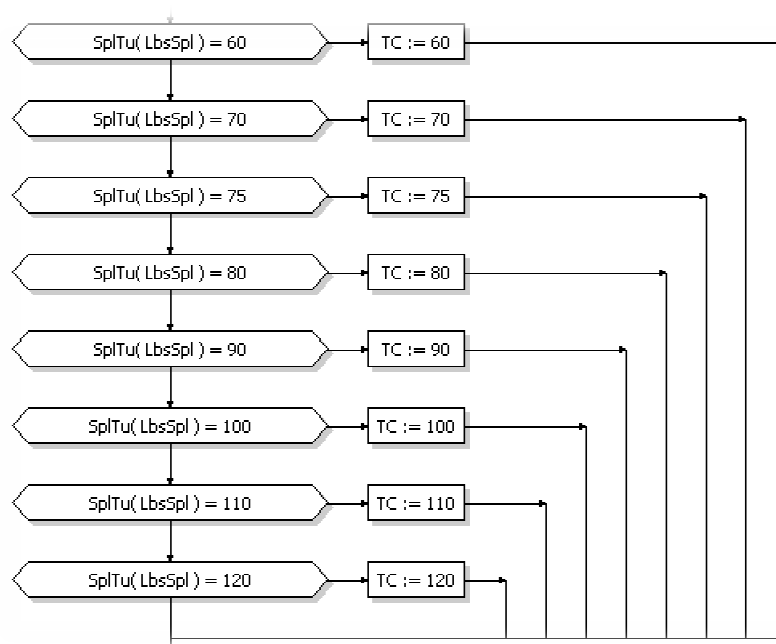
Az ábra alapján a ciklusidő nulladik másodpercében a rendszer $c1$ és $c2$ változóiban tárolja el a detektorokon áthaladt járművek mennyiségét. Ezeket összeadva, és levonva az előző ciklusban kapott mennyiséget, megkapjuk a jelenlegi periódus alatt áthaladt járművek számát. A számláló alapján dől el, hogy ezt a program melyik „T” változóban tárolja. Az ötödik ciklus elérésekor ezeket az értékeket összeadja a program, majd kezdődik a folyamat előlről.

Így a központi berendezéshez két helyről érkezik adat: az öt ciklus alatt számolt járművek mennyisége a két keresztmetszeten. Ezekből kialakít a gép egy átlagos értéket, és a továbbiakban azzal dolgozik tovább. Mivel az a célunk, hogy a ciklusidőt a forgalomnagyság alapján változtassuk, a mért detektor adatokat át kell alakítanunk forgalomnagyság adatokká. Felmerülhet a kérdés, hogy miért nem az adattovábbítás előtt végzi a gép ezeket az átalakításokat, hiszen úgy a központi gép feladata lényegesen leegyszerűsödne. A problémát az jelenti, hogy a VISSIM-ben az adattovábbítás csak *integer* (egész szám) típusú adatokkal jöhet létre, a forgalomnagyság előállítása során viszont a ciklusidővel le kell osztani az összefüggést, ami nem ad garantáltan egész típusú eredményt. [11]



39. ábra: Forgalomnagyság értékének előállítása a központi gépben

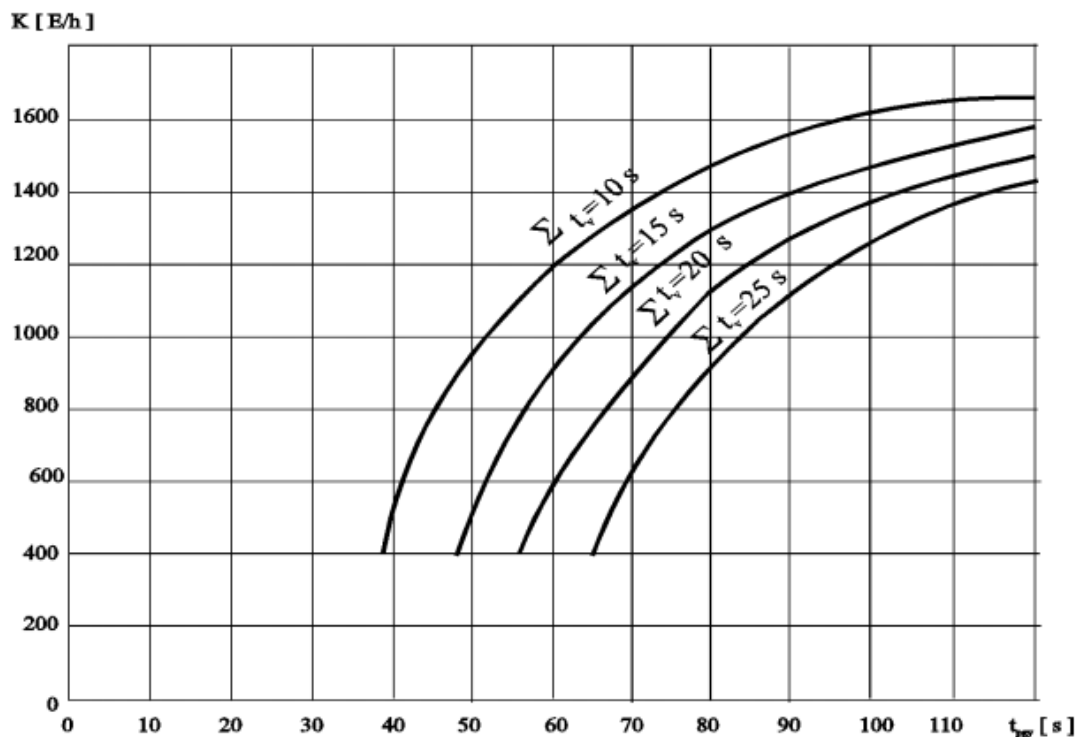
Ebből látható, hogy a központi berendezésnek szüksége van még az éppen aktuálisan futó program ciklusidejére is. Ebben az esetben is felmerült a fent említett probléma, mivel a LISA+ a ciklusidőt *float* típusú változóban tárolja. Ezért létrehoztam egy segédváltozót, amely ugyanezt az adatot tartalmazza, de egész számú formában.



40. ábra: Az aktuálisan futó ciklusidő átalakítása a sikeres kommunikáció céljából

Ennek segítségével már előállítható a forgalomnagyság aktuális értéke, amely alapja a programok kapcsolásának. Két programváltás között eltelt időtartam a ciklusidő ötszöröse, ezért nem állandó időközönként történik meg, hanem a periódusidőtől függően. Miután letelt az öt ciklusnyi idő, a gépek újból küldik a detektorok adatait, a gép újból számítja a forgalomnagyságot, majd dönt arról, hogy milyen periódusidejű új programot kell kiadni.

A döntéshez a kiszámolt forgalomnagyságot vizsgálja a rendszer, forgalomtechnikai összefüggések alapján. Tekintsük az összefüggést a forgalomnagyság és az ahhoz ajánlott ciklusidő között. Ehhez az alábbi grafikon nyújt segítséget:



41. ábra: A kapacitás, a közbenső idő összege és a periódusidő összefüggése
(forrás: [12])

Az egyes görbékhez tartozó t_v értékek a veszteségidőt [9] jelentik egy adott csomópontban. Mivel itt egy egész hálózatra szerettem volna meghatározni a forgalom alapján a ciklusidőt, ezért összes veszteségidőként az egyes csomópontokban lévő veszteségidő értékek átlagát vettem.

16. táblázat: Csomóponti veszteségidők

Csomópont száma	Veszteségidő [mp]
1	12
2	7
4	17
5	9
6	10
7	16
8	7
Átlag	11,143

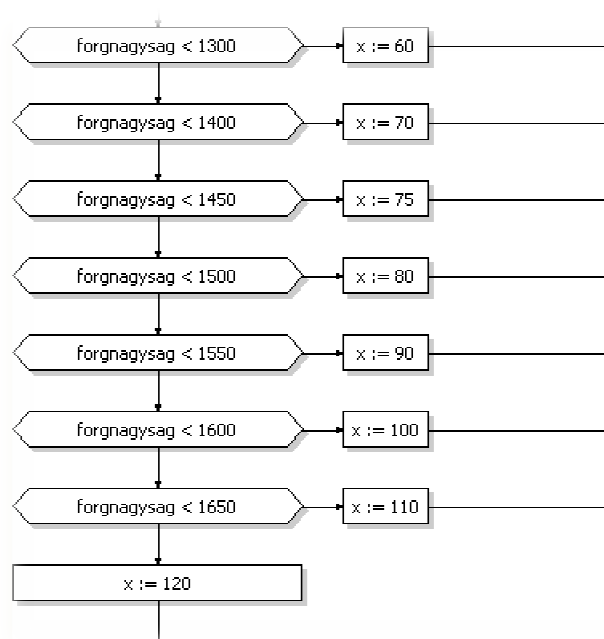
Ezt az átlagos értéket kerekítettem a grafikonban megjelenő értékekhez. Ezek után a grafikon segítségével határoztam meg azokat az intervallumokat, ahol egy adott program bekapcsolásra kerül. Tekintettel voltam továbbá arra is, hogy amíg a grafikon

értékei egysávos utakra vonatkoznak, a vizsgált szakasz kétsávos. Emiatt a központi gép által fogadott adatokat nem csak átlagolni kell, de le is kell osztani kettővel. A grafikon alapján meghatározott intervallumok a következőképpen alakultak:

17. táblázat: A programváltásokat meghatározó forgalomnagyság intervallumok

Forgalomnagyság értéke [jármű/óra]	Az aktuális program ciklusideje [mp]
0-1300	60
1300-1400	70
1400-1450	75
1450-1500	80
1500-1550	90
1550-1600	100
1600-1650	110
1650-	120

Miután a központi gép besorolta az adott forgalomnagyságot a fenti intervallumok egyikébe, meghatározza az ahhoz tartozó ciklusidő értékét. Ezt egy változó segítségével küldi el a szakaszon elhelyezett összes gépnek. Az egyes gépek fogadják ezeket az értékeket, majd a kívánt ciklusidő szerinti programot állítják be a saját csomópontjukban.



42. ábra: Programváltás a forgalomnagyság értéke alapján

Az ábrán feltüntetett „x” értéket küldi a központi a gép a többi berendezésnek.

Ezzel a programváltási folyamat lezajlott, öt ciklus múlva ismét végigfut az algoritmus, és amennyiben a kapott forgalomnagyság érték az előzőtől eltérő intervallumba esik, megtörténik az újabb programváltás.

6. Az elkészített irányítás értékelése, összehasonlítása a jelenlegi állapottal

A fentiekben bemutatott módszer alapján elkészített irányítást a VISSIM szoftverben felépített hálózati modellben teszteltem, ezért az első alfejezet a szoftver bemutatásával és használatával foglalkozik. Az ezt követő alfejezetben azt a folyamatot rögzítettem, ahogy a két használt szoftver közötti adatátvitel megtörtént. Ez egyben egy módszertant is bemutat ilyen jellegű mérnöki feladatok megvalósításához. Végül a harmadik alfejezetben a jelenlegi és a tervezett irányítás modellezésével elkészült szimulációk eredményeit értékeltem ki.

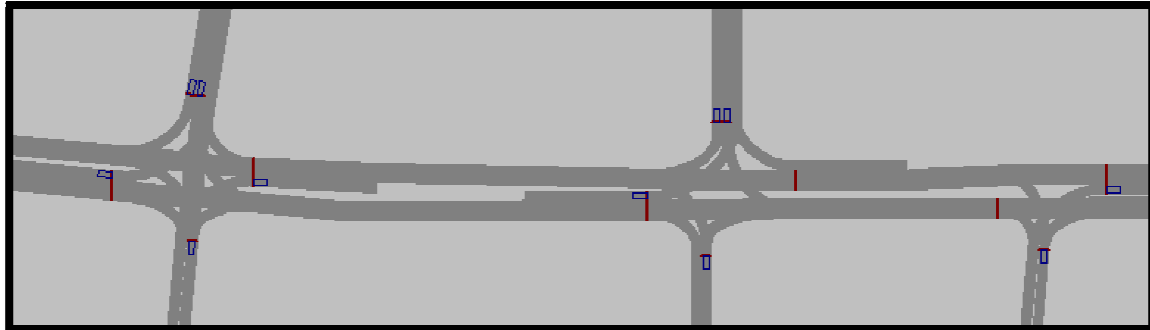
6.1. VISSIM szoftver bemutatása

A VISSIM szoftver [13] a PTV cég által forgalmazott mikroszimulációs forgalommodellező program. Segítségével kisebb hálózatok építhetők fel, majd ezek tesztelésére is lehetőség van, különböző paraméterek lekérdezésével. A program képes háromdimenziós megjelenítésre is, melynek segítségével látványos szimulációk készíthetők a tervezés során.

Első lépésként a kiválasztott hálózatot kellett felépítenem a szoftverben, szakaszokból és a csomópontokban a szakaszokat összekötő kapcsoló elemekből, úgynevezett konnektorokból. A szakaszok esetében beállítható a sávok száma, valamint sok más egyéb paraméter mellett az út típusa is, melynek függvénye a szakaszon engedélyezett maximális sebesség is. Ezek után a csomópontokban definiálnom kellett az irányítási módokat, ahol nincs jelzőlámpa, ott meg kellett határoznom az elsőbbségi viszonyokat. Ez a feladat egyszerűen megoldható a VISSIM-ben, az adott konfliktussal terhelt területre kattintva meghatározhatjuk, hogy melyik iránynak legyen elsőbbsége.

Természetesen a jelzőlámpás irányítású csomópontokban bonyolultabb a modellezés folyamata. Elsőként létre kellett hoznom a forgalomirányító berendezéseket, amelyek jelen esetben forgalomfüggő programokkal rendelkeztek, ezeket a LISA+ programból importáltam. Ezzel a témával részletesebben a következő fejezet foglalkozik (6.2. Kapcsolat a LISA+ és a VISSIM szoftverek között). Miután létrehoztam az irányításért felelős gépeket, elhelyeztem a modellben a jelzőfejeket, majd ezeket hozzárendeltem az adott forgalomirányító berendezés adott jelzőcsoportjához. Ezt a folyamatot a teljes szakaszon, azaz mind a hét csomópontban elvégeztem. A forgalomfüggő irányítás fontos elemei még a detektorok, amiket szintén a tervezésnek megfelelő pozíciókba helyeztem el. A forgalomfüggő program megfelelő működése elsősorban nem a detektorok helyétől függ, hanem annak sorszámától, ugyanis a program ez alapján azonosítja a mérőeszközöket. Forgalomfüggő irányítás létrehozására lehetőség van a VISSIM-ben is, ehhez rendelkezésre áll a COM felület, vagy a VISVAP nevű kiegészítő modul. Ezeknek a kezeléséhez a szoftver mélyebb ismerete szükséges, csakúgy mint a LISA-ban létrehozott logikák VISSIM-be való átültetéséhez. Ebben az

esetben ugyanis egy úgynevezett *external* típusú forgalomirányító berendezést kell létrehozni. Ennek módszertana a 6.2. fejezetben található. Miután elkészítettem a hálózatot, definiálnom kellett az azon közlekedő forgalom paramétereit is.

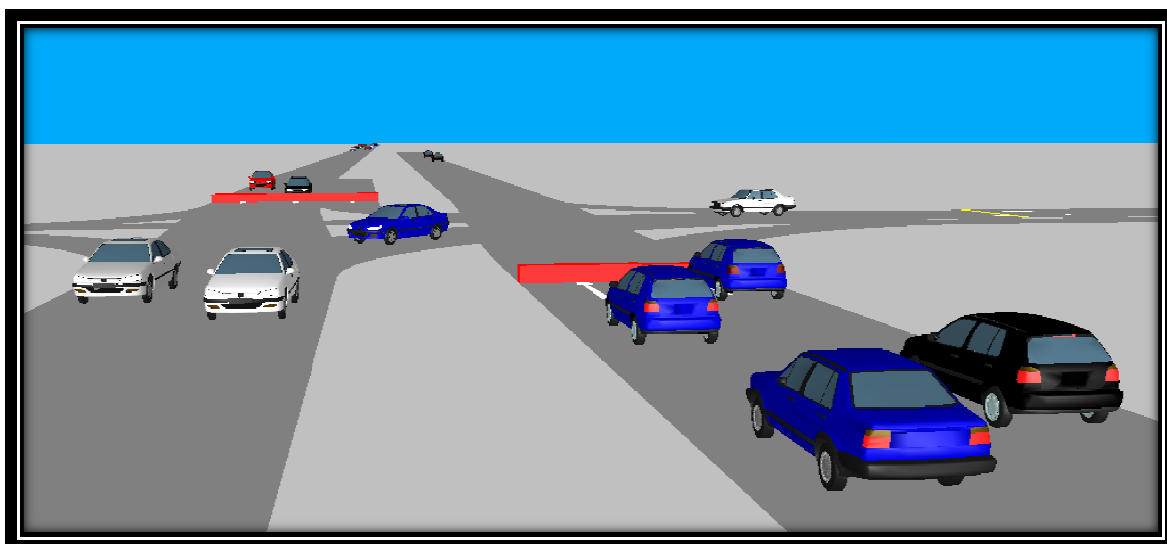


43. ábra: A modellezett szakasz egy részlete a VISSIM szoftverben

Az ábrán bordóval kerültek feltüntetésre az egyes jelzőfejek, kékkel pedig az elhelyezett detektorok. Az ábra a Sztregova köztől a Barázda utcáig mutatja be a Budafoki utat.

A forgalom létrehozásának első lépéseként meg kellett határoznom azokat a pontokat, ahol a hálózatba járművek lépnek be. Ez esetemben értelemszerűen jelentette a főirány két végét, ahol el lett vágva a szakasz, valamint a mellékirányok végeit. Itt tudtam meghatározni a belépő forgalom nagyságát is jármű/óra dimenzióban. Emellett lehetőség volt még a forgalom időbeli változtatására is. Alapesetben a szimulációs időintervallumot felosztva bizonyos időtartamokon definiálható a kezdetitől eltérő forgalomnagyság is. Ez a tulajdonság a tesztelés során rendkívül előnyös, hiszen előállíthatunk új forgalmi állapotokat is, majd megvizsgálhatjuk, hogy hogyan reagál a rendszer a változásokra. Szintén a valóság leképezésére való törekvést mutatja az a beállítás is, miszerint a megadott forgalomnagyságot nem állandó szinten biztosítja a rendszer, hanem sztochasztikus normál eloszlást valósít meg, amennyiben ezt a beállítást választjuk.

Miután létrehoztam a forgalmak kiinduló helyét és nagyságát, meg kellett határoznom az egyes csomópontokban a kanyarodó járművek volumenének megoszlását. Ezt a program úgy valósítja meg, hogy a felhasználó által létesített pontban az adott jármű meghozza a döntését a haladási irányáról, onnantól kezdve pedig igyekszik a lehető legegyszerűbb módon ezt a kijelölt mozgást véghezvinni. Természetesen ezeket a döntési pontokat minden csomópont minden ága előtt el kell helyezni, mivel itt tudjuk bevinni a modellbe a forgalomszámlálás során kapott kanyarodó irányok adatait (14. ábra). Fontos, hogy a pontokat megfelelő távolságban helyezzük el a csomóponttól, hogy a járműveknek legyen elég idejük a döntés és a megvalósítás között. Az elkészült hálózat és a rajta kialakított forgalom így már működőképes egységet alkot.



44. ábra: A modell háromdimenziós megjelenítése a VISSIM szoftverben

Ezek után következhet az értékelés szakasza, mely a szimuláció során gyűjtött adatok elemzésén alapul. Az adatgyűjtés módja alapvetően háromféle lehet a VISSIM-ben. Elhelyezhetünk a hálózatban adatgyűjtő pontokat, amelyek segítségével egy keresztmetszetben mérhetjük a különböző forgalmi paramétereket [9]. Ezeket a program - választásunk szerint - szöveges vagy adatbázis jellegű dokumentumban tárolja el számunkra a szimuláció befejeztével. Emellett mérhetjük a kiválasztott paramétereket egy adott szakaszon is. A harmadik vizsgálati módszer során nem kell megadnunk mérési pontokat, az adatok a teljes hálózatra vonatkoznak. Az olyan hálózati optimalizálási feladatoknál, mint amilyen a jelen feladat is, ezek a kifejezetten sokat mondó adatok.

A VISSIM szoftver természetesen jóval több funkcióval rendelkezik, mint amire most jelen dolgozatban kitértem, igyekeztem azokat a lépéseket bemutatni, amiket a tervezés során felhasználtam.

6.2. Kapcsolat a LISA+ és a VISSIM szoftverek között

A munkám teljességéhez elengedhetetlen volt a Schlothauer & Wauer által kifejlesztett LISA+ szoftver és a PTV AG által forgalmazott VISSIM szoftver összekapcsolása. Mivel ez a feladat korábban már más felhasználóknál is felmerült, a két cég kialakított erre a célra egy követendő metódust. Esetemben az volt a cél, hogy a LISA+ programmal elkészített forgalomfüggő logikák segítségével működő csomóponti szabályozásokat ültessem át a VISSIM-ben létrehozott hálózatra. Meg kell jegyezni, hogy a COM felület segítségével a VISSIM-ben is megoldható a forgalomfüggő detektoros irányítás [14], de tapasztalataim szerint a LISA+ forgalomfüggő irányítást lehetővé tévő környezete egy másik mérnöki szemléletmódot igényel. Ezen okból kifolyólag választottam ezt a típusú megoldási módszert a tervezésem során.

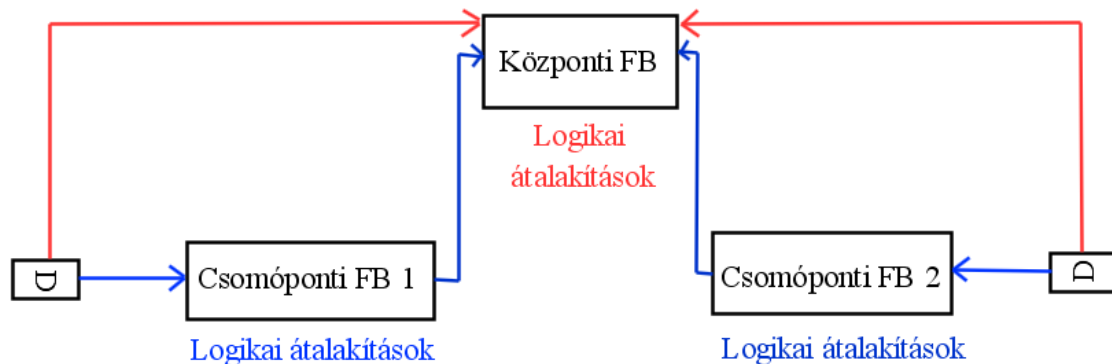
A VISSIM szoftverben amennyiben jelzőlámpás irányítást kívánunk megvalósítani, mindenek előtt a forgalomirányító berendezéseket kell definiálni. Később a lehelyezett jelzőfejeknél rendelhetőek hozzá a berendezés egyes jelzőcsoportjai az adott jelzőfejhez. Ebben az esetben viszont a jelzőcsoportokat nem a VISSIM-ben definiáltam, hanem a LISA+ programban. Miután elkészítettem a LISA+ szoftverben egy csomópont jelzőlámpás irányítását, a programból exportálnom kellett az adatokat a VISSIM számára. Az exportálás előtt szükség van az adott csomópontot irányító gép ellátására egy egyedi azonosítóval, ami alapján a két szoftvert összekötő szerver be tudja olvasni a LISA+ segítségével készült adatokat. Az exportálás során a program többek között létrehoz egy *.jar* kiterjesztésű fájlt is, amely tartalmazza többek között az általam elkészített logikákat is. Ezután visszatérhetünk a VISSIM szoftver környezetébe.

A program alapértelmezettként a forgalomirányító berendezések esetén a fix program elkészítését ajánlja fel. Ebben az esetben viszont ettől eltérő a feladatom. A VISSIM szoftverhez igényelhető egy LISA+ Add-on modul, melynek segítségével a korábban létrehozott fájljaink, azaz csomópontjaink egyszerűen beolvashatóak. Ugyanezt megtehetjük úgy is, ha a fix irányítású berendezés helyett az *external* típusút választjuk. Ettől kezdve a beimportált forgalomirányító berendezés nem mutat különbséget a szoftverben létrehozott gépekhez képest. Ugyanakkor mindenképpen figyelni kell arra, hogy az elnevezések mind a jelzőcsoportoknál, mind a detektoroknál megegyezők legyenek, mivel a program a LISA-ban létrehozott forráskódból dolgozik, azaz eltérő elnevezések esetén a logika nem fog megfelelően lefutni.

További feladatom volt a VISSIM-ben megoldani a forgalomirányító berendezések kommunikációját, hiszen ez képezi az általam kidolgozott rendszer alapját. Ehhez a szoftver több csatornát is biztosít, ahol az egyes gépek kommunikálhatnak egymással. Ezen csatornák megadásánál arra kell tekintettel lenni, hogy egy adott változó egy csatornán érkezzon az adott géphez. Tehát ha két különböző gép küld a központnak forgalmi adatokat egy változó segítségével, a kommunikáció során a cél mindkét gépnél a központi gép ugyanazon csatornája legyen. Fogalmazhatunk úgy is, hogy egy változóhoz egy adott gépnél egy csatorna tartozzon. Ezzel a témával kapcsolatban a VISSIM-ben elvégzendő feladat mindössze ennyi, de a LISA+ programban komoly előkészületek előzik meg a folyamat megvalósítását.

Mint ahogy arról a korábbiakban már szó volt (5.1. fejezet), szükség van egy központi gépre, amely irányítja a csomópontokba kihelyezett többi gép működését. Ehhez szükség van irányonként két detektorra a szakasz bevezető pontjain, melyek adatai képezik a központi gép bemenő értékeit. Ezek alapján egy meghatározott módszer segítségével (lásd 17. táblázat: A programváltásokat meghatározó forgalomnagyság intervallumok) a központi gép meghatározza az optimális ciklusidőt, és ezt küldi ki a csomóponti berendezéseknek. A két forgalomszámláló detektor a főirány mindkét irányából a vizsgált szakasz előtt helyezkedik el. A két detektorból gyűjtött információt két változóban tároltam el, mindegyiket a legközelebbi csomópontához tartozóan. Ugyanakkor az a lehetőség sem elvetendő, amikor a forgalomszámláló detektorok a

központi géphez vannak bekötve. Jelen megoldás előnye a kisebb huzalozási költség, míg a másik esetben a logika lesz egyszerűbb.



45. ábra: A forgalomszámláló detektorok két féle bekötési módja (késsel és pirossal)

Ezek után létrehoztam a LISA-ban egy fiktív csomópontot, és ehhez kötöttem hozzá a központi gépet, amely a teljes szakasz berendezéseit irányítja. Ennél a gépnél is létrehoztam természetesen a két detektor adataihoz szükséges két változót, mivel a központi berendezés ezek alapján adja ki az utasításait a többi gépnek. Nagyon fontos ugyanakkor, hogy a két-két változó mindenben megegyezzen, mivel az exportálás után a VISSIM ezek segítségével hajtja majd végre a kommunikációt. Ahhoz, hogy ez a kommunikáció tényleg megvalósuljon, szükség van még egy azonosító jelzésre a VISSIM számára, ezért ezeket a változókat a 32-es jelzésű csoportra kell állítani. Ezáltal fogja tudni a szoftver, hogy ezek a változók a gépek kommunikációjában játszanak szerepet. Figyelni kell arra is, hogy a változók sorrendje megegyezzen azzal, ahogyan őket a kommunikáció során az egyes csatornákon felhasználjuk. [11]

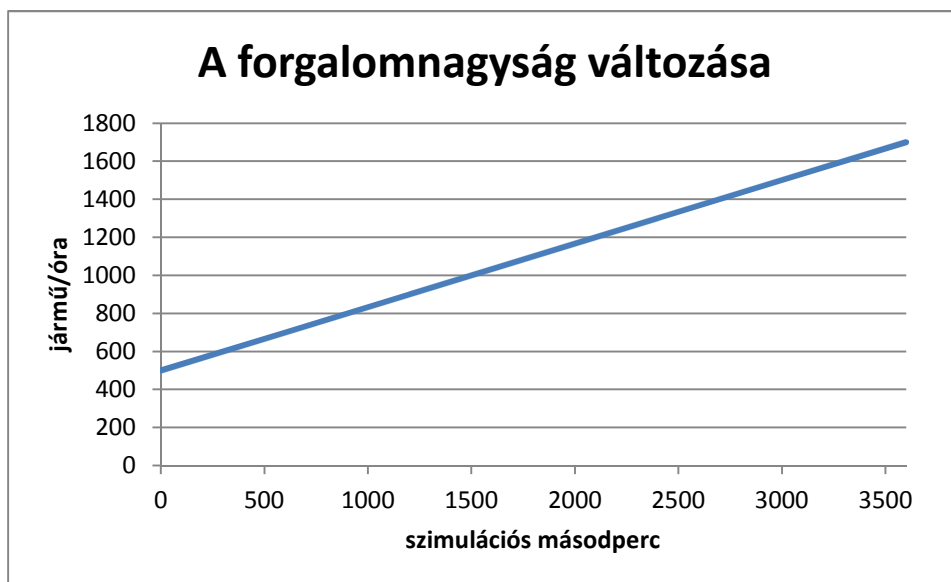
6.3. A tervezett irányítás értékelése

A tervezett irányítási módszer értékeléséhez a VISSIM-ben általam felépített modellt alkalmaztam. A modellezett szakasz mellett annak közvetlen környezetét is bevontam a vizsgálatba, ez alapján az értékelés során nem egy adott szakaszra, hanem az egész hálózatra értelmezett értékeket vizsgáltam. A szimulációt mindegyik esetben többször lefuttattam, majd az így kapott értékeket átlagolva kaptam meg azokat a számokat, amikből a rendszer működésével kapcsolatos következtetéseket le tudtam vonni.

A szimulációk futtatása során hat féle scenárióval dolgoztam, melyek többek között magukban foglalták a jelenleg alkalmazott irányítási állapotot. Emellett természetesen vizsgáltam az általam tervezett állapotot is, különböző forgalmi szituációk esetén.

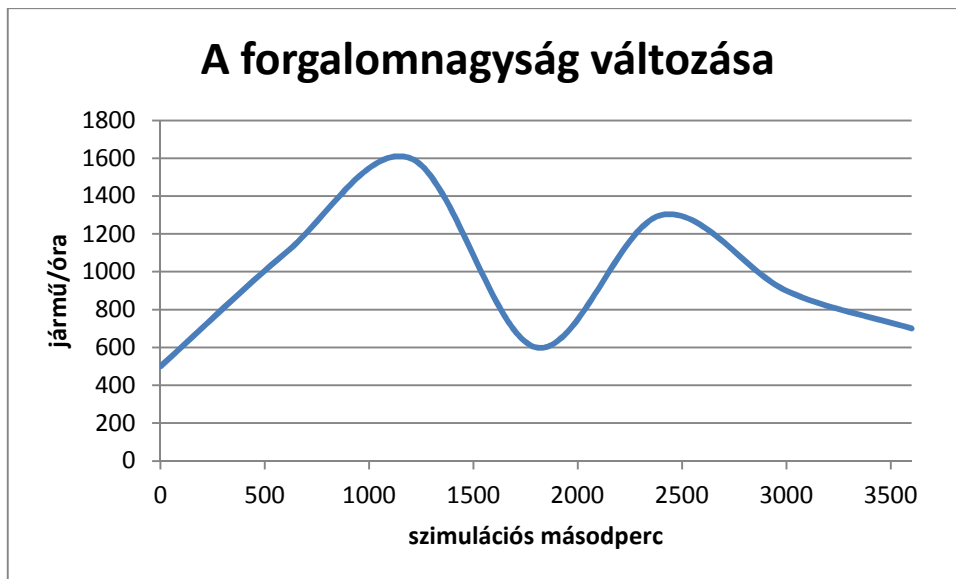
Tekintsük most át a scenáriók összetevőit:

- Jelenlegi irányítási állapot
 - o 90 másodperces periódusidő
 - o Igény detektorok alkalmazása a mellékirányokban
 - o Részben összehangolt irányítás
- Tervezett irányítási állapot
 - o Forgalomnagyság alapján változó periódusidő
 - o Igény detektorok alkalmazása a mellékirányokban
 - o Összehangolt irányítás
- Állandó jellegű forgalom
 - o A forgalomszámlálás adatai alapján kialakított forgalmi állapot
- Emelkedő jellegű forgalom
 - o A főirány forgalma emelkedő jellegű



46. ábra: A főirány forgalmának változása emelkedő jellegű

- o „Csúcsidőszak” modellezése
 - o Nagyobb mértékű terhelés vizsgálata
- Ingadozó jellegű forgalom
 - o A főirány forgalma ingadozó jellegű



47. ábra: A főirány forgalmának változása ingadozó jellegű

o A rendszer érzékenységének vizsgálata

Ezekből az összetevőkből a következő scenáriókat állítottam elő, melyek összehasonlításai adták az értékelés alapját:

- Scenárió 1: Jelenlegi irányítás alkalmazása állandó jellegű forgalom esetén
- Scenárió 2: Tervezett irányítás alkalmazása állandó jellegű forgalom esetén
- Scenárió 3: Jelenlegi irányítás alkalmazása emelkedő jellegű forgalom esetén
- Scenárió 4: Tervezett irányítás alkalmazása emelkedő jellegű forgalom esetén
- Scenárió 5: Jelenlegi irányítás alkalmazása ingadozó jellegű forgalom esetén
- Scenárió 6: Tervezett irányítás alkalmazása ingadozó jellegű forgalom esetén

Az egyes scenáriók összehasonlítására olyan értékeket választottam, amelyek jól alátámasztják egy arteriális irányítás működőképességét. Ezek a következők voltak:

- Átlagos késési idő [másodperc]
- Megállások átlagos száma [db]
- Átlagsebesség [km/h]
- Hálózaton eltöltött idő (összesen) [óra]

A szimulációkat 1 órás intervallumban futtattam le, a változó forgalmú állapotok esetén az értékek 10 perces bontásban változtak.

Elsőként az 1-es és a 2-es számú scenáriót hasonlítottam össze, melyek a mért, állandó jellegű forgalom mellett mutatják be a két irányítás közötti különbséget.

18. táblázat: Az 1-es és 2-es számú szcenárió összehasonlítása

	Irányítás	Forgalom	Átlagos késési idő [s]	Megállások száma átlagosan	Átlagsebesség [km/h]	Hálózaton eltöltött idő (összesen) [h]
SC1	Jelenlegi	Állandó	67,8	1,6	32,4	204,2
SC2	Tervezett	Állandó	61,6	1,5	33,5	200,8
			-10,1%	-4,3%	+3,2%	-1,7%

A százalékos értékek a tervezett állapot bevezetése esetén történő változásokat szemléltetik. Ebben az esetben a legkiemelkedőbb változást a késési idő csökkenésénél tapasztaltam, amely több mint 10 százalékot változott. Ez leginkább a mindkét irányban összehangolt irányításnak köszönhető, aminek következtében a többi paraméter értéke is kedvező irányba változott. Ugyanakkor meg kell jegyezni, hogy ezeknél az állandó értékű forgalomnagysággal történő futtatásoknál a jelenlegi program sem teljesített rosszul, hiszen az eltérések nem kifejezetten nagy mértékűek.

A 3. és 4. sorszámú szcenárió összehasonlításában egy növekvő forgalomnagyságú állapotot teszteltem. Az eredmények a következők lettek:

19. táblázat: A 3-as és a 4-es számú szcenárió összehasonlítása

	Irányítás	Forgalom	Átlagos késési idő [s]	Megállások száma átlagosan	Átlagsebesség [km/h]	Hálózaton eltöltött idő (összesen) [h]
SC3	Jelenlegi	Emelkedő	73,6	1,8	31,3	222,5
SC4	Tervezett	Emelkedő	59,6	1,5	33,9	205,7
			-23,5%	-20,2%	+7,9%	-8,2%

Ebben az esetben emelkedő forgalomnagyság esetén vizsgáltam az egyes paramétereket. Az előző szimulációkhoz képest azt tapasztaltam, hogy jóval nagyobbak az eltérések a jelenlegi és a tervezett irányítási állapot között. A változásokért a dinamikus irányítás a felelős, mely igazodik az emelkedő forgalomnagysághoz. Ahogyan a forgalom nagysága változik, úgy használja fel az irányítás az egyre nagyobb periódusidővel rendelkező programokat. Ennek hatására a késési idők és a megállások száma is több mint 20 százalékkal csökken, az átlagsebesség ugyanakkor majdnem 8 százalékkal megnövekszik.

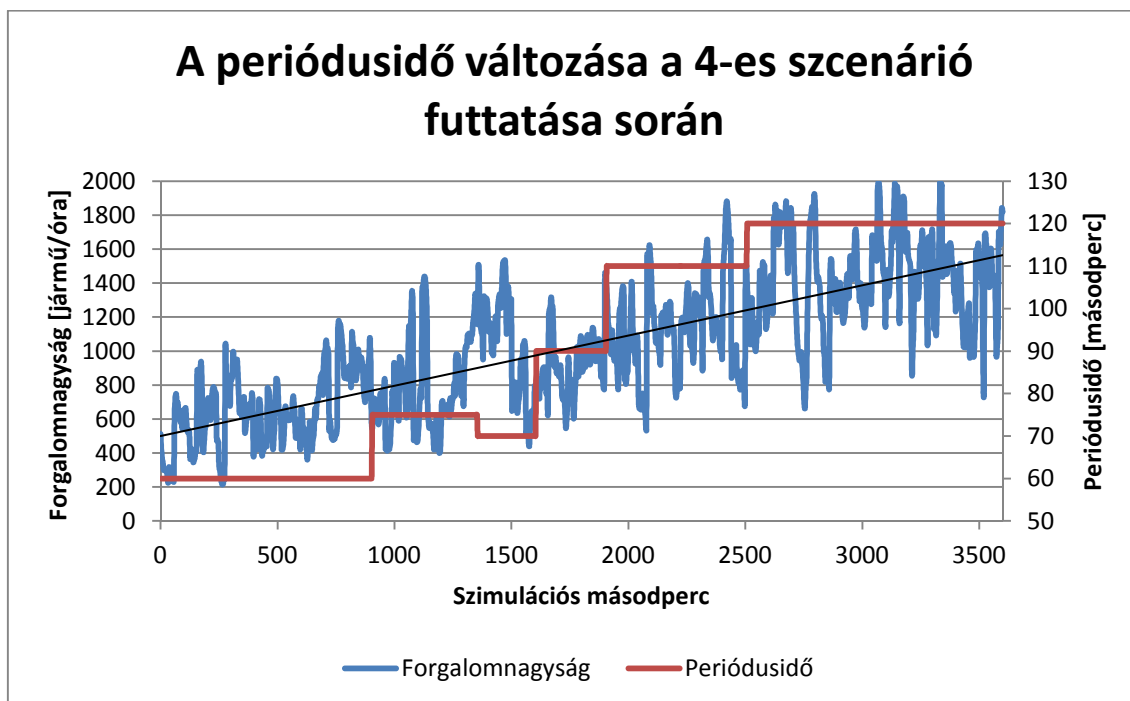
Az 5-ös és 6-os számú szcenárió során azt vizsgáltam, hogyan reagálnak az egyes irányítási módok az ingadozó forgalomra.

20. táblázat: Az 5-ös és 6-os sorszámú scenáriók összehasonlítása

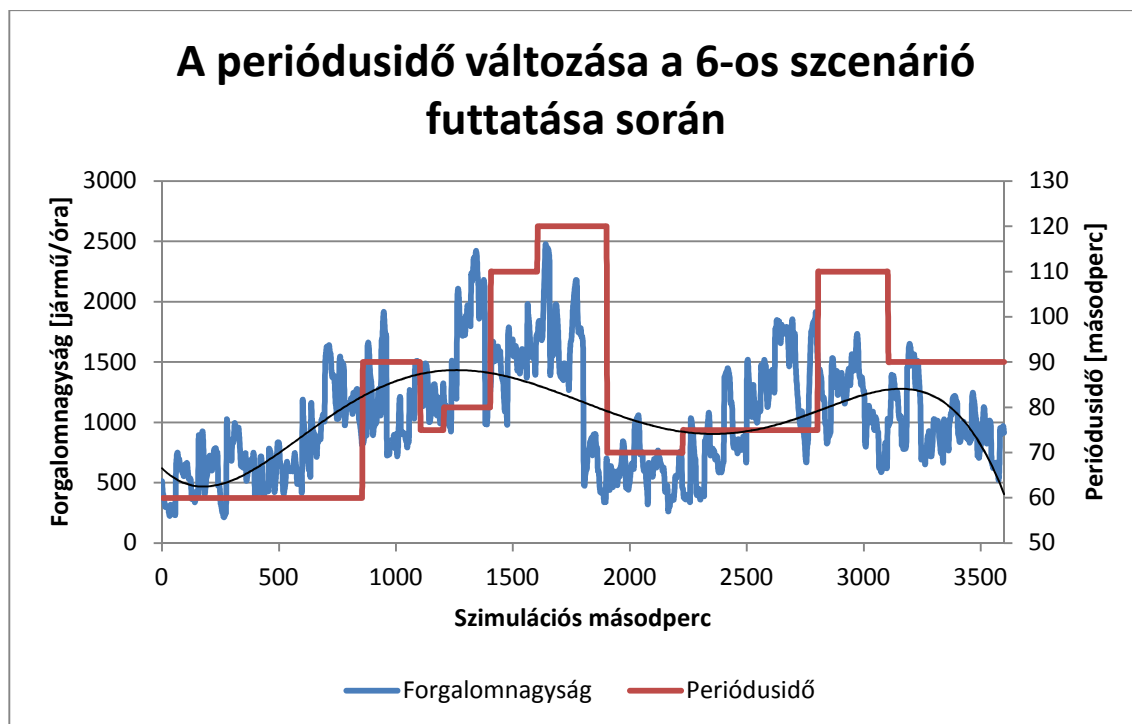
	Irányítás	Forgalom	Átlagos késési idő [s]	Megállások száma átlagosan	Átlagsebesség [km/h]	Hálózaton eltöltött idő (összesen) [h]
SC5	Jelenlegi	Ingadozó	85,7	2,1	29,8	252,1
SC6	Tervezett	Ingadozó	71,7	1,7	32,0	228,5
			-19,6%	-23,7%	+6,9%	-10,3%

Ebben az esetben kijelenthetjük, hogy az előzőekhez hasonlóan a paraméterek itt is a tervezett állapot felé hajlanak, kisebb eltérésekkel az előző két scenárióhoz képest. A késési idők és az átlagsebesség kisebb mértékű eltérése magyarázható azzal, hogy amíg a szimulációban a forgalom nagysága 10 percenként változott meg, addig a valóságban ez jóval több időt vesz igénybe. Az általam kidolgozott irányításban ugyanakkor a ciklusidők váltását úgy ütemeztem, hogy az a valóságos forgalmi változások sűrűségéhez igazolódjon.

Az alábbi két diagramban a 4-es és a 6-os scenárió esetén vizsgáltam, hogy a dinamikus irányítás hatására hogyan változik a programok ciklusideje a forgalomhoz képest. Megállapítható, hogy a ciklusidő – kisebb kilengésektől eltekintve – a forgalom jellegének megfelelően változott.

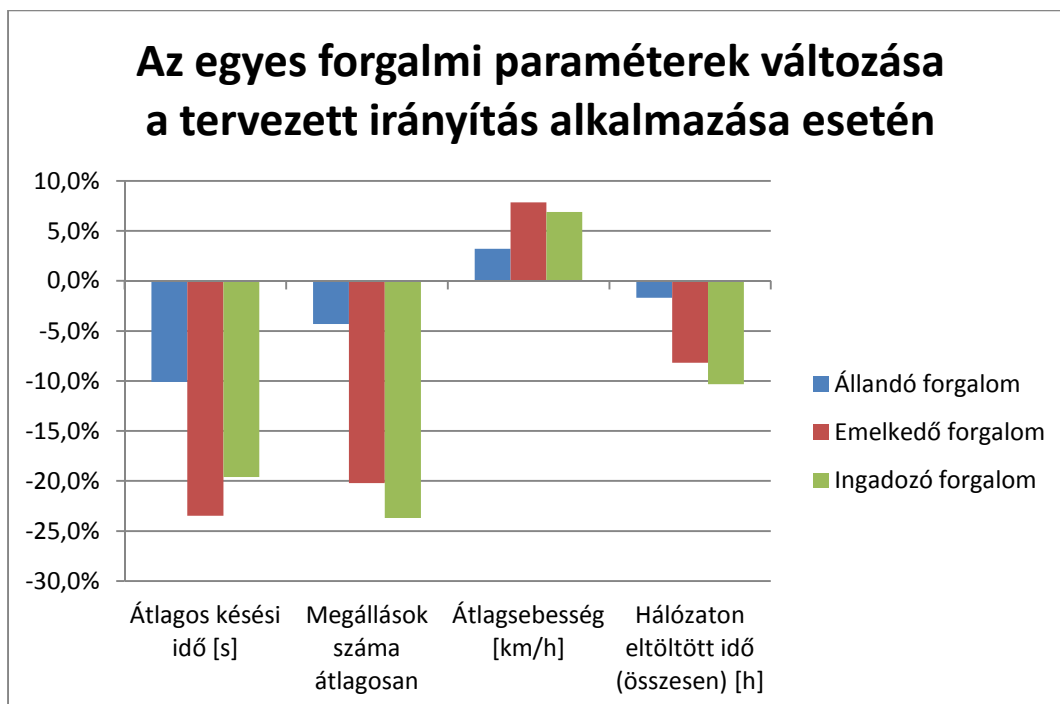


48. ábra: A periódusidő változása a 4-es scenárió futtatása során



49. ábra: A periódusidő változása a 6-os scenárió futtatása során

A következő diagram szemlélteti az eltéréseket az egyes scenáriók esetében. Mindenképpen szembeűnő, hogy a tervezett irányítás változó forgalmi paraméterek esetén hoz jobb eredményeket.



50. ábra: Az egyes forgalmi paraméterek változása a tervezett irányítás alkalmazása esetén

Összességében elmondható, hogy a tervezett irányítási állapot a forgalom változását képes lekövetni egy adott intervallumban, ezáltal hatékonyabb irányítást biztosít, mint a fix ciklusidejű állapot. A zöldidő-összehangolásnak köszönhetően csökken a szakaszon közlekedő járművek megállásainak száma, valamint késési ideje. A kevesebb lassítás és gyorsítás fázis összességében csökkenti a járművek üzemanyag fogyasztását, ami kedvező az egyéni közlekedők számára. A kevesebb megállás ugyanezen okból kifolyólag csökkenti a járművek károsanyag kibocsátását is, ezáltal csökken a szakasz szennyezettsége, a környezet levegője tisztul. Mindezek hatására a vizsgált szakasz jobb közlekedési kapcsolatot, ugyanakkor élhetőbb környezetet is biztosít a környékbeliek számára.

7. Az algoritmus továbbfejlesztésének lehetőségei

Mint a legtöbb mérnöki területen, a közlekedésmérnöki problémák megoldása esetén is mindig lehet az adott fejlesztéseket tovább és tovább javítani, csiszolni. Természetesen ez alól az általam megalkotott és vizsgált módszertan sem kivétel, ahogy azt a szimulációk is igazolták. A következőkben kitérnék pár tapasztalatomra, melyek a szimuláció futtatása közben érték, és amelyeket bizonyos megoldásokkal eszközölni lehet.

Elsőként a csomópontok összehangolásával kapcsolatban figyeltem meg különbségeket a tervezés és a szimuláció fázisa között. Amíg a tervezés során a hangolási ábrára támaszkodva határoztam meg az adott zöldidő eltolás értékeit, ez a szimuláció futtatása során nem igazolta teljes mértékben az elvárásokat. Ugyanis az egyes mellékirányokból érkező járművek számának ingadozásának hatására a két csomópont közti eljutási idő eltérhet az ideálistól, ami alapján a hangolás meg lett tervezve. Ahhoz azonban, hogy a hangolásnál figyelembe tudjuk venni a mellékirányból a főirányra csatlakozó járműveket, és az ezzel járó feltartóztatás mértékét, tudnunk kell ennek az igénynek az aktuális forgalomnagyságát. Mivel ez az érték rövid időszakok elteltével képes nagy mértékben megváltozni, célszerű valós időben mérni.

Ezért a rendszer olyan módon fejleszthető tovább, hogy az egyes csomópontok közti szakaszokon rögzítené a járművek aktuális eljutási idejét, és ez alapján számolná ki a hangolásban alkalmazott zöldidő eltolás értékét. Ehhez az egyes járművek valós idejű adataira van szükség, melynek megvalósítása történhet GPS alapon például járműflotta adatok valós idejű feldolgozása alapján, vagy a járművekben lévő mobiltelefonok cella adataiból kiindulva [15].

A rendszerrel kapcsolatban egy másik tapasztalat is felmerült a szimuláció futtatása során, amely a mágneses hurokdetektorok egy általános problémájára mutat rá. Mivel a rendszer a futó programok ciklusidejét a főirányban elhelyezett detektorok állapota alapján váltja, ezek az eszközök viszonylag fontos szerepet játszanak az irányítás sikeres megvalósulásában. A periódusidő váltása azon alapul, hogy egy adott ciklus alatt hány jármű halad át a detektoron. Kritikus forgalmi állapotok esetén torlódás alakulhat ki azon a szakaszon, ahol a detektort elhelyeztük, tehát a forgalom lassul, a rendszer érzékelése alapján kevesebb jármű halad át adott idő alatt az adott szakaszon. Szélsőséges esetben, például egy baleset miatt az álló forgalom miatt ez az érték nullára is csökkenhet. Ugyanakkor, ha a rendszer kevesebb áthaladó járművet érzékel, csökkenti a szakaszon alkalmazott ciklusidőt, ami azzal jár, hogy rövidül a főirány zöldideje, ezáltal a torlódások lassabban épülnek le.

Ennek elkerülése érdekében érdemes a detektorral nem csak a rajta áthaladó járművek számát figyeltetni, hanem a detektor időbeli foglaltságát is, amiből következtethetünk a járműoszlop haladási sebességére. Ezt a sebességet úgy is megkaphatjuk, hogy két detektort helyezünk egymás mögé, és mérjük a két érzékelés között eltelt időt.

Manapság is igen népszerű irányzat a közlekedésirányítás fejlesztésében a tömegközlekedési eszközök előnyben részesítése. Mivel a vizsgált szakaszon is található olyan autóbusz viszonylat, amely részt vesz a közösségi közlekedésben, érdemes megvizsgálni a tervezett rendszert erről az oldaláról is. Ebben az esetben a legfontosabb módosítást a zöldidő eltolások területén kell elvégeznünk, a szakaszon található megállóhelyek miatt ugyanis figyelembe kell venni a megállóba beállás, utascsere és megállóból elindulás során keletkező idővesztéseket. Ezeknek a változtatásoknak a segítségével a hangolást a buszhoz igazíthatjuk, amely által rövidülhet a menetidő és kevesebb járatra lehet szükség. Dinamikus irányítás esetén megvalósítható olyan hangolási terv készítése is, amely figyelembe veszi a buszok megállóból való indulásakor történő bejelentkezést is. Ugyanakkor meg kell jegyezni, hogy a közösségi közlekedésre optimalizált rendszer hatékonysága redukálódik az egyéni közlekedést használók számára.

8. Összefoglalás

Szakdolgozatom elkészítésének kezdetén egy zöldidő-összehangolás elkészítését terveztem megvalósítani a Budafoki útra. Ennek érdekében az érintett szakaszon forgalomszámlálást végeztem, melynek értékelésekor azt tapasztaltam, hogy a főirány forgalma jó közelítéssel mindenhol 1000 EJ/óra felett alakul, a mellékirányokból pedig pár kivételtől eltekintve egy nagyságrenddel kisebb a forgalom. Ennek hatására a jelenleg is alkalmazott forgalomfüggő irányítást kibővítettem új detektorok használatával, és kiterjesztettem kétszintűre. A kétszintű forgalomfüggő irányításhoz elkészítettem az új jelzésterveket a szakaszon található hét jelzőlámpás csomópont számára, különböző ciklusidőkkel. Elkészítettem ezután az azonos periódusidővel rendelkező programok összehangolását, mivel ezt hiányosságnak fedeztem fel a jelenlegi állapotban. A forgalomfüggő logikák segítségével megvalósítottam a detektorok forgalomfüggőségét, valamint elméletben létrehoztam egy központi forgalomirányító berendezés által működtetett irányítási rendszert is. A rendszer a főirányban elhelyezett detektorok segítségével tudja mérni az adott szakasz forgalomnagyságát, majd ezen érték alapján ad utasítást a csomópontokban elhelyezett forgalomirányító gépek számára. A forgalomnagyság érték, attól függően, hogy milyen intervallumba esik, meghatároz egy kiválasztott periódusidő értéket, melyet a központi rendszer kiküld a csomóponti gépek számára. Ezáltal bizonyos időközönként a szakaszon futó programok ciklusideje megváltozik, a forgalomtól függően.

Az algoritmust különböző forgalmi állapotok mellett teszteltem a létrehozott modellben. A forgalomnagyság értékek alapjai a forgalomszámlálás során felvett járműmennyiségek voltak, elsőként így teszteltem a megtervezett irányítási rendszert. Ebben az esetben, főleg a jelzőlámpák összehangolásának hatására a késési idők 10%-kal lecsökkentek és a szakaszon áthaladó járműveknek kevesebbszer kellett megállniuk, mint a jelenlegi irányítás esetében. A rendszert változó forgalmi állapotok esetén is teszteltem, emelkedő illetve ingadozó forgalomlefolyás mellett. Ez esetben a valós idejű periódusidő szabályozás segítségével nagyobb mértékű javulást értem el. A késési idők és a megállások száma nagyságrendileg 20 százalékkal lecsökkent, míg az átlagsebesség közel 10%-kal nőtt a szakaszon. A szimuláció során az összes jármű hálózaton eltöltött ideje 10 százalékkal lerövidült. Ezeknek az eredményeknek köszönhetően elmondható, hogy a rendszer megfelelő hatékonyságú javulást biztosított a szakaszon, így a rendszer használata kiterjeszthető.

9. Köszönetnyilvánítás

Köszönettel tartozom a szakmai segítségért Tettamanti Tamásnak és Schwanner Norbertnek, valamint mindazoknak, akik segítettek a forgalomszámlálás lebonyolításában.

10. Irodalomjegyzék

- [1] BME Építőmérnöki Kar - Út és Vasútépítési Tanszék: Úthálózatok méretezése és tervezése (TERC Kft., 2012)
- [2] N. H. Gartner: Road traffic control: Progression methods (Traffic & Transportation Systems, Pergamon Press, 1991)
- [3] Denos C. Gazis: Traffic theory (Kluwer Academic Publishers, 2002)
- [4] www.eng.wayne.edu/user_files/151/CE7670_Slides8.ppt (Tapan K. Datta, 2003)
- [5] Isaac Porche - Stéphane Lafortune: Dynamic traffic control: Decentralized and coordinated methods (in Proceedings of the 1997 IEEE Conference on Intelligent Transportation Systems, IEEE)
- [6] Schuchmann – Kisgyörgy: Közlekedéstervezés – Utak – 8. A közúti forgalom (http://www.uvt.bme.hu/targyak/kterv_I/kterv_I_II_kgysch/kterv_I_II_kgysch_8.pdf)
- [7] LISA+ szoftver (www.schlothauer.de/en/LISA.html)
- [8] Tettamanti T.: ACTROS VTC 3000. Elektronikus segédlet. (BME Közlekedésautomatikai Tanszék, Budapest, 2010)(http://www.kjit.bme.hu/images/stories/targyak/kozutir2/actros_vtc_3000.pdf)
- [9] Katkó László, Varga I., Luspay T., Tettamanti T.: Közúti közlekedési automatika. Elektronikus jegyzet. (BME Közlekedésautomatikai Tanszék, Budapest, 2007) (http://www.kjit.bme.hu/images/stories/targyak/kozutir1/kozuti_automatika_ver2.pdf)
- [10] Tettamanti Tamás: Közúti forgalomirányítási stratégiák (http://www.kjit.bme.hu/images/stories/targyak/kozutir1/forgalomiranyitas_ppt.pdf)
- [11] Schlothauer & Wauer: Schnittstelle VIAP LISA+/VISSIM
- [12] Fi István: Forgalomtechnika BSc jegyzet (http://www.epito.bme.hu/uvt/oktatas/feltoltesek/BMEEOUVAI07/forgalomtechnik_a_bsc_jegyzet.pdf)
- [13] VISSIM szoftver (<http://vision-traffic.ptvgroup.com/en-uk/products/ptv-vissim/>)
- [14] Tettamanti T., Varga I.: Közúti forgalomirányító rendszerek fejlesztése és tesztelése VISSIM-MATLAB környezetben. Acta Agraria Kaposváriensis, Volume 15, No. 3., pp. 209-219, 2011)
- [15] Ludvig Á., Tettamanti T., Varga I.: Travel time estimation in urban road traffic networks based on radio signaling data (14th International Conference on Modern Information Technology in the Innovation Processes of Industrial Enterprises, MITIP, 2012, Budapest, ISBN 978-963-311-373-8, pp. 514-527)
- [16] James H. Banks: Introduction to transportation engineering (McGraw-Hill, 2002, ISBN-10: 0072431881)
- [17] PTV Vision: VISSIM User Manual (2012)
- [18] Schlothauer & Wauer: LISA+ Manual (Version 4.0)

11. Ábrajegyzék

1. ábra: A módszer nevét adó osztópont megjelenítése az út-idő diagramban	6
2. ábra: Összefüggés a periódusidő, az összehangolási sebesség és az osztópont távolság között	7
3. ábra: Út-idő diagram a MULTIBAND módszer segítségével	10
4. ábra: A TRANSYT program felépítése	11
5. ábra: Két rétegű irányítás az ALLONS-D módszer esetében	12
6. ábra: A Budafoki út elhelyezkedése Budapesten belül	13
7. ábra: A vizsgált útszakasz	14
8. ábra: Térkép a három párhuzamos útról	15
9. ábra: A pesti oldal megközelíthetősége a Budafoki út irányából	16
10. ábra: Jelzőlámpás irányítású csomópontok elhelyezkedése	17
11. ábra: Forgalmatszámoló lap minta	19
12. ábra: Számolt irányok csomópontonként	20
13. ábra: Forgalomáramlási ábra minta	22
14. ábra: A vizsgált szakasz forgalmi terhelése [jármű/óra]	22
15. ábra: Nagyobb forgalmú mellékirányok	24
16. ábra: Budafoki út – Dombóvári út csomópont forgalomáramlási ábrája [jármű/óra]	24
17. ábra: A LISA+ grafikus tervezőfelülete	26
18. ábra: Konfliktuspontok megjelenítése a LISA+ programban	27
19. ábra: A Barázda utcai csomópont fázisai	28
20. ábra: Jelzésterv minta a LISA-ban (Újbuda Center bekötő út, $T_C = 70$ mp)	29
21. ábra: Forgalmi adatok beviteli felülete a LISA+ szoftverben	30
22. ábra: Szimulációs környezet a LISA+ szoftverben	31
23. ábra: Az összehangolt irányok jelzőcsoportjai, valamint az egyes csomópontok közötti távolság értékek [m]	32
24. ábra: Hangolási ábra (80 másodperces periódusidejű programok)	33
25. ábra: Blokkdiagram a LISA+ programban történő munkafolyamatról	34
26. ábra: Schnabel/Lohse-féle optimalizálás függvénye (Budafoki út – Dombóvári út)	35
27. ábra: HBS típusú optimalizálás függvénye (Budafoki út – Dombóvári út)	35
28. ábra: RVS típusú optimalizálás függvénye (Budafoki út – Dombóvári út)	35
29. ábra: Az optimalizálás folyamata a LISA+ programban	36
30. ábra: Hangolási ábra a zöldidő eltolások meghatározása előtt (110 másodperces periódusidő esetén)	42
31. ábra: Hangolási ábra a zöldidő eltolások meghatározása után (110 másodperces periódusidő esetén)	43
32. ábra: Irányítási stratégiák fejlődése	45
33. ábra: Kétszintű forgalomfüggő irányítás	47
34. ábra: Detektorok elhelyezkedése a szakaszon (pirossal a tervezettek)	49
35. ábra: A detektoron jelentkező igény hatására indul a fázisátmenet	50
36. ábra: A konstansok értékének megadása az éppen futó programtól függően	50

37. ábra: A központi forgalomirányító berendezés működése	51
38. ábra: A központi gépnek továbbított detektor adatok előállítása	52
39. ábra: Forgalomnagyság értékének előállítása a központi gépben	53
40. ábra: Az aktuálisan futó ciklusidő átalakítása a sikeres kommunikáció céljából	53
41. ábra: A kapacitás, a közbenső idő összege és a periódusidő összefüggése.....	54
42. ábra: Programváltás a forgalomnagyság értéke alapján	55
43. ábra: A modellezett szakasz egy részlete a VISSIM szoftverben	58
44. ábra: A modell háromdimenziós megjelenítése a VISSIM szoftverben	59
45. ábra: A forgalomszámláló detektorok két féle bekötési módja (késsel és pirossal).	61
46. ábra: A főirány forgalmának változása emelkedő jellegű.....	62
47. ábra: A főirány forgalmának változása ingadozó jellegű.....	63
48. ábra: A periódusidő változása a 4-es szcenárió futtatása során	65
49. ábra: A periódusidő változása a 6-os szcenárió futtatása során	66
50. ábra: Az egyes forgalmi paraméterek változása a tervezett irányítás alkalmazása esetén	66

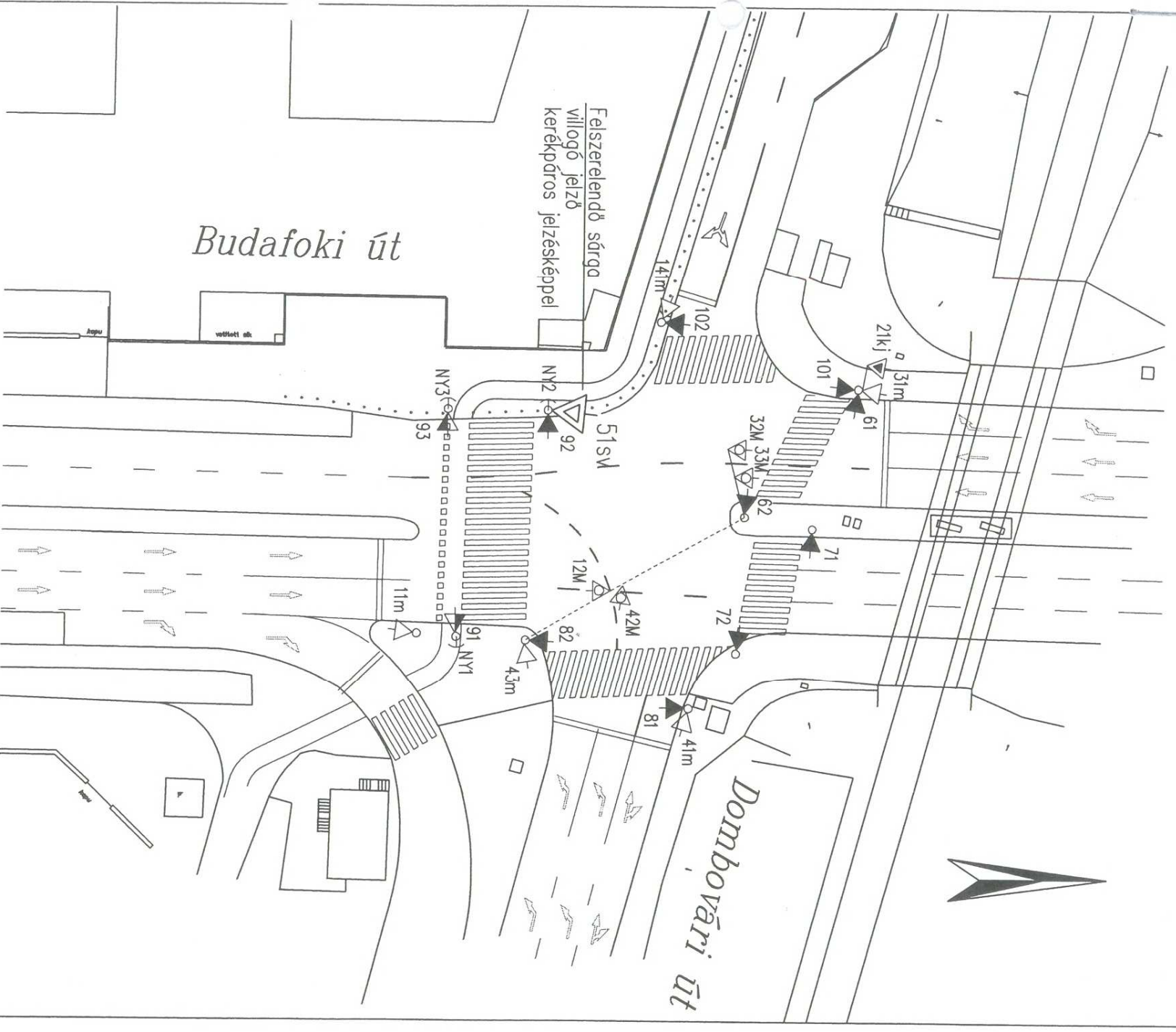
12. Táblázatjegyzék

1. táblázat: A forgalomszámlálás során számlálendő irányok száma	18
2. táblázat: Egységjármű szorzók kül- és belterületen.....	20
3. táblázat: A Budafoki út forgalma.....	23
4. táblázat: A késési idők átlaga csomópontonként a Schnabel/Lohse optimalizálási módszerrel.....	37
5. táblázat: A késési idők átlaga csomópontonként a HBS optimalizálási módszerrel ..	37
6. táblázat: A késési idők átlaga csomópontonként az RVS optimalizálási módszerrel.	37
7. táblázat: A minimális várakozási időhöz tartozó optimális periódusidő értékek (Schnabel/Lohse módszerrel)	38
8. táblázat: A minimális várakozási időhöz tartozó optimális periódusidő értékek (HBS módszerrel)	38
9. táblázat: A minimális várakozási időhöz tartozó optimális periódusidő értékek (RVS módszerrel)	38
10. táblázat: Schnabel/Lohse módszerrel készült programok értékelése.....	39
11. táblázat: HBS módszerrel készült programok értékelése	39
12. táblázat: RVS módszerrel készült programok értékelése	40
13. táblázat: Jelzőlámpás csomópontok távolsága.....	41
14. táblázat: A szakaszon megállás nélkül átbocsátott járművek száma	44
15. táblázat: Hurokdetektorok száma a vizsgált szakaszon	49
16. táblázat: Csomóponti veszteségidők	54
17. táblázat: A programváltásokat meghatározó forgalomnagyság intervallumok	55
18. táblázat: Az 1-es és 2-es számú szcenárió összehasonlítása	64
19. táblázat: A 3-as és a 4-es számú szcenárió összehasonlítása.....	64
20. táblázat: Az 5-ös és 6-os sorszámú szcenáriók összehasonlítása	65

13. Mellékletek

- Csomóponti helyszínrajzok
- Jelenlegi fázistervek
- Közbensőidő mátrixok
- Forgalomáramlási ábrák
- Tervezett fázistervek
- Hangolási ábrák (tervezett + jelenlegi)

Melléklet 1: Csomóponti helyszínrajzok



Csomópont:

Budafoki út – Dombóvári út

Csp. szám:

905

Dátum:

2008.07.

Megnevezés:

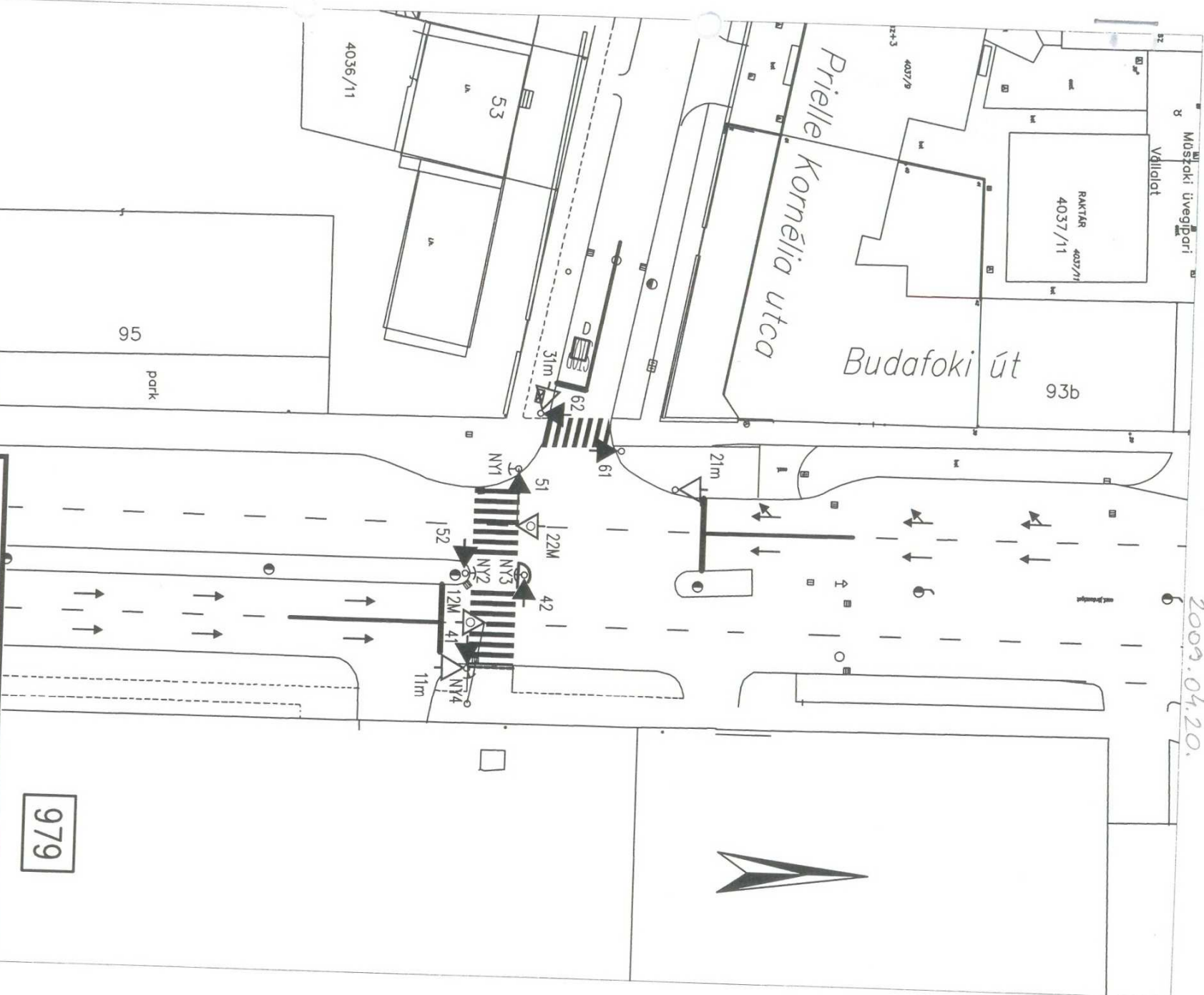
jelzőlámpa-számozási helyszínrajz

Méretarány:

1:500

C84CV

2009.04.20.



979

DATA RECORD KFT. 2132 Göd, Lenkey u. 26.

Megbízó: AIG/Lincoln Építési Kft.

Tárgy: **Budapest XI. ker. Budafoki út – Prielle Kornélia u.**

jezdőlámpás irányítósa – csomópont

Rajz címe: **Jeledőlámpás elhelyezésvázlat**

Tervező:

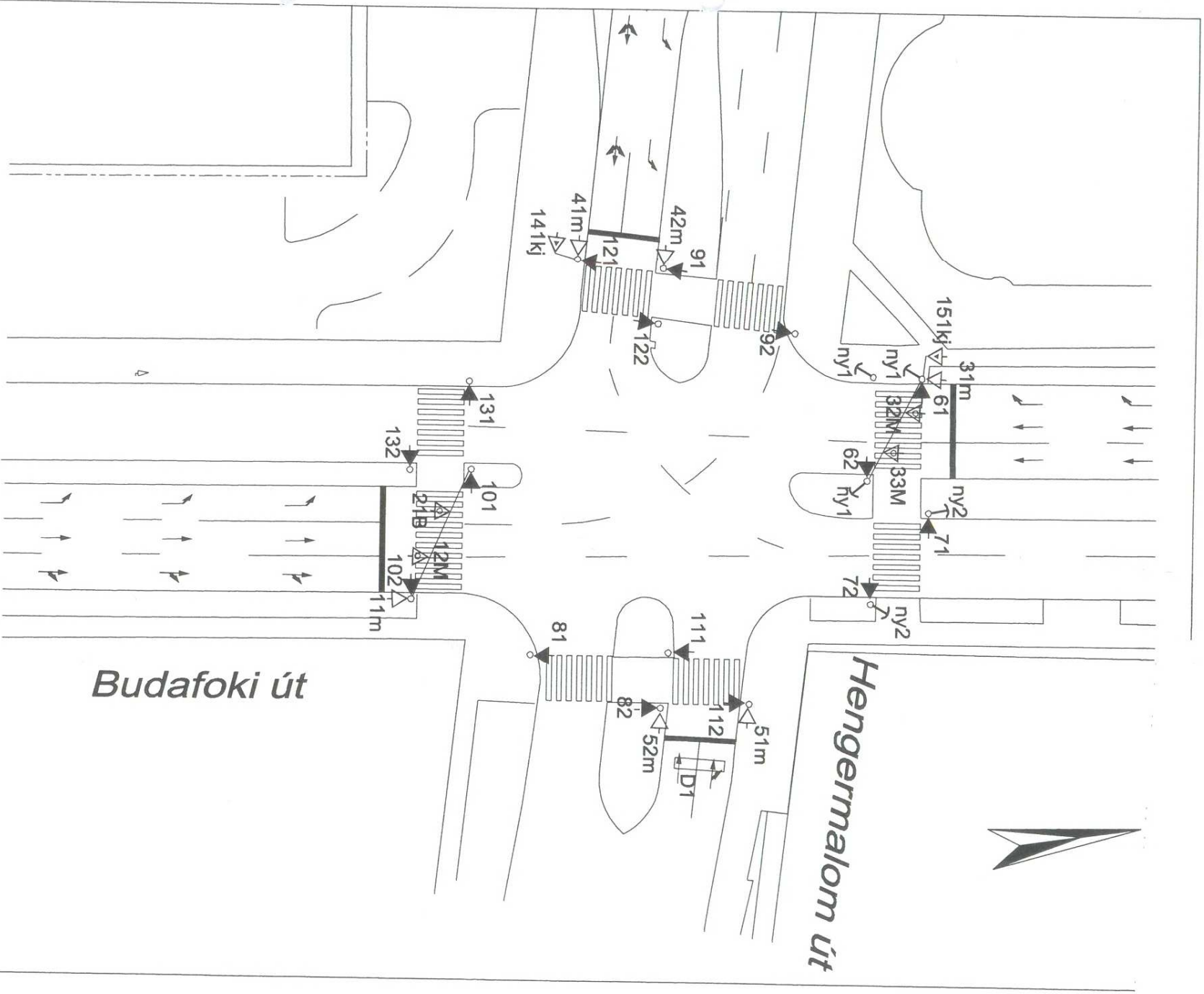
Huszár Dezső 01-7830

Dátum: 2008. aug.

Tervező: 659/08

Rajzszám: F-3

Méretarány: $M = 1 : 500$



A csomópont neve:

Budafoki út - Hengermalom út

A rajz megnevezése:

Jelzőszámozási helyszínrajz

Csomópontszám:

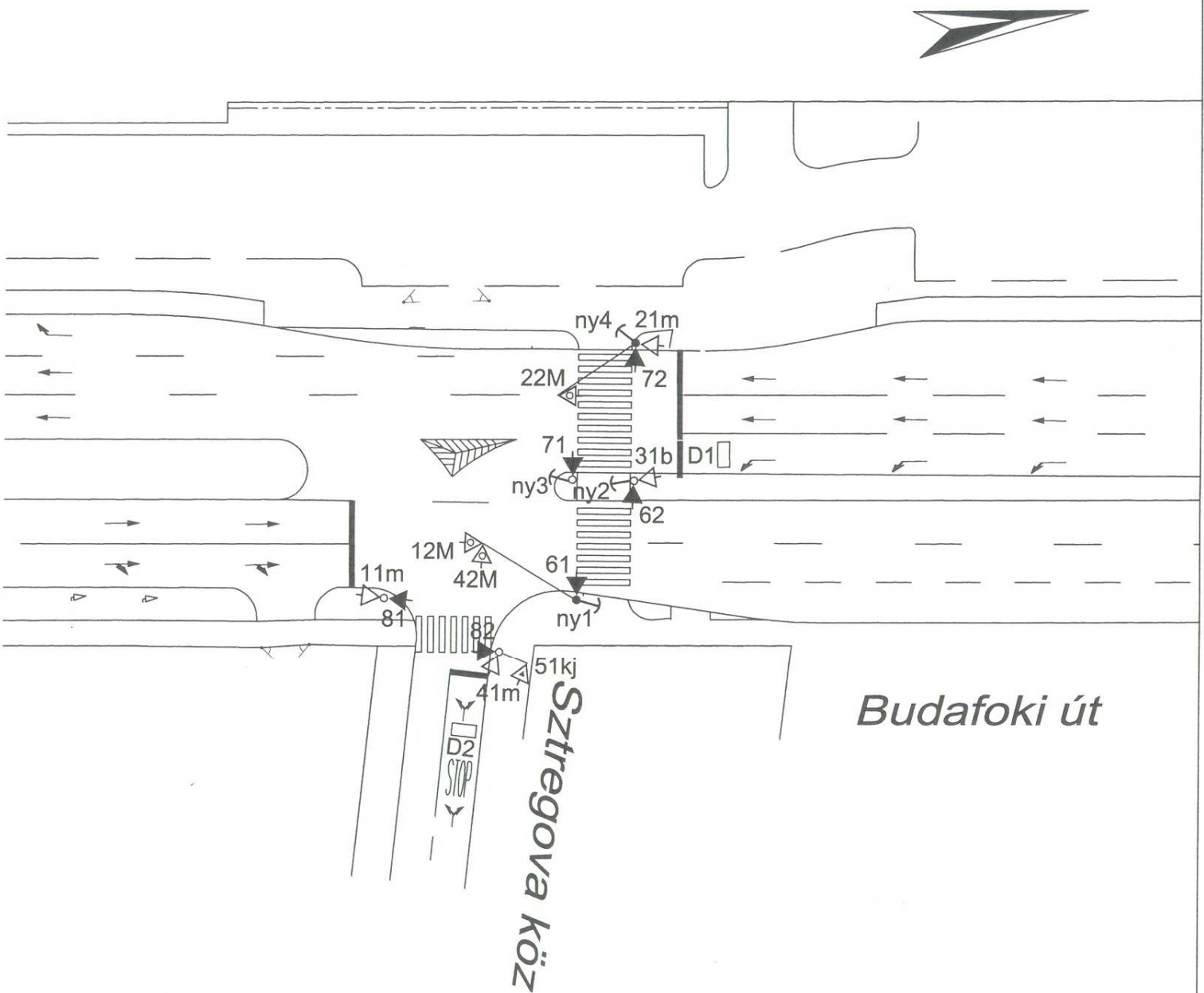
919

Dátum:

2006.07.

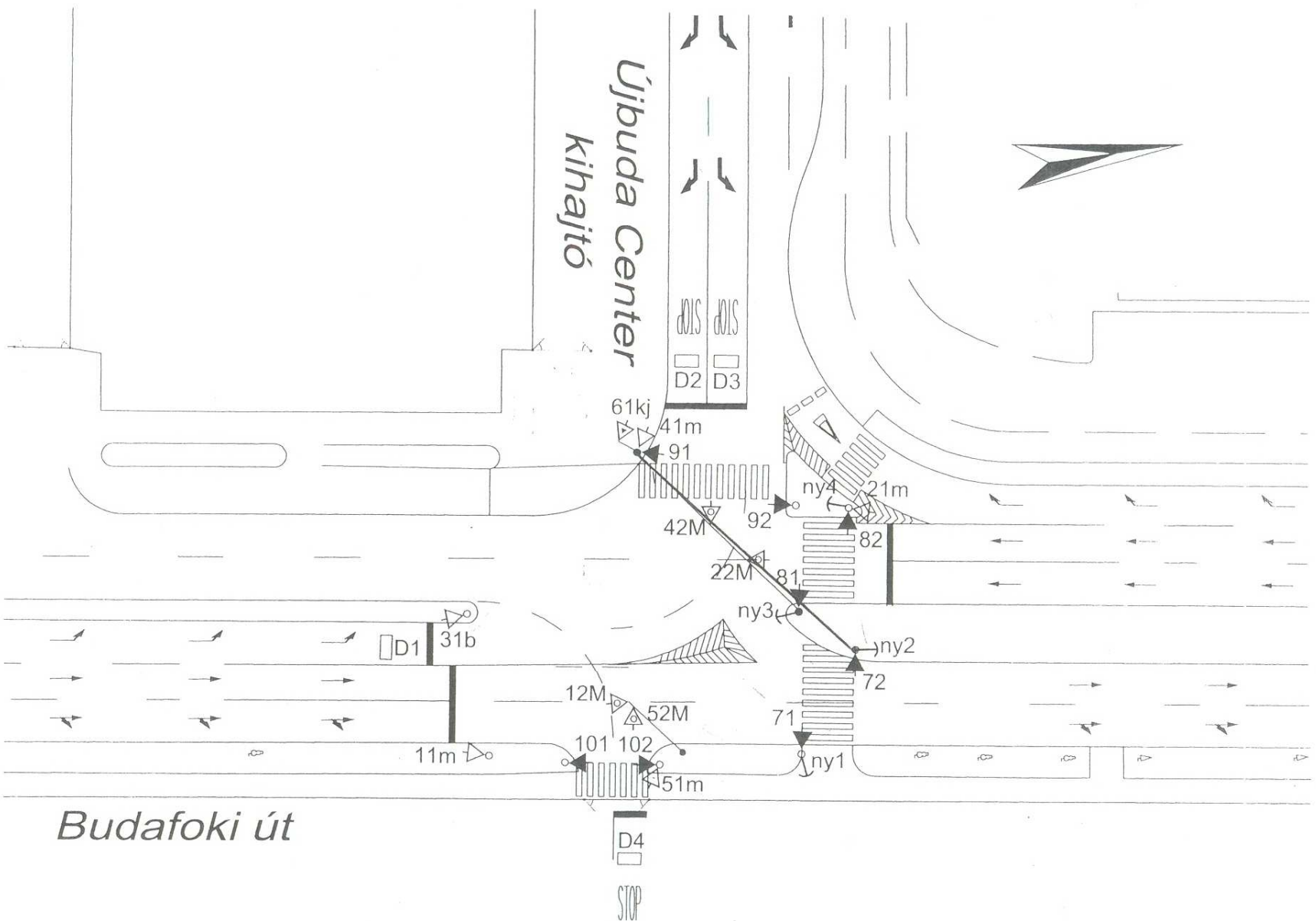
Méretarány:

1:500



981 Budafoki út - Sztregova köz
jelzőszámozási helyszínrajz
Ideiglenes forgalombahelyzés

Dátum: 2007. 11.



ÚT-TESZT
Kft.

CSONTOPONT NÉVE:

Bp. XI. Budafoki út - Újbuda Center kihajtó

Sztrégova köz csp. módosított kialakítása

jelzőlámpa számozási helyszínrajz

CSONTOPONT SZÁM:

982

DATUM:

2006.07.

TERVEZŐ:

Huszár Imre

MUNKASZÁM:

497

MÉRÉTFÁNY:

1:500

RÁJZSZÁM:

SZ-08/M

Budafoki ut 185

Budafoki ut 183

Barázdás utca

aszf.

kapu

D1

D2

D3

D4

D5

D6

Ny1

Ny2

Ny3

Ny4

31b

54m

52m

12m

122m

62m

61m

71m

81k

112

111

22m

41b

sovény

aszf.

aszf.

aszf.

aszf.

kapu

kapu

Portásfülke

Tervezett sorompó

A tervezett sorompó, a D6 és a D5 között, a Barázdás utcán.

ROADPLAN
Tervező, Kivitelező és Tanácsadó KFT.
1048 BUDAPEST IV. ker. KÖRÖSBÁNYA U. 15.

Kábelgyár, Barázda utca, Szerémi út, Hengermalom út által határolt terület

külső közúti kapcsolatait biztosító fejlesztések közlekedési terveinek
kiveteli terve

Tervszám: 3344

Csombpont never!

BUDAFOKI ÚT – BARÁZDA U.

Munka never!

Rajszám: SZ-01

Megbízói

Tervező:

Huszár Imre
01-9074

Đã turn:

2008. november

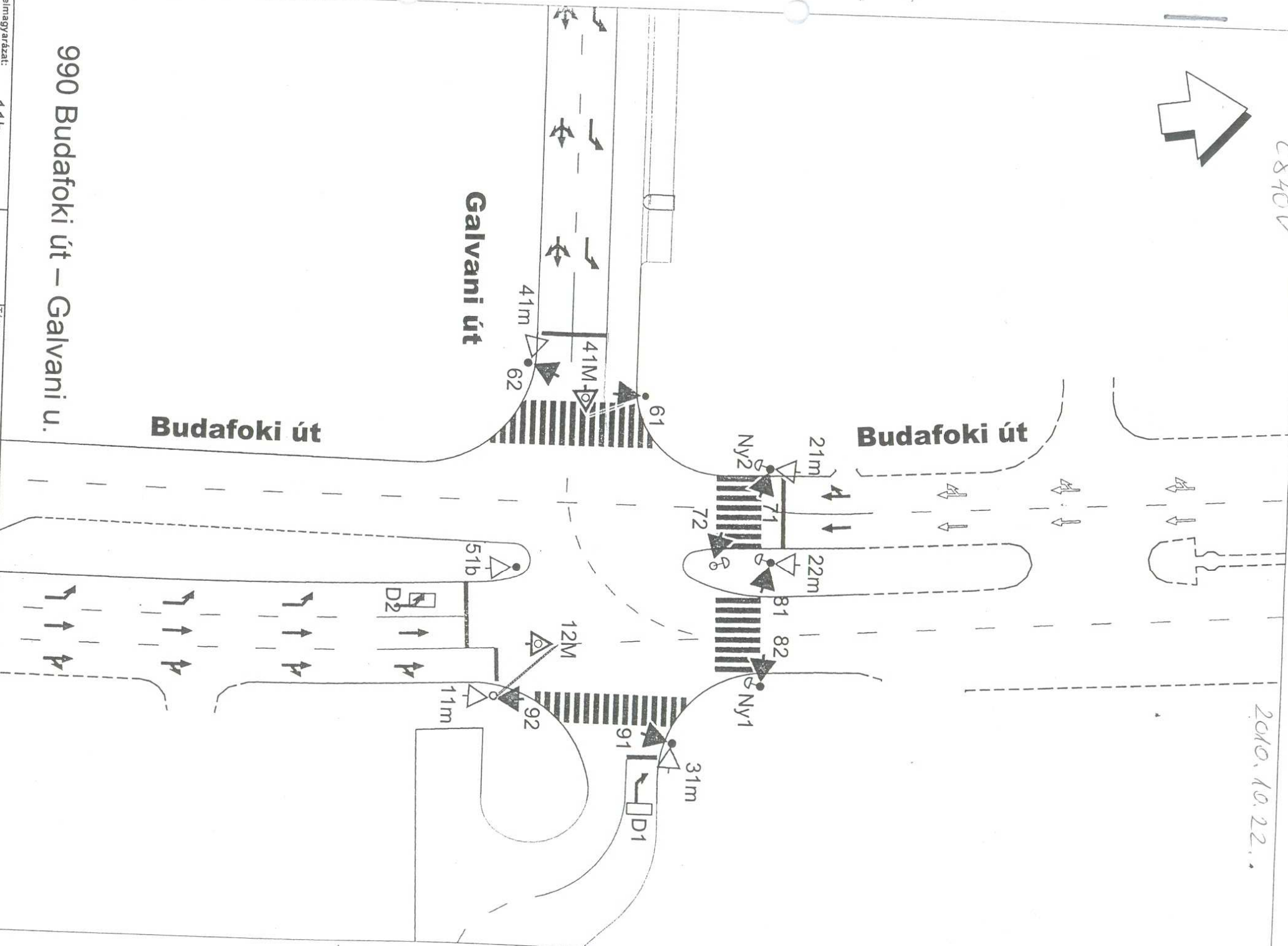
Meretany

1:500

Jelzőszámozási helyszínrajz

Csomópontszám:
983

2010.10.22.



Jelmeagyarázat:  Meglévő jelző  Tervezett jelző		11b 	
Tárgy:			
Dél-budai tehermentesítő út (Galvani út - Budafoki út)		Munkakész:	
Jelzőlámpa számozási terv		Dátum: 2009. 06. 02.	
Tervező: Ercsényi Balázs		Tervezési: 11. 09. 011	
Méretarány:		Rajzsám: K-02-12-R-06	
Lapszám:		Lapszám:	

Dátum:	Tervezési:
2009. 06. 02.	11.09.011
Tervező:	Bevezető:

Méretarány:	K-02-12-R-06
-------------	--------------

Melléklet 2: Jelenlegi jelzéstervek

JELZŐ			Tervező Huszár D.	Tervezés dátuma 2008.07.10	Jóváhagyó	Jóváhagyás dátuma	Megjegyzés		
srsz	Száma	tip	Periódusidő 90	Programváltási pont	Bekapcsolási pont 20	Szinkron pont 50	Eltolás	Zöld idő	Kapacitás
			0102030405060708090						
1	11,12	J	18313766 --D^--NY--NY					48	1920
2	21	KI	11182529 NYD--NY					18	360
3	31,32,33	J	17313566 --D^--NY--NY					49	2940
4	41,42,43	J	41172 NY					29	1740
5	51	SV	1037 NY						
6	61-62	GY	3971 NY					28	
7	71-72	GY	61274 NY					28	
8	81-82	GY	91661 NY					52	
9	91-92	GY	1223 NY					11	
10	101-102	GY	19313565 --D^--NY--NY					46	
11	141	J	12182630 NYD--NY					18	360
12	2. sor masmo		1124					13	
13	4. sor masmo		369					24	
14	6. sor masmo		270					22	
15	7. sor masmo		573					22	
16	11. sor masmo		1525					10	
									Str. száma

JELZŐ			Tervező Huszár D.	Tervezés dátuma 2008.09.02	Jóváhagyó	Jóváhagyás dátuma	Megjegyzés X=DvNY		
srsz	Száma	tip	Periódusidő 90	Programváltási pont	Bekapcsolási pont 47	Szinkron pont 40	Eltolás	Zöld idő	Kapacitás
			0102030405060708090						
1	11,12	J	26 85					90	3600
2	21,22	J	26 85					90	3600
3	31	J	119					18	360
4	41-42	GY	120					19	
5	51-52	GY	120					19	
6	61-62	GY	2485					90	
Továbbító pontok			18	37	40	80			Str. száma
Várakozási idők			10	7	0	37			3
			SYN						
Siemens 979 Vilati 979			Csomópont neve Budafoki út - Prielle K. utca 3. sz. program				Érvényesség 2009.04.20 -tól -ig		

JELZŐ			Tervező	Tervezés dátuma	Jóváhagyó	Jóváhagyás dátuma	Megjegyzés				
srsz	Száma	tip	Periódusidő	Programváltási pont	Bekapcsolási pont	Szinkron pont	Eltolás	Zöld idő	Kapa- citás		
			90		44	4					
1	11,12	J						61	2440		
2	21	J						17	340		
3	31,32,33	J						46	2760		
4	41,42	J						16	640		
5	51,52	J						5	200		
6	61-62	GY						23			
7	71-72	GY						8			
8	81-82	GY						48			
9	91-92	GY						46			
10	101-102	GY						24			
11	111-112	GY						48			
12	121-122	GY						47			
13	131-132	GY						16			
14	141	KI						14	280		
15	151	KI						15	300		
Továbbító pontok			1 4	20 29	45	65			Str. száma 3		
Várakozási idők			10 0	13 0	0	3					
			SYN								
Siemens 919 Vilati 272			Csomópont neve Budafoki út - Hengermalom u. 3. sz. program				Érvényesség 2007.10.05 -tól -ig				

JELZŐ			Tervező	Tervezés dátuma	Jóváhagyó	Jóváhagyás dátuma	Megjegyzés				
srsz	Száma	tip	Periódusidő	Programváltási pont	Bekapcsolási pont	Szinkron pont	Eltolás	Zöld idő	Kapa- citás		
			90		78	20					
			0102030405060708090								
1	11,12	J	31 73							48	1920
2	21,22	J	47 82							55	2200
3	31	J	36 47							11	220
4	41,42	J	55 66							11	220
5	51	KI	37 47							10	200
6	61-62	GY	54 66							12	
7	71-72	GY	52 73							21	
8	81-82	GY	28 71							47	
Továbbító pontok			17 20 27 44 61 79						Str. száma 3		
Várakozási idők			22 0 4 5 2 0								
			SYN								
Siemens			Csomópont neve					Érvényesség			
981			Budafoki út - Sztregova köz					2007.12.10 -tól			
Vilati			3. sz. program					-ig			
1001											

JELZŐ			Tervező	Tervezés dátuma	Jóváhagyó	Jóváhagyás dátuma	Megjegyzés	Zöld idő	Kapa- citás		
srsz	Száma	tip	Periódusidő	Programváltási pont	Bekapcsolási pont	Szinkron pont	Eltolás				
			90		87	20					
			0102030405060708090								
1	11,12	J						90			
2	21,22	J						90			
3	31	J						10			
4	41,42	J						14			
5	51,52	J						10			
6	61	KI						11			
7	71-72	GY						12			
8	81-82	GY						29			
9	91-92	GY						90			
10	101-102	GY						90			
Továbbító pontok			17 20	29	41	58	80		Str. száma		
Várakozási idők			12 0	6	5	2	0		3		
			SYN								
Siemens			Csomópont neve				Érvényesség				
982			Budafoki út - Újbuda Center kihajtó				2007.12.10 -tól				
Vilati											
1002			3. sz. program				-ig				

JELZŐ			Tervező	Tervezés dátuma	Jóváhagyó	Jóváhagyás dátuma	Megjegyzés		
srsz	Száma	tip	Periódusidő	Programváltási pont	Bekapcsolási pont	Szinkron pont	Eltolás	Zöld idő	Kapacitás
			90		70	10			
			<						

JELZŐ			Tervező Főmterv	Tervezés dátuma 2009.06.02	Jóváhagyó	Jóváhagyás dátuma	Megjegyzés		
srsz	Száma	tip	Periódusidő 90	Programváltási pont	Bekapcsolási pont 5	Szinkron pont 25	Eltolás	Zöld idő	Kapacitás
1	11,12	J						71	2840
2	21,22	J						69	2760
3	31	J						12	240
4	41,42	J						7	280
5	51	J						9	180
6	61-62	GY						66	
7	71-72	GY						23	
8	81-82	GY						11	
9	91-92	GY						65	

Melléklet 3: Közbensőidő mátrixok

[illegible]

	Tervező	Tervezés dátuma	Jóváhagyó	Jóváhagyás dátuma	Megjegyzés
--	---------	-----------------	-----------	-------------------	------------

SZABAD JELZÉS VÉGÉN KIHALADÓ			SZABAD JELZÉS ELEJÉN BEHALADÓ														
ssz	száma	jele	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	11,12	J				5	7		9		5						
2	21	J			7	6	6			9	5					7	
3	31,32,33	J		6		7	5	5						9	7		
4	41,42	J	7	6	7		8		10	9			5				
5	51,52	J	4	4	7	5				8		5				7	
6	61-62	GY			8											8	
7	71-72	GY	5			5											
8	81-82	GY				5											
9	91-92	GY		5			5									5	
10	101-102	GY	9	9													
11	111-112	GY					7										
12	121-122	GY				7										7	
13	131-132	GY			5											5	
14	141	KI			5								5	8			
15	151	KI		4			5	5		7							

Siemens Vilati	919 272	Csomópont neve Budafoki út - Hengermalom u. Tiltás ill. közbenső idő mátrix	Érvényesség -tól -ig
-------------------	------------	---	----------------------------

[illegible]

[illegible]

Siemens	Csomópont neve	Érvényesség
982	Budafoki út - Újbuda Center kihajtó	-tól
Vilati	Tiltás ill. közbenső idő mátrix	-ig
1002		

Siemens 983 Vilati 2983	Csomópont neve Budafoki út - Barázda u. Tiltás ill. közbenső idő mátrix	Érvényesség -tól -ig
---	---	----------------------------

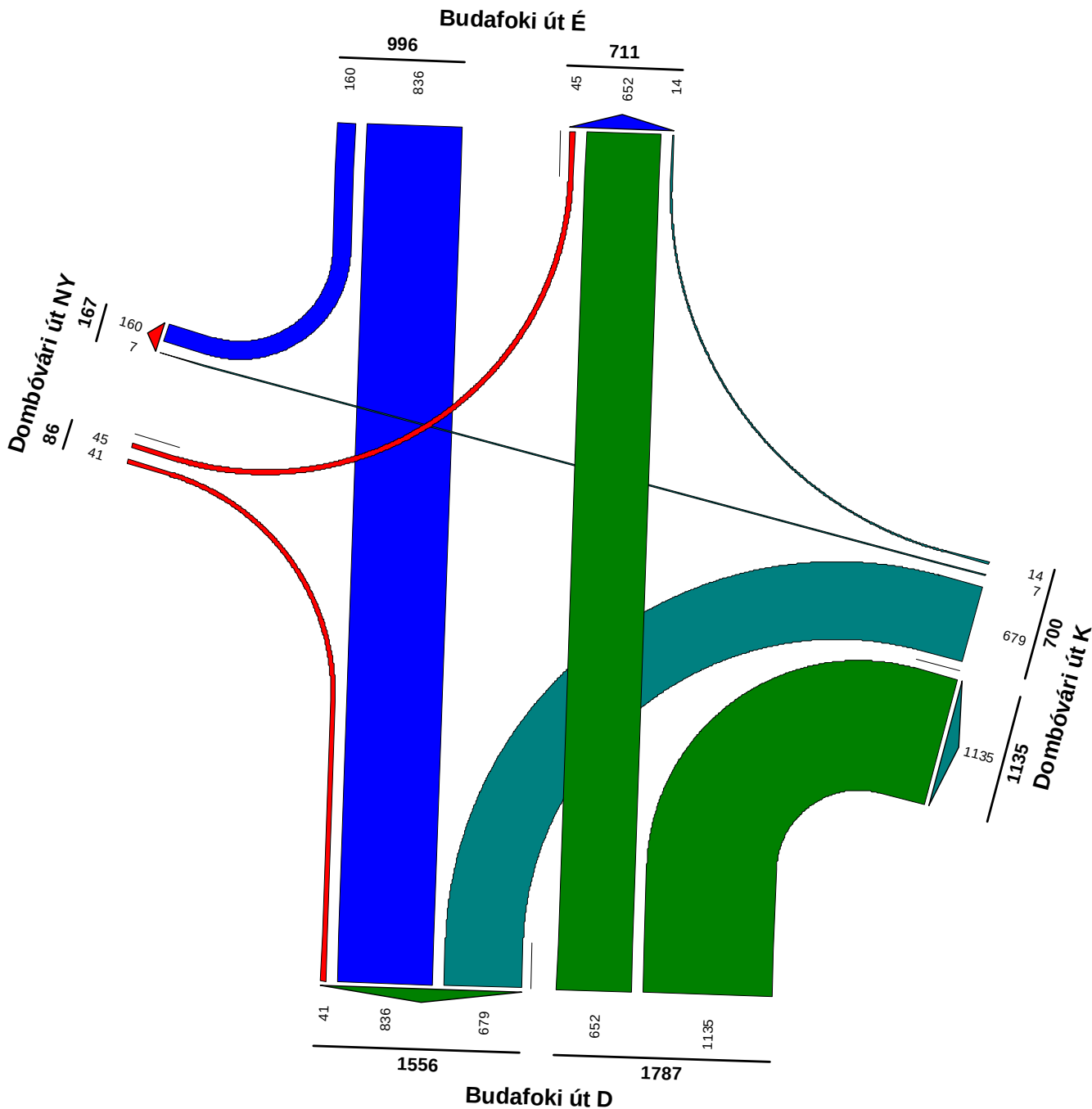
Siemens 990 Vilati 1021	Csomópont neve Budafoki út - Galvani u. Tiltás ill. közbenső idő mátrix	Ervényesség 2010.10.22 -tól -ig
----------------------------------	---	---------------------------------------

Melléklet 4: Forgalomáramlási ábrák

Forgalomáramlási ábra

LISA+

15:00-16:00



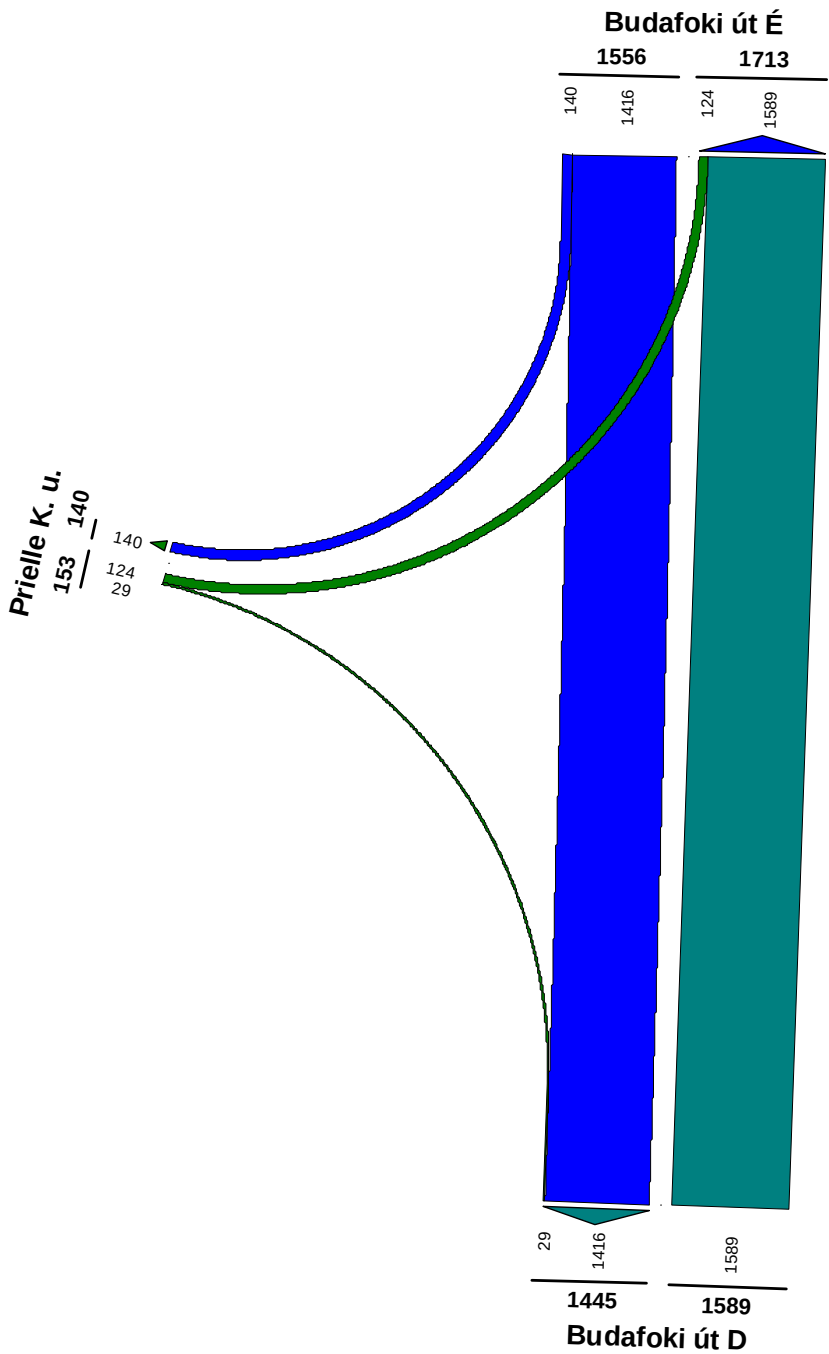
From\To	2	4	1	3
2			836	160
4	14		679	7
1	652	1135		
3	45		41	



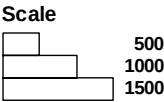
Forgalomáramlási ábra

LISA+

15:00-16:00



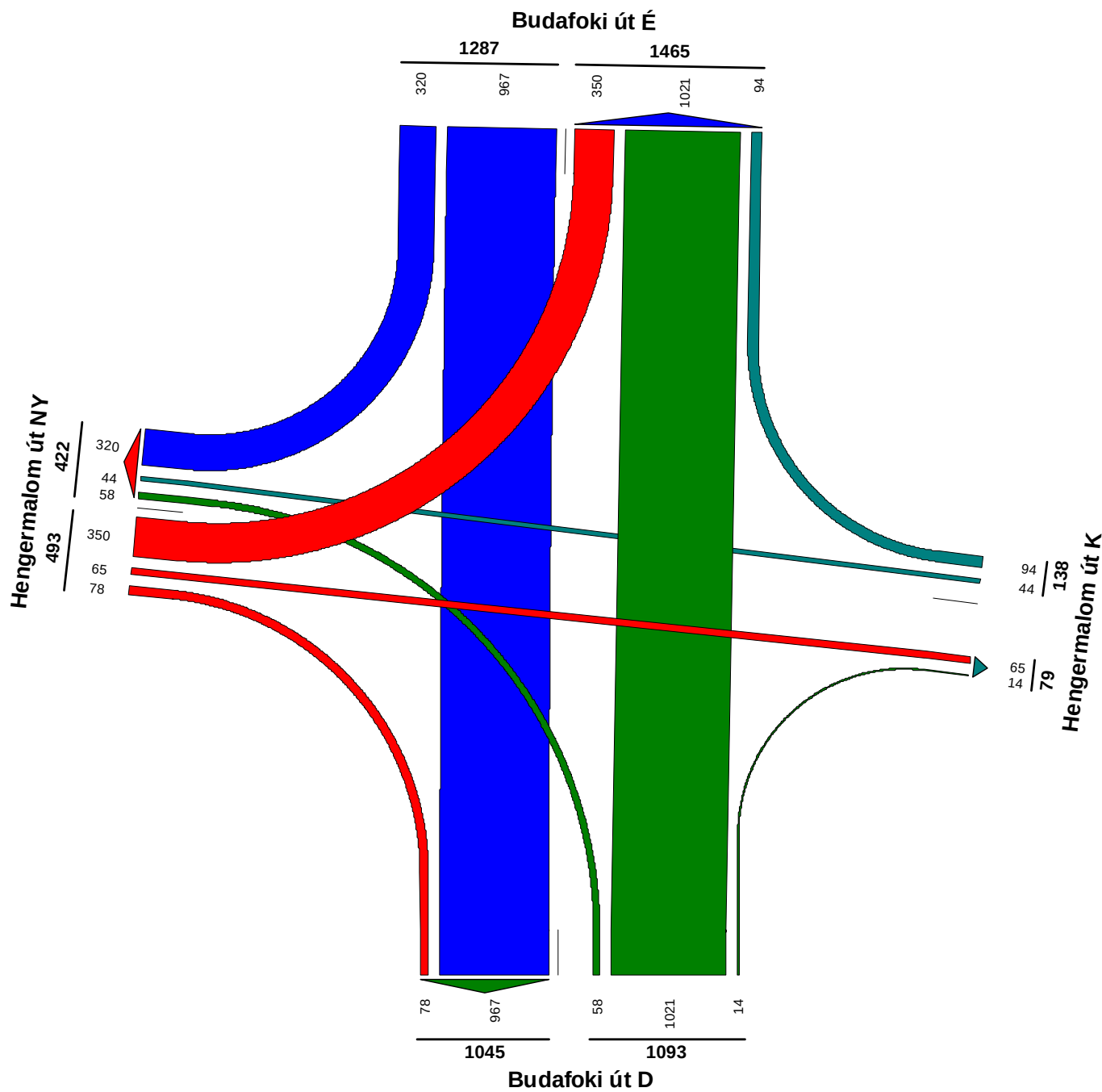
From\To	2	1	3
2		1416	140
1	1589		
3	124	29	



Forgalomáramlási ábra

LISA+

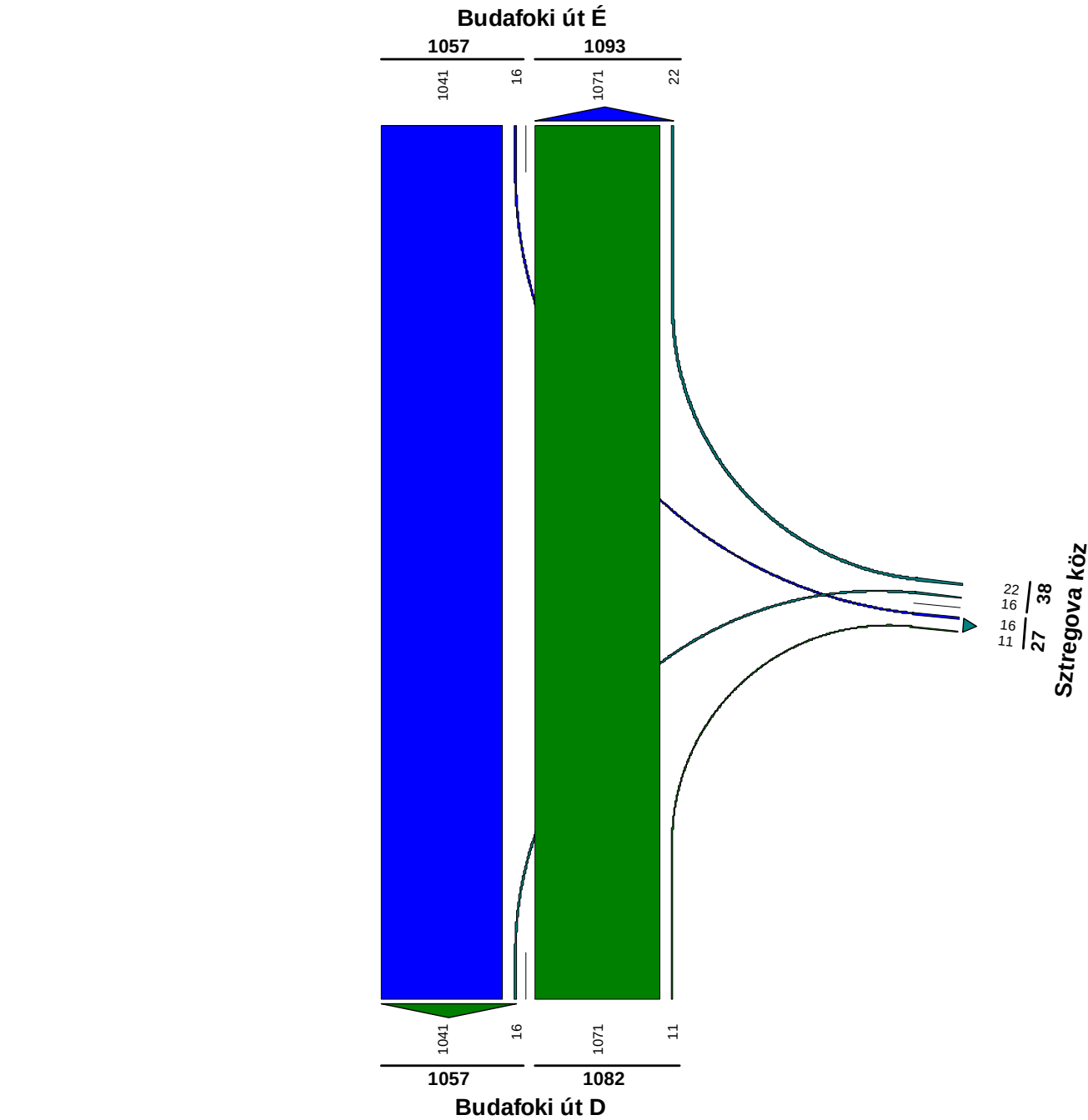
15:00-16:00



Forgalomáramlási ábra

LISA+

15:00-16:00

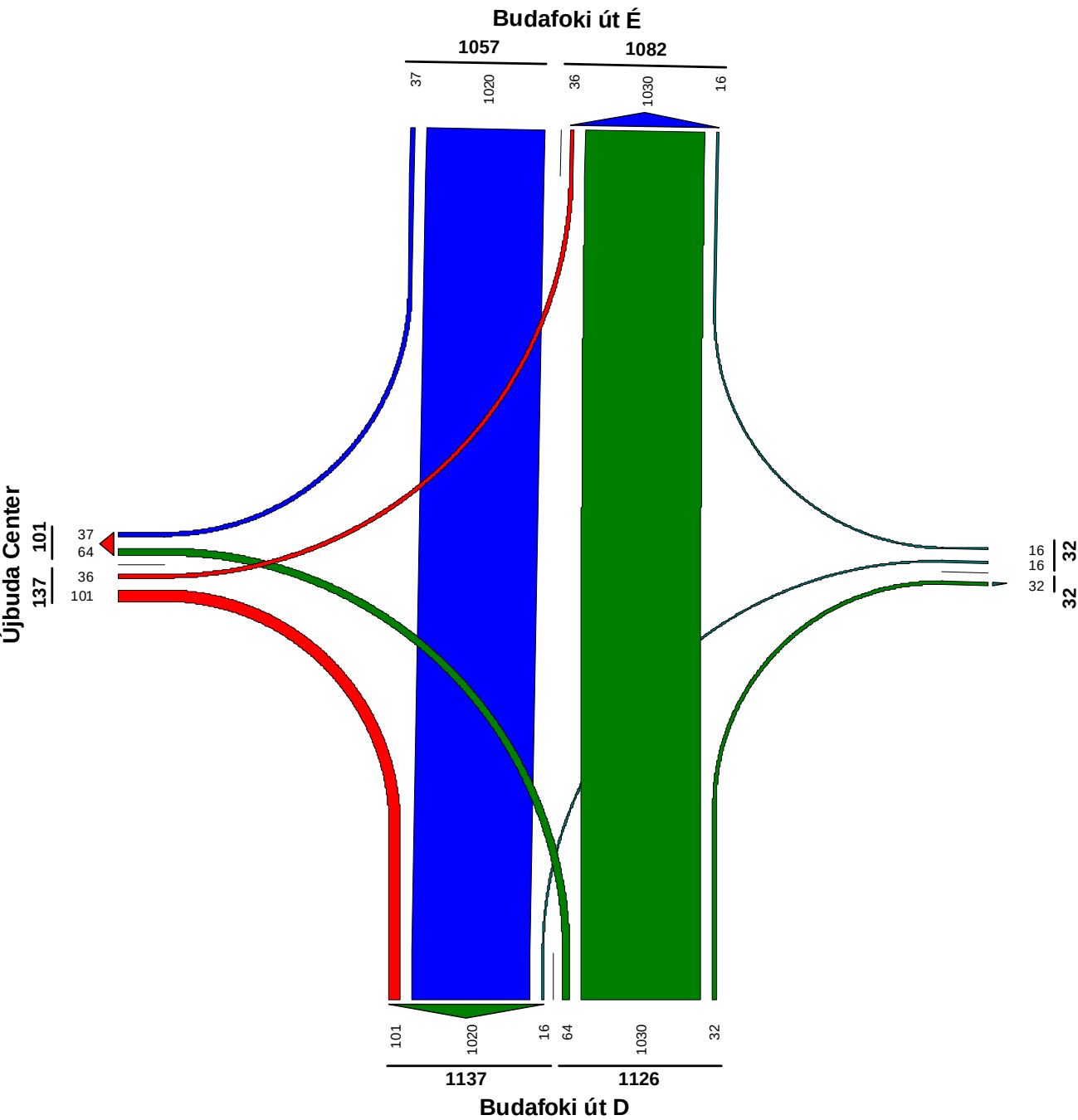


From\To	1	3	2
1		16	1041
3	22		16
2	1071	11	

Forgalomáramlási ábra

LISA+

15:00-16:00

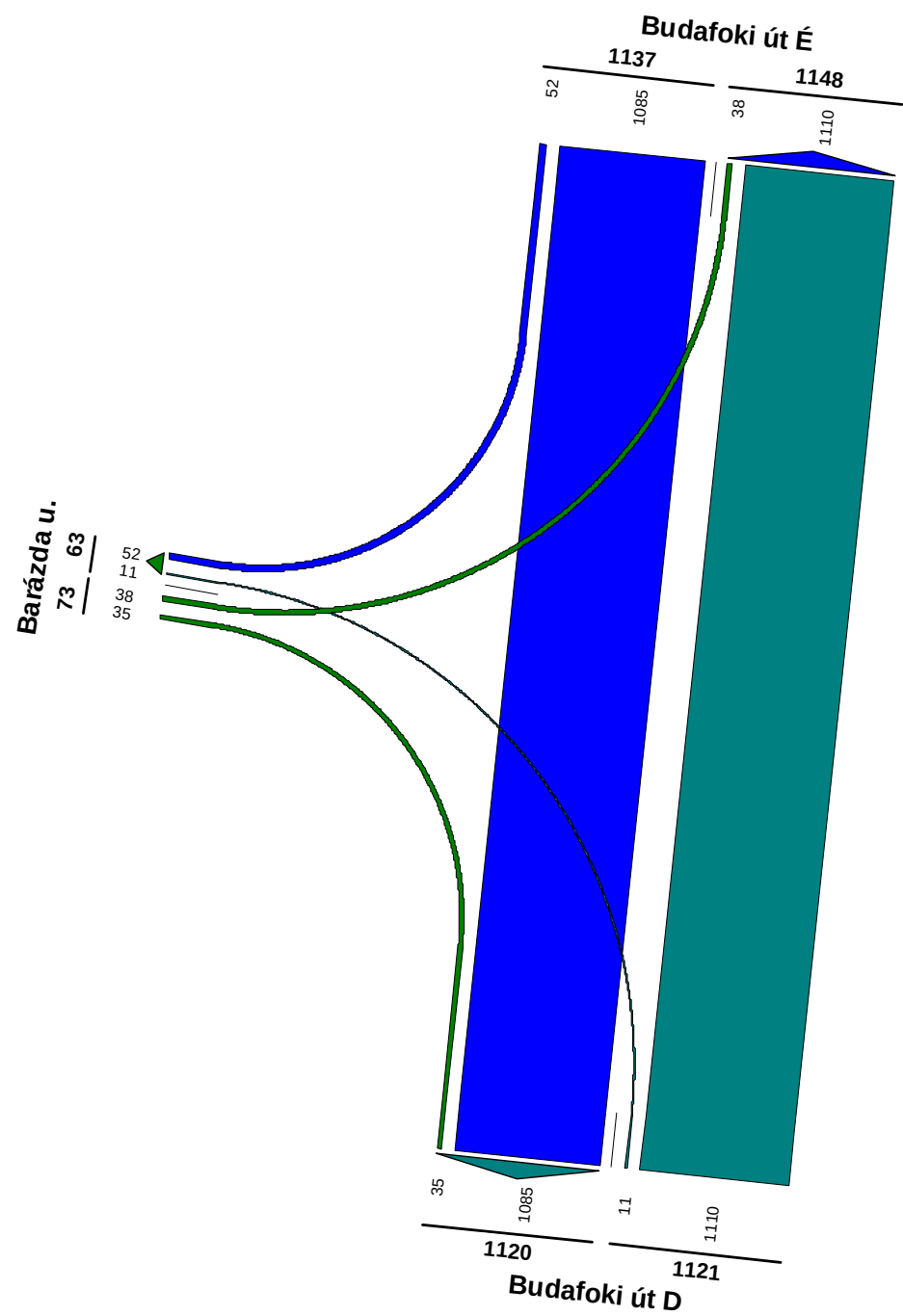


From\To	2	4	1	3
2			1020	37
4	16		16	
1	1030	32		64
3	36		101	

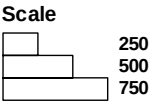
Forgalomáramlási ábra

LISA+

15:00-16:00



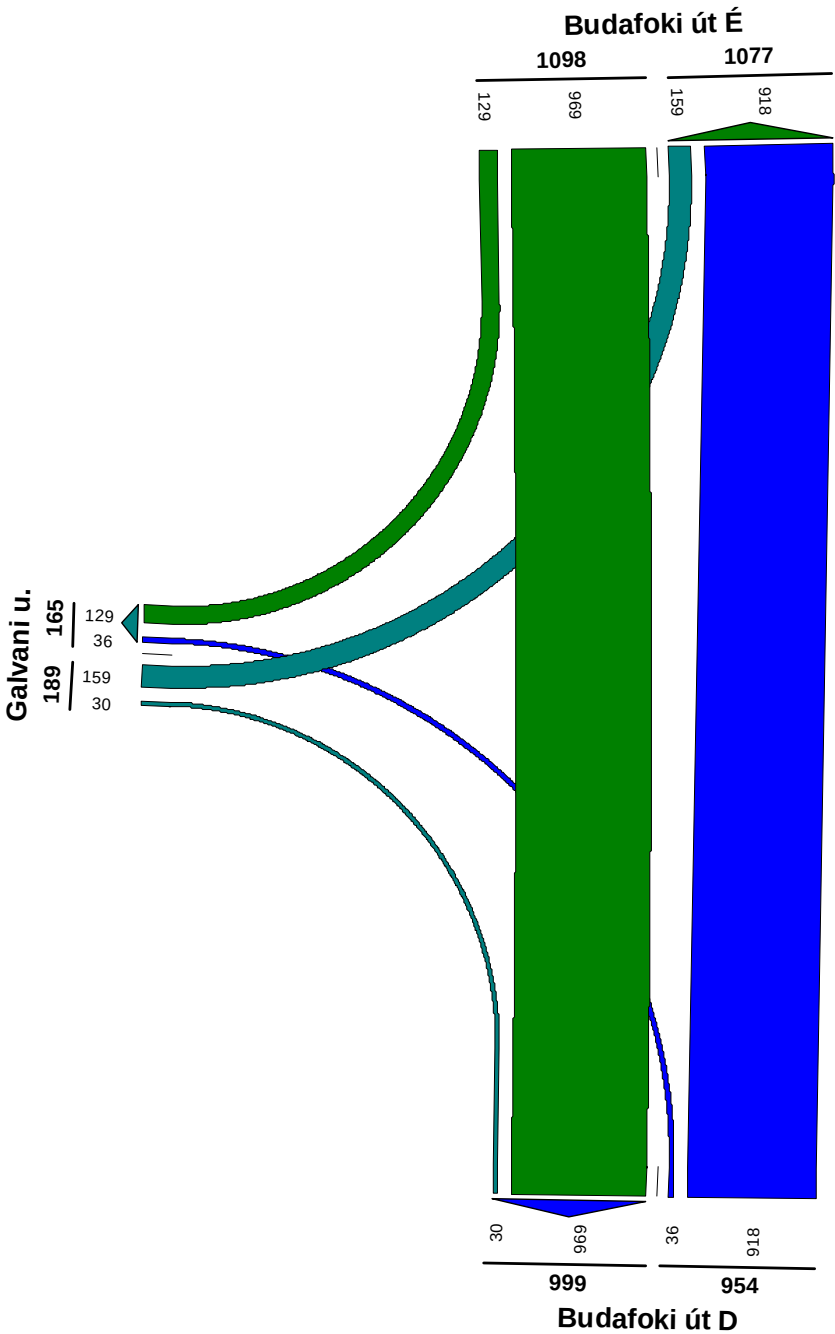
From\To	2	1	3
2		1085	52
1	1110		11
3	38	35	



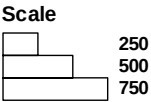
Forgalomáramlási ábra

LISA+

15:00-16:00



From\To	2	3	1
2		36	918
3	30		159
1	969	129	

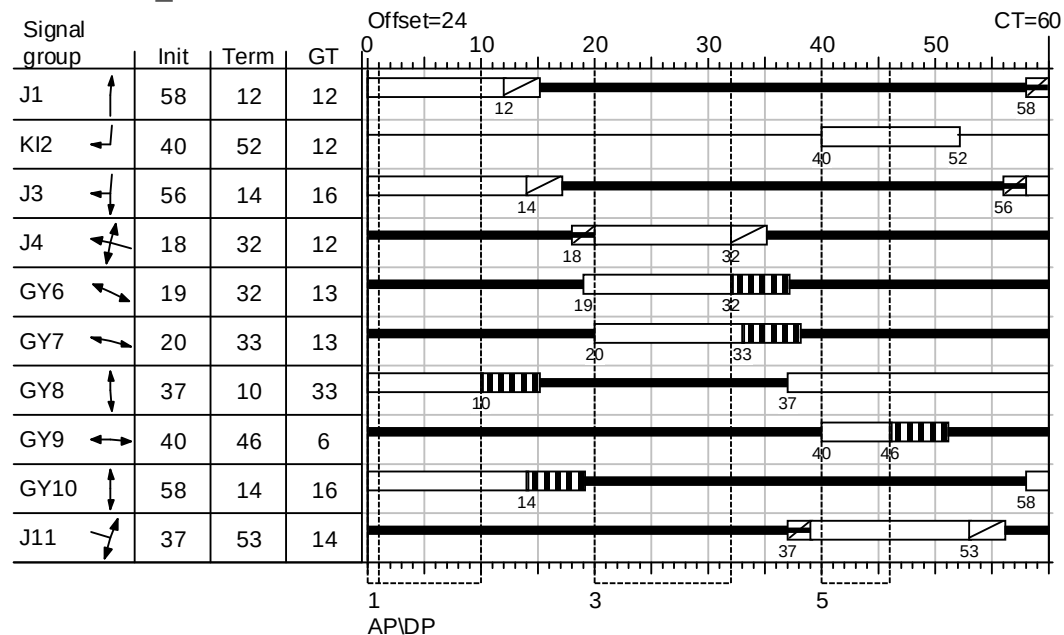


Melléklet 5: Tervezett jelzéstervek

Jelzésterv - 1. csomópont - 60 másodperces ciklusidő

LISA+

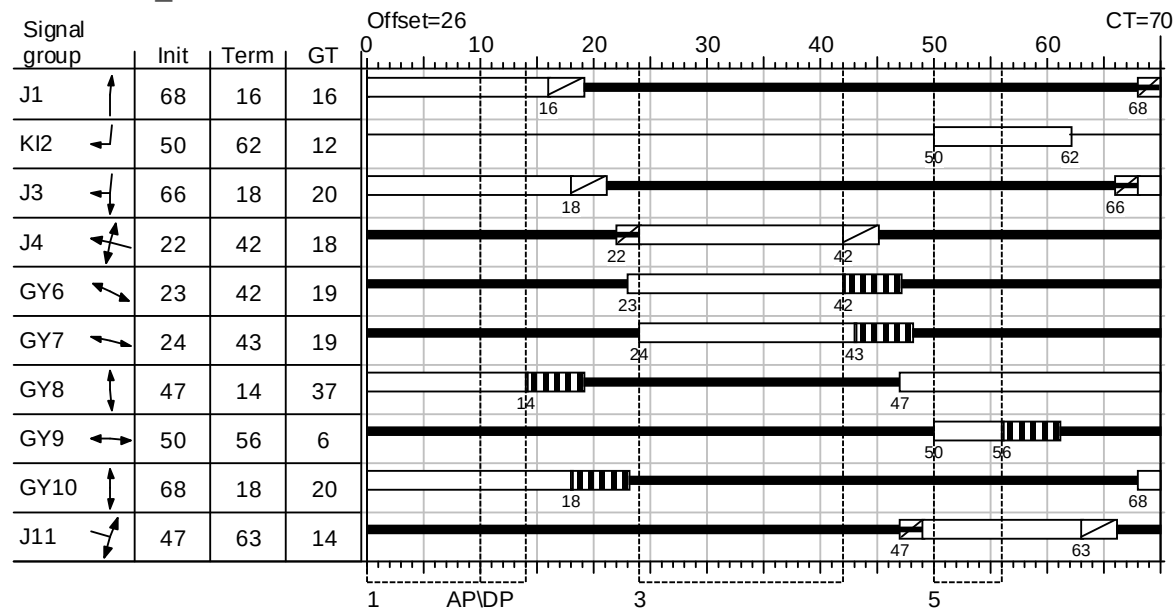
_STP 60 H



Jelzésterv - 1. csomópont - 70 másodperces ciklusidő

LISA+

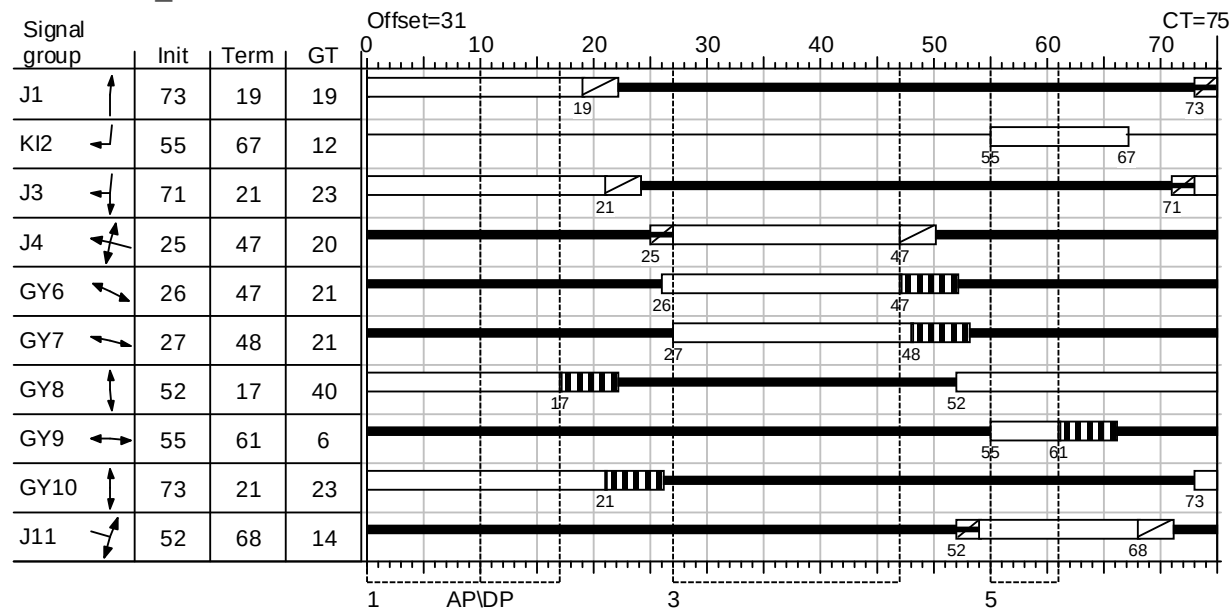
_STP 70



Jelzésterv - 1. csomópont - 75 másodperces ciklusidő

LISA+

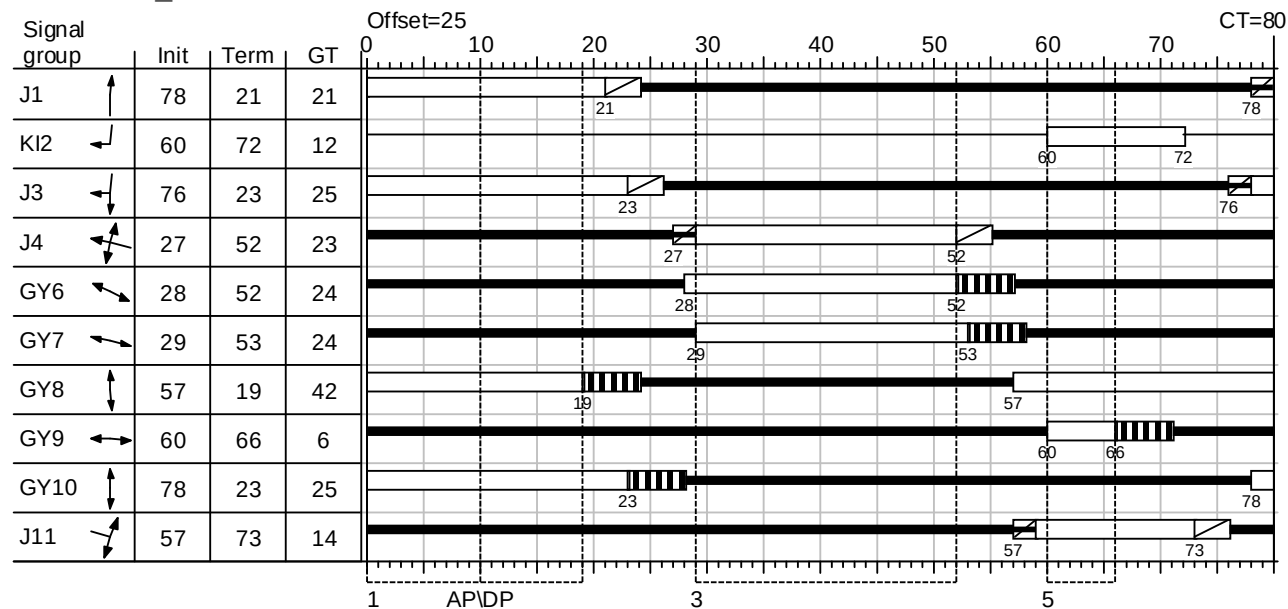
_STP 75



Jelzésterv - 1. csomópont - 80 másodperces ciklusidő

LISA+

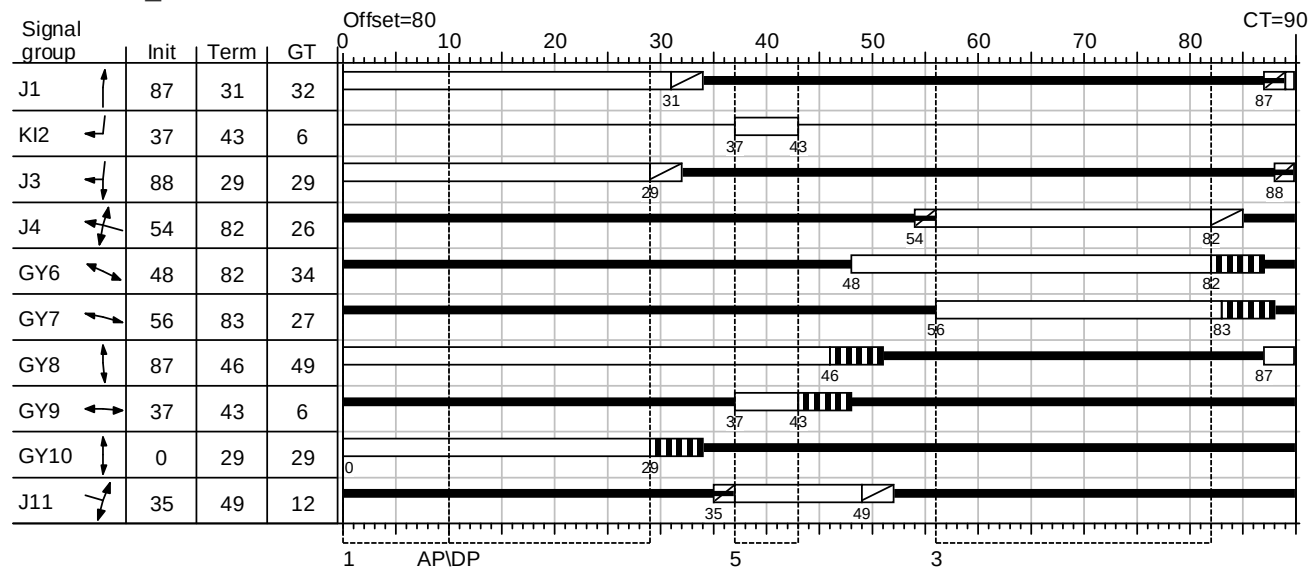
_STP 80 R



Jelzésterv - 1. csomópont - 90 másodperces ciklusidő

LISA+

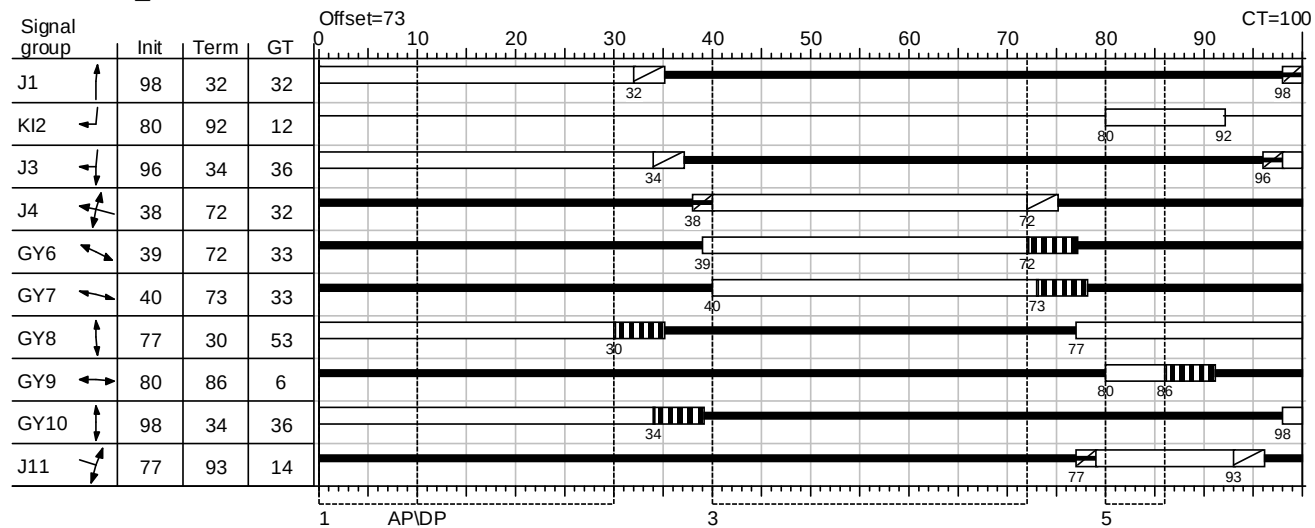
_STP 90 R



Jelzésterv - 1. csomópont - 100 másodperces ciklusidő

LISA+

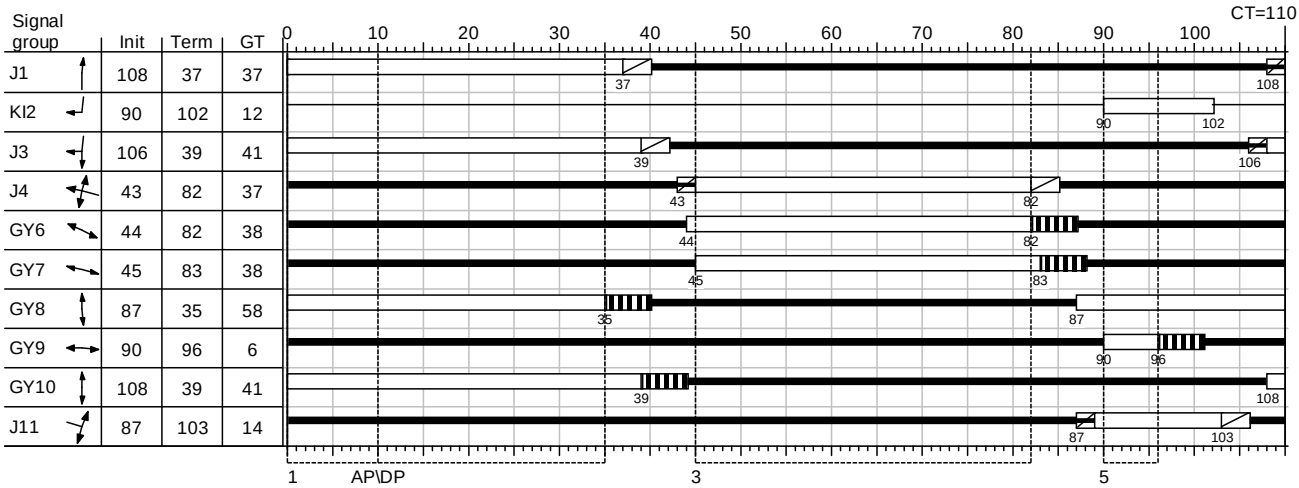
_STP 100 R



Jelzésterv - 1. csomópont - 110 másodperces ciklusidő

LISA+

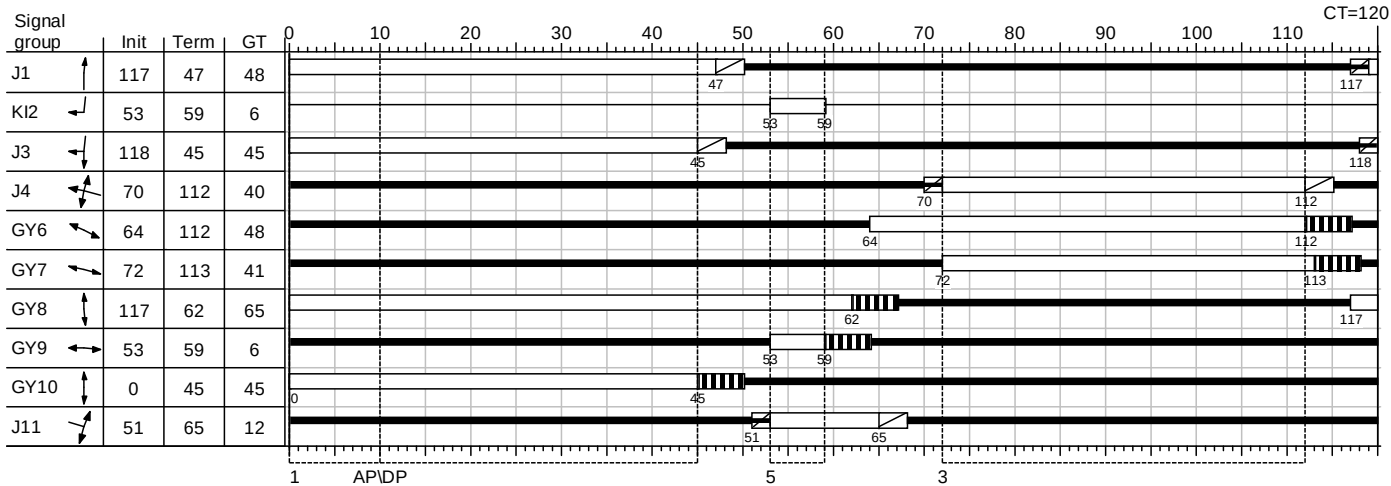
_STP 110 R



Jelzésterv - 1. csomópont - 120 másodperces ciklusidő

LISA+

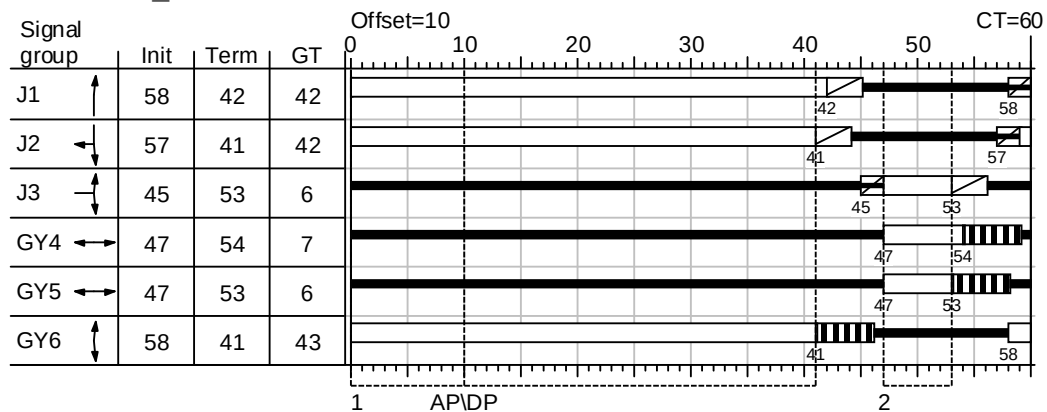
_STP 120 R



Jelzésterv - 2. csomópont - 60 másodperces ciklusidő

LISA+

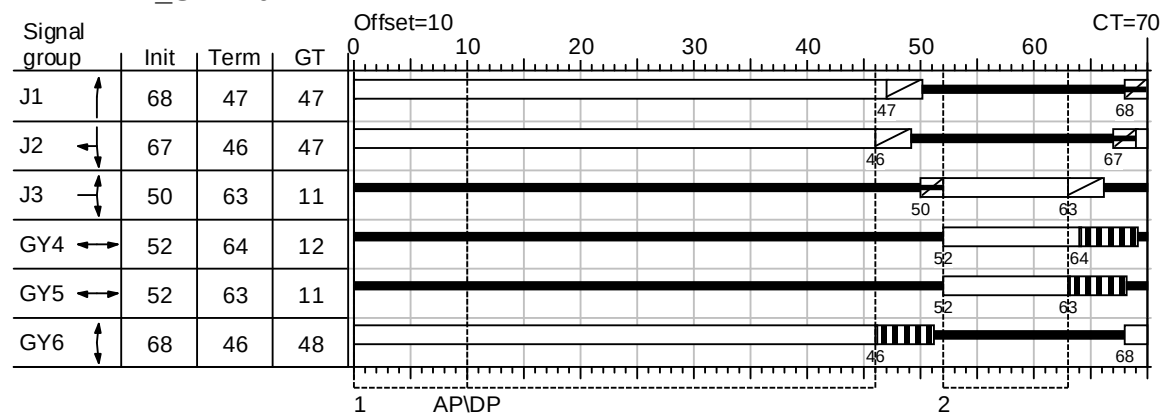
_STP 60 H



Jelzésterv - 2. csomópont - 70 másodperces ciklusidő

LISA+

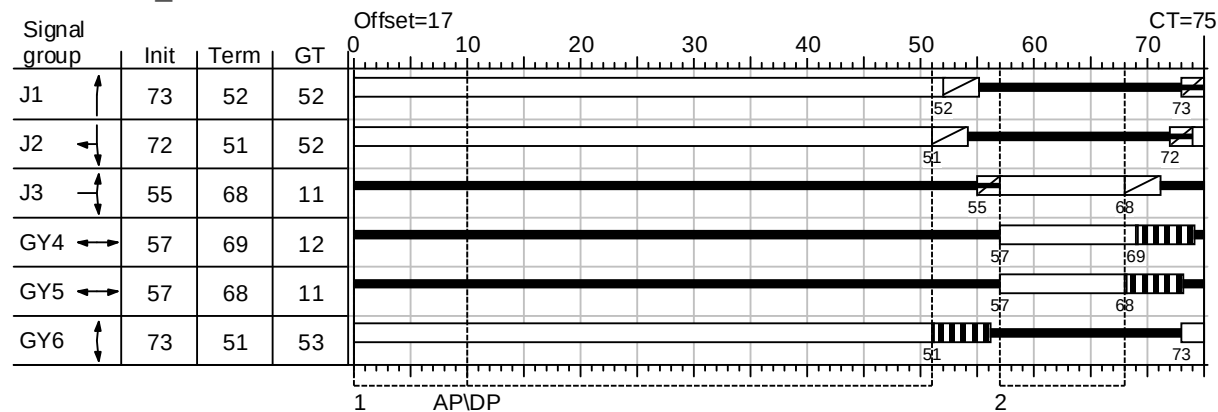
_STP 70



Jelzésterv - 2. csomópont - 75 másodperces ciklusidő

LISA+

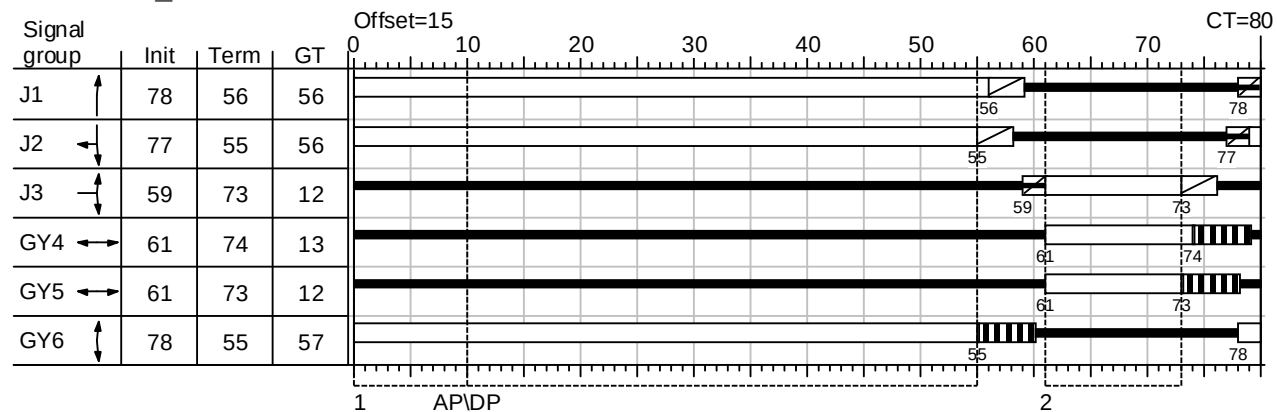
_STP 75



Jelzésterv - 2. csomópont - 80 másodperces ciklusidő

LISA+

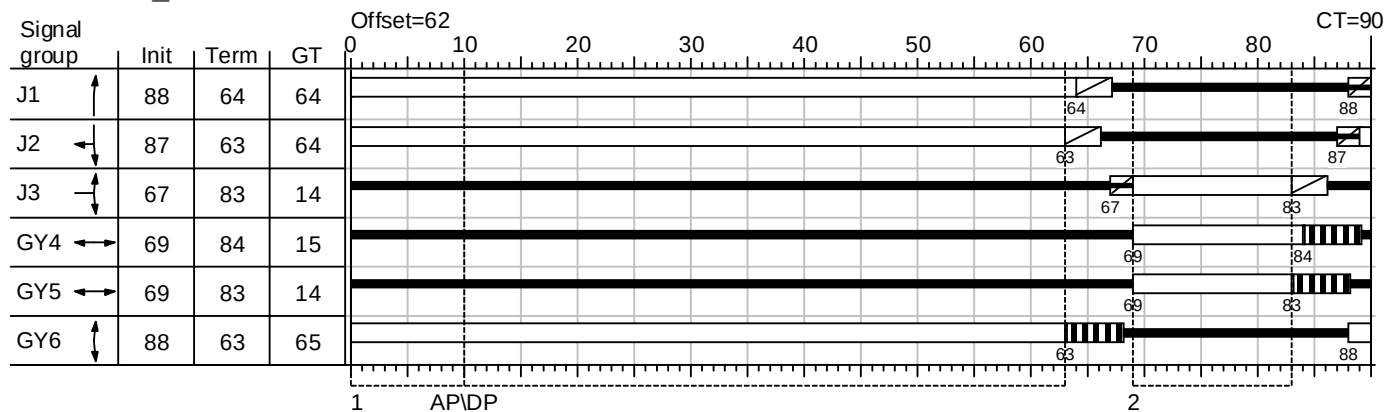
_STP 80 R



Jelzésterv - 2. csomópont - 90 másodperces ciklusidő

LISA+

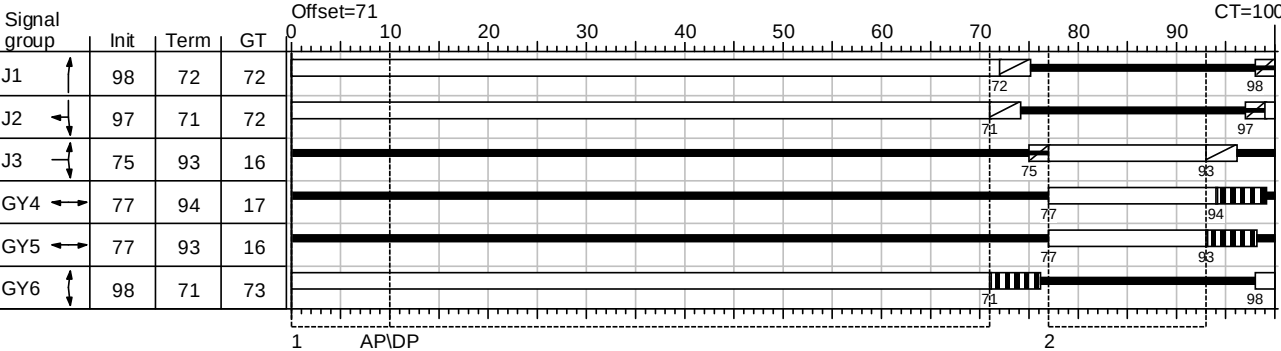
_STP 90 R



Jelzésterv - 2. csomópont - 100 másodperces ciklusidő

LISA+

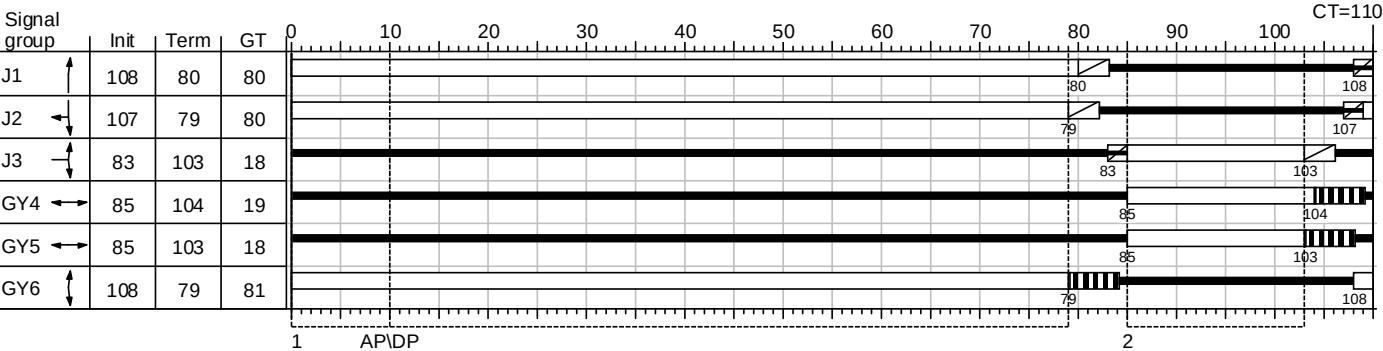
_STP 100 R



Jelzésterv - 2. csomópont - 110 másodperces ciklusidő

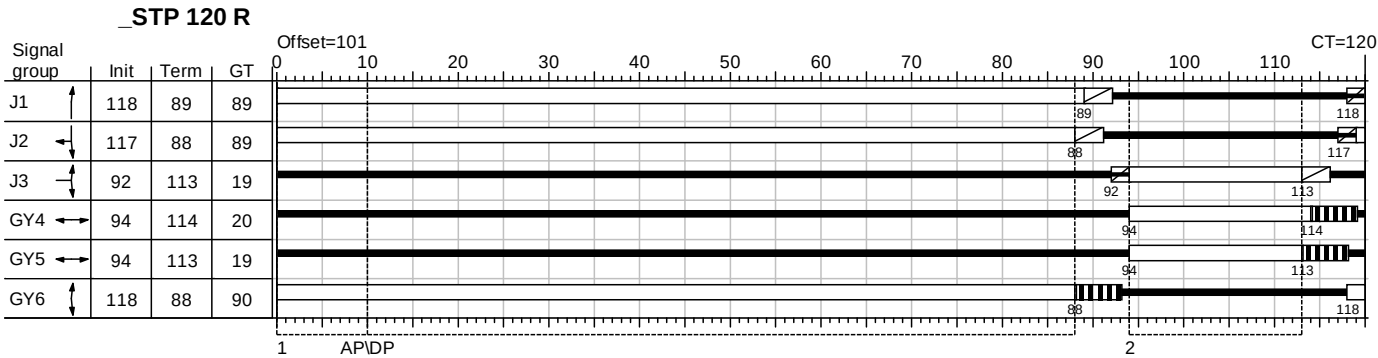
LISA+

_STP 110 R



Jelzésterv - 2. csomópont - 120 másodperces ciklusidő

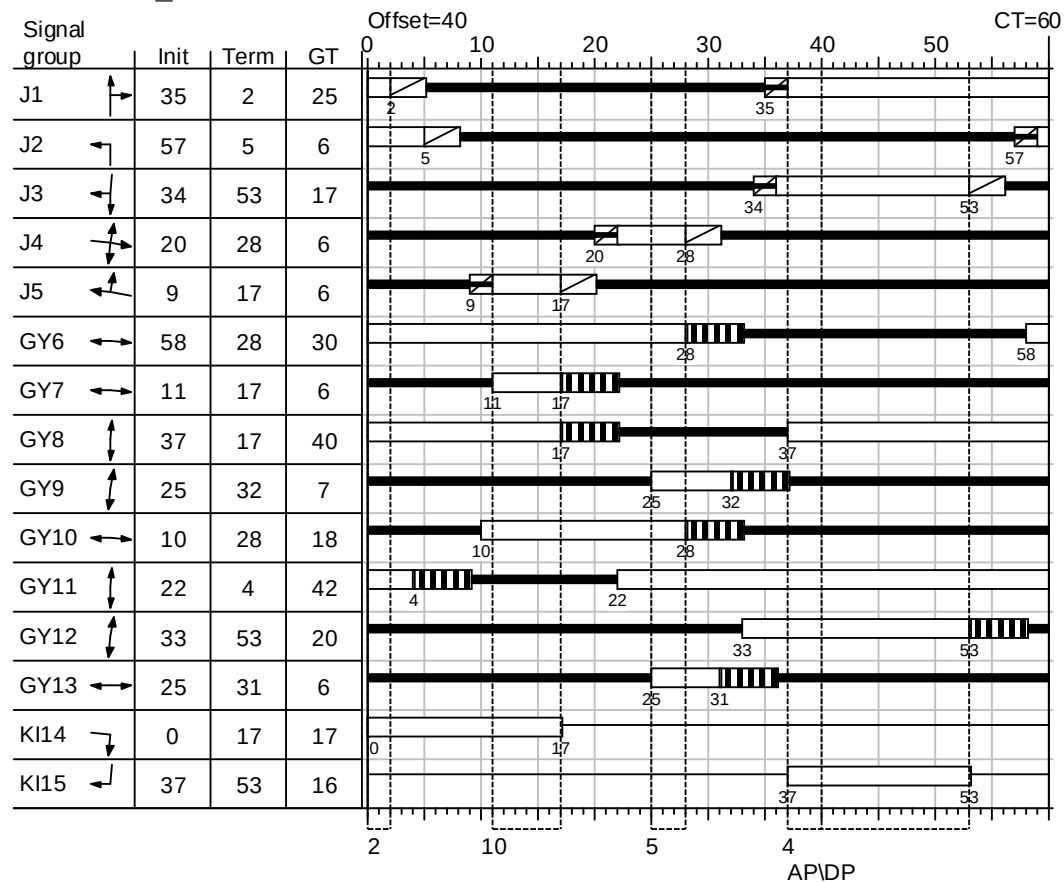
LISA+



Jelzésterv - 4. csomópont - 60 másodperces ciklusidő

LISA+

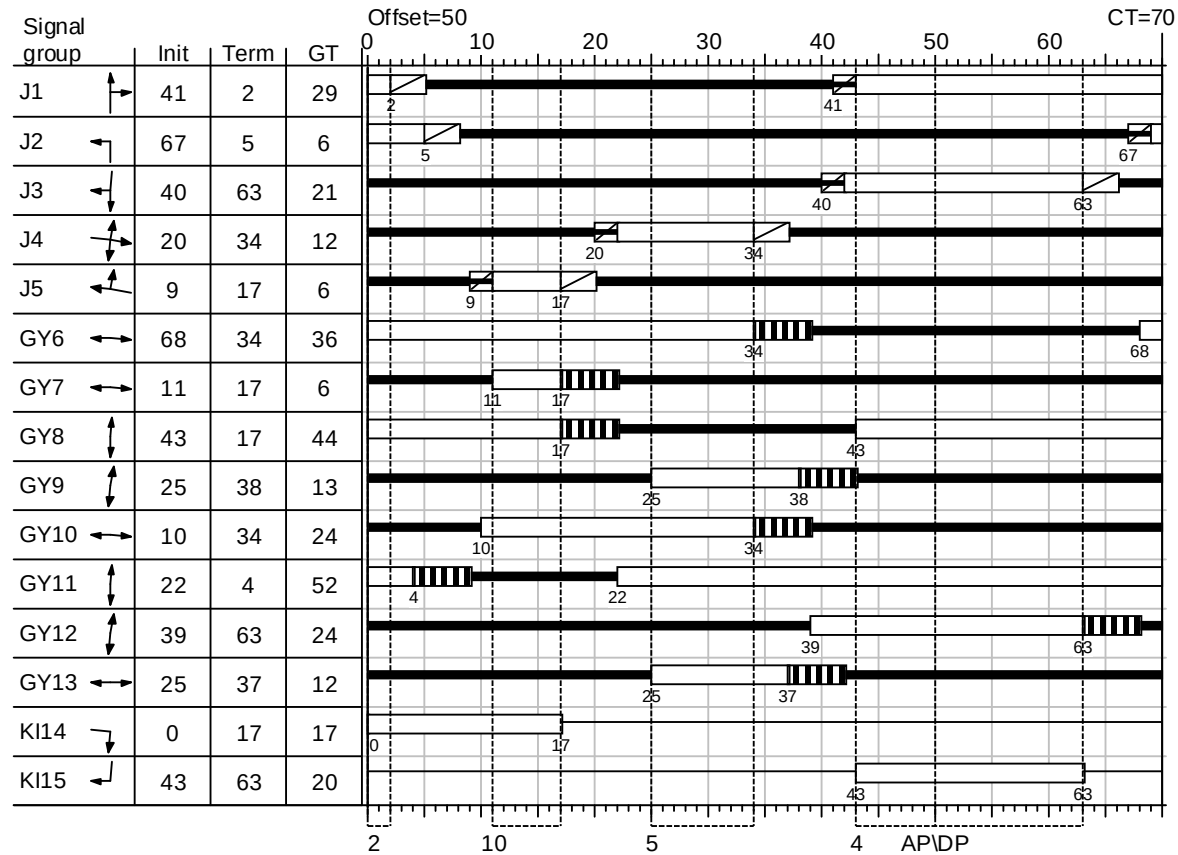
_STP 60 H



Jelzésterv - 4. csomópont - 70 másodperces ciklusidő

LISA+

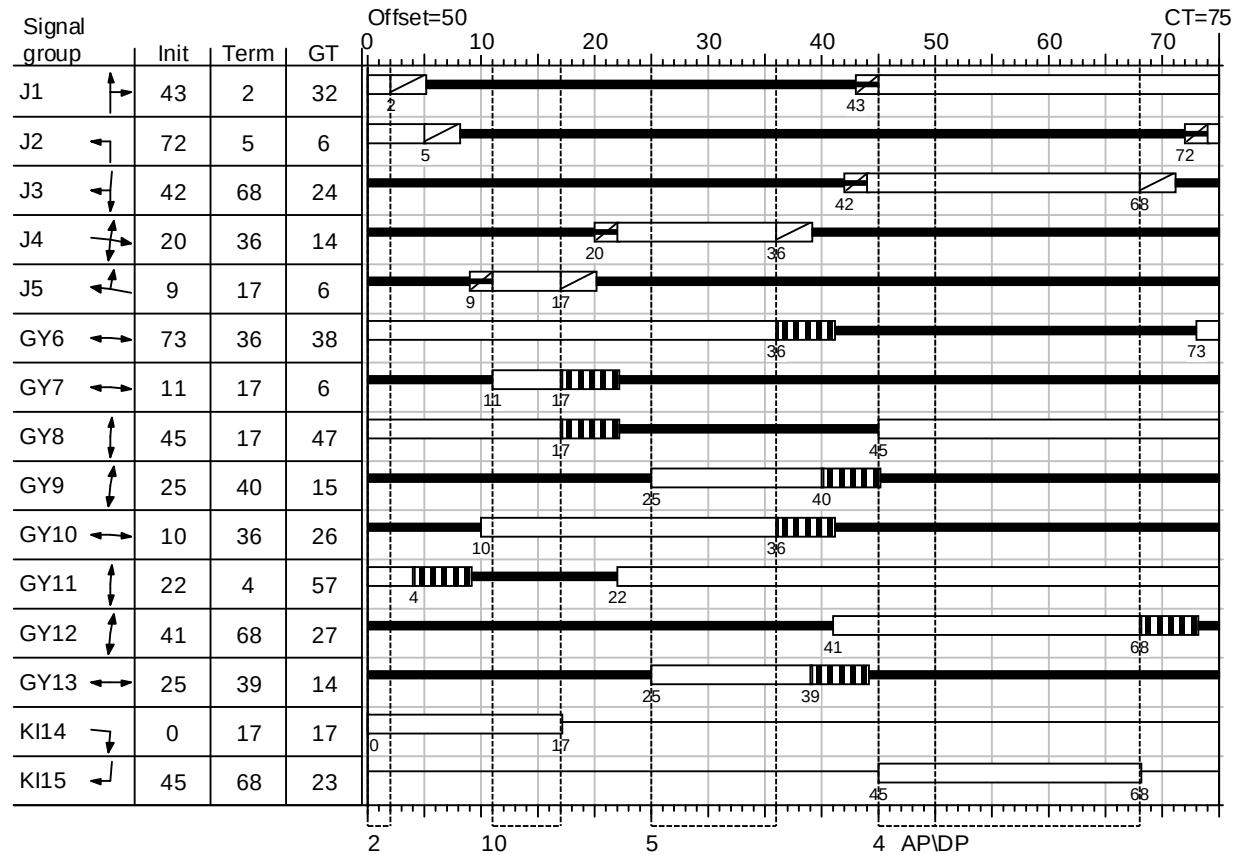
_STP 70



Jelzésterv - 4. csomópont - 75 másodperces ciklusidő

LISA+

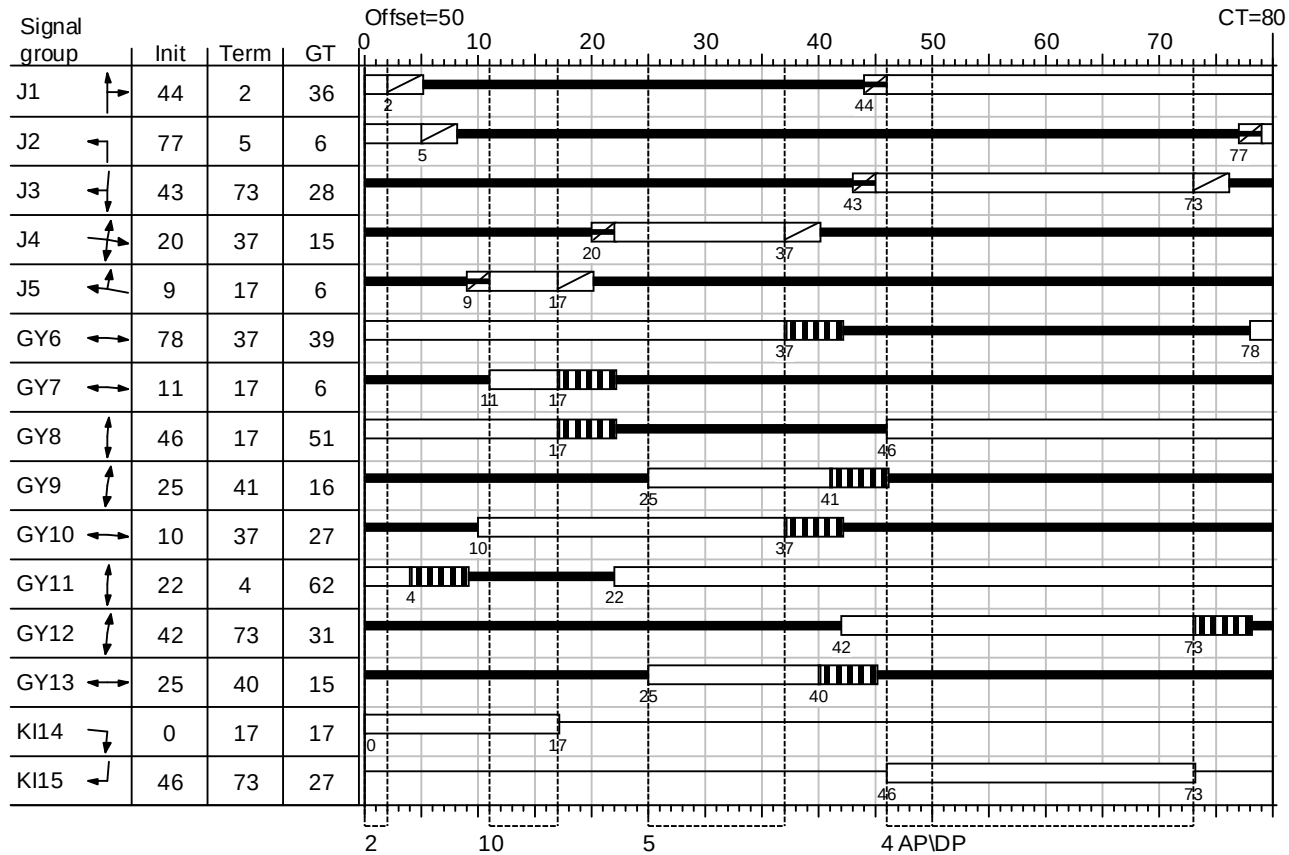
_STP 75



Jelzésterv - 4. csomópont - 80 másodperces ciklusidő

LISA+

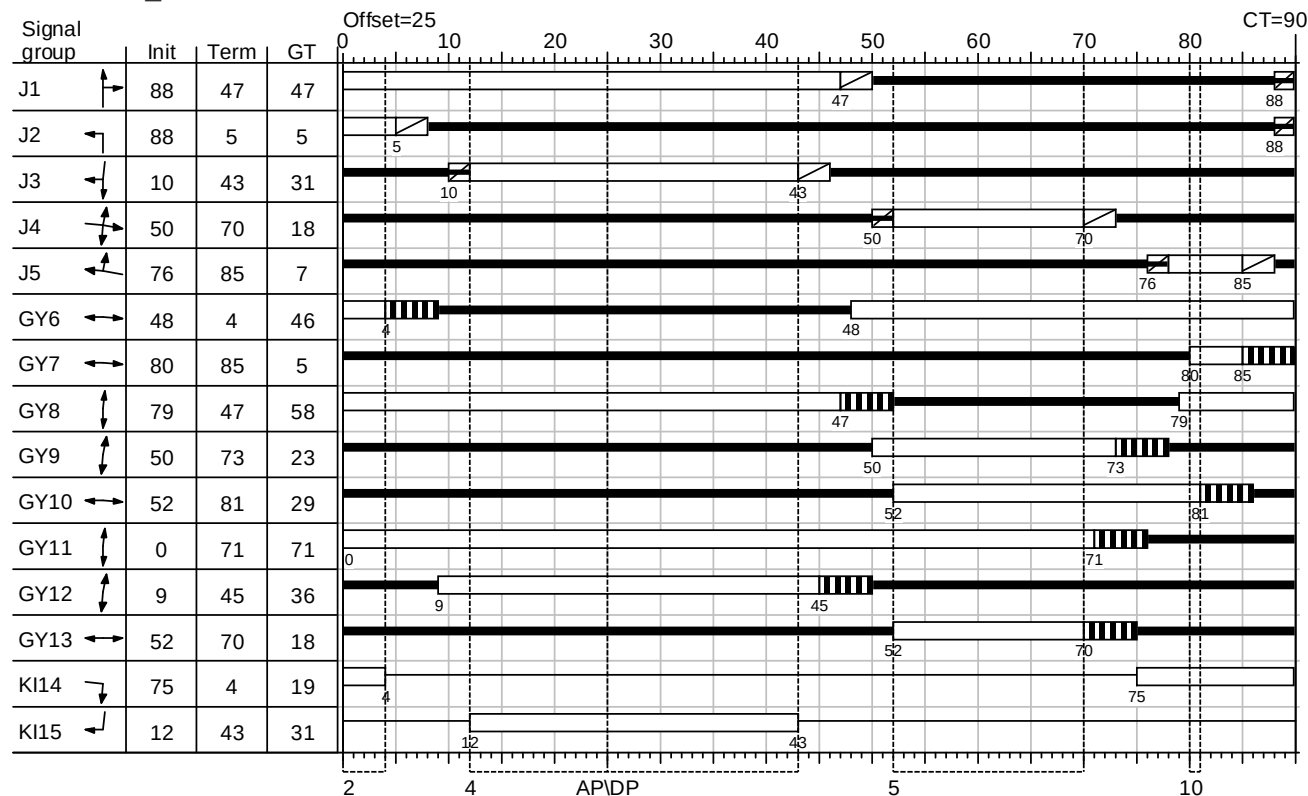
_STP 80 R



Jelzésterv - 4. csomópont - 90 másodperces ciklusidő

LISA+

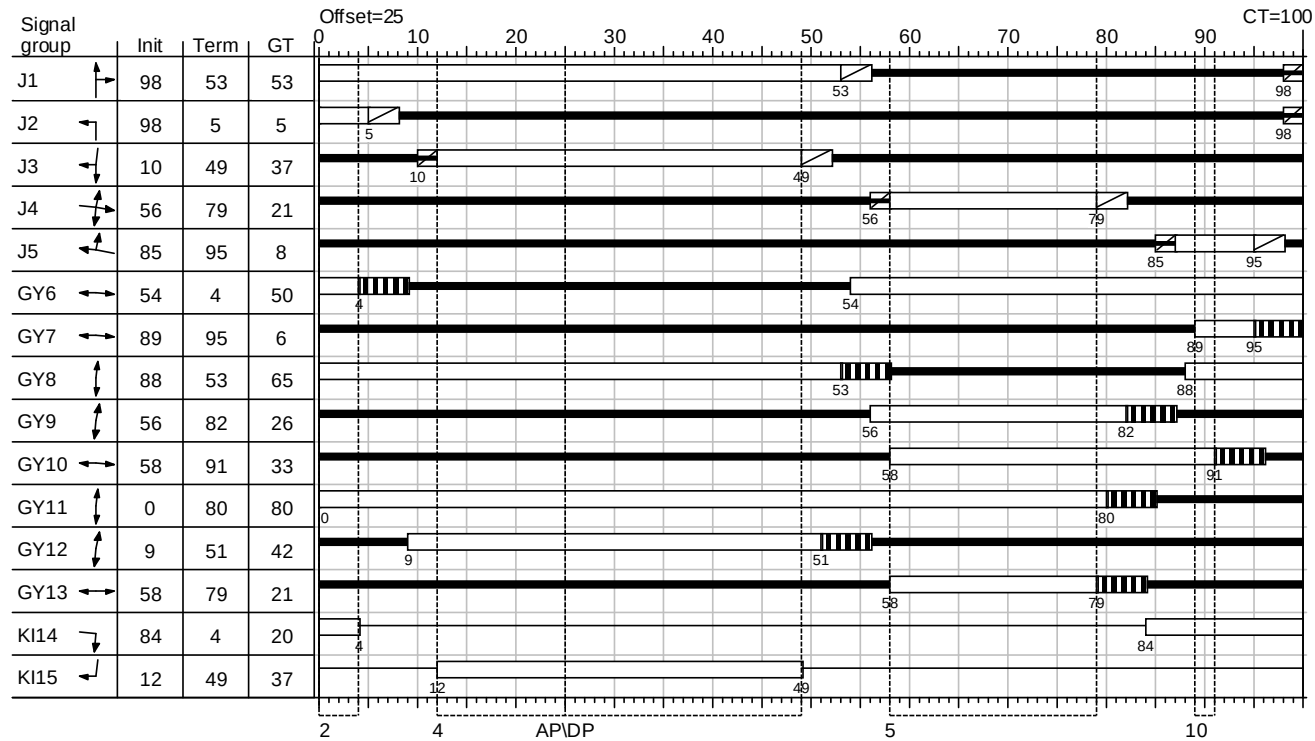
_STP 90 R



Jelzésterv - 4. csomópont - 100 másodperces ciklusidő

LISA+

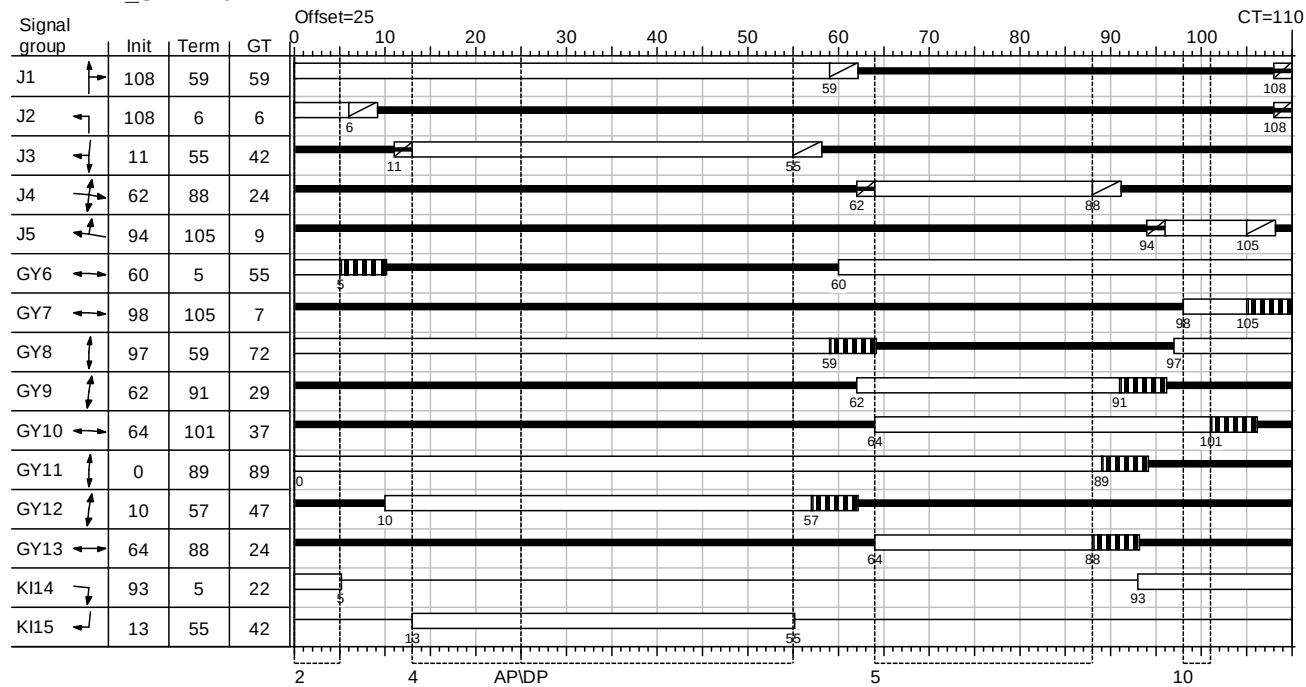
_STP 100 R



Jelzésterv - 4. csomópont - 110 másodperces ciklusidő

LISA+

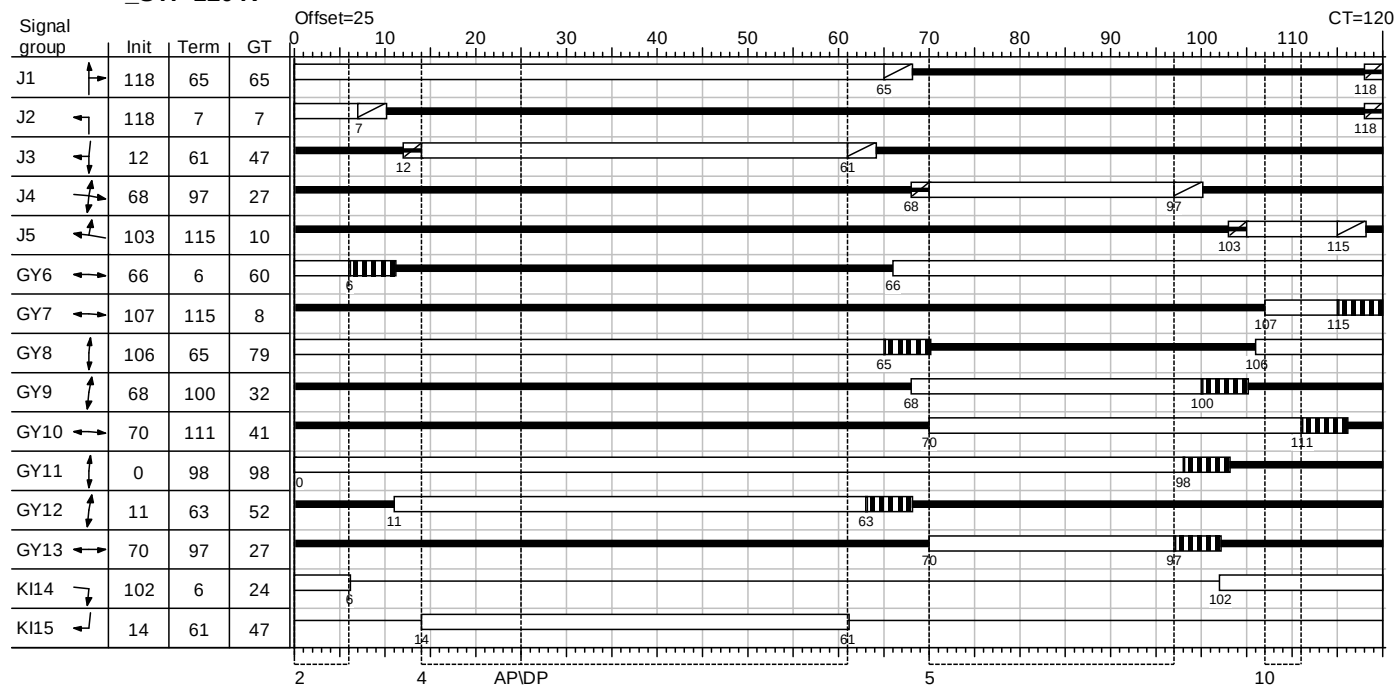
_STP 110 R



Jelzésterv - 4. csomópont - 120 másodperces ciklusidő

LISA+

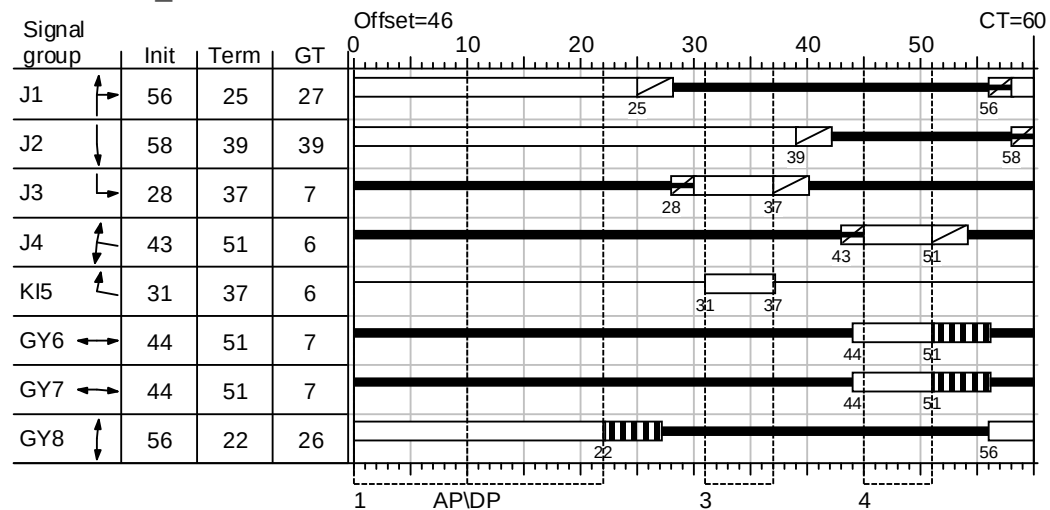
_STP 120 R



Jelzésterv - 5. csomópont - 60 másodperces ciklusidő

LISA+

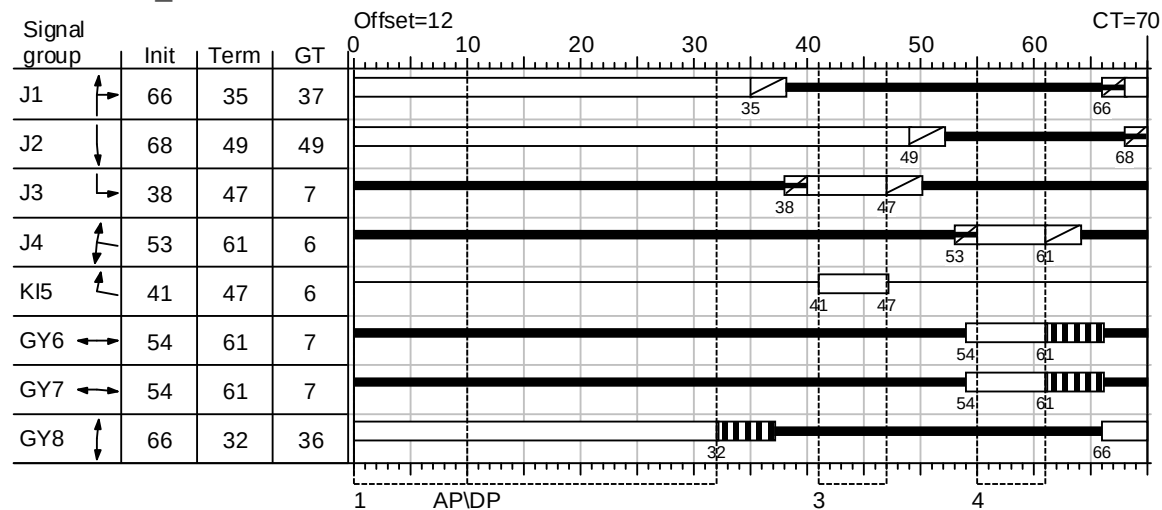
_STP 60 H



Jelzésterv - 5. csomópont - 70 másodperces ciklusidő

LISA+

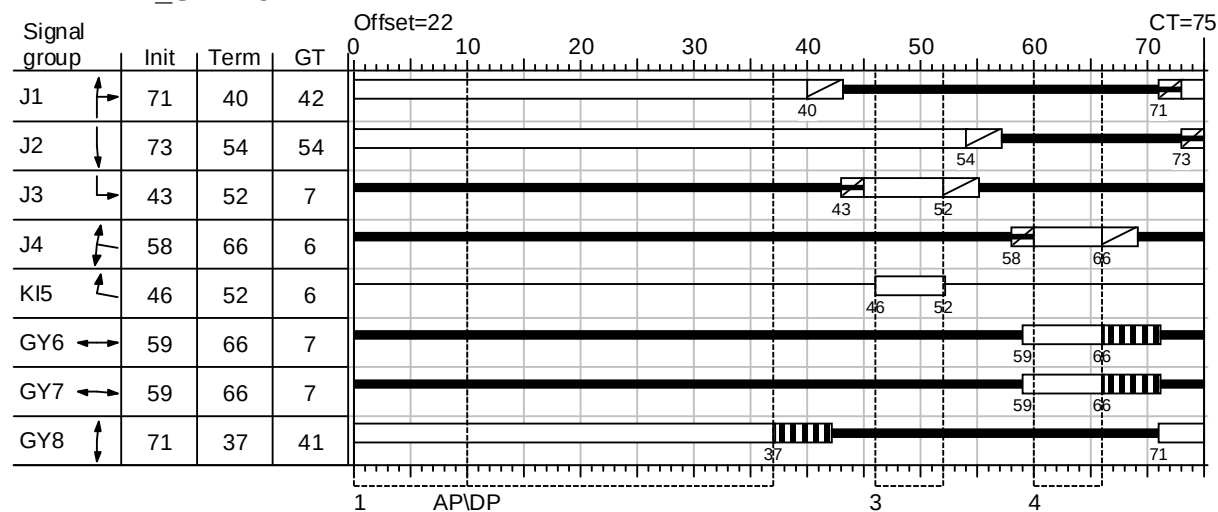
_STP 70



Jelzésterv - 5. csomópont - 75 másodperces ciklusidő

LISA+

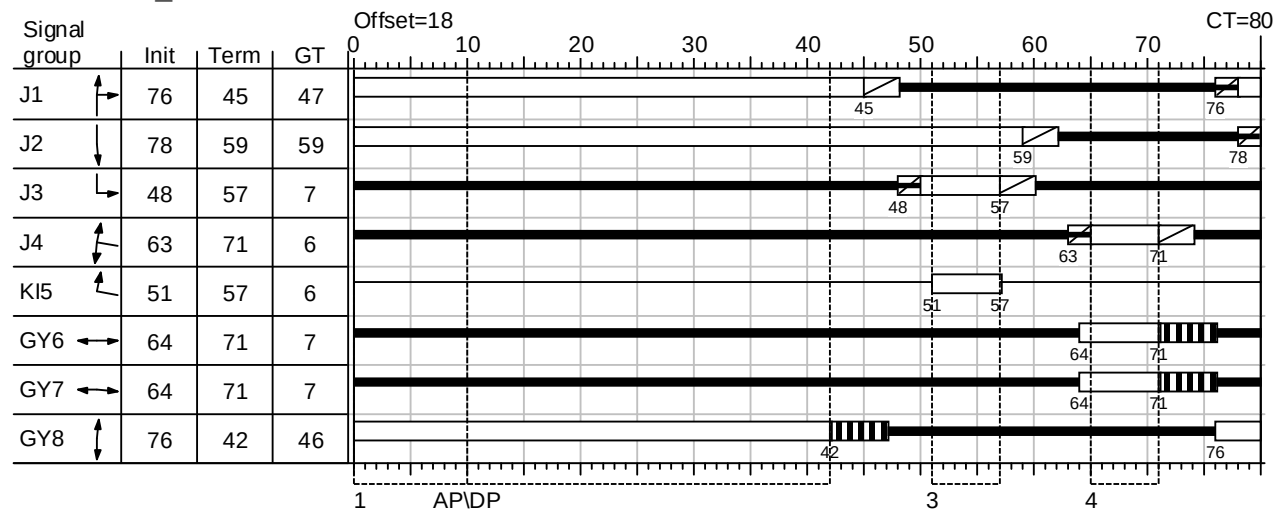
_STP 75



Jelzésterv - 5. csomópont - 80 másodperces ciklusidő

LISA+

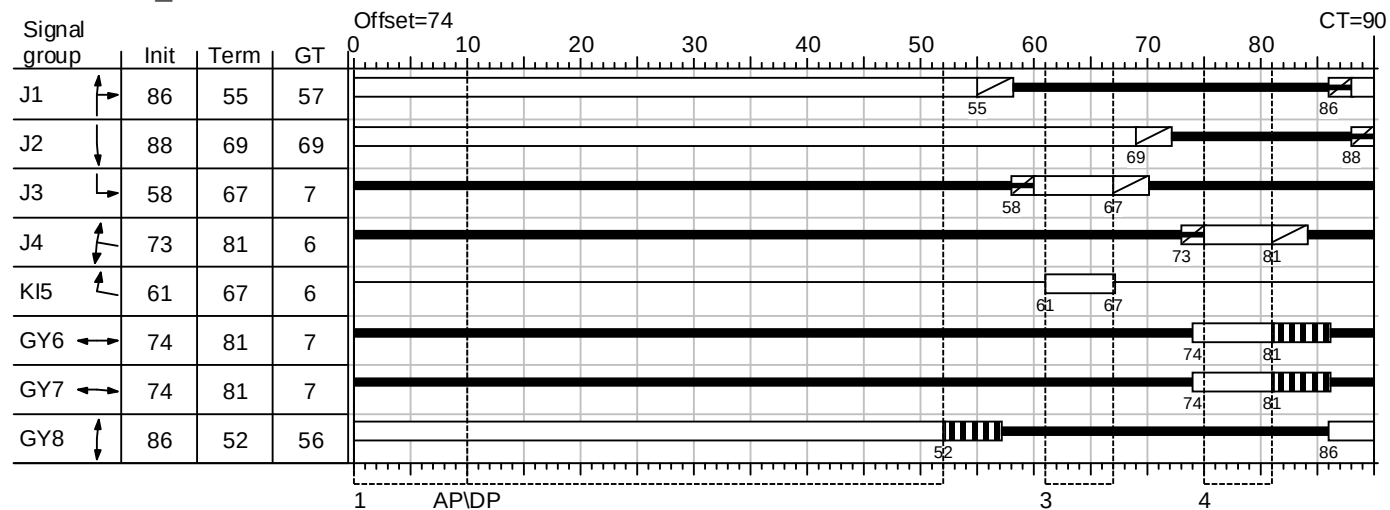
_STP 80 R



Jelzésterv - 5. csomópont - 90 másodperces ciklusidő

LISA+

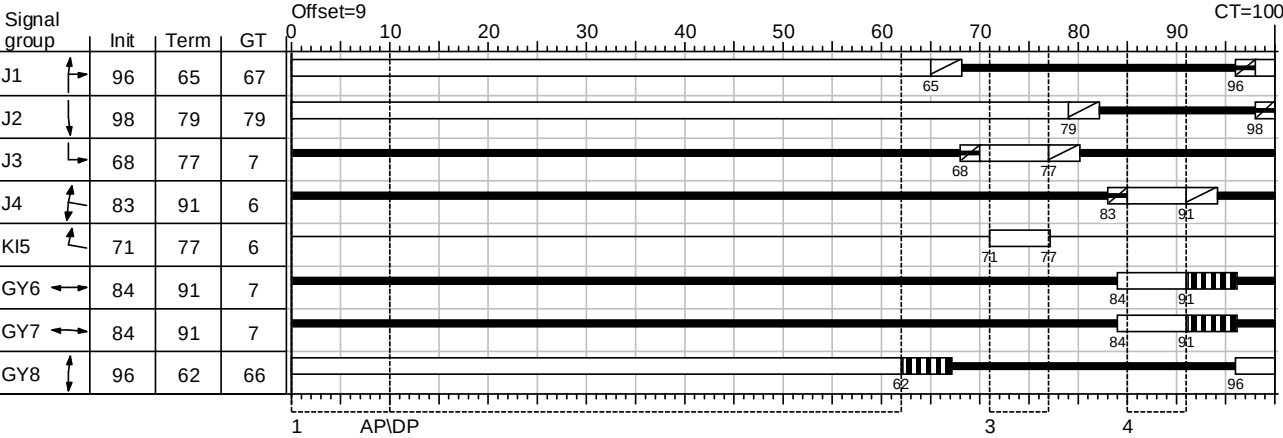
_STP 90 R



Jelzésterv - 5. csomópont - 100 másodperces ciklusidő

LISA+

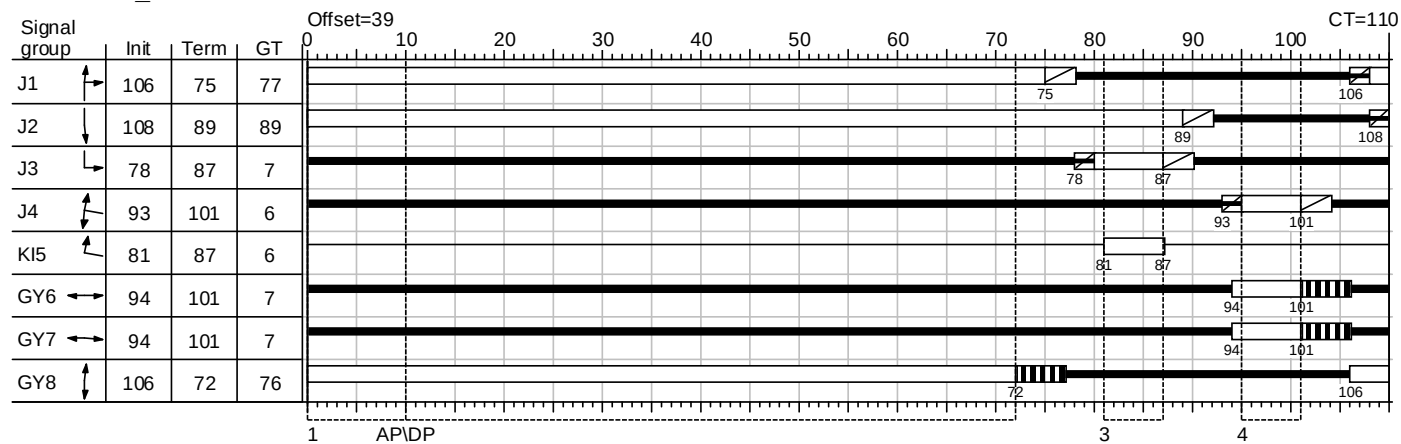
_STP 100 R



Jelzésterv - 5. csomópont - 110 másodperces ciklusidő

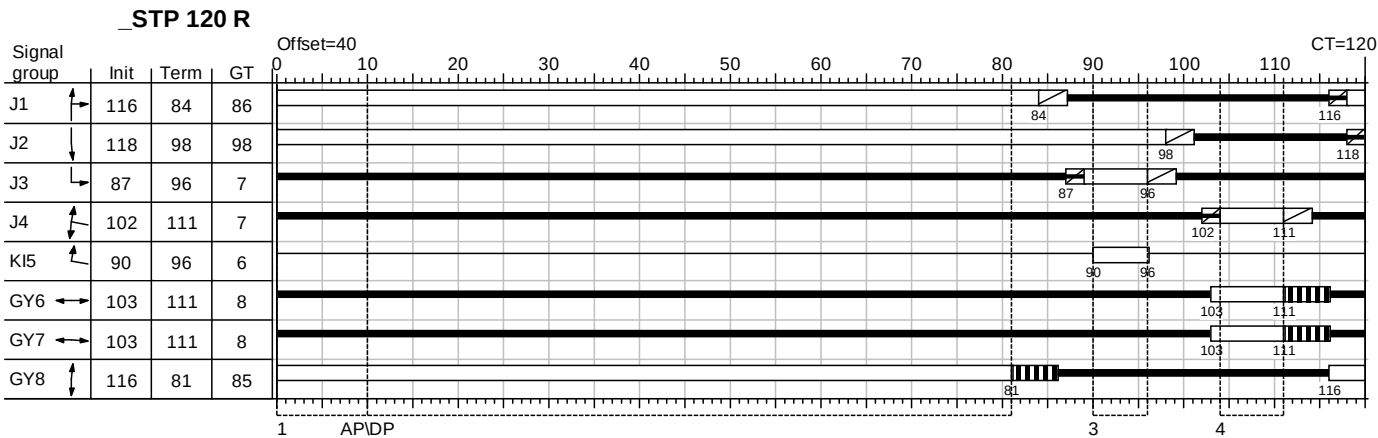
LISA+

_STP 110 R



Jelzésterv - 5. csomópont - 120 másodperces ciklusidő

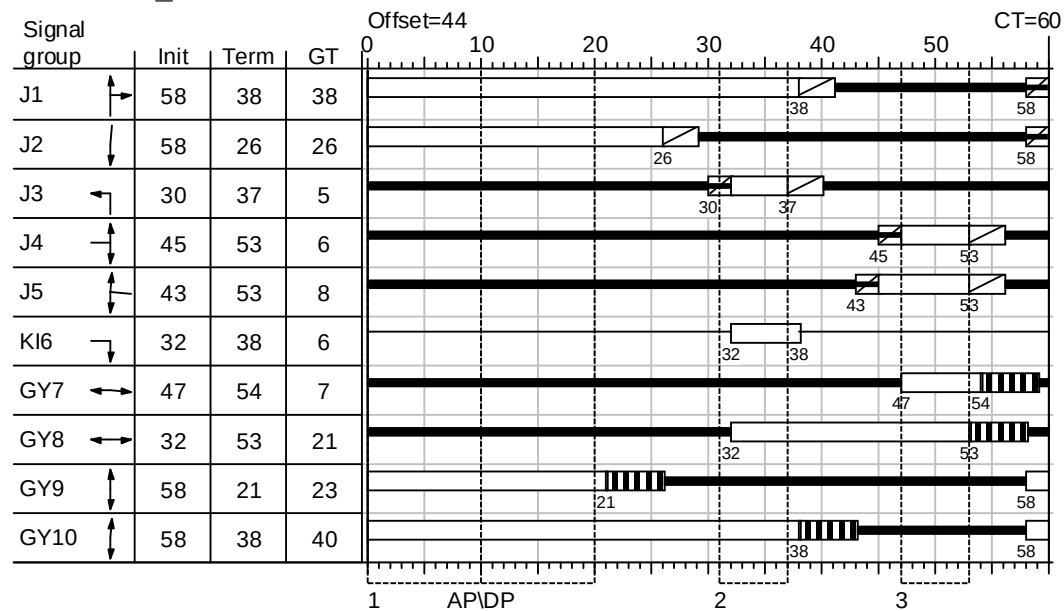
LISA+



Jelzésterv - 6. csomópont - 60 másodperces ciklusidő

LISA+

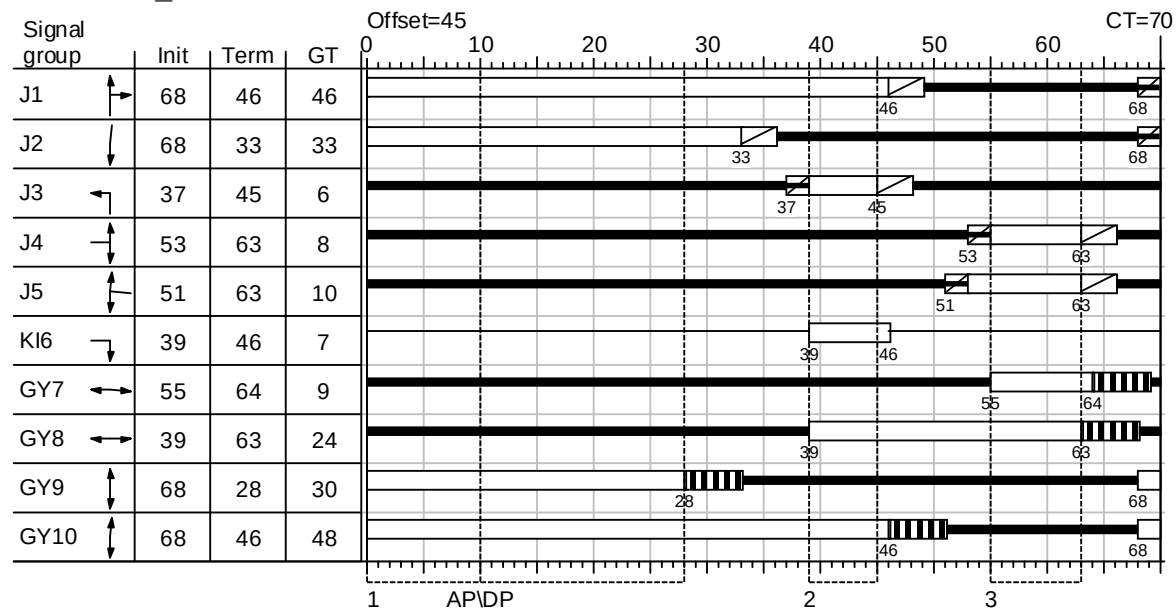
_STP 60 H



Jelzésterv - 6. csomópont - 70 másodperces ciklusidő

LISA+

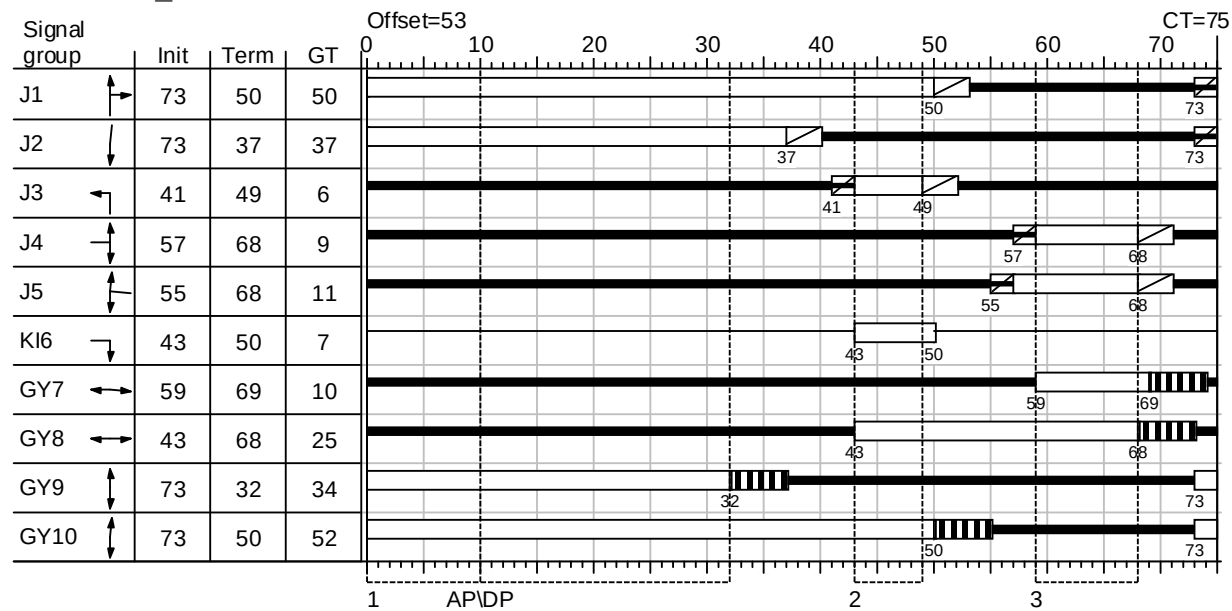
_STP 70



Jelzésterv - 6. csomópont - 75 másodperces ciklusidő

LISA+

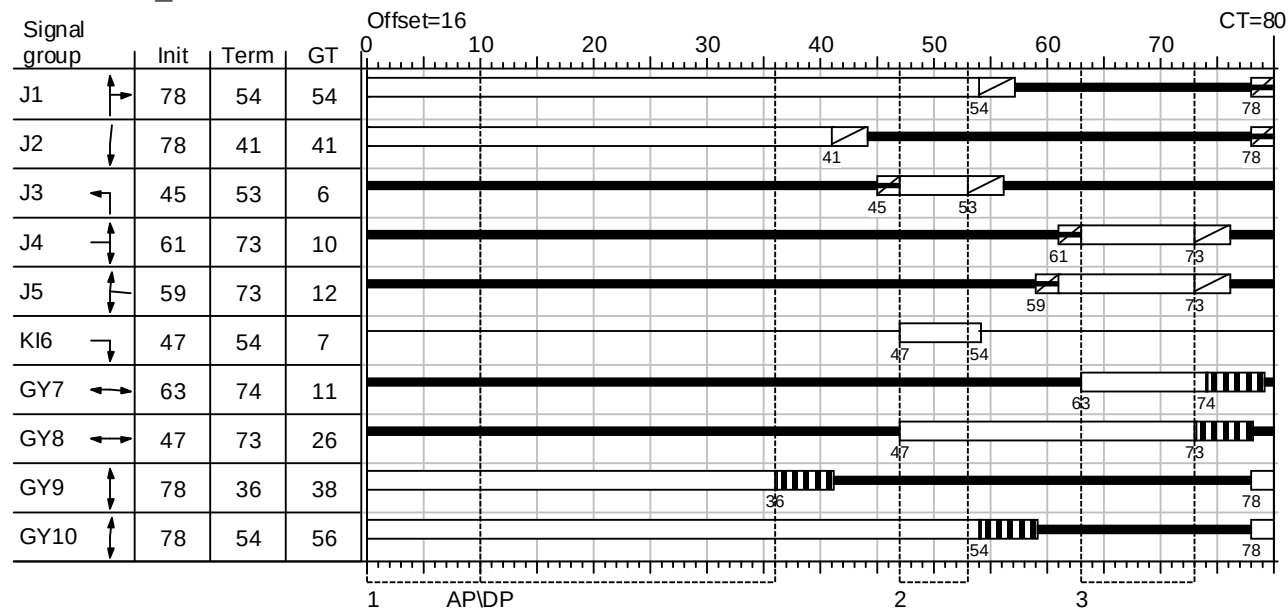
_STP 75



Jelzésterv - 6. csomópont - 80 másodperces ciklusidő

LISA+

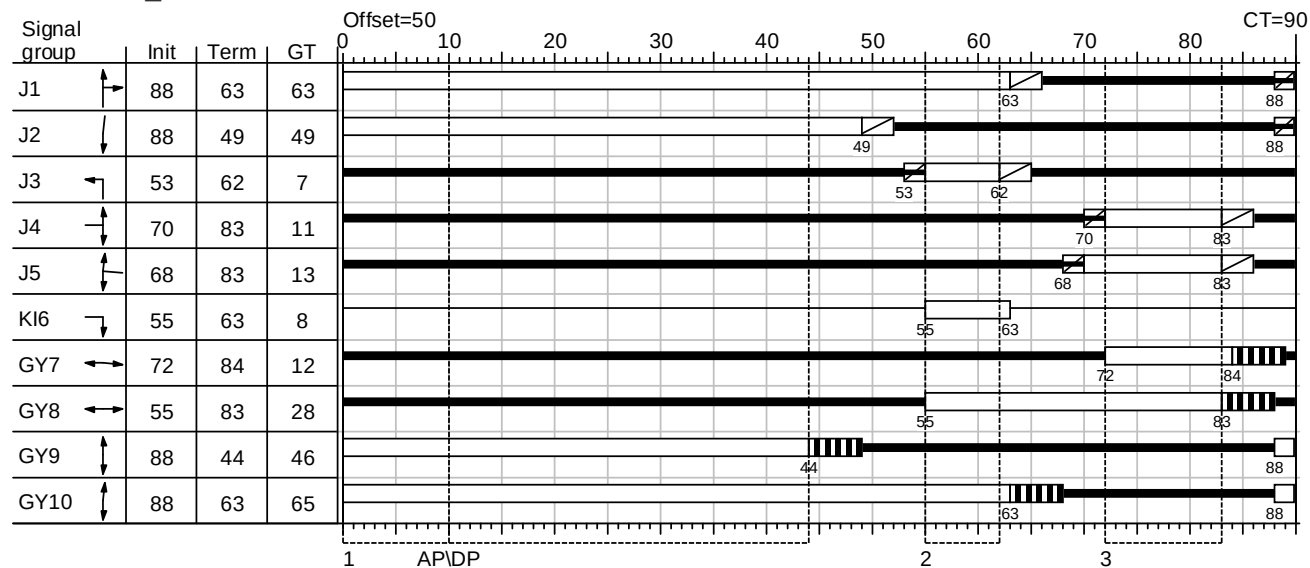
_STP 80 R



Jelzésterv - 6. csomópont - 90 másodperces ciklusidő

LISA+

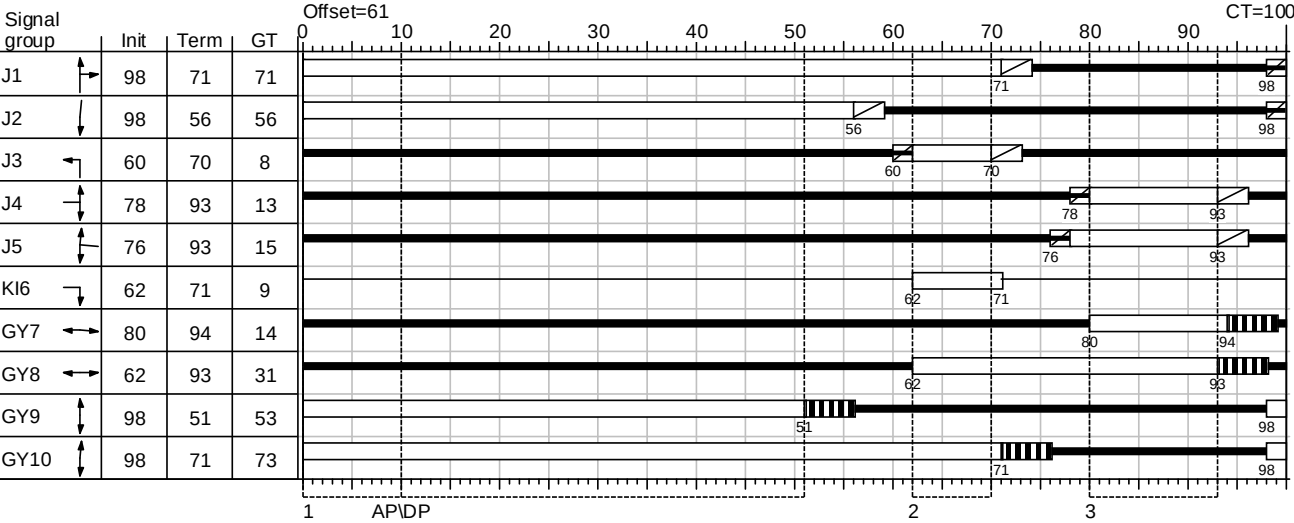
_STP 90 R



Jelzésterv - 6. csomópont - 100 másodperces ciklusidő

LISA+

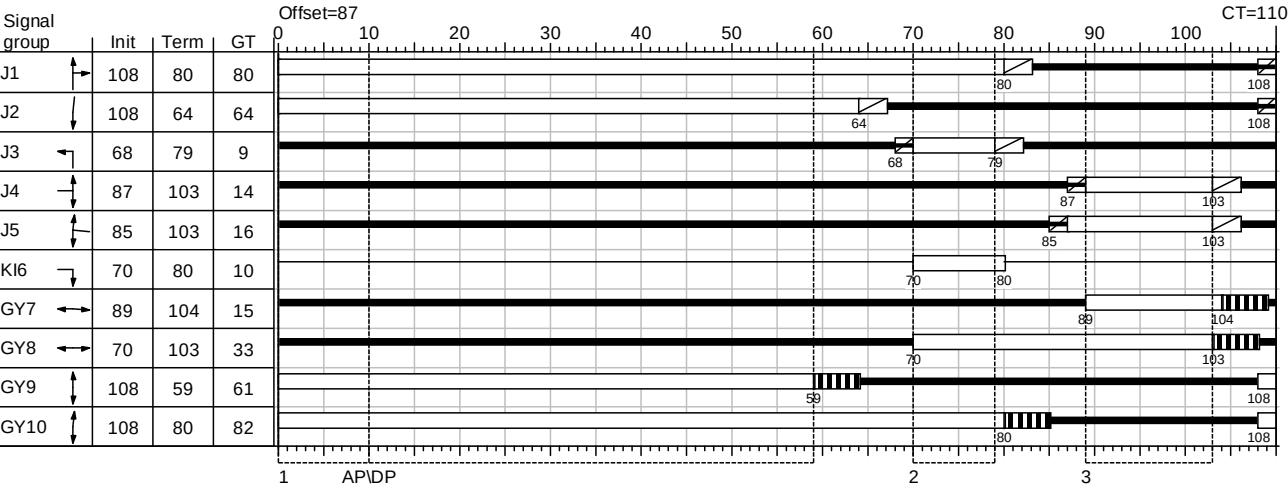
_STP 100 R



Jelzésterv - 6. csomópont - 110 másodperces ciklusidő

LISA+

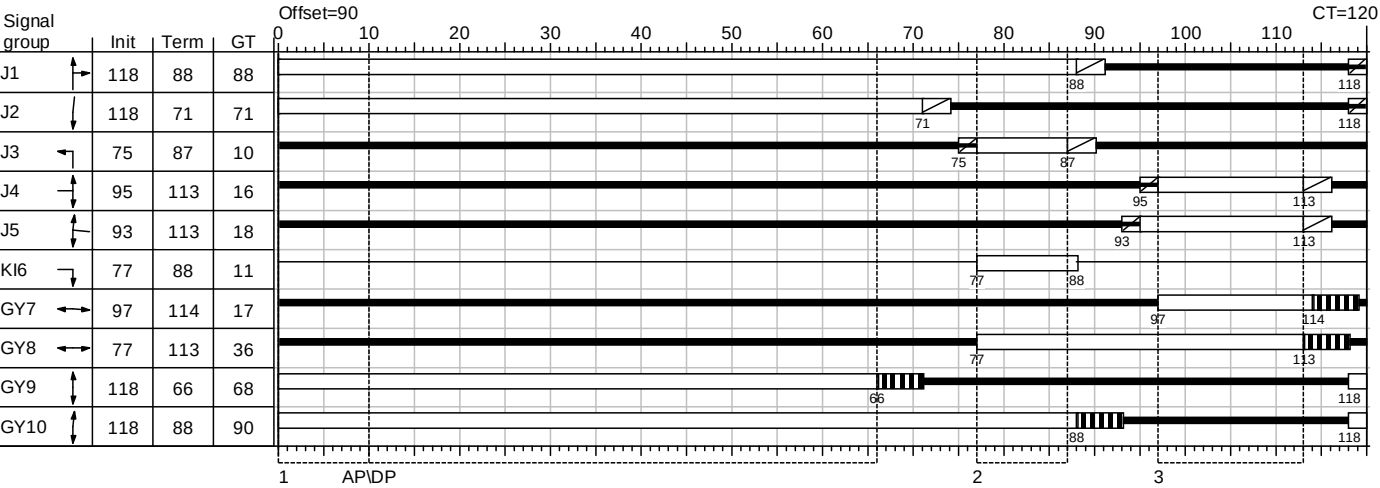
_STP 110 R



Jelzésterv - 6. csomópont - 120 másodperces ciklusidő

LISA+

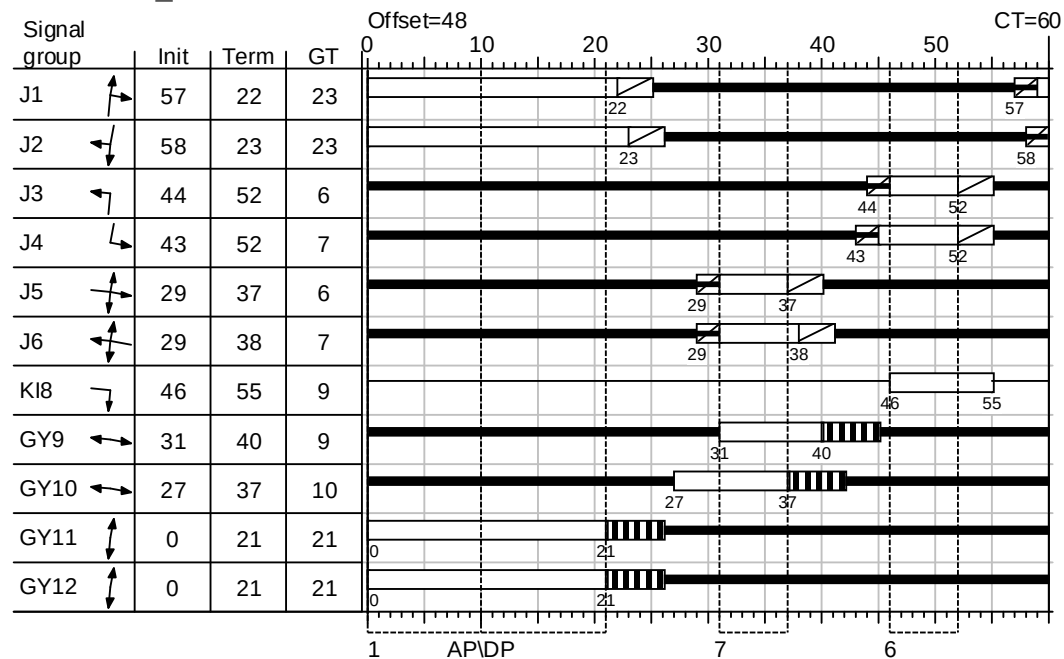
_STP 120 R



Jelzésterv - 7. csomópont - 60 másodperces ciklusidő

LISA+

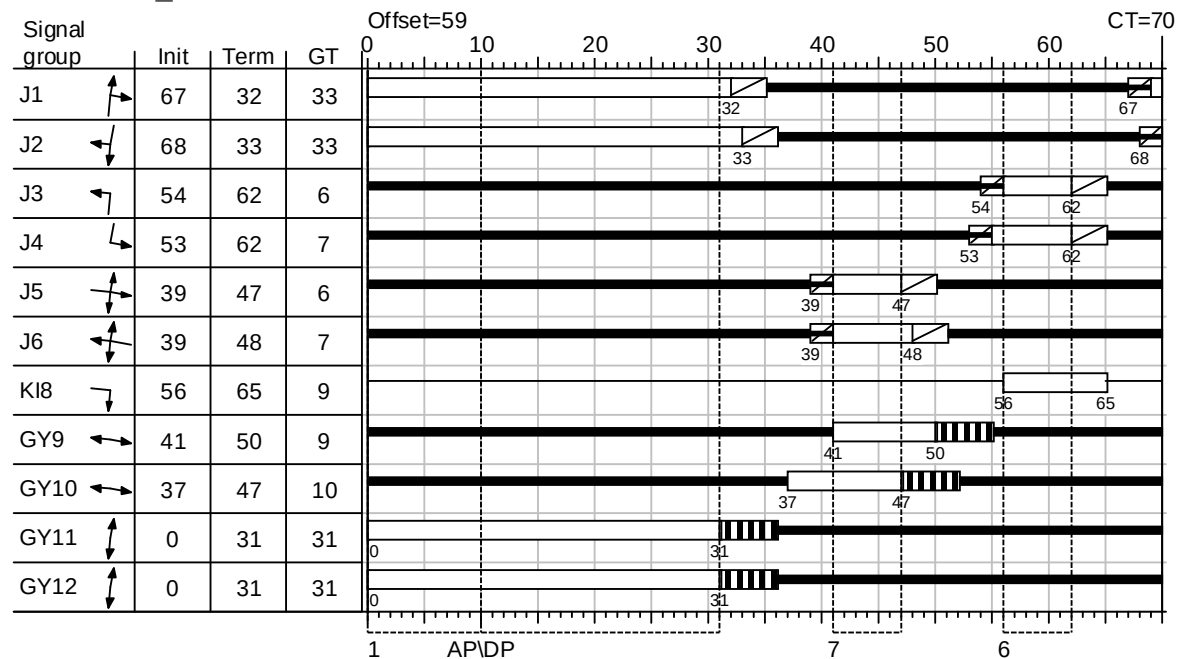
_STP 60 H



Jelzésterv - 7. csomópont - 70 másodperces ciklusidő

LISA+

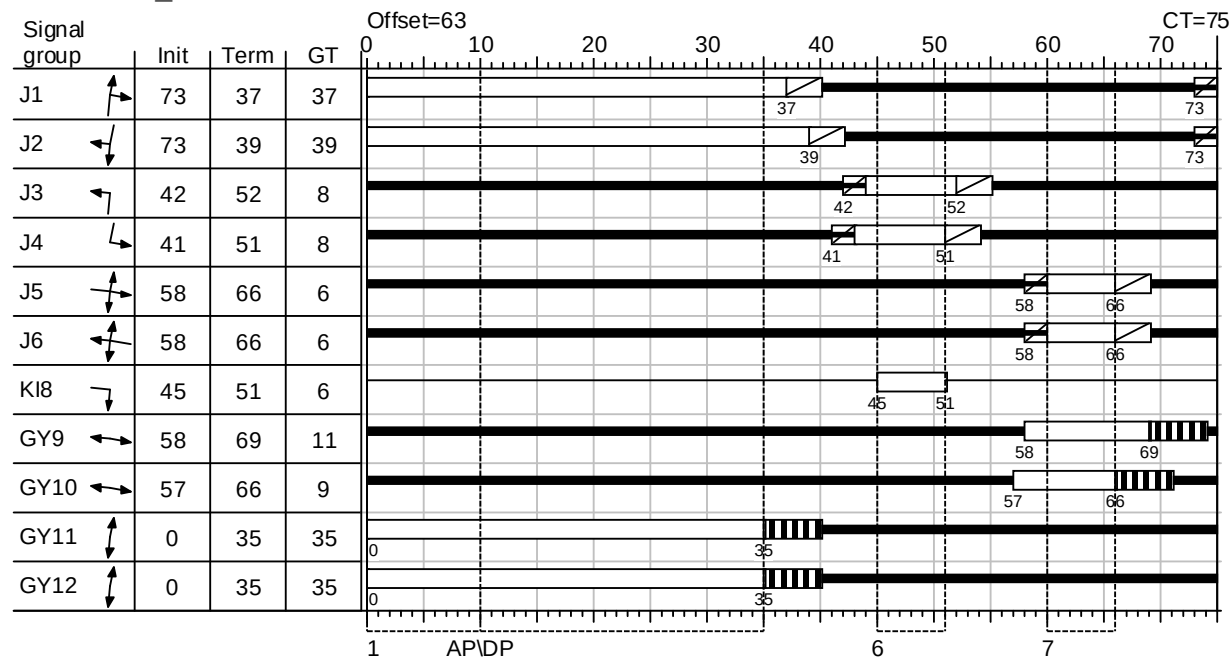
_STP 70



Jelzésterv - 7. csomópont - 75 másodperces ciklusidő

LISA+

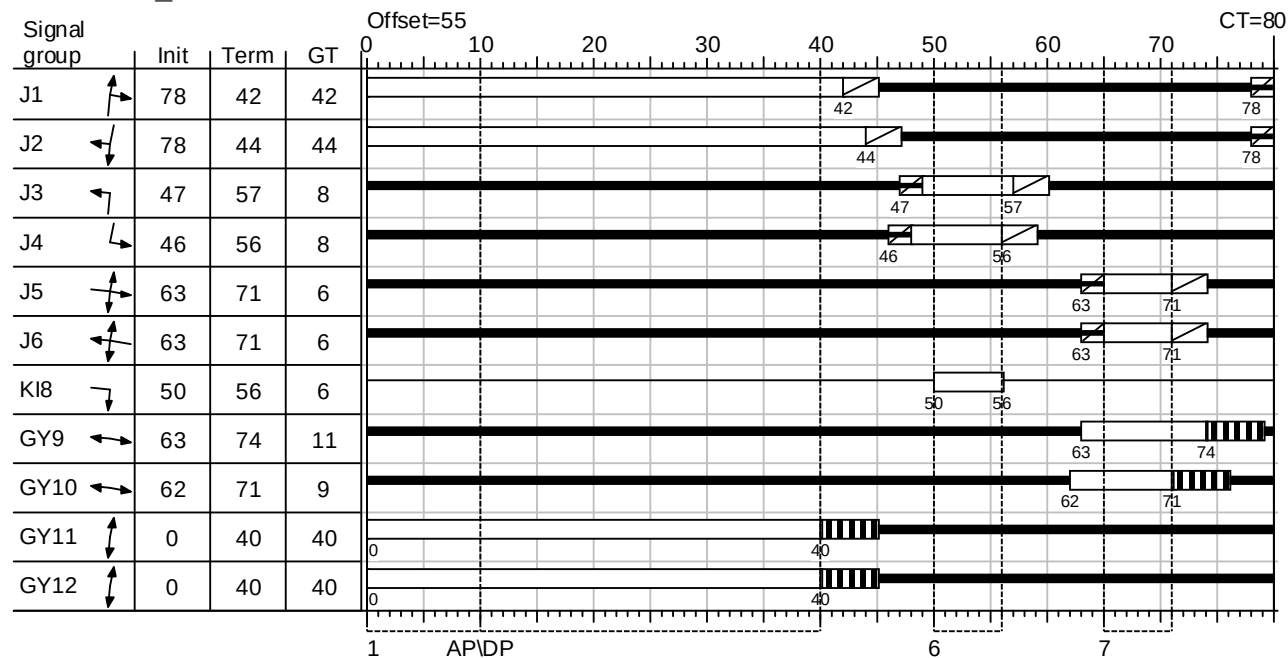
_STP 75



Jelzésterv - 7. csomópont - 80 másodperces ciklusidő

LISA+

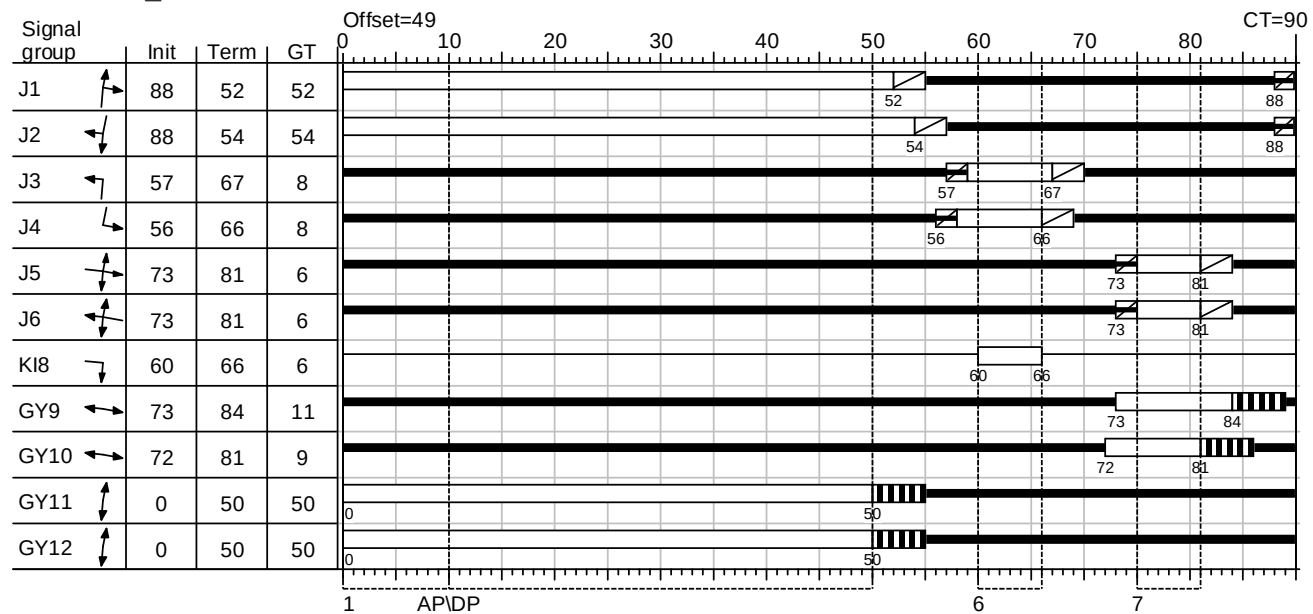
_STP 80 R



Jelzésterv - 7. csomópont - 90 másodperces ciklusidő

LISA+

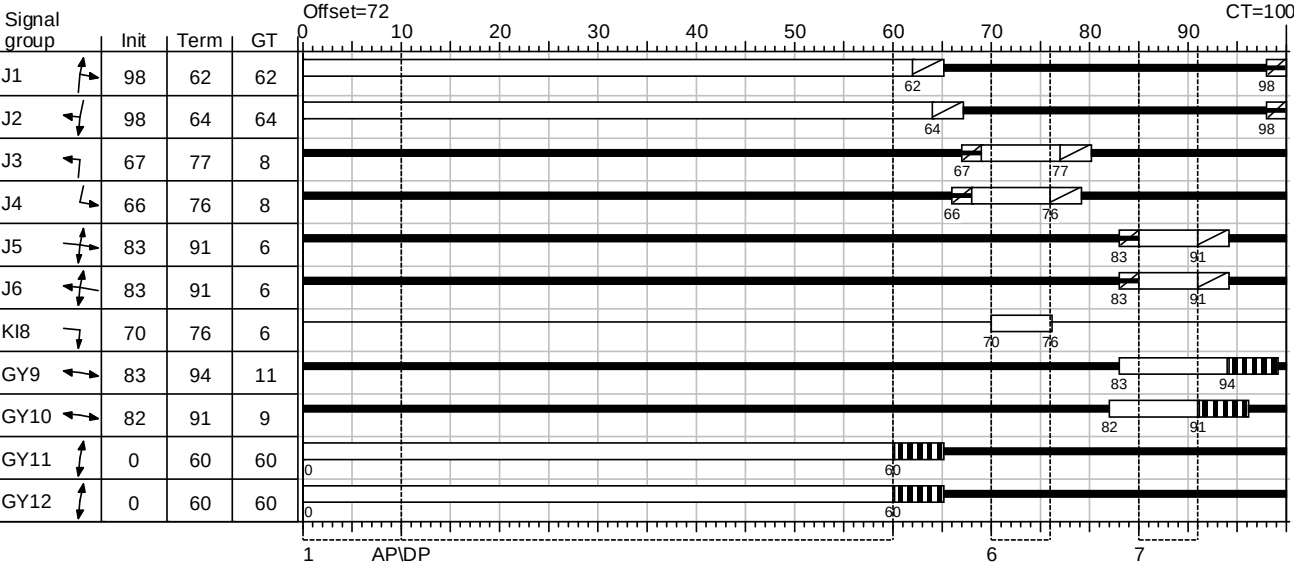
_STP 90 R



Jelzésterv - 7. csomópont - 100 másodperces ciklusidő

LISA+

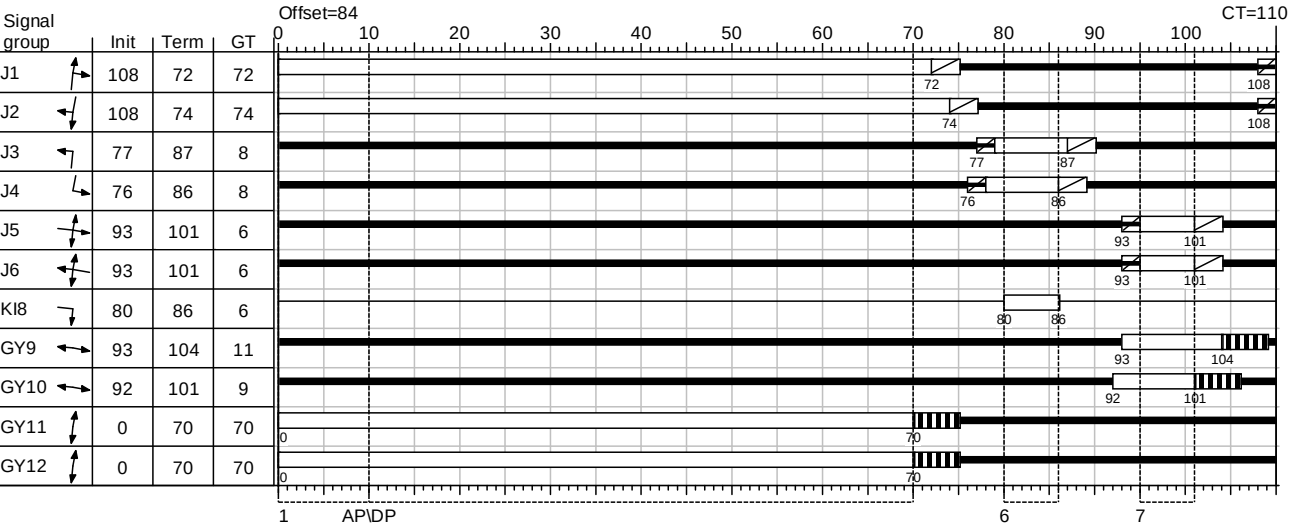
_STP 100 R



Jelzésterv - 7. csomópont - 110 másodperces ciklusidő

LISA+

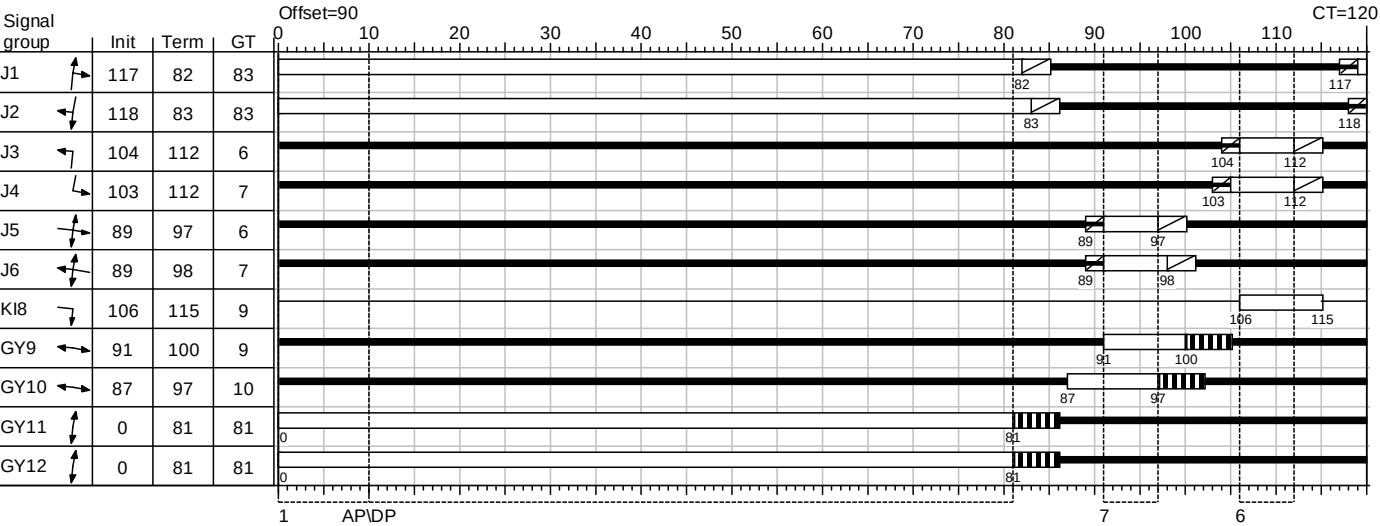
_STP 110 R



Jelzésterv - 7. csomópont - 120 másodperces ciklusidő

LISA+

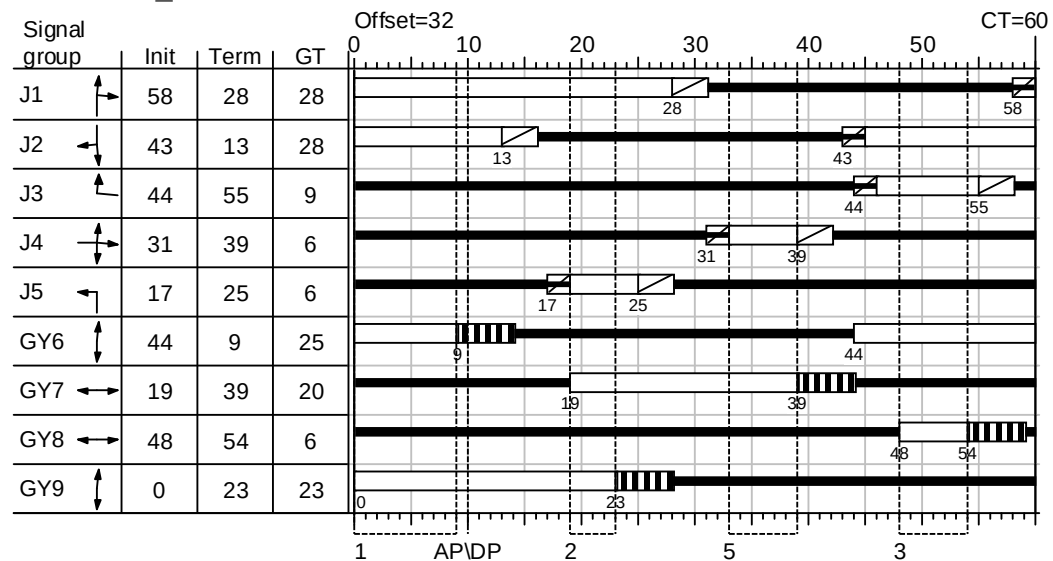
_STP 120 R



Jelzésterv - 8. csomópont - 60 másodperces ciklusidő

LISA+

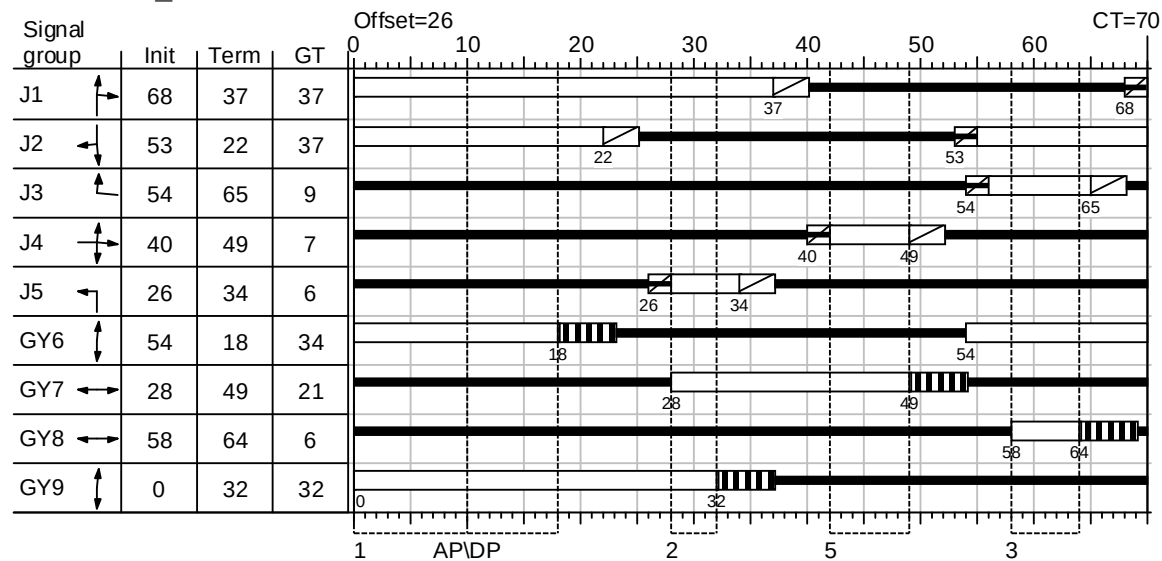
_STP 60 H



Jelzésterv - 8. csomópont - 70 másodperces ciklusidő

LISA+

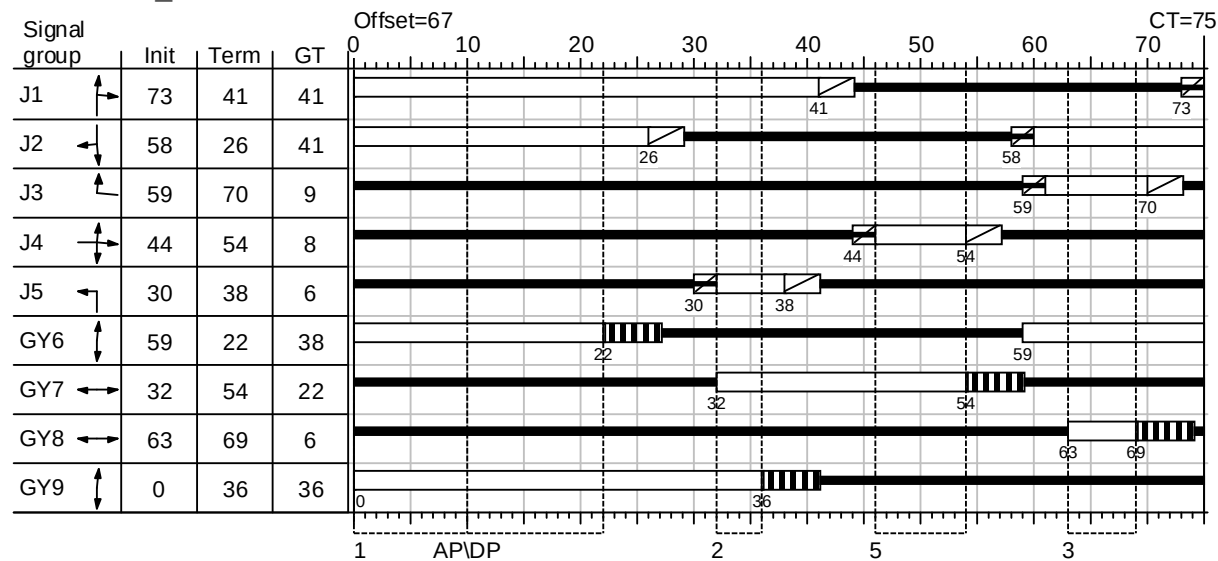
_STP 70



Jelzésterv - 8. csomópont - 75 másodperces ciklusidő

LISA+

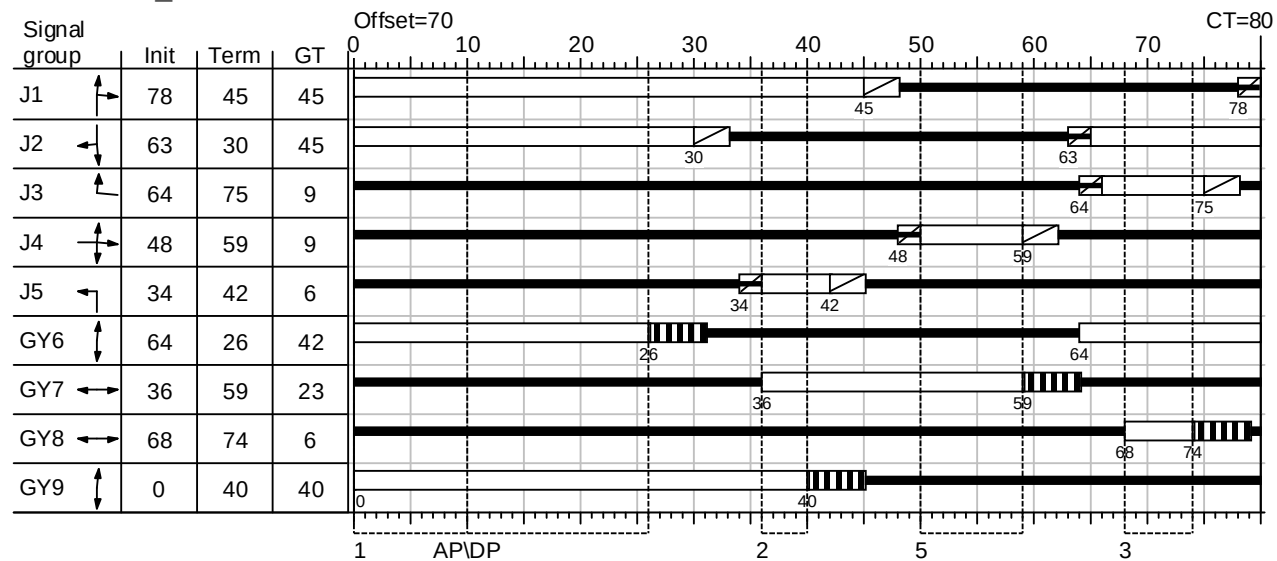
_STP 75



Jelzésterv - 8. csomópont - 80 másodperces ciklusidő

LISA+

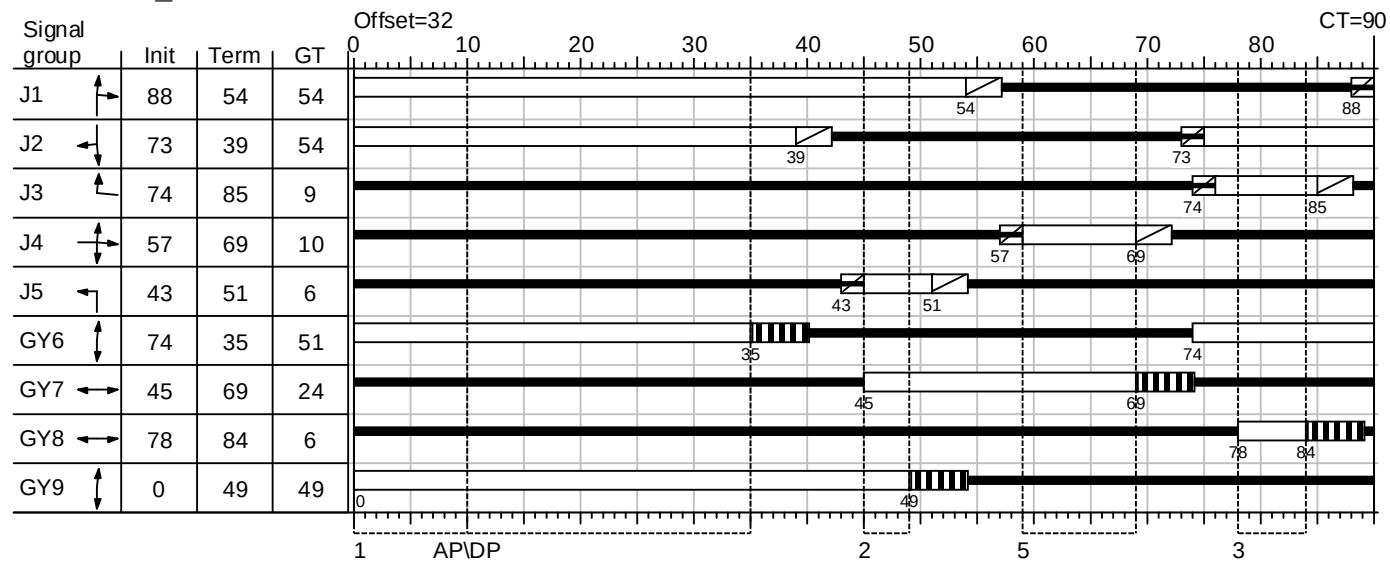
_STP 80 R



Jelzésterv - 8. csomópont - 90 másodperces ciklusidő

LISA+

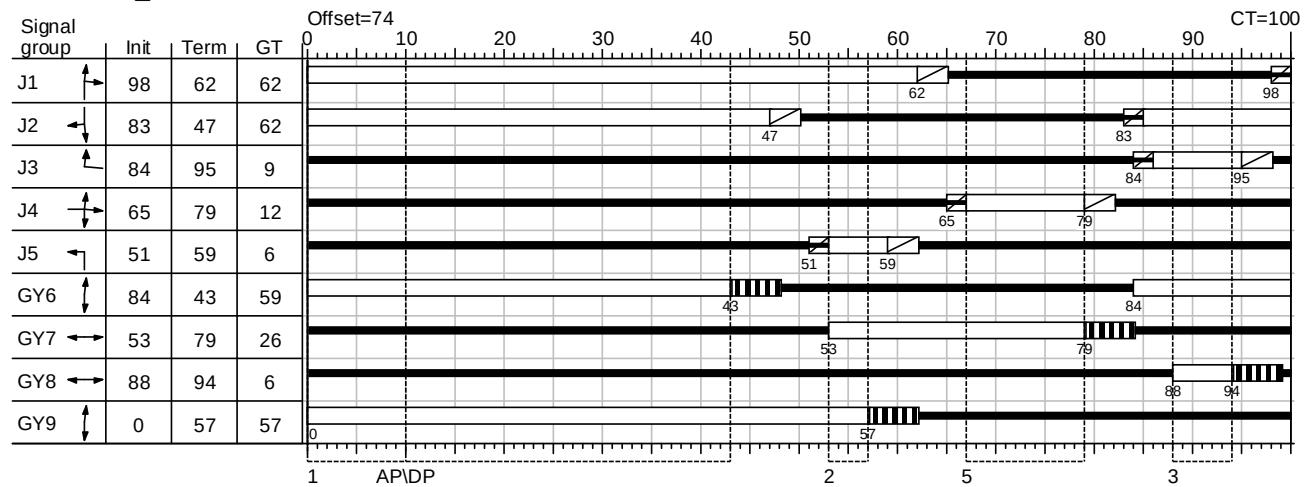
_STP 90 R



Jelzésterv - 8. csomópont - 100 másodperces ciklusidő

LISA+

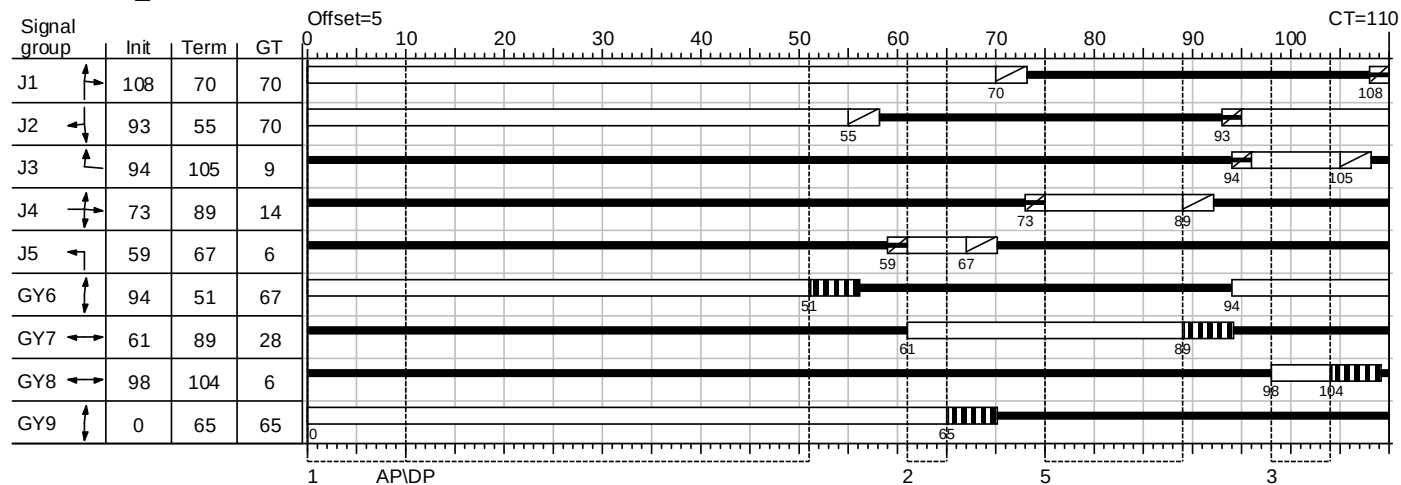
_STP 100 R



Jelzésterv - 8. csomópont - 110 másodperces ciklusidő

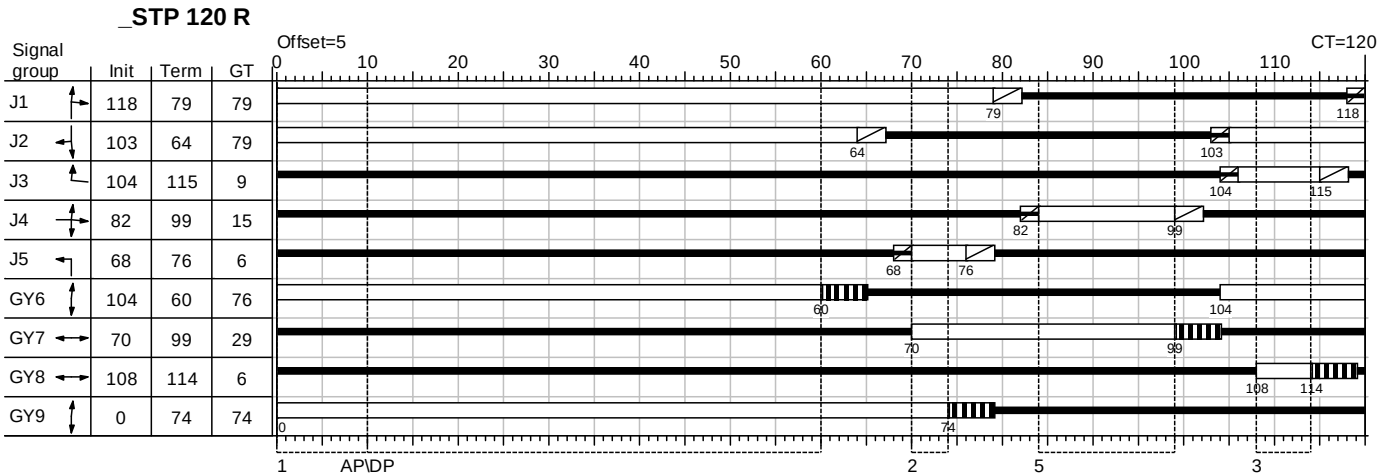
LISA+

_STP 110 R



Jelzésterv - 8. csomópont - 120 másodperces ciklusidő

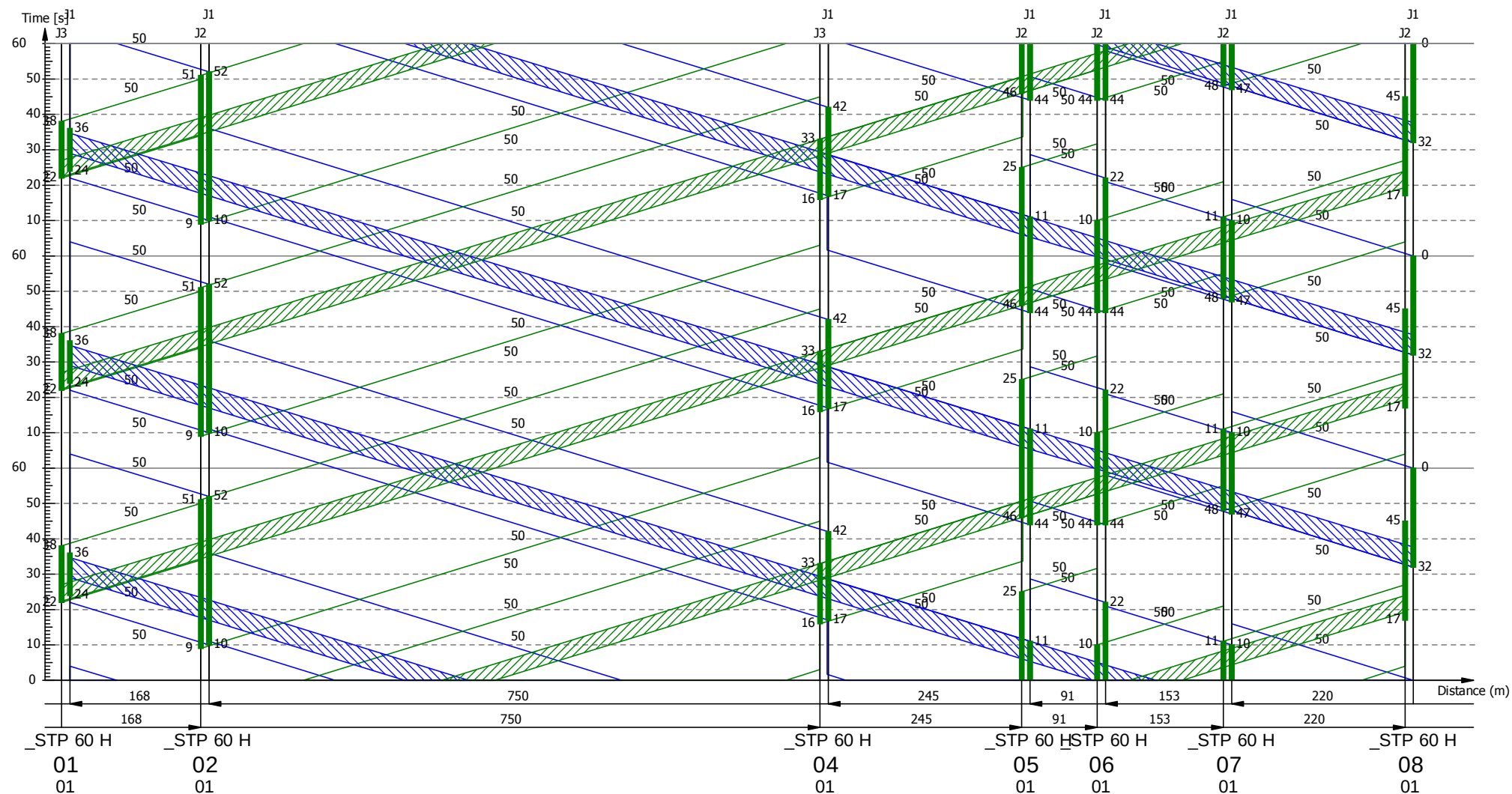
LISA+



Melléklet 6: Hangolási ábrák

Hangolási ábra - 60 másodperces ciklusidő

LISA+



--

--	--

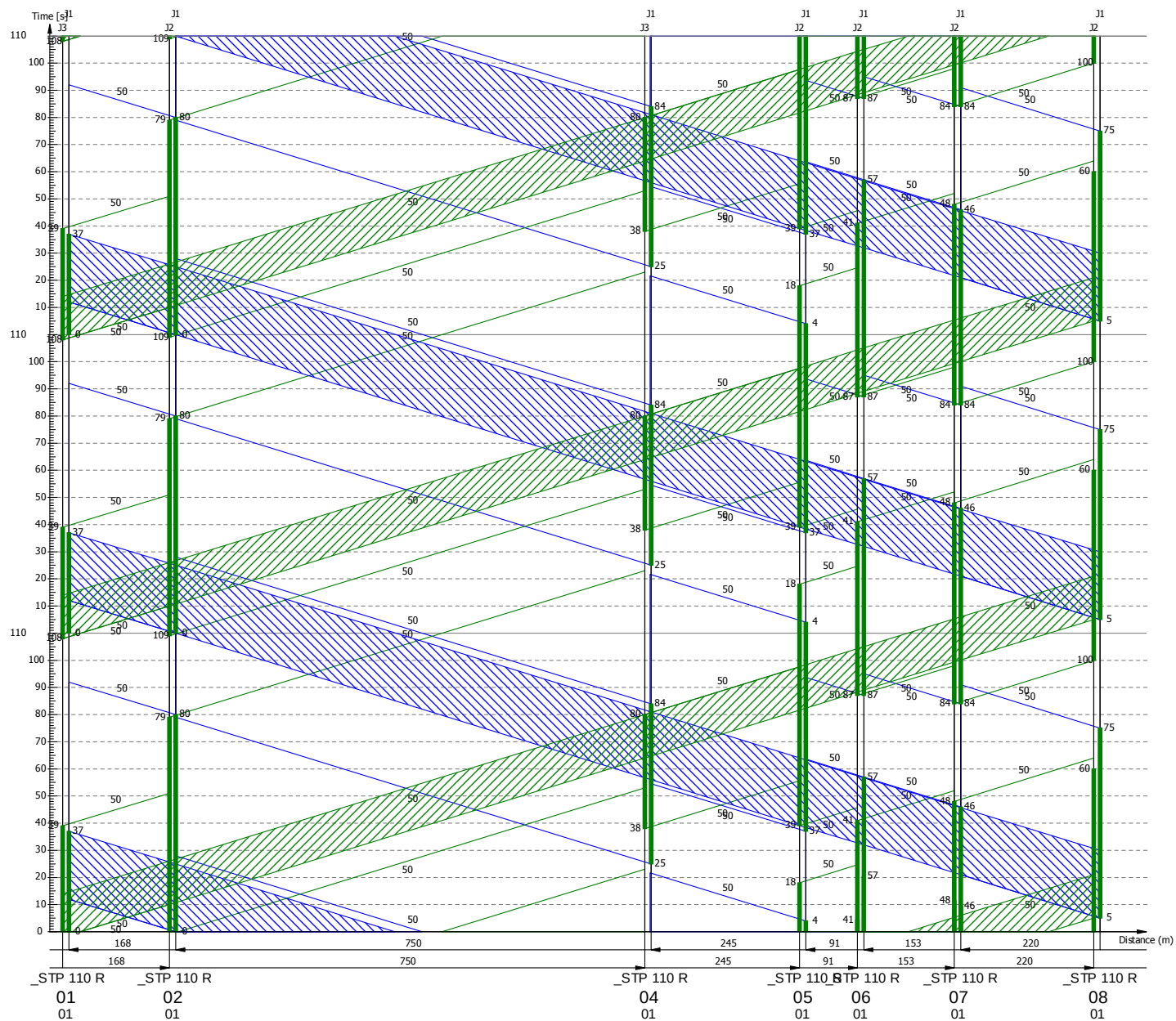
--

--	--

--

Hangolási ábra - 110 másodperces ciklusidő

LISA+



--

Hangolási ábra - Jelenlegi állapot

LISA+

