

# Közúti forgalommodellezési gyakorlatok

## 5. kiadás

**3-5. fejezetek a Vissim és Visum 2023-as verzióihoz aktualizálva**

**6. fejezet a QGIS 3.28-as verzióhoz aktualizálva**

Dr. Horváth Márton Tamás, Dr. Tettamanti Tamás, Dr. Varga István

Lektorálta: Dr. Esztergár-Kiss Domokos

BME, Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék



M Ű E G Y E T E M 1 7 8 2

Budapest, 2024

## Tartalomjegyzék

<b>1. modul: Matlab alapok, gyakorlatok.....</b>	<b>5</b>
1. lecke: A program bemutatása .....	6
2. lecke: Változók kezelése .....	7
3. lecke: Függvények .....	13
4. lecke: Vezérlési szerkezetek.....	15
5. lecke: Grafikus megjelenítés .....	17
6. lecke: Simulink .....	24
7. Irodalom a 1. modulhoz .....	29
<b>2. modul: Forgalommodellezés gyakorlatok Matlabban.....</b>	<b>30</b>
1. lecke: Lineáris járműkövetési modell.....	31
2. lecke: Járműkövetési modell stabilitásvizsgálata .....	39
3. lecke: Makroszkopikus forgalom-modellezés .....	45
4. lecke: Lökéshullámok modellezése az LWR modell segítségével.....	54
5. Irodalom a 2. modulhoz .....	59
<b>3. modul: Vissim mikroszkopikus forgalomszimulátor gyakorlatok .....</b>	<b>60</b>
1. lecke: A Vissim bemutatása.....	61
1. A mikroszkopikus járműkövetési modell.....	61
2. A Vissim járműkövető modellje (Wiedemann modell).....	62
2. lecke: Hálózat felépítése.....	65
1. A jobb egérgomb funkció – objektumok létrehozása a Vissim modellben .....	65
2. Térképháttér behívása .....	65
3. Úthálózat létrehozása .....	67
4. Gyakorlás .....	69
3. lecke: Forgalom generálása .....	70
1. Járművek és gyalogosok beállítása .....	70
2. Vegyes járművezetői viselkedés beállítása – autonóm járműforgalom modellezése .....	71
3. Időintervallumok („Time intervals”) .....	72
4. Forgalmi igények és útvonalak generálása .....	73
5. Gyakorlás .....	76
4. lecke: Elsőbbségi szabályok beállítása.....	77
5. lecke: Jelzőlámpás forgalomirányítás .....	78
6. lecke: Közösségi közlekedés modellezése .....	81

1. Megállóhelyek definiálása.....	81
2. Viszonylatok definiálása .....	82
3. Tömegközlekedési járművek adatainak beállítása.....	85
7. lecke: Vissim futtatási eredmények kiértékelése, videó készítése .....	86
1. Szimulációs futtatások kiértékelése .....	86
2. Szimulációs videók készítése .....	88
8. Irodalom a 3. modulhoz .....	91
<b>4. modul: Vissim forgalomszimuláció COM programozással.....</b>	<b>92</b>
1. lecke: Vissim forgalomszimuláció lehetőségeinek kiterjesztése COM felület segítségével .....	93
1. A Vissim-COM programozás célja .....	93
2. A Vissim-COM programozás alaplépései .....	93
2. lecke: Vissim-COM szerver létrehozása Matlabban.....	97
3. lecke: Vissim-COM metódusok.....	98
4. lecke: Vissim hálózat betöltése .....	100
5. lecke: Általános szimulációs beállítások Vissim-COM programban .....	101
1. Objektumok tulajdonságai .....	101
2. Objektumok attribútumai .....	101
6. lecke: Szimuláció futtatása .....	104
7. lecke: Forgalom generálása .....	105
8. lecke: Jelzőlámpás irányítás, detektorok .....	107
9. lecke: Futtatás közbeni kiértékelés .....	111
4.19. A „DataCollectionMeasurements” mérésre vonatkozó alattribútumok (PTV, 2017).....	112
4.20. Kiértékelési beállítások Data Collection Point esetén (Evaluation configuration) és eredmények (alul) .....	113
10. lecke: Minta Vissim-COM kód .....	114
11. Irodalom a 4. modulhoz .....	115
<b>5. modul: VISUM makroszkopikus forgalomszimulációs gyakorlatok.....</b>	<b>116</b>
1. lecke: A Visum bemutatása .....	117
2. lecke: Hálózat felépítése.....	120
1. A háttér előkészítése és behívása.....	121
3. lecke: Hálózati elemek hozzáadása .....	124
1. Node .....	124

<b>3. Turn</b> .....	125
<b>4. lecke: Zónák kialakítása</b> .....	127
<b>5. lecke: OD mátrix előállítása</b> .....	129
<b>6. lecke: Forgalom ráterhelés a hálózatra</b> .....	132
<b>7. Irodalom az 5. modulhoz</b> .....	134
<b>6. modul: QGIS alapok, gyakorlatok</b> .....	135
<b>1. lecke: Térinformációs rendszerek alapjai</b> .....	136
<b>2. lecke: Vetületi rendszerek</b> .....	141
<b>3. lecke: Vektoros rétegek alapvető műveletei, jellemzői</b> .....	145
<b>4. lecke: Különböző típusú shape-fájlok szerkesztése</b> .....	151
<b>5. lecke: Rétegek megjelenítési beállításai</b> .....	158
<b>6. lecke: Lekérdezések és adatkapcsolatok</b> .....	164
<b>7. lecke: Egyebek: nyomtatás, pluginok</b> .....	172
<b>8. Hasznos linkek, a mintafeladatok forrásai</b> .....	177



# 1. modul: Matlab alapok, gyakorlatok

## Cél:

A fejezet célja a Matlab és beépített környezetének, a Simulink-nek bemutatása és alapvető funkcióinak ismertetése.

## Követelmények:

Ön akkor sajátította el megfelelően a tananyagot, ha képes

- a mátrixokkal definiált műveletek elvégzésére.
- a kapott eredmények ábrázolására.
- egyszerű algoritmusok Matlab-ban történő megvalósítására.
- dinamikus rendszerek szimulációjára Simulink-ben.

## Időszükséglet:

A tananyag elsajátításához körülbelül 4-6 órára lesz szüksége.

## Kulcsfogalmak:

Matlab  
Simulink

## 1. lecke: A program bemutatása

A Matlab ('Matrix Laboratory') egy programozási nyelv és fejlesztői környezet. Alapvetően műszaki számítások elvégzéséhez, algoritmusok fejlesztéséhez és teszteléséhez, szimulációk készítéséhez tervezték. Más matematikai nyelvekkel összehasonlítva (Maple, Mathematica) a Matlab nyelvi szinten csak numerikus számításokat támogat, de a Symbolic Math Toolbox segítségével szimbolikus számításokat is végezhetünk vele.

A Matlab egy mátrixorientált nyelv: alapeleme a kétdimenziós mátrix, ennek speciális eseteként kezeli a skalárértékeket, a sor- és oszlopvektorokat. A mátrix elemei a double változók, melyek a komplex számok mellett két speciális érték (NaN: not a number, definiálatlan érték, inf: infinity, végtelen) megadására is alkalmasak. A hivatkozott változók automatikusan allokalódnak egy ún. workspace-ben.

A Matlab-ban - hasonlóan más interpreteres nyelvekhez - kétféleképpen dolgozhatunk:

- Parancsokat adunk az interpreternek az ún. 'command window'-n keresztül. Ez az interaktív mód, ekkor azonnal választ kapunk az értelmezőtől.
- Az ún. m-fájlok segítségével. Ez a hagyományos, megszokott mód, amikor a felhasználó megírja a programot egy szövegszerkesztő segítségével, és azt később futtatja az interpreterrel. Az m-fájlok Matlab kódot tartalmazó szöveges fájlok .m kiterjesztéssel.
- Mindkét esetben a program válasza a 'command window'-ban jelenik meg, ha nem akarunk megjeleníteni egy parancshoz tartozó választ, a parancs végére pontosvesszőt (;) teszünk.

A különböző szakterületek számítási feladatait számos toolbox segíti, melyek az adott területhez kapcsolódó előre definiált függvényeket, objektumokat tartalmaznak. Függvényeket, objektumokat saját magunk is készíthetünk Matlab alatt, így létrehozhatunk saját toolboxot is. Néhány beépített Matlab toolboxot példaként említünk:

- Math, statistics and optimization
- Signal processing
- Control system
- System identification
- Computational biology
- Data acquisition
- Symbolic math
- Financial

A felsorolt toolboxok sokszínűsége is mutatja, hogy a Matlab egy valóban széles körben használható környezet. Ebben a fejezetben a program használatának alapjait ismertetjük, mellyel a 2. fejezetben található modellezési és szimulációs feladatok megoldása érthetővé válik.

Megjegyzés a program használatához: a Matlab legfontosabb parancsa a help, melyet egy ismeretlen utasítás előtt a command window-ban meghívva pontos leírást kapunk annak használatáról, paraméterezéséről, példával illusztrálva. Így a Matlab használata az alapok elsajátítása után könnyen tanulható autodidakta módon. A program részletes help-je, felhasználói fóruma a [www.mathworks.com](http://www.mathworks.com) oldalon érhető el.

## 2. lecke: Változók kezelése

Az utasítások kétféleképpen nézhetnek ki:

- kifejezés

A kifejezések műveleti jelekből, konstans-, függvény- és változónevekből épülhetnek fel. Ha nem adunk meg változót, akkor a Matlab automatikusan az ans (answer, válasz) változóba rakja bele az eredményt.

pl. `>>sin(2)` az alábbi választ adja:

```
ans=0.9093
```

- változó=kifejezés

Az értékadó kifejezések eredménye egy-egy mátrix lesz, amely értéként hozzárendelődik a megadott változóhoz.

Ekkor, pl. `>>a=2` az alábbi választ adja:

```
a=2
```

- A 'workspace' jellemzői, kezelése:

- a változókat külön nem deklaráljuk: azok az első használatkor deklarálódnak, az interpreter egy megfelelő méretű memóriát foglal le nekik az ún. 'workspace'-ben, és a változók élettartama az interpreter futásának ideje.
- a változók listája a `whos` paranccsal kérhető le.
- egy változó a `clear` paranccsal törölhető, így az általa foglalt memória felszabadul. Az interpreterből történő kilépés során (`quit`) az összes változó elvész.
- adatok mentése, betöltése: a `save` és a `load` segítségével lehet elmenteni és visszatölteni a workspace összes változóját. A tároláshoz a program a `.mat` adatállományt használja.

### 1. Alapértelmezett változók: a mátrixok

A program alapértelmezett változó struktúrája a mátrix, így az alábbiakban áttekintjük a mátrixok definiálásának és hivatkozásának szabályait:

- a változók megadásakor ügyeljünk a betűméretre: a Matlab megkülönbözteti a kis- és nagybetűket:

pl. az `>> A_x=1; a_x=1` parancs két változót definiál:

```
A_x=1  
a_x=1
```

- a mátrixok elemeit szögletes zárójelek között adjuk meg, sorfolytonosan: a soron belüli elemeket vessző, vagy szóköz választja el egymástól, a sorokat pontosvessző választja el:

pl. `>> A=[3 4 5; 1 2 3]`

Ekkor az alábbi mátrixot kapjuk:

$$A = \begin{bmatrix} 3 & 4 & 5 \\ 1 & 2 & 3 \end{bmatrix}$$

- ha az összetartozó szintaktikai egységeket nem tudjuk egy sorban megadni, a sor végén elhelyezett három pont (...) a következő sor értékmegadását folytatólagosan az adott sorhoz kapcsolja:

pl. `a >> A= [ 3 4 5 ...  
1 1; 2 4 3 1 6]`

parancs az alábbi mátrixot adja:

$$A = \begin{bmatrix} 3 & 4 & 5 & 1 & 1 \\ 2 & 4 & 3 & 1 & 6 \end{bmatrix}$$

- a mátrix egy eleme lehet másik mátrix is, így blokkmátrixok megadása is lehetséges:

pl. `a >> A=[1 2; 3 1]`

`M=[A A]`

utasítás az alábbi mátrixot adja:

$$M = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 3 & 1 & 3 & 1 \end{bmatrix}$$

- definiálható az üres mátrix is, a `[]` jelöléssel;
- egy vektor vagy mátrix dimenzióinak lekérdezéséhez a `length` és a `size` utasításokat használhatjuk:

pl. `>>length(M)`

`ans = 4`

`>>size(M)`

`ans = 2 4`

Speciális mátrixok megadása:

- az egységmátrix az `eye(n)` paranccsal adható meg, ahol  $n$  a mátrix dimenziója:

pl. `>> I=eye(2)`

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- egy  $n \times m$ -es null-mátrixot a `zeros(n,m)` paranccsal definiálhatunk:

pl. `>> Z=zeros(3,2)`

$$Z = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- tiszta 1-elemű mátrix a `ones(n,m)` paranccsal definiálható:

pl. `>> N=ones(3,2)`

$$N = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- egy  $(0,1)$  intervallumbeli egyenletes eloszlású véletlen számokból álló mátrix a `rand(n,m)` paranccsal adható meg:

pl. `>> R=rand(2)`

$$R = \begin{bmatrix} 0.9841 & 0.2547 \\ 0.5612 & 0.1768 \end{bmatrix}$$

- ekvidisztáns lépésközű sorozatot megadhatunk vektorral a kezdő indexérték, lépésköz, ill. befejező indexérték definiálásával a következőképpen:

kezdőérték:lépésköz:befejező indexérték

pl. `>> v=1:3:19`

$$v = [1 \ 4 \ 7 \ 10 \ 13 \ 16 \ 19]$$

Mátrixok közötti műveletek:

- az összeadás, kivonás, szorzás a mátrixokra vonatkozó kommutativitási szabályok szerint alkalmazható a `+`, `-`, `*` jelekkel:

pl. az `>> M=[1 2; 3 4]; N=[1 2; 2 1]; P=M*N`

$$P = \begin{bmatrix} 5 & 11 \\ 4 & 10 \end{bmatrix}$$

- transzponált mátrix az aposztróf karakter (') alkalmazásával adható meg:

pl. az >> M=[1 2; 3 4]; M=M'

utasítás válasza:

$$M = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

- egy mátrix inverze az inv() paranccsal adható meg:

pl. az >> M=[1 2; 3 4]; M=inv(M)

utasítás válasza:

$$M = \begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix}$$

- mátrix hatványa a ^ karakterrel definiálható:

pl. >> M=[1 2; 3 4]; M^3

$$M = \begin{bmatrix} 37 & 54 \\ 81 & 118 \end{bmatrix}$$

- a hatványozás végezhető elemenként is, a műveleti jel elé a pont (.) karaktert kitéve:

pl. >> M=[1 2; 3 4]; M.^3

$$M = \begin{bmatrix} 1 & 8 \\ 27 & 64 \end{bmatrix}$$

- a szorzás is végezhető elemenként is, a műveleti jel elé a pont (.) karaktert kitéve:

pl. az >> M=[1 2; 3 4]; N=[1 2; 2 1]; P=M.\*N

$$P = \begin{bmatrix} 1 & 4 \\ 6 & 4 \end{bmatrix}$$

Mátrixelemek hivatkozása, kijelölése (ezek a szabályok a mátrixelemek definiálása során is alkalmazhatók):

- egy mátrix elemeire az indexekkel hivatkozunk, melyet gömbölyű zárójelbe teszünk M(n,m) formában.

pl. `A=[1 4 7; 3 2 8]; A(2,3)` az A23 elemet hívja meg:

`ans=8`

- skalár változó esetében nem kell indexet megadni:

pl. `>> a=8; a`

`ans=8`

- vektorok esetében elég egy indexet megadni:

pl. `>> v=[2 4 6 8]; v(3)` válasza:

`ans=6`

- rövidített kijelölés: ha egy vektor helyére kettőspontot teszünk, akkor az első vektor helyén a mátrix minden sorát, a második vektor helyén pedig mindegyik oszlopát jelöljük ki. pl. `A(:,[1 3])` az A mátrix első és harmadik oszlopát adja meg:

`>> A=[1 4 7; 3 2 8]; A(:,[1 3])`

$$A = \begin{bmatrix} 1 & 7 \\ 3 & 8 \end{bmatrix}$$

- bármelyik vektor helyén megadhatunk intervallumot is, pl. `A(1:2,:)` az A mátrix első kettő sorát emeli ki:

`>> A=[1 4 7; 3 2 8; 2 1 3]; A=(1:2,:)`

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 3 & 2 & 8 \end{bmatrix}$$

## 2. Logikai változók, operátorok

A Matlab a logikai igaz és hamis állítások reprezentálására a 0 és 1 értékeket használja. Természetesen logikai értékek is mátrixként tárolódnak. Létrehozhatunk logikai tömböt pl.

- $x = [true, true, true, false, false]$ ; utasítással,
- vagy a  $x = [10, 45, 78, 34] > 40$ ; kifejezéssel.

A Matlab logikai műveletei:

Logikai operátor	Jelentés
==	egyenlő
<	kisebb
>	nagyobb
<=	kisebb, vagy egyenlő
>=	nagyobb, vagy egyenlő
~=	nem egyenlő
&	elemenkénti logikai és
&&	logikai és (skalárokra)
	elemenkénti logikai vagy
	logikai vagy (skalárokra)
~	elemenkénti logikai tagadás
xor	logikai kizáró vagy



### 3. lecke: Függvények

#### 1. A Matlab beépített függvényei

A Matlab számos beépített függvényt kínál. Ezek elsősorban a toolboxok elemeiként kerülnek a Matlab eszköztárába, de magunk is készíthetünk függvényt. A legtöbb aritmetikai függvény megtalálható a Math toolbox részeként. pl. *sin, cos, sqrt, log, round*, stb.

A függvényeket a következő szintaxissal hívjuk meg:

kimeneti változók=függvénynév(bemeneti változók)

Ügyeljünk a kimeneti és bemeneti változók dimenzióhelyes megadására. Egyes skálár függvényeknek létezik mátrix-mátrixfüggvényként megvalósított verziója is, ezek m-re végződnek. pl. az exponenciális függvény `exp` parancsa helyett alkalmazzuk az `expm` hívást mátrixokra.

pl. Az  $f(x) = e^x$  függvény  $x = 25$ -ben vett helyettesítési értékéhez futtassuk az alábbi kódot:

```
>> x=25;  
e_ad_x=exp(x);
```

A parancsokra az alábbi választ kapjuk:

```
ans= 7.2005e+10
```

Számítsuk ki a  $x \in \mathbb{R}^n$  esetén vett  $g(x) = [e^{x_1} \dots e^{x_n}] \in \mathbb{R}^n$ , vektor-vektorfüggvényt a  $v = [3 \ 2]$  vektorra!

```
>> v=[3 2];  
M=diag(v);           % diagonális mátrixszá alakítjuk a vektort  
e_ad_M=expm(M);  
e_ad_v=diag(e_ad_M) % a mátrix főátlóját vektorra alakítjuk
```

```
ans= 20.0855  7.3891
```

## 2. Saját függvények létrehozása

A függvények voltaképpen speciális .m fájlok. A Matlab számára saját magunk is megadhatunk függvényeket, egy általunk készített .m fájlban, melyet bármikor meghívhatunk az általunk definiált függvénynévvel. A függvények szerkesztésének szintaxisa:

```
function [kimeneti változók]=függvénynév(bemeneti változók)
...
kimeneti változók=...
```

A saját függvényünk működtetéséhez az alábbi szabályoknak kell teljesülniük:

- a kódban szereplő függvénynévnek meg kell egyeznie a függvénykódot tartalmazó .m fájl nevével. (tehát: függvénynév.m-ként mentendő).
- a kód végén valamennyi kimeneti változó értékét meg kell adnunk.
- a függvényünk által használt változók lokálisak (azaz, csak az itt deklarált változókat alkalmazza). Ha egyes változókat szeretnénk hozzáférhetővé tenni más függvények, vagy a parancsablak számára, azt a változó definiálása előtt a global paranccsal tehetjük meg, a következő szintaxissal: global változónév.

Az alábbiakban tekintsünk egy egyszerű példát: készítsünk olyan függvényt, mely megadja egy általunk megadott mátrix sajátértékeinek átlagának négyzetét, és ennek kiszámítja egészrészét. A függvényt mentjük el sajáttertek\_jatek.m néven.

```
function y=sajattertek_jatek(M)
EI=eigs(M) % mátrix sajátértékei
EI_atlag=mean(EI) % sajátértékek átlaga
EI_atlag_negyzet=EI_atlag^2 % az átlag sajátérték négyzete
y=floor(EI_atlag_negyzet) % a keresett egészrész
```

A fenti kódot futtassuk az  $A=[13 \ 22; 0 \ 8]$  mátrixra, azaz hívjuk meg a mentett függvényünket!

```
>> A=[13 22; 0 8];
sajattertek_jatek(A)
```

Ekkor az alábbi válaszokat kapjuk a programtól:

```
EI = 13, 8
EI_atlag = 10.5
EI_atlag_negyzet = 110.25
y = 110
```

## 4. lecke: Vezérlési szerkezetek

A programozás alapvető vezérlési szerkezeteit a Matlab-ban is használhatjuk. Lásd az alábbi példákat!

### if/else/elseif

Az if utasítás először az utána következő logikai kifejezést értékeli ki. Ha a kifejezés értéke igaz, az utána következő utasításokra adja át a vezérlést. Az else kulcsszó után megadhatjuk azokat az utasításokat is, amelyeket akkor hajt végre, ha a kifejezés értéke hamis volt. Használhatunk még elseif-eket is további feltételes végrehajtási utak definiálására. Több feltétel együttes teljesülése esetén csak az első igaz feltétel utáni utasítások kerülnek végrehajtásra. A szerkezetet az end kulcsszó zárja le. A logikai kifejezést mindig lusta módon értékeli ki, azaz pl. A && B esetén a B-t már nem értékeli ki, ha A hamis, ill. A || B esetén B-t már nem értékeli ki, ha A igaz.

pl. az alábbi feltételrendszer a számok 2-vel és 4-gyel való oszthatóságát vizsgálja:

```
if rem(n,2) ~= 0
    disp('n páratlan');
elseif rem(n,4) ~= 0
    disp('n nem osztható 4-el');
else
    disp('n osztható 4-el');
end
```

### switch/case

A switch többirányú elágazást tesz lehetővé egy változó vagy kifejezés értékétől függően. A kifejezés, ami szerint elágazunk skalár vagy string típusú lehet. A case és az otherwise kulcsszavak után írjuk az egyes ágakat, a szerkezetet az end kulcsszó zárja le.

A case kifejezés után szerepelhet skalár, sztring vagy skalárok és sztringek cella tömbje. A Matlab a C nyelvvel szemben megengedi, hogy a case kulcsszavak után string konstansokat, sőt változóneveket írjunk. A switch valódi többirányú elágazás, azaz nincsen szükség a C nyelvből ismert break utasításokra az egyes végrehajtási ágak elhagyásához. A fenti if példa switch változata:

```
switch (rem(n,4) == 0) + (rem(n,2) == 0)
    case 0
        disp('n páratlan');
    case 1
        disp('n nem osztható 4-el');
    case 2
        disp('n osztható 4-el');
    otherwise
        disp('Ez lehetetlen!');
end
```

## for

A for ciklust előre rögzített számú iteráció végrehajtására használhatjuk. Szintaxisa:

```
for (változó) = (vektor)
    (utasítások)
end
```

A ciklus végrehajtásakor a változó a vektor elemein fut végig. A program a vektor egy másolatával dolgozik, azaz a változó által bejárt értékek nem változnak, ha a ciklusmagban módosítjuk a vektor-t. Ha a vektor skalár típusú, akkor a ciklusmag egyszer fut le, ilyenkor a változó értéke a vektor értékével lesz egyenlő. Ha a vektor mátrix típusú, akkor a változó mérete a vektor első dimenziójával fog megegyezni, értékeit pedig a többi dimenzió balról-jobbra bejárása során fogja felvenni.

Például egy  $2 \times 3 \times 4$  dimenziós T mátrixra a változó sorban a T(1,1,1), T(2,1,1), T(1,2,1), T(2,2,1),... értékeket fogja felvenni.

A klasszikus számlálós ciklus megvalósításához így egy névtelen vektort kell definiálnunk. Az alábbi példa 2-től 10-ig kiír minden 3. számot.

```
for i = 2:3:10
    disp(i);
end
```

## while

A while segítségével feltételes ciklusokat definiálhatunk. A while után következő feltétel teljesülése esetén lépünk be a ciklusba. A szerkezetet itt is az end kulcsszó zárja. A következő összetett példa a  $p(x)=x^3-2x-5$  polinom egyik gyökét közelíti intervallumfelezéssel.

```
a = 0; fa = -Inf;
b = 3; fb = Inf;
while b-a > 0.1*b
    x = (a+b)/2;
    fx = x^3-2*x-5;
    if sign(fx) == sign(fa)
        a = x; fa = fx;
    else
        b = x; fb = fx;
    end
end
```

## 5. lecke: Grafikus megjelenítés

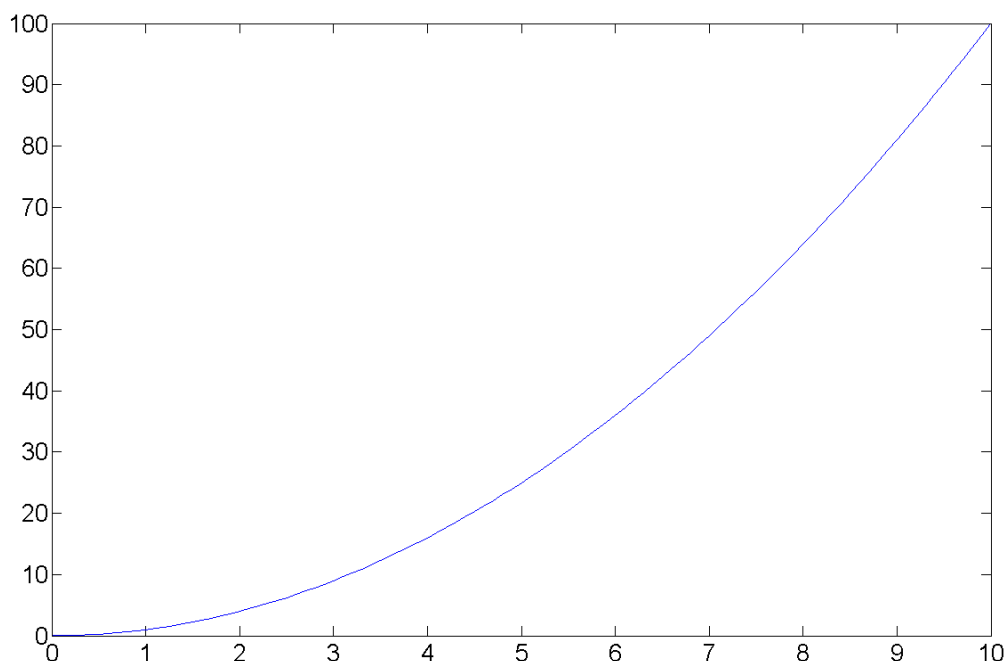
A Matlabban végzett számítások és szimulációk eredményét célszerű grafikusan is megjeleníteni. Mivel a Matlab egy alapvetően numerikus számításokra kitalált szoftver, ezért a függvény rajzolás diszkrét értékek mellett, numerikusan valósítható meg: tehát voltaképpen mátrixok elemeit rajzoljuk ki, a mátrix indexeihez rendelve az értelmezési tartomány és értékkészlet felvett elemeit.

### 1. Egyváltozós függvények rajzolása

A plot parancs szolgál az egydimenziós vektorok megjelenítésére, így ezt használjuk az egyváltozós függvények rajzolásához is. A parancs argumentumai a grafikon megjelenítésének igen széleskörű beállítását teszik lehetővé. A parancs alap szignatúra a `plot(x,y)`, ahol `x` az `x` tengely elemeit tartalmazza, `y` pedig az `x`-hez tartozó függvényértékeket. Amennyiben csak `plot(y)` formában hívjuk meg a függvényt akkor az `x` tengely az indexeket fogja tartalmazni. Alapértelmezetten a Matlab az adatokat egy összefüggő kék vonal segítségével jeleníti meg.

pl. Ábrázoljuk az  $f(x)=x^2$  függvényt az  $x=[0,10]$  értelmezési tartományon! Az alábbi kód felhasználásával az 1.1. ábrán látható diagramot kapjuk:

```
x=0:0.1:10;  
y1=x.^2;  
plot(x,y1)
```



1.1. ábra: az  $f(x) = x^2$  függvény ábrázolása

Az alapértelmezett plot vonalrajzoló parancs további utasításokkal egészíthető ki:

A diagram négyzethálóját a grid paranccsal adhatjuk meg: `plot(x,y),grid`.

Egyazon ábrán újabb függvény ábrázolása a hold on paranccsal lehetséges.

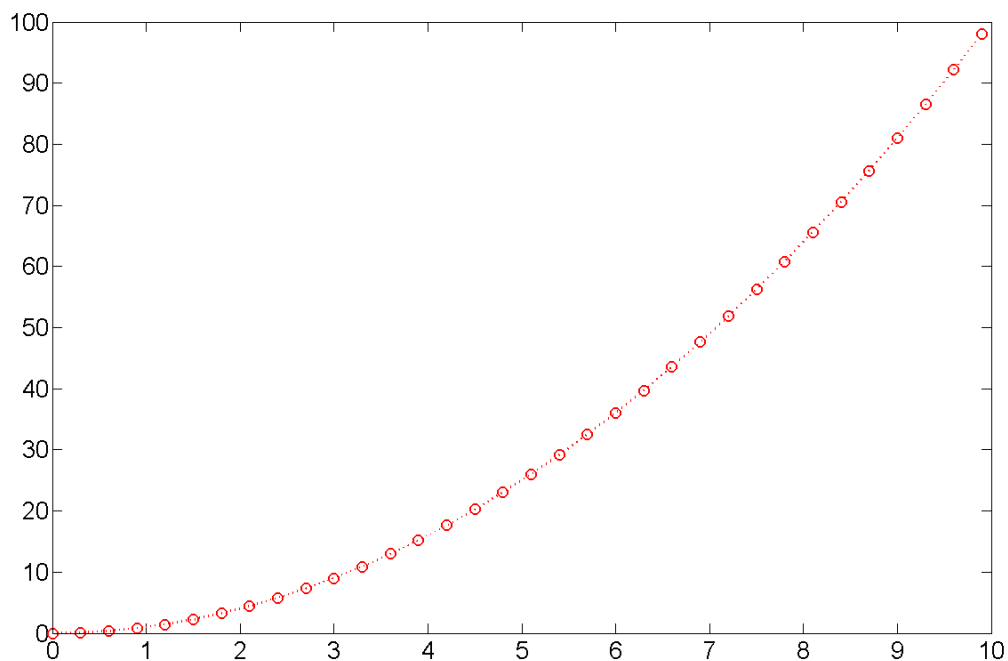
A tengelyeknek feliratot az xlabel, ylabel, a vonalaknak címet a legend paranccsal adhatunk.

## 2. Megjelenítési stílus beállítása

További paraméterként vonal specifikációkat adhatunk meg, mellyel állítható a vonal színe, vastagsága, mintája, valamint az egyes adatpontok színe, alakja és mérete. A vonal stílusára vonatkozó beállítások a megjelenítendő értelmezési tartomány, értékészlet argumentumok után adhatóak meg.

pl.:

```
x=0:0.3:10  
y=x.^2;  
plot(x,y,'or')
```



1.2. ábra: az  $f(x) = x^2$  függvény egy másik ábrázolása

A fenti példában is látható a vonal tulajdonságait egy string segítségével adhatjuk meg (1. 2. ábra) Az első karakter a vonal tulajdonságait adja meg (a : szimbólum a pontozott vonalhoz vezet), a következő karakterek az adatpontok tulajdonságait (az o szimbólum az adatpontok körrel való kiemelését adja), az utolsó karakter (jelen esetben r) pedig a színt.

### A pont, a vonal és a szín beállítási lehetőségei:

- Az alábbi táblázatban szereplő szimbólumokkal a pont stílust adhatjuk meg:

szimbólum	pont stílus
'+'	kereszt
'o'	kör
'*'	csillag
'.'	pont
'x'	X
's'	négyzet
'd'	rombusz
'^', 'v', '<'	háromszögek
'p', 'h'	öt- és hatágú

- Az alábbi táblázat az egyes vonalstílusokat előidéző szimbólumokat foglalja össze:

szimbólum	vonal
'_'	sima
'--'	szaggatott
'...'	pontozott
'-.'	pontozott

- A gyakran használt színek parancsait az alábbi táblázat összegzi:

szimbólum	szín
'r'	piros
'g'	zöld
'b'	kék
'c'	cián
'm'	magenta
'y'	sárga
'k'	fekete
'w',	fehér

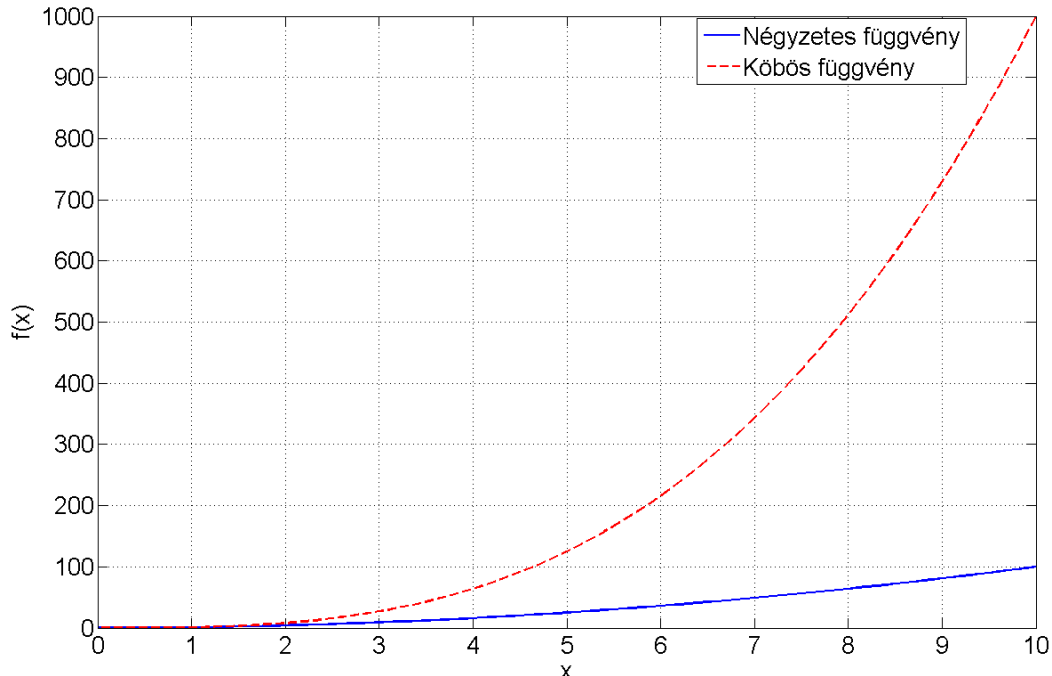
A fentiekén kívül természetesen bármilyen egyéb szín alkalmazható, ha az RGB kódját egy 3 elemű vektorként adjuk meg.

- Ezen kívül még további tulajdonságokat is megadhatunk a vonal és pont stílusokban az alábbi szintaxissal: 'parancs', adat. A parancsokat az alábbi táblázat foglalja össze:

parancs	tulajdnonság
LineWidth	vonalvastagság
MarkerEdgeColor	pont
MarkerFaceColor	pont kitöltő színe
MarkerSize	pont mérete

A felsorolt beállítási lehetőségek bemutatásához futtassuk le az alábbi kódot (melynek eredményeként kaphatjuk az 1.3. ábrán látható diagramot):

```
x=0:0.1:10;
y1=x.^2;
y2= x.^2;
plot(x,y1,'LineWidth',2),grid
hold on
plot(x,y2,'r--','LineWidth',2)
xlabel('x')
ylabel('f(x)')
legend('Négyzetes függvény','Köbös függvény')
```



1.3. ábra: az  $f(x) = x$  és  $g(x) = x^3$  függvények ábrázolása



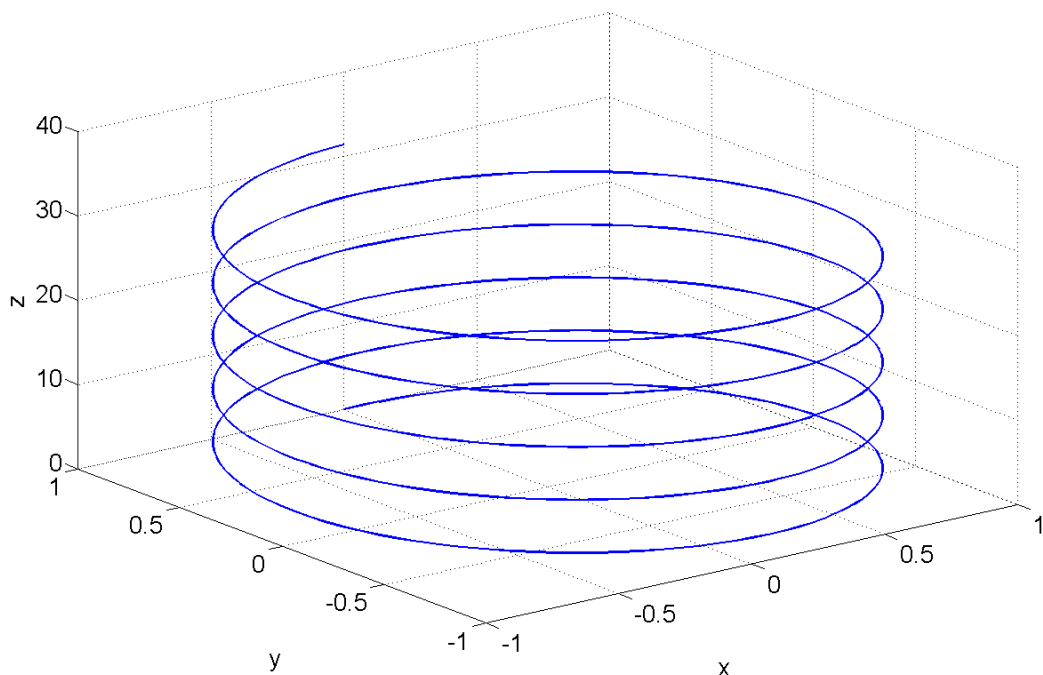
### 3. Adatok ábrázolása 3D-ben

#### A plot3 parancs

A plot3 paranccsal lehetőség nyílik kétváltozós függvények térben történő ábrázolására. Hasonlóan a plot parancshoz, meg kell adni a függvény argumentumainak vektorait, és az értékkészlet vektorát. Ügyeljünk a három megadott vektor azonos dimenziójára!

pl. adjunk meg egy csavarvonalat, mely 5 fordulatot emelkedik! Használjuk az alábbi kódot:

```
z=0:pi/50:10*pi;  
x=sin(z);  
y=cos(z);  
plot3(x,y,z),grid
```



1.4. ábra: csavarvonal ábrázolása a plot3 paranccsal

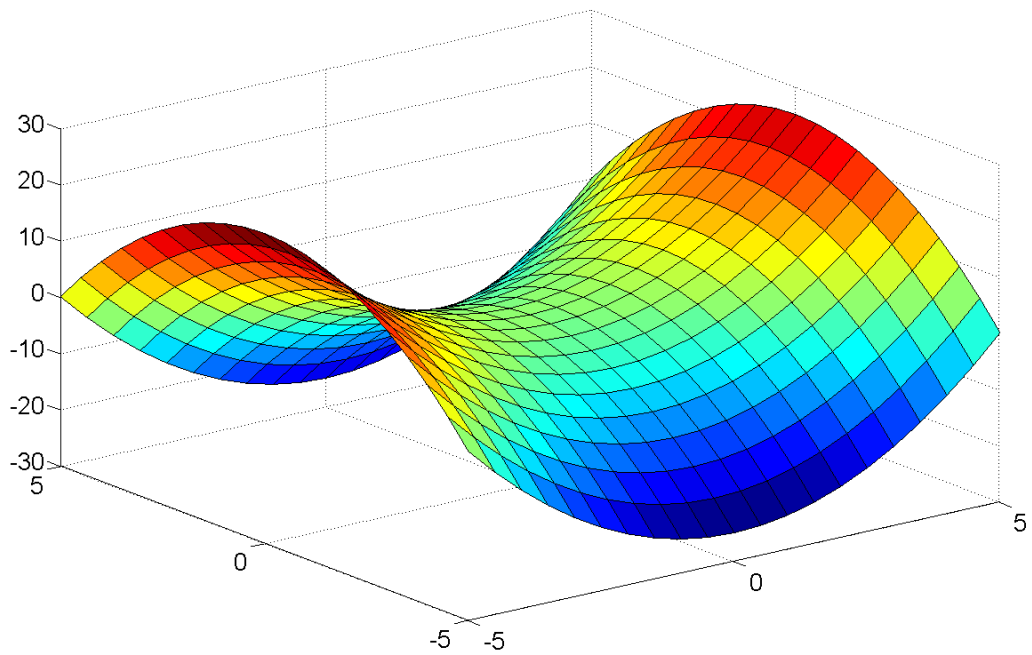
#### A surf és mesh parancsok

Felületek térbeli ábrázolására alkalmas a surf (és a mesh) parancs. A két parancs csak a megjelenítésben különbözik, szintaxisuk megegyező. A függvény argumentumában a függvény értékkészlete, és argumentumának értelmezési tartományai szerepelnek. A függvény értékkészletét mátrixként adjuk meg, melyet az adott értelmezési tartományra felírt függvény értékeivel töltünk fel. Tekintsük az alábbi példát.

Ábrázoljuk az  $f(x,y)=x^2-y^2$  függvényt a  $[-5, 5] \times [-5, 5]$  értelmezési tartományon! Ehhez alkalmazzuk az alábbi kódot:

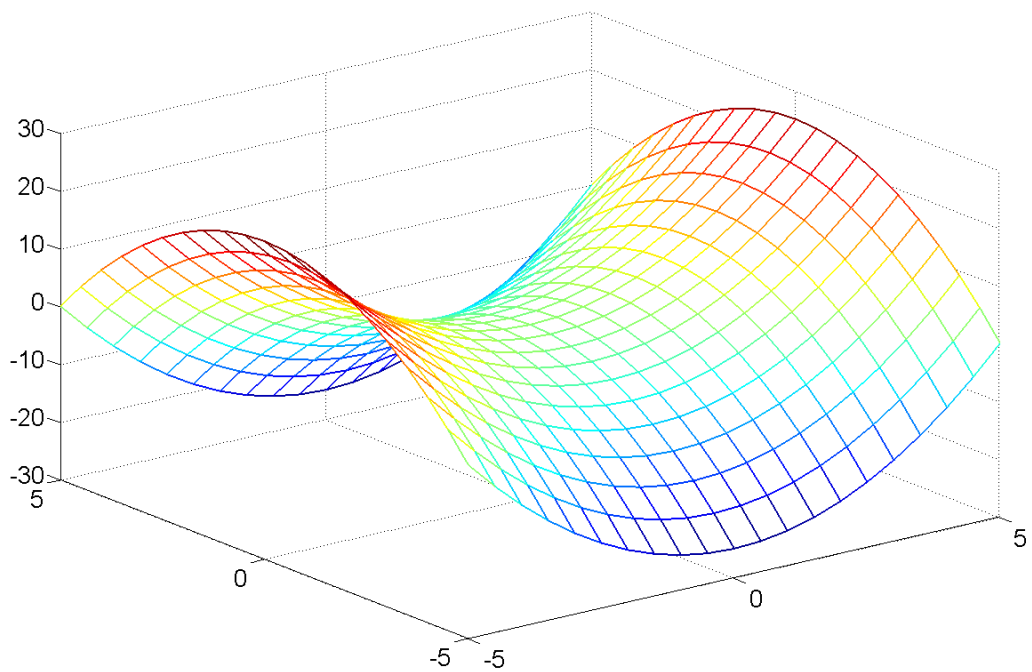
```
[X,Y]=meshgrid(-5:0.5:5);  
Z=X.^2-Y.^2;  
surf(X,Y,Z)
```

A surf paranccsal az 1.5. ábrán látható eredményt kapjuk.



1.5. ábra:  $x^2 - y^2$  függvény ábrázolása a surf paranccsal

A mesh parancs használatával a kapott felület hálóként jelenik meg, ezt ábrázolja a 1.6. ábra.



1.6. ábra:  $x^2 - y^2$  függvény ábrázolása a mesh paranccsal

## 6. lecke: Simulink

A Simulink egy blokkdiagram alapú környezet, mely modell alapú tervezések megvalósítására és dinamikus rendszerek szimulációjára használható. A szimuláció motorjaként a Simulink változtatható lépésközü, közönséges differenciálegyenletek megoldására alkalmas numerikus módszereket alkalmazó megoldókat, ún. ODE-solvereket használ.

A Simulink környezet a Matlabból a simulink parancs segítségével, vagy az eszköztárból a Simulink ikonjára kattintva nyitható meg.

A Simulink modellek struktúrája grafikusán, ablakban szerkeszthető, előre definiált blokkkelem-tár alapján, amit a Simulink Library-ben találunk. A Matlab toolboxok speciális elemeket adnak a Simulink blokkkelem-tárába is. A szimulációk paraméterei és eredményei a Matlabon keresztül is kezelhetők, de lehetséges csak Simulink környezetben definiálni a rendszer paramétereit, illetve csak itt megjeleníteni a szimulációk eredményét.

A továbbiakban egy egyszerű példán keresztül mutatjuk be a Simulink használatát.

**Feladat:** Modellezze le az alábbi dinamikus rendszert, ahol  $x(t)$  a rendszer állapotát,  $u(t)$  a rendszer gerjesztését jelöli. A rendszert az alábbi kezdeti érték problémával adjuk meg:

$$\begin{aligned} dx(t)/dt &= u(t) \\ u(t) &= A \sin(\omega t) \\ u(0) &= 0, x(0) = 0 \end{aligned}$$

**Megjegyzés:** a feladatban szereplő rendszer egy integrátor, mely bemenet hatására az alábbi kimenetfüggvényt adja:

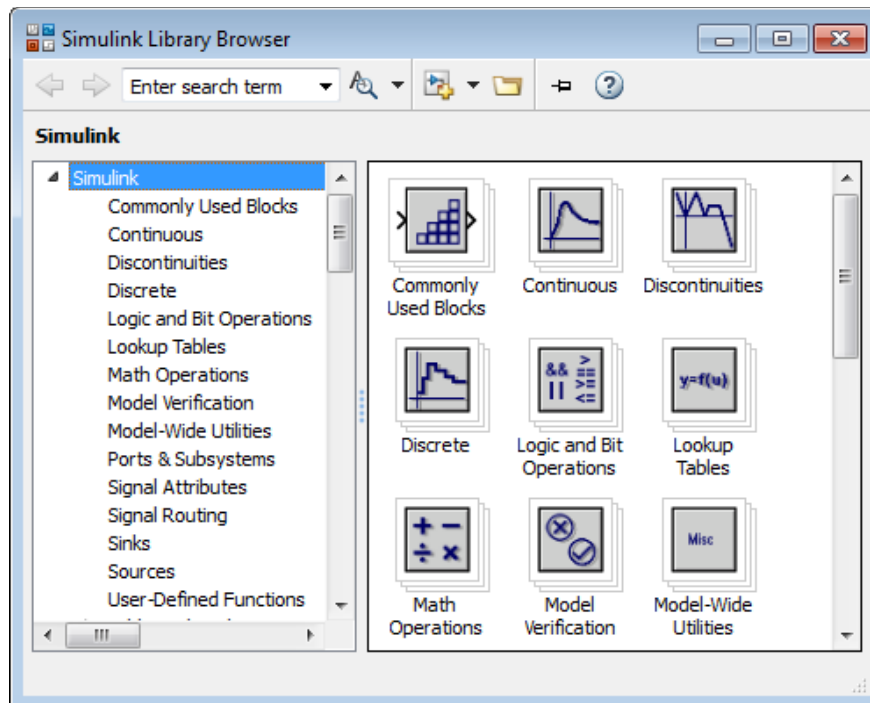
$$x(t) = A \cos(\omega t)$$

A szimulációs modell megvalósításához az alábbi blokkokra van szükségünk (melyeket a Simulink Library-ből választhatunk ki):

- szinuszjel (Sine wave): a rendszer bemenete.
- integrátor (Integrator): a rendszer átviteli függvényét adja.
- multiplexer (Mux): a két jel együttes megjelenítését biztosítja.
- szkóp (Scope): a be- és kimeneti jelek megjelenítéséhez.

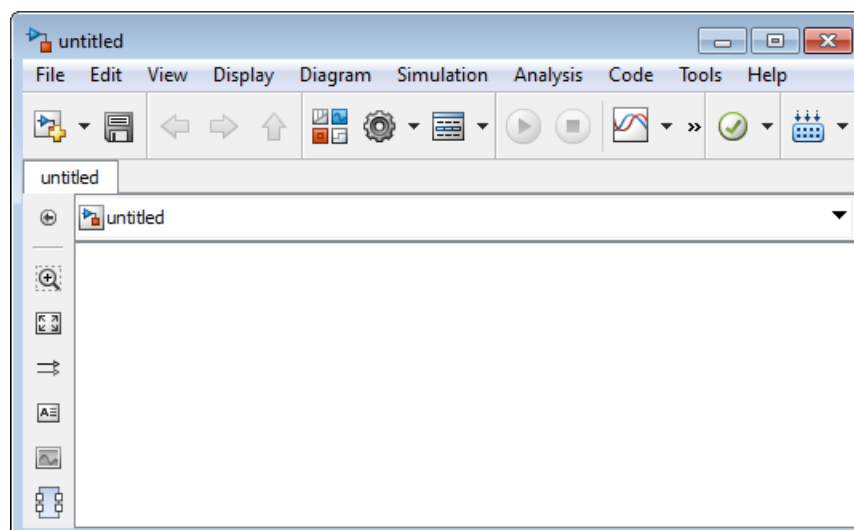
## 1. A modellépítés lépései:

1. Nyissuk meg a Simulink Library Browsert, az alábbi ikonra kattintva: . A megnyitott browsert a 1.7. ábra illusztrálja.



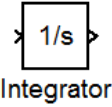



1.7. ábra: Simulink Library Browser

2. Nyissunk meg egy új munkalapot (1.8. ábra).



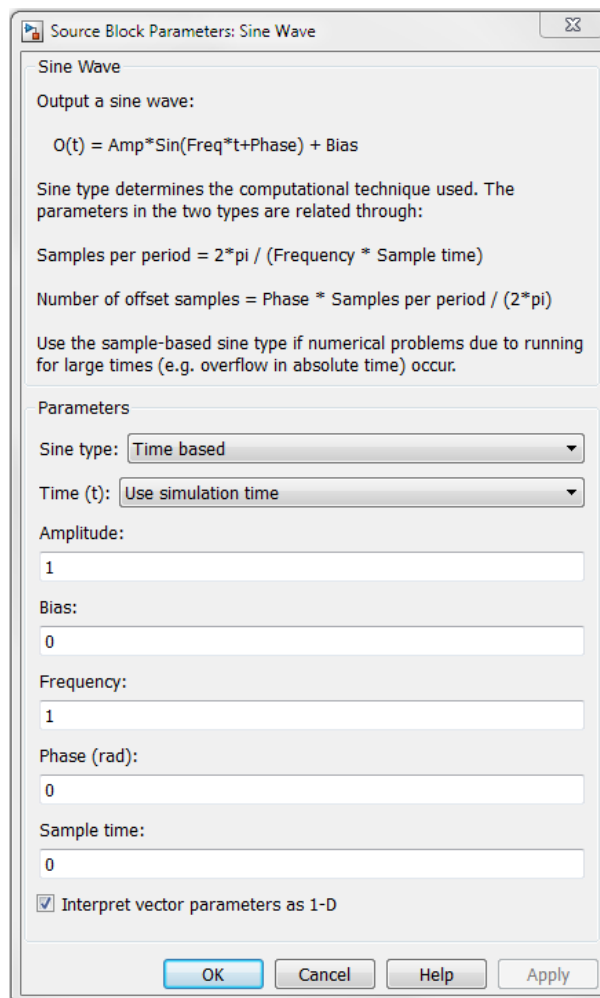
1.8. ábra: üres munkalap

3. Helyezzük el az alábbi blokkokat a munkalapon:

Integrator	
Sine wave	
Mux	
Scope	

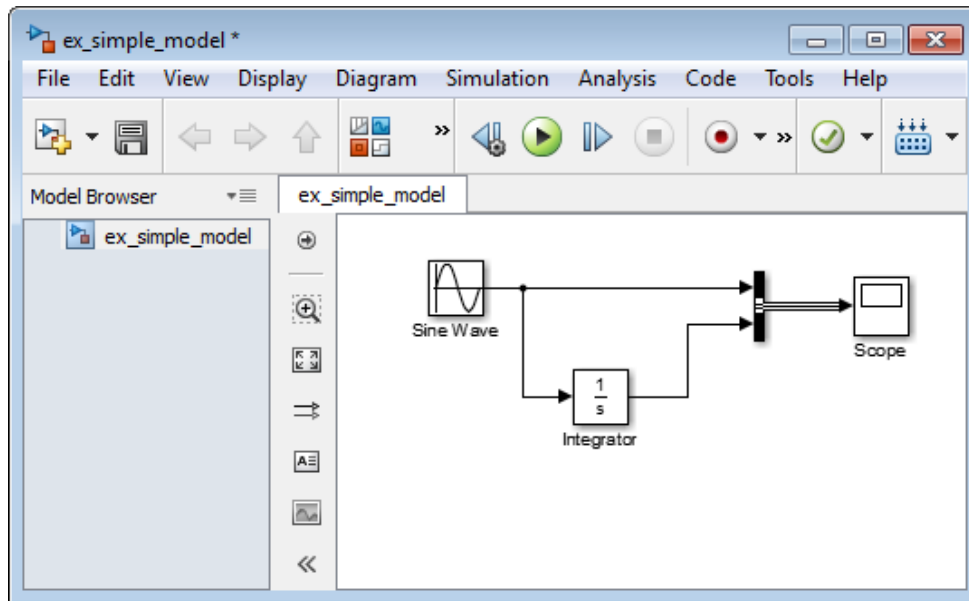
1.1. táblázat: felhasznált Simulink elemek

4. A Sine wave elem számra adjuk meg az alábbi paramétereket:  $\omega=1\text{rad/s}$ ,  $A=1$ . Az 1.9. ábrán látható ablak az elemre kattintva nyílik meg.



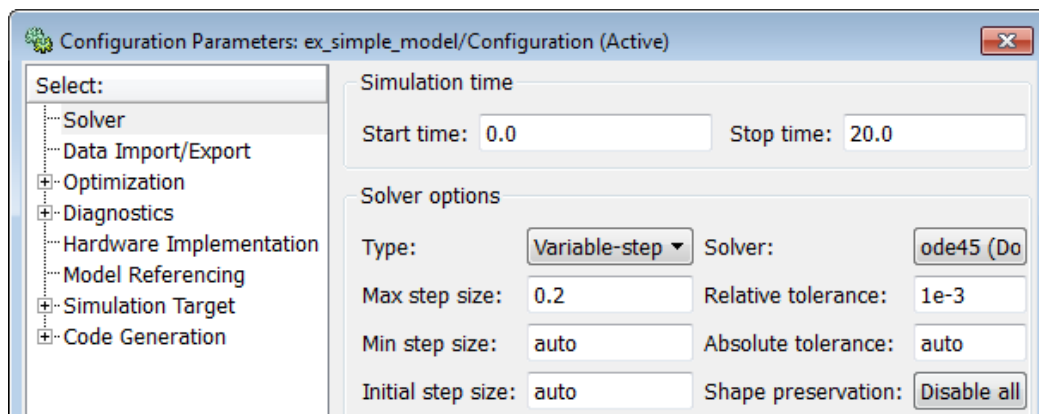
1.9. ábra: szinuszjel paramétereinek megadása

5. Kössük össze a munkalapról helyezett elemeket a 1.10. ábrán látható elrendezés szerint.



1.10. ábra: összekötött elemek



6. Adjuk meg a konfigurációs paramétereket. A Simulation > Model Configuration Parameters menüpontra kattintva a 1.11. ábrán látható ablak nyílik meg.

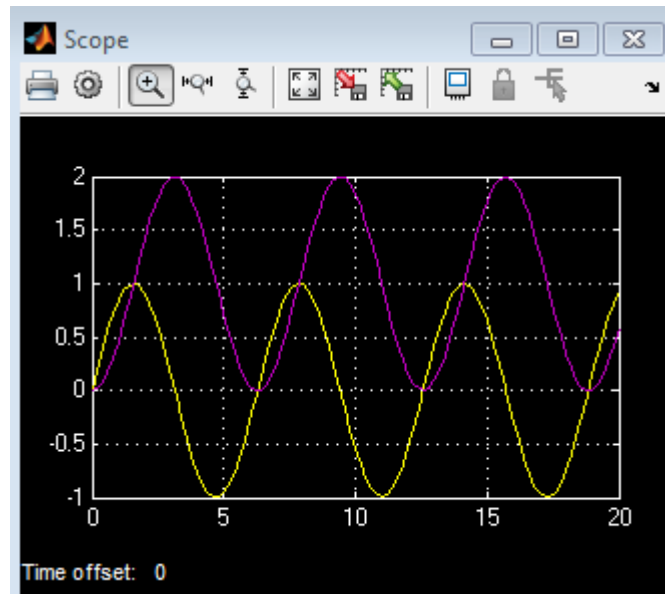


1.11. ábra: konfigurációs paraméterek

Ebben az ablakban az alábbi beállításokat végezzük el:

- start time, stop time: segítségével a szimuláció időtartama állítható be.
- a folytonos idejű szimulációhoz válasszuk ki a Variable-step típust 0.2 s legnagyobb szimulációs lépéssel.

7. Futtassuk le a szimulációt a modellen található  gomb segítségével. A szimuláció szüneteltetéséhez használjuk a  gombot.
8. Elemezzük a szimuláció eredményeit. A Scope blokkra kattintva a rendszer kimenete és bemenete egyazon diagramon megtekinthető (1.12. ábra). Sárgával a rendszer bemenete, lilával a rendszer kimenete jelenik meg.



1.12. ábra: szimuláció eredménye



## 7. Irodalom a 1. modulhoz

- [1] Lantos Béla: Irányítási rendszerek elmélete és tervezése I-II., Akadémiai Kiadó, 2001.
- [2] Schmidt János, Szendrő Balázs, Hullár Béla, Pintér Balázs, Kovács Péter, Koleszár Ádám: Programnyelvek: Matlab. <http://nyelvek.inf.elte.hu/leirasok/Matlab/>

## 2. modul: Forgalommodellezés gyakorlatok Matlabban

### Cél:

A tananyag célja a forgalmi modellek két szintjének: a mikroszkopikus és makroszkopikus tárgyalási módnak bemutatása és megvalósítása egyszerűsített modellek segítségével.

### Követelmények:

Ön akkor sajátította el megfelelően a tananyagot, ha képes

- a forgalmi modellek mikroszkopikus és makroszkopikus tárgyalásmódjának megkülönböztetésére.
- az alapvető modellek feltételezéseinek matematikai megfogalmazására.

### Időszükséglet:

A tananyag elsajátításához körülbelül 4-6 órára lesz szüksége.

### Kulcsfogalmak:

- Simulink
- LWR modell
- járműkövetési modell

## 1. lecke: Lineáris járműkövetési modell

A leckében egy egyszerű mikroszkopikus forgalmi modellt valósítunk meg Matlab Simulink környezetben.

### Elméleti áttekintés

A lineáris járműkövetési modellben [1] az  $i$ . jármű mozgását az alábbi egyenlet írja le:

$$\underbrace{a_i(t + \tau_i)}_{\text{reakció}} = \underbrace{\lambda_i}_{\text{érzékenység}} \cdot \underbrace{(v_{i-1}(t) - v_i(t))}_{\text{inger}} \quad (1)$$

A modell szerint az  $i$ . jármű gyorsulása az  $i - 1$ . és az  $i$ . jármű közötti sebességkülönbség lineáris függvényeként adódik, a lineáris függvény együtthatója a  $\lambda_i$  együttható, mely az  $i$ . jármű vezetőjére jellemző érték. A járművezetőt leíró másik paraméter a vezető reakcióideje  $\tau_i$ .

A járművek sebessége és gyorsulása a jármű elmozdulásának deriváltjaként is felírható. Differenciál alakban a fenti összefüggés a következő:

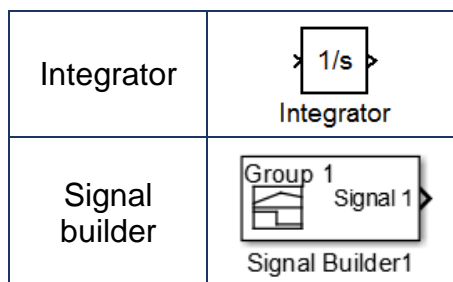
$$\frac{d^2 x_i(t + \tau_i)}{dt^2} = \lambda_i \cdot \left[ \frac{dx_{i-1}(t)}{dt} - \frac{dx_i(t)}{dt} \right] \quad (2)$$






Egy másodrendű lineáris differenciálegyenlethez jutottunk, mely a követő jármű mozgásának dinamikus leírását adja. A jármű  $a_i(t)$ ,  $v_i(t)$ ,  $x_i(t)$  függvényének kinyeréséhez az  $a_i(t)$  gyorsulás függvényen kívül az  $a_1(0)$ ,  $v_1(0)$ , illetve  $x_1(0)$  kezdeti értékek megadása is szükséges. Így voltaképpen egy kezdeti érték probléma megoldását keressük, melyet a (2) egyenlet és az  $a_1(0)$ ,  $v_1(0)$ , illetve  $x_1(0)$  kezdeti értékek adnak meg.

A feladat a (2) egyenlet által leírt és rendszerdinamika és kezdeti érték probléma megvalósítása Simulinken két jármű esetére.

### Modell megvalósítás Simulinkben

A modell felépítéséhez az alábbi Simulink blokkokra van szükség:



Time delay	 Transport Delay
Gain	 Gain
Demux	
Mux	
Scope	 Scope

**2.1. táblázat: felhasznált Simulink elemek**

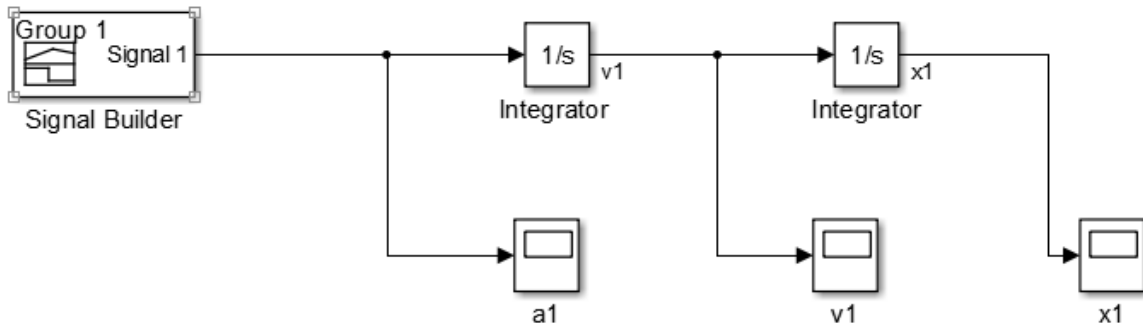
A rendszerben az elől haladó jármű mozgásegyenleteit az alábbi összefüggésekkel definiáljuk:

$$\begin{aligned}
 a_1 &= a_1(t) \\
 v_1 &= v_1(0) + a_1 t \\
 x_1 &= x_1(0) + v_1(0)t + \frac{a_1}{2} t^2
 \end{aligned}
 \tag{3}$$

ahol az  $a_1(t)$  függvény az alábbi:

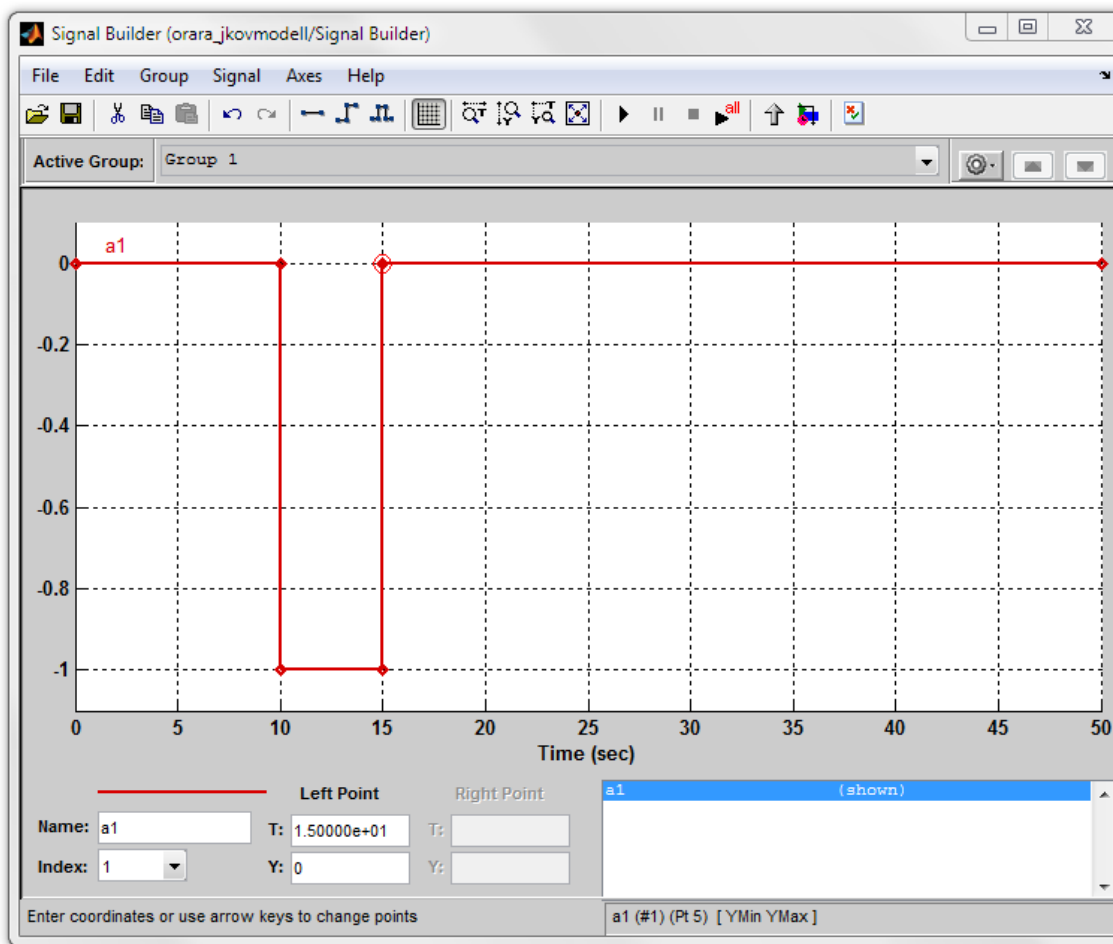
$$a_1 = \left\{ \begin{array}{l} -1 m/s^2, \quad \text{ha } 10 \leq t \leq 15 \\ 0 m/s^2, \quad \text{ha } t < 10 \text{ és } t > 15 \end{array} \right\}
 \tag{4}$$

A felsorolt Simulink blokkok segítségével építjük fel a 2.1. ábrán látható elrendezést, így megkapjuk az elől haladó jármű mozgását szimuláló modellt!



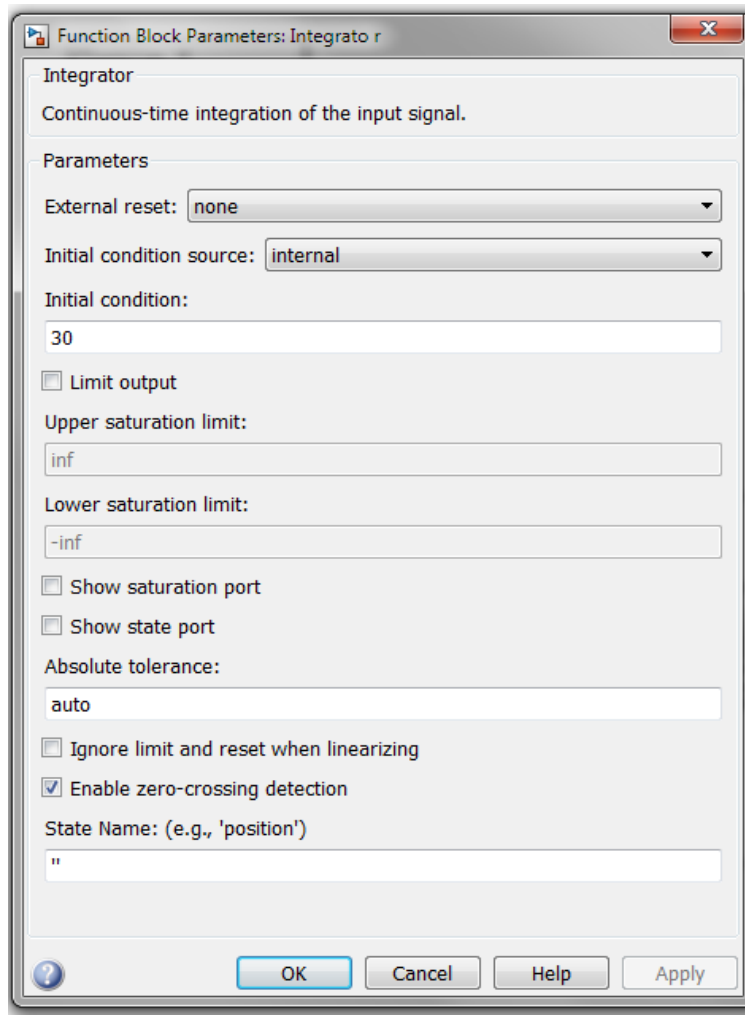
2.1. ábra: elől haladó jármű mozgásának szimulációja

A rendszer gerjesztéseként szolgáló  $a_1(t)$  gyorsulást a „Signal Builder” blokkban definiálhatjuk, ezt mutatja be a 2.2. ábra.



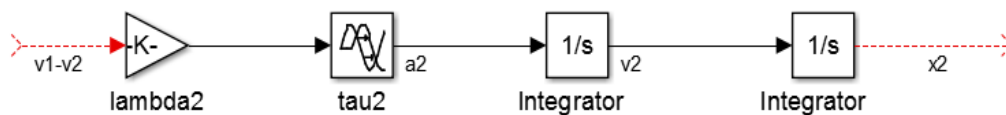
2.2. ábra: az  $a_1(t)$  függvény megadása

A  $v_1(0)$ ,  $x_1(0)$  kezdeti értékeket az integrátor blokkok „Initial condition” mezőiben definiálhatjuk (2.3. ábra). Adjuk meg a következő kezdeti értékeket:  $v_1(0) = 30\text{m/s}$ ,  $x_1(0) = 100\text{m}$ .



2.3. ábra:  $v_1(0)$  kezdeti érték megadása

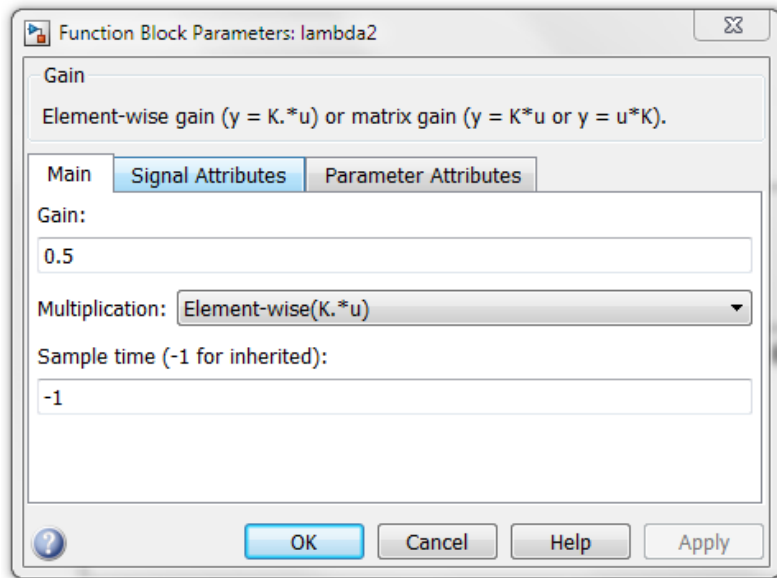
A 2.4. ábrán szereplő blokkokat a követő jármű változóit leíró blokkokkal egészítjük ki: ez a  $v_2(t)$  és  $x_2(t)$  függvényeket adó integrátorokat, továbbá (1) egyenlet jobb oldalán szereplő  $\lambda_i$ , ( $i=2$ ) érzékenységi együtthatót és bal oldalán található  $\tau_i$ , ( $i=2$ ) reakcióidőt reprezentáló blokkokat jelenti.



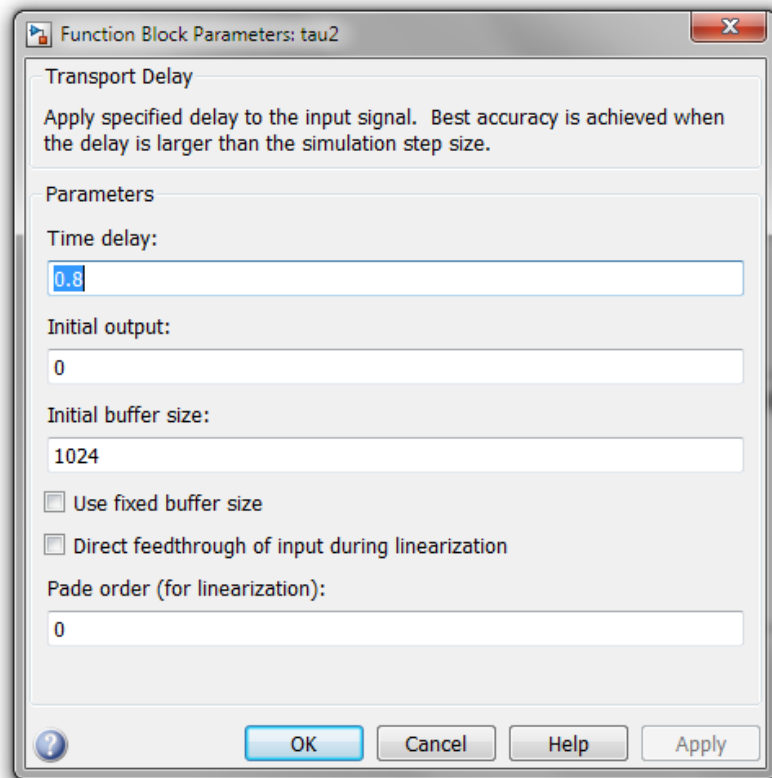
2.4. ábra: követő jármű modellje

Hasonlóan az elől haladó járműhöz, a követő jármű esetén is adjuk meg a következő kezdeti értékeket (a két integrátor blokkban):  $v_2(0) = 40\text{m/s}$ ,  $x_2(0) = 70\text{m}$ . Látható, hogy  $v_2(0) > v_1(0)$ , így egy utolérési szituációt modellezünk.

A kezdeti értékek megadását követően adjuk meg a  $\lambda_2$  és  $\tau_2$  paraméterek értékét a következőképpen:  $\lambda_2 = 0.5$  (2.5. ábra);  $\tau_2 = 0.8\text{s}$  (2.6. ábra).

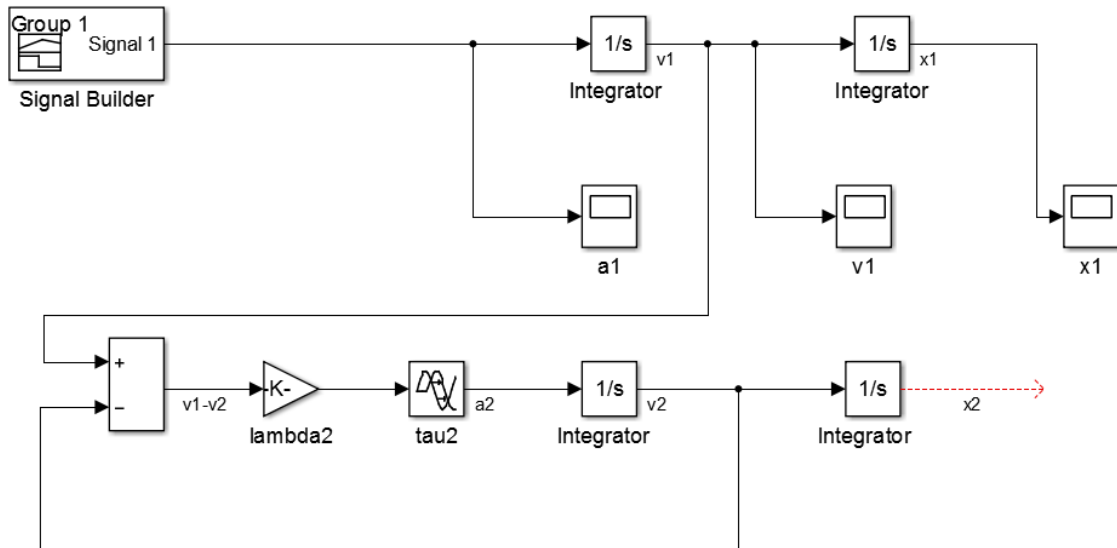


2.5. ábra: a követő jármű érzékenységi paraméterének ( $\lambda_2$ ) beállítása



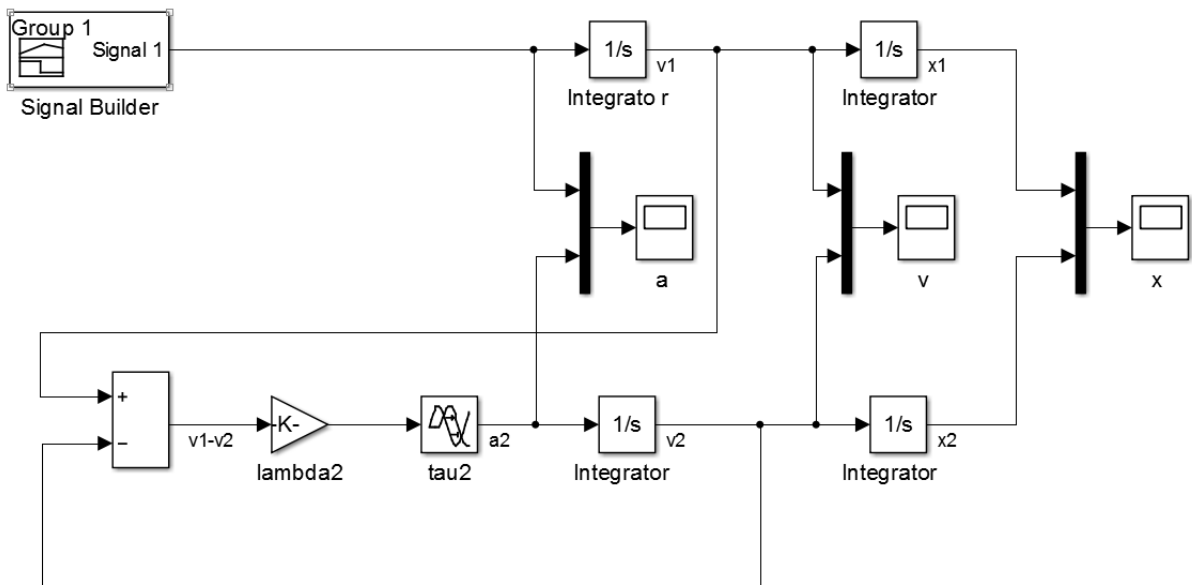
2.6. ábra: a követő járműre jellemző reakcióidő ( $\tau_2$ ) megadása

A  $\lambda_2$  blokk bemenete a  $v_1(t) - v_2(t)$  sebességkülönbség, mely a követő jármű gerjesztését adja. A 2.1. és 2.4. ábrák összevonásával a sebességkülönbség, mint gerjesztés megadható (2.7. ábra).



2.7. ábra: elöl haladó és követő jármű modelljének összekapcsolása

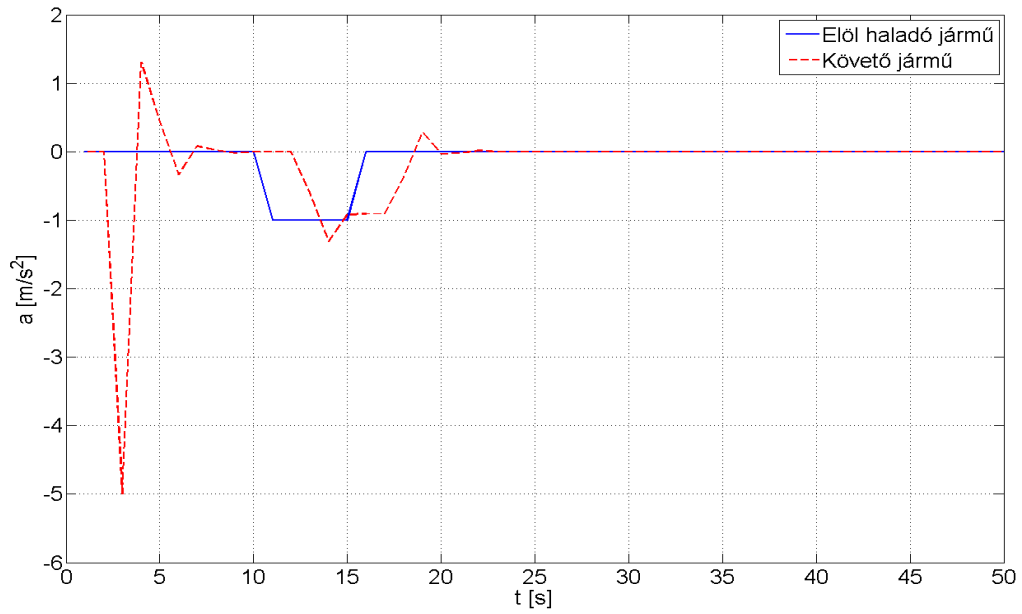
A két jármű mozgását leíró változók együttes megjelenítéséhez a „Scope” blokkok elé „Mux” blokkokat helyezünk (2.8. ábra).



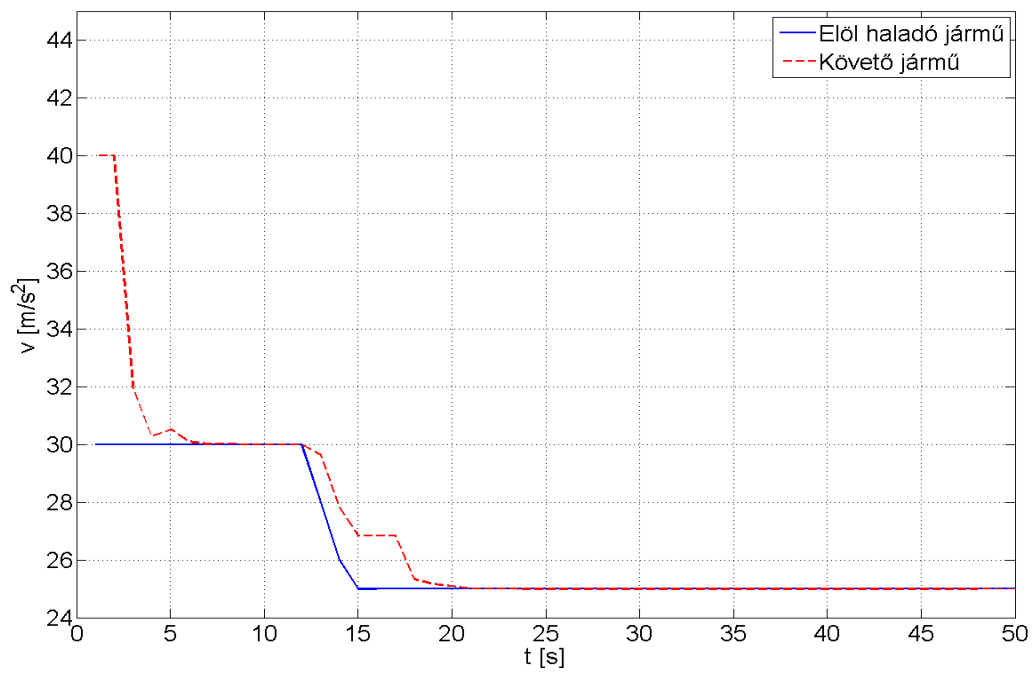
2.8. ábra: járműkövetési modell megvalósítása két járműre

Így a szimuláció futtatása után megjeleníthető a két jármű  $a(t)$ ,  $v(t)$ ,  $x(t)$  függvénye (lásd: 2.9-2.11. ábrák). A diagramokat megtekintve látszik, hogy a követő jármű 5s után éri utol az elöl haladó járművet, és ezt követően azonos sebességgel haladnak. Az elöl haladó jármű 10s után fékezni kezd, melyet a követő jármű szintén megtesz, a reakcióidőnek megfelelő késéssel.

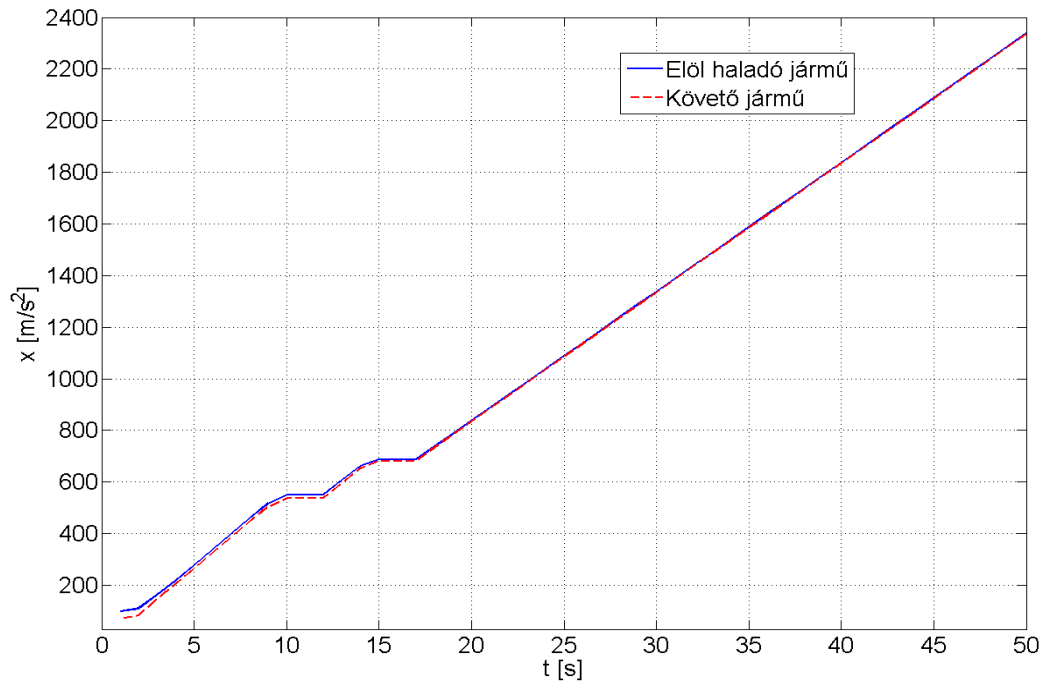




2.9. ábra: gyorsulás-idő diagramok



2.10. ábra: sebesség-idő diagramok



**2.11. ábra: elmozdulás-idő diagramok**

## 2. lecke: A járműkövetési modell stabilitásvizsgálata

Az 1. feladatban bemutatott szimuláció során a követő jármű biztonságos reakciót mutat: a gyorsulásában jelentkező változás kis mértékű és késése nem vezetett sem ütközéshez, sem oszcillációhoz a követő jármű sebesség-idő függvényében (2.9. ábra). A tanúsított reakció jellegét alapvetően a járművezetőre jellemző paraméterek:  $\lambda$  és  $\tau$  határozzák meg. Sejthető, hogy nagy reakcióidő és érzékenység értékek esetén jelentős mértékű és késésű gyorsítás jelentkezik. Ez adott esetben a két jármű ütközését is maga után vonhatja, vagy a követő jármű reakciójában oszcillációkhoz vezethet. Szintén sejthető, hogy kis reakcióidejű, kismértékű beavatkozások stabil viselkedést mutatnak.

Ebben a feladatban a járműkövetési modell stabilitását vizsgáljuk. A rendszert stabilnak nevezük, ha egy véges bemenőjelre adott válasza  $t \rightarrow \infty$  esetén 0-hoz tart\*.

---

\* A BIBO stabilitás ekvivalens definíciója szerint a rendszer stabil, ha egy energiakorlátos  $u(t)$  bemenetre  $(\int_0^\infty |u(t)|^2 dt < \infty)$  adott rendszerkimeneti jelre  $(y(t))$  teljesül, hogy  $\lim_{t \rightarrow \infty} y(t) = 0$ .

## Elméleti áttekintés

Az [1] munka a járműkövetési modell stabilitását is vizsgálja, és a  $\lambda$  és  $\tau$  paraméterek függvényében kritériumokat fogalmaz meg a követés stabilitására vonatkozóan. Az alábbiakban ezt tekintjük át, és a feladatban az előzőekben felépített Simulink modellen végzünk stabilitásanalízist. A lokális stabilitás vizsgálatával két jármű által meghatározott járműkövetési rendszeren értelmezhető a stabilitás. Fogalmazzuk át a (2) egyenletet a két jármű közötti sebességkülönbség dinamikus leírására. A sebességkülönbséget  $y(t)$ -vel jelölve:

$$y(t) = \frac{dx_{i+1}(t)}{dt} - \frac{dx_i(t)}{dt} \quad (5)$$

Az alábbi differenciálegyenlethez jutunk:

$$\frac{dy(t + \tau)}{dt} = -\lambda y(t) \quad (6)$$

Bontsuk fel  $y(t)$ -t az alábbi két tag összegére, ahol

$$\begin{aligned} y_0(t) & \text{ a megoldás } t > \tau \text{ esetén} \\ z(t) & \text{ a megoldás } t < \tau \text{ esetén} \end{aligned} \quad (7)$$

Ekkor (6) az alábbi formában írható fel:

$$\frac{dy(t + \tau)}{dt} = -\lambda(y_0(t) + z(t)) \quad (8)$$

A (8) differenciálegyenlet Laplace-transzformálásával<sup>†</sup> és a (7) egyenlet felhasználásával az alábbi egyenlethez jutunk:

$$Z(s) = -\frac{\lambda Y_0(s)}{se^{s\tau} + \lambda} + \frac{Y_0(\tau)}{se^{s\tau} + \lambda} \quad (9)$$

A rendszer stabilitásának vizsgálatához keressük a (9) egyenlet  $Y_0(s) \rightarrow Z(s)$  közötti átvitel pólusait az alábbi egyenlet gyökeiként:

$$se^{s\tau} + \lambda = 0 \quad (10)$$

---

<sup>†</sup> Egy adott  $f(t)$ ,  $t \in [0, \infty)$  függvény Laplace transzformáltját az alábbi improprius integrállal definiáljuk:

$$F(s) = \int_0^{\infty} e^{-st} f(t) dt$$

A (10) egyenlet bal oldala egy Lambert-féle logaritmus szorzatfüggvény. Az (10) egyenlet megszámlálhatóan végtelen megoldással rendelkezik, tehát szabályszerűség írható fel a végtelen sok megoldás tagjai között. Ezek vizsgálatát az alábbiakban a  $\lambda$  és  $\tau$  paraméterek függvényében végezzük. Emlékezzünk, hogy az egyenlet gyökei a rendszer pólusai! A legnagyobb valós részű  $s_0$  gyökből következtetünk a rendszer stabilitására. A  $\lambda$  és  $\tau$  paraméterek függvényében az alábbi stabilitási tulajdonságokat kapjuk:

- ha  $\lambda \cdot \tau \leq e^{-1}$ , az  $s_0$  gyök valós és negatív, tehát stabil viselkedést kapunk: a rendszer reakciója monoton csökkenő  $t \rightarrow \infty$  esetén.
- ha  $e^{-1} < \lambda \cdot \tau < \pi/2$ , az  $s_0$  gyök komplex, negatív valós résszel. A rendszer így elnyomja a gerjesztést, tehát csökkenő reakciót kapunk  $t \rightarrow \infty$  esetén. A reakció azonban nem monoton csökkenő, hanem szinuszos.
- ha  $\lambda \cdot \tau = \pi/2$ , az  $s_0$  gyök komplex, 0 valós résszel. A rendszer ekkor a stabilitás határán van: a gerjesztést nem nyomja el, de nem is erősíti.  $t \rightarrow \infty$  esetén a reakció konstans amplitúdójú szinuszos jel.
- ha  $\lambda \cdot \tau > \pi/2$ , az  $s_0$  gyök pozitív valós részű. A rendszer ekkor instabil: a gerjesztést erősíti, így  $t \rightarrow \infty$  esetén a reakció növekvő amplitúdójú szinuszos jel.

### Stabilitás vizsgálat Simulinkben

Az alábbiakban az 1. feladatban felépített szimulációs rendszeren vizsgáljuk a lineáris járműkövetési modell stabilitását. A jobb összehasonlíthatóság kedvéért mindkét járműre adjunk meg 30 m/s kezdeti sebességet! A stabilitási esetek vizsgálatához alkalmazzuk az 2.2. táblázatban megadott paramétereket.

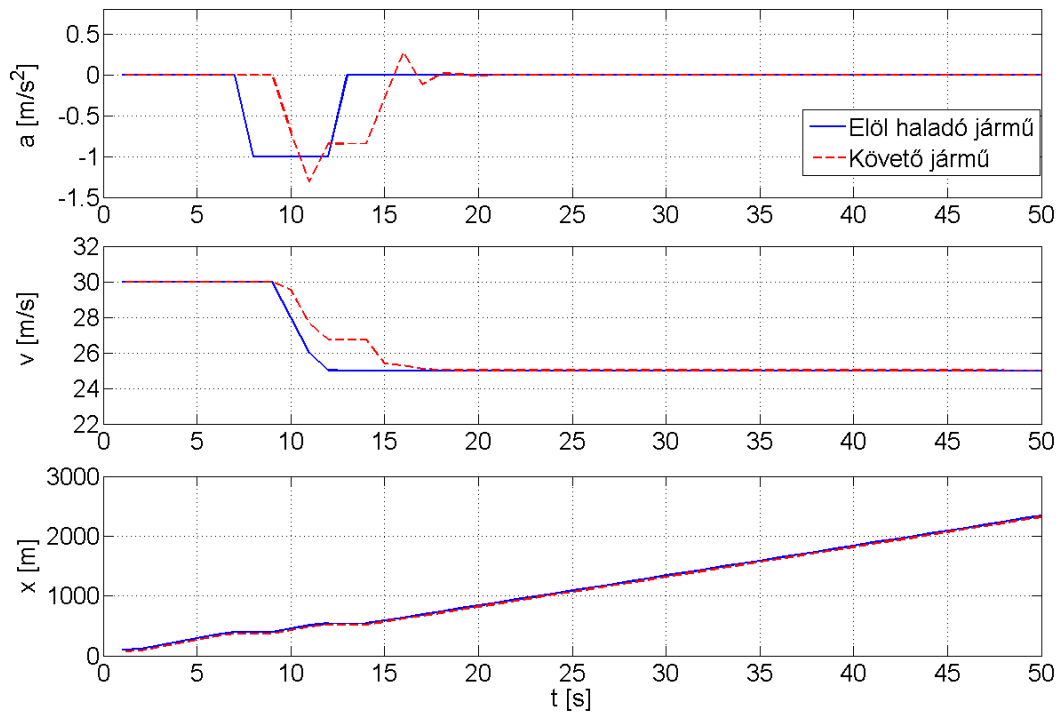
	$\lambda$	$\tau$	$\lambda \cdot \tau$
1.eset	0,5	0,6	0,3
2.eset	1	0,8	0,8
3.eset	1,571	1	1,571
4.eset	1,8	1,5	2,7

2.2. táblázat: szimulációs paraméterek

Látható, hogy a megadott paraméterekkel a stabilitási esetek mindegyike vizsgálható. Az alábbiakban alkalmazzuk a megadott paramétereket, és vizsgáljuk a rendszer stabilitását!

#### 1. eset – stabil viselkedés

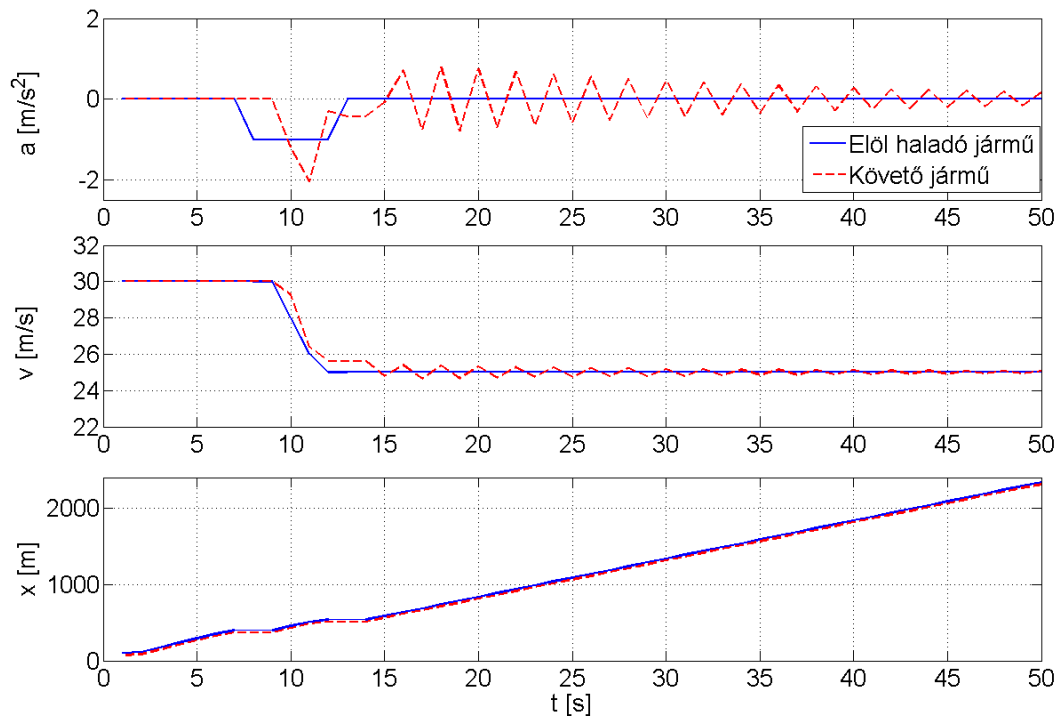
A szimuláció eredménye a 2.12. ábrán látható. A követő jármű az öt érő ingert nem erősíti fel, kis amplitúdójú reakciójával gyorsan alkalmazkodik az előtte haladó jármű sebességéhez.



2.12. ábra:  $a(t), v(t), x(t)$  diagramok - 1. eset

## 2. eset – stabil viselkedés, csökkenő oszcilláció a rendszer kimenetén

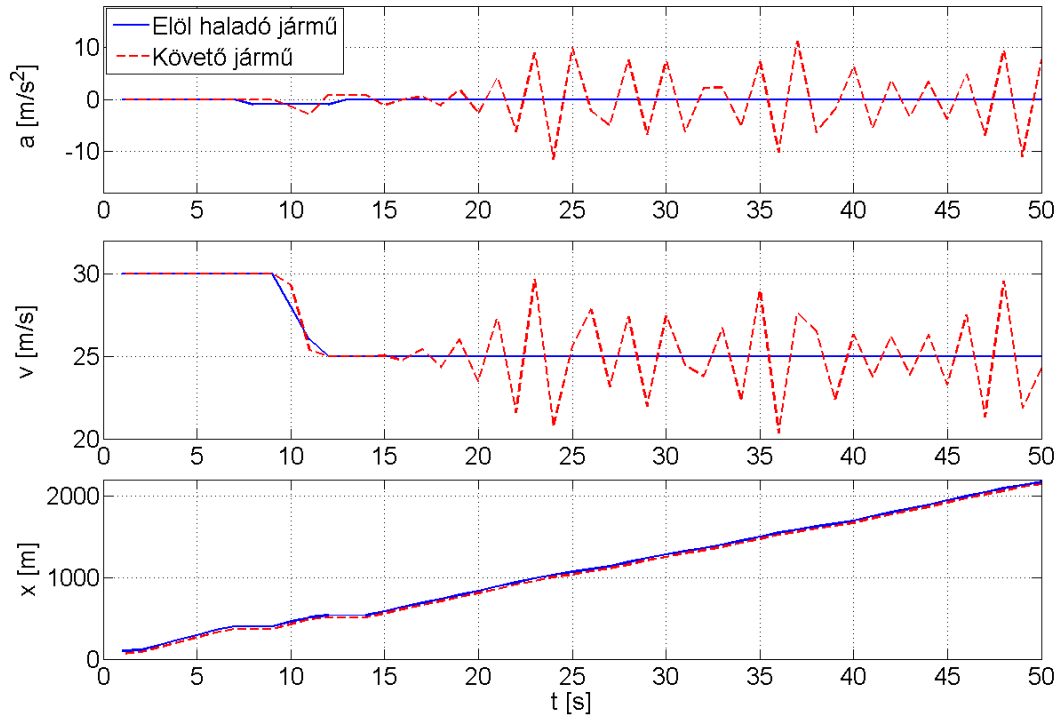
Ebben az esetben ugyan rendszerelméleti definíció szerint stabil viselkedést mutat a követő jármű (tehát  $t \rightarrow \infty$  esetén a sebességkülönbség is tart a 0-hoz), de nemkívánatos oszcilláció jelenik meg a reakciójában a megnövelt érzékenységre és reakció eredményeként (2.13. ábra).



2.13. ábra:  $a(t), v(t), x(t)$  diagramok – 2. eset

### 3. eset – oszcilláció a rendszer kimenetén

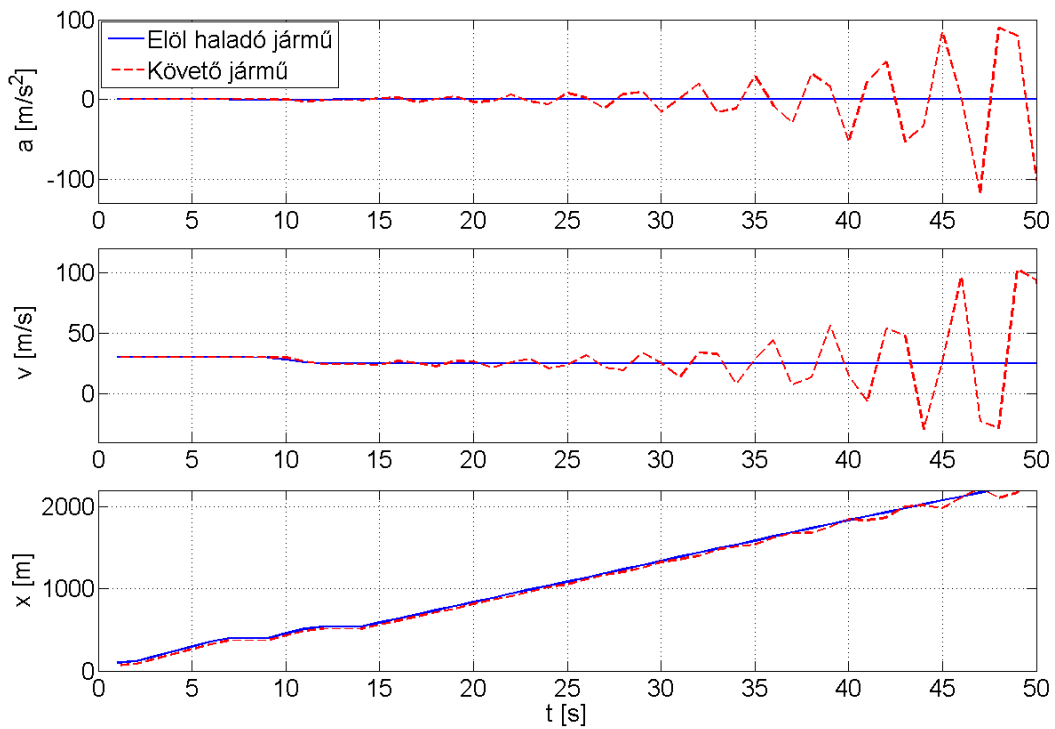
A stabilitás határán a követő jármű reakciója 0 körüli oszcilláció, melynek amplitúdója korlátos. Az oszcilláció eredményeként a jármű nem vesz fel konstans sebességet.



2.14. ábra:  $a(t)$ ,  $v(t)$ ,  $x(t)$  diagramok - 3. eset

#### 4. eset – instabil viselkedés, növekvő oszcilláció a rendszer kimenetén

Instabil viselkedést láthatunk a 2.15. ábrán. A jármű reakciójában jelen lévő oszcilláció amplitúdója nem korlátos,  $t \rightarrow \infty$  esetén  $\infty$ -hez tart. A növekvő negatív gyorsulásértékek hatására a követő jármű negatív sebességet is felvesz  $t = 40$ s után. A  $t = 47$ s időpontban a követő jármű utoléri és meg is előzi az elől haladó járművet, így a mozgása az elől haladó jármű gyorsulása, sebessége és pozíciója körüli monoton növekvő amplitúdójú oszcillációvá válik.



2.15. ábra:  $a(t)$ ,  $v(t)$ ,  $x(t)$  diagramok - 4. eset

A négy esettanulmány alapján levonható az a következtetés, hogy a stabil viselkedésű járművezető kis beavatkozásokat tesz kis reakcióidővel.



### 3. lecke: Makroszkopikus forgalom-modellezés

A leckében egy makroszkopikus forgalmi modellt valósítunk meg a diszkrét LWR forgalmi modellel Simulinkben.

#### Elméleti áttekintés

Az 1955-ben megalkotott Lighthill-Whitham-Richards féle forgalmi modell (a továbbiakban LWR modell) az autópálya forgalomban jelen lévő lökeshullámok dinamikus leírására hozták létre [2], [3]. A modell részletes leírása a [4] könyvben található, az alábbiakban egy rövid összefoglalását adjuk a modellnek.

A modell makroszkopikus, azaz a forgalomban résztvevő járműegyedek viselkedését figyelmen kívül hagyva a forgalom egészének dinamikus leírását adja. Ehhez az alábbi aggregált forgalmi változókat használja:

Az időben folytonos forgalomsűrűség ( $\rho(t)$ ): adott  $t$  pillanatban egy vizsgált hosszúságú  $L$  útszakaszra eső  $N(t)$  járműmennyiség.

$$\rho(t) = \frac{N(t)}{L} \quad (11)$$

A térben folytonos forgalomnagyság ( $q(x)$ ): adott  $x$  térbeli ponton egy vizsgált  $T$  időszak alatt áthaladó  $M(x)$  járműmennyiség.

$$q(x) = \frac{M(x)}{T} \quad (12)$$

A forgalmi átlagsebesség kétféleképp adható meg:

- időben folytonos átlagsebesség ( $v_t(x)$ ): a vizsgált  $L$  útszakaszon az adott  $t$  időpontban elhelyezkedő járművek  $v_i(t)$  pillanatnyi sebességeinek átlaga:

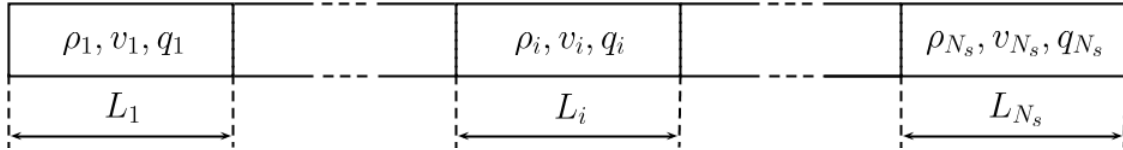
$$v_t(x) = \frac{\sum_{i=1}^{M(x)} v_i(x)}{M(x)} \quad (13)$$

- térben folytonos átlagsebesség ( $v_s(t)$ ): az adott  $x$  ponton áthaladó járművek  $v_i(x)$  sebességeinek átlaga:

$$v_s(t) = \frac{\sum_{i=1}^{N(t)} v_i(t)}{N(t)} \quad (14)$$

A fenti változók vizsgálati tartományát (a térben folytonos jellemzők esetén a  $T$  vizsgált időszakot, az időben folytonos jellemzőknél az  $L$  vizsgált útszakaszt) minden határon túl

finomítva térben és időben folytonos  $\rho(x, t), q(x, t), v_s(x, t), v_t(x, t)$  változókat kapunk. Folytonos  $(x, t)$  tartományon  $v_s(x, t)$  és  $v_t(x, t)$  megegyezik, a továbbiakban  $v(x, t)$ -vel jelöljük.



**2.16. ábra: térben diszkrétizált autópálya hálózat**

A folytonos makroszkopikus forgalmi változók értelmezése kiterjeszhető véges tér-idő ablakokra is. Egy úthálózat véges számú útszakaszra osztható (lásd 2.16. ábra), ekkor a hálózat véges számú térben diszkrét  $(\rho_i, v_i, q_i)$  makroszkopikus változó hármassal írható le. A numerikus stabilitást szem előtt tartva [3] az  $L_i$  szakaszhoz ismeretében megfelelő  $T_S$  mintavételi idő választható, így térben és időben diszkrét változókhoz jutunk. Így adott  $i$  szakaszon a  $k$ . diszkrét lépésben az alábbi diszkrét makroszkopikus változókat használjuk:

$$\rho_i(k) = \rho_i([kT_S, (k+1)T_S]) \quad (15)$$

$$v_i(k) = v_i([kT_S, (k+1)T_S]) \quad (16)$$

$$q_i(k) = q_i([kT_S, (k+1)T_S]) \quad (17)$$

A diszkrét és folytonos forgalmi változók értelmezése után tekintsük át az LWR modell feltételezéseit és matematikai alakját.

Az LWR modell az alábbi feltételezésekre épül:

1. A forgalomra teljesül a megmaradási elv: adott időszakban egy vizsgált útszakaszon a járműszám megváltozása egyenlő a behaladó és kihaladó forgalomnagyságok különbségével.
2. A forgalom sebessége a forgalom sűrűségének monoton csökkenő függvénye.
3. A forgalmi állapotváltozók között fennáll a fundamentális összefüggés: adott  $(x, t)$  pontban a forgalom nagysága kifejezhető a forgalomsűrűség és a forgalmi átlagsebesség szorzataként.

A továbbiakban az LWR modellt térben és időben diszkrét módon formalizáljuk, térben visszafelé képzett (backward difference) numerikus approximációval. A modell feltételezéseit az alábbi egyenletekkel írhatjuk le, a 2.16. ábrán látható hálózat  $i$ . szakaszára:

A járművek megmaradása az alábbi diszkrét differencia-egyenlettel adható meg:

$$\rho_i(k+1) = \rho_i(k) + \frac{T_S}{L_i} \cdot [q_{i-1}(k) - q_i(k)] \quad (18)$$

A forgalom sebessége a forgalom sűrűségének monoton csökkenő függvénye:

$$v_i(k) = V(\rho_i(k)) \quad (19)$$

ahol  $V(\rho_i(k))$  jelöli az egyensúlyi sebesség-sűrűség összefüggést, melyre az alábbi függvényt alkalmazzuk az [5] munka nyomán:

$$V(\rho_i(k)) = v_{free} \exp\left(-\frac{1}{a} \left(\frac{\rho_i(k)}{\rho_{krit}}\right)^a\right) \quad (20)$$

ahol  $v_{free}$ ,  $a$ ,  $\rho_{krit}$  modellparaméterek.

A forgalmi állapotváltozók közötti összefüggést a fundamentális egyenlettel írjuk le:

$$q_i(k) = \rho_i(k) \cdot v_i(k) \quad (21)$$

Az egyensúlyi sebességfüggvényt (20) a fundamentális egyenletbe (21) helyettesítve megkapjuk az úthálózat egyensúlyi kapacitásfüggvényét:

$$q_i(k) = Q(\rho_i(k)) = V(\rho_i(k))\rho_i(k) \quad (22)$$

Az LWR modellt alkotó (18)-(22) egyenletek egy térben-időben diszkrét dinamikus leírását adják egy autópálya hálózatnak. Az alábbiakban ezek megvalósítását mutatjuk be Simulink környezetben.

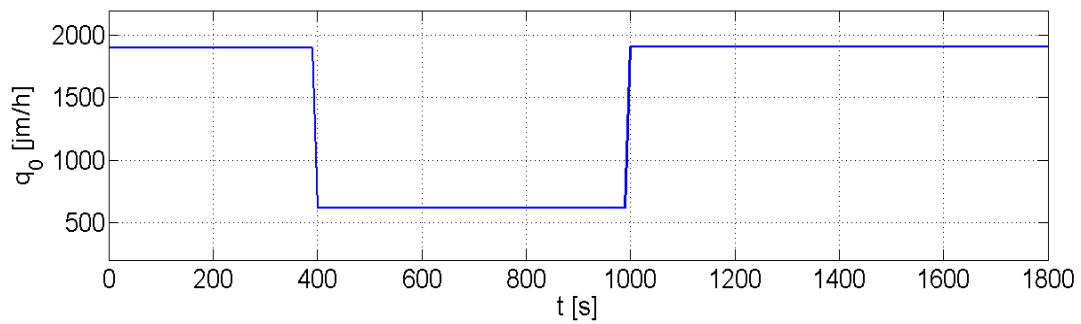
### Modell építése Simulinkben

Alkalmazzuk a (18)-(22) egyenleteket az alábbi paraméterekkel:

$L_i$	1 km
$T_S$	10 s
$\rho_{krit}$	25 jn/km
$v_{free}$	130 km/h
$a$	2,5



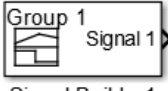


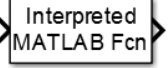



**2.3. táblázat: modellparaméterek**

A rendszerre a 2.17. ábra szerinti forgalomnagyság gerjesztést adjuk (1900 J/h-ról indítva).



2.17. ábra: a rendszer gerjesztése

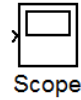
A modell építéséhez az alábbi Simulink elemeket alkalmazzuk:

- |                             |  |
|-----------------------------|--|
| Clock                       |     |
|                             | Clock  |
| Subsystem                   |    |
|                             | Subsystem  |
| Signal builder              |  |
|                             | Signal Builder1  |
| Input                       |   |
|                             | In   |
| Output                      |   |
|                             | Out  |
| Interpreted Matlab function |  |
|                             | fcn  |
| Matlab function             |   |
|                             | MATLAB Function  |
| Time delay                  |   |
|                             | Transport Delay  |
| Demux                       |   |

Mux



Scope



#### 2.4. táblázat: felhasznált Simulink blokkok

A szakasz dinamikus modelljét a (18) differenciaegyenlet adja. Ezt az egyszerűség kedvéért beépített Matlab függvénnyel valósítjuk meg, ez tartalmazza a 2.3. táblázatban megadott modellparamétereket is. A Simulink modellre elhelyezett „Interpreted Matlab Function” blokkba az alábbi kódot helyezzük el:

```
function y=lwr(u)
rho_i_k =u(1);           % rho_i (k)
q_h_k=u(2);             % q_{i-1}(k)

% Parameterek megadása
Ts=0.0028;
L=1;
v_free=130;
rho_cr=25;
a=2.5;

% v_i(k):
v_i_k = v_free.*exp(-(1/a).*((rho_i_k)/rho_cr).^a);

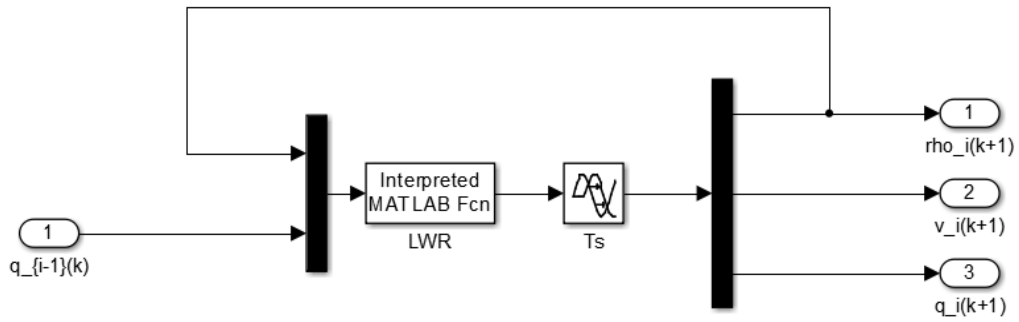
% rho_i(k+1):
rho_i_k1 = rho_i_k + (Ts/L)*(q_h_k - rho_i_k*v_i_k);

% v_i(k+1):
v_i_k1 = v_free.*exp(-(1/a).*((rho_i_k1)/rho_cr).^a);

% q_i(k+1):
q_i_k1=v_i_k1*rho_i_k1;

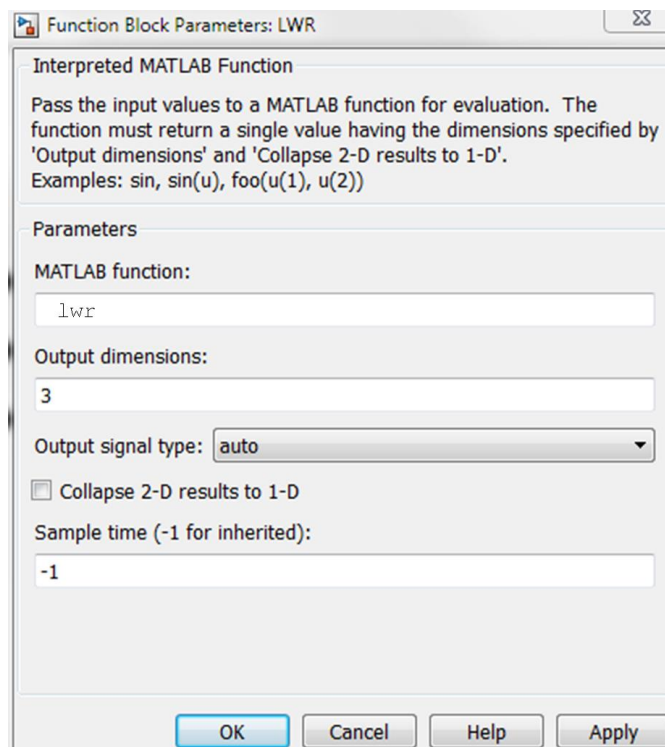
y=[rho_i_k1;v_i_k1; q_i_k1];
```

Építsük fel egy szakasz modelljét az 2.18. ábrán mutatott elrendezés szerint. A „Transport delay” blokkban adjunk meg a gerjesztésnek megfelelő kezdeti érték feltételeket és a mintavételi időt.

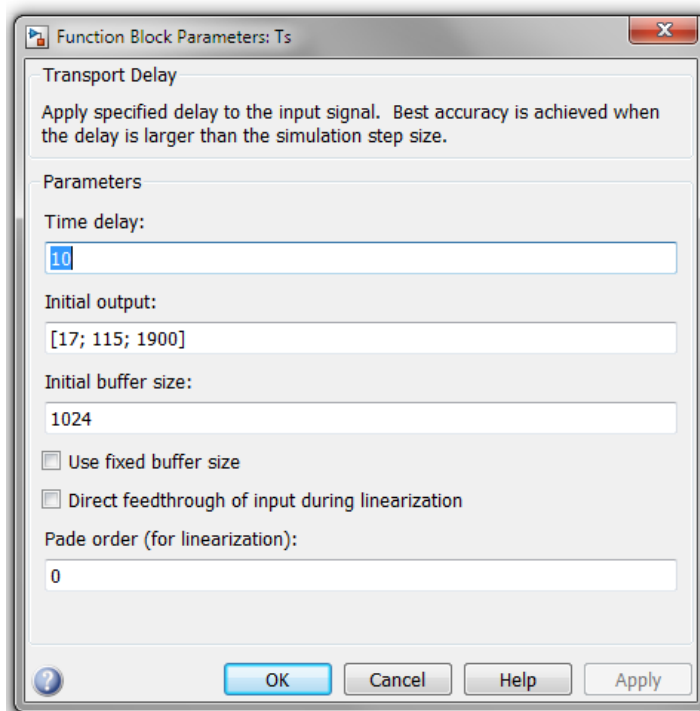


2.18. ábra: egy autópálya szakasz modellje

Az „Interpreted MATLAB Function” blokkban adjuk meg a rendszerdinamikát leíró „lwr.m” Matlab függvényt (2.19. ábra) (figyeljünk a kimeneti dimenziók számára is).

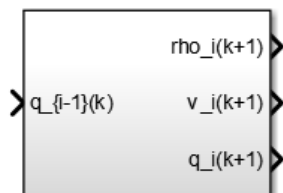


2.19. ábra: Matlab függvény meghívása Simulinkben



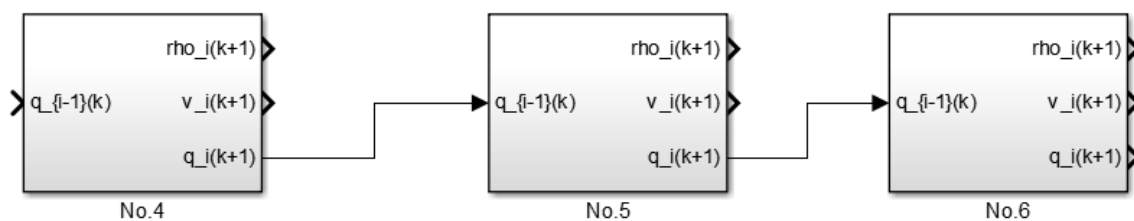
2.20. ábra: „Transport Delay” blokk paraméterezése

Az így elkészített - egy szakaszt szimuláló - modellünket helyezzük el egy „Subsystem” blokkba, mely a 2.21. ábrán látható formát nyeri!



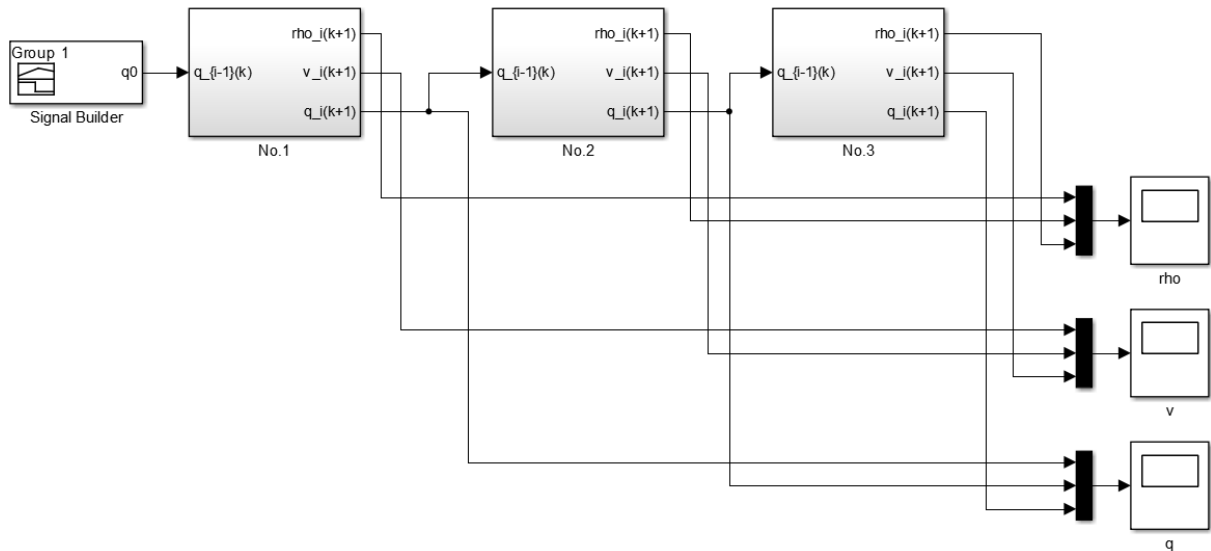
2.21. ábra: egy általános i. szakasz modellje alrendszerként

Az egyes szakaszokat reprezentáló alrendszereket fűzzük össze a 2.22. ábrán bemutatott módon (figyelembe véve a rendszert leíró (18)-(22) egyenleteket).



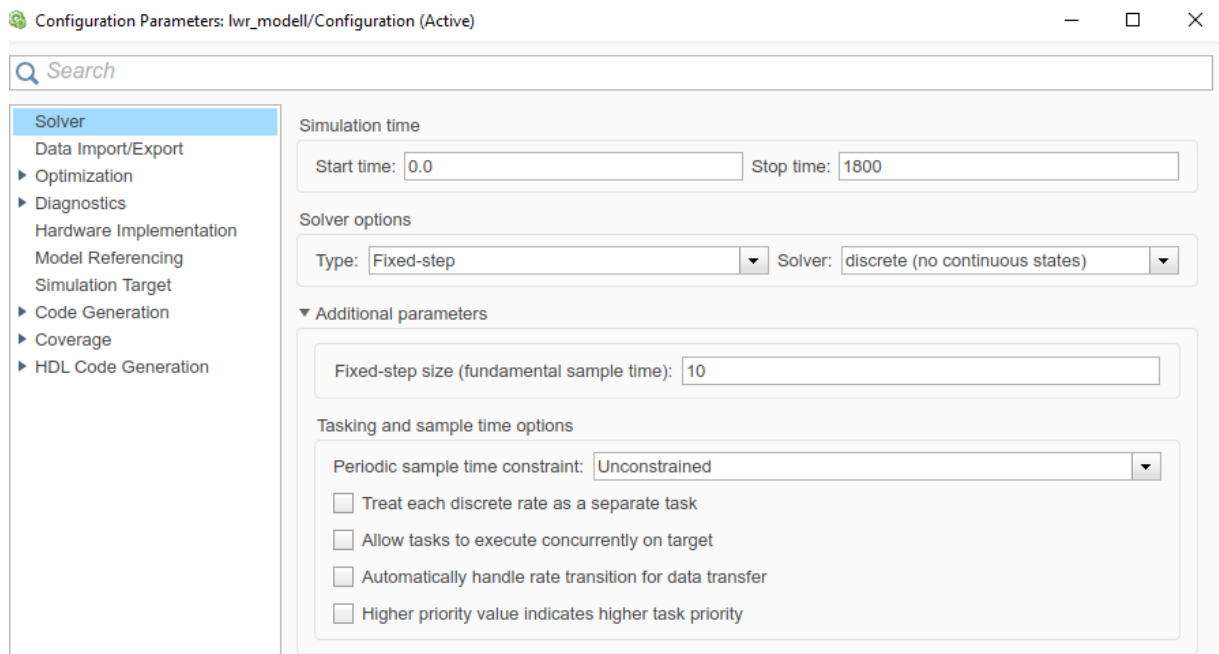
2.22. ábra: összefűzött alrendszerek

A szimulációhoz szükséges gerjesztést a „Signal Builder” blokkal adjuk meg, és az egyes szakaszok forgalmi változóit egy-egy „Scope” blokkban rajzoljuk ki. A teljes szimulációs rendszer áttekintő blokkvázlata a 2.23. ábrán látható.



**2.23. ábra: három szakaszból álló autópálya hálózat szimulációs vázlata**

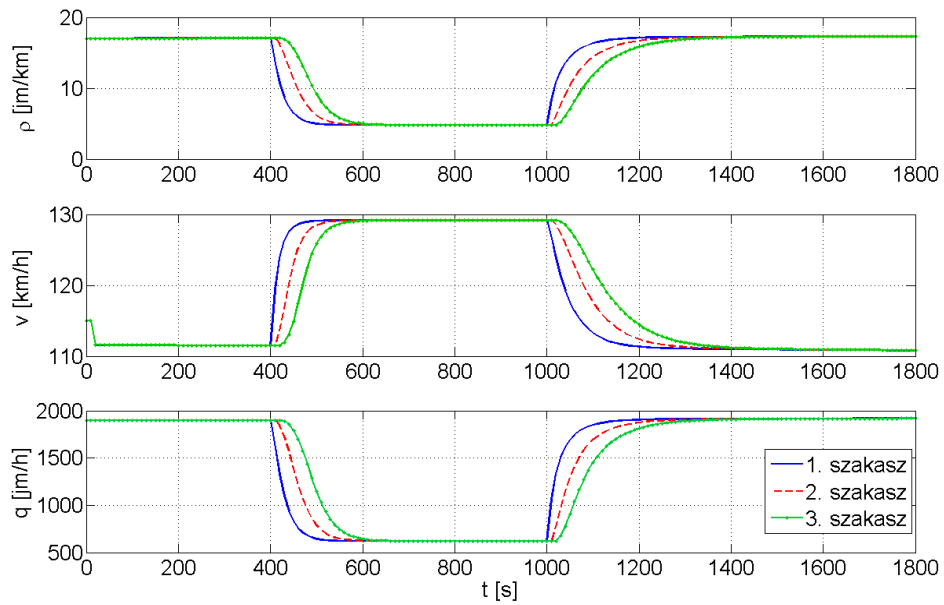
A simulink szimuláció beállításai a következők legyenek:



**2.24. ábra: a Simulink szimuláció paramétereinek beállítása**

Az összeállított Simulink modellen futtassunk szimulációt, a 2.17. ábrán megadott gerjesztéssel. A futtatás eredményét a 2.25. ábrán szemléltetjük.





**2.25. ábra: szimuláció eredménye**

A szimulációban a forgalom nagyság értékének függvényében változásokat tapasztalunk az egyes autópálya szakaszokon 400s és 1000s után. Vegyük észre, hogy a tranziensek jellege erősen függ a tranziens indulásakor jellemző forgalom sebességétől.

## 4. lecke: Lökéshullámok modellezése az LWR modell segítségével

### Elméleti áttekintés

Az LWR modell a főpályán érkező forgalom nagyság hatásának modellezésére alkalmas, eredeti formájában azonban nem képes a hátrafelé terjedő nagy sűrűségű közegek – ún. lökéshullámok szimulációjára. A lökéshullámok elméletéről részletesen a [2] és [3] művekben olvashatunk. Egy általános  $i$ . szakasz esetén a térben hátrafelé haladó információ tartalmazásához szükséges az  $i+1$ . szakasz forgalmi állapotának bevonása az  $i$ . szakasz forgalmát modellező differenciaegyenlet peremfeltételei közé. Az előző leckében felépített modellben ez nem történt meg, mivel a térben visszafelé tartó 'backward difference' numerikus approximációt végeztük.

Módosítsuk az eredeti modellünket: alkalmazzuk a (19) egyenlet helyett az alábbi összefüggést a forgalmi átlagsebesség modellezésére:

$$\hat{v}_i(k) = v_{i+1}(k) = v_{free} \exp\left(-\frac{1}{a} \left(\frac{\rho_{i+1}(k)}{\rho_{krit}}\right)^a\right) \quad (23)$$

ill. legyen

$$\hat{\rho}_i(k) = \frac{\rho_i(k) + \rho_{i+1}(k-1)}{2} \quad (24)$$

Tehát az  $i$ . szakaszon kialakuló forgalomsebességet az  $i$ . és  $i+1$ . szakaszon lévő forgalomsűrűségek függvényeként adjuk meg, így a szakaszt követő információ közvetve a rendszerdinamikát jellemző differenciaegyenlet peremfeltételébe került. A feladat az így módosított rendszermodell Simulinkben való megvalósítása, lökéshullám modellezése.

### Lökéshullámok modellezése Simulinkben

A továbbiakban az 1. feladatban elkészített 3 szakaszból álló autópálya hálózatot alkalmazzuk, ennek módosításait mutatjuk be.

A rendszert az alábbi változókkal gerjesztjük:

- Az első útszakaszra lépő forgalom nagysága  $q_0 = 1900 \text{ jm/h}$  legyen (Azaz az első Signal Builder blokkot módosítsuk!)
- Hozzunk létre majd egy másik Signal Builder blokkot is és definiáljuk az utolsó szakasz után következő, 4. szakasz sűrűségét:

$$\rho_4 = \begin{cases} 70 \frac{\text{km}}{\text{km}}, & \text{ha } 500 \leq t \leq 600 \\ 10 \frac{\text{jm}}{\text{km}}, & \text{ha } t < 500 \text{ és } t > 600 \end{cases} \quad (25)$$

ahol  $\rho_4$  a 4. útszakasz

Ezzel egy konstans forgalomra ható nagy sűrűségű gerjesztést adunk, voltaképpen dirac-delta gerjesztést adunk a rendszerre, mely a forgalomban történő hirtelen fékezés hatására kialakuló nagy lokális forgalomsűrűséget utánozza. A hirtelen sűrűségugrás hatására hátrafelé terjedő lökéshullám alakul ki.

A szimulációhoz módosítsuk az 1. feladatban felépített Simulink modellt az alábbi lépéseket követve:

Módosítsuk az  $i$ . szakaszt leíró LWR függvényt a következőre:

```
function y=lwr_lokes(u)
%% fgv. bemenetek
rho_i_k=u(1);
q_h_k =u(2); %h : i-1-edik szakaszra vonatk.
rho_i1_k=u(3); %rho_{i+1}(k-1)

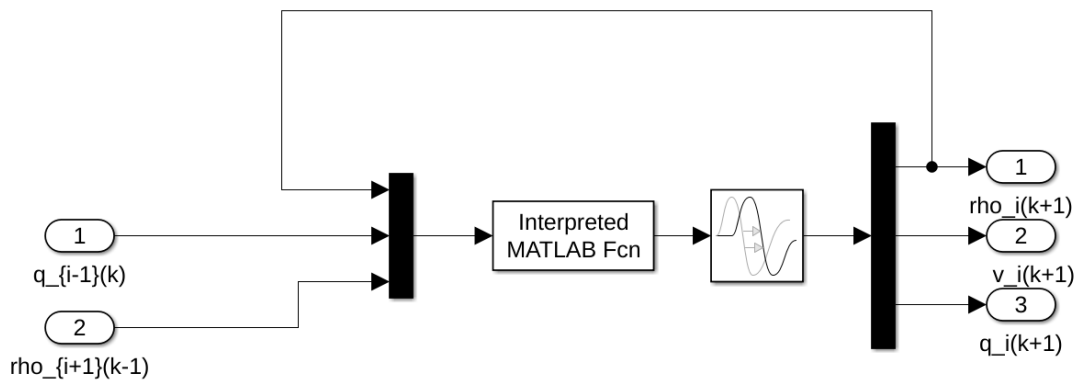
%% param.
Ts=10/3600; %J/h-ra kalkulalunk
L=1;
v_free=130;
rho_cr=25;%
a=2.5;

%% változók def.
v_hat_k=v_free*exp(-(1/a)*(rho_i1_k/rho_cr)^a);
rho_hat_k=0.5*rho_i_k+0.5*rho_i1_k;
rho_i_k1=rho_i_k+(Ts/L)*(q_h_k - rho_hat_k*v_hat_k); %rho_i(k+1)
v_i_k1=v_free*exp(-(1/a)*(rho_i_k1/rho_cr)^a); %v_i(k+1)
q_i_k1=v_i_k1*rho_i_k1; %q_i(k+1)

y=[rho_i_k1;
   v_i_k1;
   q_i_k1];

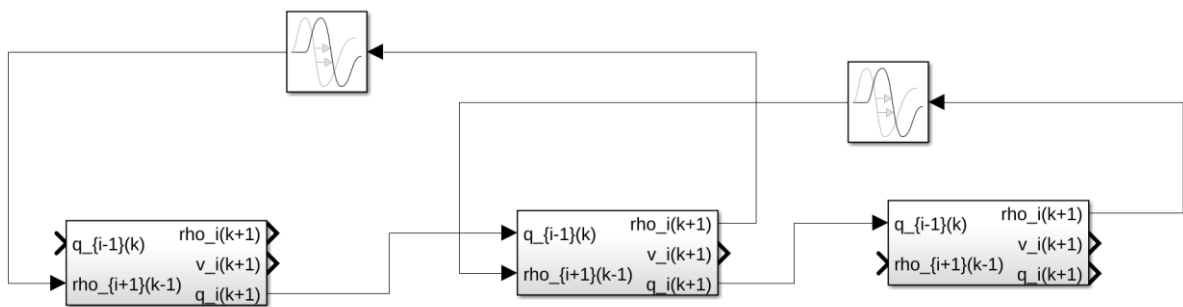
end
```

Az  $i$ . szakasz Simulink modelljén változtassuk meg az általános szakasz modelljének bemenet-számát a 2.26. ábra szerint.



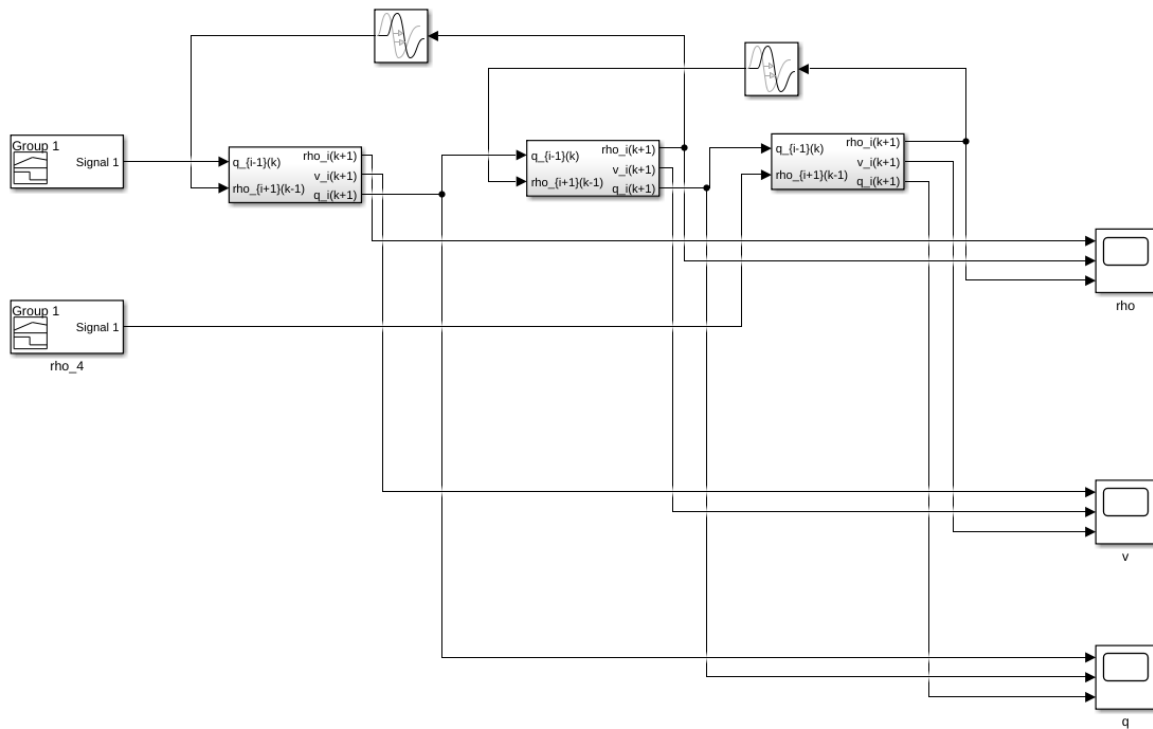
2.26. ábra: szakaszmodell módosítása

A módosított szakaszokat kössük össze a 2.27. ábrán bemutatott módon.



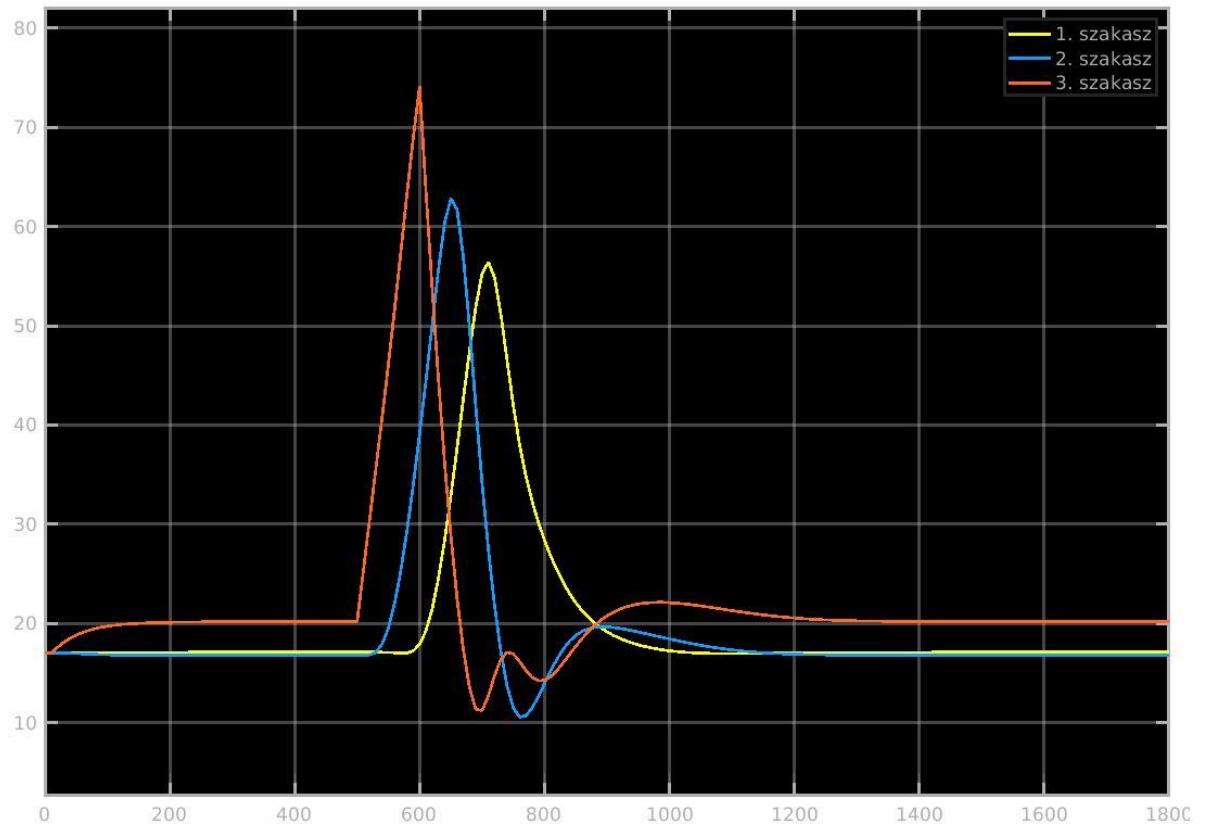
2.27. ábra: módosított szakaszok összekötése

A módosítások után az hálózat Simulink modelljét a 2.28. ábrának megfelelően fejezzük be.



2.28. ábra: módosított rendszer

Futtassunk szimulációt az elkészített rendszeren a (25) egyenlet által megadott gerjesztéssel. A szimuláció eredményeként látszik, hogy hátrafelé terjedő lökéshullámot generáltunk az autópályán.



2.29. ábra: szimuláció eredménye – a sűrűség megváltozása a 3 szakaszon

## 5. Irodalom a 2. modulhoz

- [1] Gazis, D. C., Herman, R. and Rothery, R. W. (1963): Analytical Methods in Transportation: Mathematical Car-Following Theory of Traffic Flow, Proceedings of the American Society of Civil Engineers, Engineering of Mechanical Division. Vol. 6, pp. 29-46.
- [2] M.J. Lighthill, G.B. Whitham: On kinematic waves II.: A theory of traffic flow on long crowded roads. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 229 (1955): pp. 317-345.
- [3] P.I. Richards: Shock waves on the highway. Operations Research, (4) (1956): pp. 42-51.
- [4] R. Courant, K. Friedrichs, H. Lewy: Über die partiellen Differenzgleichungen der mathematischen Physik. Mathematische Annalen, 100 (1928), (1), pp. 32-74.
- [5] Luspay Tamás – Tettamanti Tamás – Varga István: Forgalmirányítás: Közúti járműforgalom modellezése és irányítása. Typotex, 2011.
- [6] A. Messmer, M. Papageorgiou: METANET – A macroscopic simulation program for motorway networks. Traffic Engineering and Control, 31, 1990. pp. 466-470.

## 3. modul: Vissim mikroszkopikus forgalomszimulátor gyakorlatok

### Cél:

A tananyag célja a Vissim mikroszkopikus forgalomszimulátor bemutatása és alapvető kezelésének ismertetése gyakorlati példákon keresztül.

A jegyzet megírásához a szoftver 2023-as verzióját használtuk.

Fontos kiemelni, hogy az alábbi gyakorlatok csak a legfontosabb kezelési műveleteket fedik le. A program teljes és részletes ismertetése a Vissim User Manual című dokumentumban található (PTV, 2023).

### Követelmények:

Ön akkor sajátította el megfelelően a tananyagot, ha képes

- a járműkövetési forgalommodellezés elméletének megfogalmazására,
- a Vissim forgalommodelljének bemutatására,
- a Vissim forgalomszimulátor alapvető kezelésére.

### Időszükséglet:

A tananyag elsajátításához körülbelül 3-4 órára lesz szüksége.

### Kulcsfogalmak:

- mikroszkopikus közúti forgalomszimuláció,
- járműkövetési forgalommodell (Wiedemann modell),
- Vissim szimulátor gyakorlatok.



## 1. lecke: A Vissim bemutatása

A Vissim közúti forgalomszimulátor a karlsruhe-i székhelyű PTV Group (www.ptvgroup.com) terméke. A Vissim mikroszkopikus forgalommodellézést valósít meg, azaz a járműviselkedés individuális megközelítésén alapul. A szimulált közlekedési hálózat működése így részletesebben vizsgálható, és képi megjelenítése is valósághűen megvalósítható.



3.1. ábra: 3D-s Vissim pillanatkép (PTV, 2023)

A Vissim ma már széles körben elterjedt szoftver, amely hiteles referenciát jelent bármilyen közúti irányítási feladat megvalósítása során; legyen az valós közlekedésfejlesztési projekt vagy kutatás (Takács, 2004).

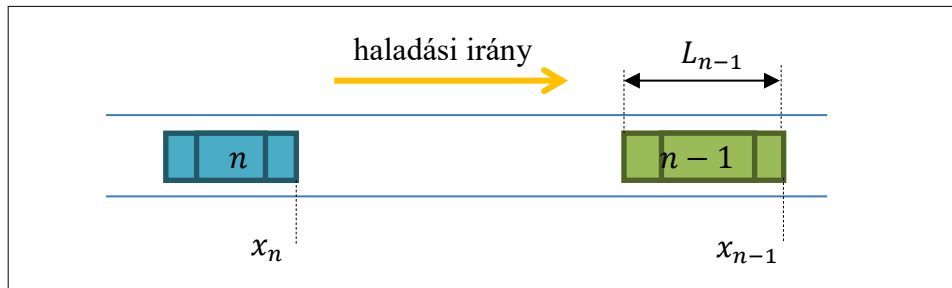
### 1. A mikroszkopikus járműkövetési modell

A Vissim egy ún. pszicho-fizikai járművezető-viselkedési modellre épül, amelynek elméleti alapja a Wiedemann járműkövetési modell (Wiedemann, 1974). A modell részletesebb bemutatása előtt (lásd következő fejezet) azonban a mikroszkopikus járműkövetési modellosztály ismertetése szükséges.

A járműkövetési modellek azon a feltételezésen alapulnak, hogy az egyedi járművek mozgását alapvetően az előttük haladó gépkocsi viselkedése határozza meg. Ezt az összefüggést felhasználva matematikai egyenletekkel írható le az egyes járművek dinamikája. A járműkövetési modell alapjelölései a következők (szemléltetésként lásd a 3.2. ábrát):

- $x_n$ : az  $n$ -edik jármű pozíciója, [m]
- $v_n$ : az  $n$ -edik jármű sebessége, [m/s]
- $a_n$ : az  $n$ -edik jármű gyorsulása, [m/s<sup>2</sup>]
- $\Delta x = x_{n-1} - x_n$ : követési távolság, [m]

- $\Delta v = v_n - v_{n-1}$ : sebességkülönbség, [m/s]
- $v_n^{desired}$ : az  $n$ -edik jármű kívánt sebessége (desired speed), [m/s]
- $L_{n-1}$ : az  $n - 1$ -edik jármű hossza, [m]
- $T$ : reakció idő, [s]



3.2. ábra: A járműkövetési modell jelölései

A járműkövetési modellek a követő autó viselkedését működési régiókkal írják le, tipikusan: szabadáramlási, követési, fékezési területekkel. A járművek megpróbálják elérni a számukra optimális (kívánt) sebességet - az adott régióban alkalmazott szabályrendszer szerint működve. Ebben a koncepcióban a követő jármű végig az előtte haladó járműtől függően választja meg az aktuális haladási sebességét.

## 2. A Vissim járműkövető modellje (Wiedemann modell)

A Vissim járműszimulációjának alapvető koncepciója - a járműkövetési modellosztálynak megfelelően - az egymást követő járművek interakciója, amelyben mindig csak a követő autós reagál. A gyorsabban haladó követő jármű vezetője akkor kezd lassítani, amikor az egyéni érzékelési küszöböt eléri az előtte haladó, lassabb jármű percepciója. Mivel a követő autó nem tudja pontosan meghatározni az elől haladó gépkocsi sebességét, így addig lassít, amíg egy másik érzékelési küszöböt el nem ér, és ekkor fog újra gyorsítani. A szimuláció tehát a lassulás-gyorsulás kapcsolatán alapuló, folyamatos iterációs eljárás.

A Wiedemann modellt a 3.3. ábrán látható működési régiók írják le, amelyekhez a következő küszöbértékek definiálása szükséges (Olstam & Tapani, 2004):

- $AX$ : a kívánt távolság álló járművek között (pl. lámpa előtti sor). Ez a küszöbérték magába foglalja az elől haladó jármű hosszát, valamint a megkívánt távolságot a követő jármű eleje és az elől haladó vége között:

$$AX = L_{n-1} + AX_{add} + AX_{mult} \cdot z_n$$

ahol  $AX_{add}$  és  $AX_{mult}$  hangolási paraméterek, és  $z_n$  egy járművezetőtől függő normál eloszlású paraméter [0,1] intervallumban.

- $d$ : a kívánt minimum követési távolság alacsony sebességkülönbségnél.

$$d = AX + BX$$

$$BX = (BX_{add} + BX_{mult} \cdot z_n) \cdot \sqrt{v}$$

ahol  $BX_{add}$  és  $BX_{mult}$  hangolási paraméterek, és  $v$  a következőképpen definiált:

$$v = \begin{cases} v_{n-1}, & \text{ha } v_n > v_{n-1} \\ v_n, & \text{ha } v_n \leq v_{n-1} \end{cases}$$

- $SDX$ : a maximális követési távolság, amely a fent definiált minimális követési távolságot felhasználva  $[1.5, 2.5] \cdot d$  intervallumban mozog.
- $SDV$ : közelítési küszöbérték, amikor a követő jármű vezetője észreveszi, hogy egy lassabban haladó kocsihoz közelít.

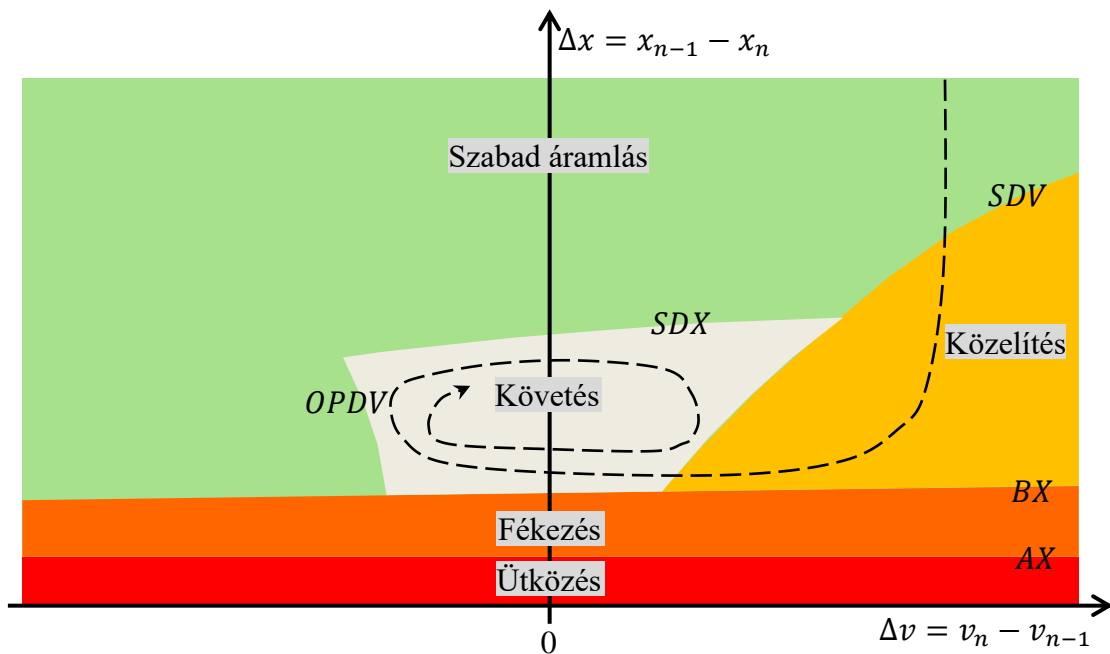
$$SDV = \left( \frac{\Delta x - L_{n-1} - AX}{CX} \right)^2$$

ahol  $CX$  egy hangolási paraméter.

- $OPDV$ : a növekvő sebességkülönbség küszöbértéke, amikor a követő jármű vezetője érzékeli, hogy lassabban halad, mint az előtte lévő jármű.

$$OPDV = SDV \cdot (-OPDV_{add} - OPDV_{mult} \cdot w_n)$$

ahol  $OPDV_{add}$  és  $OPDV_{mult}$  kalibrációs paraméterek, és  $w_n$  egy normál eloszlású hangolási paraméter.



3.3. ábra: Különböző küszöbértékek és régiók a Wiedemann modellben

A Wiedemann modell működési tartományai:

1. Szabad áramlási tartomány: ebben a zónában a jármű az őt körülvevő forgalomtól függetlenül mozog, ezért a számára meghatározott kívánt sebességet maximális gyorsulással éri el. Az alkalmazandó  $b_{max}$  maximális gyorsulás egy kalibrációs paraméter.

2. Közelítési tartomány: az *SDV* küszöbön áthaladva a jármű érzékeli, hogy egy nála lassabb autót fog „beérni”. Az ütközés elkerülése érdekében a követő jármű az alábbi negatív gyorsulási értékkel lassít le:

$$b_n = \frac{1}{2} \cdot \frac{(\Delta v)^2}{d - (\Delta x - L_{n-1})} + b_{n-1}$$

ahol  $b_{n-1}$  a követett autó lassulási értéke.

3. Követési tartomány: az *OPDV*, *SDX*, *SDV*, és *BX* határvonalak által közrefogott zóna. Amennyiben a jármű az *SDV* vagy a *BX* határon keresztül érkezik ebbe a tartományba,  $-b_{null}$  lassulási érték alkalmazandó. Ha pedig az autó az *OPDV* vagy az *SDX* küszöbön keresztül hajt be a zónába,  $b_{null}$  gyorsulással kell számolni.  $b_{null}$  egy kalibrációs paraméter.

4. Fékezési tartomány: amennyiben a követő jármű eleje és a követett autó vége közötti távolság kisebb, mint  $d$ , az alábbi negatív gyorsulás alkalmazandó:

$$b_n = \frac{1}{2} \cdot \frac{(\Delta v)^2}{d - (\Delta x - L_{n-1})} + b_{n-1} + b_{min} \cdot \frac{d - (\Delta x - L_{n-1})}{BX}$$

ahol  $b_{min}$  egy hangolási paraméter.

5. Ütközési tartomány: ez tulajdonképpen csak elméleti terület, mivel a Vissimben a járművek nem tudnak ebbe a zónába eljutni.

A 3.3. ábrán - szaggatott vonallal - egy követő jármű mozgása is be van jelölve a szemléletesség kedvéért. A berajzolt trajektória jól reprezentálja a különböző működési területeken való mozgást: a jármű egy szabadáramlási helyzetből kiindulva közelít meg egy másik (előtte haladó) autót elérve a közelítési tartományt, majd a követő zónába jutva mozog tovább.

## 2. lecke: Hálózat felépítése

### 1. A jobb egérgomb funkció – objektumok létrehozása a Vissim modellben

A Vissimben nagy jelentősége van a jobb egérgombnak, mivel az objektumok létrehozását azzal végezzük. A telepítés utáni első alkalommal a program megkérdezi a jobb egérgombhoz rendelt alapfunkciót. Ez lehet az új objektum létrehozása vagy a (Windows környezetben megszokott) legördülő menü megjelenítése. E kettő közül bármelyik kiválasztható a **jobb egérgomb** alapfunkciójaként, és a másik pedig a **Ctrl + egér jobb gombhoz** lesz hozzárendelve.

A jobb egérgomb e beállítása utólag is módosítható az „Edit/User Preferences” menüben a „GUI/Network Editor” menü alatt.

A Vissim alaplogikája objektumok létrehozására a modellben a következő:

- a „Network Objects” menüsávban kiválasztjuk az adott elemet (pl. a legelső a „Links”);
- a „Network Editor” ablakon belül tetszőleges helyre kattintunk az egér jobb gombjával (a másik opció szerint Ctrl + egér jobb gombbal).


Ez a létrehozási logika természetesen az összes objektumra igaz a „Network Objects” menüsávban!


A következő fejezetekben egységesen úgy mutatjuk be a példákat, hogy az egér jobb gomb alapfunkciója az objektum létrehozás és a Ctrl + egér jobb gombhoz pedig a legördülő menü funkció van rendelve.

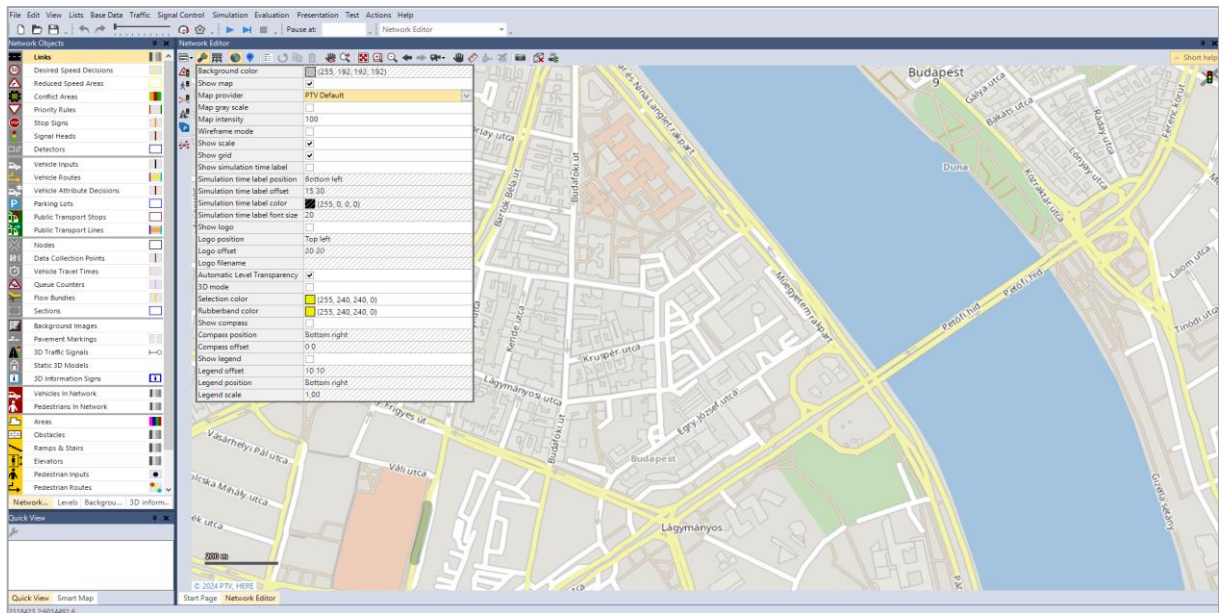
### 2. Térképháttér behívása

A Vissim hálózati elemeinek létrehozása előtt hasznos lehet egy alap háttér betöltése (pl. térkép vagy csomóponti helyszínrajz). A Vissimbe egyaránt betölthetők online térképszolgáltatók (Open Street Map, Bing Maps) térképei, ill. raszter- és vektorgrafikus képek is (pl. bmp, jpg, dwg, shape, stb.) .

Elsőként vegyük végig egy térképszolgáltató által biztosított háttértérkép beállítását:

1. a „Network Editor” ablak tetején található eszköztárból kiválasztjuk a „villáskulcs” ikont () , amely az alapvető grafikai beállításokat tartalmazza;
2. a legördülő menüben beállítjuk a térképszolgáltatót („Map provider”) és a „Show map” funkcióval a térkép láthatóságát (lásd 3.4. ábra);
3. ezután tetszőleges helyen elkezdhetjük az úthálózatunk megrajzolását, a Vissim ugyanis innentől fogva megfelelően lesz beskálázva, azaz bármekkora nagyításban is van a „Network Editor” ablakunk (térképünk) a „Linkek” méretarányosan kerülnek benne megrajzolásra.

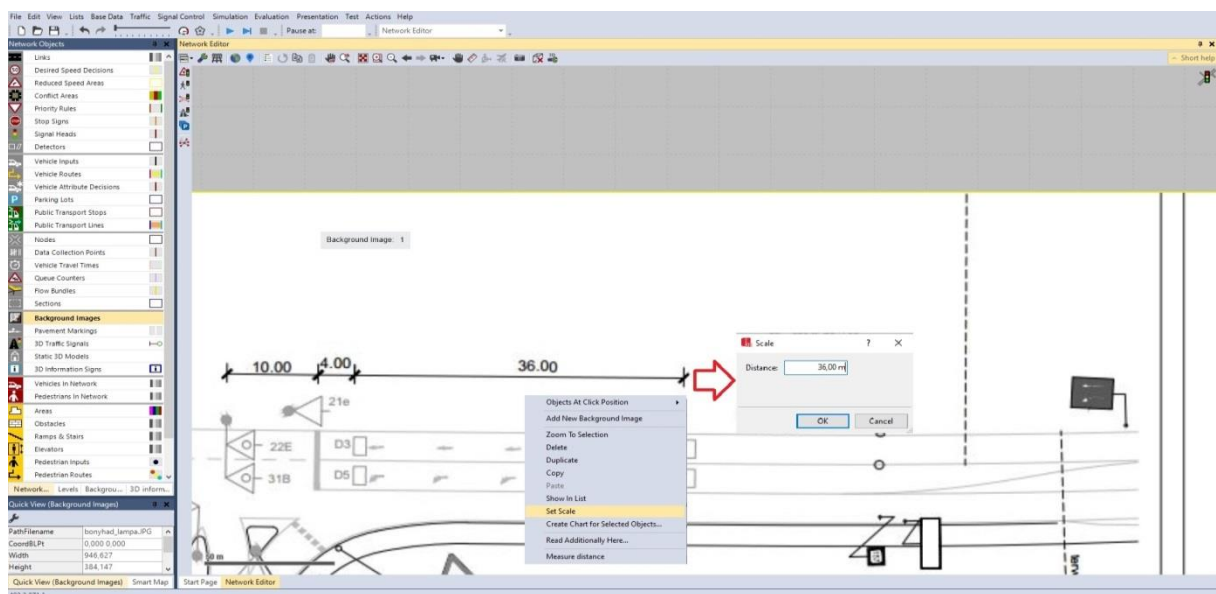
A térképháttér egyébként közvetlenül a „Network Editor” ablak eszköztárának térkép ikonjával () is ki- és bekapcsolható.



3.4. ábra: Háttér megjelenítése és beszkálázása

A folytatásban vegyük végig egy kép háttérként való beállítását:

1. a „Network Objects” menüsávból kiválasztjuk a „Background Images” menüt;
2. a „Network Editor” ablakon belül tetszőleges helyre kattintunk az egér jobb gombjával, ezáltal behívható az adott képfájl;
3. a kép betöltése azonban – a térképháttérrel ellentétben - már nem jár automatikus skálázással természetesen, emiatt a hátteret utólagosan kell a Vissim geometriájához méretezni a legördülő menü (Ctrl + jobb egérgomb benyomásával a képre kattintva) „Set scale” parancsával (lásd 3.5. ábra).
4. A beszkálázást legegyszerűbben a háttérkép egy ismert méretű elemével tudjuk megtenni, pl. egy térkép megadott léptékével. Ugyanakkor javasolt minél hosszabb táv alapján skálázni, mert így a méretezési hiba kisebb lesz. A skálázáshoz a jobb egérgombot tartjuk végig lenyomva, és úgy húzzuk végig a léptéken.



3.5. ábra: Képháttér beszkálázása



## Gyakorlás:

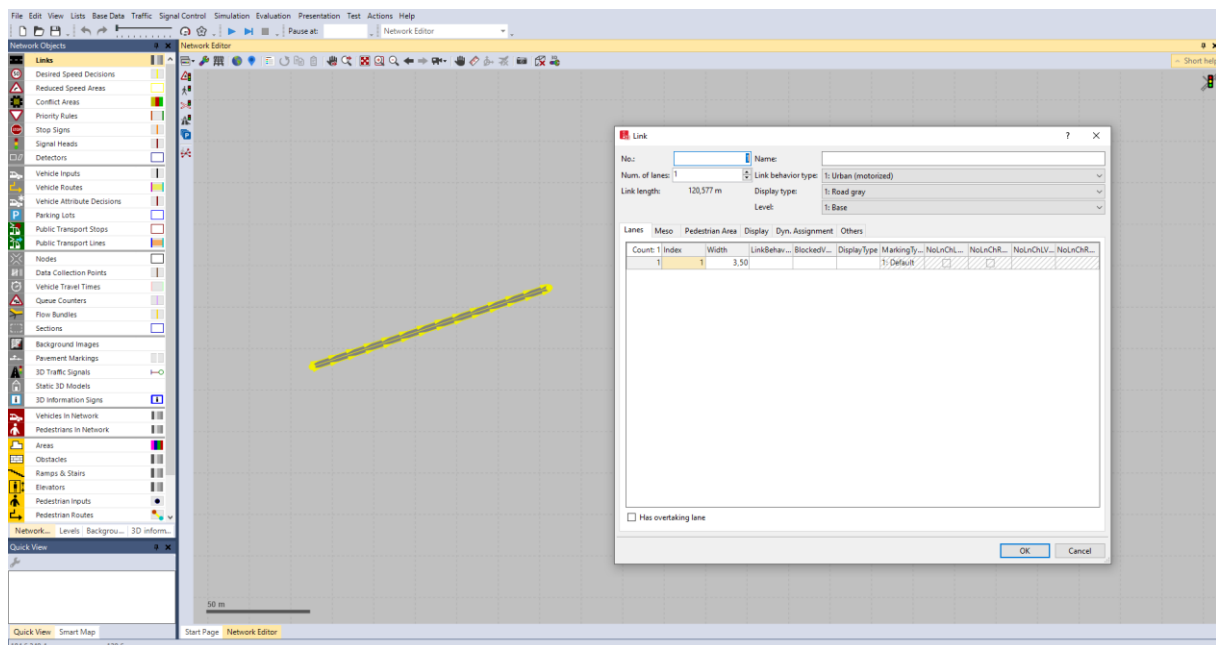
- Állítsuk be az Bing térképet háttérként és rajzoljunk be segítségével egy tetszőleges budapesti belvárosi útszakasz (lásd következő fejezet)!
- Figyeljük meg különböző nagyításban a méretarány megfelelőségét!
- Vissimbe Hívjunk be egy méretezett vagy léptékkel ellátott helyszínrajzot (lásd pl. 3.5. ábra) a fent leírt módon!
- A betöltés után skálázzuk be a hátteret!

## 3. Úthálózat létrehozása

A Vissim úthálózat útszakaszokból („Links”) és az őket összekötő kapcsoló elemekből („Connectors”) áll össze. Úthálózat létrehozásához a „Network Objects” menüsáv „Links” menüjét kell kiválasztani (lásd 3.6. ábra).

Először a „Linkeket” (útszakaszokat) kell létrehozni az alábbi lépések szerint:

- jobb egérgombbal kattintás a felületre, amivel tulajdonképpen a szakasz kezdőpontját adjuk meg;
- majd - az egérgombot lenyomva tartva - elhúzás tetszőleges irányba és hosszon;
- a gomb felengedésével felugrik egy „Link” ablak (lásd 3.6. ábra), ahol részletes beállításokat végezhetünk az adott szakaszra vonatkozóan, és ezután létrejön az útszakasz.

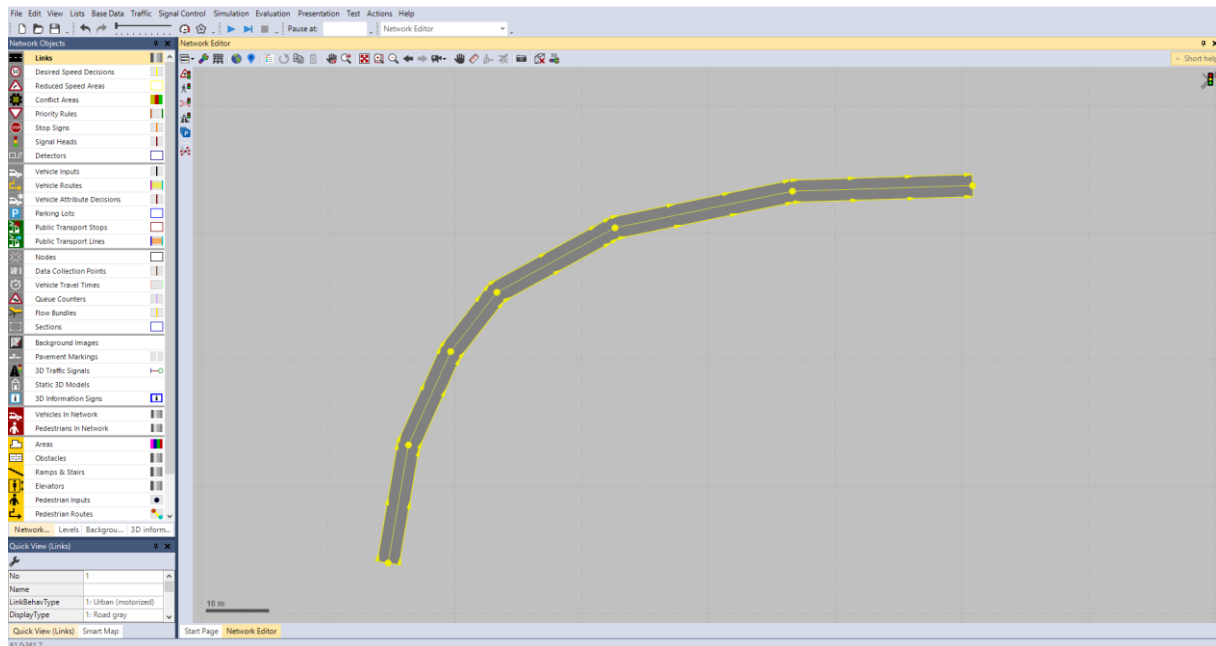


3.6. ábra: „Link” létrehozása

A kész szakaszt természetesen utólag is módosíthatjuk. Duplán kattintva (bal egérgombbal) a meglévő szakaszra a fent említett „Link” ablak ugrik fel.

Ezen kívül a létrehozott útszakasz geometriája is könnyedén változatható:

- a „Link” eleje és vége tetszőlegesen nyújtható, ill. rövidíthető - a szakasz elejének vagy végének a megragadásával,
- rákattintva a szakaszra bal egérgombbal, a „Link” mozgathatóvá válik („drag and drop” módszerrel),
- az útszakaszokat köztes pontok beszúrásával tudjuk ívesíteni (a meglévő szakaszra kattintva jobb egérgombbal); a köztes pontokat pedig bal egérgombbal lehet ezután mozgatni (lásd 3.7. ábra).



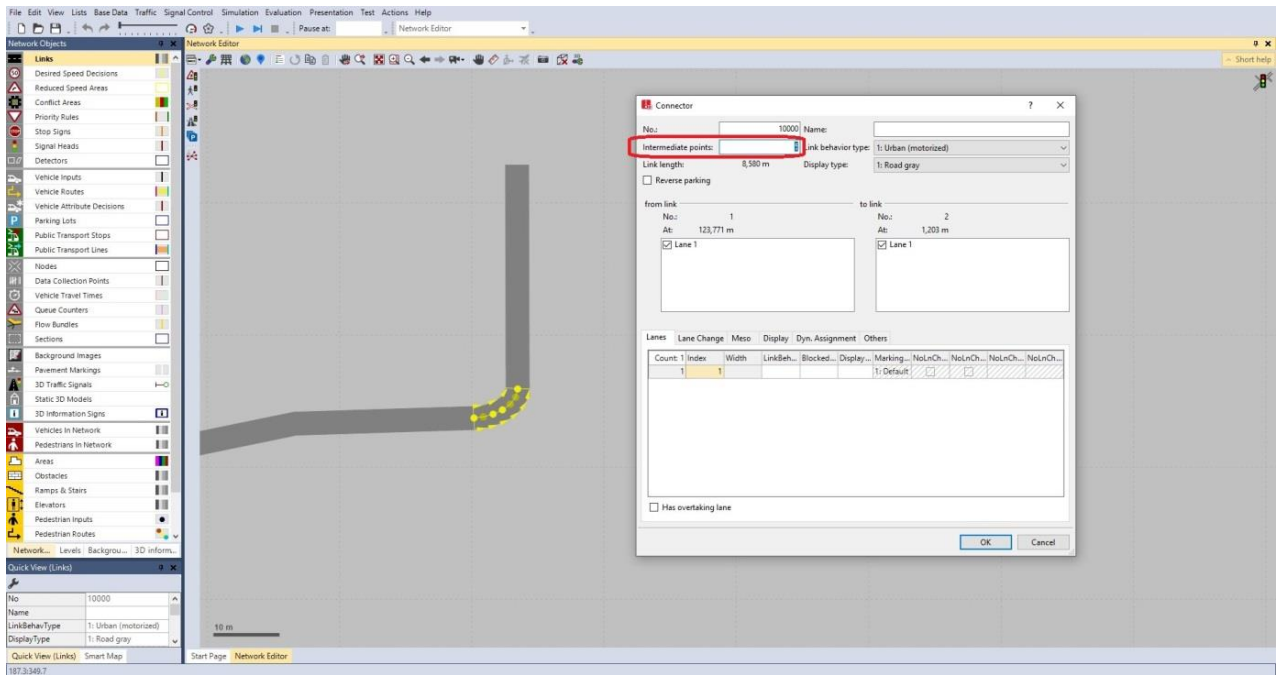
3.7. ábra: „Link” ívesítése

A „Linkek” létrehozása után lehetséges a kapcsoló elemek („Connectors”) alkalmazása. Tulajdonképpen a kapcsolók keltik életre a hálózatot, hiszen segítségével teremthető meg a fizikai kapcsolat az egyes útszakaszok között.

Kapcsoló elemek létrehozása két „Link” összekötésére az alábbiak szerint történik:

- jobb egérgombbal kattintás a kiinduló szakaszra (ekkor már megjelenik a „Connector”), amivel tulajdonképpen a kapcsoló kezdőpontját adjuk meg,
- majd - az egérgombot lenyomva tartva - ráhúzzuk a nyilat a cél „Link”-re (ez lesz a kapcsoló végpontja),
- a gomb felengedésével felugrik egy „Connector” ablak (lásd 3.8. ábra), ahol részletes beállításokat végezhetünk az adott kapcsolóra vonatkozóan, és ezután létrejön a két szakaszt összekötő „Connector”.





### 3.8. ábra: „Connector” létrehozása

A kapcsoló elemeket a „Link”-hez hasonlóan ívesíthetjük, illetve további lehetőségként az „Intermediate points” opciót (lásd a 3.8. ábra) alkalmazva automatikus ív rajzolása lehetséges (ez kanyarodó ívek létrehozásakor előnyös).

Ha egy „Connector” egy adott „Link”-hez annak végpontjánál kapcsolódik, akkor ügyeljünk arra, hogy a „Connector” a lehető legkisebb mértékben lógjon rá a „Link”-re. Amennyiben ugyanis a „Link” és a „Connector” átfedésben vannak, az átfedett „Link” területre helyezett egyéb hálózati elemek nem működnek megfelelően (pl. Signal head, Desired speed decision).

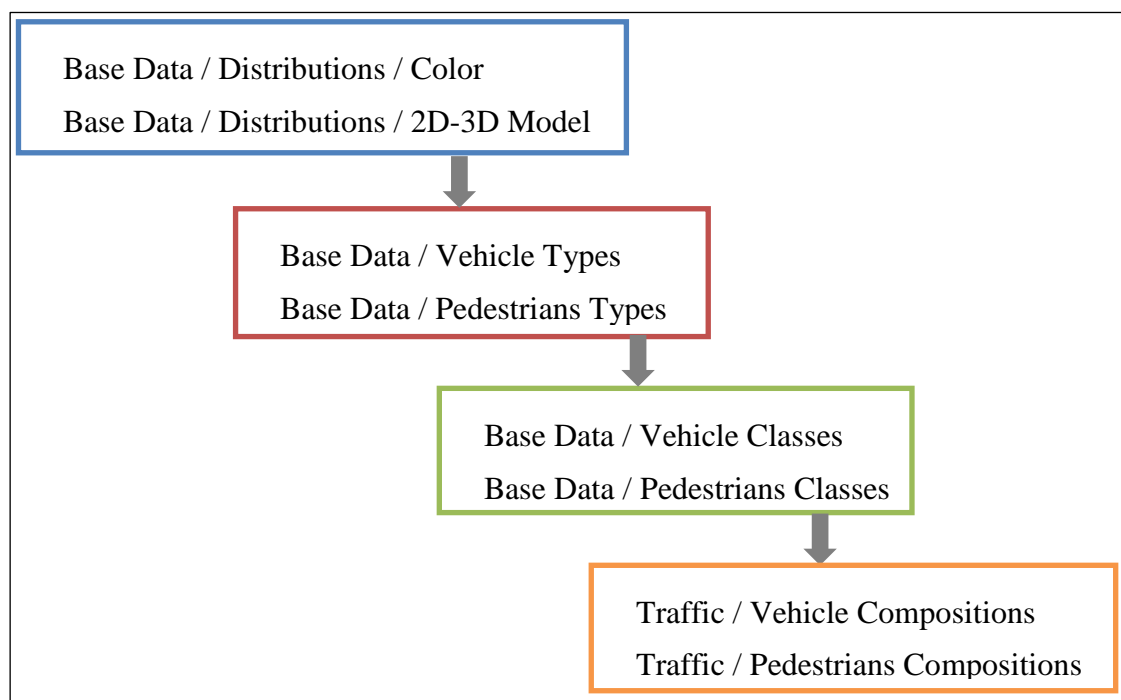
#### Gyakorlás

- Egyszerű úthálózat gyanánt hozzunk létre egy klasszikus X-alakú csomópontot („Linkek” és kapcsolók megfelelő alkalmazásával)!
- Az úthálózatban kétirányú utcák szerepeljenek!
- Készítsünk egy gyalogos szakaszt is (gyakorlatilag egy zebrát) a csomópont egyik ágára (egyszerűen az adott „Linke”-re helyezve a gyalogos „Linket” keresztbe)!

### 3. lecke: Forgalom generálása

#### 1. Járművek és gyalogosok beállítása

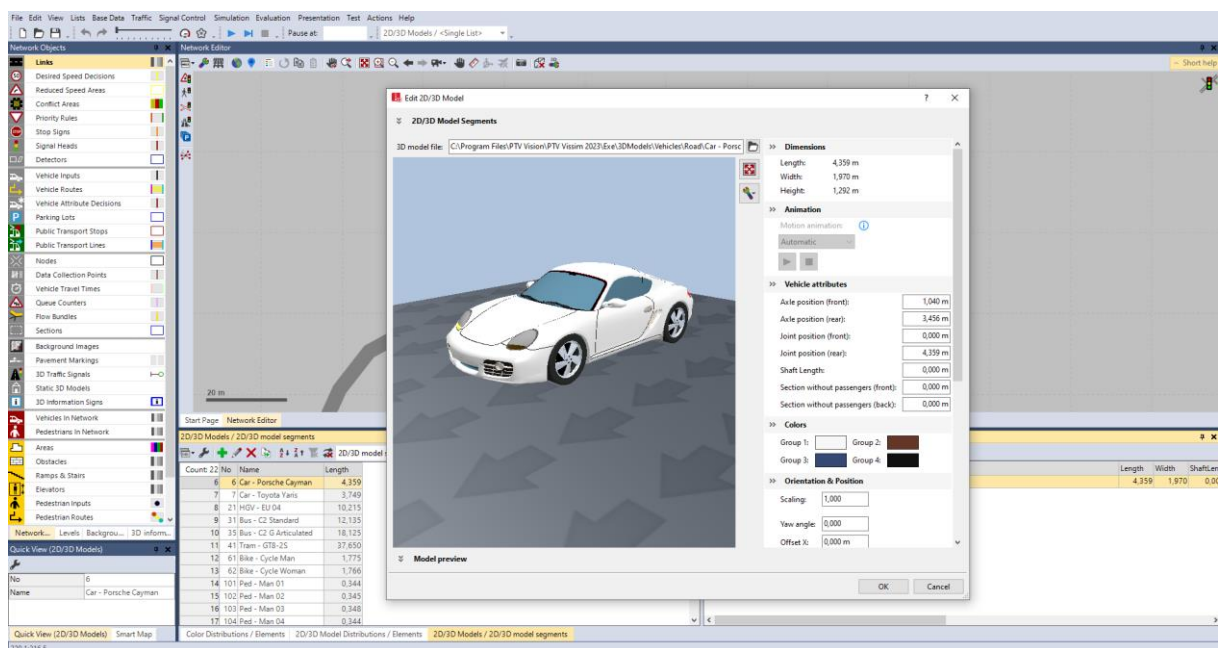
A Vissimben részletes beállítási lehetőség áll rendelkezésre a szimulációban alkalmazott járműparkot illetően. A forgalom megadása alapvetően kompozíciókban történik („Vehicle Compositions”). Egy kompozíció gyakorlatilag különböző tulajdonságú járművek, különböző arányú összetételét jelenti (szabodon konfigurálható módon). A kompozíciók beállítása előtt azonban végig kell menni a részletesebb járműbeállításokon is. A beállítási logika megértését a 3.9. ábrával segítjük.



**3.9. ábra: A jármű-, ill. gyalogos-beállítás hierarchikus logikája Vissimben a menüpontok szerint**

- Az egyéni járműszínek definiálása a „Base Data/Distributions/Color” menüben történik.
- A „Base Data/Distributions/2D-3D Model” menüben egyrészt az alkalmazott járművek eloszlása állítható be.
  - A „2D-3D Model” almenüben továbbá a járművek geometriai beállításai konfigurálhatók. Ennek részeként 3D-s járművek („v3d” kiterjesztéssel) betöltése lehetséges a Vissim telepítési mappa „3DModels” almappájából (lásd 3.10. ábra). További 3D-s autók letöltése lehetséges innen: <https://us-resources.ptvgroup.com/en-us/ptv-support/ptv-vissim-3d-vehicle-models>.

Fontos megjegyezni, hogy amennyiben több elemű járművet szeretnénk létrehozni (pl. villamos), azt is a fenti felületen lehet megtenni a járműegységek csoportba foglalásával.



3.10. ábra: „v3d” járművek betöltése a Vissimbe

- A „Base Data/Vehicle Types” menüben járműtípusokat definiálhatunk. Ennek során a „Base Data/Distributions/Color” és „/2D-3D Model” menükben (már korábban) meghatározott színeket és járműmodelleket választhatjuk ki. Továbbá a kívánt és maximális gyorsulási paraméterek is itt állíthatók be.
- A „Base Data/Vehicle Classes” menüben az előző lépésben létrehozott típusokat választhatjuk ki. Az így beállítható járműosztályok több funkcionalitás használatakor jönnek szóba (pl. „Routes”, „Desired Speed Decisions”).

A fenti folyamat elvégzésére leginkább azért van szükség, hogy tetszőleges járműösszetételt alkalmazhassunk a forgalom generálásakor. A „Traffic/Vehicle Composition” menüben ugyanis az előzőleg definiált járműtípusokból („Vehicle Types”) lehet járműösszetételt létrehozni.

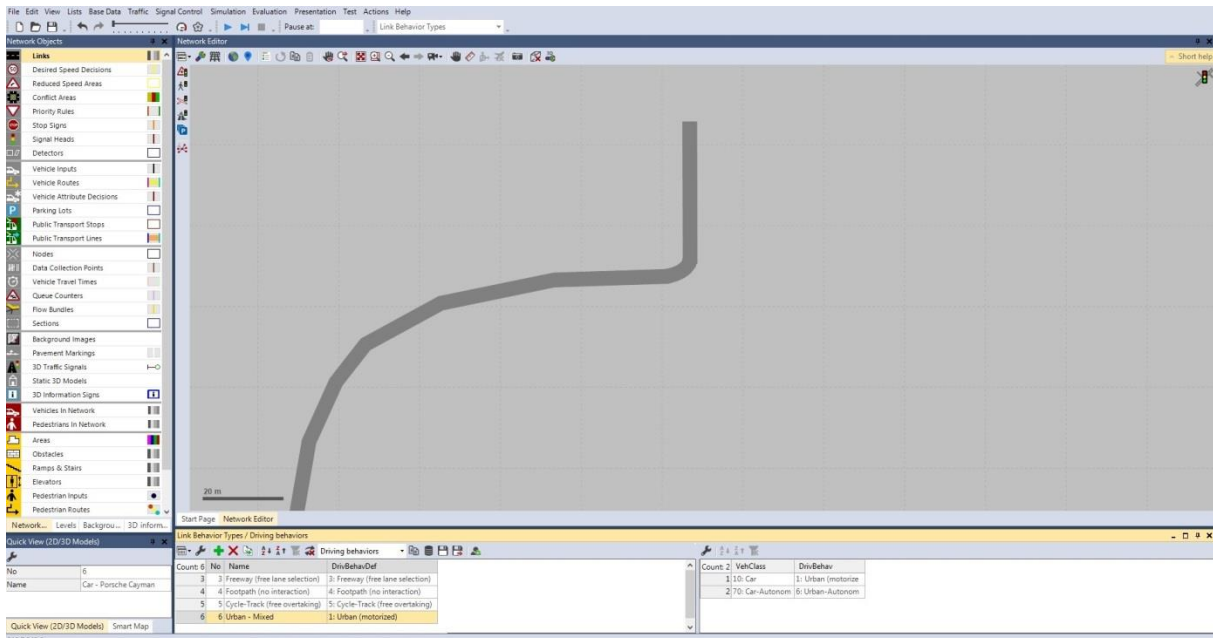
A gyalogosok („Pedestrians”) konfigurálása a járművek beállításával megegyező logikával történik.

## 2. Vegyes járművezetői viselkedés beállítása – autonóm járműforgalom modellezése

A Vissimben a járműparaméterek beállításán felül lehetséges a járművezetői viselkedés beállítása is, egyszerre akár többféle magatartás keverésével. Utóbbi szempont kiemelten fontos olyan esetekben, amikor a járműfolyamot nem tekintjük homogénnek. A valóságban ilyen példa lesz várhatóan az autonóm és a hagyományos járművek keveredésének időszaka.

A beállítás a „Base Data/Link Behavior Types” menüpontban lehetséges. A vegyes járművezetői viselkedés beállításának előfeltétele, hogy definiáljunk többféle járműosztályt a „Base Data/Vehicle Classes” menüpontban, valamint hozzájuk kapcsolódóan 1-1 járművezetői viselkedést a „Base Data/Driving Behaviors” menüpontban. Ezt követően vegyíthetők a különböző járműosztályok (autonóm és hagyományos), a 3.11. ábra szerint.

Az alsó egység bal oldalán a „+” ikonra kattintva vehető fel új járművezetői viselkedéscsoport, míg a jobb oldalon a „+” ikonra kattintva megadható, hogy mely járműcsoportra („VehClass”) milyen járművezetői magatartás („DrivBehav”) legyen jellemző. Amely járműcsoportra nem állítunk be itt értéket, az a „DrivBehavDef” járművezetői magatartás szerint fog viselkedni (lásd az alsó egység bal oldalán).



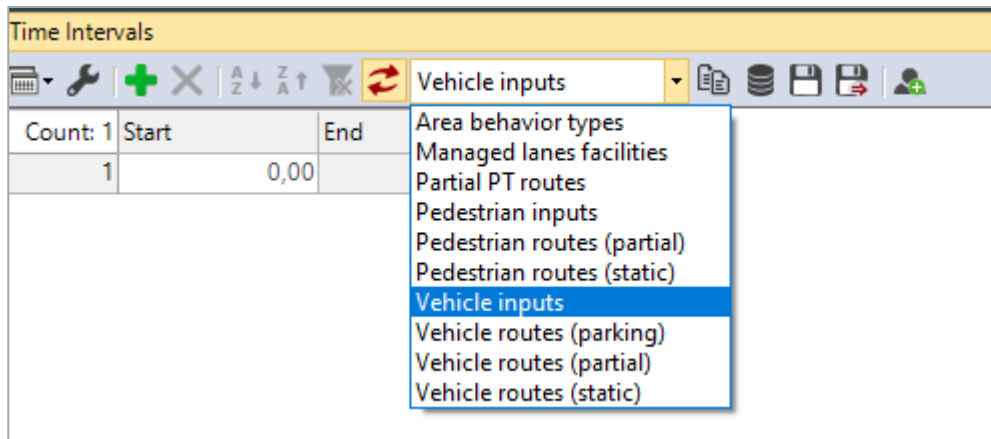
3.11. ábra: Vegyes járművezetői viselkedés beállítása

A fentebb beállított vegyes viselkedés az útszakaszokra vonatkozik. Tehát az útszakaszokra vonatkozóan be kell állítani a vegyes járművezetői viselkedést, ezt a Link tulajdonságai között tudjuk megtenni, a „Link Behavior Type” beállítandó a megfelelő értékre (lásd: 3.6. ábra). A vegyes járművezetői viselkedés értelemszerűen akkor fog megvalósulni, ha az útszakaszra vegyes forgalmat bocsátunk. Ezt részletesen lásd a Forgalmi igények és útvonalak generálása c. fejezetben.

### 3. Időintervallumok („Time intervals”)

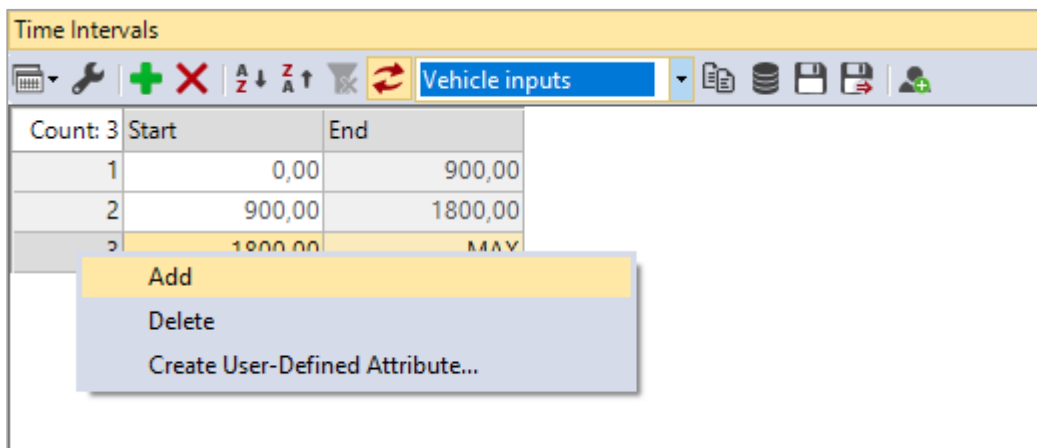
Az időintervallum beállításának külön jelentősége van a Vissimben, mivel számos objektum (nem csak e fejezetben bemutatottak) az időintervallumok szerint is konfigurálható. Ezek közül a legfontosabbak: az útvonal beállítások („Routes”) és a jármű vagy gyalogos bemenetek („Vehicle Inputs”, „Pedestrian Inputs”).

Az időintervallum beállítása a „Base Data/Time Intervals” menün keresztül lehetséges. A menüpontra való kattintással a Vissim ablak alsó sávjában megjelenő „Time Intervals” ablakban lehet intervallumokat beállítani az adott objektumra vonatkozóan (lásd 3.12. ábra).



3.12. ábra: Objektum kiválasztása „Time Intervals” beállításhoz

Alapbeállításaként egyetlenegy intervallum jelenik meg a táblázatban: 0 másodperctől a „MAX” értékig, amely a szimulációs periódust jelenti (alapbeállításaként 3600 sec). További időintervallumot úgy tudunk hozzáadni a táblázathoz, hogy az utolsó táblázatsorba jobb egérgombbal kattintva a legördülő menüből az „Add” parancsot választjuk (lásd 3.13. ábra).



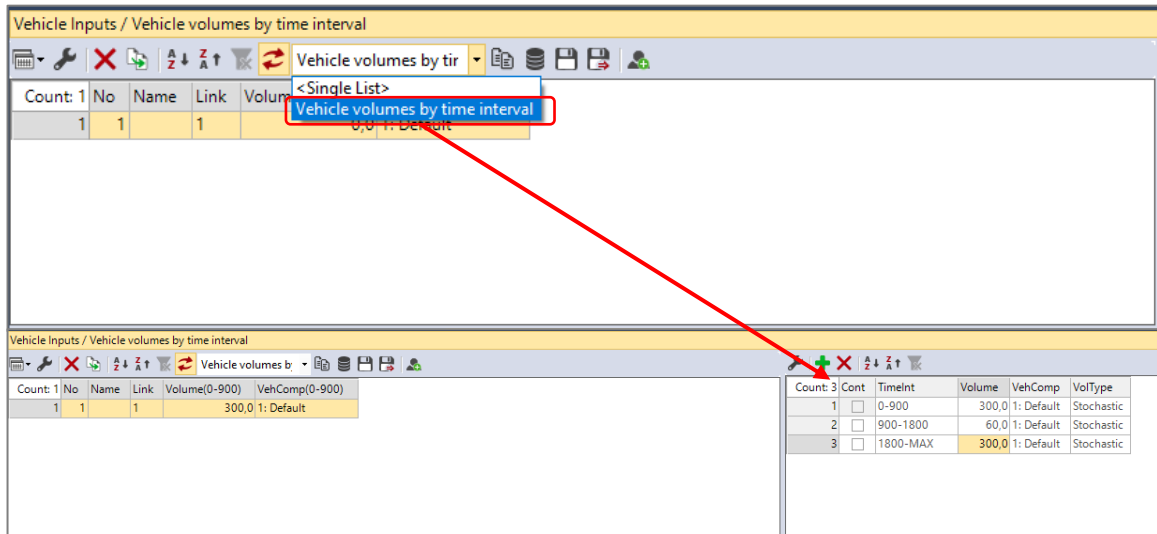
3.13. ábra: Időintervallum hozzáadása

#### 4. Forgalmi igények és útvonalak generálása

A Vissim úthálózatba a „Network Objects” menüsáv „Vehicle Inputs” funkciójával generálhatunk forgalmi igényeket. Járműforgalmi bemenet csak az adott „Link” elején képezhető (automatikusan oda rakja le a Vissim) az alábbiak szerint:

- jobb egérgombbal kattintás az adott szakaszra, ekkor egy fekete csík kerül az „Link” bemeneti végére jelezve, hogy egy „Vehicle Input” lett hozzárendelve;
- a jármű bemenetek listáját jelenítsük meg alul a következő módon: a „Network Objects” menüsáv „Vehicle Inputs” funkciójára kattintva (jobb egérgombbal) a legördülő menüből válasszuk a „Show List” parancsot;
- ezután a megjelent alsó sávban az adott forgalmi igény beállításai következnek;

- ekkor érdemes a legördülő menüből az időintervallumos megjelenítésre („Vehicle volumes per interval”) váltani (lásd 3.14. ábra);



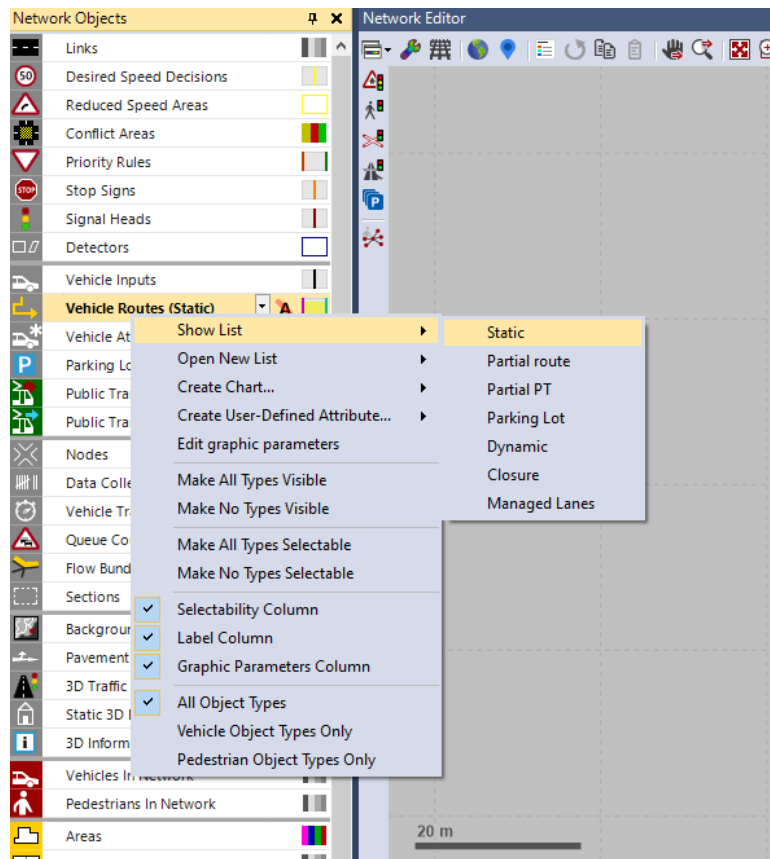
**3.14. ábra: Forgalmi igények beállítása**

- a „Vehicle Inputs” menüsávban a „Link” automatikusan kitöltődik azon szakasz számával, amire kattintottunk előzőleg (de ez akár módosítható is),
- a menüsáv utolsó két oszlopában a forgalomnagyság (jármű/óra) és a járműkompozíció (fenti példában: „Default”) beállítása szükséges,
- a kompozíciónál egyébként a „Traffic/Vehicle Compositions” menüben definiált járműösszetételek közül lehet választani (lásd Járművek és gyalogosok beállítása c. fejezet),
- fontos, hogy a jobb oldalon megjelenített részen az adott „Vehicle Input” objektumhoz tartozó - előzőleg már definiált időintervallumokhoz (lásd előző fejezet) - külön-külön forgalomnagyság állítható be (ezzel dinamikus igényforgalmat generálhatunk).

Realisztikus forgalomszimuláció elérése érdekében a forgalmi igények beállítása mellett a járművek által választott útvonalak (honnan-hová kapcsolatok) beállítása szükséges. Ez a „Network Objects” menüsáv „Routes” funkciójával lehetséges (lásd 3.15. ábra).

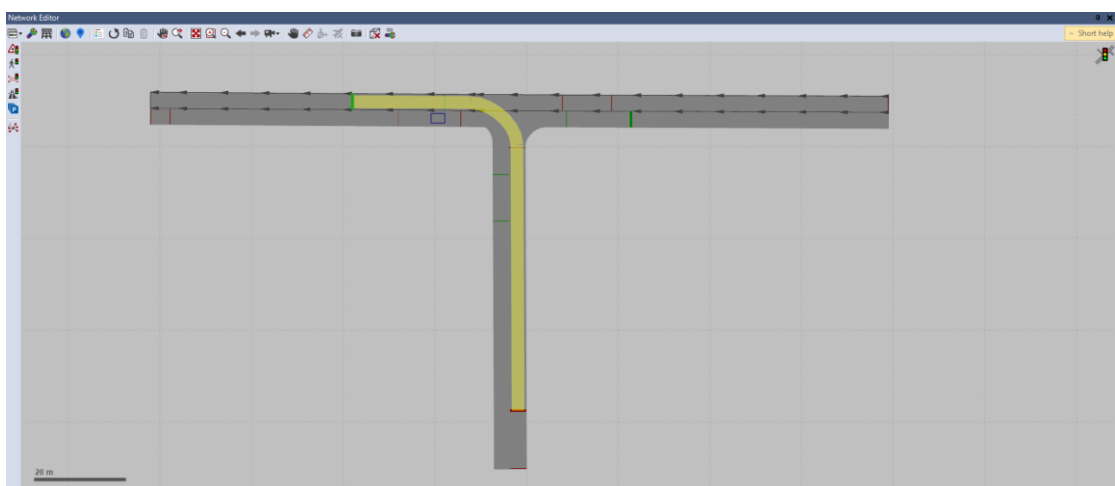
Az útvonalak beállításának menete:

- az útvonalak listáját jelenítsük meg alul a következő módon: a „Network Objects” menüsáv „Vehicle Routes” funkciójára kattintva (bal egérgombbal) a legördülő menüből válasszuk a „Show List/Static” parancsot (lásd 3.15. ábra), mivel alapesetben a statikus útvonalválasztást alkalmazzuk;



3.15. ábra: Statikus útvonalak listájának megnyitása

- bal egérgombbal a kiindulási pontot tartalmazó „Link” kijelölése;
- jobb egérgombbal a kiindulási pont lerakása a szakaszon;
- ezután a célpont lerakása történik bármelyik egérgombbal (lásd 3.16. ábra);



3.16. ábra: Útvonalak beállítása



- egy kiindulási ponthoz természetesen akármennyi célpont választható, így az előző pontbeli műveletet annyiszor végezzük el, ahányszor szükséges;
- fontos, hogy az utolsó végpont beállításakor vagy az „Esc” gombot nyomjuk meg, vagy kattintsunk a háttér üres területére – ezzel jelezve, hogy befejeztük az adott kezdőponthoz tartozó célpontok definiálását;
- az így kialakított útvonalak közötti forgalmi megoszlás pedig utólagosan állítható be az alsó sáv jobb oldalán (lásd 3.17. ábra).

Count	No	Name	Link	Pos	AllVehTypes	VehClasses
1	1		1	4,306	<input checked="" type="checkbox"/>	
2	4		2	60,467	<input checked="" type="checkbox"/>	
3	5		3	12,748	<input checked="" type="checkbox"/>	

Count	VehRoutDec	No	Name	DestLink	DestPos	RelFlow(0-MAX)	VehRoutDec/CombineStaRoutDec
1	5	1	1	104,908		20,000	<input type="checkbox"/>
2	5	2	2	117,041		30,000	<input type="checkbox"/>

Az 1. és 2. útvonal megoszlása:  
40% - 60%

**3.17. ábra: Útvonalak forgalmi megoszlásának beállítása**

Fontos megjegyezni, hogy nagyobb úthálózati modell esetén, érdemes az „Combine static routing decisions” opciót alkalmazni, amely az egymás utáni útválasztásokat kapcsolja össze (3.17. ábra jobb szélső oszlop, lásd részletesebben a Vissim Manualt: [2]). Az összeolvasztott útvonalak esetén sokkal simább forgalomlefolys várható a Vissim szimuláció alatt, pl. elkerülhetők a kanyarodás előtt történő túl késő sávváltások, amelyek miatt indokolatlan és nem realiztikus torlódások alakulhatnak ki.

A „Routes” mellett az útvonalválasztás a dinamikus ráterhelési opcióval is beállítható a „Traffic/Dynamic Assignment” menüpont segítségével.

## 5. Gyakorlás

Az előző leckében elkészített hálózatban dolgozva végezzük el az alábbiakat!

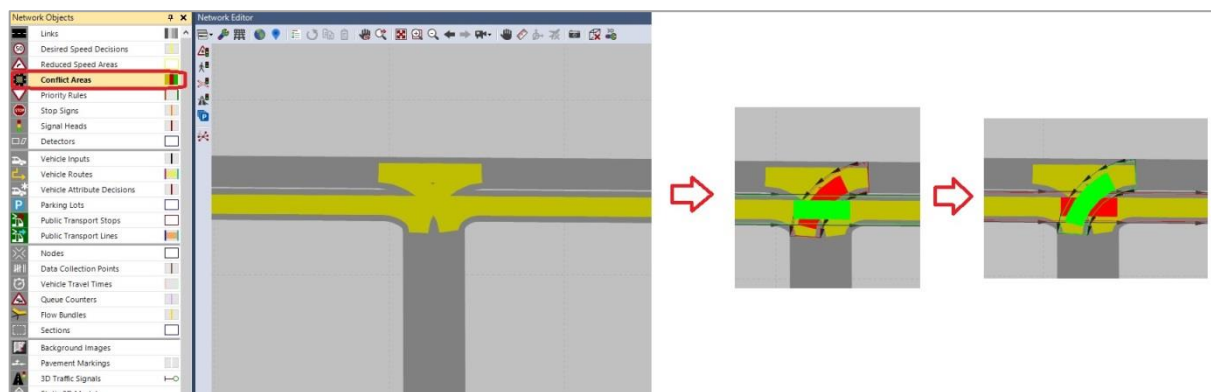
- Generáljunk forgalmakat „Vehicle Input” funkcióval úgy, hogy a csomópont 4 ágából rendre piros teherautók, sárga furgonok, kék motorkerékpárok és zöld személygépkocsik érkezzenek csak!
- Állítsuk be az útvonalakat úgy, hogy minden ágban legyen érvényes a következő fordulási ráta megoszlás: az egyenesen haladók aránya 70%, a balra kanyarodóké 10%, a jobbra kanyarodók aránya pedig 20%!



## 4. lecke: Elsőbbségi szabályok beállítása

A Vissimben három lehetőség is adott az elsőbbségi szabályok beállítására (lásd 3.18. ábra):

- elsőbbségadás kötelező tábla,
- stop-tábla,
- „Conflict Areas” (konfliktus zónák) alkalmazása.



3.18. ábra: Elsőbbségi viszonyok beállítása „Conflict Areas” funkcióval

Az opciók közül az utóbbi használata javasolt a legtöbb esetben az alábbiak szerint:

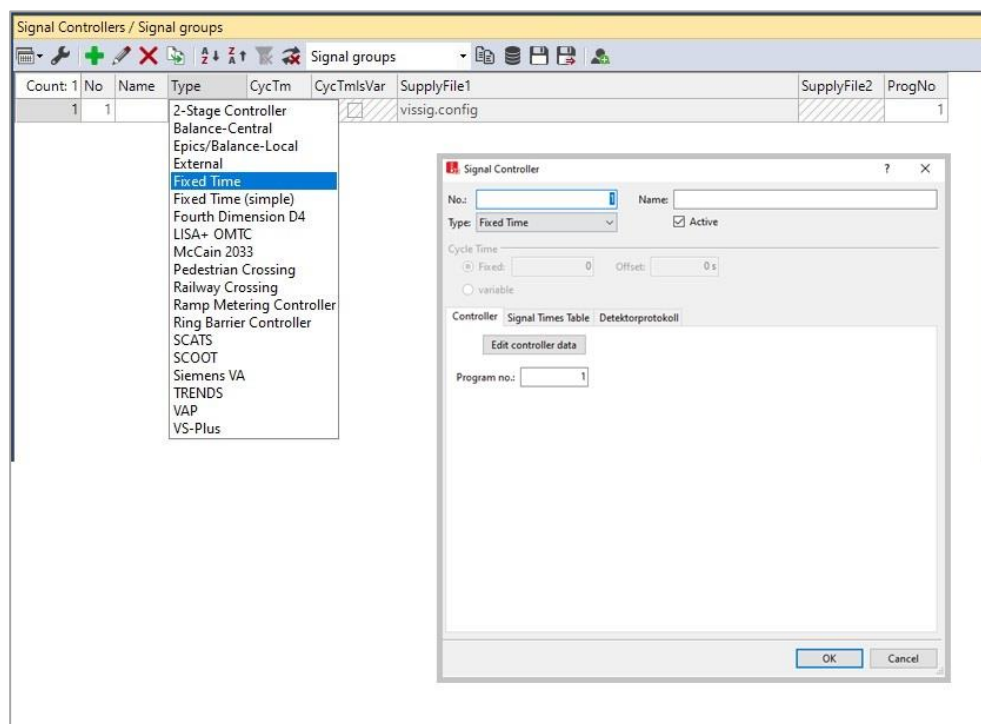
- „Conflict Areas” funkció kiválasztása az eszköztárból,
- bal egérgombbal a konfliktus zónába való kattintás, amely tipikusan az egymást keresztező „Connector” elemekből adódik ki (több egymást átfedő konfliktus területek között a Tab billentyűvel lehet váltani),
- ekkor az alapesetnek megfelelően a keresztező irányok azonos rangúak: mindkettő sárga lesz,
- a jobb egérgombbal belekattintva azonban tetszőlegesen változtathatók az egyes irányok közötti elsőbbségi viszonyok (lásd 3.18. ábra),
- az így kialakított elsőbbségi viszonyok tulajdonságait utólag finomíthatjuk: „Conflict Areas” funkcióban, jobb egérgombbal a szürke területre való kattintással áttekinthetők és szerkeszthetők az elsőbbségi viszonyok láthatóságra és érzékelési időkre vonatkozó paraméterei.

Gyakorlás:

- Az előző leckében elkészített hálózatban dolgozva állítsunk be olyan elsőbbségi szabályokat a csomópontban, amely az egyik keresztező irány elsőbbségét biztosítja!
- Ellenőrzésképpen futtassunk le szimulációt!

## 5. lecke: Jelzőlámpás forgalomirányítás

A Vissim több lehetőséget is nyújt a szabályozó logika, azaz a forgalomirányító berendezések szimulációjára (lásd 3.19. ábra). Rögzített idejű vezérlés esetén az ún. „Fixed time” jelzésvezérléssel egyszerűen definiálható a forgalomtechnika. Saját, forgalomfüggő logika megvalósításhoz többek között a Vap modul (Vehicle Actuated Programming) használható. A Vap modulhoz a Visvap tervező szoftvermodullal (a Vissim szoftver mellé jár licenctől függően) készíthető tetszőleges szabályozás folyamatábrás módon, amihez a hálózatban felvett hurokdetektorokat is fel lehet használni. A Visvapban elkészített logikát egy „vap” kiterjesztésű fájlba exportálva lehet közvetlenül meghívni a Vissimben. További lehetőségként választható az ún. „External” jelzésvezérlés, amely az előzőhöz hasonlóan tetszőleges, saját logika megalkotására ad lehetőséget. Ugyanakkor mindezt már C++ programnyelven kell elkészíteni a Vissim által nyújtott Signal Controller API (szintén licenctől függő) segítségével, ami gyakorlatilag a szabályozó forráskód-szintű programozását jelenti. Mivel a C++, mint objektumorientált nyelv, magas szintű programozási lehetőséget nyújt, bőven túllépi a grafikus Visvap alapú megoldás korlátait. A Signal Controller API-val elkészített jelzési programot „.dll” kiterjesztésű fájlba kell lefordítani, aminek elérési útját a „vap” fájlhoz hasonlóan a felhasználó adja meg a Vissim grafikus felületén keresztül. Ezekon a megoldásokon túl a Vissim további, kereskedelmi célú szabályozási modulok alkalmazását is lehetővé teszi, pl. SCATS, SCOOT, SIEMENS VA, LISA+OMTC.



3.19. ábra: Jelzőlámpás irányítás beállítása

Az alábbiakban a rögzített idejű jelzőlámpás irányítás alkalmazását ismertetjük. Elsőként a csomóponti forgalomirányító berendezés definiálása szükséges (lásd 3.19. ábra):

- a „Signal Control/Signal Controllers” menü megnyitásával a Vissim ablak alján megjelenik a forgalomirányító berendezések listája;

- a „+” ikonra kattintva új forgalomirányító berendezés felvétele;
- a „Type” legördülő menüből a „Fixed time” irányítási típus kiválasztása;
- a táblázat megfelelő sorára duplán kattintva bal egérgombbal, majd a felugró ablakban az „Edit controller data” parancsra kattintva felugrik egy szerkesztő ablak, amelyen belül lehetséges a közbensőidő mátrix, a jelzőcsoportok, ill. a fázisterv felvitele és módosítása;
- a táblázat megfelelő sorára jobb gombbal kattintva lehetséges továbbá a jelzőlámpa programok optimalizálásának elindítása is az „Optimize Signal Control” funkcióval.

A virtuális forgalomirányító berendezés elkészítését követően lehetőség nyílik a jelzőfejek elhelyezésére az útszakaszokon:

- a „Network Objects” menüsávból kiválasztjuk a „Signal Heads” funkciót;
- bal egérgombbal az adott „Link” kijelölése;
- jobb egérgombbal a jelzőfej lerakása, amivel együtt egy „Signal Head” ablak ugrik fel, amelyben a jelzőfejhez tartozó berendezés (SC: „Signal Controller”) és jelzőcsoport („Signal Group”) állítható be.

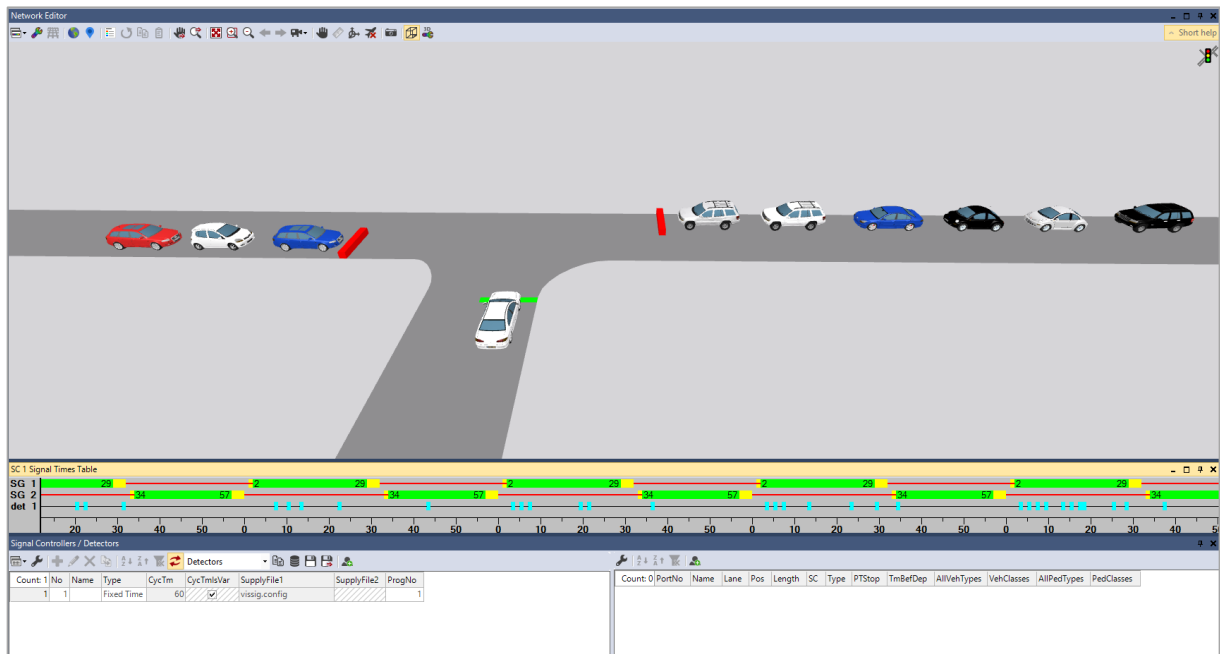
Fontos megjegyezni, hogy a jelzőfejet sose helyezzük a „Connector”-ra, mert az esetenként hibás működést eredményezhet (piros jelzés figyelmen kívül hagyása)!

The image shows a 'Signal Head' configuration window. The fields are as follows:

- No.: 1
- Link - lane: 2 - 1
- SC - Signal group: 1 - 1: Signal group 1
- At: 24,969 m
- Type: Circular
- Rate of compliance: 100,00 %
- Discharge record active
- Is block signal
- Amber speed: 0,00 km/h
- Show label
- Vehicle classes:
  - All vehicle types
  - 10: Car
  - 20: HGV
  - 30: Bus
  - 40: Tram
  - 50: Pedestrian
  - 60: Bike

3.20. ábra: Jelzőfej beállítása

A beállított jelzéstervet futás közben is ellenőrizhetjük az „Evaluation/Window/Signal Times Table” menüpont segítségével (lásd 3.21. ábra).



3.21. ábra: Jelzőlámpa fázistervének futás közbeni megfigyelése

Gyakorlás:

- Az előzőleg elkészített csomóponthoz készítsünk egy egyszerű kétfázisú „teli zöldes” jelzőlámpás forgalomirányítást 60 mp-es ciklus idővel!
- Gyalogos jelzőket is vigyünk fel!
- Ellenőrzésképpen futtassunk le szimulációt, és futtatás közben jelenítsük meg a fázistervet!

## 6. lecke: Közösségi közlekedés modellezése

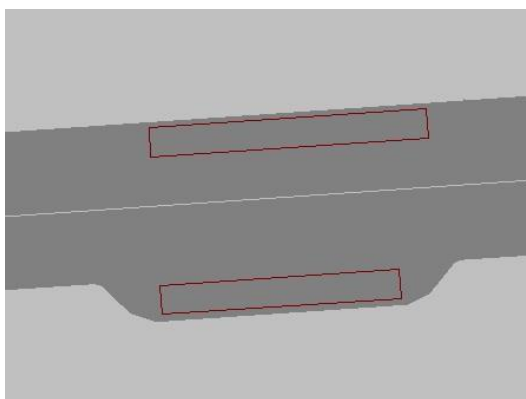
A közösségi közlekedés modelljének elkészítése során a következő sorrendet érdemes betartani:

1. a megállóhelyek definíciója,
2. a viszonylatok definíciója.

Ugyan a Vissim fordított sorrendet is engedélyez, gyakorlati szempontból szerencsésebb ezt elkerülni.

### 1. Megállóhelyek definiálása

A Vissimben kétféle megállóhely létrehozása lehetséges: forgalmi sávban történő megállás, illetve autóbuszöböl. Bármelyiket is szeretnénk alkalmazni, a hálózat felépítése során azt nem kell külön kialakítani, a program automatikusan létrehozza. A kétféle megállóhely közti különbséget mutatja a 3.22. ábra.



3.22. ábra: Megállóhelyek típusai Vissimben

Egy megállóhely létrehozásának lépései a következők:

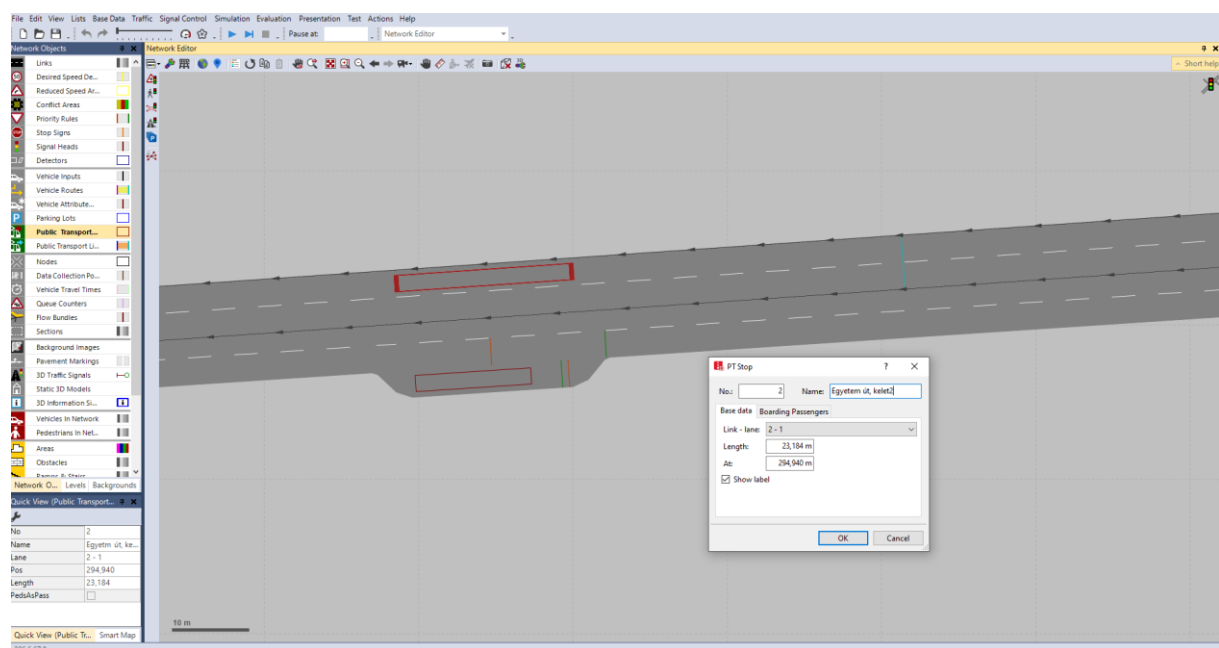
1. először a bal oldali listában kattintsunk a „Public Transport Stop” ikonra;
2. ezt követően bal egérgombbal jelöljük ki a megfelelő szakaszt;
3. majd a kijelölt szakaszon a jobb egérgombot lenyomva tartva húzzunk egy - nagyjából a tervezett megálló hosszúságával egyező - vonalat;
4. a felugró ablakban az alábbi beállítások lehetségesek:
  - a. „No. „: a megállóhely egyedi azonosítószáma
  - b. „Name”: a megállóhely neve, a későbbi könnyebb azonosíthatóság miatt érdemes beszédes nevet adni, ami az irányt is tartalmazza (pl. Blaha Lujza tér, Keleti pu. felé)
  - c. „Length”: a megállóhely hossza
  - d. „Link - lane”: a megállóhely melyik linken sávban helyezkedik el (mindig az 1-es a külső sáv)
  - e. „At”: a „Link”-en belül a megállóhely kezdetét kijelölő hely
  - f. Ha a Show label melletti négyzetet kijelöljük (pipát teszünk), akkor megjelenik a megállóhely egyedi azonosítószáma, ha a rétegen a címkézést aktiváljuk.

A „Boarding Passengers” fülre váltva tudjuk beállítani a megállóban várakozó utasok számát, jobb egérgomb és az „Add” kiválasztásával.

Autóbuszöböl létrehozásához Ctrl + jobb klikkel kattintsunk a megállóra a hálózaton, majd válasszuk az „Add lay-by stop” opciót.

Érdemes megjegyezni, hogy a „Street” típusú megállóhely pozíciója a későbbiekben bármikor változtatható, ehhez bal egérgombbal kattintsunk a „Link”-re, majd ezután bal egérgombbal megragadva tudjuk odébb húzni (a módszer egyébként az egyéb hálózati elemekre - jelzőfej, detektor, stb. - is alkalmazható.) A megállóhely adatait később is tudjuk módosítani, ehhez kattintsunk duplán a bal egérgombbal az adott elemre.

A fent leírtakat mutatja a 3.23. ábra.



3.23. ábra: Megállóhely létrehozása

## 2. Viszonylatok definiálása

Új viszonylatot az alábbi lépésekkel tudunk létrehozni:

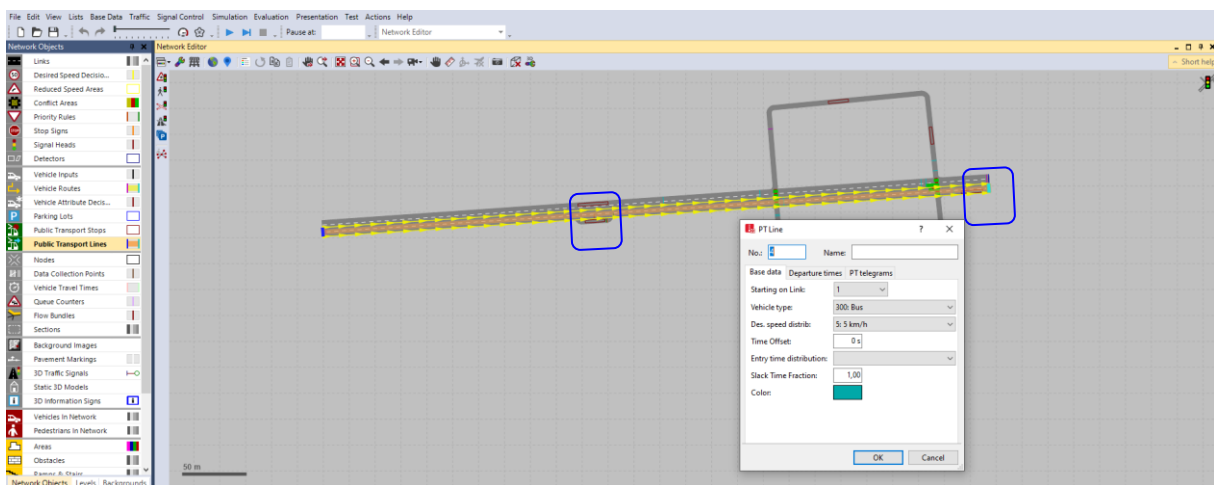
1. kattintsunk a bal oldali listában a „Public Transport Lines” ikonra;
2. kattintsunk a jobb egérgombbal arra a szakaszra, ahonnan a viszonylat indul (ekkor megjelenik a szakasz belépő végén egy sötétkék vonal);
3. kattintsunk bal egérgombbal arra a „Link”-re a célpontra, ahol a viszonylat elhagyja a hálózaton (vagy ahol a végállomása van);
4. a felugró ablakban végezzük el a beállításokat, Base Date fül:
  - a. „No. „: a viszonylat egyedi azonosítószáma a Vissim-hálózatban
  - b. „Name”: a viszonylat neve (érdemes az irányt is beírni, pl. 7E, Blaha Lujza tér felé)
  - c. „Vehicle type”: a viszonylat járműveinek fajtája
  - d. „Des. speed distrib.”: a jármű haladási sebességtartománya zavartalan forgalomáramlás esetén
  - e. „Time Offset”: az első megállóból való indulást megelőzően ennyivel lép be a hálózatba a jármű. Tehát, ha az első megállóból való első indulás ideje 120 s

(ennek beállítását lásd a következő pontokban) és a „Time Offset” értéke 100 s, akkor a jármű a 20. másodpercben lép be a hálózatba, elmegy az első megállóig, majd ott várakozik egészen a 120. másodperc leteltéig. Így, ha a „Time Offset” értéke kisebb, mint a megállóba való eljutás ideje, akkor semmilyen hatást nem fogunk érzékelni a szimuláció futtatása során.

- f. „Slack Time Fraction”: ha a jármű előbb végzett az utascserével, akkor elvileg megengedhető, hogy a megállót az előírt indulási idő előtt elhagyja. Ha a „Slack Time Fraction” 1 értékű, akkor kivárja. Ha egynél kisebb, akkor előbb el fog indulni (minél kisebb, annál előbb). Ez csak akkor működik, ha a megállóhoz „Departure time offset” van rendelve.
  - g. „Color”: a viszonylat járműveinek színe
5. A Departure Times fülön a jobb egérgomb megnyomása után a felugró ablakban: az „Add...” kiválasztása után egyenként adhatunk meg meneteket, míg a „Generate Start Times...” kiválasztásával egyszerre több menetet adhatunk meg. Alább az utóbbi esetet részletezzük, ami alapján az előbbit is kezelni lehet:
- i. „Begin”: az első menet indulási ideje
  - ii. „Rate”: két menet között eltelt indulási idő
  - iii. „End”: az az időpont, ami után már nem indul újabb menet (nem feltétlenül esik egybe az utolsó menet indulásával!)
  - iv. „Course”: opcionális lehetőség, mellyel minden egyes indulási idő azonosítható, bővebb leírásért lásd a használati utasítást.
  - v. „Occupancy”: a hálózatba belépéskor a járművön tartózkodó utasok száma.

**Megjegyzés:** amennyiben az első megállóból történő első (vagy első néhány) indulás időpontja kisebb, mint amennyi időre a járműnek szüksége van elérni az első megállót, valamint befejezni az utascserét, úgy a jármű a 0. szimulációs másodpercben fog belépni a hálózatba, a megállóba érve azonnal elvégzi az utascserét, majd rögtön továbbindul. (Ekkor „természetesen” ez az első néhány menet a menetrendhez képest végig késésben lesz.)

A leírtak a 3.24. ábrán láthatók.



3.24. ábra: Viszonylat létrehozásának lépései

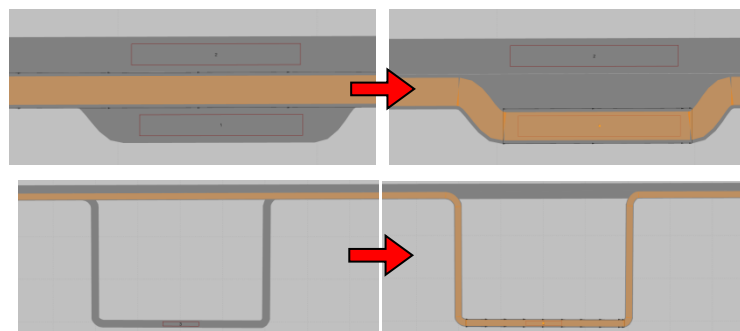
Az ábrán késsel bejelöltük a viszonylat által érintendő megállókat is. Látható, hogy a viszonylat vonalának kijelölése után alapértelmezetten a bal oldali buszöblöt nem érinti, a jobb oldali, forgalmi sávban lévő megállót viszont igen. Így a szimuláció során csak utóbbiban fognak megállni a buszok, előbbiben nem.

A problémát az alábbi módon kezelhetjük:

- kattintsunk a jobb egérgombbal a sárga területre a megálló közelében, ezzel létrehozunk egy közbenső pontot
- e közbenső pontot a bal egérgombbal ragadjuk meg, és húzzuk be a megállóba.

Egy hasonló probléma lehet, amikor a megadott kiinduló- és célpont között a Vissim a valóságostól eltérő útvonalat jelöl ki. A problémát a megállók kérdéséhez teljesen hasonlóan rendezhetjük: létrehozunk a töréspontot, majd a megfelelő „Link”-re húzzuk.

A két bemutatott probléma kezelési módját szemlélteti a 3.25. ábra.



3.25. ábra: Tömegközlekedési útvonalak módosítása

A lecke elején említettük, hogy érdemes először a megállókat létrehozni és csak ezt követően a viszonylatokat. Ennek oka, hogy az útvonalak létrehozásakor a megállókat a viszonylatok automatikusan igénybe veszik, ellenkező esetben viszont magunknak kell minden egyes megállót minden egyes viszonylat számára aktiválni.

Előfordulhat, hogy a hálózatunkon expressz járatok is szerepelnek, melyek nem állnak meg minden megállóban. Az egyes megállóhelyek kihagyását - egyéb más funkciókkal együtt - a „Public transport line stops” ablakban tudjuk beállítani, amelyet az alsó sáv jobb oldali rekeszében a legörülő menüből tudjuk kiválasztani. Kattintsunk bal egérgombbal és az „Add” paranccsal adjunk hozzá egy új megállót. Itt az „Active” jelölőnégyzet deaktiválásával tudjuk elérni, hogy az adott viszonylat járművei a megállóban soha ne álljanak meg. Ezzel szemben a „SkipPoss” jelölőnégyzettel azt engedélyezhetjük, hogy a jármű ne álljon be a megállóba, ha nincs fel- és leszálló utas sem. (Mindennek csak a „PT Stop active” jelölőnégyzet aktiválása mellett van értelme.) Ezt mutatja a 3.26. ábra.

Count	No	Lane	Pos	Length	PedsAsPass	
1	20000	-1	2,971	15,000	<input type="checkbox"/>	
2	2	-1	294,940	23,184	<input type="checkbox"/>	
3	4	-1	494,495	16,613	<input type="checkbox"/>	
4	5	20002	-1	0,100	14,797	<input type="checkbox"/>
5	6	6	-1	56,754	15,526	<input type="checkbox"/>
6	7	5	-1	103,666	13,248	<input type="checkbox"/>

Count	PTLine	PTStop	Active	SkipPoss	ChgDir	Dir	DepOffset	PedsAsPass	DwellTmDef
1	7E_Ujpalota	1: Egyetem út, kelet fe...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0,0	<input type="checkbox"/>	Distribution
2	15E_Bem tér	1: Egyetem út, kelet fe...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0,0	<input type="checkbox"/>	Distribution
3	4	1: Egyetem út, kelet fe...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0,0	<input type="checkbox"/>	Distribution

3.26. ábra: Megálló kihagyásának beállítása



### 3. Tömegközlekedési járművek adatainak beállítása

A tömegközlekedési járművek, illetve azok üzemeltetési módja (például elsőajtós felszállás) igen lényeges hatással vannak az utascsere lebonyolódására, így ezen tényezőket is érdemes lehet figyelembe venni a szimuláció készítésekor. Az alábbi lépéseket követhetjük ilyen esetben:

1. válasszuk ki a „Base Data\Vehicle types” opciót
2. az alul megjelenő ablakban kattintsunk jobb egérgombbal a közösségi közlekedési járműre, és válasszuk az „Edit” opciót, majd a felugró ablak „Special” fülén a „PT Parameters” gombot
3. az újabb felugró ablakban végezzük el a beállításokat
  - a. „Alighting Time”: egy utasra eső fajlagos leszállási idő. Ha egy utas átlagos leszállási ideje 6 s és a járművön 4 ajtó van, amit leszállásra lehet igénybe venni, akkor  $6/4=1,5$  s-ot írjunk.
  - b. „Boarding Time”: egy utasra eső fajlagos felszállási idő, hasonlóan az előzőhöz.
  - c. „Total Dwell Time”:
    - i. „Additive”: ha az ajtókon a fel- és a leszállás is engedélyezett,
    - ii. „Maximum”: ha a le- és felszállás elkülönített ajtókon történik.
  - d. „Clearance Time”: a jármű megállóba való beállításának, valamint az ajtónyitások és zárások összesített ideje.
  - e. „Capacity”: a jármű befogadóképessége.

Fontos kiemelni, hogy ha beállítjuk egy járatnál a „Departure time offset” értékét, akkor az alapján áll elő a mérvadó indulási idő (a meghatározott megállóknban, ahol a beállítás hatása érvényesül).

A leírtakból következik, hogy ha a hálózatunkon többféle tömegközlekedési jármű fut (pl. szóló- és csuklós buszok), akkor azoknak többféle járműkategóriát kell létrehozni, majd azoknál a beállításokat egyenként elvégezni. Fontos azonban megjegyezni, hogy csak olyan szintig pontosítsuk az egyes beállításokat, amelyet a szimulációs feladat megkövetel. Ha a buszokat, villamosokat csak illusztrációs céllal futtatjuk a hálózaton, megfontolandó, hogy biztosan szükség van-e a nagyon aprólékos beállításokra.

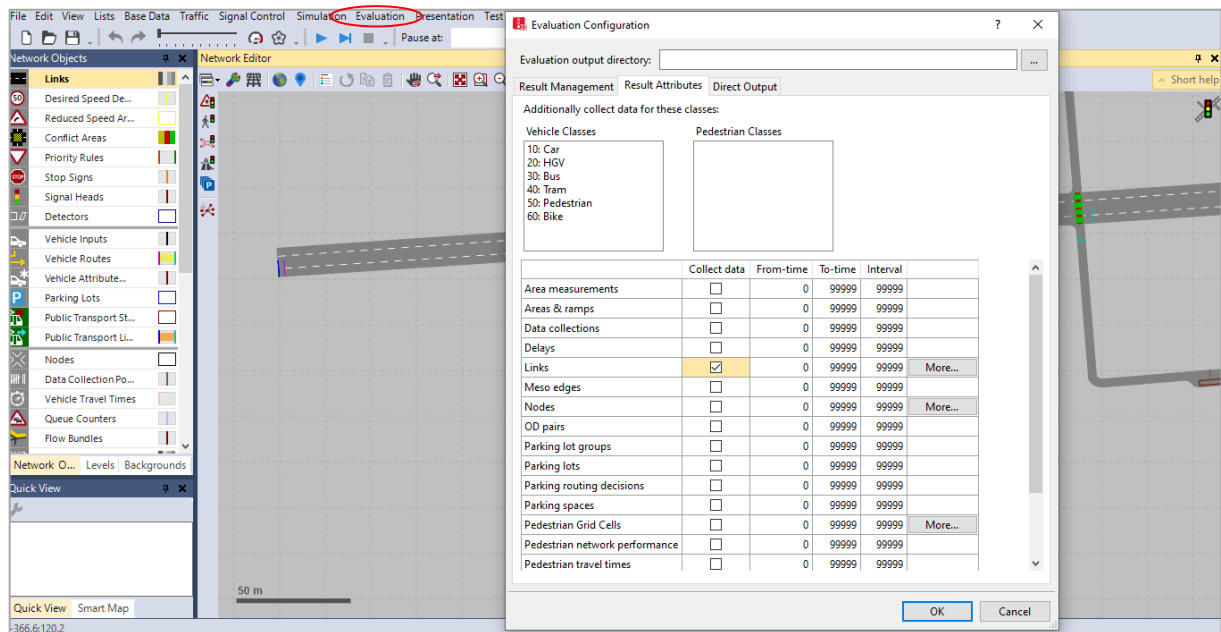
#### Gyakorlás:

- Az előzőleg elkészített hálózatunkba készítsünk tömegközlekedési megállókat és viszonylatokat:
  - legyenek megállók a forgalmi sávokban és autóbuszöbölben is,
  - legyenek hagyományos és expressz járatok is,
  - legyenek szóló és csuklós buszok is, a járművek jellemzőit is állítsuk be!
- Ellenőrzésképpen futtassunk le szimulációt!

## 7. lecke: Vissim futtatási eredmények kiértékelése, videó készítése

### 1. Szimulációs futtatások kiértékelése

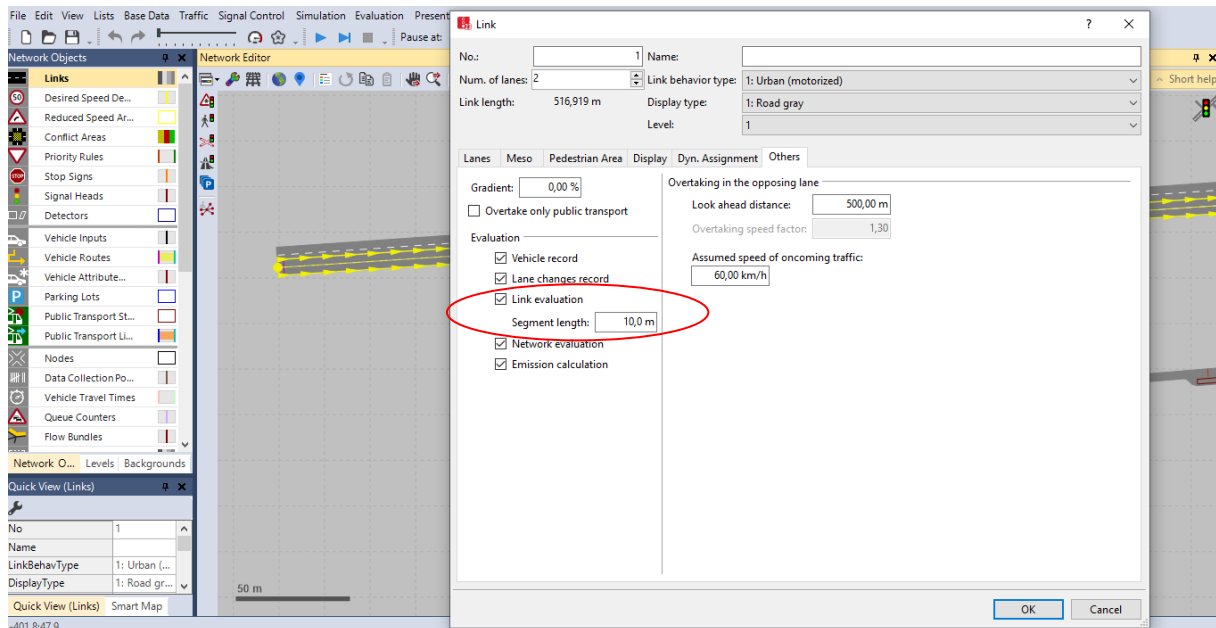
A Vissim szimulációk legfőbb gyakorlati haszna a grafikus megjelenítésen túl a futtatási eredmények kiértékelése. Ehhez az „Evaluation” menüpont elemeit lehet használni, amelyek közül példaként az „Evaluation/Configuration” menü „Result Attributes” fülét emeljük ki (lásd 3.27. ábra). Ebben a menüpontban számos kiértékelési lehetőség adott, de mindegyik esetében az eredmények megjelenítési módja azonos. A linkeknél a „More” gomb megnyitása után a „Per lane segment” opció kiválasztásával sávonkénti kiértékelést kérhetünk. A „From time” és „To time” értékek adják meg, hogy mettől meddig történjen kiértékelés, míg az „Interval” azt adja meg, hogy hány másodpercenként legyenek aggregálva a mért értékek. A kiértékelni kívánt elem mellett a „Collect data” jelölőnégyzetbe be kell tenni a pipát.



3.27. ábra: Kiértékelési lehetőségek az eredmények fájlba történő exportálásával

A fenti opciók közül példaként a „Link Evaluation” funkciót mutatjuk be részletesebben.



Fontos megjegyezni, hogy a „Link Evaluation” funkció csak azon szakaszokra vonatkozóan ad kiértékelést, amelyeknél a „Link evaluation” engedélyezve van. Ezt az útszakaszok létrehozásakor (vagy utólagosan a „Linkre” duplán kattintva) tehetjük meg a 3.28. ábrán bemutatott módon.

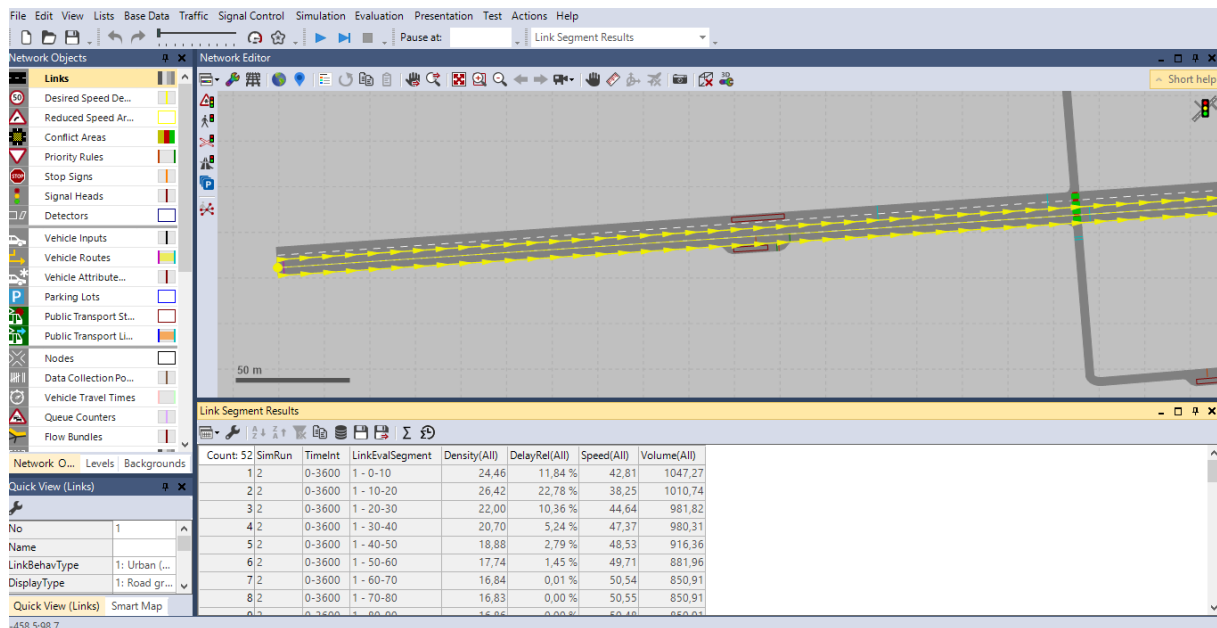


3.28. ábra: „Link Evaluation” engedélyezése az útszakasz beállításai között

Amennyiben az útszakaszt teljes hosszában szeretnénk kiértékelni, ügyeljünk arra, hogy a „Segment Length” megadásakor a szakaszhosszal („Link Length”) pontosan egyező (vagy felfele kerekített) értéket adjunk meg! Ha kisebb értéket adunk meg, az útszakasz ekkora egységekre kerül felbontásra, és a kiértékelés minden egységre elkészül.

A Link kiértékelés menete:

- Evaluation/Result Lists/Link Results menü kiválasztásával az alsó sávot figyeljük, amely a 3.29. ábrán látható.
- A felületen kiválaszthatjuk a „Select attributes” gombot (  ).
- Majd beállítjuk a megjelenítendő paramétereket.
- A futtatás után azt eredmények textfájlba menthetők a mentés (  ) gombbal. A mentés a Vissim projekt fájl mappájába kerül „.att” kiterjesztéssel (pontos vesszővel tagolva).




3.29. ábra: Link kiértékelés megjelenítése a Vissim grafikus felületén

## 2. Szimulációs videók készítése

Gyakran előfordul, hogy a szimulációnak nem csak a számszerű eredményeit kell ismertetnünk, hanem azt valamilyen animáció formájában is be kell mutatni. Ennek alapjait ismertetjük most.

A Vissim-videó (Storyboard) készítésénél két feladatunk van:

- definiálunk néhány fix képkockát, úgynevezett „keyframe”-et,
- meghatározzuk, hogy a képkockákat milyen kameramozgás kösse össze.

A videó elkészítése 3D-s környezetben célszerű. A 2D és 3D mód között a „Network Editor” eszköztárának ikonjával (  ) válthatunk, vagy a **Ctrl + D** billentyűkombináció is használható.

Egy „keyframe” definiálása során nem csak a pillanatnyi nézetet határozzuk meg, hanem azt is, hogy a kamera milyen módon mozogjon a következő fix képkockára, függetlenül attól, hogy azt beállítottuk-e már.

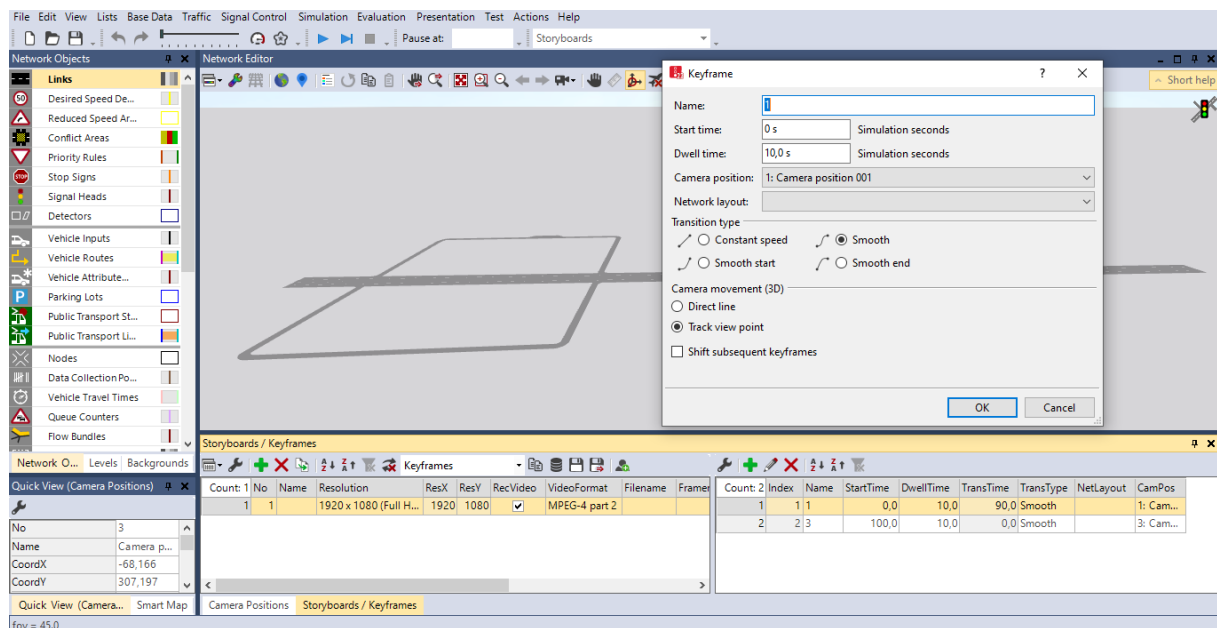
Egy „keyframe” beállításának lépései az alábbiak:

1. állítsuk a 3D-nézetet a kívánt pozícióba; (vagy a „Presentation/Camera Positions” menüpontban adjuk meg az adatokat)
2. kattintsunk a „Presentation\Storyboards” menüpontra;
3. az alsó sáv bal oldali rekeszében vehető fel („+” ikonnal) egy storyboard, ami a keyframe-eket tartalmazza, míg a jobb oldalon a storyboardhoz tartozó keyframe-ek vehetők fel, szintén a „+” ikonnal;
4. az alsó sáv jobb oldali rekeszébe kattintsunk jobb egérgombbal és az „Add” paranccsal („+” ikon) adjunk hozzá egy új „keyframe”-et,
5. ezt követően végezzük el a beállításokat a felugró ablakban (az alábbiakon felül a „villáskulcs” ikonra kattintva egyéb beállítási lehetőségek is előhívhatók):
  - a. „Name”: a „keyframe” neve

- b. „Starting time”: a „keyframe”-hez hányadik másodpercben érjen el a kamera
  - c. „Dwell time”: a „keyframe” pozíciójában hány másodpercig maradjon a kamera
  - d. „Transition Type”: a kamera mozgásának sebességprofilját adja meg az adott és a követő „keyframe” között
    - i. „Constant speed”: két keyframe között a kamera egyenletes sebességgel mozog (0-ról ugrásszerűen felgyorsul a sebességre, majd a célhoz érkezve ugrásszerűen lelassít)
    - ii. „Smooth”: a kamera sebessége szinuszfüggvény 0 és  $\Pi$  között felvett értékei szerint alakul két „keyframe” között. Lassan indul, majd felgyorsul, aztán visszalassul. Ez kelti a legtermészetesebb hatást.
    - iii. „Smooth end”: a mozgás első felében lineáris, míg második felében a sebességről fokozatosan lassul
    - iv. „Smooth start”: a mozgás első fele fokozatosan gyorsuló, második fele egyenletes sebességű. A mozgást inkább kombinálni érdemes a „Smooth end” mozgással, a kettő között lineáris haladással.
  - e. „Shift subsequent Keyframes”: ha több „keyframe” közé szeretnénk egy újabbat beszúrni, akkor a mögöttes „keyframe”-eket eltolja.
6. újabb „keyframe” definiálása előtt mozgassuk a kamerát a kívánt pozícióba, majd ismételjük meg az előző lépéseket.

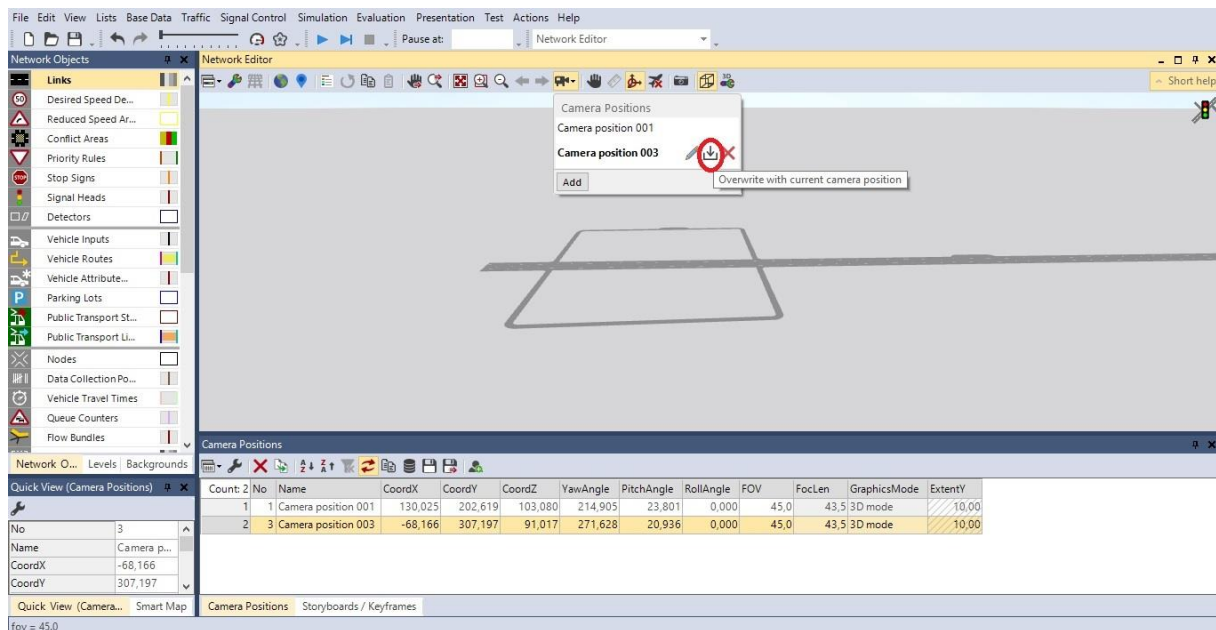
A beállított kameramozgást leellenőrizhetjük a Storyboards/Keyframes alsó, bal oldali egységében a Storyboardra jobb klikkel kattintva, majd a „Preview with video/simulation speed” lehetőségek egyikét választva.

A fent leírtak a 3.30. ábrán láthatók.



3.30. ábra: „Keyframe” beállítása

Egy már definiált „keyframe”-nél a kamera állása utólag is módosítható. Ehhez a Network Editor ablakban állítsuk be a kívánt nézetet, majd válasszuk ki a „Camera Position Selection” ikont. Ezt követően válasszuk ki az „Overwrite with current Camera Position” ikont a módosítani kívánt keyframe mellett. Ezt mutatja be a 3.31. ábra.



3.31. ábra: „Keyframe” módosítása

A videófelvevő módot a „Presentation\Record Videos” menüpontból aktiválhatjuk, akár a szimuláció indítása előtt, vagy akár közben.

## 8. Irodalom a 3. modulhoz

- [1] Olstam, J. J. & Tapani A. Comparison of car-following models, VTI Meddelande 990A, Swedish National Road Transport Research Institute. Project Code 40503 and 40485, 2004
- [2] PTV, VISSIM 2023 User Manual. PTV Planung Transport Verkehr AG, Germany, 2023
- [3] Takács M. Forgalmuszimulációs elemzések egységes vizsgálati módszertanának kidolgozása, Tervezési útmutató, Gazdasági és Közlekedési Minisztérium, 2004
- [4] Wiedemann R. Simulation des Straßenverkehrsflusses Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe, Heft 8, 1974

## 4. modul: Vissim forgalomszimuláció COM programozással

### Cél:

A tananyag célja a Vissim programozásának bemutatása. A Vissim-COM felületen keresztül gyakorlatilag tetszőleges (COM kezelésre képes) programnyelven hatékonyan programozhatjuk a forgalomszimulációkat.

Fontos kiemelni, hogy az alábbi gyakorlatok csak a legfontosabb kezelési műveleteket fedik le. A Vissim-COM programozás teljes és részletes ismertetése a Vissim GUI felületből elérhető internetes súgóban (Online Help) található.

### Követelmények:

Ön akkor sajátította el megfelelően a tananyagot, ha képes

- megfogalmazni a Vissim-COM programozás célját, és alaplogikáját (hierarchikus objektum modell),
- a Vissim-COM alapvető programozására Matlab környezetben.

### Időszükséglet:

A tananyag elsajátításához körülbelül 4-5 órára lesz szüksége.

### Kulcsfogalmak:

- Vissim forgalomszimuláció,
- COM (Component Object Model),
- Vissim-COM programozás,
- Matlab.



# 1. lecke: Vissim forgalomszimuláció lehetőségeinek kiterjesztése COM felület segítségével

## 1. A Vissim-COM programozás célja

Az alábbiakban egy olyan szimulációs keretrendszert mutatunk be, amely hatékonyan alkalmazható közúti forgalomirányítási problémák vizsgálatához, ill. kapcsolódó kutatáshoz. A keretrendszer egyik építőeleme az előző modulban már bemutatott Vissim mikroszkopikus szimulátor. Egy másik építőelem pedig egy tetszőleges programozási nyelv, amely képes COM (Component Object Model) kezelésre [1]. A két építőelem a szimulátorhoz tartozó COM interfész segítségével kapcsolható össze. A COM a Microsoft által kifejlesztett technológia a komponens alapú fejlesztés támogatására, amely a szoftverek közti kommunikációt teszi lehetővé. Bár több platformon is megvalósították, elsősorban a Microsoft Windows operációs rendszerében használják.

A Vissim-COM hierarchikus felépítésű, objektumorientált modellt valósít meg, amiben kód szinten válnak elérhetővé és programozhatóvá a grafikus felületen nyújtott funkciók és paraméterek [2]. A COM felületet lehet script programnyelveken (pl. Matlab Script, Visual Basic Script, Java Script), illetve objektumorientált nyelveken is programozni (pl. Microsoft Visual Studio programcsalád nyelvei, Java, Delphi, C++). A programozás első lépéseként egy COM klienszt kell létrehozni a választott nyelven. Majd céltól függően a forgalomszimulációs folyamat egyes részeit vagy akár egészét el lehet készíteni és automatizálni a COM-on keresztül.

## 2. A Vissim-COM programozás alaplépései

A gyakorlatban egy COM felületen fejlesztett, saját logikát használó Vissim szimulációt a következő lépésekben lehet felépíteni:

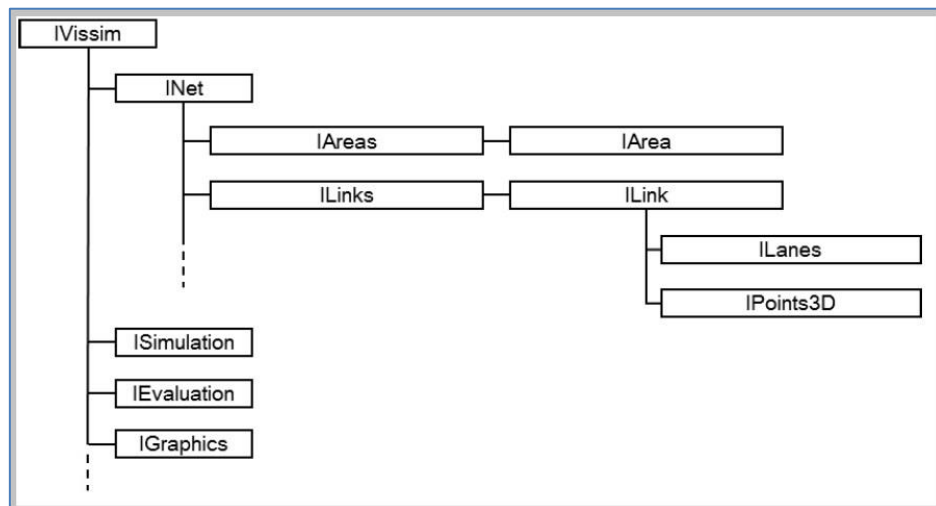
1. az adott közlekedési hálózat felépítése a Vissim grafikus felületén keresztül minden részletével együtt (útszakaszok, jelzőlámpák, detektorok, járművek, stb.), amely egy Vissim projekt fájlban („inpx” kiterjesztéssel) menthető el,
2. COM kliens létrehozása a választott programnyelven,
3. COM interfészen keresztül a szimuláció részletes programozása Vissim-COM specifikus parancsokkal, ehhez néhány példa alább:
  - speciális szimulációs beállítások (pl. többszöri futtatás egymás után változó paraméterekkel),
  - a járművek valamilyen feltételhez kötött viselkedése,
  - futás közbeni kiértékelés (pl. „Link Evaluation”),
  - saját jelzőlámpa irányítási logika leprogramozása,
4. szimuláció futtatása COM programból.

A Vissim-COM programozás alaplogikájának megértéséhez tekintsük a 4.1. ábrát, amely Vissim-COM objektum modelljének egy részletét szemlélteti. Mint látható, a Vissim-COM szigorú objektum hierarchiára épül, amelyben két fajta objektum típus különböztethető meg:

- kollekción (tömb, lista): egyedi objektumok tárolására alkalmas; a kollekción elnevezései (angol nyelv szerinti) többes számban szerepelnek az objektum modellben, pl. „Links”.

- tároló: egyetlen objektum tárolására alkalmas; a tárolók elnevezései egyes számban láthatók az objektum modellben, pl. „Link”.

A Vissim-COM programozásban az ‘I’ jelölés mindig az objektum-interfészre utal, azaz amelyen keresztül az adott objektumok elérhetők. Az objektumok hierarchikus struktúrába vannak rendezve, amelynek a tetején a fő Vissim objektum áll (ebből csak egyetlen egy lehet). A főobjektum hozzáférési felülete az IVissim interfész.



**4.1. ábra: A Vissim-COM objektum modellje [2]**

A teljes Vissim-COM objektum-modell hierarchiájának leírása a Vissimből közvetlenül elérhető online súgóban található (Help\Online Help...\Vissim-COM\Objects). Tekintsük meg például az IVissim interfész „helpjét” a 4.2. ábrán!

The screenshot shows the Vissim COM Interface Documentation window. The search bar contains 'ivissim'. The left sidebar shows a list of 51 topics, with 'Ivissim' selected. The main content area shows the 'Ivissim' interface documentation, including a 'Summary' section, 'Public Methods', 'Public Properties', and 'See Also' sections. The 'Public Properties' section is expanded, showing a table of properties.

Property Name	Description
AttValue	Set attribute value
Evaluation	Returns an Evaluation object instance.
Graphics	Returns a Graphics object instance.
InternalTests	Internal test function.
LicenceInfo	Returns a LicenceInfo object instance.
Net	Returns a Net object instance.
Optima	Returns a Optima object instance.
Presentation	Returns a Presentation object instance.
ScenarioManagement	Returns a ScenarioManagement object instance.
Simulation	Returns a Simulation object instance.

#### 4.2. Vissim objektumok elérése (Vissim online help)

A „Public Properties” címke alatt tíz tulajdonságot láthatunk, amelyek értékeket adnak vissza a Vissim főobjektumról. Ez azt jelenti tehát, hogy pl. az „IEvaluation”, „IGraphics”, „INet”, „IPresentation” és „ISimulation” interfészeken keresztül az „Evaluation”, „Graphics”, „Net”, „Presentation” és „Simulation” objektumok elérhetők.

A különböző alacsonyabb szintű objektumok eléréséhez tehát a 4.1. ábrán látható hierarchiát kell követni. Az objektum-modell logikájának megértéséhez tekintsünk egy példát, amely egy adott útszakasz („Link”) kódszintű elérését ismerteti:

1. Az „IVissim” főobjektum-interfész alatt közvetlenül szerepel (többek között) az „INet” interfész, amely a szimulációhoz kapcsolódó összes hálózati funkcionalitást fogja össze.
2. Az „INet” objektum alatt kollekciónak szerepelnek.  
A „Links” kollekciónak a hálózatban (korábban grafikusán) definiált összes linket (útszakaszt) tartalmazza. A „Links” kollekciónak az „INet” interfész publikus tulajdonsága. A „Link” objektumokat a „Links” tartalmazza, amelyhez az „ILinkCollection” az elérési felület. Ugyanez igaz pl. az „IAreaCollection” kollekciónak és az „Areas” objektumokra.
3. Egy adott „Link” objektum eléréséhez tehát a „Vissim”, „Net”, és „Links” objektumok definiálásán keresztül vezet az út. Az „ILinkContainer” konténer (és konténerek

általában) publikus tulajdonsága az „ItemByKey”, amely lehetővé teszi az „ILink” felülethez és ezáltal adott „Link” objektumhoz való hozzáférést.

4. Az adott „Link” elérését követően pedig már alkalmazhatók a „Link” objektumra vonatkozó Vissim-COM metódusok (pl. „RecalculateSpline”), ill. attribútum lekérések/módosítások (pl. „NAME”).

A fenti példát Visual Basic Script nyelven mutatjuk be. Itt gyakorlatilag a 10-es számú Link elérését láthatjuk, ahol az aposztróf (') jel után a kommentek olvashatók:

Set vis = CreateObject('Vissim.Vissim')	'Vissim főobjektum létrehozása
Set vnet = vis.Net	'Net objektum létrehozása
vis.LoadNet(„D:\Example\test.inpx”)	'Hálózat betöltése
Set links = vnet.Links	'Links kollekció objektum létrehozása
Set link_10 = links.ItemByKey(10)	'A 10-es Link lekérése objektumként

### 3. Matlab alkalmazása Vissim-COM programozáshoz

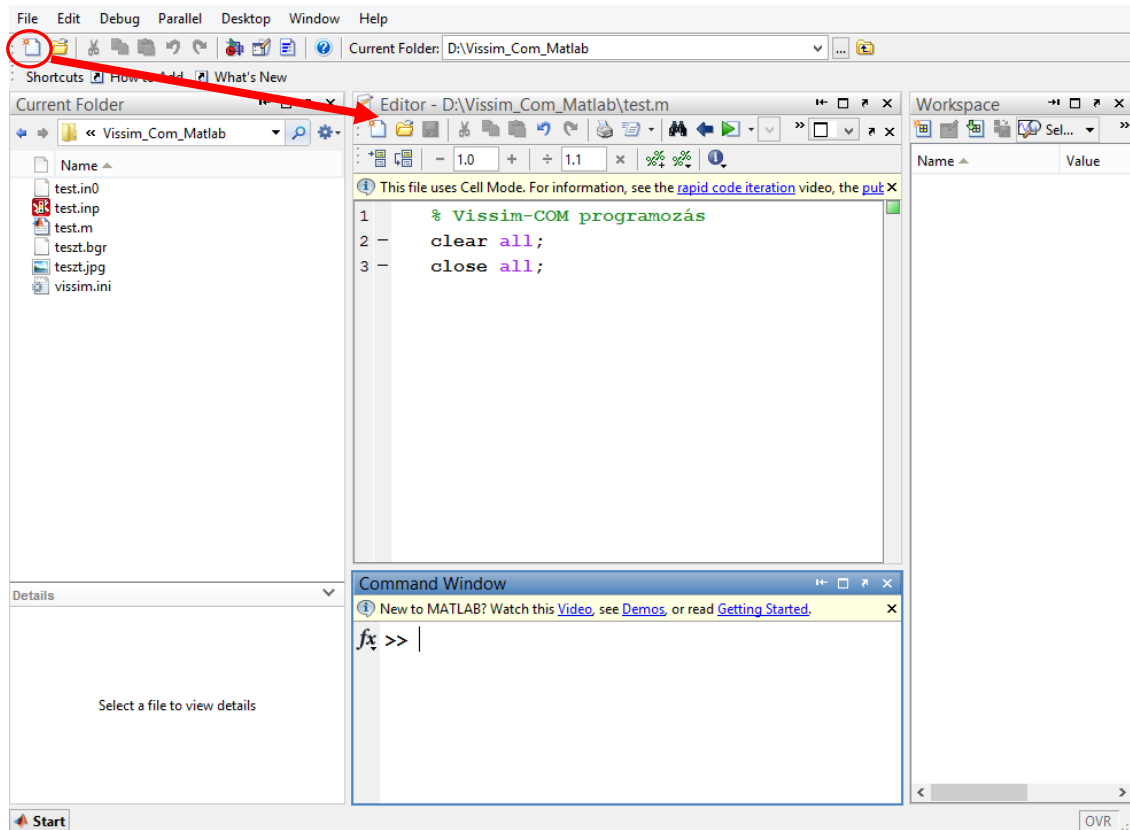
Az alábbi fejezetekben Matlab Script nyelven ismertetjük a Vissim-COM programozást részletesebben. Mindehhez a Vissim-COM Interface Manual (PTV VISSIM, online help) és [2] szoftverdokumentáció nyújt nagy segítséget. Jóllehet e kézikönyv példái Visual Basic nyelven adottak, azok kis programozási ismerettel könnyen átültethetők más programnyelvbe (így Matlab környezetbe is), hiszen a COM programozás azonos elv szerint történik - bármely nyelven is íródik.

A Matlab alkalmazásának egyik fő előnye a Matlab Script nyelv egyszerűsége. Egy másik nagyon fontos szempont pedig, hogy a Matlab (mint gyakorlati célokra gyártott matematikai szoftvercsomag) rengeteg beépített funkcionalitással rendelkezik. Így például optimalizálási feladatokat egyszerű Matlab parancsok segítségével oldhatunk meg, statisztikai függvények hívhatók meg tetszőlegesen, vagy egyszerű mátrixkezelés valósítható meg. A Matlab által nyújtott funkciók alkalmazásával rengeteg idő és energia takarítható meg összehasonlítva azt egy másik programnyelven történő megvalósítással. Amennyiben tehát a Vissim forgalomszimulációkat COM-on keresztül programozzuk, de speciális műveleteket is akarunk végezni (pl. optimalizálás), érdemes a Matlab Scriptet választanunk a Vissim-COM programozás alapnyelveként.

Fontos technikai információ, hogy bármilyen nyelven is szeretnénk Vissim-COM programot készíteni, működéséhez regisztrálnunk kell a Vissimet COM-szerverként az operációs rendszerünkben (ezáltal más alkalmazások is elérhetik a Vissim COM objektumait). Ezt a Vissim szoftver „Help/Register COM server” menüponttal tehetjük meg (a Vissim telepítése után elég egyszerű).

## 2. lecke: Vissim-COM szerver létrehozása Matlabban

Script fájl („.m” kiterjesztésű textfájl) készítése Matlabban a „File/New/Script” menüponttal vagy az eszköztárban található fehér lap ikonnal lehetséges (lásd 4.3. ábra).



4.3. ábra: Matlab Script fájl („.m” kiterjesztéssel) létrehozása

A Matlab Script kódban a % jel után a kommentek írhatók.

Érdeemes az „.m” fájlt két alapvető paranccsal indítanunk:

```
clear all;  
close all;
```

Az első kitörli a Matlab Workspace tartalmát, azaz az aktuálisan használt változókat és értékeiket. A törlés igen hasznos olyan hibák elkerülésére, amiket például az előző futtatásból visszamaradt változók értékei okoznak. A második parancs pedig az aktuálisan megnyitott Matlab ablakok (pl. diagramok) bezárását végzi el egy lépésben.

Új COM (más néven ActiveX) szerver létrehozása az „actxserver” Matlab paranccsal lehetséges:

```
vis=actxserver('VISSIM.vissim')
```

Ha részletesebb információt szeretnénk egy adott Matlab utasításról, használhatjuk a Matlab parancssort („Command Window”) „help” parancsot írva az utasítás elé, pl.:

```
help actxserver
```

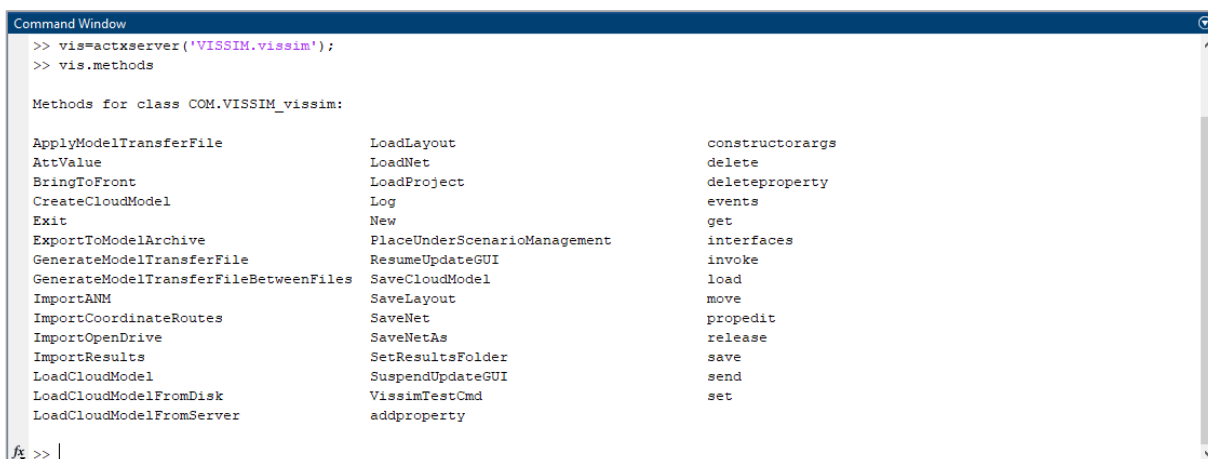
### 3. lecke: Vissim-COM metódusok

A Vissim-COM szerveren létrehozott objektumok metódusai is lekérhetők a parancssorban. Az objektumok listája a Vissim-COM Interface Manual (PTV VISSIM, online help) kézikönyvben megtalálható. Minden egyes objektumra külön-külön kérhetünk le metóduslistát, ha a „Command Window”-ba az objektum nevét, majd ponttal elválasztva a „methods” utasítást írjuk:

```
{ Vissim-COM objektum neve }.methods
```

A fenti módszer csak akkor alkalmazható, ha a kapcsos zárójelbe írt objektum már előzőleg definiálva van.

Példaként tekintsük a 4.4. ábrát, amely a „vis” nevű objektumra (előző fejezetben definiált „Vissim” főobjektum) alkalmazható, és az összes metódus listázását szemlélteti.



```
Command Window
>> vis=actxserver('VISSIM.vissim');
>> vis.methods

Methods for class COM.VISSIM_vissim:

ApplyModelTransferFile      LoadLayout                  constructorargs
AttValue                    LoadNet                     delete
BringToFront               LoadProject                 deleteproperty
CreateCloudModel           Log                         events
Exit                       New                         get
ExportToModelArchive       PlaceUnderScenarioManagement interfaces
GenerateModelTransferFile  ResumeUpdateGUI            invoke
GenerateModelTransferFileBetweenFiles SaveCloudModel             load
ImportANM                  SaveLayout                  move
ImportCoordinateRoutes     SaveNet                     propedit
ImportOpenDrive            SaveNetAs                   release
ImportResults              SetResultsFolder           save
LoadCloudModel             SuspendUpdateGUI           send
LoadCloudModelFromDisk    VissimTestCmd              set
LoadCloudModelFromServer  addproperty
```

4.4. ábra: A „vis” objektum metóduslistájának lekérése Matlabban

A fenti metóduslistából további példaként tekintsük az „invoke” parancsot a 4.5. ábrán.



```
Command Window
>> vis.invoke
AttValue = Variant AttValue(handle, ustring)
New = void New(handle)
LoadNet = void LoadNet(handle, Variant(Optional))
LoadProject = void LoadProject(handle, ustring)
PlaceUnderScenarioManagement = void PlaceUnderScenarioManagement(handle, ustring, ustring)
SaveNet = void SaveNet(handle, Variant(Optional))
SaveNetAs = void SaveNetAs(handle, Variant(Optional))
LoadLayout = void LoadLayout(handle, Variant(Optional))
SaveLayout = void SaveLayout(handle, Variant(Optional))
ImportANM = void ImportANM(handle, Variant(Optional))
ImportCoordinateRoutes = void ImportCoordinateRoutes(handle, ustring, bool, bool, Variant(Optional))
BringToFront = void BringToFront(handle)
Exit = void Exit(handle)
ImportResults = void ImportResults(handle, Variant(Optional))
VissimTestCmd = void VissimTestCmd(handle, int32, ustring, ustring, ustring)
SuspendUpdateGUI = void SuspendUpdateGUI(handle)
ResumeUpdateGUI = void ResumeUpdateGUI(handle, Variant(Optional))
GenerateModelTransferFile = void GenerateModelTransferFile(handle, ustring, ustring, Variant(Optional))
GenerateModelTransferFileBetweenFiles = void GenerateModelTransferFileBetweenFiles(handle, ustring, ustring, Variant(Optional))
ApplyModelTransferFile = void ApplyModelTransferFile(handle, ustring)
SetResultsFolder = void SetResultsFolder(handle, ustring)
Log = void Log(handle, MessagePriority, ustring)
ImportOpenDrive = void ImportOpenDrive(handle, ustring, Variant(Optional))
ExportToModelArchive = void ExportToModelArchive(handle)
LoadCloudModel = void LoadCloudModel(handle, ustring)
LoadCloudModelFromServer = void LoadCloudModelFromServer(handle, ustring)
LoadCloudModelFromDisk = void LoadCloudModelFromDisk(handle, ustring)
CreateCloudModel = void CreateCloudModel(handle, ustring)
SaveCloudModel = void SaveCloudModel(handle, bool)
```

4.5. ábra: A Matlab parancssor válasza egy Vissim-COM objektum „invoke” parancsára

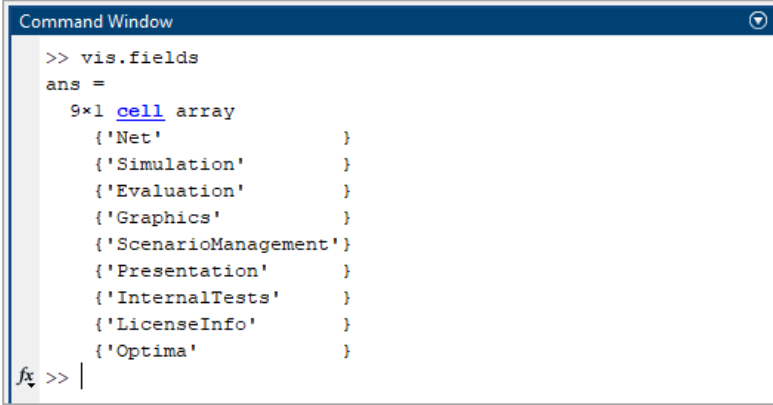
Mint látható, a lista az elérhető metódusokat sorolja fel a visszatérési érték típusával és a hozzájuk tartozó bemeneti függvényparaméterrel. A „variant” egy olyan változó típus,

amely többféle típust is magába foglal. A „void” jelentése pedig, hogy nincs visszatérési értéke a metódusnak.

Természetesen ezzel az eljárással analóg módon a Vissim-COM bármely objektumára vonatkozóan is listázhatók az elérhető metódusok.

Egy másik hasznos Matlab parancs, a „fields”, alkalmas egy objektum tulajdonságainak felfedésére.

{Vissim-COM object name}.fields



```
Command Window
>> vis.fields
ans =
    9x1 cell array
    {'Net'           }
    {'Simulation'   }
    {'Evaluation'   }
    {'Graphics'     }
    {'ScenarioManagement'}
    {'Presentation' }
    {'InternalTests' }
    {'LicenseInfo'  }
    {'Optima'       }
```

#### 4.6. A Matlab parancssor válasza egy Vissim-COM objektum „fields” parancsára

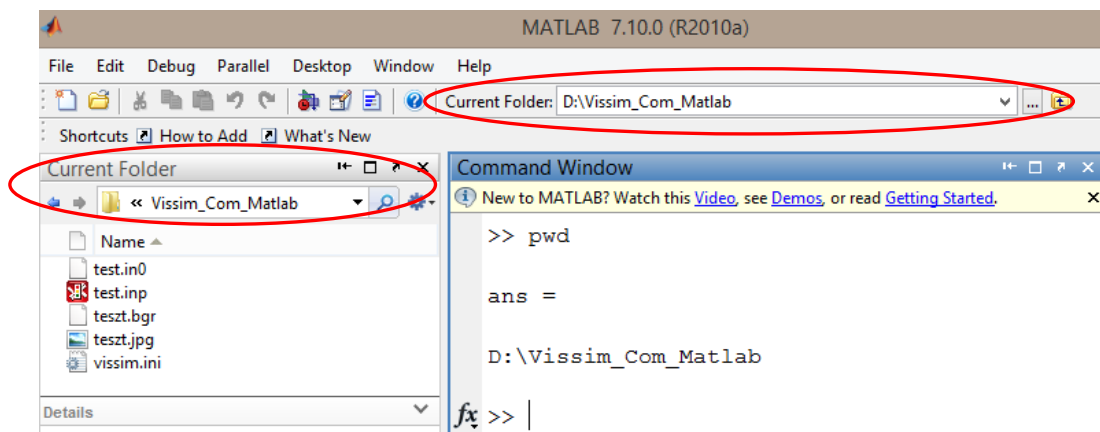
## 4. lecke: Vissim hálózat betöltése

Vissim-COM programozás esetén a szimulációs hálózatot és elemeit először a Vissim grafikus felületén kell létrehozni. Az így kapott „inpx” kiterjesztésű projektfájlt és „layx” kiterjesztésű „Layout” fájlt aztán a COM programból a „LoadNet” és a „LoadLayout” metódusokkal lehet betölteni.

A metódusok alkalmazásakor a Vissim fájlokat meg lehet adni abszolút elérési úttal, ami a fájlok pontos helyét a meghajtó-betűjellel és a tartalmazó mappák neveivel határozza meg, pl.

```
vis.LoadNet('D:\Vissim_Com_Matlab\test.inpx');  
vis.LoadLayout('D:\Vissim_Com_Matlab\test.layx');
```

Emellett lehetőség van relatív elérési út megadására is, ami egy praktikusabb megoldás. Ehhez csupán a Matlab „pwd” parancsát kell meghívunk, amely visszaadja az aktuális mappa abszolút elérési útvonalát (lásd 4.7. ábra).



4.7. ábra: A „pwd” parancs alkalmazása Matlab parancssorban

A hálózat betöltése relatív útvonallal a következőképpen történik:

```
access_path=pwd;  
vis.LoadNet([access_path '\test.inpx']);  
vis.LoadLayout([access_path '\test.layx']);
```

A relatív elérési út alkalmazása rendkívül hasznos, amennyiben pl. a Vissim projektet nem mindig ugyanazon számítógépen akarjuk futtatni. Elég a projektmappát átmásolni az adott gépre, és onnan megnyitni a Matlab Script fájlt. Ilyenkor a kód futtatása előtt nem szükséges a Vissim projekt mappa teljes útvonalának frissítése.

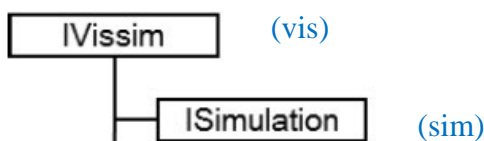


## 5. lecke: Általános szimulációs beállítások Vissim-COM programban

Az alábbiakban az objektum tulajdonságok és attribútumok beállítási menetét ismertetjük. Az érthetőség kedvéért konkrét példaként a szimulációs beállításokat ismertetjük, de természetesen a tulajdonságok és attribútumok szerkesztésének metódusa minden más objektum esetén is azonos.

A szimulációs beállításokhoz a „Simulation” objektumot kell először definiálni, amely a főobjektum alatt foglal helyett a Vissim-COM hierarchia modellben (lásd 4.8. ábra). Ezt a fentebb már definiált, „vis” nevű főobjektumon keresztül tehetjük meg:

```
sim=vis.Simulation;
```



4.8. ábra: A „Simulation” a „Vissim” főobjektum alatt áll a hierarchiában [2] (a zárójelben a mintakódban használt objektumnév olvasható)

### 1. Objektumok tulajdonságai

Minden Vissim-COM objektumnak vannak tulajdonságai (Property). Az objektumok tulajdonságait a „get” metódussal kérhetjük le. „Simulation” objektum esetén pl. ez csupán egy tulajdonságot jelent (lásd 4.9. ábra).



4.9. ábra: A „Simulation” objektum tulajdonságainak lekérése „get” paranccsal

### 2. Objektumok attribútumai

Az objektumoknak ún. attribútumai („Attribute”) is vannak. Az attribútumok eléréséhez az „AttValue” metódust kell használni.

Az attribútumok listája elérhető a Vissim online súgóban (Help\Online Help...).

>> Vissim - COM > [Simulation](#)

## Simulation Attributes

[Overview](#) [Collapse All](#)

☰ **Summary**

Identifier	Short name	Long name
<a href="#">Comment</a>	Comment	Comment
<a href="#">IsRunning</a>	IsRunning	Is running
<a href="#">IsSimBreakAtActive</a>	IsSimBreakAtActive	Is simulation break at active
<a href="#">MesoSectsForMicroSim</a>		
<a href="#">NumCores</a>	NumCores	Number of cores
<a href="#">NumRuns</a>	NumRuns	Number of runs
<a href="#">RandSeed</a>	RandSeed	Random seed
<a href="#">RandSeedIncr</a>	RandSeedIncr	Random seed increment
<a href="#">RetroSync</a>	RetroSync	Retrospective synchronization
<a href="#">SimBreakAt</a>	SimBreakAt	Simulation break at
<a href="#">SimMode</a>	SimMode	Simulation mode
<a href="#">SimPeriod</a>	SimPeriod	Simulation period
<a href="#">SimRes</a>	SimRes	Simulation resolution
<a href="#">SimSec</a>	SimSec	Simulation second
<a href="#">SimSpeed</a>	SimSpeed	Simulation speed
<a href="#">SimTmOfDay</a>	SimTmOfDay	Simulation time (time of day)
<a href="#">StartDate</a>	StartDate	Start date
<a href="#">StartTm</a>	StartTm	Start time
<a href="#">StartTmOfDay</a>	StartTmOfDay	Start time (time of day)
<a href="#">UseAllCores</a>	UseAllCores	Use all cores
<a href="#">UseMaxSimSpeed</a>	UseMaxSimSpeed	Use maximum simulation speed
<a href="#">VolumeIncrDynAssign</a>	VolumeIncrDynAssign	Volume increment (dynamic assignment)

#### 4.10. A „Simulation” objektum attribútum-listája (PTV Online Help)

Vannak csak olvasható (read-only), valamint írható/olvasható (editable) attribútumok. Az azonosító („Identifier”) oszlop elemeire kattintva a „Type” és az „Editable” mezők megmutatják az adott attribútum szerkeszthetőségét, ill. azt, hogy a szimulációs futtatás alatt is módosítható-e. Példaként tekintsük az alábbi ábrát!

SimRes		SimPeriod	
Value type	unsigned int32	Value type	durationInSeconds
Editable	true	Editable	true
Simulation Behavior	EditableDuringSim	Simulation Behavior	ReadOnlyDuringSim
Type	Optional	Type	Optional
Minimum	1	Minimum	1
Maximum	20	Maximum	2678400
Default value	10	Default value	3600

SimSec	
Value type	preciseDurationInSeconds
Editable	false
Simulation Behavior	ReadOnlyDuringSim
Type	Calculated
Minimum	0

#### 4.11. A „Simulation” objektum szerkeszthető és nem szerkeszthető attribútumai (PTV Online Help)

Az „AttValue” metódus alkalmazásának szintaktikája a kiolvasás („get”) és módosítás („set”) esetén az alábbi:

```
sim.get('AttValue', {'attribútum'});
sim.set('AttValue', {'attribútum'}, {beállítandó érték});
```

A 4.11. ábrához kapcsolódóan példaként írhatjuk az alábbi:

```
sim.get('AttValue', 'SimSec');           %(A SimSec csak olvasható, így a
                                           “set” nem megengedett.)
```

Egy másik példa a szimuláció hosszának beállítására vonatkozik:

```
period_time=3600;
sim.set('AttValue', 'SimPeriod', period_time);   %(Szerkeszthető, de csak futáson
                                                    kívül.)
sim.get('AttValue', 'SimPeriod')                 %(A válasz 3600 lesz.)
```

További példaként említjük meg a „Simulation Resolution” tulajdonságot. Ez azt mondja meg, hogy a Vissim járműkövetési modellje hányszor fusson le egy másodperc alatt a szimuláció során:

```
step_time=3;
sim.set('AttValue', 'SimRes', step_time);        %(A ‘SimRes’ attribútum
                                                    beállítása.)
sim.set('AttValue', 'SimRes')                    %( A válasz 3 lesz.)
```

A Vissim attribútumok kiolvasására létezik egy alternatív szintaxis is, amely teljesen egyenértékű a „get” metódus alkalmazásával (lásd alábbi példa):

```
sim.AttValue('SimPeriod') ⇔ sim.get('AttValue', 'SimPeriod')
```

## 6. lecke: Szimuláció futtatása

A szimuláció futtatása háromféle módon történhet Vissimben:

„RunContinuous”: folyamatos futtatás,

„RunSingleStep”: léptető futtatás (minden lépésben a „Simulation Resolution”-nek megfelelő lépésközi idő („step time”) kerül leszimulálásra,

„RunMulti”: sok egymás utáni futtatás.

A fentiek közül a „RunSingleStep” metódust emeljük ki, mivel legkönnyebben ez teszi lehetővé a Vissim szimuláció futtatás közbeni tetszőleges manipulációját (pl. forgalmi igények folyamatos változtatása).

A „RunSingleStep” alkalmazása „for” ciklusban javasolt. Az alábbi példában futtatunk egy szimulációt, amely minden egyes szimulációs lépésben kiírja az addig eltelt szimulációs időt (a „ period\_time „ és „ step\_time” változókat korábban már definiáltuk).

```
for i=0:(period_time* step_time)
    sim.RunSingleStep;
    sim.get('AttValue', 'SimSec');
end
```

„RunSingleStep” használatakor a beállított szimulációs sebesség („Simulation Speed”) nincs hatással a futtatás gyorsaságára. Ekkor a szimuláció léptetve halad a Matlab „for” ciklusának megfelelően, minden lépésközt maximális sebességgel futtatva. Emiatt a fenti módszer alkalmazása esetén a sebesség szabályozását (pl. ha direkt lassítani szeretnénk vizuális megfigyelés okán), a Matlab „pause” parancsával biztosíthatjuk. Az alábbi példában egy 500 ms-os szünetet iktatunk be minden lefuttatott lépés után:

```
for i=0:(period_time*step_time)
    sim.RunSingleStep;
    pause(0.5);
end
```

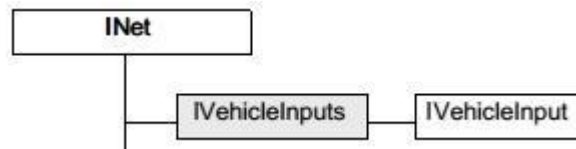
## 7. lecke: Forgalom generálása

A Vissim-COM lehetőséget biztosít a forgalmi igények dinamikus változtatására. Ez igen hasznos amennyiben például:

- sok szimulációt akarunk futtatni különböző forgalmi gerjesztések mellett (akár „MultiRun” metódussal),
- egy napszak forgalmi változását lekövetve szeretnénk (futtatás közben is) változó forgalmi igényt generálni.

Ehhez első körben a „Net” objektumot kell létrehozni, amely a főobjektum alatt foglal helyet a Vissim-COM hierarchia modellben (lásd 4.12. ábra). Ezt a fentebb már definiált, „vis” nevű főobjektumon keresztül tehetjük meg:

```
vnet=vis.Net;
```



4.12. ábra: A „VehicleInput” objektum helye a Vissim-COM hierarchiában [2]

Ezután a „VehicleInputs” kollekción kell létrehozni, amely tartalmazza az összes (a Vissim grafikus felületén definiált) járműbemenetet („VehicleInput”):

```
vehins=vnet.VehicleInputs;
```

A „VehicleInputs” kollekción keresztül bármelyik „VehicleInput” objektum elérhetővé válik alkalmazva a „ItemByKey” metódust, pl.:

```
vehin_1=vehins.ItemByKey(1);
```

Az adott „VehicleInput” objektum tetszőlegesen szerkeszthető az „AttValue” metóduson keresztül (a 4.13. ábrán látható attribútumoknak megfelelően).

>> Vissim - COM > VehicleInput

### VehicleInput Attributes

Overview Collapse All

Summary

Identifier	Short name	Long name
Cont	Cont	Continued
DelayLatent	DelayLatent	Delay (latent)
DemLatent	DemLatent	Demand (latent)
Label	Label	Label
LabPosRel		
LabPosRelPt	LabPosRelPt	Label position relative
LabPosRelWktPoint	LabPosRelWktPoint	Label position relative (WKT)
LabPosRelX	LabPosRelX	Label position relative (x)
LabPosRelY	LabPosRelY	Label position relative (y)
LatWGS84	LatWGS84	Latitude (WGS 84)
Link		
LongWGS84	LongWGS84	Longitude (WGS 84)
Name	Name	Name
No	No	Number
ShowLabel	ShowLabel	Show label
TimeIntVehVols		
VehComp	VehComp	Vehicle composition
VolType	VolType	Volume type
Volume	Volume	Volume
WktLoc	WktLoc	WKT location
WktLocWGS84	WktLocWGS84	WKT location (WGS84)
Zone		

4.13. ábra: A „VehicleInput” objektum attribútum táblázata (PTV VISSIM online help)

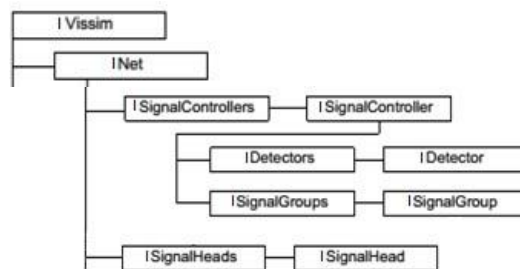
Egy példa a forgalomnagyság attribútum módosítására:  
`vehin_1.set('AttValue', 'Volume(1)', 600);`

ahol a „Volume(1)” 1-es értéke az első időintervallumra vonatkozik.

## 8. lecke: Jelzőlámpás irányítás, detektorok

Egyedi jelzőlámpás irányítás programozása is megvalósítható COM felületen keresztül. Mindamellet el kell mondani, hogy saját logika programozásához könnyebb megvalósítási lehetőséget nyújt a korábban már említett Vissim modul (folyamatábrás programozás) vagy Signal Controller API interfész (C++ nyelven való programozás). Ennek fő oka, hogy ez utóbbiak beépítetten kezelik a közbensőidőket és a szabályozáshoz szükséges hurokdetektorokat.

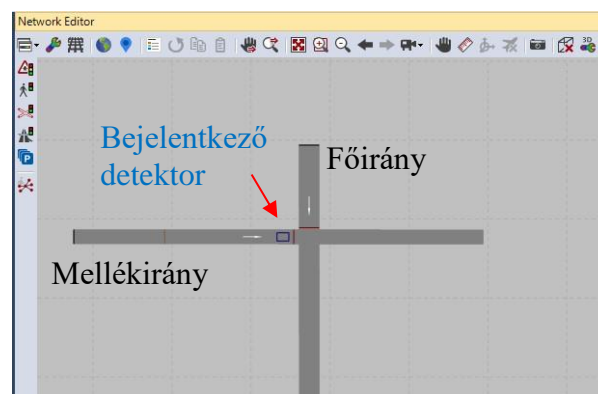
A jelzőlámpás irányítás helye a Vissim-COM objektum modellben a 4.14. ábrán látható.



4.14. ábra: A jelzőlámpás irányításhoz kapcsolódó elemek a Vissim-COM objektum modellben [2]

A Vissim-COM-os jelzőlámpás szabályozást most egy egyszerű példával mutatjuk be.

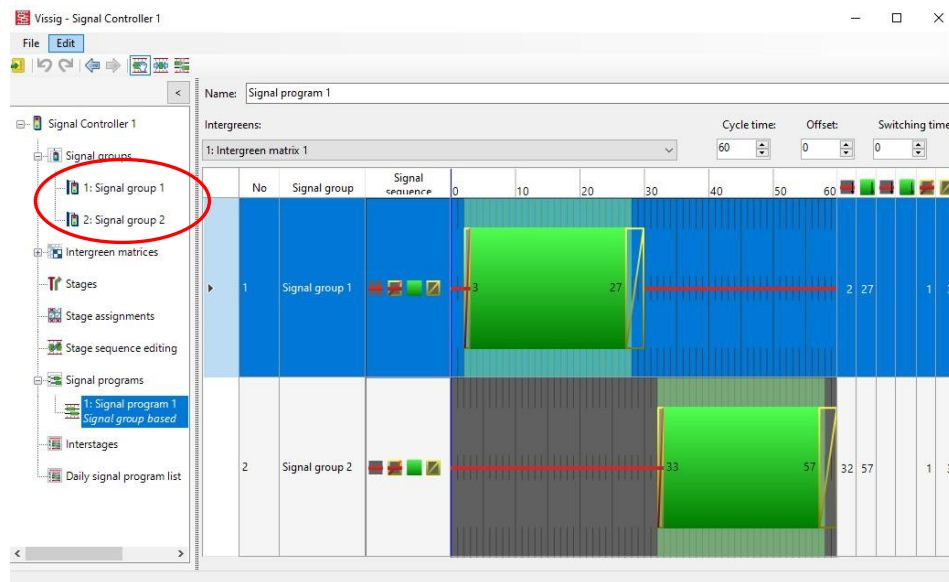
Legyen adott egy egyszerű jelzőlámpás csomópont (lásd 4.15. ábra), ahol egy egyirányú mellékutca keresztez egy szintén egyirányú főútvonalat! A csomópontban két jelzőcsoport működik. Alapból a főútvonali jelzőcsoport állandó zöld jelzéssel üzemel. A mellékutca jelzőcsoportja csak hurokdetektoros bejelentkezés esetén kap zöldfázist. A hurokdetektor foglaltságát a logika 20 másodpercenként ellenőrzi. A betétfázis hossza is 20 másodperc.



4.15. ábra: Egyszerű csomópont Vissimben bejelentkezéses forgalomirányítással

Elsőként a Vissim grafikus felületén kell létrehozni a szükséges elemeket:

- forgalomirányító berendezés definiálása a „Signal Control/Edit Controllers” menüpontban „Fix time” szabályozót választva (ettől még később forgalomfüggővé tesszük a COM programból!),
- jelzőcsoportok („Signal Group”) létrehozása egy adott jelzéstervvel (lásd 4.16. ábra),



4.16. ábra: „Signal Group” létrehozása a Vissim grafikus felületén jelzéstervvel együtt

- jelzőfejek („Signal Head”) elhelyezése a fő- és a mellékirányon,
- bejelentkező detektor elhelyezése a mellékirányon („Detector”).

A fentiek után a Vissim-COM programban definiálni kell a csomóponti forgalomirányító berendezést a „SignalControllers” kollekción keresztül:

```
scs=vnet.SignalControllers;
sc=scs.ItemByKey(1);           %(Signal Controller 1)
```

Létre kell hozni a jelzőcsoportok objektumait a „SignalGroups” kollekción keresztül:

```
sgs=sc.SGs;
sg_1=sgs.ItemByKey(1);       %(Signal Group 1)
sg_2=sgs.ItemByKey(2);       %(Signal Group 2)
```

Továbbá definiálni kell a bejelentkezéshez szükséges hurokdetektor objektumot:

```
dets=sc.Detectors;
det_all=dets.GetAll;         %(Először minden detektort lekérünk.)
det_1=det_all{1};           %(Az első detektor elérése.)
```



A jelzőcsoportok által kiadott jelzéseképeket a „SignalGroup” objektum „State” attribútumán keresztül lehet vezérelni a megfelelő kódok alapján (lásd 4.17. ábra), pl. az 1-es jelzőcsoport piros kijelzésének kódja:

```
sg_1.set('AttValue', 'State', 1);
```

>> Vissim - COM > SignalizationState Enumeration

### SignalizationState Enumeration

▼ Collapse All

Members

Member	Value	Summary
SignalizationStateAlternatingRedGreen	10	
SignalizationStateAmber	4	
SignalizationStateFlashingAmber	7	
SignalizationStateFlashingGreen	9	
SignalizationStateFlashingRed	8	
SignalizationStateGreen	3	
SignalizationStateGreenAmber	11	
SignalizationStateOff	5	
SignalizationStateRed	1	
SignalizationStateRedAmber	2	
SignalizationStateUndefined	6	

Copyright © 2023 PTV Planung Transport Verkehr GmbH

PTV VISSIM

4.17. ábra: A „SignalGroup” objektum „State” attribútumához kapcsolódó jelzések kódok (PTV VISSIM, online help)

A hurokdetektor állapotának lekérése pedig szintén „AttValue” metóduson keresztül történhet többféle attribútum szerint, pl.

```
det_1.get('AttValue', 'Detection');
det_1.get('AttValue', 'Impulse');
det_1.get('AttValue', 'Occup');           %(Occupancy)
det_1.get('AttValue', 'Presence');
```

A forgalomfüggő logika teljes megvalósítása a fentiek mellett a Matlab „rem” parancsával történik (amely két szám osztás utáni maradékát adja vissza):

```
for i=0:( period_time*step_time)
    sim.RunSingleStep;
    if rem(i/step_time,20)==0           % ellenőrzés minden 20. másodpercben
```

```

demand=det_1.get('AttValue', 'Presence'); % detektorfoglaltság ellenőrzése: 0/1
if demand==1 % igény vezérelt betétprogram
    sg_1.set('AttValue','SigState',1); % főirány piros (1)
    sg_2.set('AttValue','SigState',3); % mellékirány zöld (3)
else % nincs igény -> főirány zöld
    sg_1.set('AttValue', 'SigState', 3);
    sg_2.set('AttValue', 'SigState', 1);
end
end
end

```

A fenti példában az egyszerűség kedvéért elhanyagoltuk a közbenső időt a két fázis között, ill. átmeneti jelzéseképeket sem alkalmaztunk. Ezek megvalósítása további programozást igényel.

## 9. lecke: Futtatás közbeni kiértékelés

A Vissim-COM programozás egyik nagy előnye a futás közbeni kiértékelés lehetősége.

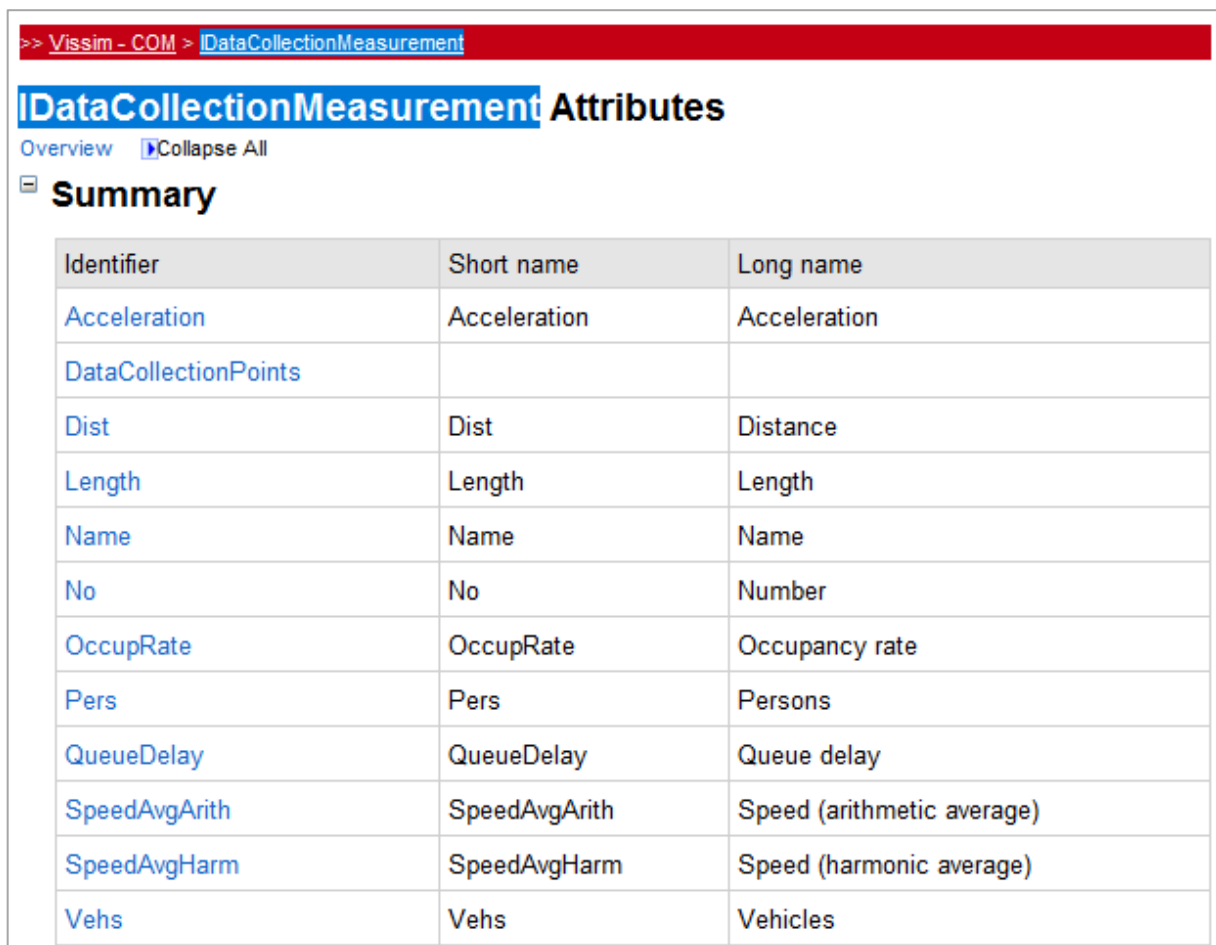
A számos kiértékelési lehetőség közül most példaként a „DataCollection” objektumon keresztüli értékelést mutatjuk be.

Az adatgyűjtő pontok („Data Collection Points”) hatékonyan alkalmazhatók a Vissim grafikus felületén. Tetszőlegesen helyezhetők el az úthálózatban, ill. számos paraméter mérésére alkalmasak az adott keresztmetszetben (pl. gyorsulás, járműszám, foglaltság).

A „Data Collection Measurements” kollekción keresztül és a „ItemByKey” metódus alkalmazásával érhetjük el az adott mérőpontot:

```
datapoints=vnet.DataCollectionMeasurements;  
datapoint1=datapoints.ItemByKey(1);
```

A mérőpontok Vissim-COM felületen történő (akár futás közbeni) kiértékeléséhez a „GetResult” metódus használható a megfelelő paraméterekkel együtt (lásd 4.18. ábra).



The screenshot shows the Vissim-COM interface for the DataCollectionMeasurement class. The title bar indicates the current object is DataCollectionMeasurement. Below the title, there is a section for 'Attributes' with an 'Overview' tab and a 'Collapse All' button. A 'Summary' section is expanded, showing a table of attributes with three columns: Identifier, Short name, and Long name.

Identifier	Short name	Long name
Acceleration	Acceleration	Acceleration
DataCollectionPoints		
Dist	Dist	Distance
Length	Length	Length
Name	Name	Name
No	No	Number
OccupRate	OccupRate	Occupancy rate
Pers	Pers	Persons
QueueDelay	QueueDelay	Queue delay
SpeedAvgArith	SpeedAvgArith	Speed (arithmetic average)
SpeedAvgHarm	SpeedAvgHarm	Speed (harmonic average)
Vehs	Vehs	Vehicles

4.18. ábra: Az „IDataCollectionMeasurement” interfészen keresztül lekérdezhető attribútumok (PTV VISSIM, online help)

Egy ilyen lekérdezésre példa az alábbi kód:

```
datapoint1.get('AttValue', 'Speed(19, 2, All)');
```

ahol a zárójelben található elemek sorrendben:

- lekérdezendő paraméter,
- a szimulációs futtatás sorszáma (a példában 19),
- az időintervallum száma (a példában 2),
- járműosztály („Vehicle Class”), ahol az „All” érték az összes osztályt magába foglalja.

A méréseket tehát több futtatásra, időintervallumra és különböző járműosztályokra vonatkozóan is el tudjuk végezni. A COM segítségével lehetőség van az összes mérési eredmény elérésére az alábbi ábrán látható alattribútumok („sub attributes”) megadásával. Azaz a fenti kódot tekintve például a megadott konstansok (19 és 2) helyett periodikus alattribútumok is megadhatók.

Sub attribute #	Possible Input	Description	
1	SimulationRun	1, 2, 3, ... or Current	number according to the attribute <i>No</i> of simulation runs (see <i>Simulation Runs</i> list) or <i>Current</i> for the simulation run currently running
		Avg, StdDev, Min, Max	aggregated value of all simulations runs
2	TimeInterval	1, 2, 3 ... or Last	index of one specific time interval (the index for time intervals always starts at 1 which refers to the first time interval) or <i>Last</i> to reference to the last completed time interval.
		Avg, StdDev, Min, Max	aggregated value of all time intervals of one simulation
		Total	sum over all time intervals of one simulation
3	VehicleClass	10, 20, ... or All	one or more vehicle class numbers (according to the attribute <i>No</i> of <i>Vehicle Classes</i> ) or <i>All</i> . Data is shown only for the vehicle classes defined here.

#### 4.19. A „DataCollectionMeasurements” mérésre vonatkozó alattribútumok (PTV, 2023)

A 4.19. ábrán ismertetett első két alattribútum esetén általában a „Current” és a „Last” alattribútumokat érdemes használni, pl.

```
datapoint1.get('AttValue', 'Speed(Current, Last, All)');
```

Az adatgyűjtő pontokkal történő mérés kritikus feltétele Vissim-COM esetén is, hogy a Vissim grafikus felületén előzőleg a „Data Collections\Collect data” funkció be legyen jelölve (és megfelelően konfigurálva) az „Evaluation\Configuration...\Result Attributes” menüben. Az időintervallum beállítás is igen fontos „Data Collection” alkalmazásakor. A mért paraméterre vonatkozóan minden mérési időintervallumban egy átlagos értéket kapunk. Azaz minél kisebb intervallumot állítunk be, annál pontosabb mérést kapunk egy időintervallumra vonatkozóan. A beállítást szemlélteti a 4.20. ábra.

The screenshot displays the VISSIM software interface. The main window shows a network editor with a cross-shaped road layout. The 'Evaluation Configuration' dialog box is open, showing the 'Direct Output' tab. The 'Data collections' row is highlighted, with a red circle around the 'Collect data' checkbox and another red circle around the 'Interval' value of 20. The 'Data Collection Results' table is visible at the bottom of the main window.

Count	SimRun	TimeInt	DataCollectionMeasurement	Acceleration(All)	Dist(All)
1	13	0-20	1		
2	13	20-40	1		
3	13	40-60	1		
4	13	60-80	1	0,69	26,97
5	13	80-100	1		

4.20. Kiértékelési beállítások Data Collection Point esetén (Evaluation configuration) és eredmények (alul)

## 10. lecke: Minta Vissim-COM kód

Az átláthatóság kedvéért a fentiekben ismertetett mintapéldákból egy teljes Matlab Script kód kerül alább bemutatásra.

```
%% Vissim-COM programming - example code %%
clear all;
close all;
clc; % Clears the command window
%% Create Vissim-COM server
vis=actxserver('VISSIM.vissim');
%% Loading the traffic network
access_path=pwd;
vis.LoadNet([access_path '\test.inpx']);
vis.LoadLayout([access_path '\test.layx']);
%% Simulation settings
sim=vis.Simulation;
period_time=3600;
sim.set('AttValue', 'SimPeriod', period_time);
step_time=3;
sim.set('AttValue', 'SimRes', step_time);
%% Define the network object
vnet=vis.Net;
%% Setting the traffic demands of the network
vehins=vnet.VehicleInputs;
vehin_1=vehins.ItemByKey(1);
vehin_1.set('AttValue', 'Volume(1)', 1500); % main road
vehin_2=vehins.ItemByKey(2);
vehin_2.set('AttValue', 'Volume(1)', 100); % side street
%% The objects of the traffic signal control
scs=vnet.SignalControllers;
sc=scs.ItemByKey(1);
sgs=sc.SGs; % SGs=SignalGroups
sg_1=sgs.ItemByKey(1);
sg_2=sgs.ItemByKey(2);
dets=sc.Detectors;
det_all=dets.GetAll;
det_1=det_all{1};
%% Access to DataCollectionPoint object
datapoints=vnet.DataCollectionMeasurements;
datapoint1=datapoints.ItemByKey(1);
%% Access to Link object
links=vnet.Links;
link_1=links.ItemByKey(1);
%% Running the simulation
verify=20; % verifying at every 20 seconds
%Evaluation\Configuration...\Interval in the Vissim GUI
for i=0:(period_time*step_time)
    sim.RunSingleStep;
    if rem(i/step_time, verify)==0 % verifying at every 20 seconds
        demand=det_1.get('AttValue', 'Presence'); %get detector occupancy:0/1
        if demand==1 % demand -> demand-actuated stage
            sg_1.set('AttValue', 'SigState', 1); % main road red (1)
            sg_2.set('AttValue', 'SigState', 3); % side street green (3)
        else % no demand on loop -> main road's signal is green
            sg_1.set('AttValue', 'SigState', 3);
            sg_2.set('AttValue', 'SigState', 1);
        end
        % Query the avg. speed and vehicle number at the end of each eval. interval:
        datapoint1.get('AttValue', 'Vehs(Current, Last, All)')
        datapoint1.get('AttValue', 'Speed(Current, Last, All)')
    end
end
%% Delete Vissim-COM server (also closes the Vissim GUI)
vis.release;
disp('The end')
```

## **11. Irodalom a 4. modulhoz**

[1] Box D. Essential COM, Addison-Wesley, ISBN 0-201-63446-5, 1998

[2] PTV Vissim 2023, Introduction to the COM API, PTV Planung Transport Verkehr AG, Germany, 2023

## 5. modul: VISUM makroszkopikus forgalomszimulációs gyakorlatok

### Cél:

A tananyag célja a Visum makroszkopikus forgalomszimulációs szoftver bemutatása és alapvető kezelésének ismertetése gyakorlati példákon keresztül.

A jegyzet megírásához a szoftver 2023-as számú verzióját használtuk. A program teljes és részletes ismertetése a Visum User Manual című és a Visum QuickStar dokumentumokban található meg.

### Követelmények:

Ön akkor sajátította el megfelelően a tananyagot, ha képes

- a klasszikus négylépcsős makroszkopikus forgalommodellezés elméletének megfogalmazására,
- a Visum forgalomszimulátor alapvető kezelésére a négylépcsős forgalommodellezésnek megfelelően.

### Időszükséglet:

A tananyag elsajátításához körülbelül 4-6 órára lesz szüksége.

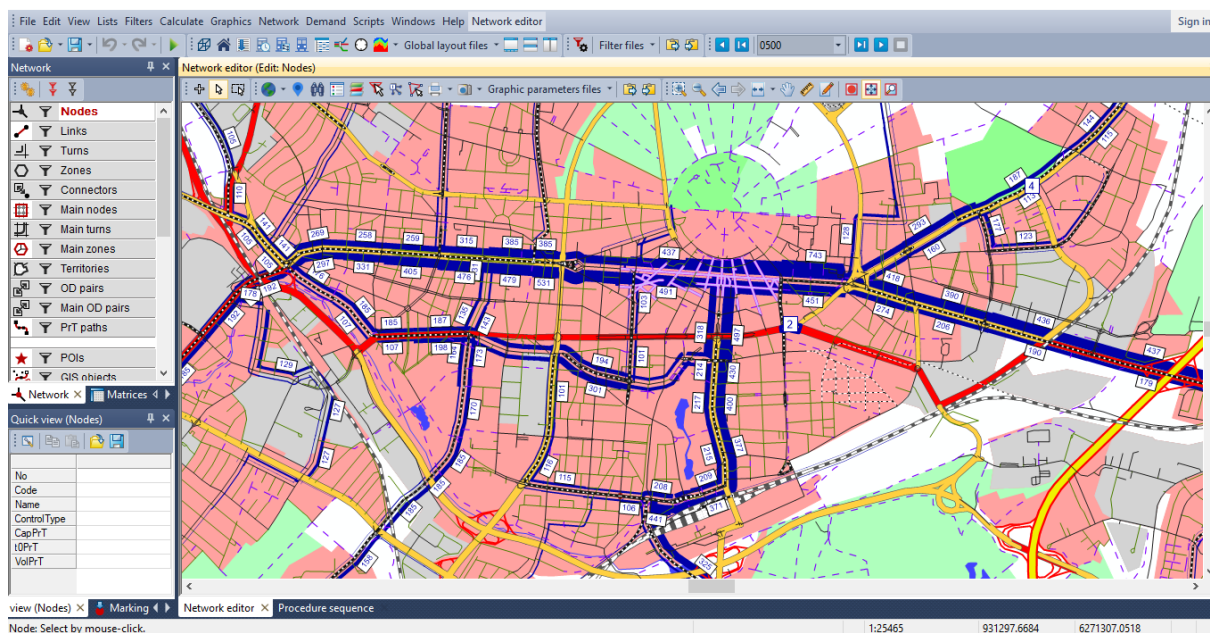
### Kulcsfogalmak:

- makroszkopikus közúti forgalomszimuláció
- négylépcsős forgalmi modell
- Visum szimulációs gyakorlatok



## 1. lecke: A Visum bemutatása

A Visum közúti forgalomszimulátor szintén a karlsruhe-i székhelyű PTV Group (www.ptvgroup.com) terméke, ahogy a Vissim is. A makroszkopikus szintű forgalmi modellezés célja a közlekedési hálózaton lebonyolódó forgalom hálózati szintű leképezése (szakaszok forgalmi terhelésének megállapítása); nagyobb terület (pl. város stb.) modellezése hosszabb időtávra, a közlekedés egészére jellemző folyamatok vizsgálata. Vizsgálható a forgalmi áramlatok, a beavatkozások hatékonysága, a levegő és zajszennyezettség, a tömegközlekedési menetrend és hálózat optimalizálás és a közlekedési módok integrált kezelése.



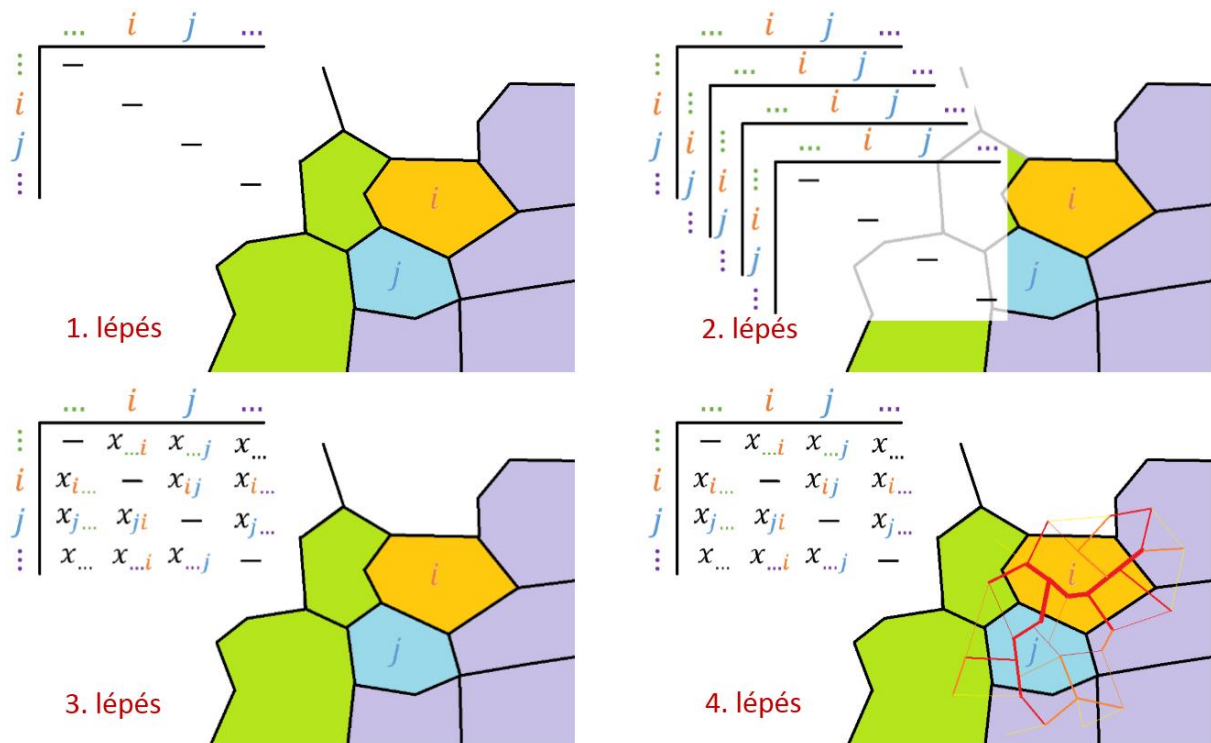
5.1. ábra: Visum pillanatkép (forrás: PTV mintafáj)

Visumban a közlekedés modellezése elsősorban a közlekedéstervezéssel kapcsolatos intézkedések, építkezések forgalmi és egyéb hatásainak szimulációjával (pl. útlezárás hatása, forgalom átrendeződése) történik. A programmal lehetséges az útdíjak hatásainak prognosztizálása, valamint az egyes közlekedési rendszerek (közösségi - egyéni, pl.: személygépjármű, tehergépjármű stb.) forgalmának elkülönült vizsgálata. Az aktuális számlálási adatok összehasonlíthatók a szimuláció során nyert eredményekkel. A szimuláció alatt a célforgalmi mátrix adatait ráterheljük az úthálózatra és az egyes útszakaszokra megkapjuk a forgalom terhelési adatokat.

A Visumban a tömegközlekedés modellezése során a viszonylathálózat és menetrend elemezhető és tervezhető. Tömegközlekedés specifikus jellemzők szemléltetése (eladott jegyek, felszálló/leszálló utasok száma zónánként, megállóként stb.), vagy az utasszámok és más jellemzők kiértékelése lehetséges közlekedési rendszerenként, viszonylatonként, üzemeltetőnként, megállóként, útszakaszonként. A grafikus szemléltetési lehetőség a jelenlegi és tervezett változat, illetve több tervvariáns eltéréseinek bemutatására is alkalmas. Lehet vele jármű- és járművezető szükségletet számítani, illetve költség-haszon elemzés alapjául szolgáló kimeneti adatokat előállítani. Üzemeltetői mutatók kinyerhetők a viszonylati költségek számításához.

A Visum a hagyományos négylépcsős modellre épül. A klasszikus négylépcsés közlekedési modellezés az alábbi elemekből áll (lásd 5.2. ábra):

1. lépés a forgalomkeltés (trip generation): A forgalomkeltés célja az adott területről kiinduló és az oda beérkező forgalmak meghatározása, azaz annak definiálása, hogy hány utazás indul ki és érkezik be az egyes körzetekbe. Ez a célforgalmi mátrix (OD, azaz Origin-Destination matrix) peremeinek, azaz a sor- és oszlopösszegek meghatározását jelenti.
2. lépés a forgalomszétosztás (traffic distribution): A forgalomszétosztás során az előző pontban meghatározott kiinduló és érkező forgalmakat osztjuk szét, azaz a kiinduló utazásokhoz célpontokat definiálunk. Ez gyakorlatilag az OD-mátrix belsejének kitöltése a honnan-hova utazások számaival.
3. lépés a forgalommegosztás (modal split): A forgalommegosztás célja az utazók közlekedési eszközválasztásának modellezése. Ez általában a különböző közlekedési módok közötti megosztást jelenti.
4. lépés a forgalmi ráterhelés (traffic assignment): Ebben a fázisban az előzőekben meghatározott forgalmi igények közlekedési hálózatra való ráterhelése történik - másképp fogalmazva az átlag forgalmak úthálózaton történő szétterítése. Ez gyakorlatilag egy optimalizálási feladat megoldását jelenti különböző korlátozó feltételek mellett, amelynek eredményeként a közlekedési hálózat elmeinek átlagos forgalomnagyságát és utazási idejét kapjuk meg. Az így nyert adatokból további forgalomtechnikai paramétereket lehet kiszámítani vagy megbecsülni.



5.2. ábra: A négylépcsős forgalommodell

A négy lépésből az első három az igényoldalt jelenti tehát annak meghatározását, hogy mekkora igény mutatkozik a közlekedési rendszer igénybevételére. Az igény modell során a mátrixok mutatják a közlekedési igényeket. A mátrixban lévő utazásszámok a teljes közlekedési rendszerre vagy annak egy részére (gyalogosok, tömegközlekedés, egyéni közlekedés stb.), társadalmi csoportokra (hallgatók, munkavállalók, nyugdíjasok stb.) és utazási célokra (bevásárlás, pihenés, munkába járás stb.) vonatkoznak.

A negyedik lépés a ráterhelés, amely meghatározza, hogy a felmerülő keresletet hogyan tudja kiszolgálni a rendelkezésre álló vagy a tervezett kínálat (úthálózat, közforgalmú közlekedési szolgáltatás). Ennek a kínálatnak a megjelenítésére szolgál a közúthálózatot, illetve a közforgalmú közlekedési rendszert ábrázoló gráf. A közlekedési rendszerben bekövetkező változások (pl. új út építése) hatással van a közlekedőkre (pl. utazási idő), az üzemeltetőkre (járműszám, bevétel, költség), a közösségre (akiknek előnye származik belőle, de ők fizetik meg), a tömegközlekedési szolgáltatóra és a környezetre. A hatások meghatározása különböző ráterhelési eljárásokkal, ún. skim mátrixokkal (ellenállás mátrixokkal), környezeti modellel (zaj és légszennyezés) és üzemeltetői modellel (üzemeltetési és finanszírozási igények) történhet.

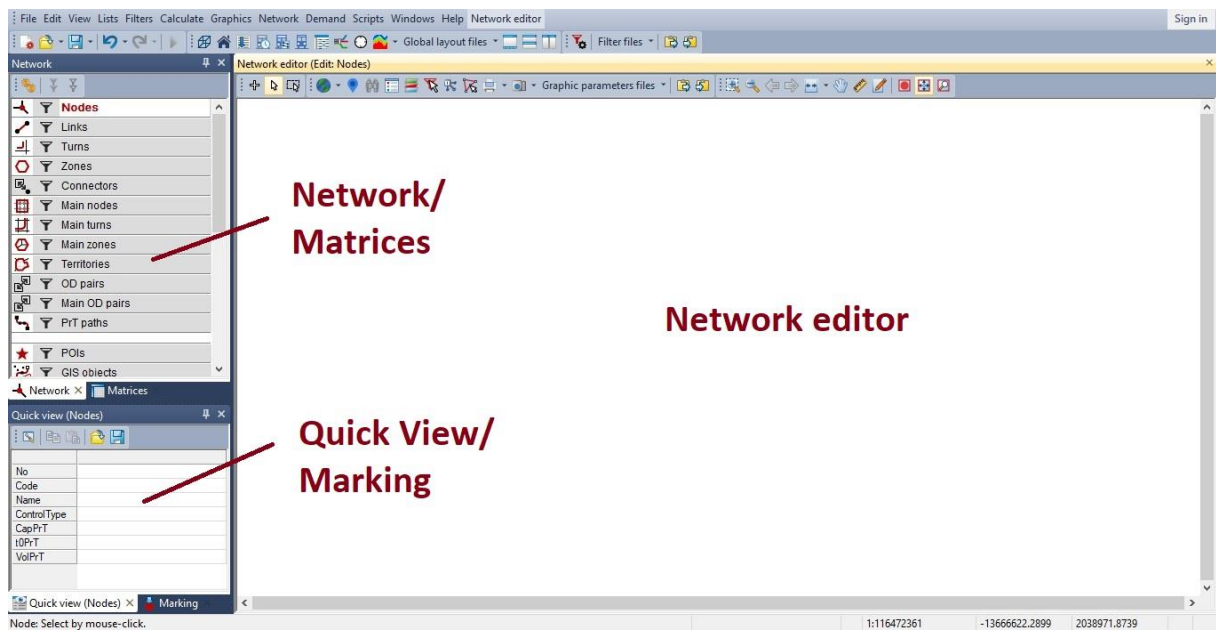
A következő fejezetekben ezzel a negyedik lépéssel, azaz a forgalom ráterheléssel fogunk foglalkozni. A módszer alapvető ismeretéhez a továbbiakban csak az egyéni közlekedést vesszük figyelembe és azon belül nem vizsgáljuk a forgalom megosztást. Az egyénit PrT-vel és a közösségi közlekedést PuT-vel jelöli a program, ez egyes beállításoknál ezeket meg kell különböztetni, így a továbbiakban mi a PrT-t fogjuk használni. A közlekedési hálózat felvétele után, zónákra osztjuk a hálózatot, amik a ki- és beérkező körzeteket jelentik. Utána előállítjuk a szétosztási mátrixot, azaz a célforgalmi mátrix belsejét vesszük fel és általa megkapjuk a kiinduló és beérkező forgalmak nagyságát. A hálózatra történő ráterhelést követően megkapjuk az egyes útszakaszok forgalmi terhelését.

## 2. lecke: Hálózat felépítése

A Visumban a hálózati modellobjektumok az alábbiak:

- Node; csúcspont azaz a közúti hálózat csomópontjai
- Zone; a forgalmi igények kiinduló- és célkörzetei
- Link; él a megengedett közlekedési rendszer egy irányított eleme, ami például egyéni közlekedésnél lehet a sebesség és kapacitás, tömegközlekedésnél az utazási idő alapján súlyozhatunk. Mindkét irány külön objektumnak számít és különböző éltípusok definiálhatók.
- Turn; kanyarodási lehetőség beállítása
- Connector; konnektorok a zónák közötti kapcsolat (Zónából kilépés, zónákba belépés), tömegközlekedés esetében a kapcsolat olyan csúcsponttal, amelyiknél megállóterület és megálló is van.

Ebben a következő leckékben ezen objektumok felvételével ismerkedünk meg. Indítsuk el a Visum programot! Alaphelyzetben az alábbi ablakok aktívak: Network editor, Network, Marking, Quick view és Matrices. Az ablakokat aktiválni a View menüben lehet. A leírtak az 5.3. ábrán láthatók.



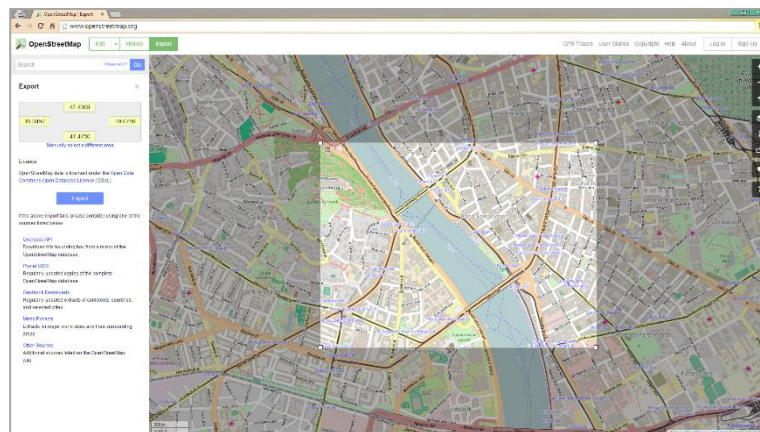
5.3. ábra: Aktív ablakok

A Network editor ablakban történik a hálózat felépítése és a Network ablakban választhatjuk ki a hálózati elemeket.

## 1. A háttér előkészítése és behívása

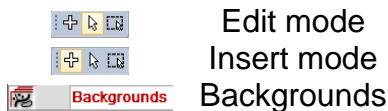
A Vissimhez hasonlóan a hálózati elemek létrehozásához használható háttértérkép, amelyre történik az elemek fölvétele. A Visumban betölthetők már kész shapefájlok (lásd 6. QGIS alapok, gyakorlatok fejezetet) vagy OpenStreetMap (OSM) térkép is, amely tartalmazza a linkeket és nodokat. A következőkben nézzük meg, hogyan lehet egy OSM fájlt letölteni a <https://www.openstreetmap.org> honlapról.

Az oldal felső menüsorában jelöljük ki az Export gombot, ahogy az 5.4. ábrán is látható. Ezután a térkép ablakban állítsuk be a letölteni kívánt térképet vagy adjuk meg az oldalsó sávban a szélességi és hosszúsági koordinátákat. Végül Az Export fülön kattintsunk az Export gombra.



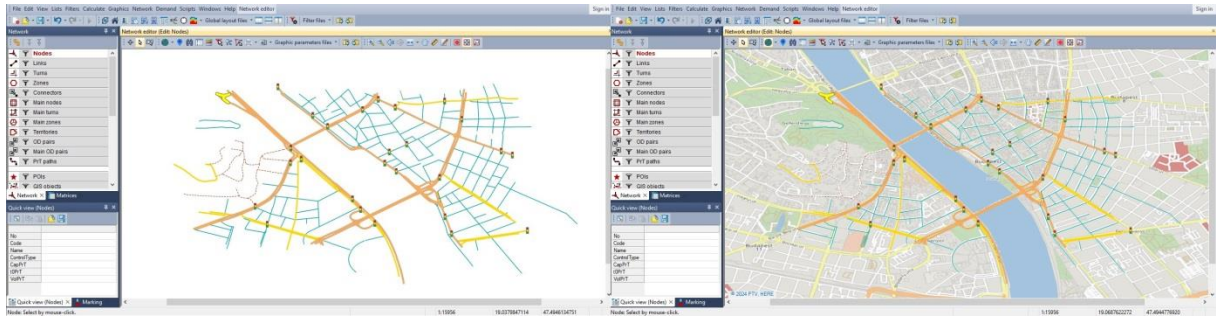
5.4. ábra: OpenStreetMap letöltése

A Visumba a térkép betöltése a File/Import/OpenStreetMap menü alatt történik. (A Configuration set-nél válasszuk az Urban road network-öt Student licenc-es Visumnál, különben nagyméretű lesz a fájlunk, amit később nem lehet menteni.) Alapbeállításaként azok a hálózati elemek látszódnak, amelyek aktiválva vannak a Network ablakban, azaz a név melletti ikon hátere fehér és nem szürke. Állítsuk be úgy a térképet, hogy elsősorban a Node-ok és Link-ek látszódnak! Ha szerkeszteni akarjuk őket, akkor az Edit mode legyen kijelölve a Network ablakban, ha új elemet szeretnénk hozzáadni, akkor pedig az Insert mode-ra kattintsunk.



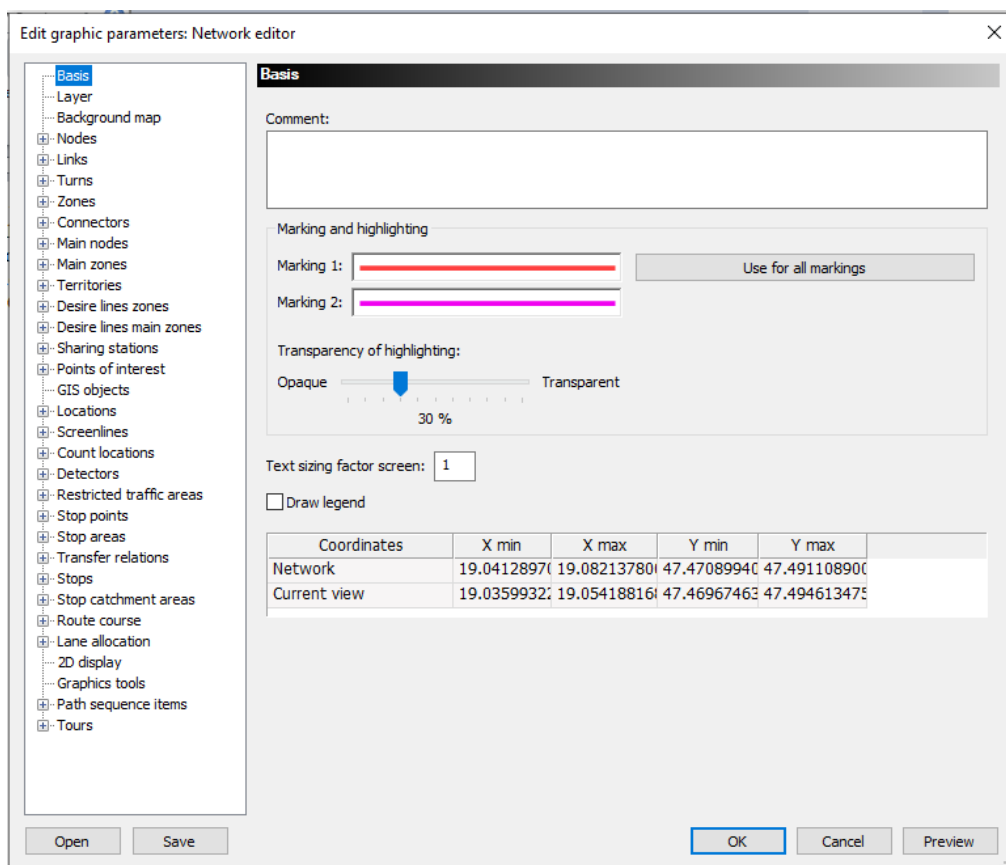
Ha a program indításakor nem fogadtuk el az OSM licenc szerződés feltételeit, akkor a hálózatunk mögé utólag is tehetünk térképet, ehhez maradjunk Insert mode-ban. A Network ablakban található Backgrounds fülre kattintva (vagy Edit/Network objects/Backgrounds menü alatt) és utána a Network editor ablakba hozzáadhatjuk a térképet, használhatjuk az Internet maps-et és az adott felbontásban (ahogy a Network editor ablakban a hálózat látszik) automatikusan mögé helyezi grafikusán az OSM térképet (lásd 5.5. ábra).





5.5. ábra: Az OSM térkép file importálása és a háttér megjelenítése

Az egyes hálózati elemek grafikus megjelenítését magunk is módosíthatjuk, például a linkek színét saját útkategóriák alapján is beállíthatjuk, de ugyanígy a node-oknál is jelölhetjük a különböző típusú csomópontokat színezéssel. Ezt a Graphics/Edit graphic parameters menüben állíthatjuk, a felugró ablak bal szélén a fastruktúrából kiválaszthatjuk a megfelelő hálózati elemet és a Displayben beállíthatjuk az egyes paramétereket, ahogy az 5.6. ábrán látszik.



5.6. ábra: Hálózati elemek grafikus beállítása

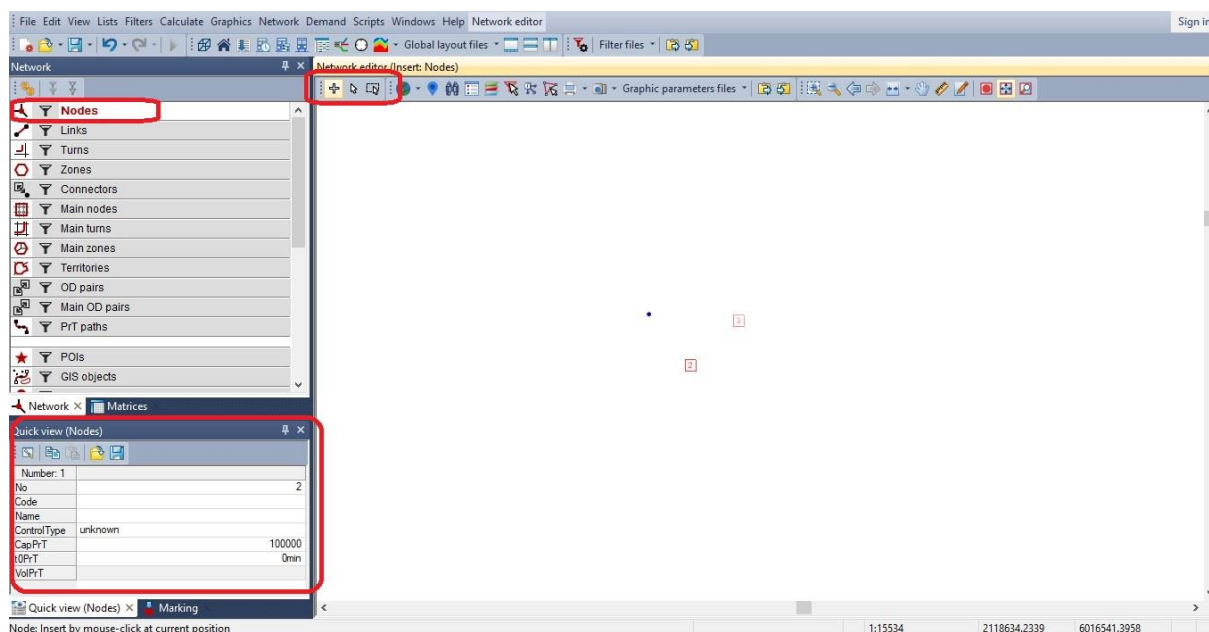
Az egyes hálózati elemek megjelenítési módjait állíthatjuk a Ctrl+G billentyű kombináció megnyomásával is. Választhatunk címkét, az egyes elemek megjelenítésének színét, kategóriákat képezhetünk attribútum szerint, stb.

### Gyakorlás:

- Töltsünk le egy OSM fájlt az alábbi koordináta paraméterekkel: 19.0450 - 19.0750, 47.4750 - 47.4900.
- Ezt a térképet töltsük be a Visumba és helyezzünk mögé grafikus térképet is.

### 3. lecke: Hálózati elemek hozzáadása

Az előző leckében felvett hálózatban meglévő úthálózati elemeket módosíthatjuk vagy adhatunk hozzájuk újabb elemeket. Ebben a leckében node, link és turn felvételére mutatunk példát, illetve ezen hálózati elemek beállítási lehetőségeit vizsgáljuk.



5.7. ábra: Node felvételéhez szükséges beállítások

Figyeljünk arra, hogy Edit mode szerkeszteni tudjuk az általunk kiválasztott elemet, illetve ha új elem szeretnénk hozzáadni, akkor pedig az Insert mode-ra kattintsunk, ahogyan a 5.7. ábra szemlélteti.

#### 1. Node

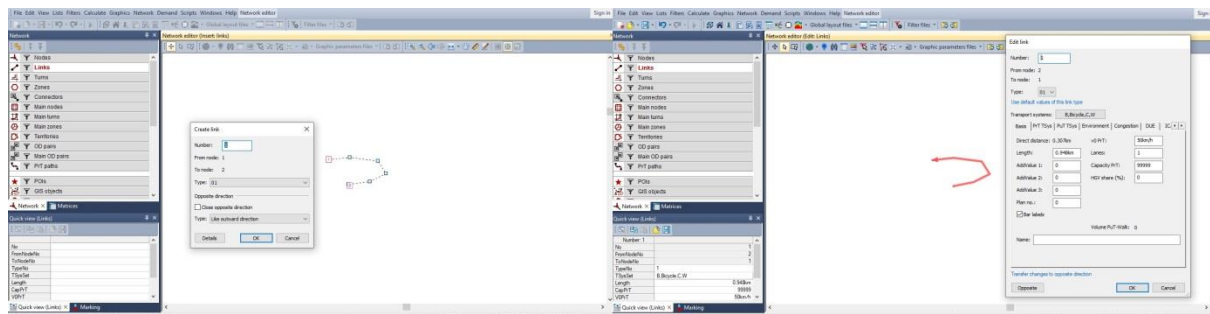
Először node-okat kell létrehozni. A Network ablakban jelöljük ki a Nodes fület és a térképen a megfelelő helyre kattintva elhelyezhetünk node-ot, a Quick View ablakban adhatunk nevet és egyéb beállításokat is elvégezhetünk. A meglévő elem adatait az Edit mode kiválasztásával az adott node-ra kattintva lehet előhívni, ekkor megjelenik a node azonosító száma a térképen. A node-ra duplán kattintva felugrik a Junction editor (node ID) ablak.

#### 2. Link

A Network editor ablakban látható vonalszakasz mindkét forgalomiránynak megfelelően tartalmaz link-et. A menetirány szerinti oldalára kattintva érhető el az adott irányba mutató link és a kijelölést követően vastag vagy vékony piros nyílra vált, annak megfelelően, hogy az adott menetirány létezik-e, azaz aktív vagy passzív-e a link. Létrehozása a Node létrehozáshoz hasonló módon történik. A Network ablakban jelöljük ki a Links fület és a térképen egy adott node-tól indulva töréspontosan elhelyezhetünk linket, amíg végpontként



egy másik node-ra nem kattintunk. Linket létrehozhatunk úgy is, hogy csak a kezdő és a végpontnak megfelelő node-okra kattintunk, ekkor legrövidebben egy egyenest helyez el a program. Create link felugró ablakban a számát, illetve az egyes irányok típusát állíthatjuk be, OK gombbal létrehozható. A link kijelölése után az Edit... kiválasztásával változtathatjuk a beállításokat, és itt már nevet is adhatunk neki (lásd 5.8. ábra).

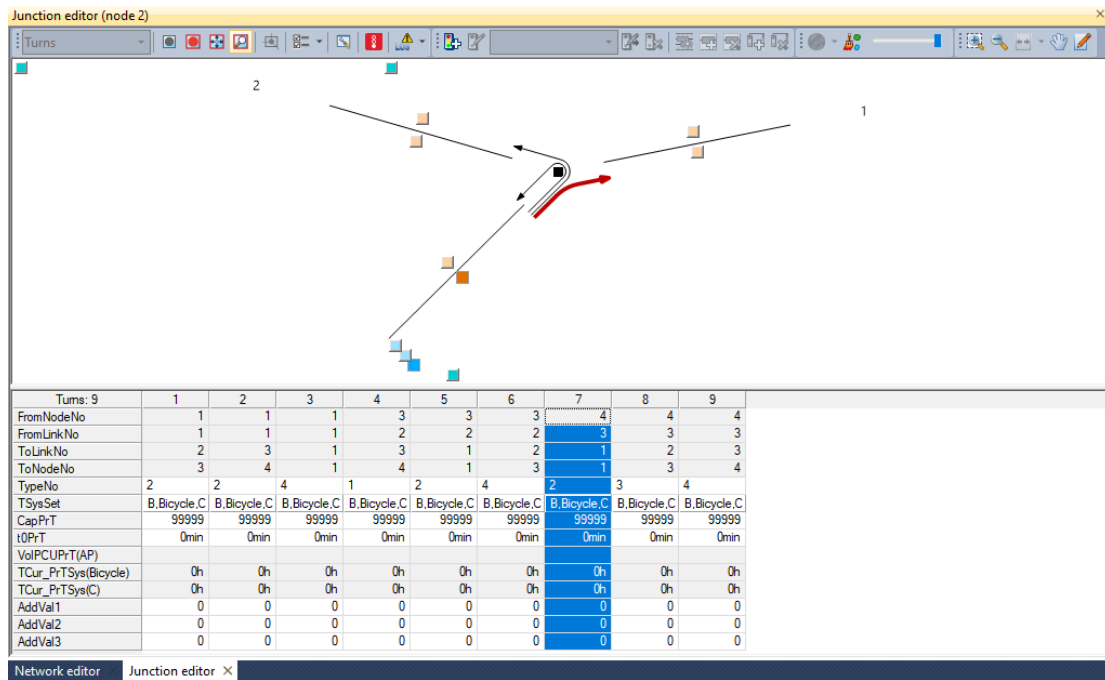


5.8. ábra: Link felvétele (Create link) és szerkesztése (Edit link)

Node-ot, linket, és a többi hálózati elemet keresés alapján is kiválaszthatjuk, ha tudjuk az azonosító számukat. A Network ablakban a keresni kívánt elem típus legyen kijelölve! Majd a Network editor ablakban jobb klikkelés az egérrel, és a felugró ablakból válasszuk a Find... gombot! Itt megadhatjuk a számát vagy listából ki is választhatjuk, a listában az elemre jobbal klikkelve, majd a Zoom opciót választva a térképnézet az adott elemhez ugrik.

### 3. Turn

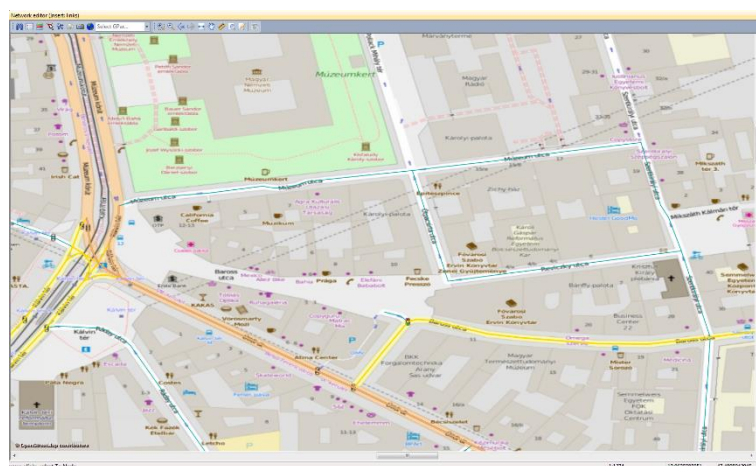
Turn-ök beállításához Edit mode-ban kell lennünk és kijelölni a Turn fület. Ezután a megfelelő node-ra kell kattintani duplán, ami után megjelenik a Junction editor ablak, ahol a vizsgált csomópont feketével, a szomszédosak kékkel jelöltek. Ha a paraméterablakban egy turn-höz tartozó oszlopot kijelölünk, akkor azt az 5.9. ábrán vastag piros nyíl jelöli, valamint mellette megjelenik az abból az irányból (amit narancssárga négyzet jelöl) lehetséges többi turn fekete nyila. Ha megszüntetni szeretnénk egy fordulási lehetőséget, akkor a TSysSet beállításnál töröljük az összes közlekedési módot, azaz tiltjuk ezeket. Közlekedési mód a szoftver értelmezésében nem csak az egyéni és közösségi közlekedés közötti megosztást jelentheti, hanem megkülönböztethetünk pl. járműtípusokat is (személy- és tehergépjárművek). Ha az oszlopfőn jobb egérgombra előjövő ablakban bepipáljuk a Show also turns of blocked links és a Show also closed turn megjelenítési lehetőségeket, akkor megjelenik az összes lehetséges turn, és a TSysSet-tel újra aktiválhatjuk ezeket.



5.9. ábra: Turn beállítás

#### Gyakorlás:

- Helyezzünk egy Node-ot a Múzeum utca és a Szentkirályi utca sarkára.
- Hozzunk létre linkeket, az itt található mellékutcák középvonalára (Múzeum utca, ill. a Szentkirályi utca Múzeum utca és Reviczky utca közötti szakasza), és az utcanevek alapján nevezzük el őket! Figyeljünk az utcák irányítására és annak megfelelően kezdjük el felvenni! Ha az utca nem egyenes, akkor töréspontokat helyezhetünk el két node között (lásd 5.10. ábra).

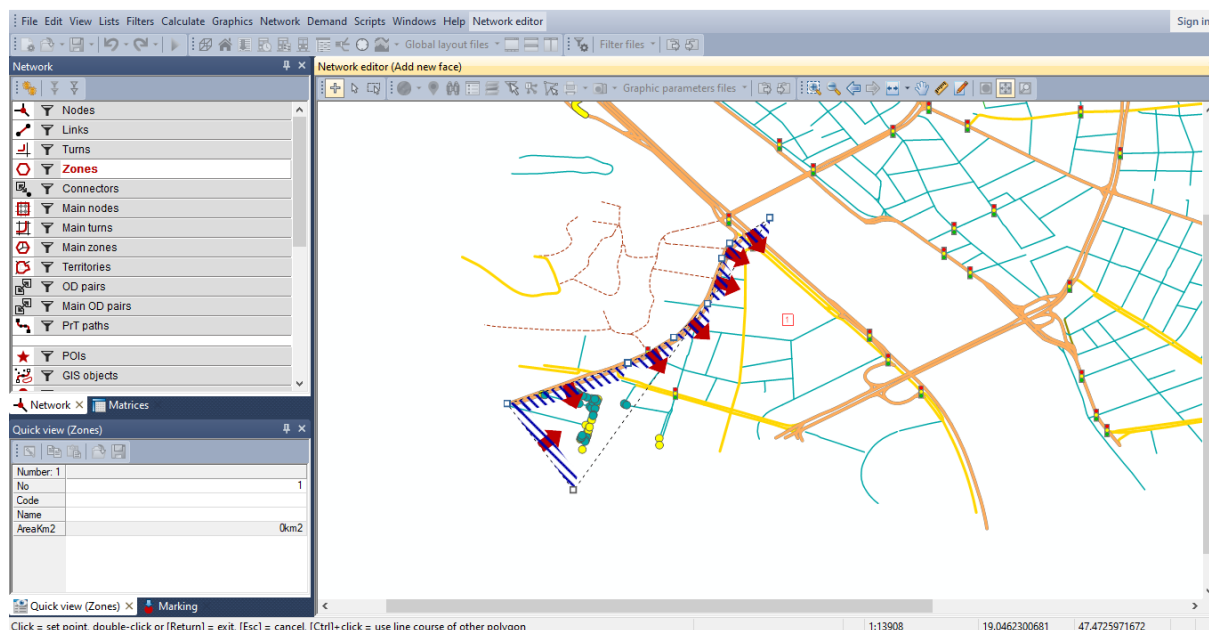


5.10. ábra: Múzeum utca és a Szentkirályi utca környékére felvett node és linkek

- Vizsgáljuk meg az újonnan létrehozott csomópontot! Látható, hogy az adott node-hoz kapcsolódó linkek irányításának megfelelően létrejöttek a turn-ök. Gyakorlásképpen inaktíváljunk turn-öket és utána állítsuk is vissza őket!

## 4. lecke: Zónák kialakítása

Egy zóna létrehozásához először ki kell jelölni a zóna közepét, azaz a kiinduló és célkörzetek centroidját, aminek nem kell egybe esnie a zóna geometriai közepével. A centroid kijelölése lehet megadni a zónahatárokat. A sraffozott vonal közepén látható piros nyíl a zóna közepe felé mutasson, ahogy ez az 5.11. ábrán is látszik. Bal egérgombbal lehet tördelni a vonalat, a jobb egérgombbal pedig befejeződik a zónaterület kijelölése. A bal alsó, Quick view ablakban adhatók meg a tulajdonságok.



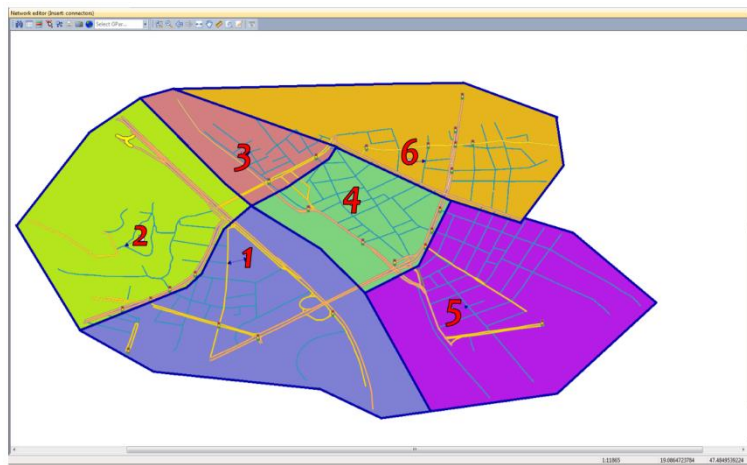
5.11. ábra: Zóna felvétele

Ahhoz, hogy a Network editor ablakban a zónák neve legyen látható, az alábbi beállítást kell végrehajtani. Edit mode-ban a Network ablakban jobb egérgombbal felugró ablakból válasszuk ki az Edit graphic parameters opciót, majd a fastruktúras menüben álljunk a Zones/Display-re és az Active fülnél a Content-nél legördülő menüből válasszuk ki a Name-t vagy azt a paramétert, amit megjeleníteni szeretnénk.

A zónákat kapcsolnunk kell a hálózathoz, erre van a Connectors, azaz a csomópontokat (nodes) és a körzetközpontokat (zones) kötjük össze vele. A létrehozott zónákhoz kapcsolódik az OD-mátrix, a zónák jelentik a vizsgált körzeteket és az egyes körzetekhez tartozó forgalmi igényeket kell a szimuláció során a hálózatra ráterhelni. Azaz a konnektorok (Connectors) azok a hálózati elemek, amelyekkel az OD-mátrix egyes celláiban meghatározott forgalmi igényeket a hálózatra bocsátjuk. Egy zónához minimum egy, de akár több konnektor is kapcsolható. Legyünk Edit Mode-ban és válasszuk ki a Connectors fület. A Network editor ablakban klikkeljünk bal egérgombbal a zóna centroidjára, majd szintén bal egérgombbal a hálózat azon pontjára (a zónán belül), ahová a konnektort csatlakoztatni kívánjuk. A bal alsó, Quick View ablakban megadhatjuk, hogy a konnektor mely közlekedési módok számára vehető igénybe. Ez a jelöléssel megadjuk azt is, hogy az egyéni vagy a közösségi közlekedési módot választók igényeit terheljük-e rá a hálózatra az adott konnektoron keresztül.

### Gyakorlás:

- Az 5.12. ábrán látható elrendezésben hozzunk létre 6 darab zónát, az alábbi nevekkel: BME, Gellért-hegy, Belváros, Belső-Ferencváros, Középső-Ferencváros, Józsefváros!
- Határok:
  - 1-BME: Duna közepe, Bartók Béla út mentén (de nem tartozik hozzá)
  - 2-Gellért-hegy: Duna közepe, Bartók Béla út
  - 3-Belváros: Duna közepe, Vámház körút
  - 4-Belső-Ferencváros: Duna közepe, Ferenc körút, Üllői út mentén, Vámház körút mentén
  - 5-Középső-Ferencváros: Duna közepe, Ferenc körút mentén, Üllői út mentén
  - 6-Józsefváros: Üllői út

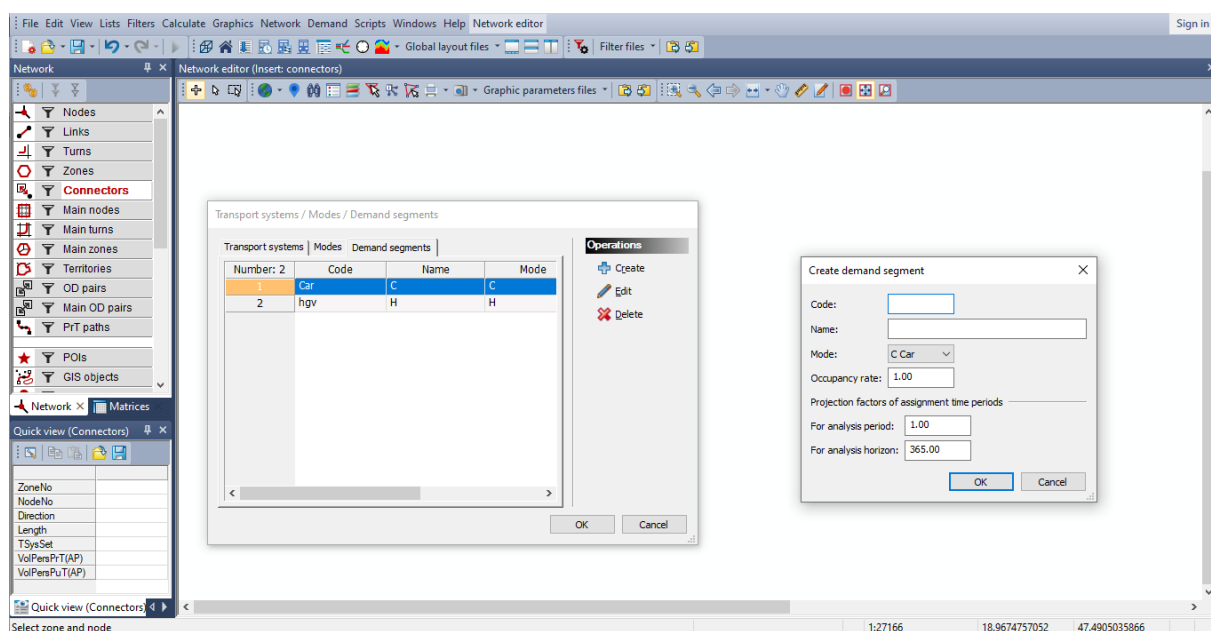


5.12. ábra: Zónák határai

## 5. lecke: OD mátrix előállítás

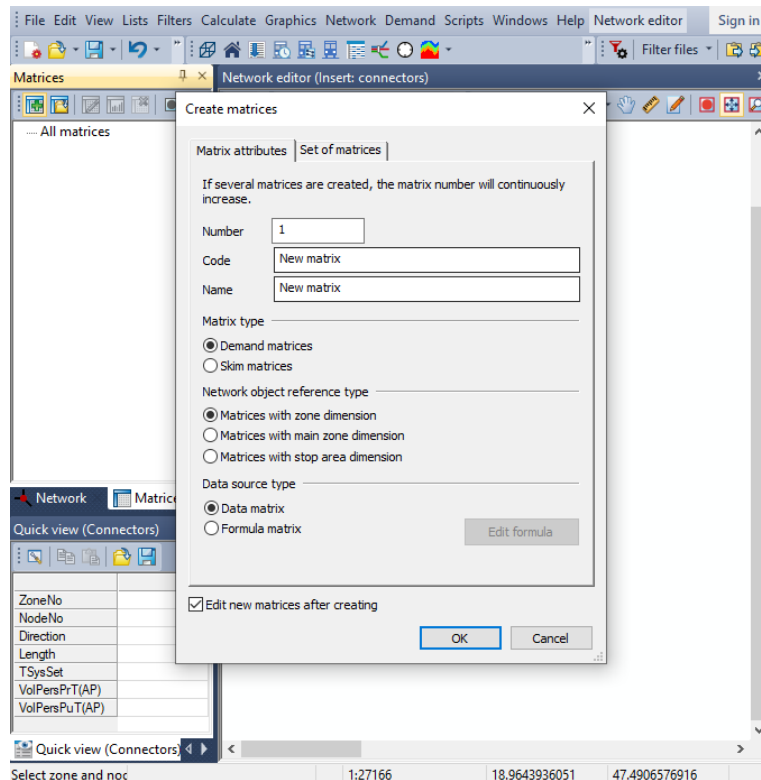
Az OD mátrixok úgynevezett igényszegmensekhez (demand segment) kapcsolhatók. Az egyes forgalmi módokat különböztetjük meg vele, ilyenek pl. az egyéni közlekedők igényszegmensei (pl.: külön a személygépjárművek vagy a kis, közepes és nehéz tehergépjárművek). A programban található 4 féle alapbeállítás; Bus, Car, Hgv és PuT Walk (busz, személygépjármű, tehergépjármű és közösségi közlekedésre rágyaloglás), de ezekhez mi magunk is hozzáadhatunk.

A főmenü Demand résznél válasszuk ki a Transport systems / Modes / Demand segments menüt, és a Create gombbal nyissuk meg a Create demand segment ablakot! Állítsuk be a paramétereiket! (lásd 5.13. ábra)



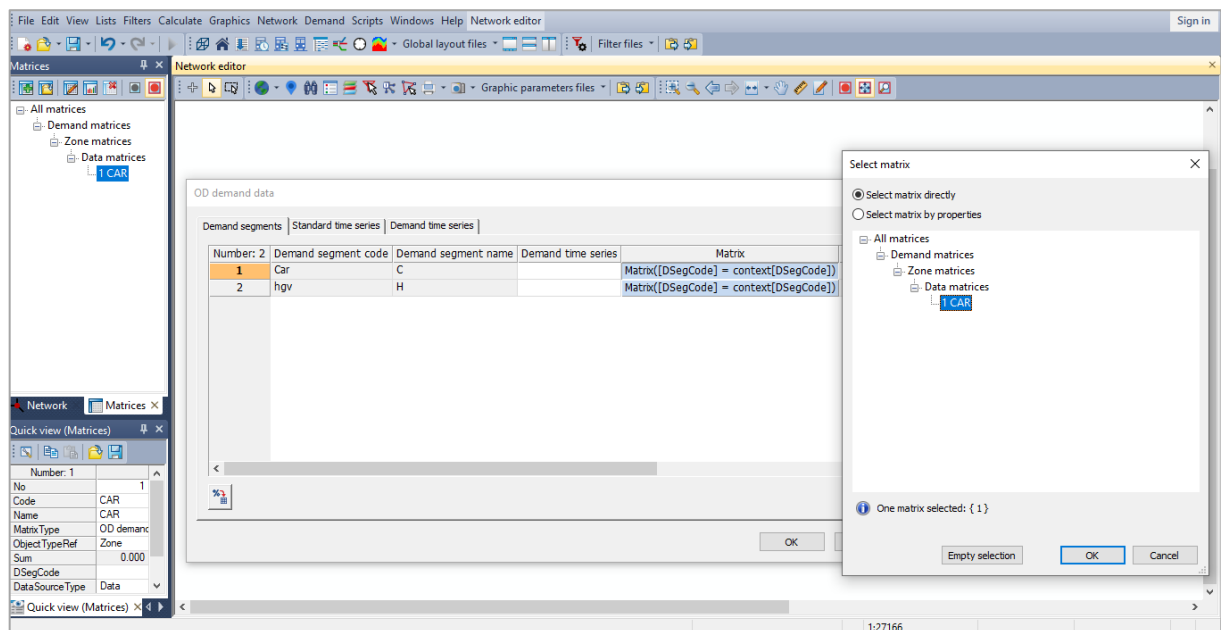
5.13. ábra: Demand segment megadása

A Network ablaknál váltsunk az alatt megjelenő Matrices ablakra és kattintsunk a Create matrices gombra! Adjunk kódot és nevet a mátrixnak, és a Matrix type-nál jelöljük ki a Demand matrix for zones-t! Az OK megnyomással elkészül a mátrixunk (lásd 5.14. ábra). Mátrixot külső forrásból is meg lehet hívni, akár Excel táblázat is importálható, továbbá az előállított mátrixot külön el is lehet menteni.



5.14. ábra: OD mátrix létrehozása

Rendeljük össze a demand segment-et az OD mátrixszal! Nyissuk meg a Demand / Demand data menüt, a felugró ablakból válasszuk a Demand segments fület! Az általunk választott demand segment sorában rendeljük hozzá a mátrixot! (lásd 5.15. ábra)



5.15. ábra: Demand segment és OD mátrix összerendelése

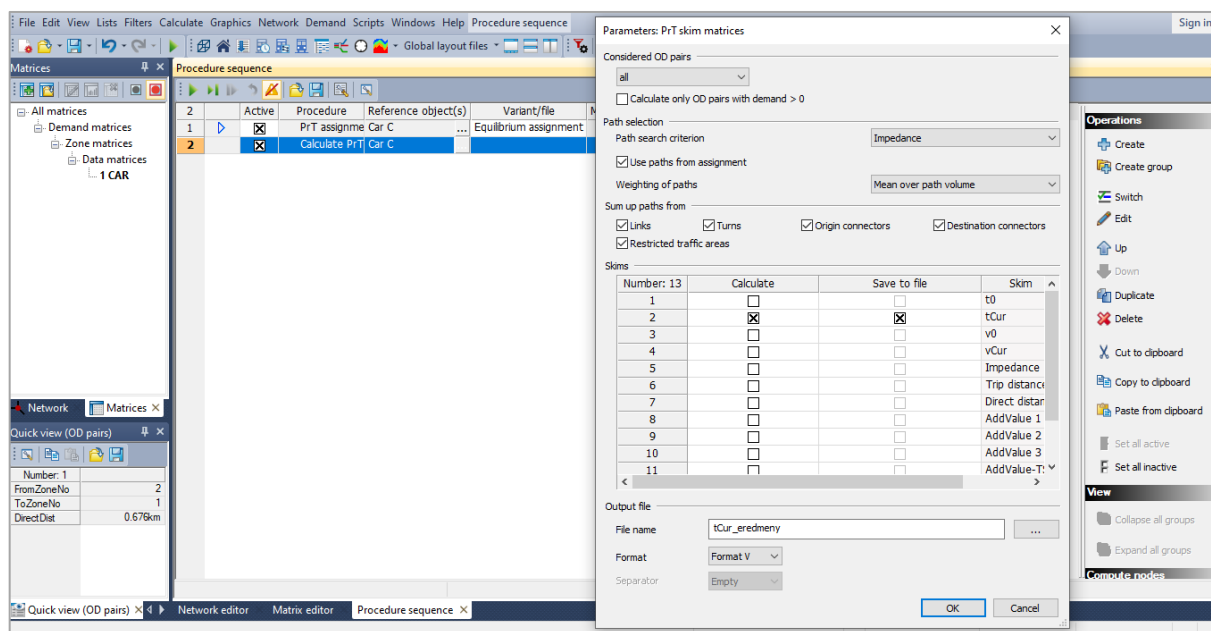
### Gyakorlás:

- Hozzuk létre egy új igény szegmenst, adjuk meg a kódját, nevét és a gördülő menüből válasszuk ki közlekedési módot (pl.: Code: 01; Name: First DS; Mode: C CAR)!
- Hozzuk létre egy OD mátrixot, és töltsük ki egyforma (pl. 100) forgalomszétosztási értékekkel!

## 6. lecke: Forgalom ráterhelés a hálózatra

A Főmenüből válasszuk ki Calculate / Procedure sequence menüt és az ablak jobb oldalán található Create gombra kattintsunk! A felugró Procedure ablakból jelöljük ki az Assignments közül a PrT assignment-et és OK gombbal hozzuk létre! A Reference object(s) oszlopban válasszuk ki a demand segment-et (a kód és neve jelenik meg), valamint a Variant/File oszlopban jelöljük ki az Equilibrium assignment-et!

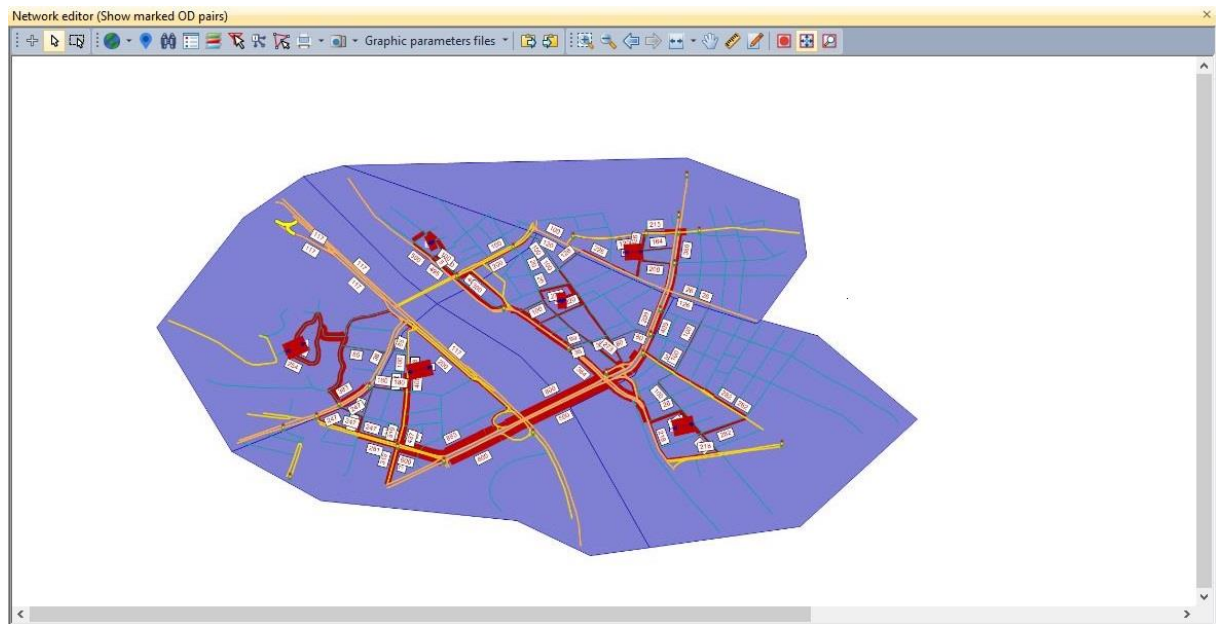
Az egyes zónákhoz különböző indikátorokat rendelhetünk, pl. vizsgáljuk meg az egyes zónák közötti átlagos utazási időket (tCur). A szimuláció futása során a program létrehoz egy úgynevezett skim mátrixot, amelynek sor és oszlopai az OD mátrixszal egyeznek meg, belsejét az általunk választott indikátor alapján tölti ki. Egy újabb Create gombbal hozzunk létre számítással egy skim matrix-ot, azaz a Procedure ablakban található fastruktúrából válasszuk ki a Matrices / Calculate PrT skim matrix-ot! Itt is a References object(s)-hez rendeljük hozzá az általunk használni kívánt demand segment-et! Kétszer kattintsunk a sor elejére és felugrik a Parameters: PrT skim matrices ablak, ahol jelöljük ki a tCur sorban a Calculate oszlopot és utána a Save to file-t is, adjuk meg a file nevét, esetleg elérési útját és OK gombbal zárjuk be az ablakot!



5.16. ábra: Parameters: PrT skim matrices ablak

Futassuk le a számolást a Start - Run all active operations gombra kattintva vagy a főmenüben a Procedure sequence / Start gombbal! Az eljárás futása során a Matrices ablakban a Demand matrices mellett létrehoz egy Skim mátrixot, ahol a számított értékek szerepelnek!





5.17. ábra: A hálózat a ráterhelt forgalommal

Gyakorlás:

- Végezzük el a forgalom terhelés számítását először konstans értékekkel, majd az módosított adatokkal is!
- Figyeljük meg a változást a térképen!

## 7. Irodalom az 5. modulhoz

- [1] Horváth R.: Rendszerdinamika mint a közlekedési rendszerek igénymodellezésének új lehetősége, Doktori értekezés, 2012
- [2] PTV, VISUM 2023 User Manual PTV Planung Transport Verkehr AG, Germany, 2023
- [3] PTV, Vision Tutorial VISUM 2023 QuickStart PTV Planung Transport Verkehr AG, Germany, 2023

## 6. modul: QGIS alapok, gyakorlatok

### Cél:

A modul célja a térinformációs rendszerekbe (GIS: Geographic Information System) történő bevezetés, a közlekedésmérnöki területen használt alapvető GIS adatállomány-típusok ismertetése, illetve az ezekkel végezhető elemi műveletek bemutatása a QGIS szoftverben.

A középpontban a gyakorlati alkalmazások bemutatása áll, különös hangsúlyt fektetve a makroszkopikus forgalmi modellek hálózati adatainak előállítására. Az elméleti háttér ismertetése csak olyan mértékű, ami feltétlenül szükséges a gyakorlati használat alátámasztására, illetve a megértés elősegítésére.

### Követelmények:

Ön akkor sajátította el megfelelően a tananyagot, ha képes

- különböző shape-adatállományok létrehozására, illetve módosítására,
- a szoftveres környezet rétegeinek kezelésére,
- a shape-adatállományok megjelenítésére vonatkozó alapvető beállítások (címkézés, szűrés, stílusbeállítás) elvégzésére,
- térbeli és logikai lekérdezések alkalmazására.

### Időszükséglet:

A tananyag elsajátításához körülbelül 360 percre lesz szüksége.

### Kulcsfogalmak:

- térinformációs rendszerek,
- shape-fájl,
- réteg,
- QGIS gyakorlatok.

## 1. lecke: Térinformációs rendszerek alapjai

A lecke a térinformációs rendszerek fogalmát, valamint a QGIS térinformatikai programot mutatja be olyan alapszinten, hogy a későbbi fejezetek gyakorlatai mögötti tartalom láthatóvá és érthetővé váljon.

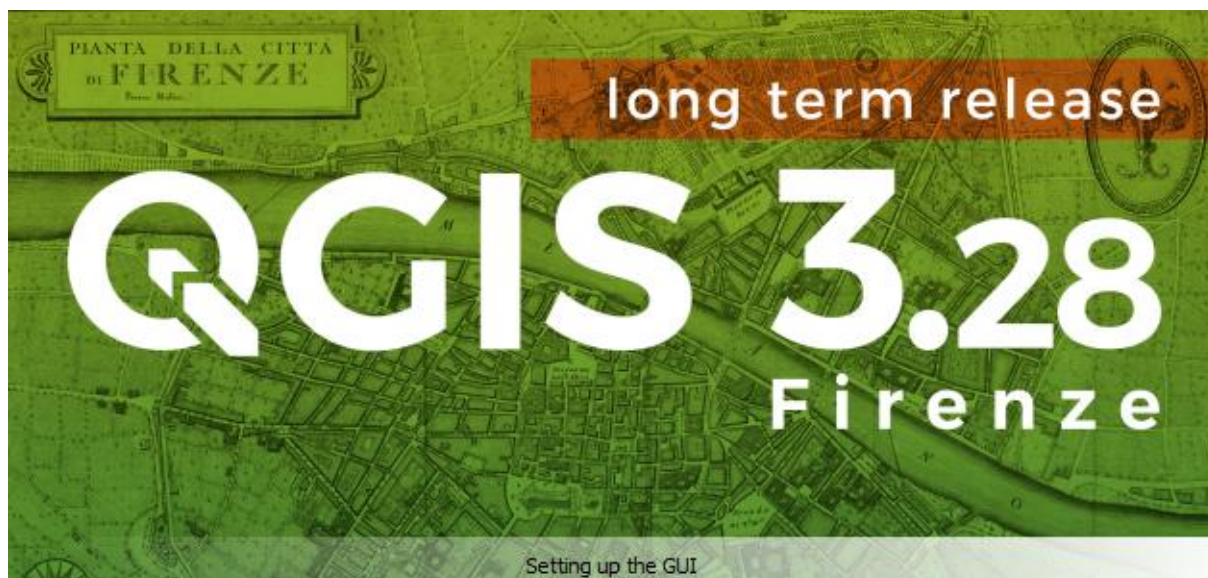
### A térinformációs rendszerek

Definíció szerint: „A térinformációs rendszerek azok az információs rendszerek, amelyekben a tárolt adatok földrajzi helyhez vagy objektumhoz vannak hozzárendelve, és a keresések, lekérdezések a földrajzi elhelyezkedés alapján is elvégezhetők. [4]” Gyakran találkozhatunk a témában a geodatbázis kifejezéssel, ami olyan adatbázis, melyben az egyes rekordoknak az attribútumaikon felül olyan tulajdonságaik is vannak, amik azok térbeli kiterjedésére, illetve elhelyezkedésére vonatkoznak.

A térinformációs rendszerek tanulmányozása során találkozhatunk az angol elnevezésből (Geographic Information System) származó GIS mozaikszóval. A modul során bemutatásra kerülő QGIS szoftver elnevezése is innen származik. A Q a korábbi „Quantum GIS” elnevezésből ered, melyet már csak a köznyelvben használnak időnként.

### A QGIS

A QGIS egy ingyenes, nyílt forráskódú térinformációs rendszer, amelynek pár havonta jelennek meg újabb és újabb verziói. Emiatt a jegyzet célja nem a legmélyebbre menő szoftverbemutatás, hanem olyan általános ismeretek átadása, amelyek később más térinformációs szoftverekben is alkalmazhatóak lesznek majd a mérnöki tervezési munkák során.



6.1. ábra: A QGIS 3.28-as verziója

---

[4] Munkácsiné Dr. Lengyel E., Dr. Tóth J., Dr. Csiszár Cs., Juhász J.: Közlekedési Informatika, elektronikus jegyzet, Budapest, BME, Közlekedésüzemi Tanszék, 2004

A szoftver letölthető a [www.qgis.com](http://www.qgis.com) webhelyről. Ugyanitt található egy szinte mindenre kiterjedő dokumentáció, valamint gyakorlati feladatok is.

## Digitális térképek fajtái

Amennyiben a valós világot digitális térképekre kívánjuk leképezni, alapvetően két lehetőségünk van:

1. Vektoros digitális térképet alkalmazunk. Ekkor a valós világ elemeit egy ponttal vagy valamilyen alakú szakasszal vagy felületdarabbal jelenítjük meg.
2. Raszteres digitális térképet alkalmazunk. Ekkor a valós világ fölé egy felületet helyezünk, és ezt elemi egységnyi kis részekre bontjuk.

A raszteres térkép által megjelenített kép részletesebb, szebb, viszont a térkép elemei közötti kapcsolatokat nem tudjuk benne tárolni. Utóbbira képes a vektoros térkép, igaz a vizuális jellemzői elmaradnak a raszteresétől. Emiatt azonban kisebb az ilyen fájlok mérete, ami előnyös lehet.

A különböző térképekkel végzett munka során a közlekedésmérnöki gyakorlatban általában a kapcsolatok kezelése hangsúlyos, így a továbbiakban kizárólag a vektoros térképekkel foglalkozunk. Ennek standard megjelenítési formája az Esri shape-fájl, bár egyéb, szoftverspecifikus adatállományok is léteznek.

## Fájltípusok

Jelen alfejezet célja kizárólag a makroszkopikus forgalmi modellezésben alkalmazott vektoros fájl típusok olyan szemmel történő bemutatása, hogy a gyakorlati alkalmazás mögötti tartalom megértését elősegítse. A részletes elméleti háttér ismertetése nem cél.

A térinformációs rendszerek, így a QGIS használata során is, két nagy alapvető fájlcsoportot érdemes ismerni, a shape-fájlt és a QGIS projekt-fájlt.

### 1. Shape-fájl

A jegyzetben bemutatott példákban a shape-fájl - amit a kifejlesztő cégre utalva Esri shape-fájlnak is neveznek - szolgál a térinformációs rendszer definíciójában megfogalmazott geodatabázis eltárolására. Fontos kiemelni, hogy az elnevezéssel ellentétben a shape-fájl nem egy fájl takar, hanem többet. Amikor egy shape-fájlt elmentünk egy adott néven, igazából több fájl mentünk azonos névvel, de eltérő kiterjesztéssel. A „több” az kötelezően 3-at jelent, melyek az alábbiak:

- „.shp”: shape-formátum, ami az elemek geometriai adatait tartalmazza,
- „.shx”: shape index formátum, ami az elemek indexeit tartalmazza,
- „.dbf” formátum, ami az egyes rekordok attribútumait tartalmazza (így pl. Excelben is megnyitható).

Ezenfelül még egyéb kiterjesztésű fájlokat is tartalmazhat egy shape, ilyen például a „.prj”, ami a koordináta-rendszerhez kapcsolódó információkat tárolja.

Ilyen módon, ha például egy „minta” nevű shape-fájlt a számítógépen másolni akarunk, akkor azt a minta.shp, minta.shx és a minta.dbf (illetve, ha van, akkor minta.prj, minta.ixs, stb.) fájlok együttes másolásával tehetjük meg.

A fájlkiterjesztésből nem olvasható ki, de az egyes rekordok térbeli kiterjedése alapján a shape-fájlokat három csoportra bonthatjuk:

- point (pont) shape-ek - ilyenek például a PTV Visum node-jai vagy zone centroidjai,
- line (vonal) shape-ek - ilyenek például a PTV Visum linkjei vagy connectorai,
- polygon (sokszög) shape-ek - ilyenek lehetnek a PTV Visum zone-jai.

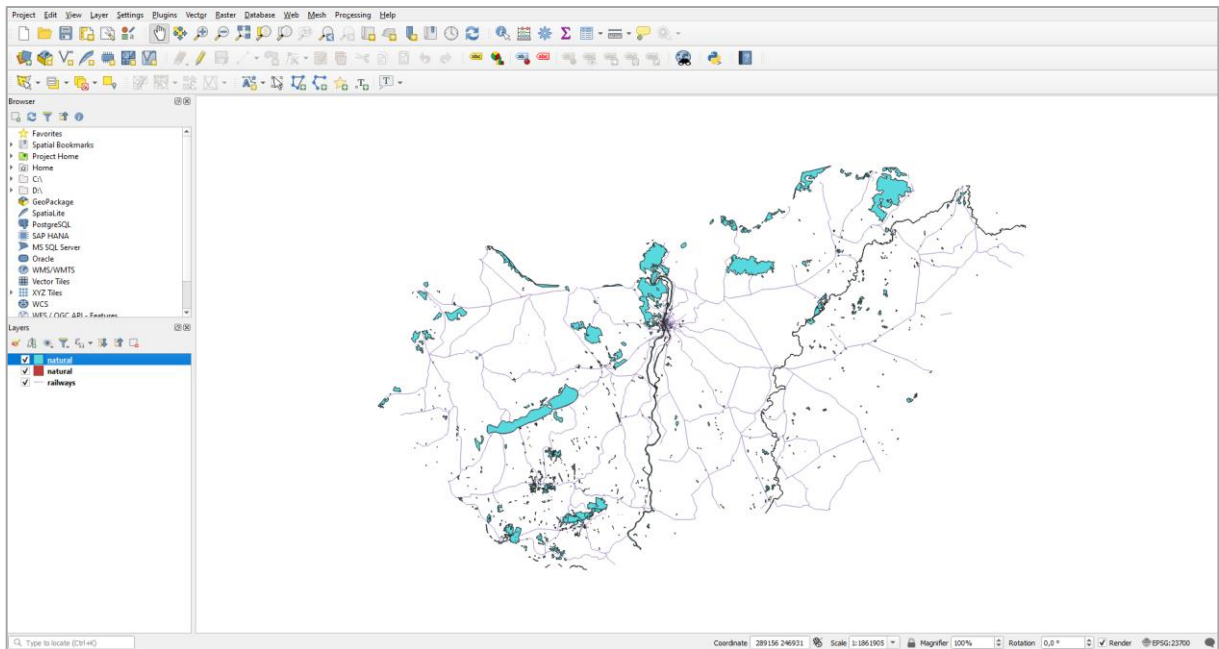
Egy shape-en belül a fenti adattípusokból csak egyféle lehet, tehát point és line egyazon shape-fájlban nem szerepelhet. Éppen emiatt már a fájl létrehozásakor előre meg kell határoznunk, hogy az milyen típusú lesz. A fájl rekordjai (pl. egy-egy pont, vonal) attribútumainak létrehozásakor pedig - egy adatbázis létrehozásához hasonlóan - azok adattípusait kell meghatározni. Ezek az adatbázis-kezelési, illetve programozási előtanulmányok alapján már ismertek (string, integer, number with decimal places, stb.). Az adattípus megválasztása, illetve ismerete a különböző adatbázis-kezelési műveletek elvégezhetőségét döntően befolyásolhatja, pl. string típusú „6”-hoz integer típusú 2-t hozzáadva nem garantált, hogy 8-at fogunk kapni.

Fontos kiemelni, hogy a shape-fájl egy standard formátum, tehát nem csak QGIS-ben, hanem egyéb szoftverekben is megnyitható, szerkeszthető, pl. ArcGIS-ben vagy PTV Visumban is.

## 2. QGIS projekt fájl

A QGIS projekt fájl ezzel szemben a QGIS speciális fájlformátuma: egy a shape-fájlok szerkesztésére szolgáló keret fájl, amiben különböző beállítási konfigurációkat tudunk elmenteni, illetve magukat a shape-eket szerkeszteni. Tehát a tényleges adat mindig a shape-ben van, a projekt fájl tulajdonképpen csak megjelenítésre szolgál, illetve a különböző adatműveletek elvégzését segíti.

A projekt fájl egysége a réteg („layer”). Bármilyen műveletet akarunk elvégezni egy shape-fájlon, azt egy rétegben tároljuk el. Egy rétegben csak egy shape-fájl lehet, viszont egy shape-fájl egyidejűleg több rétegbe is behívható. A rétegek szerepét később mutatjuk be részletesen, az eddigi ismeretek a 6.2. ábrán láthatók. Figyeljük meg a bal oldalon, hogy a „natural” nevű shape-fájlt kétszer is behívtuk, két különböző rétegbe, míg a „railways” nevűt csak egyszer.

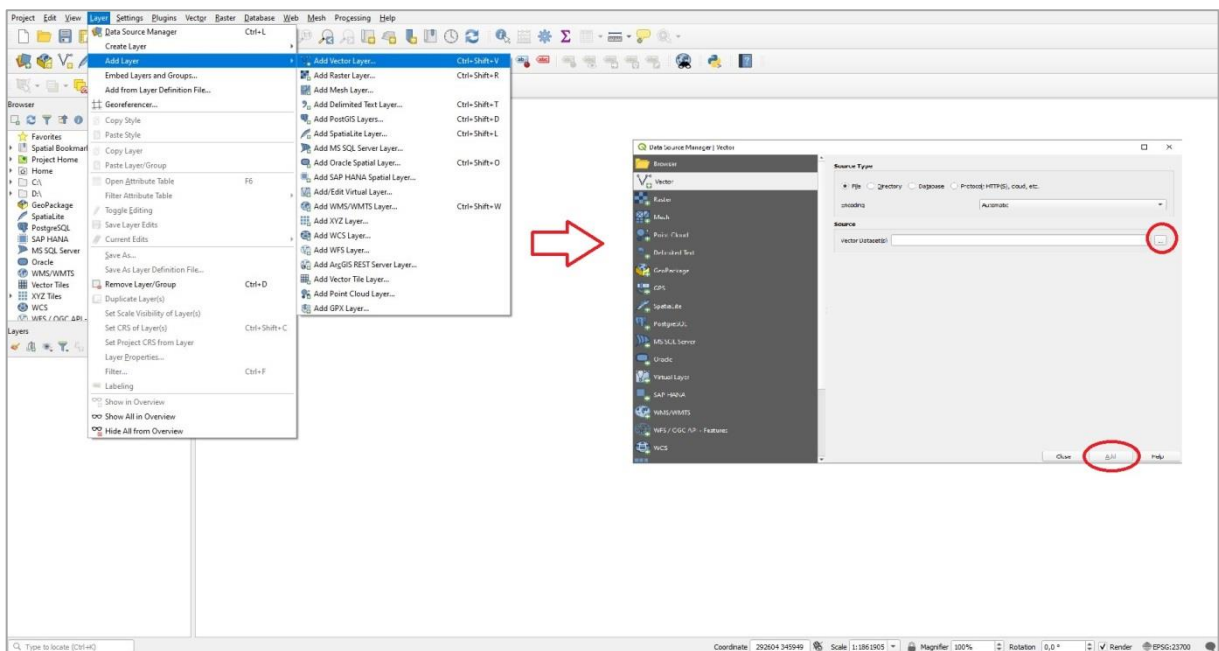


6.2. ábra: Rétegek elhelyezkedése

Az egyes rétegek a shape-fájlok behívásával automatikusan létrejönnek. A shape-fájl behívását az alábbi módon tehetjük meg:

- kattintsunk a „Layer\Add layer\Add Vector Layer” menüpontra
- kattintsunk a „Browse” gombra, majd válasszuk ki shape-fájl fájljai közül az „.shp” kiterjesztésűt
- kattintsuk az „Add” gombra

A leírtakat mutatja a 6.3. ábra.



6.3. ábra: Vektoros réteg hozzáadása

Fontos kiemelni, hogy a réteg nem azonos a shape-fájllal, hiszen előbbi egy kezelési felület a projekten belül, míg utóbbi maga a tényleges adat.

A térinformációs rendszerek alapfogalmainak bemutatása után rátérünk a gyakorlati alkalmazások ismertetésére. Az előbbieken ugyan rögtön az egyes shape-fájlok behívásával kezdtünk, magyarországi térképek esetén ezt megelőzően a koordináta-rendszer beállítását érdemes elvégezni, ezt a folyamatot, illetve az ehhez szükséges minimális elméleti háttérrel mutatja be a következő fejezet.

Annak érdekében, hogy gördülékenyen használhassuk a programot, Windowst használó számítógépeken célszerű a QGIS elindításakor a tálcát valamelyik szélre állítani, különben előfordulhat, hogy egyes menüpontok nem lesznek láthatóak.



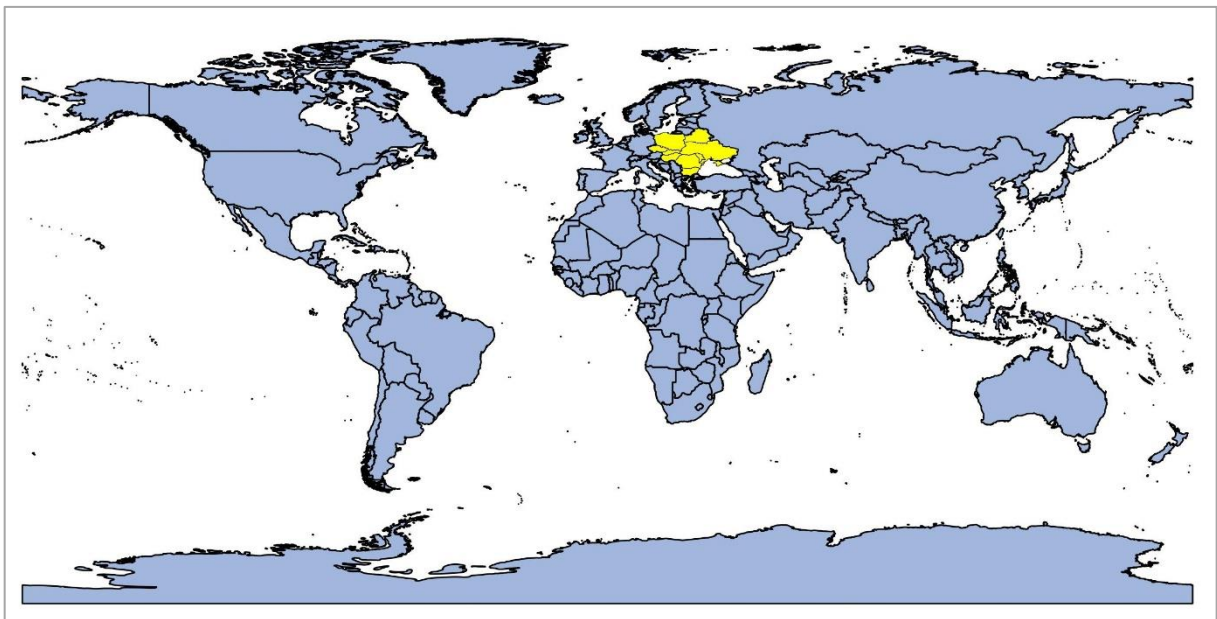
## 2. lecke: Vetületi rendszerek

A lecke a gyakorlati használat szempontjából legfontosabb vetületi rendszereket mutatja be először általánosan, majd pedig a QGIS szoftveres környezetben alkalmazva.

### 1. Vetületi rendszerek

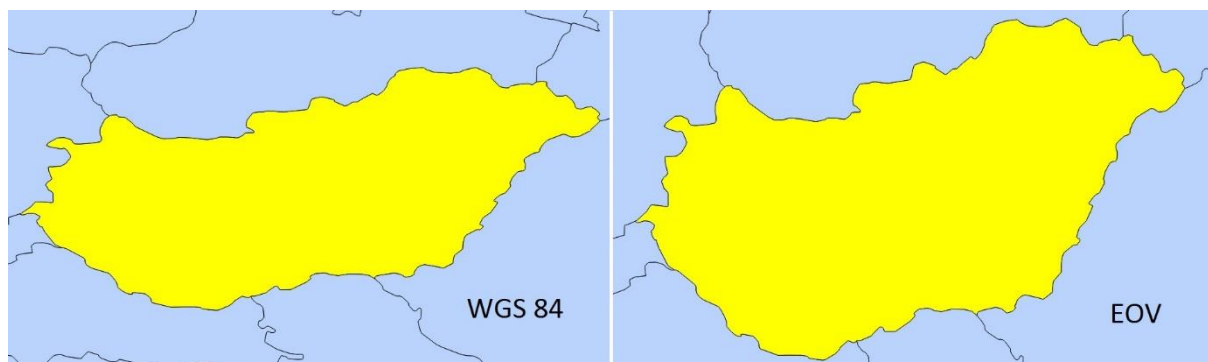
A korábbi tanulmányokból ismert probléma, hogy a Földfelszín felületét le kell képezni valamilyen síkfelületre, ami torzításokkal jár. Ennek mértéke adott vetületi rendszert alkalmazva a Föld egyes pontjain eltérő. Fordított gondolkodással: a Föld adott területét síkfelületre leképezve eltérő mértékű torzulást tapasztalunk az eltérő vetületi rendszerek alkalmazása esetén. (Megjegyzendő, hogy léteznek vetület nélküli rendszerek is, de a QGIS gyakorlatok szempontjából ezek nem relevánsak.)

Az egész világ leképezése során általában a WGS 84 (World Geodetic System - geodéziai világrendszer) vetületi rendszert alkalmazzák, aminek koordináta-rendszere a GPS-mérések koordináta-rendszere is. Ennek eredményét mutatja a 6.4. ábra.



6.4. ábra: A világ WGS 84 vetületi rendszerben

Ennek a vetületi rendszernek a torzítása az Egyenlítőtől távolodva egyre nő, Magyarországon így már elég jelentős. Emiatt, ha például csak Budapestet szeretnénk ábrázolni, célszerű más vetületi rendszert alkalmazni. Az EOVS (Egységes Országos Vetület) vetületi rendszert direkt olyan módon határozták meg, hogy a hazai viszonyokban a lehető legkisebb legyen a torzulás mértéke. Ezt szemlélteti a következő 6.5. ábra, melynek bal oldala WGS 84-ben, jobb oldala EOVS-ben ábrázolja Magyarországot.



**6.5. ábra Magyarország képe különböző vetületi rendszerekben**

A gyakorlatban csak Magyarországot ábrázoló térképek esetén az EOV, az egész világot ábrázoló térképek esetén a WGS 84 szerinti vetületi rendszer alkalmazása ajánlott. Természetesen az országok nagy részének szintén vannak nemzeti vetületi rendszerei, így egy adott ország ábrázolásakor általában a nemzeti vetületi rendszer használata lehet indokolt.

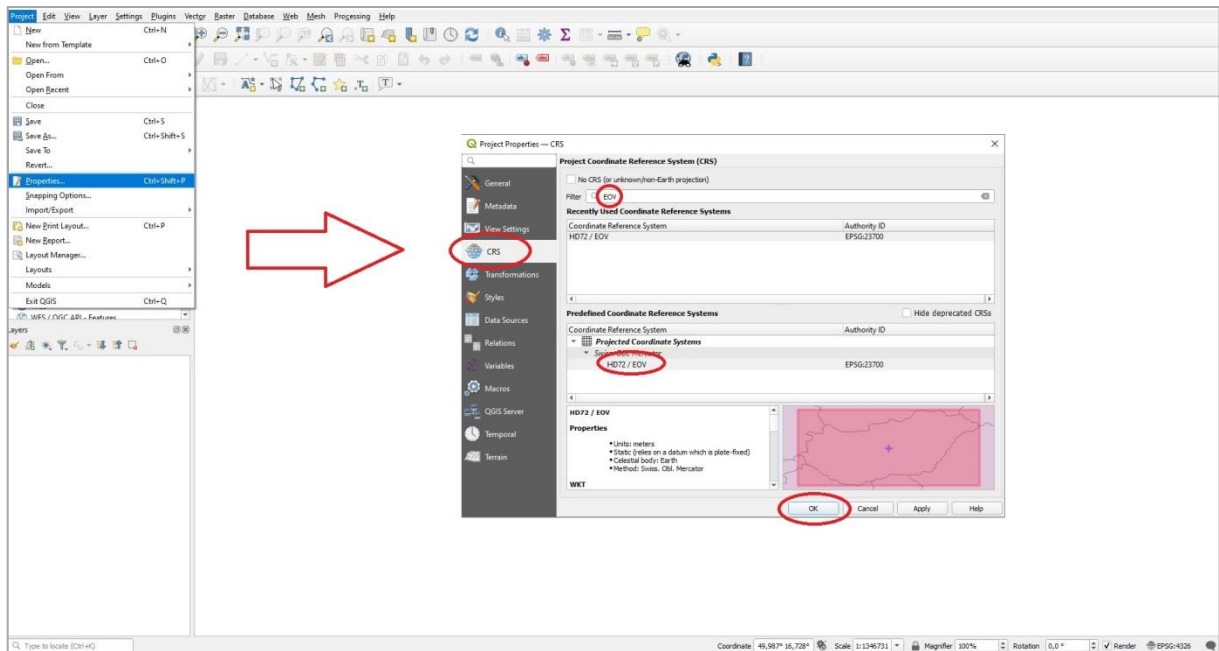
## **2. Koordináta-rendszer beállítása QGIS-ben**

A koordináta-rendszer beállítását két szinten tudjuk megtenni, egyrészt magának a projektnek, másrészt pedig az egyes rétegeknek a szintjén.

A QGIS alapértelmezett koordináta-rendszere a WGS 84. Így ha ebben a koordináta-rendszerben szeretnénk dolgozni, a projektfájl alapbeállítását fogadjuk el. Ellenkező esetben (pl. magyarországi helyszínek esetén), szükséges a WGS 84-ből a kívánt koordináta-rendszerbe történő transzformáció. Ehhez célszerűen új projektfájl létrehozásakor (de akár később is lehet) állítsuk be a project koordináta-rendszerét a „Project\ Properties” menüpont alatt az alábbi lépéseket követve:

- a „CRS” fülön a „Filter” ablakba írjuk be a projektfájl koordináta-rendszerét, ami Magyarországot ábrázoló térképek esetén EOV
- ekkor a középső ablakban megjelenik a HD72/EOV, EPSG:23700 azonosítójú koordináta-rendszer, amit válasszunk ki a bal egérgombbal
- a lap alján kattintsunk az OK gombra

Ezt a folyamatot mutatja a 6.6. ábra.



6.6. ábra: A projekt koordináta-rendszerének beállítása

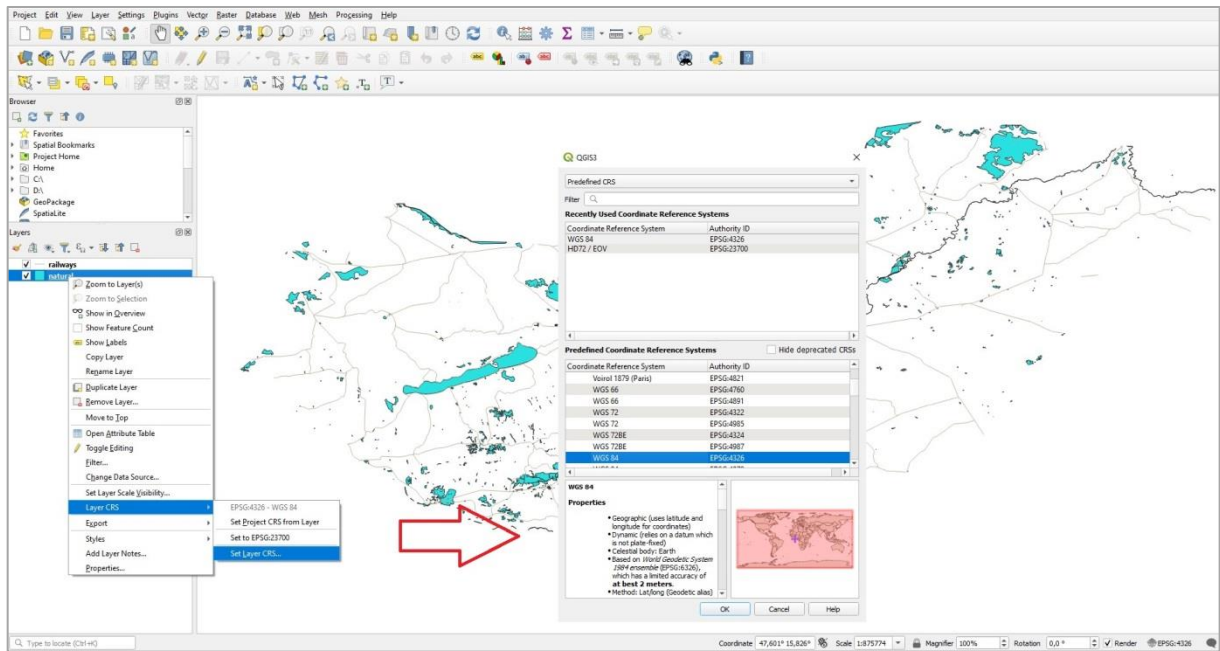
Az előbbi lépéssorozat egy WGS 84-ből EOJ-be történő konverzió. Tehát csak akkor kapjuk a kívánt eredményt, ha a rétegfájljaink WGS 84 koordináta-rendszerben vannak.

Az előző fejezetben szó esett róla, hogy az egyes shape-fájloknak opcionálisan lehet egy „prj” kiterjesztésű fájlja, ami a vetületi rendszerre vonatkozó információt tartalmazza. Amennyiben ilyen van, úgy ha egy shape-fájlt egy rétegbe behívunk, annak vetületi rendszere a „prj” fájllel egyezik. Amennyiben „prj” fájl nem található, a shape behívásakor a QGIS rá fog kérdezni a koordináta-rendszerre.

A rétegfájlok koordináta-rendszerének beállítása módosítható is, az alábbi módon történik:

- jobb egérgombbal kattintsunk az adott rétegre a listában,
- válasszuk ki a „Layer CRS/Set layer CRS” opciót,
- a felugró ablakban a „Predefined Coordinate Reference Systems” listából válasszuk ki a kívánt koordináta-rendszert (pl. WGS 84, EPSG:4326)!

Ezt mutatja a 6.7. ábra.



6.7. ábra: A réteg koordináta-rendszerének beállítása

#### Gyakorló feladatok:

- Internetről letöltött (pl.: <http://www.mapcruzin.com/free-hungary-arcgis-maps-shapefiles.htm>) tetszőleges, Magyarországot ábrázoló shape-fájlt hívjunk be QGIS-be, és állítsuk a réteg koordináta-rendszerét WGS 84-be (EPSG: 4326). Mentjük a projektfájlt, majd állítsuk át a koordináta-rendszerét HD72/EOV, EPSG:23700 -re!
- Állítsuk át a shape-et tartalmazó réteg koordináta-rendszerét is EOV-re! Összegezzük a látottakat! Mi okozza a változást?

### 3. lecke: Vektoros rétegek alapvető műveletei, jellemzői

Ebben a leckében a rétegekkel kapcsolatos alapismereteket mutatjuk be, a megjelenítési és kijelölési lehetőségeket középpontba helyezve.

Ahogy arról már szó esett, a réteg az a kezelési felület, amin keresztül QGIS-ben (és általában a térinformációs szoftverekben) szerkeszteni tudjuk a különböző - esetünkben shape - fájlokat. Tehát mindig, minden réteg mögött van egy shape-fájl, ami az adatokat tartalmazza. Ilyen módon, ha egy rétegből pár elemet kitörölünk, akkor valójában a shape-fájlból távolítjuk el az elemeket. Viszont ha csak a megjelenítési beállításokon változtatunk, például a réteget sárga helyett zöld színnel jelöljük, az nem érinti az adattartalmat, ilyen módon a shape-fájl elemeit sem.

#### 1. Rétegek létrehozása

Vektoros réteg létrehozására két alapvető lehetőségünk van: vagy egy már meglévő shape-fájlt hívunk be, és ezzel létrejön a réteg, vagy pedig „üresen” hozunk létre egyet, és annak tartalmát mi alakítjuk ki. Mivel minden réteg mögött van egy shape-fájl, utóbbi esetben a réteg mögé létrehozunk egy üres shape-et is, és azt kezdjük el szerkeszteni.

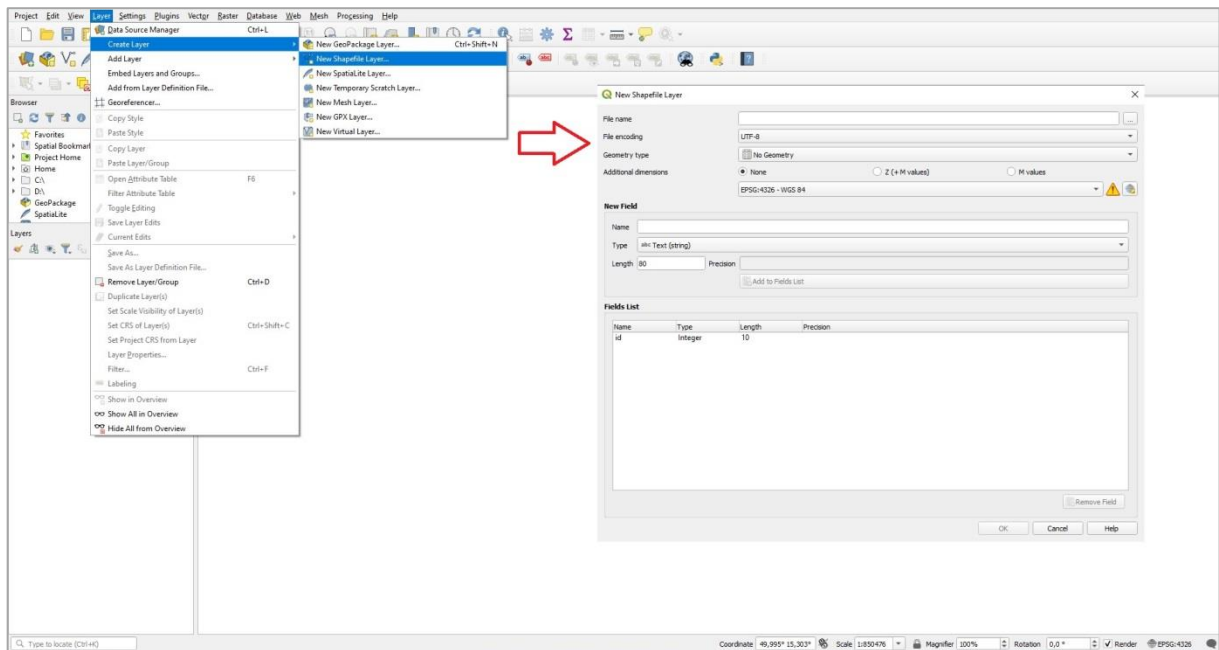
Meglévő shape-fájlból réteg létrehozására az első leckében már láttunk példát, így most az új rétegek és hozzájuk tartozóan új shape-fájlok kialakítási lépéseit mutatjuk be. Mivel a shape-fájl egy adatbázis is, érdemes az elején létrehozni olyan attribútumokat, melyekről tudjuk, hogy a későbbiekben szükségünk lehet rájuk. A lépések az alábbiak:

- kattintsunk a „Layer\Create Layer\New Shapefile Layer” menüpontra;
- a „File name” ablakban tudjuk megadni a fájl nevét, az ablak mellett található „...” gombra kattintva tudjuk megadni a fájl mentési helyét;
- a „File encoding” pont alatt az adatbázisunk szöveges részeinek kódolási rendszerét választhatjuk ki; érdemes az alapbeállítást meghagyni, esetleg az UTF-8-at kiválasztani;
- válasszuk ki a létrehozni kívánt shape-fájl (és ezzel együtt a réteg) fajtáját a „Geometry type” legördülő listából;
- az „Additional dimensions” alatti legördülő listából a koordináta-rendszer beállításánál érdemes az EPSG:4326 - WGS 84-et választani, amit szükség esetén a projektfájl koordináta-transzformációjával EOVB-be tudunk konvertálni az előző fejezetben bemutatott módon (megjegyzés: ha egy réteg koordináta-rendszere EOVB, és ezen végezzük el a projektfájlon belül a WGS 84-EOVB transzformációt, akkor az eredmény már nem EOVB lesz!);
- a „New Field” egységben tudjuk az adatbázistáblánk mezőit felvenni:
  - az attribútum nevébe („Name”) lehetőleg ne írjunk ékezetes betűket és szóközt (bár a QGIS engedi);
  - a típust („Type”) úgy válasszuk meg, hogy az összhangban legyen a bevinni kívánt információ tartalommal;
  - a „Length” menüpontban azt adhatjuk meg, hogy a mezőbe maximum hány karaktert lehessen beírni (tizedes törtek esetén a tizedes vessző 0 db

karakternek számít, a tizedes jegyek számát a „Precision” mezőbe lehet beírni);

- az „Add to Fields List” gombbal tudjuk az attribútumot hozzáadni az adattáblához;
- amennyiben egy attribútumot el akarunk távolítani, úgy az „Fields List” listában kattintsunk erre az elemre a bal egérgombbal, majd nyomjuk meg a „Remove Field” gombot;
- kattintsunk az OK gombra!

A lépéseket a 6.8. ábra mutatja.



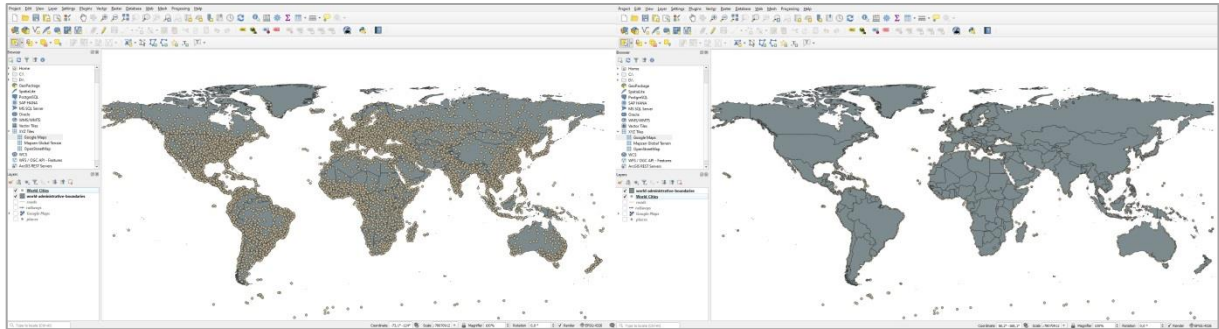
6.8. ábra: Új réteg létrehozása

- ezután a felugró ablakban mentsük el a shape-fájlt;
- a bal oldalon, a rétegek listájában meg fog jelenni a réteg.

## 2. Rétegek megjelenítése, csoportosítása

Ahogy arról már szó esett, egy projektfájlba több réteg is behívható. A rétegek mindig valamilyen sorrendben egymáson helyezkednek el, a bal oldali „Layers” listában föntebb lévő kitakarja a lentebb lévő. A réteg a bal egérgomb lenyomva tartása mellett a listában föntebb, vagy lentebb húzható. Ezt a folyamatot illusztrálja az 6.9. ábra: a bal oldalon a városok rétege az országok rétege alatt van, így az ország poligonja kitakarja őket. A jobb oldalon a sorrend fordított, így a városok látszanak.



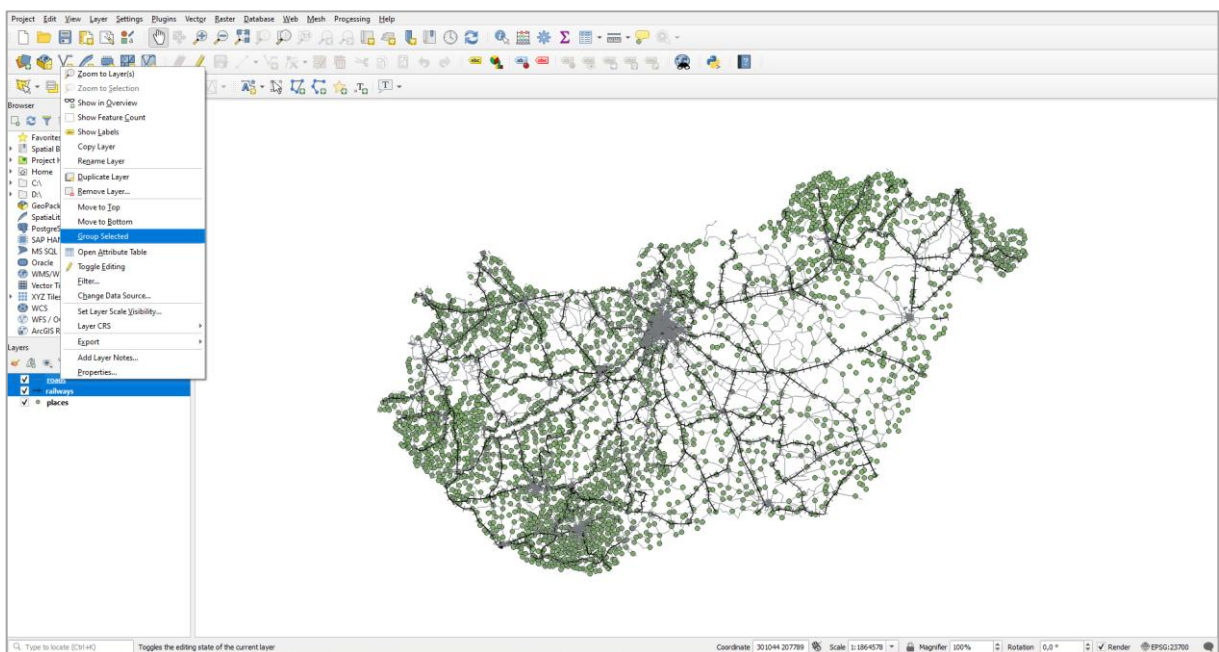


**6.9. ábra: Rétegek megjelenítése különböző sorrendben**

Amennyiben egy rétegre éppen nincsen szükségünk, annak megjelenítését ki tudjuk kapcsolni a „Layers” listában a réteg neve melletti jelölőnégyzetre kattintva.

Gyakran előfordul, hogy több réteget együttesen szeretnénk kezelni, például, ha bizonyos rétegeket csak háttérnek hívunk be, legtöbbször a tájékozódás segítése miatt. Erre nyújt hasznos lehetőséget a rétegek csoportosításának lehetősége. Az alábbi példában a „lényeg” a magyarországi nagyvárosok ábrázolása, amihez háttérként behívtuk főbb a magyar utakat és vasutakat. Utóbbi kettőből képeztünk csoportot az alábbi módon (lásd a 6.10. ábrát):

- a Shift billentyűt lenyomva bal egérgombbal kijelöljük a csoportosítani kívánt rétegeket, majd közülük valamelyikre a jobb egérgombbal kattintva kiválasztjuk a „Group Selected” opciót;
- innentől a csoportba sorolt rétegek bizonyos funkciói együttesen kezelhetők, pl. a listán belül föntebb/lentebb mozgathatók, vagy ki- és bekapcsolhatók.



**6.10. ábra: Rétegek csoportosítása**

Egy adott rétegre jobb egérgombbal kattintva különböző egyéb opciókból is választhatunk, ebből emelünk ki néhány gyakran alkalmazottat.

- „Remove”: ezzel a réteg eltávolítható a projektfájlból;

- „Rename”: ezzel a réteg átnevezhető, ami a könnyebb eligazodást szolgálja sok réteg esetén; itt már ékezetes karakterek és szóköz is alkalmazható;
- „Properties”: itt a legkülönbözőbb beállítások tehetők meg, például a vonalak vastagsága, színek, címkék, hogy csak a legalapvetőbbeket említsük, a későbbiekben még részletesebben is bemutatjuk;
- „Open attribute table”: itt lehet az egyes shape-fájlok adattartalmát módosítani, ami a gyakorlati alkalmazások egyik központi része; a későbbiekben még részletesebben is bemutatjuk.

### 3. Kijelölési lehetőségek

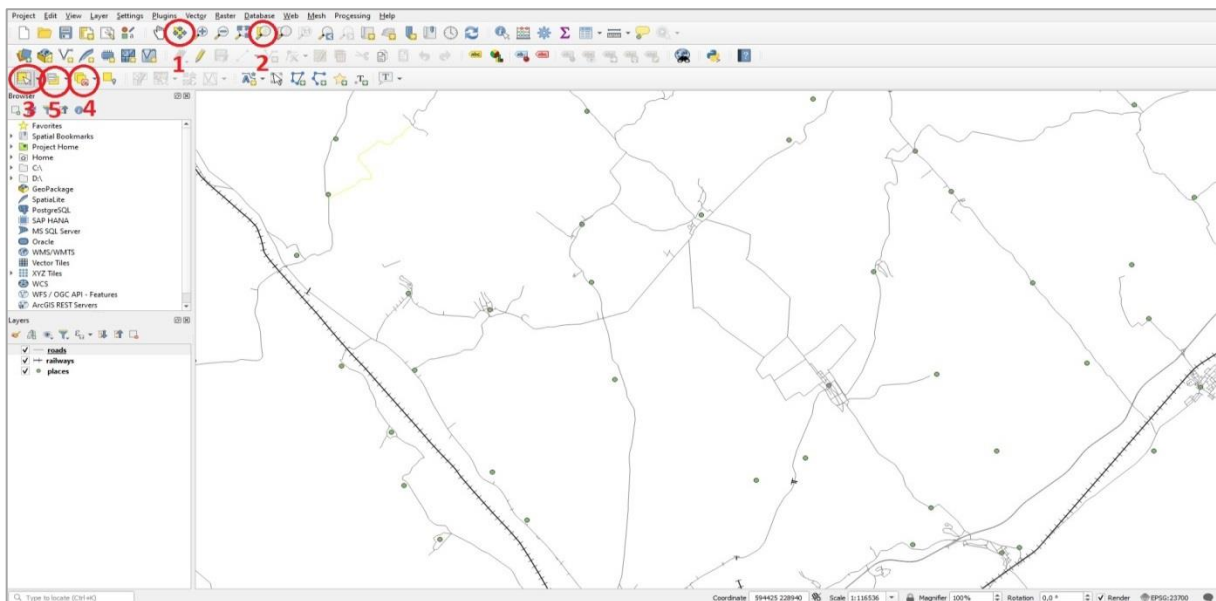
A térinformációs rendszerek meghatározásából kiindulva kétféle alapvető lehetőségünk van az elemek közül történő kiválasztásra:

- térbeli kijelölés,
- vagy: attribútumok szerinti kijelölés lekérdezés alapján.

A térbeli kijelölésekhez és általában a QGIS használatához elengedhetetlen a nézetek változtatásának ismerete. A nézet húzását a görgő benyomásával, vagy a klasszikus pásztázó ikonra kattintás után a bal egérgomb nyomva tartásával és húzásával tudjuk megtenni. A méretarány kicsinyítését és nagyítását az egyik, illetve másik irányba történő görgetéssel tudjuk változtatni, vagy az ikonsorban a nagyító, illetve kicsinyítő ikonokra kattintás után a képernyőre téglalapot rajzolva. Ehhez a bal egérgomb folyamatos nyomva tartása mellett kell kijelölnünk a téglalap átlóját.

Általánosan igaz, és nem csak a kijelölések esetén, hogy a bal oldali „Layers” listából mindig ki kell jelölnünk azt a réteget, amelyikkel éppen dolgozni szeretnénk.

Ökölszabályként elmondható, hogy a QGIS ikonjai közül azok állnak kapcsolatban a kiválasztott elemekkel vagy a kiválasztás módjával, amelyekben egy sárga téglalapot látunk szaggatott szürke kerettel.



6.11. ábra: Kijelölési lehetőségek



A 6.11. ábrán láthatók az egyes kijelölési lehetőségek. Az ábra számokkal jelölt pontjaihoz kapcsolódó legfontosabb tudnivalók:

1. „Pan Map to Selection”: a képernyőt a kijelölt terület fölé húzza.
2. „Zoom to Selection”: a képernyőt a kijelölt területre nagyítja.
3. A térbeli kijelölések ebben a pontban hajthatók végre (a kijelölendő réteg bal oldali listából történő kiválasztása után):
  - a. „Select Feature(s)”: egy elem kiválasztására a bal egérgomb lenyomásával. A shift billentyű lenyomva tartásával több elem is kiválasztható.
  - b. „Select Features by polygon”: sokszög alakú kijelölés, a bal egérgomb sorozatos lenyomásával tudjuk a sokszög nagyságát és alakját meghatározni. Az utolsó pont megadása után a jobb egérgombot kell egyszer lenyomni.
  - c. „Select Features by Freehand”: szabadkézi kijelölés, ahol az első pontot a bal egérgomb lenyomásával adjuk meg, majd felengedés után a kurzort végigvezetjük a kívánt íven. A minta megrajzolása után ismét benyomjuk a bal egérgombot.
  - d. „Select Features by Radius”: kör alakú kijelölés, a bal egérgombot lenyomva kijelöljük a kör középpontját. A gomb felengedése után a kurzorral húzzuk meg a kör sugarát, végül ismét megnyomjuk a bal egérgombot..
4. Kijelölések megszüntetése
  - a. „Deselect Features from All Layers”: a kijelölések megszüntetése az összes rétegen.
  - b. „Deselect Features from the Current Active Layer”: a kijelölések megszüntetése az aktuális rétegen.
5. Attribútumok értéke alapján lehet kijelöléseket végrehajtani ebben a pontban
  - a. „Select Features by Value”: attribútumok konkrét értéke alapján lehet szűrni (pl. kisebb, nagyobb, egyenlő).
  - b. „Select Features by Expression”: az attribútumok alapján összetettebb kijelölések hajthatók végre ebben a pontban. A kijelöléshez az SQL nyelv minimális ismerete szükséges. Erről a lehetőségről a későbbiekben még részletesebben is szó esik.
  - c. „Select All Features”: az összes elem kijelölése a rétegen.
  - d. „Invert Feature Selection”: a rétegen kijelölt elemek és nem kijelölt elemek között a kijelölés felcserélése.

#### Gyakorló feladatok:

- Hozzunk létre négy új shape-fájlt és hozzájuk tartozó réteget az alábbiak szerint:
  - Név: NODE, típus: point, kötelező attribútum: AZONOSITO, CENTROID (ennek értéke 0 vagy 1 lesz a későbbiekben) egyéb attribútum tetszőlegesen hozzáadható
  - Név: LINK, típus: line, kötelező attribútumok: AZONOSITO, TIPUSKOD SEBESSEG, HOSSZ, egyéb attribútum tetszőlegesen hozzáadható
  - Név: CONNECTOR, típus: line, kötelező attribútumok: AZONOSITO, HOSSZ, egyéb attribútum tetszőlegesen hozzáadható
  - Név: ZONE, típus: polygon, kötelező attribútumok: AZONOSITO, LAKOSSZAM, egyéb attribútum tetszőlegesen hozzáadható

- Ügyeljünk a megfelelő adattípusok megválasztására és az adathosszakra! Gondoljuk át, hová várunk szöveges, hová számértékeket, illetve reálisan előfordulhat-e tizedes érték az adott mezőben!
- Képezzünk az azonos típusú két shape-fájl rétegéből egy csoportot!
- Nevezzük át az egyes rétegeket Csomópont, Szakasz, Konnektor és Zóna nevekre!
- Állítsuk be a rétegek sorrendjét az objektumok várható térbeli kiterjedése szerint úgy, hogy majd az adatok feltöltése után az egyes rétegek közül egyik se legyen teljesen kitakarva!
- Kapcsoljuk ki a létrehozott csoport láthatóságát!
- Mentsük el a munkát, mert a következő fejezetekben ezzel fogunk dolgozni.

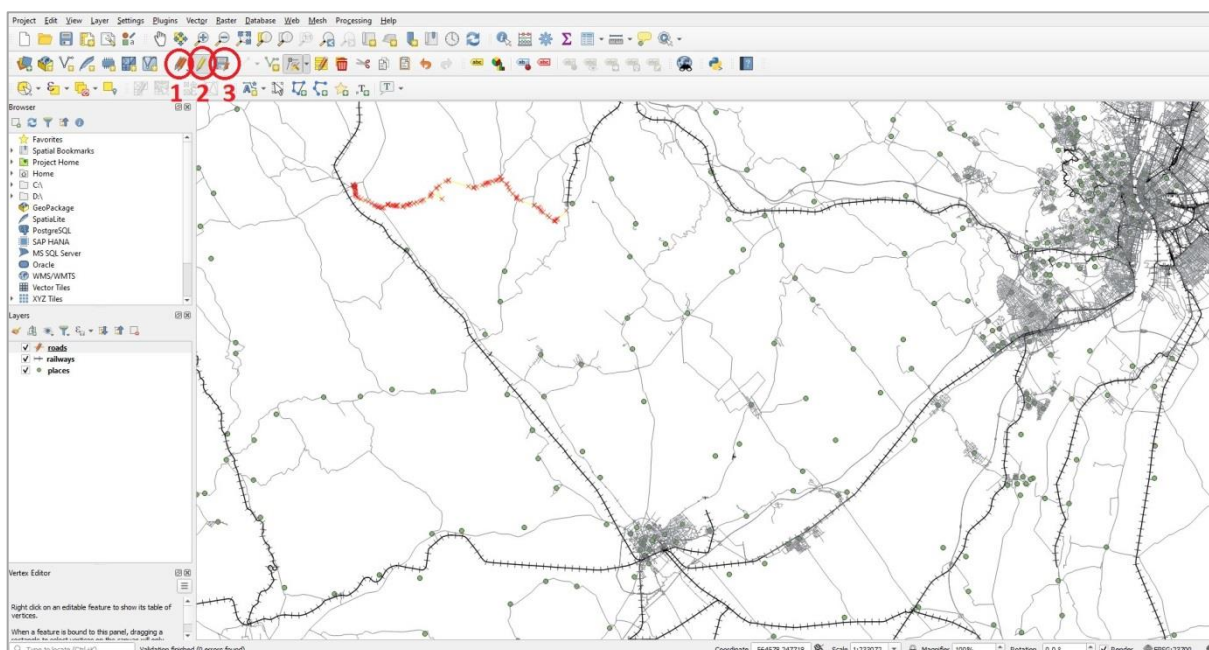
## 4. lecke: Különböző típusú shape-fájlok szerkesztése

Ebben a fejezetben bemutatjuk, hogyan lehet az egyes különböző típusú shape-fájlokat szerkeszteni, új elemeket felvenni, illetve a meglévők térbeli kiterjedését és attribútumaik adattartalmát módosítani.

### 1. A szerkesztő mód

Bármilyen szerkesztési, módosítási művelet végrehajtása előtt szükséges a szerkeszteni kívánt réteg kiválasztása a bal oldali listából, majd a „Toggle Editing” - szerkesztő mód bekapcsolása. Egyszerre több réteg is szerkesztő módba kapcsolható. A módosításokat a „Save for Selected/All Layer(s)” gombokkal tudjuk elmenteni. A hibásnak vélt módosítást a „Current Edits” menüpont alatt a „Rollback” paranccsal tudjuk visszavonni, akár csak a kijelölt rétegekre („Rollback for Selected Layers”), akár az összes rétegre („Rollback for All Layers”) vonatkozóan is. Az egyes említett menüpontok elhelyezkedése látszik a 6.12. ábrán.

1. „Current Edits” (és alatta „Rollback”)
2. „Toggle Editing”
3. „Save Layer Edits”



6.12. ábra: Szerkesztési műveletek

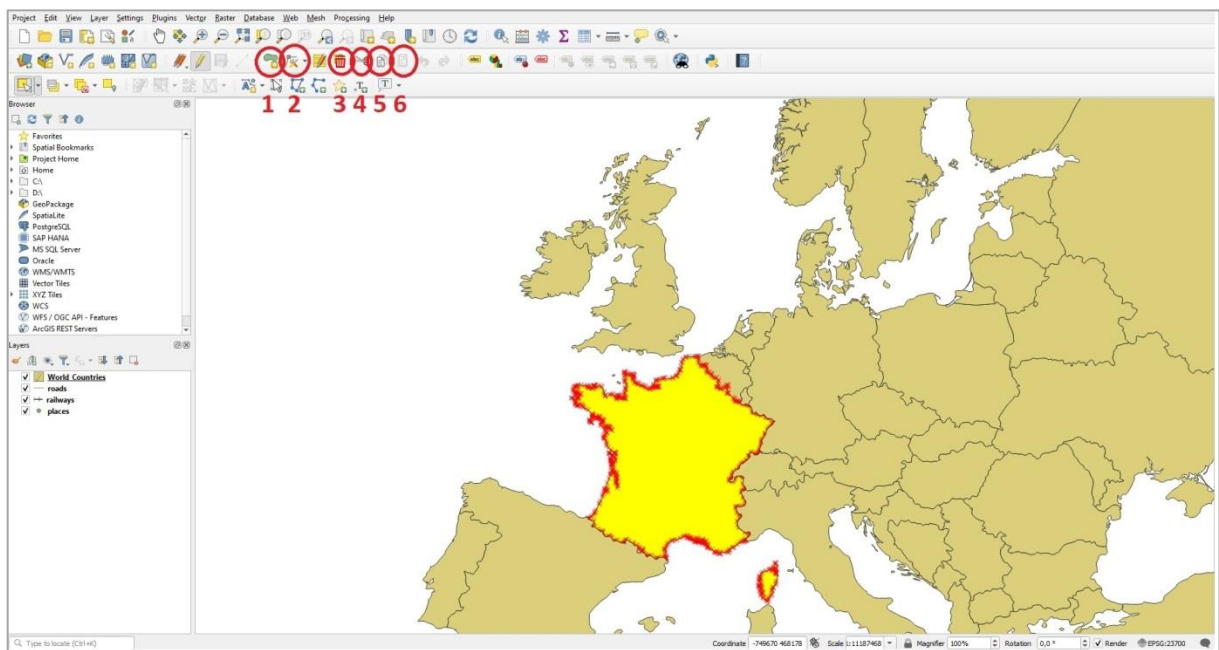
A szerkesztési műveletek befejezése után érdemes a „Toggle Editing” gombot ismételten megnyomni, ezzel kilépünk a szerkesztő módból.

## 2. Térbeli módosítások

Az új elem létrehozásának, illetve egy már meglévő alakja módosításának tudnivalói a shape-fájl típusától függenek, így ezeket az ismereteket külön-külön tekintjük át az egyes fájljajták alapján.

A 6.13. ábra szemlélteti a kijelölt rétegen a térbeli módosítások során használt ikonokat, melyek jelentései a következők:

1. „Add Feature” - új elem létrehozása
2. „Vertex Tool” - pontok pozíciójának változtatása, melyek lehetnek point shape alkotóelemei, vonalas shape elemeinek végpontjai vagy sokszög shape elemeinek csúcsai (tehát ezzel a funkcióval az objektumok térbeli kiterjedését változtathatjuk meg)
3. „Delete Selected” - a kijelölt elem törlése (a kijelölés mikéntjéről lásd az előző fejezetet)
4. „Cut Features” - adott elem kivágása a shape-fájlból
5. „Copy Features” - adott elem másolása
6. „Paste Features” - adott elem beillesztése, melynél az alábbiakat érdemes betartani:
  - a. Egyezzen a kivágás/másolás rétegtípusa a beillesztés rétegtípusával. (A QGIS ellenkező esetben átkonvertálja a beillesztendő elemet a célréteg típusává, ami torzulásokat eredményez.)
  - b. A beillesztés réteget kapcsoljuk szerkesztő módba (a „Toggle Editing” gombbal).



6.13. ábra: Térbeli módosítási lehetőségek

A következőkben leírtakra általánosan igaz, hogy a szerkesztést megelőzően a bal oldali listából válasszuk ki a szerkeszteni kívánt réteget, majd aktiváljuk a szerkesztő módot (a „Toggle Editing” gombbal).

## Pontok kezelése

- Létrehozás: az „Add Feature” gomb megnyomása után bal egérgombbal a kívánt helyre kattintunk. Az ezután felugró ablakban megadhatjuk a shape-fájl egyes attribútumainak értékét.

## Vonalak kezelése

- Létrehozás: a QGIS-ben töröttvonalakat tudunk létrehozni. Adott sorrendben csúcspontokat jelölünk ki, ahol két egymást követő csúcspontot egy-egy egyenes szakasz köt össze. Az „Add Feature gomb” megnyomása után bal egérgombbal kijelöljük a töröttvonal csúcspontjait, majd miután az összeset kijelöltük, az ablak tetszőleges helyére kattintunk egyet a jobb egérgombbal. A csúcspontokat vertexnek nevezik.
- Töröttvonal alakjának megváltoztatása: a „Vertex Tool” gomb megnyomása után bal egérgombbal rákattintunk a módosítani kívánt elemre (ekkor a csúcspontok színe pirosra vált), majd a mozgatni kívánt csúcspontot a kurzorral tudjuk mozgatni.
- Új csúcspont felvétele: a „Vertex Tool” gomb megnyomása után kattintsunk bal egérgombbal duplán a vertex létrehozási helyére, majd kattintsunk még egyet.
- Csúcspont törlése: a „Vertex Tool” gomb megnyomása után kattintsunk bal egérgombbal a töröttvonalra (ekkor a csúcspontok színe pirosra vált), melyről a csúcspontot törölni szeretnénk, majd a bal egérgomb lenyomva tartásával és a kurzor húzásával jelöljük ki a törölni kívánt csúcspontot vagy csúcspontokat (ekkor ezek a csúcspontok kékre válnak), majd nyomjuk le a Delete billentyűt.

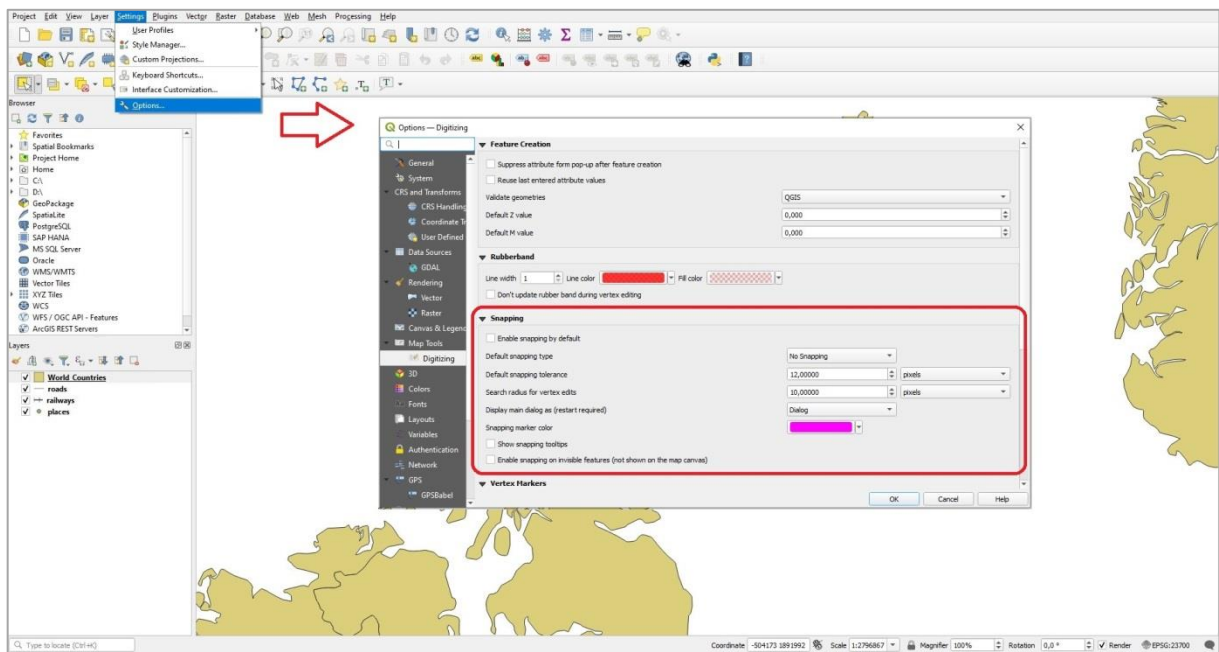
## Sokszögek kezelése

- Létrehozás: az „Add Feature” gomb megnyomása után bal egérgombbal kijelöljük a sokszög csúcspontjait, majd az ablak tetszőleges helyére kattintunk egyet a jobb egérgombbal.
- Az egyéb funkciók mindenben egyeznek a vonalas shape-fájloknál leírtakkal.

## 3. A „Snapping” funkció

Új elemek létrehozásánál, vagy már meglévők módosításánál szükségünk lehet arra, hogy szomszédos elemek pontosan illeszkedjenek egymáshoz, ezt segíti elő a „Snapping” funkció. Ekkor az újonnan létrehozandó elem csúcspontját az illesztendő elem csúcspontja vagy éle magához vonzza. A funkció rétegek között is érvényes, nem csak egy rétegen belül.

A funkciót a „Settings\Options” gombra kattintva, majd a Map Tools/Digitizing opciót választva tudjuk testre szabni. Ezt mutatja a 6.14. ábra.



6.14. ábra: A „Snapping” funkció

Itt tudjuk beállítani, hogy a rétegeken található elemek csúcsa (vertex), éle (segment), vagy egyéb nevezets pontja (pl. centroidja) vonzza-e magához a kurzort, miközben rétegeket szerkesztünk. A „Tolerance” alatt pedig a vonzás távolsága állítható be, minél nagyobb a számérték, annál távolabbról érezhető már a vonzó hatás.

#### 4. Attribútumok és mezők módosítása

Akár attribútumot kívánunk módosítani, akár új mezőt létrehozni vagy törölni, mindig aktiváljuk előtte a már bemutatott szerkesztő funkciót („Toggle Editing”).

Az egyes rekordok attribútumait, illetve a mezőket az „Attribute Table”-ben tudjuk szerkeszteni. Ehhez jelöljük ki a szerkeszteni kívánt shape-fájlt a bal oldali listából, majd a jobb egérgomb lenyomása után válasszuk az „Open Attribute Table” lehetőséget.

Adatmódosítás esetén a megfelelő cellába kattintsunk a bal egérgombbal, majd írjuk át az értéket.

Új mezőt a „New Field” (1) gombot megnyomva tudunk létrehozni. A mező típusának és egyéb jellemzőinek megadásakor a korábban leírtakat tartjuk szem előtt.

Már meglévő mező törléséhez először nyomjuk meg a „Delete Field” (2) gombot, majd ezt követően a felugró ablakból válasszuk ki a törölni kívánt mezőt.

Amennyiben egy mező értékét több rekordra vonatkozóan, adott szabály szerint akarjuk módosítani, célszerű a „Field calculator” (3) funkció alkalmazása, amivel ezeket a műveleteket el tudjuk végezni. Mindehhez aktiváljuk a szerkesztő módot!

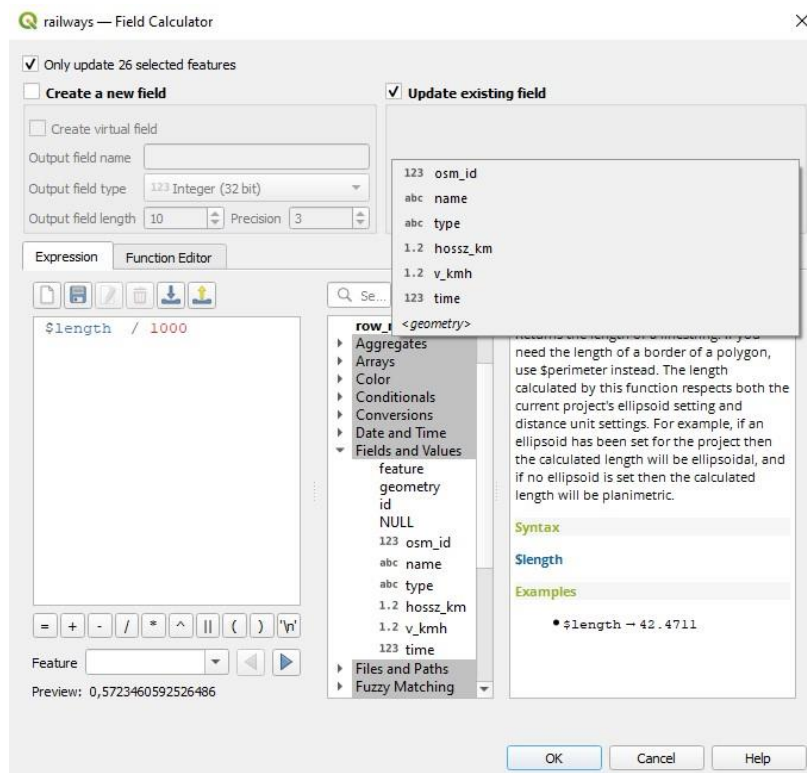
Az előbb bemutatott funkciók gombjait mutatja a 6.15. ábra.



	osm_id	name	type	hossz_km
1	4903395	NULL	tram	NULL
2	11013624	NULL	tram	NULL
3	14513814	NULL	tram	NULL
4	14514022	NULL	tram	NULL
5	14514150	NULL	tram	NULL
6	24020799	NULL	tram	NULL
7	24022004	NULL	tram	NULL
8	24052436	NULL	tram	NULL
9	24054341	NULL	tram	NULL
10	24210197	NULL	tram	NULL
11	24252292	NULL	tram	NULL
12	24262169	NULL	tram	NULL
13	24268440	NULL	tram	NULL
14	24268442	NULL	tram	NULL
15	24268457	NULL	tram	NULL

6.15. ábra: Az attribútumtábla mezőinek módosítási lehetőségei

A „Field calculator” felépítése látható a 6.16. ábrán. A mintában egy vonalas shape-fájl elemeinek hosszát számoljuk ki km-ben.



6.16. ábra: A „Field calculator”

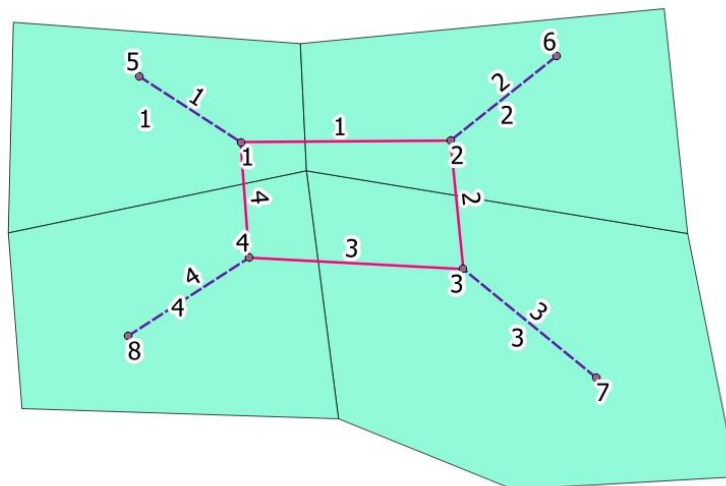
A legfelső sorban lévő jelölőnégyzet aktiválásával a művelet csak a kijelölt rekordokon hajtódik végre, egyébként pedig az összesen. A második blokkban a „Create a new field” opciót választva az eredmény számára új mezőt hozhatunk létre (a korábban már bemutatottak szerint), míg a jobb oldali „Update existing field” opció bejelölése esetén az eredménnyel egy már meglévő mező tartalmát írjuk felül. A példában ezt tesszük: a hossz\_km mezőbe fogjuk kérni az eredményt. A harmadik, „Function list” nevű blokkban állítható be, hogy mi alapján legyen kiszámolva a mező értéke. Hossz számítása esetén a „Geometry” fül alatt a „\$length” (hossz) értékre kattintsunk duplán. A hosszt a rendszer mindig a koordináta-rendszer egységében számolja, ami esetünkben méter. Ezért az alsó, „Expression” nevű blokkban kiegészítjük az értéket manuálisan „\$length/1000”-re. A legelső sorban az „Output preview” felirat utáni értékből láthatjuk a művelet eredményét.

Érdemes kiemelni, hogy a 6.16. ábrán szereplő középső blokkban lehetőségünk van a „Fields and Values” pontban a mezőneveket is kiválasztani, és ezáltal az „Expression” ablakba beírni. Ilyen eshetőségre példa, ha eljutási időt akarunk számolni órában. Ekkor az alábbi lépéseket célszerű megtenni:

- Mivel a mezők között nincs olyan, aminek neve az eljutási időre utal, ezért az első blokkban a „Create a new field” opciót választjuk, vagyis az eredmény számára új mezőt hozunk létre. Kitöltjük a mező kívánt nevével és egyéb jellemzőivel.
- A „Function list” blokkból a „Fields and Values” pontból kiválasztjuk a hossz\_km mezőt, majd az „Expression” ablakba beírunk egy „/” osztásjelet, majd ismét a „Fields and Values” pontból kiválasztjuk a v\_kmh (sebesség km/h-ban) mezőt. Így a képlet: „hossz\_km” / „v\_kmh”.

#### Gyakorló feladatok:

- Az előző lecke végén elmentett gyakorlófeladatot nyissuk meg, és vegyünk fel elemeket az egyes rétegekre. Az alábbi állapotig jussunk el (az egyes rétegek színeivel és a feliratokkal még ne foglalkozzunk, hagyjuk meg a QGIS által sorsolt színeket):



6.17. ábra: Az elkészítendő feladat kódolása

- Node: a 8 db pont
- Link: a 4 db folyamatos rózsaszín vonal
- Connector: a 4 db szaggatott lila vonal



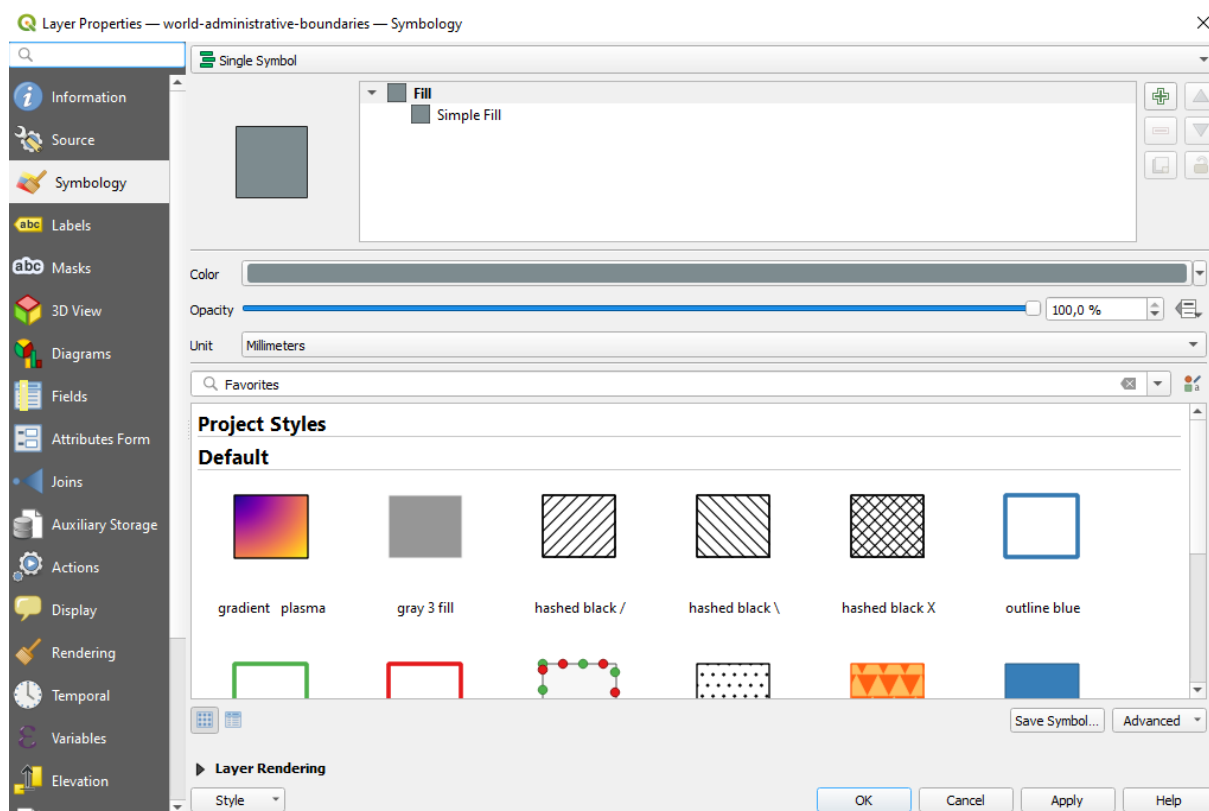
- Zone: a 4 db sokszög
- Minden esetben, tehát rétegeken belül és rétegek között is pontos illeszkedést állítsunk be, tehát ne csak látszólag, hanem valójában is egy pontban találkozzanak ez elemek!
- Töltsük fel az attribútumtáblák AZONOSITO mezőit a 6.17. ábrán látható számozás szerint!
- „Field calculator” segítségével számítsuk ki a vonalas shape-ek szakaszainak hosszait!
- A ZONE shape-ben a LAKOSSZAM legyen az AZONOSITO ezerszerese, „Field calculator”-ral számoljunk!
- Másoljunk, majd illesszünk be elemet ellentétes rétegtípusba (pl polygont node-ba), és vizsgáljuk meg az eredményt!
- Mentsük el a munkát, mert a következő fejezetekben ezzel fogunk dolgozni!

## 5. lecke: Rétegek megjelenítési beállításai

A leckében a rétegek megjelenítésével kapcsolatos alapvető tudnivalókat mutatjuk be. Ismertetjük a színek beállításának lehetőségét, illetve azok lehetséges kapcsolatát a rétegek adattartalmával. Ezenfelül kitérünk a címkézésre és a különböző diagramok felvételének módjára is.

### 1. Rétegek színének beállítása

A rétegek színének beállításához kattintsunk a bal oldalon a réteg nevére jobb egérgombbal, majd bal egérgombbal a „Properties” menüpontra, ezután a felugró ablakban a „Symbology” fülre.

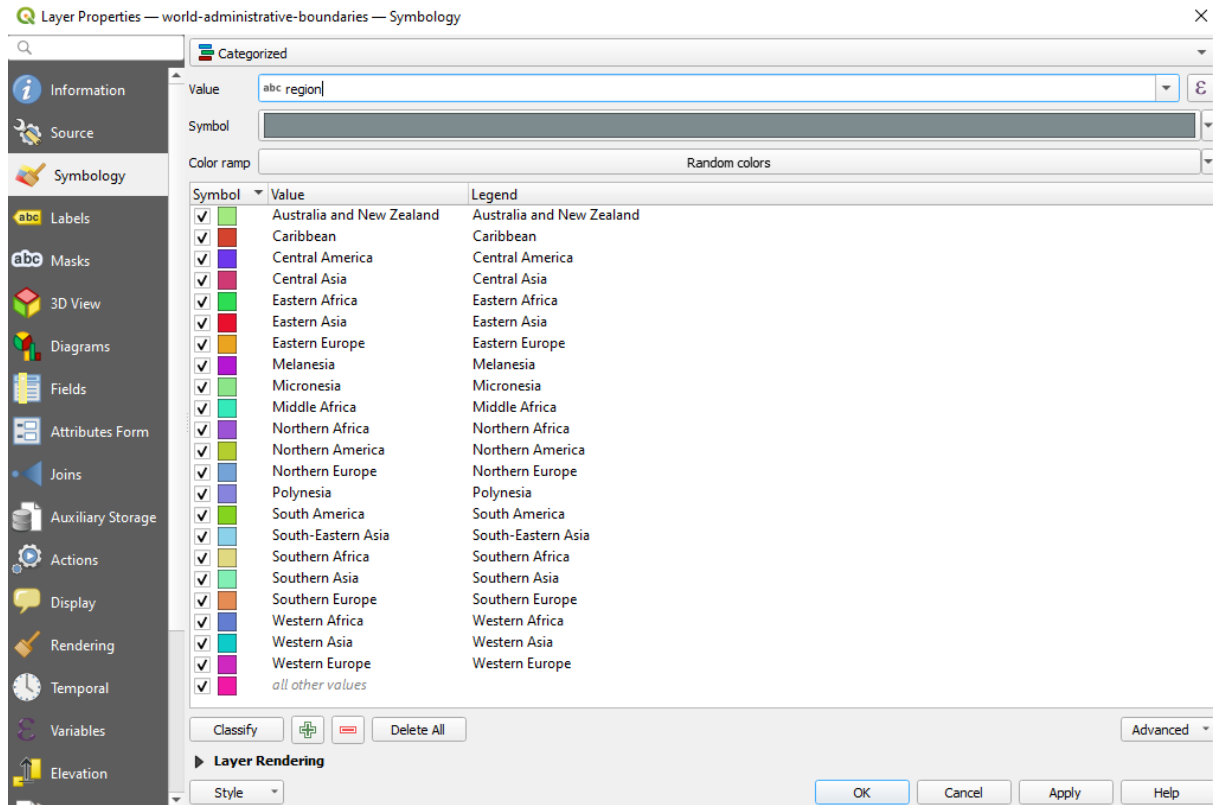


6.18. ábra: Réteg színének beállítása a „Single Symbol” opcióval

A felső egységben lévő legördülő listában meg tudjuk adni, hogy a kiválasztott réteg színezése milyen elv szerint történjen. Ezek közül mutatjuk be a három legalapvetőbb lehetőséget.

1. „Single Symbol” opció: az egész réteg egységes színű, a kívánt színt a felső egységben lévő „Color” felirat mögötti színes (az ábrán homokszínű) téglalapra kattintva tudjuk módosítani. Ez látható a 6.18. ábrán.
2. „Categorized” opció: a réteg színezése egy adott attribútum összes felvett értéke alapján külön-külön színekkel történik. A kívánt attribútumot a „Value” legördülő listából tudjuk kiválasztani, míg a színskálát a „Color ramp” menüpontban. Ezek után a „Classify” gombot benyomva tudjuk az értékeket a közepén található ablakba kilistázni. Ha egy adott elemet törölni szeretnénk a listából, akkor kattintsunk rá a bal

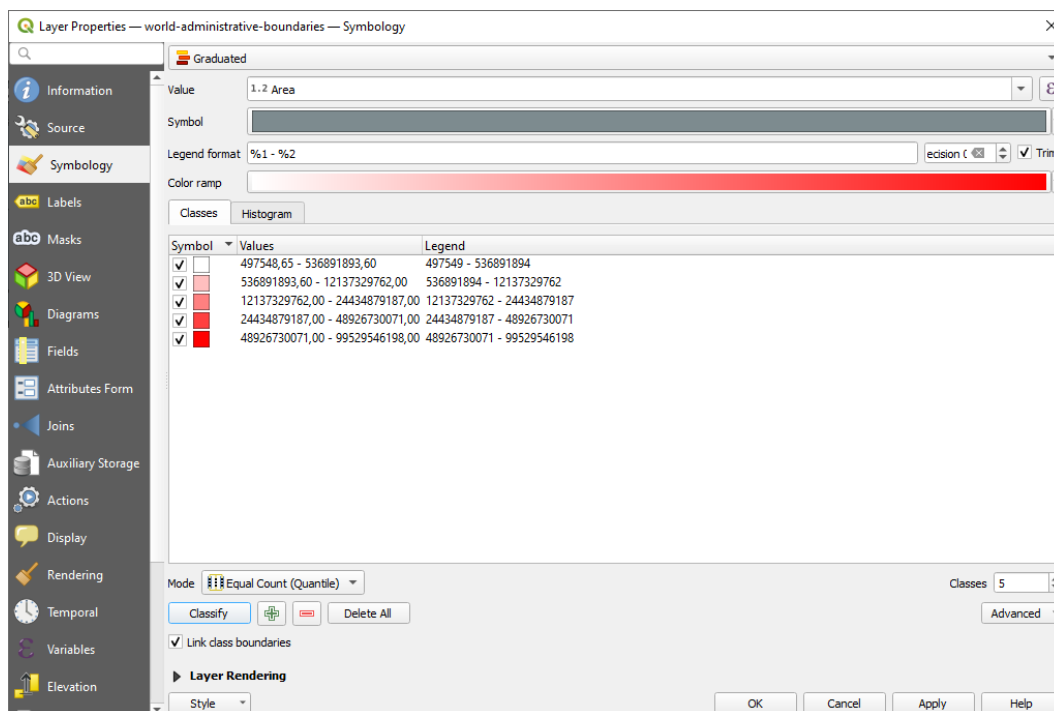
egérgombbal, majd a „Delete” gomb megnyomásával töröljük ki. Ha minden elemet törölni szeretnénk, akkor a „Delete all” gombot nyomjuk meg. Az ablakban a „Label” oszlopban lévő elemekre bal egérgombbal duplán kattintva adhatunk címkét az adott kategóriának, így majd a réteglistában az adott kategória a címke nevével fog megjelenni. Ezek a lehetőségek láthatók az 6.19. ábrán.



6.19. ábra: Réteg színének beállítása a „Categorized” opcióval

3. „Graduated” opció: a „Categorized” opcióhoz hasonlóan itt is a réteg színezése egy adott attribútum értékei alapján történik, de az előzővel ellentétben nem az összes felvett érték egyenkénti megkülönböztetésével, hanem a felvett értékek szerinti, valamilyen elv szerint kialakított kategóriákba rendezéssel. A „Classes” menüpontban megadhatjuk, hogy hány kategóriát képezzen a rendszer, míg a „Mode” megadásával a kategóriák kialakításának elvét választhatjuk meg (például a kategóriaképzés a felvett értékek alapján vagy az adott osztályközbe eső elemszám alapján legyen egyenletes). A „Value” oszlopban bármelyik értékre bal egérgombbal duplán kattintva a kategóriahatárokat egyedileg is tudjuk szerkeszteni. Ezek a lehetőségek láthatók a 6.20. ábrán.

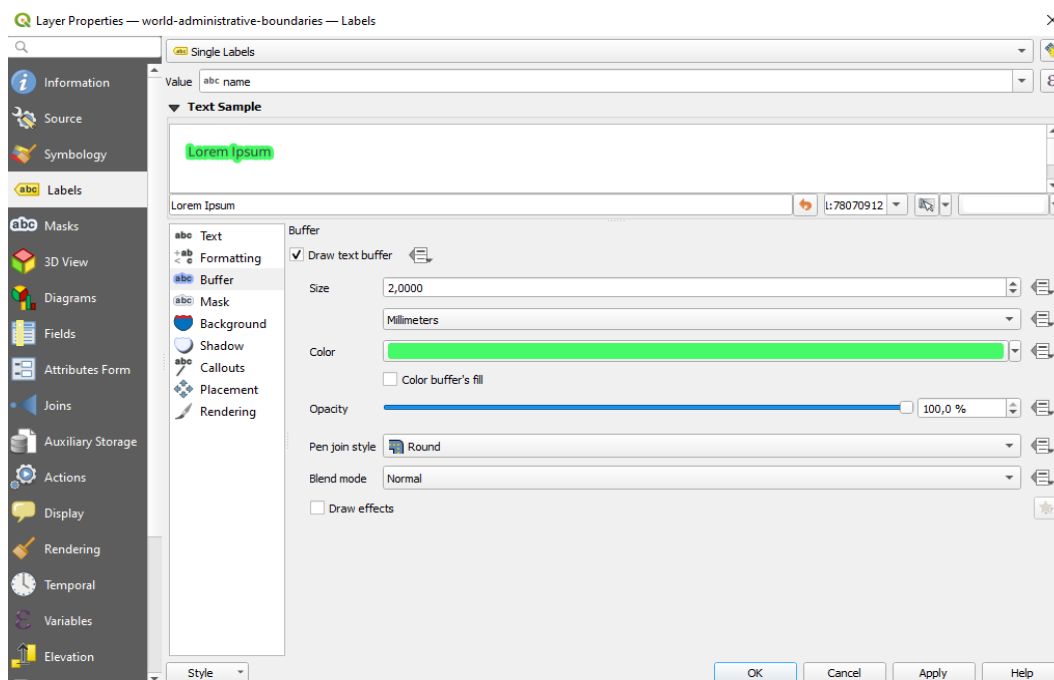
Általánosan, minden rétegszínezési módra érvényes lehetőség a réteg átlátszóságának állítása, amit a „Symbol” melletti színes téglalapra kattintva az „Opacity” menüpontban tehetünk meg. Az érték minél nagyobb, a réteg annál inkább átlátszó. Ez a lehetőség akkor hasznos, ha több réteg helyezkedik el egymáson, és mind az alsó, mind a felső egység információtartalmát egyszerre kívánjuk megmutatni.



6.20. ábra: Réteg színének beállítása a „Graduated” opcióval

## 2. Rétegek címkézése

A címkézés funkció segítségével a térképi megjelenítés során az egyes rétegek egy-egy attribútumának értéket szövegesen is ki tudjuk írni a képernyőre. Ehhez kattintsunk az adott réteg nevére a bal oldali listában a jobb egérgombbal, majd válasszuk ki a „Properties” opciót, ezután a felugró ablakban jelöljük ki a „Labels” funkciót. A menüpont a 6.21. ábrán látható.

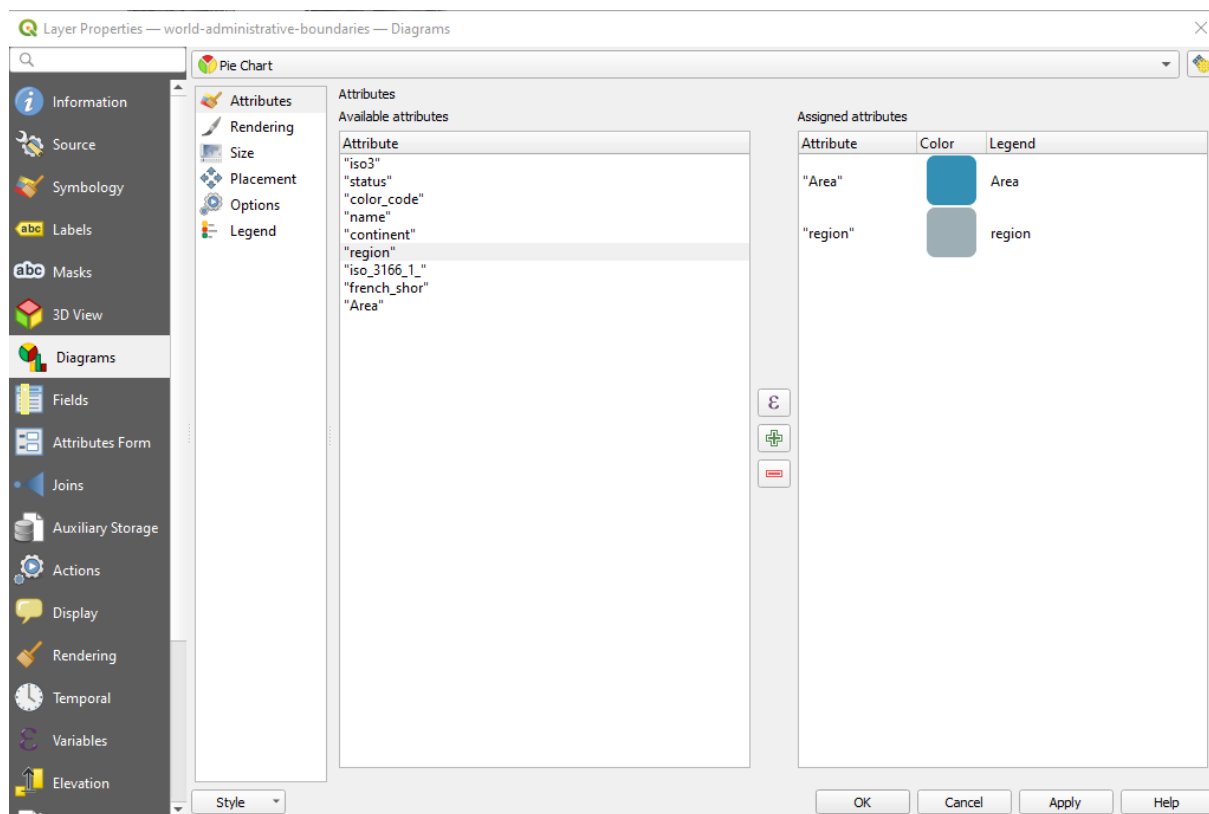


6.21. ábra: Réteg címkézése

A legfelső sorban a legördülő listából tudjuk kiválasztani, hogy a címkézést melyik attribútum szerint kívánjuk megtenni. A bal oldalon található listában a szövegszerkesztőből már ismert tulajdonságait állíthatjuk a megjelenítendő feliratnak, kiegészítve olyan jellemzőkkel, mint például a térbeli elhelyezkedés. Egy funkciót érdemes külön kiemelni, melynek neve „buffer”. Ez szegélyt jelent, ami a 6.21. ábrán látható mintaszöveg esetében zöld színű. Megjegyzendő, hogy „buffer” alkalmazható a különböző rétegek elemein is, például egy sokszög határától adott távolságon belül eső objektumok szemléltetésére. A címkék gyakorlati alkalmazására látható példa az előző lecke utolsó, 6.17. ábráján, amivel a gyakorlófeladatot illusztráltuk. Abban az esetben fehér buffert alkalmaztunk.

### 3. Diagramok megjelenítése

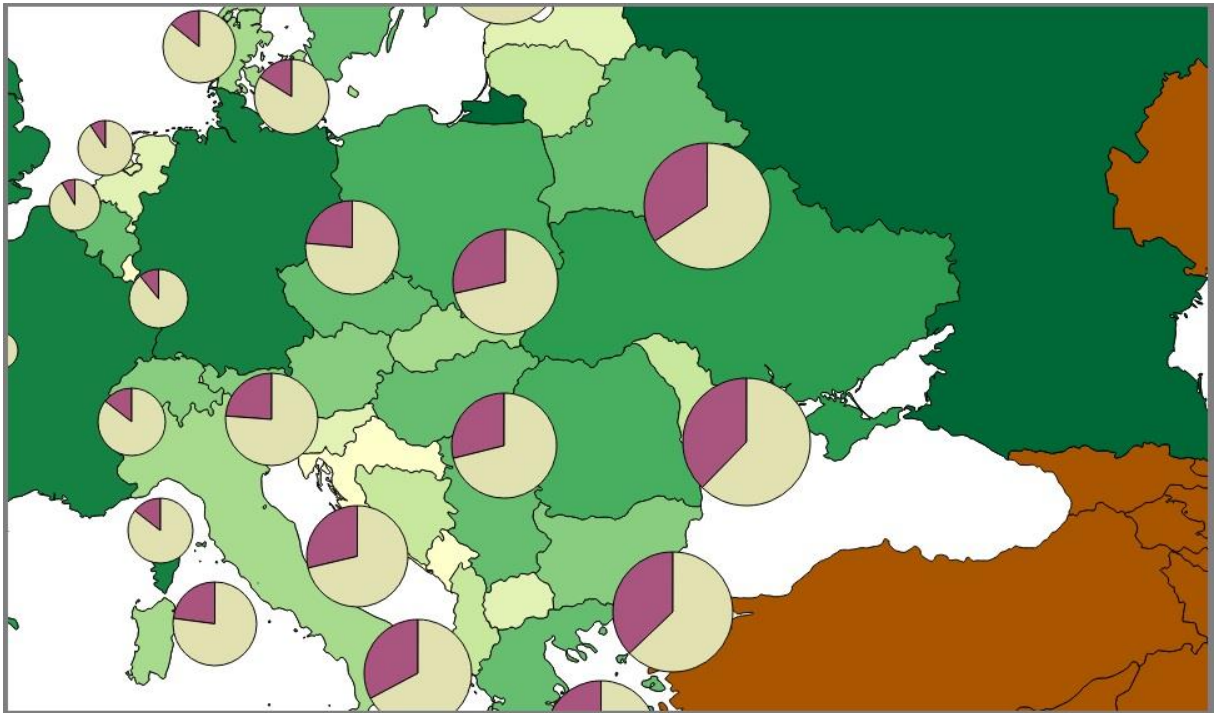
A diagram funkció megnyitása az előbbiekhöz hasonlóan történik. A bal oldali réteglisztában kattintsunk az adott réteg nevére jobb egérgombbal, majd válasszuk a „Properties” opciót, ezután a felugró ablakban a „Diagrams” menüpontot. Ezt az állapotot mutatja a 6.22. ábra.



6.22. ábra: A diagramszerkesztő beállításai

A felső sorban választhatjuk ki a diagram típusát, ami lehet kör, oszlop vagy szöveg. A megjeleníteni kívánt attribútumokat a középső egységben tudjuk kiválasztani. Bal egérgombbal jelöljük ki őket, majd a „+” gomb megnyomásával adjuk őket hozzá a jobb oldali listához. Amennyiben mégsem szeretnénk a megjelenítést, a „-” gombbal tudjuk eltávolítani a nem kívánt elemeket. Az „ε...” gomb segítségével egy a korábban már megismert „Field calculator”-hoz teljesen hasonló felület nyílik meg, ami lehetővé teszi, hogy ne csak direkt, hanem számított értékek is megjeleníthetők legyenek. Az egyes

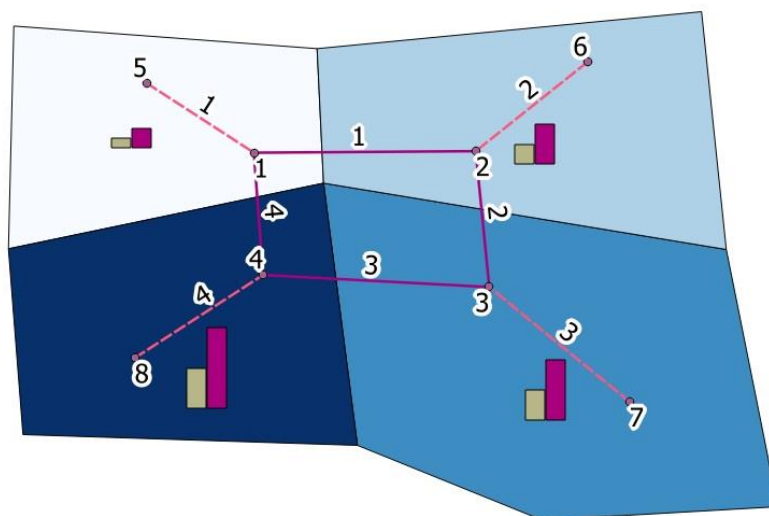
megjelenített attribútumok színeit a „Color” oszlopban a megfelelő színes téglalapra bal egérgombbal kattintva tudjuk módosítani. Érdeemes megjegyezni, hogy a diagramok mérete diagramtípustól függően változtatható, illetve lehetőségünk van a méretet valamilyen attribútum szerint változóra választani (például a diagram mérete egy adott terület népességszámával legyen arányos). A „Size” mezőben a maximális méretet adjuk meg, amivel arányos lesz a többi diagram mérete. Egy ilyen beállításra mutat példát a 6.23. ábra.



6.23. ábra: Diagramok alkalmazása

Gyakorló feladatok:

- Az előző lecke végén létrehozott állománnyal dolgozzunk!
- Hozzuk létre a 6.24. ábrát!



6.24. ábra: A létrehozandó megjelenítés

- A beállítások az alábbiak:
  - Minden rétegen címke az azonosító megjelenítésével 18,25-es betűméret, 2-es bufferrel.
  - A ZONE réteg megjelenítésének típusa Graduated, 4 osztállyal, a lakosság szerint.
  - Az oszlopdiagram a zónához kapcsolódik, a méretet meghatározó attribútum az AZONOSITO, a maximális méret 10, homokszínnel jelenjen meg az azonosító értéke, lilával a lakosság 500-ad része!
- Mentsük el a munkát, mert a következő fejezetekben ezzel fogunk dolgozni.

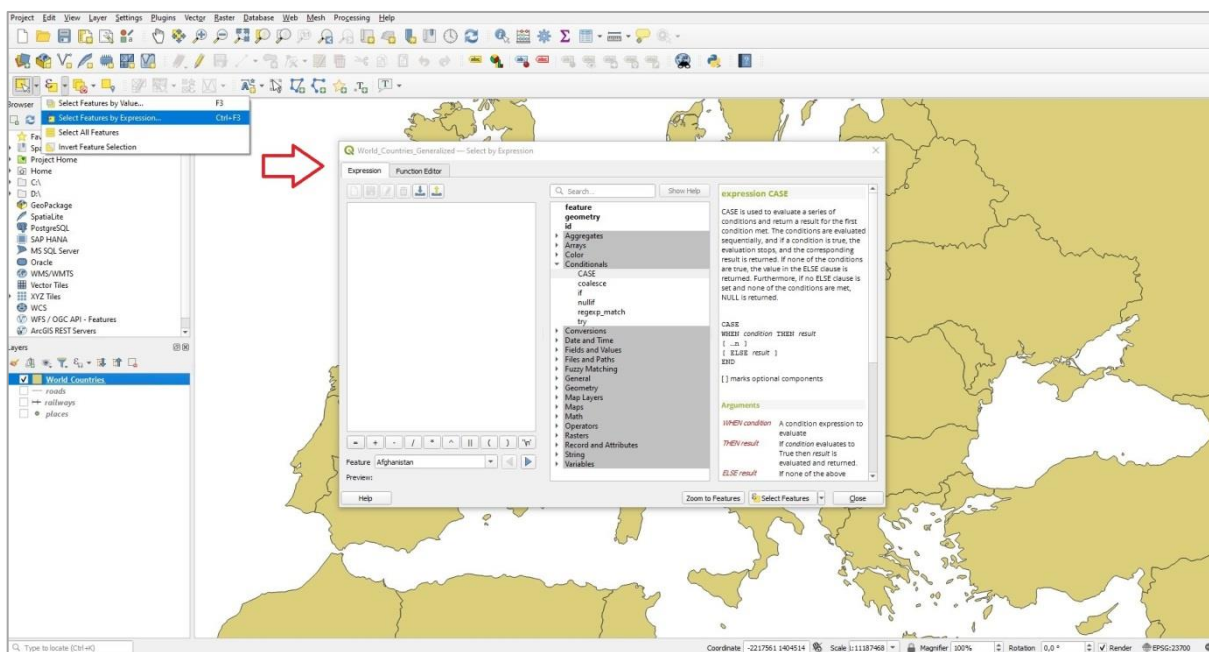
## 6. lecke: Lekérdezések és adatkapcsolatok

A lecke első részében áttekintjük azokat a lehetőségeket, amikor egy réteg elemeinek kijelölését nem a képernyőn történő direkt kattintással, hanem valamilyen attribútum vagy az elhelyezkedésre vonatkozó tulajdonság alapján tesszük meg. A második részben kitérünk az ilyen módon kiválasztott elemek új rétegbe, illetve új shape-fájlba történő exportálásának lehetőségére, továbbá bemutatjuk a CSV-fájlok kezelését QGIS-ben.

### 1. Lekérdezések a kifejezés-szerkesztő használatával

A kijelölési lehetőségekről szóló alfejezetben már utaltunk a kifejezés-szerkesztő funkcióra, melynek segítségével egy rétegből annak valamilyen attribútumai szerint tudunk kijelölni bizonyos elemeket.

A bal oldali listából válasszuk ki a megfelelő réteget, majd kattintsunk a „Select Features by Expression” menüpontra. Ezután egy, a korábban már megismert „Field calculator”-hoz teljesen hasonló felületen tudjuk beállítani a kiválasztás módját. A lekérdezések az SQL-nyelv szabályai szerint történnek, amivel korábbi tanulmányaink során már találkoztunk. Érdekes a bal egérgombot egyszer megnyomva rákattintani az adott lehetőségre, mert ekkor a jobb oldali ablakban az alkalmazást segítő leírást és példákat is láthatunk. A leírt folyamatot mutatja a 6.25. ábra.

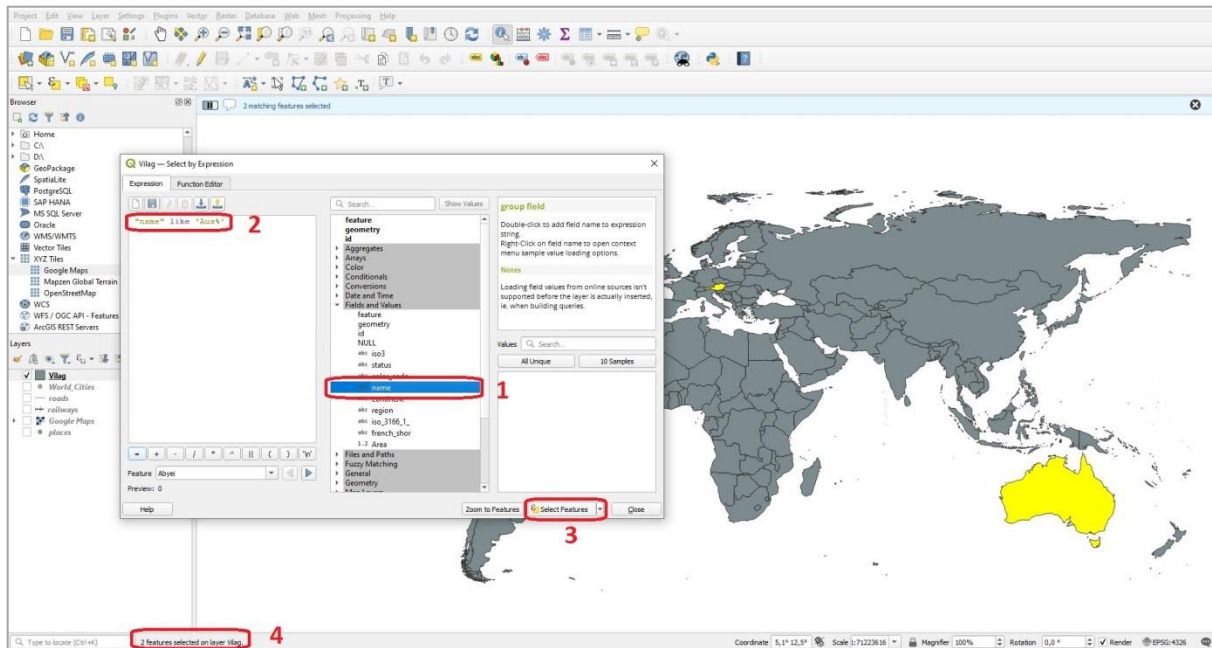


6.25. ábra: A kifejezés-szerkesztő

A következőkben a kifejezés-szerkesztő alkalmazására mutatunk be egy példát, melynek során a világ országaiból kiválasztjuk azokat, amelyek neve „Aus” betűkkel kezdődik. A minta shape-ben az országneveket a NAME mező tartalmazza. Mivel ez szöveges adattípusú, és mi ebben egy részletet keresünk, ezért a „like” operátort alkalmazzuk, majd e



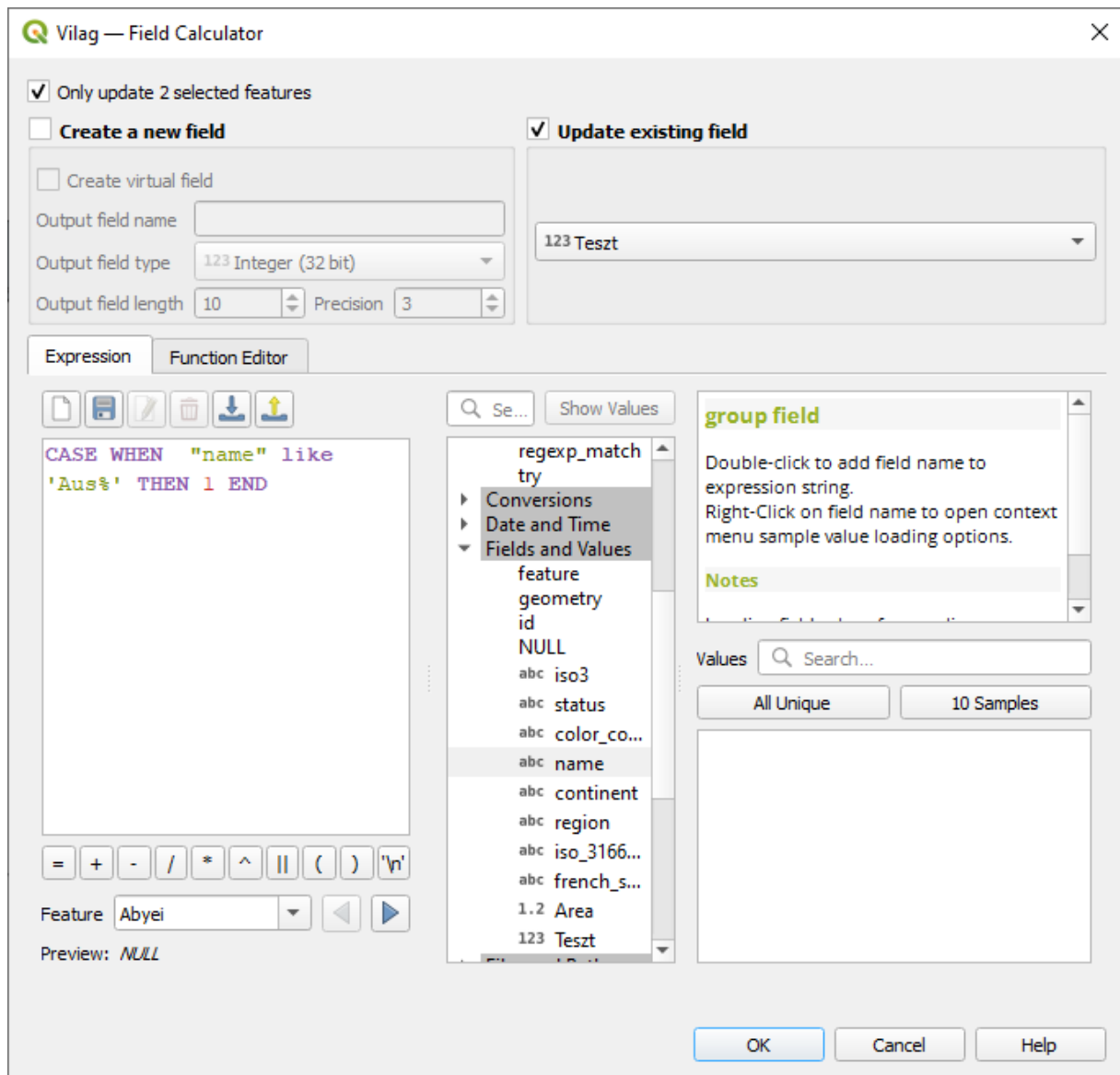
mögé az 'Aus%' kifejezést írjuk. Az aposztrófokra szintén a név szöveges adattípusa miatt van szükségünk, a %-kal pedig az „Aus” mögött lévő tetszőleges számú és fajtájú karaktert helyettesítjük. Ezután a bal egérgombbal megnyomjuk a „Select” gombot, amire a képernyő bal alsó sarkában megjelenik a „2 features selected on layer Vilag” felirat, ami mutatja, hogy a kifejezéssel 2 találatot értünk el (Ausztria és Ausztrália), mindezt a térképen kijelölve is láthatjuk. Ezt mutatja a 6.26. ábra.



6.26. ábra: A kifejezés-szerkesztő alkalmazása

Természetesen a most bemutatott lekérdezési lehetőségek alkalmazhatók a „Field calculator” használatkor is, csak ekkor feltételként szerepelnek. Például, ha a „Teszt” mezőt Ausztria és Ausztrália esetén egy 1-es értékkel akarjuk feltölteni, akkor a „Field calculator”-ba a „CASE WHEN „NAME” LIKE 'Aus%' THEN 1 END” kifejezést kell beírunk (valamint az „Update existing field” opciónál a Teszt mezőt bejelölnünk).

Az esetlegesen bonyolultnak tűnő szintaktikához segítséget kapunk: a 6.26. ábrán látható - a „Field calculator”-hoz teljesen hasonló - felugró ablakban a „Conditionals” pont alatt megtaláljuk a CASE opciót, amire bal egérgombbal duplán kattintunk, így a megfelelő szintaktikával (sötétkék betűkkel) azonnal hozzáadódik az alsó ablakban lévő „Expression List”-hez. Ezt mutatja a 6.27. ábra, miután a program által mutatott helyekre beírtuk a megfelelő kifejezéseket.

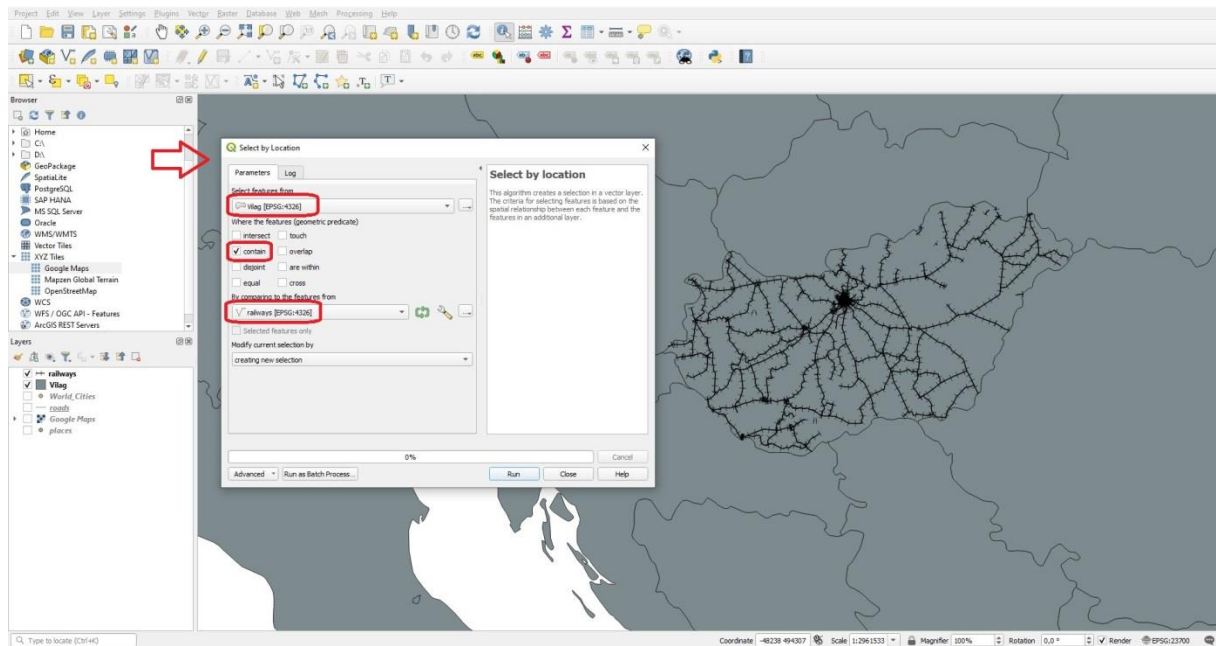


6.27. ábra: Lekérdezésen alapuló kitöltés a „Field calculator”-ban

## 2. Térbeli lekérdezések

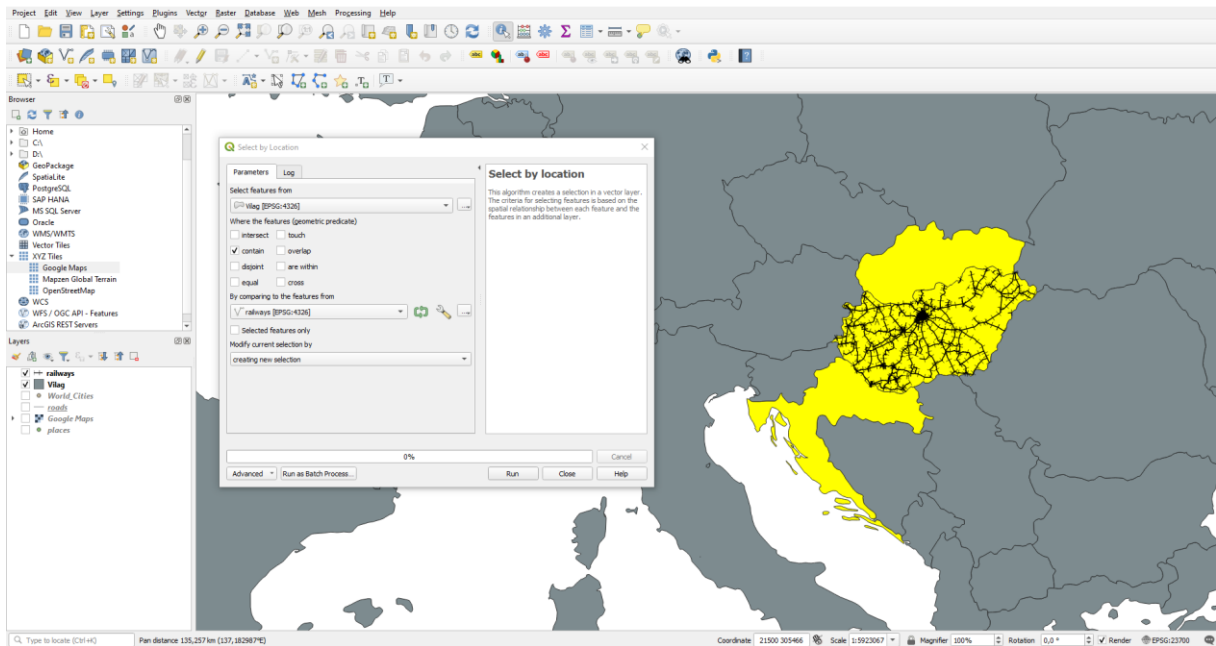
A térbeli lekérdezések segítségével egy adott réteg elemei közül azok valamely más réteghez képesti pozíciója alapján tudjuk kiválasztani a megadott feltételt teljesítőket. A következő mintapéldában szereplő két réteg: a világ országai és Magyarország vasútjai. A feladat során azt akarjuk kideríteni, hogy a világ országai rétegen szerepel-e olyan vasúti szakasz, ami nincs Magyarország területén.

A térbeli lekérdezés módját a 6.28. ábrán látható módon tudjuk kiválasztani.



6.28. ábra: A térbeli lekérdező mód

Először bal egérgombbal kattintsunk a felső menüsoron a „Select by location” gombra, ami után felugrik az azonos nevű ablak. (Amennyiben a funkció nem érhető el vagy nem működik, akkor lapozzunk az Egyebek: nyomtatás, pluginok fejezethez, a Pluginok egységhez!) A felső „Select features from” egységben tudjuk megadni, hogy melyik rétegből akarunk elemeket kiválasztani, míg a harmadik, „By comparing the features from” egységben azt, hogy melyik másik réteggel való kapcsolat alapján. A kapcsolat típusát a két egység között lévő „Where the features (geometric predicate)” menüpontban tudjuk kiválasztani. A kapcsolatok lehetséges típusa az érintett rétegek fajtájától (vonalas-polygon, polygon-polygon, stb.) függ. Az alsó „Modify current selection by” egységben pedig azt tudjuk megadni, hogy a kiválasztás eredményét mire használjuk. A Spatial Query ablak négy egységéből kiválasztott elemeket összeolvasva értelmes angol mondatot kapunk, így ellenőrizhető, hogy milyen eredményt várhatunk. Az eddigieket a konkrét példába átültetve: a „Vilag” nevű rétegből akarjuk kiválasztani azokat az elemeket, amik tartalmazzák a „railways” nevű réteg valamely elemét. Az „Apply” gomb benyomásával hajthatjuk végre a parancsot (ami sokáig is eltarthat). Az eredmény a 6.29. ábrán látható.

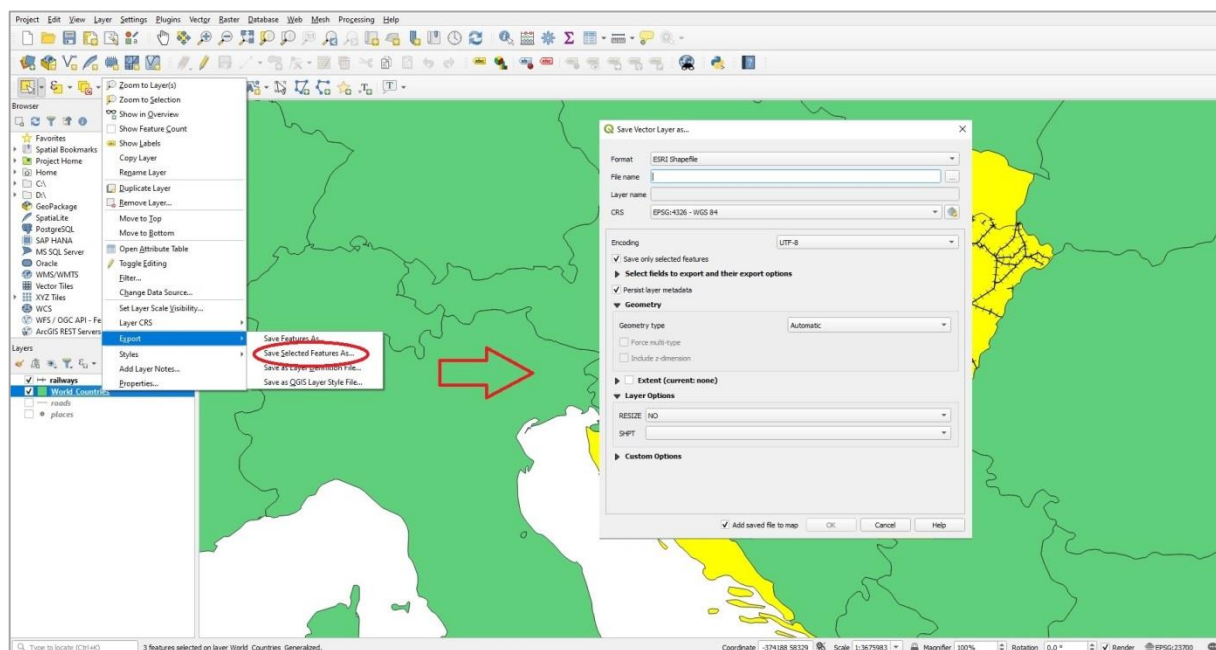


6.29. ábra: Térbeli lekérdezés eredménye

A 6.29. ábráról megállapíthatjuk, hogy a magyarországi vasutak rétegében Horvátországban és Szlovákiában van önálló szakasz Magyarországon kívül.

### 3. Adatok exportálása

Adatok exportálásához kattintsunk jobb egérgombbal a bal oldali listában a megfelelő réteg nevére, majd válasszuk az „Export/Save Selected Features As...” funkciót. Ez látható a 6.30. ábrán.



6.30 ábra: Adatexportálás

A felugró ablakban a „Format” menüpont alatt választhatjuk ki, hogy milyen fájl típusba történjen az exportálás. Ez esetünkben leggyakrabban az ESRI Shapefile lesz, de a következő alfejezetben mutatunk példát más típusra is. A „File name” ablakban tudjuk megadni a mentés helyét, illetve a shape-fájl nevét, mindezt a „Browse” gomb megnyomásával célszerű megtenni. A „CRS” pontban a mentett shape-fájl koordináta-rendszerét adhatjuk meg, ha nem felel meg az alapértelmezett beállítás, ami a forrásréteg koordináta-rendszerét ajánlja fel.

Egy igen fontos funkciót takar a „Save only selected features” jelölőnégyzet. A korábbiakban, illetve ebben a leckében is már számos lehetőséget láttunk arra, hogy shape-fájlok elemeit kijelöljük (akár manuálisan, akár lekérdezések útján), eddig azonban az eredményeket nem mentettük. Ezt tudjuk megtenni a jelölőnégyzet aktiválásával. (Ellenkező esetben a réteg teljes tartalmát mentjük.)

Az alsó sorban megjelenő „Add saved file to map” jelölőnégyzet aktiválásával az exportált fájl réteggé hozható hozzá a projektfájlhoz.

#### 4. Point shape-fájlok exportálása és importálása CSV-fájlokból

Az exportálási folyamat bemutatása során utaltunk rá, hogy az ESRI Shapefile formátum helyett mást is választhatunk. Erre egy point shape-fájloknál előforduló példa a strukturált szövegfájl, a CSV-formátum. Ekkor a réteg attribútumait egy textfájlba mentjük ki. Azonban CSV-fájlok esetén ennél sokkal fontosabb az importálás lehetősége. Sokszor előfordul, hogy adatokat valamilyen mérőeszközből CSV-formátumban tudunk kinyerni vagy esetleg Excelben szerkesztünk, majd ezeket hívjuk be valamilyen térinformációs rendszerbe. Erre mutatunk egy példát a következőkben. (Amennyiben Excelben szerkesztünk adatokat, úgy ügyelni kell rá, hogy a mentés CSV formátumban történjen.)

A mintában a Pest megyei autószalonok CSV-állományát hívjuk be. Ennek egy részlete látható a 6.31. ábrán:

```
latitude;longitude;marka;kereskedo;ir_szam;telepules;cim;tel_szam;honlap
47.449589;18.971533;Alfa Romeo;Autóház Budaörs/Néző M5 Kft.;2040;Budaörs;Kinizsi u. 28.;23/455-455;www.autohazbudaors.hu
47.391493;18.934221;Alfa Romeo;Autócentrum Szabó Kft.;2030;Érd;Kis-Duna u. 9.;23/524-028;www.fiatszabo.hu
47.60983;19.336193;Alfa Romeo;Darázs II. Kft.;2100;Gödöllő;Dózsa Gy. út 160.;28/525-200;www.darazsauto.hu
47.762277;19.140497;Alfa Romeo;Release Autó Kft.;2600;Vác;Szt. László út 11.;27/510-300;www.qauto.hu
47.379202;19.357885;Alfa Romeo;Atlasz Autó Kft.;2225;Üllő;Pesti út hrsz. 3337/6.;29/522-522;www.atlaszauto.hu
47.452382;18.907682;Audi;Planet Autó M1 Kft.;2045;Törökbálint;Tó u. 2.;23/456-789;www.planetseat.hu
47.455182;18.962399;BMW;Rack Autó Kft.;2040;Budaörs;Csata út. 23.;23/416-771;www.bmw-rackautobudaors.hu
47.530184;18.925095;Chevrolet;Interat;2092;Budakeszi;Budaörsi út 2.;23/535-032;www.chevroletinterat.hu
```

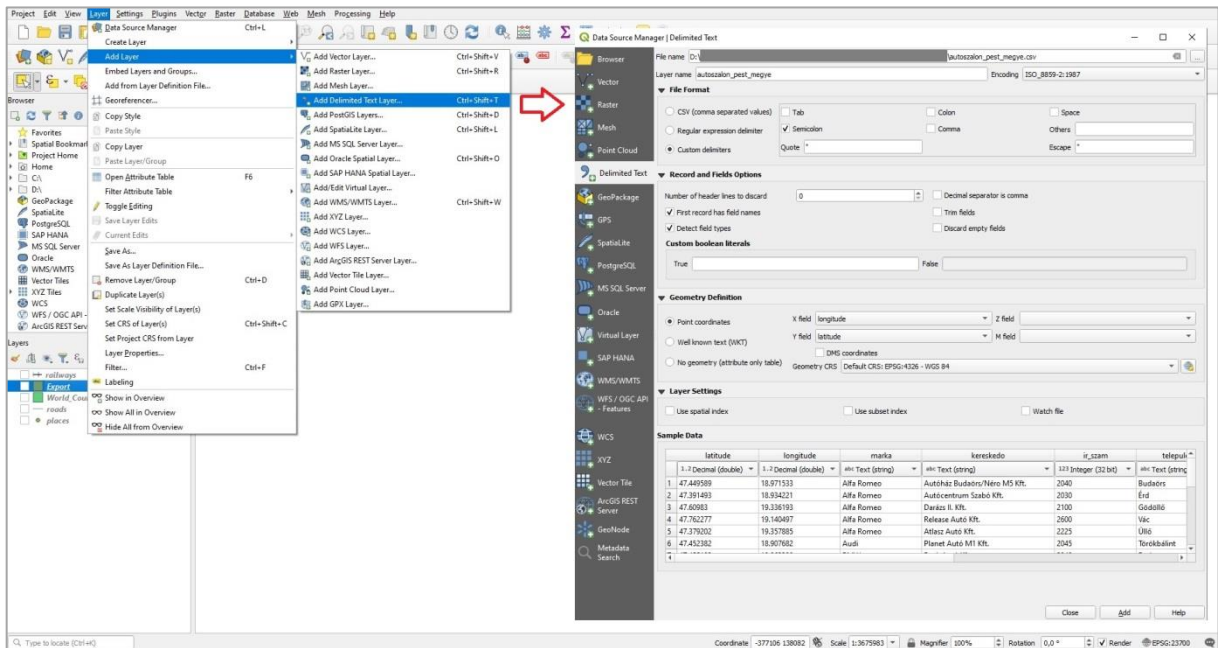
6.31. ábra: Minta CSV-adatállomány

A QGIS-ben CSV-fájl behívásakor kattintsunk a „Layer/Add Layer/Add Delimited Text Layer” gombra.

A felugró ablakban a „Browse...” gomb megnyomása után tudjuk kiválasztani a behívandó állományt. A „Layer name” menüpontban adjuk meg a QGIS-ben megjelenő réteg nevét. A „File format” csoportban a „Custom delimiters” pontban tudjuk megjelölni az egyes elemeket egymástól elválasztó karaktert, ami a példában a pontosvessző (semicolon). Amennyiben az adatsorunknak van fejléce - esetünkben ez igaz -, akkor aktiváljuk a „First record has field names” jelölőnégyzetet. Az „X field” és „Y field” legördülő listából a megfelelő koordinátákat választjuk ki. Ha van longitude és latitude nevű, a megfelelő GPS-koordinátákat tartalmazó mező, akkor a rendszer ezt automatikusan felismeri (ez igaz a



példában is). Beállítandó még a koordináta-rendszer, ami GPS-koordináták longitude és latitude adatai esetén WGS 84. A folyamat a 6.32. ábrán látható.



6.32. ábra: CSV-állomány importálása

CSV-adatállományok esetén érdemes figyelni a tizedesvessző jelölésére. Az amerikai programok többségében ponttal jelölik, míg Magyarországon vesszővel. Emiatt megeshet, hogy néhány program nem ismeri fel a magyar jelölést. A QGIS-ben (lásd a 6.32. ábrát) a „Decimal separator is comma” jelölőnégyzet aktiválásával ez a probléma áthidalható. Érdemes megjegyezni azt is, hogy a CSV-fájlokban az elválasztó karakter többféle lehet, leggyakrabban vessző, pontosvessző és tabulátor. Figyeljünk, hogy az elválasztó karakter ne egyezzen a tizedesjelölő karakterrel, különben az adatok közti határt nem tudja megállapítani a program.

### Gyakorló feladatok:

- Az előző lecke végén létrehozott állománnyal dolgozzunk!
- A NODE réteg CENTROID mezőjét „Field calculator” segítségével töltsük fel az alábbiak szerint:
  - 0 legyen az érték, ha az AZONOSITO mező értéke kisebb 5-nél,
  - 1 legyen az érték egyéb esetben.
- A LINK réteg TIPUSKOD és SEBESSEG mezőjét „Field calculator” segítségével töltsük fel az alábbiak szerint:
  - ha az AZONOSITO értéke 3-nál kisebb, akkor a TIPUSKOD legyen 10, a SEBESSEG pedig 50,
  - ha az AZONOSITO értéke 3, akkor a TIPUSKOD legyen 20, a SEBESSEG pedig 30,
  - ha az AZONOSITO értéke 4, akkor a TIPUSKOD legyen 30, a SEBESSEG pedig 90!

- Kifejezés-szerkesztő alkalmazásával jelöljük ki azokat a linkeket, amelyek sebessége 45 és 55 közé esik!
- Exportáljuk az előbbi kijelölést új rétegbe, melynek neve legyen LINK\_45\_55, és mentjük új shape-fájlba!
- Térbeli lekérdezéssel jelöljük ki azokat a zónákat, amelyeken megtalálható a LINK\_45\_55 réteg valamely eleme!
- Exportáljuk azokat a node-okat egy CSV-fájlba (x és y-koordinátákkal együtt), amelyek CENTROID értéke 1, és nevezzük el ZONE\_CENTROID-nak!
- Importáljuk be a ZONE\_CENTROID.csv-t, majd ellenőrizzük térbeli lekérdezéssel, hogy a centroidok egybeesnek-e a NODE réteg megfelelő elemeivel!

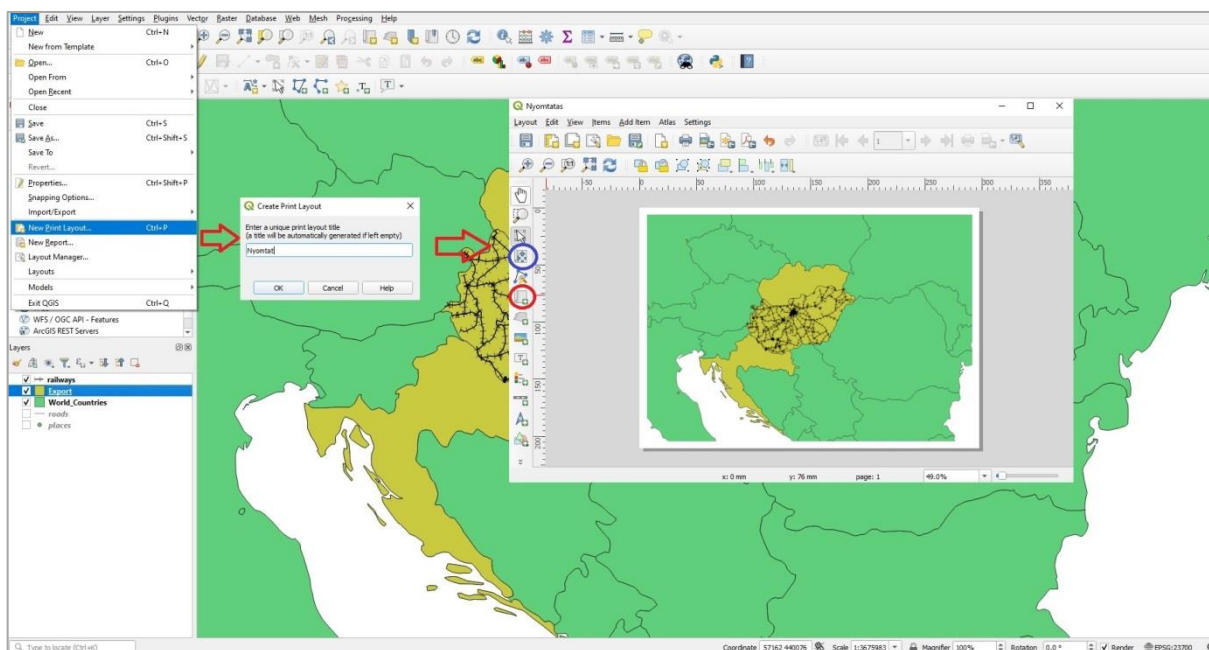
## 7. lecke: Egyebek: nyomtatás, pluginok

Ebben a leckében olyan lehetőségeket mutatunk be, melyekkel a munkánk magasabb színvonalra emelhető, illetve megkönnyíthető. Az első részben megmutatjuk, hogy milyen módon tudjuk az elkészített ábrákat igényes formában kinyomtatni, míg a másodikban a QGIS kiegészítőiről, a pluginokról ejtünk röviden szót.

### 1. Nyomtatás

A QGIS-ben számos különféle lehetőségünk van a nyomtatás beállítására, így jelen alfejezetben az alapokat mutatjuk be, amik alapján az alapvető műveletek elvégezhetők.

Nyomtatni egy úgynevezett nyomtatási kompozíció beállítása után tudunk. Egy projekten belül több beállítás is elmenthető, amik a „Project\Layouts” menüpontból hívhatók elő. Új kompozíciót létrehozni a „Project\New Print Layout” menüpontban tudunk. A felugró ablakban egy egyedi nevet szükséges megadni, ami után megnyílik a kezelői felület. Az „Add Map” ikonra kattintva tudjuk az üres lapra behívni a projektfájl térképét. Ez egy téglalapban fog elhelyezkedni, melyet úgy tudunk létrehozni, hogy a bal egérgombot a tervezett téglalap egyik átlójának végpontjában lenyomjuk, majd nyomva tartjuk és az átló másik végpontjáiig húzzuk, majd ott felengedjük. Ekkor fog megjelenni a képen is látható nézet - ami egyezik a projektfájl aktuális nézetével -, addig csak egy üres lapot látunk. Ezeket a lépéseket mutatja a 6.33. ábra.



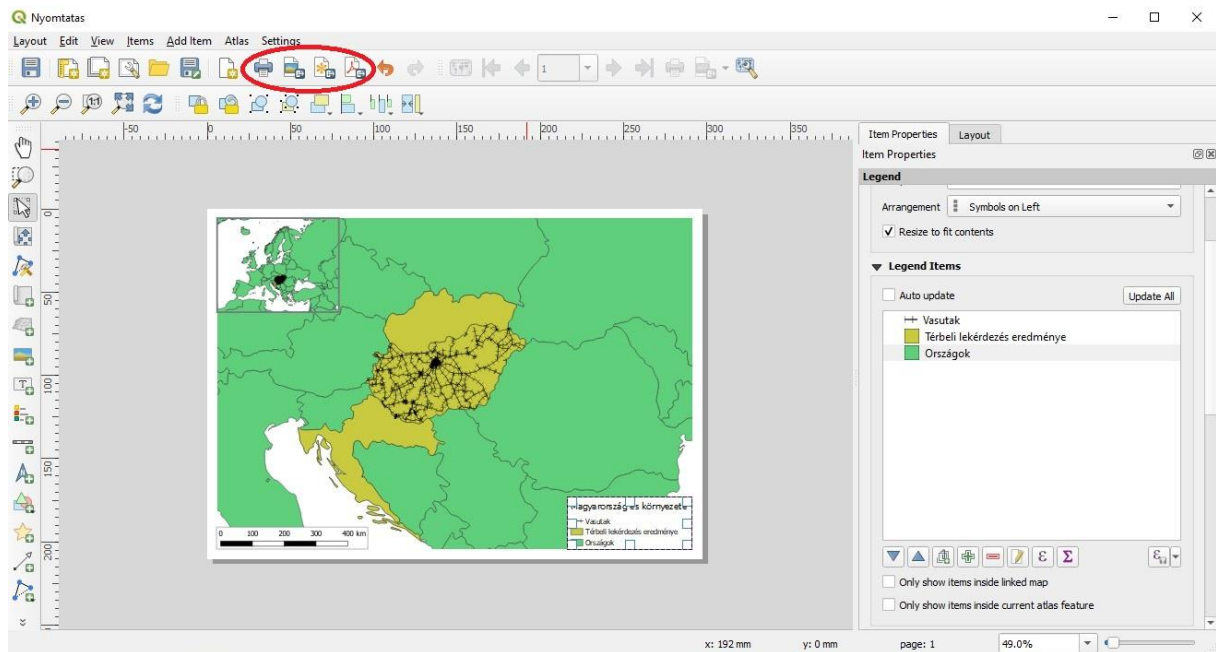
6.33. ábra: A nyomtatási kezelőfelület

Az ábra kereteit bármikor módosíthatjuk, ehhez a bal egérgombbal duplán kell kattintani a képre, ekkor a sarokpontoknál, illetve az oldalak felezőpontjainál lévő fehér négyzeteket (lásd a 6.33. ábrán) bal egérgombbal mozgatva lehet a szegélyeket mozgatni. A kereten belül a rétegek a késsel jelölt „Move item content” ikonra kattintás után mozgathatók a bal



egérgomb nyomva tartásával. Az egér görgetésével pedig nagyítható, illetve kicsinyíthető a méretarány.

Az ábrára rendkívül sokféle kiegészítő elem tehető, például méretarány, jelmagyarázat, cím, stb. Ezek elhelyezkedésének, kiterjedésének változtatása az ábra tulajdonságainál leírtakkal egyezik. Ehhez a „View/Panels/Item properties” lehetőséget kell aktiválni, majd a képre kattintani. Az ekkor megjelenő „Item properties” fülön szabadon beállítható az összes felirat, a betűtípus, a színek, jelmagyarázat esetén a szereplő rétegek nevei, illetve az is, hogy egyáltalán fel legyenek-e tüntetve a jelmagyarázatban. Egy lapra több kép is behívható. Egy mintát láthatunk a 6.34. ábrán.



6.34. ábra: Nyomatási kompozíció

A kész kompozíciót különböző fájlformátumokba menthetjük, illetve közvetlenül nyomtathatjuk a 6.34. ábrán bejelölt ikonokra kattintva.

## 2. Pluginok

A pluginok a QGIS kiegészítői, melyeket bárki programozhat. Arra szolgálnak, hogy olyan funkciókat biztosítsanak, amiket az alapprogram nem tud. A pluginok minden felhasználó számára szabadon elérhetők.

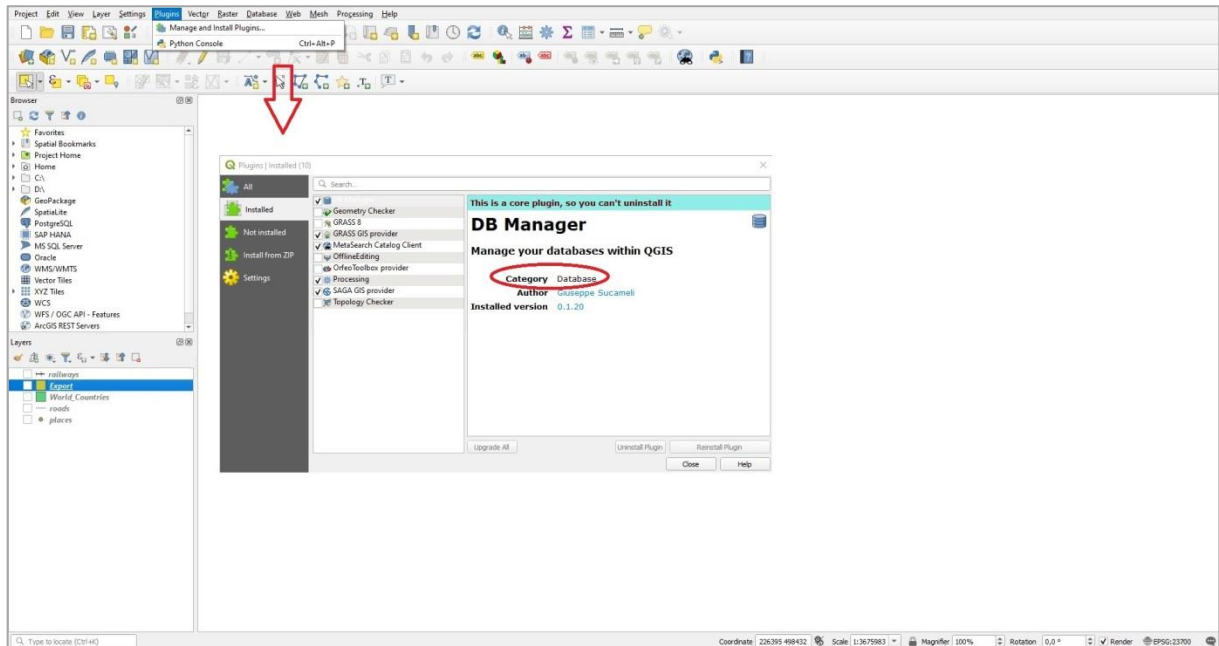
Egy plugin alkalmazásához két lépést kell megtenni:

1. a plugin le kell töltenünk és installálnunk, valamint
2. a plugin engedélyeznünk kell.

Kétféle plugin létezik. A „core plugin”-ok már a QGIS számítógépre történő telepítéskor installálásra kerülnek, így ezeket csak aktiválni kell. Az „external plugin”-ok esetében viszont mindkét lépést magunknak kell megtennünk.

A kezelőfelület a „Plugins\Manage and Install Plugins...” menüpont alatt található. A felugró ablakban az „Installed” fülön a már telepített, míg a „Not installed” fülön a

telepíthető pluginok listáját találjuk. Az egyes pluginok engedélyezése a megfelelő jelölőnégyzet aktiválásával lehetséges. Piros színnel jelennek meg azok a pluginok, melyekkel valami probléma van. A leírtakat mutatja a 6.35. ábra.

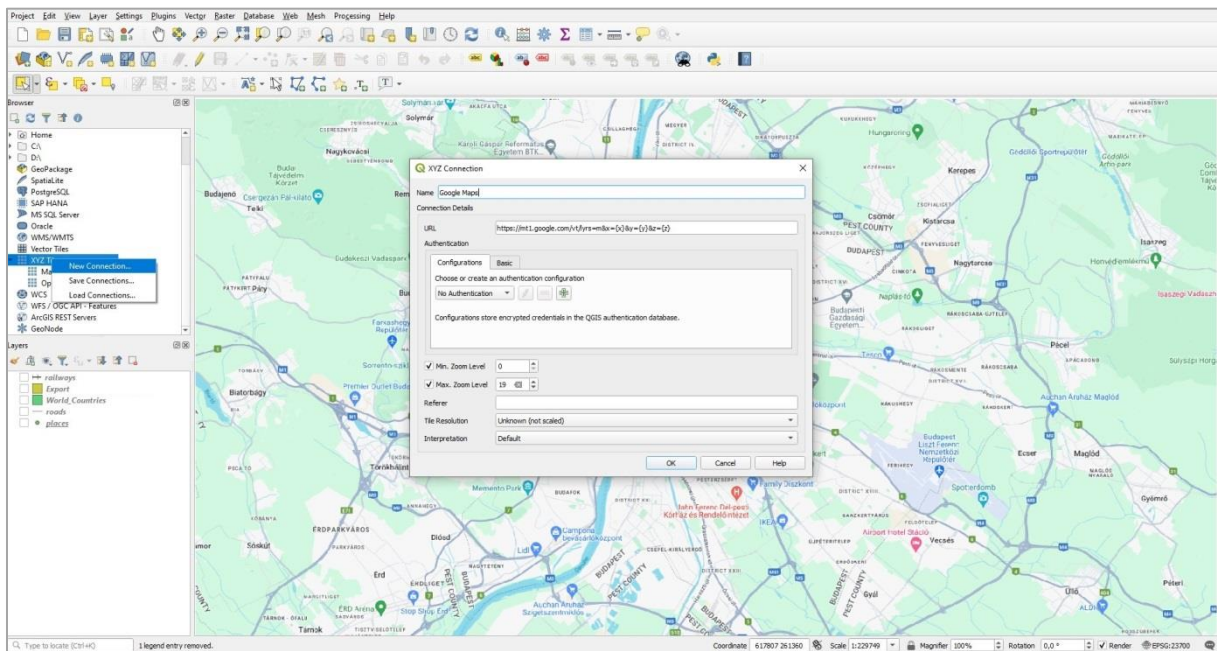


6.35. ábra: Pluginok kezelőfelülete

Az ábrán csak core pluginok láthatók. A plugin leírásánál a jobb oldali ablakban a „Category:” pontban látható az adott plugin besorolása. Ez azért fontos, mert ez alapján tudjuk majd megtalálni a QGIS menüjében, ha használni akarjuk.

### 3. Standard térkép behívása

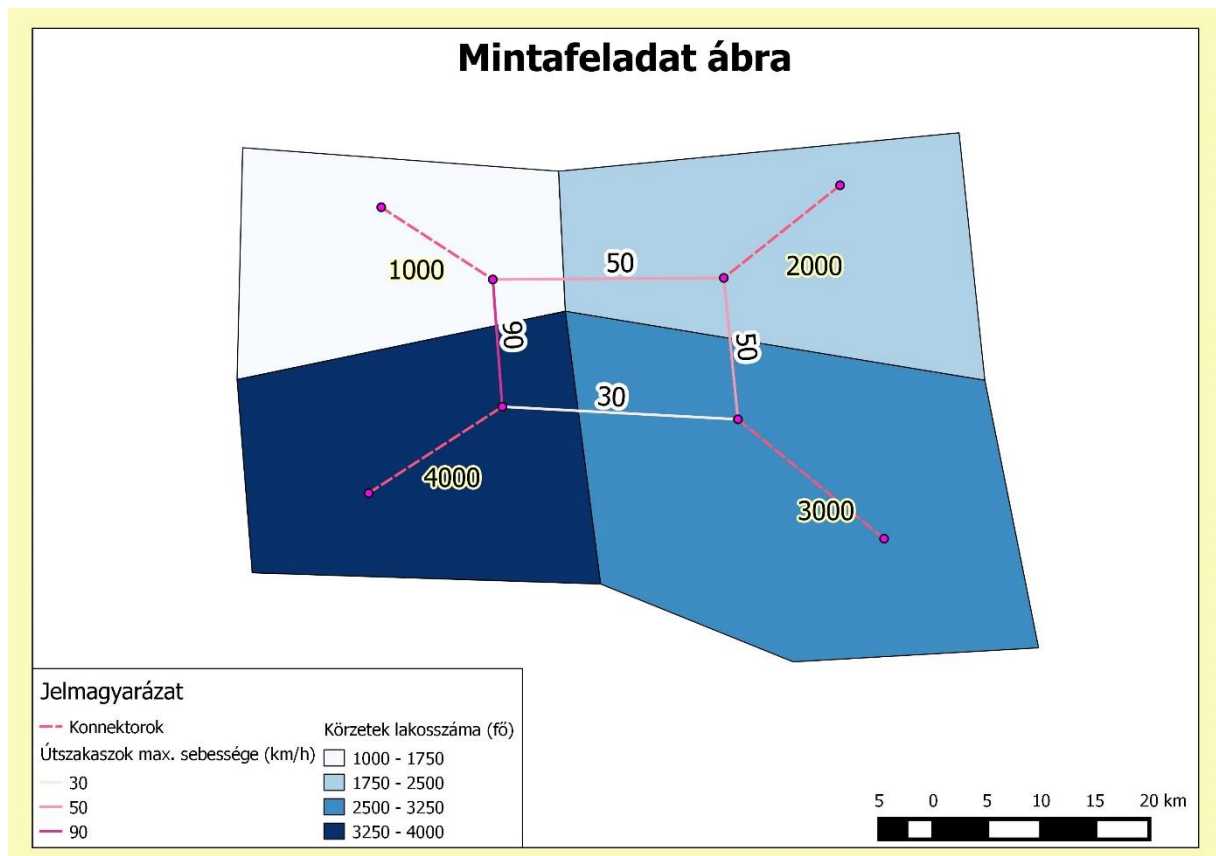
Munkáink során gyakran előfordul, hogy valamilyen internetes térképhez igazodva szeretnénk elkészíteni a geodatabázisunkat. Ehhez nyújt segítséget az „XYZ Tiles” funkció, amellyel internetes térképek hívhatók be háttérként egy önálló rétegbe. Ehhez a bal oldali ablakban az „XYZ Tiles” menüpontra jobb klikkel kattintva kiválasztjuk a „New Connection” opciót, majd a felugró ablakban megadjuk a szükséges adatokat. A „Name” ablakban megadjuk a nevet, pl. Google Maps. Ezt követően az URL ablakban meg kell adni az adott térkép hivatkozási címét, mely a QGIS weboldaláról tölthető le: [https://docs.qgis.org/3.10/en/docs/user\\_manual/managing\\_data\\_source/opening\\_data.html?highlight=xyz%20tiles#using-xyz-tile-services](https://docs.qgis.org/3.10/en/docs/user_manual/managing_data_source/opening_data.html?highlight=xyz%20tiles#using-xyz-tile-services). Google Maps esetén a hivatkozási cím a <https://mt1.google.com/vt/lyrs=m&x={x}&y={y}&z={z}>. A leírtakat mutatja be a 6.36. ábra.



6.36. ábra: Az XYZ Tiles funkció használata

#### Gyakorló feladatok:

- Az előző lecke végén létrehozott állománnyal dolgozzunk!
- Készítsük el a 6.37. ábrát „Print Composer” segítségével A/5-ös méretben, majd mentjük el .jpg formátumban!



6.37. ábra: Gyakorló feladat ábra

- Ügyeljünk a lehető legpontosabb egyezésre!
  - Figyeljünk a címkék elhelyezésére (a zónáké jobbra lefelé eltolt), a buffer méretére és színére!
  - Az útszakaszok sebességi kategóriáinál Categorized, a körzetek lakosságának megjelenítésénél Graduated megjelenítési módot alkalmazzunk!
  - A jelmagyarázat két oszlopot tartalmazzon!
  - A cím legyen félkövér!

## 8. Hasznos linkek

- [1] Online help: [https://docs.qgis.org/3.28/en/docs/user\\_manual/index.html](https://docs.qgis.org/3.28/en/docs/user_manual/index.html)
- [2] Magyarország minta shp - utak, vasutak, természetvédelmi területek, stb.:  
<http://www.mapcruzin.com/free-hungary-arcgis-maps-shapefiles.htm>
- [3] Világ minta shp - országok közigazgatási határai:  
<https://public.opendatasoft.com/explore/dataset/world-administrative-boundaries/export/>
- [4] Világ nagyvárosai minta shp:  
<https://www.naturalearthdata.com/downloads/10m-cultural-vectors/10m-populated-places/>
- [5] Open Street Map (Világ shp térkép): <http://www.openstreetmap.org>