

Intelligens rendszerek gyakorlat

Ezen a gyakorlaton megismerkedhetünk a Fuzzy Logic Toolbox használatával. A Toolbox használatához indítsuk el a MATLAB R2016a-t.

A célja a mai gyakorlatnak, hogy megismerjük, hogyan lehet ezzel a Fuzzy Logic Toolbox-al egy Fuzzy szabályzót létrehozni és használni. Először is meg kell nézzük, hogy mi is ez a Toolbox, mik a részei, mit lehet vele csinálni majd, megnézzük azt is hogyan lehet az elkészült szabályzót kipróbálni tesztelni.

Kicsit előre utalva az elkészült szabályzót a Simulink segítségével lehet majd kipróbálni.

A MATLAB és a Simulink átjárható, tehát a parancssoros környezet és a grafikus Simulink-es környezet. Az átjárást az alábbi két utasítást biztosítja:

- `assignin(ws, 'var', val)`

A ws mindig a base lesz. A base workspace-ben található var változóhoz hozzárendeli a val értéket.

- `[a1, a2, a3, ...] = evalin(ws, expression)`

A ws mindig a base lesz. A base workspace-ben található expression-t fogja lefuttatni és visszaadja a futtatás eredményét az [a1, a2, a3, ...]-nak.

Mindkét esetben a parancsok beírása után az F1 segítségével érhető el a help.

A Fuzzy szabályzó Simulink-es használatához először is meg kell tervezzünk egy szabályzót, minden fontos tulajdonságot be kell állítani. Amit nem állítunk be az default értéken marad, de akkor is be lesz állítva. Ha ezzel végeztünk, akkor megfelelő módon ki kell értékelnünk, azaz működtetni kell bírunk, futtatnunk kell. Lehet futtatni MATLAB parancssorban is, de lehet futtatni Simulink segítségével is.

Két alapvető típusú Fuzzy szabályzó van. Az egyik a Mamdani míg a másik a Takagi Sugeno. A Fuzzy Logic Toolbox esetében az alapértelmezett a Mamdani Fuzzy szabályzó.

A Fuzzy Control Toolbox-hoz azaz a Fuzzy szabályzó konfigurálásához egy GUI áll a rendelkezésünkre. Ezen a gyakorlaton ezzel fogunk megismerkedni. Amikor létrehozunk egy default Fuzzy szabályzót, ezt azzal meg lehet tenni hogy parancssorba beírjuk, hogy :

- `fuzzy`

Ennek hatására létrejön egy üres Fuzzy szabályzó. Egy olyan Fuzzy szabályzó fog létrejönni, ami nem futtatható és le se lehet fordítani. Ennek az az oka, hogy ebből hiányoznak a szabályok.

Alapvetően 3 szintjét különböztetjük meg a Fuzzy szabályzóknak:

- 1. szint: default eset, ami annyira hiányos, hogy le sem tud fordulni, azaz ha ilyen szabályzót akarunk Simulink-en keresztül felhasználni, akkor hibaüzenetet kapunk. Egyszerűen nem fog működni a szimuláció, és értelem szerűen MATLAB parancssorban sem fog elindulni. (0 db szabály)

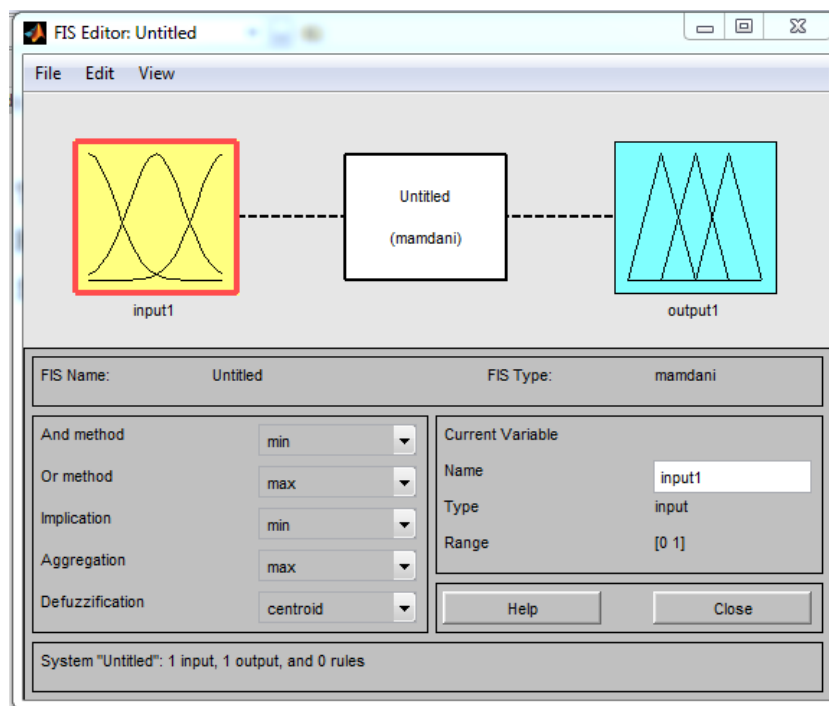
- 2. szint: ez a Fuzzy szabályzó már épen lefordítható, de igazából értelme még nincsen. Itt nem fogunk warning-ot vagy hibaüzenetet kapni, fog futni ki fog adni a kimenetén valamit csak értelmetlen dolog lesz a kimeneten. Lényegében egy konstans. (1 db szabály)
- 3. szint: olyan szintig fel van paraméterezve hogy nem csak fordítható, de értelmes szabályozást is tud csinálni. Ehhez az kell, hogy a szabálybázis minimum 2 szabályt tartalmazzon. Másképpen a legegyszerűbb szabálybázisú Fuzzy szabályzónak minimum 2 szabályt kell tartalmaznia.

Ha már egy Fuzzy szabályzóban van legalább 2 szabály (két elemű szabálybázissal rendelkező Fuzzy szabályzó) akkor azzal már komoly feladatokat lehet megoldani.

Írjuk be a MATLAB parancssorba hogy:

- fuzzy

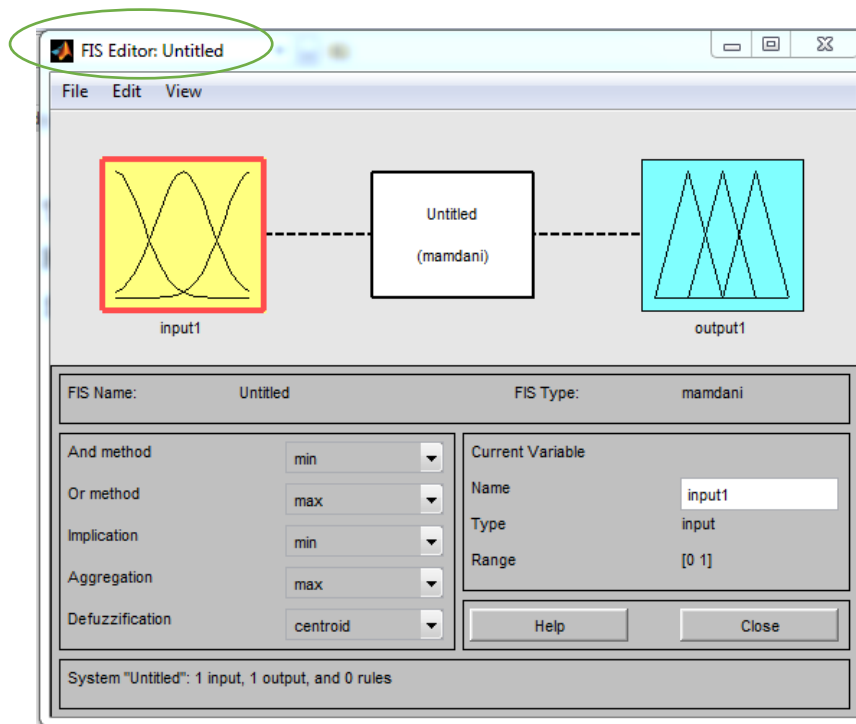
Ha mindent jól csináltunk, akkor létrejön egy üres Fuzzy szabályzó a háttérben, a memóriában, illetve ennek a háttérben formálódó Fuzzy szabályzónak a konfigurátora (Fuzzy Logic Designer). Ennek az első ablaka az alábbi módon néz ki:



Ennek a konfigurátornak összesen 5 ablaka van.

Az első 3 ablak segítségével beállításokat tudunk rajta végezni, míg az utolsó 2-n debug-olásokat, ellenőrzéseket lehet végezni. Ezen az utolsó két ablakon nem lehet olyan módosítást eszközölni, amelynek hatása lesz a későbbiekben a megszerkesztett Fuzzy szabályozókra.

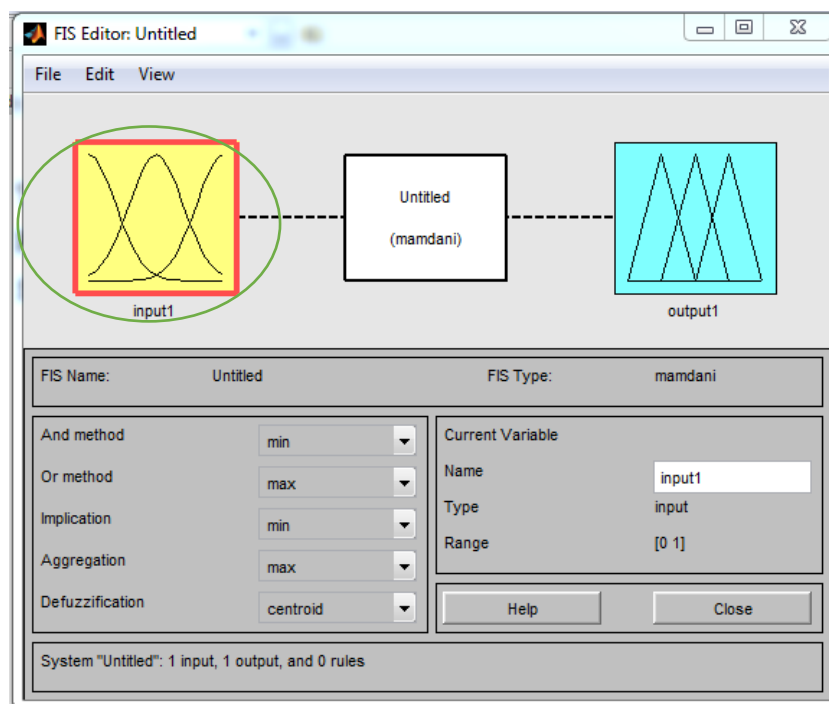
Az ablakok elrendezése monitor 4 sarkába történjen, még így sem fog kiférni minden ablak. Az első ablakban legfelül látható a Fuzzy szabályzó neve ahogy az alábbi ábrán is látható:



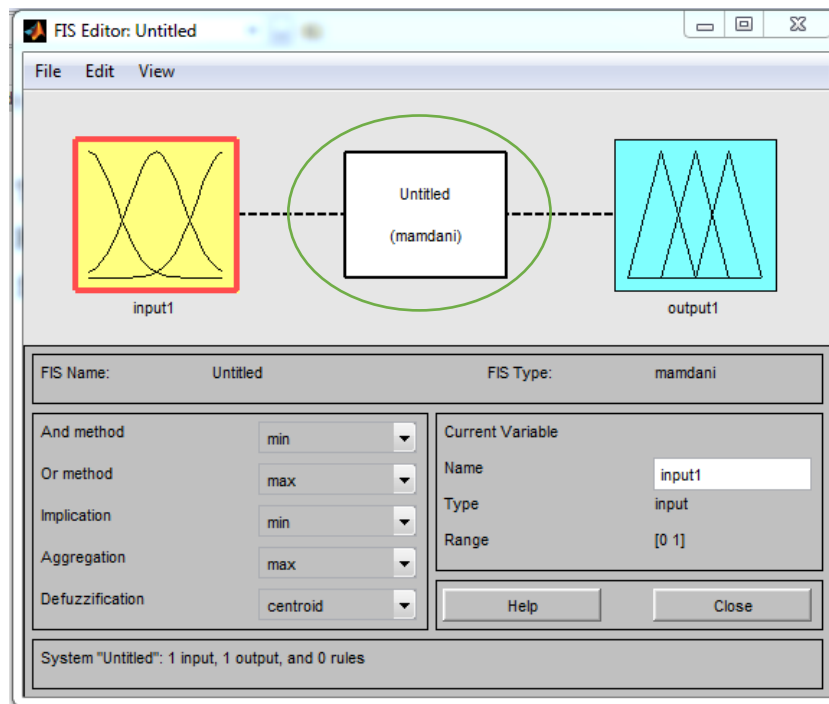
Van számos default paraméter, név, mechanizmus, amikkel az a baj hogy nagyon általánosak. Figyelni kell arra hogy ezeket mindet átnevezzük, különben problémákba fogunk ütközni.

Ökölszabály: Amit el lehet nevezni azt, nevezzük is el.

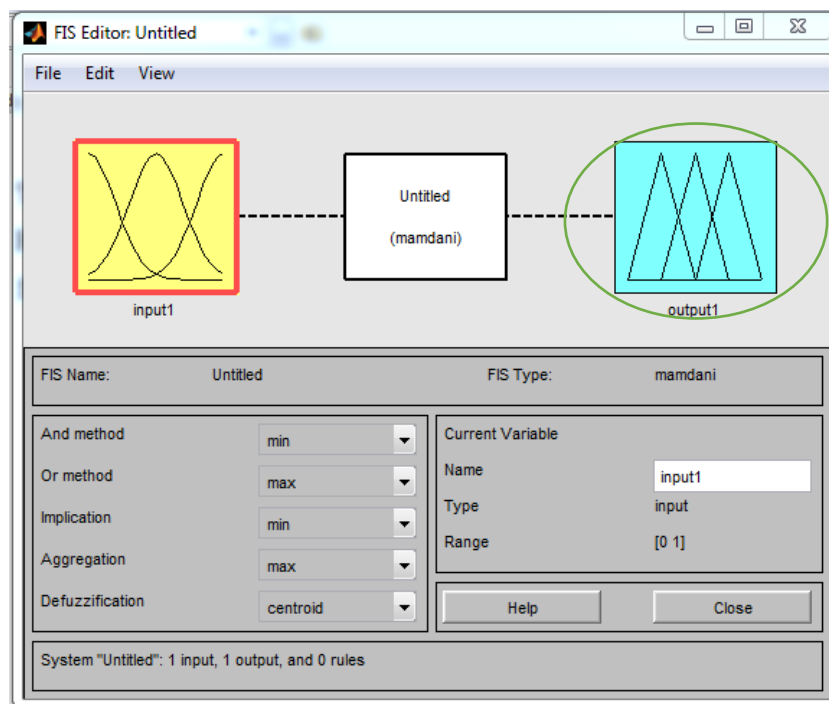
Az alap Fuzzy Logic Designer-ünk egy egyszerű egy bemenetű egy kimenetű Fuzzy szabályzót hoz létre.



A szabályzó bemenete



A szabálybázis és a következtető rendszerünk



A kimenet

Az alap szabályzót lehet bővíteni:

- Edit -> Add Variable -> Input, Output

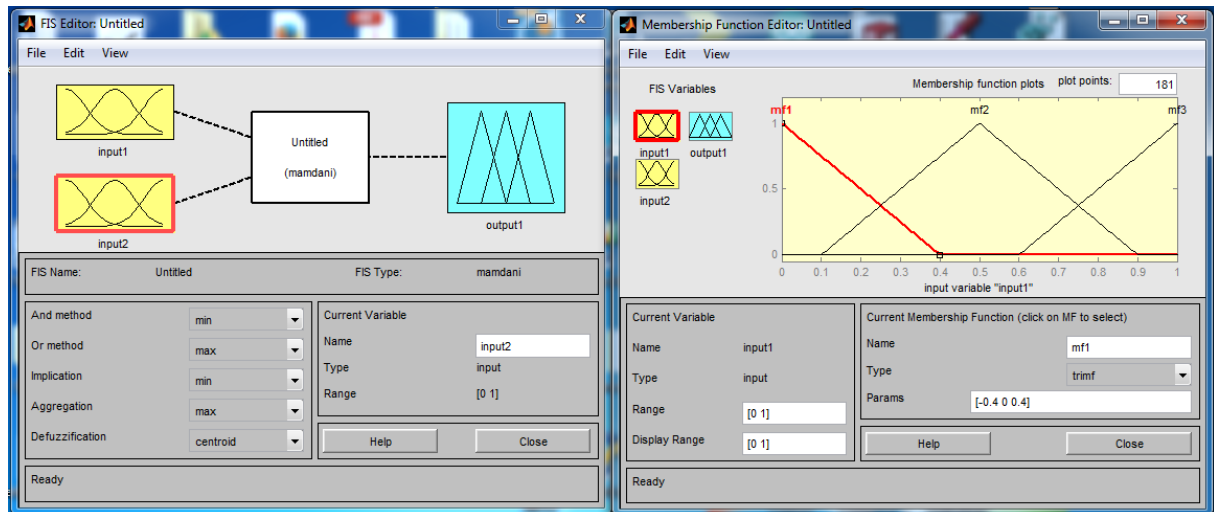
Változó eltávolítása

- Edit -> Remove Selected Variable
- vagy kijelöltjük és delete

Az Edit menüpontban az utolsó két lehetőség az attól függ, hogy a konfigurátor melyik ablakában vagyunk. Az ablak előtt és utáni 1-1 ablakot lehet a segítségével megnyitni.

Az Edit menüpontban található még az Undo, azonban ez maximum egy lépést tud visszavonni.

A bemenetet, kimenetet vagy szabálybázist kijelölni egérrel és kattintással lehet. A default szelekciók megmaradnak a következő ablakban is:



Ha hozzáadunk a Fuzzy szabályzóhoz egy új bemenetet és hiába van az input2 kiválasztva a tagsági függvény szerkesztő ablakban mégis az input1 van default-ként kijelölve. A tagsági függvény szerkesztő akkor is előjön, ha duplán kattintunk a bemenetekre vagy a kimenetekre.

A File menüpont alatt az alábbiak találhatóak:

- File -> New FIS -> Mamdani, Sugeno: ennek segítségével egy új Fuzzy szabályzót hozhatunk létre vagy Mamdani vagy pedig Sugeno félélt
- File -> Import -> From Workspace, From File: ennek segítségével változókat importálhatunk
- File -> Export -> From Workspace, From File: ennek segítségével változókat exportálhatunk

Ha az 5-ből bármelyik ablakot, amit utoljára nyitottunk meg bezárjuk, akkor feldobja, hogy akarok-e menteni. A Fuzzy szabályzót csak a bezáráskor lehet elnevezni.

Nézzük meg egy konkrét feladatot, amely legyen a borraivaló autómata, amely segít eldönteni hogy mekkora borraivalót adjunk. A borraivaló mértékét a kiszolgálás ideje (várakozás) és a felszolgált étel minősége (minőség) alapján határozhatjuk meg.

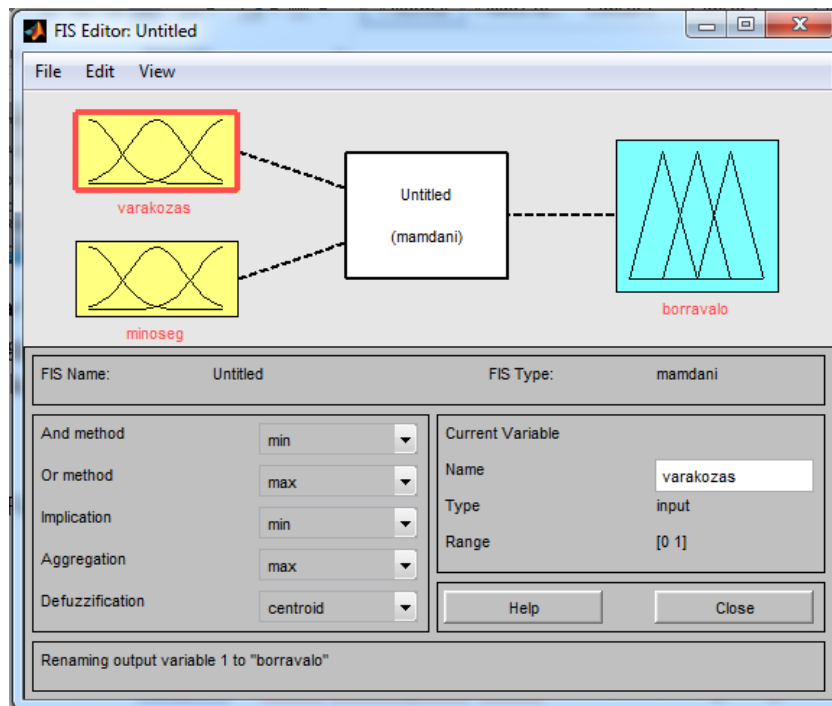
Várakozás: 0....10 (0- hosszú várakozás, 10 rövid várakozás)

Minőség: 0....10 (0 – rossz étel, 10 jó étel)

Ez alapján maximum 10% borraivalót lehet adni

Borraivaló: 0....10

Csináljuk meg ezt a szabályzót. Nevezzük el a be és a kimeneteket. Ne használjunk ékezeteket és alulvonást sem (alsó indexet csinál belőle)



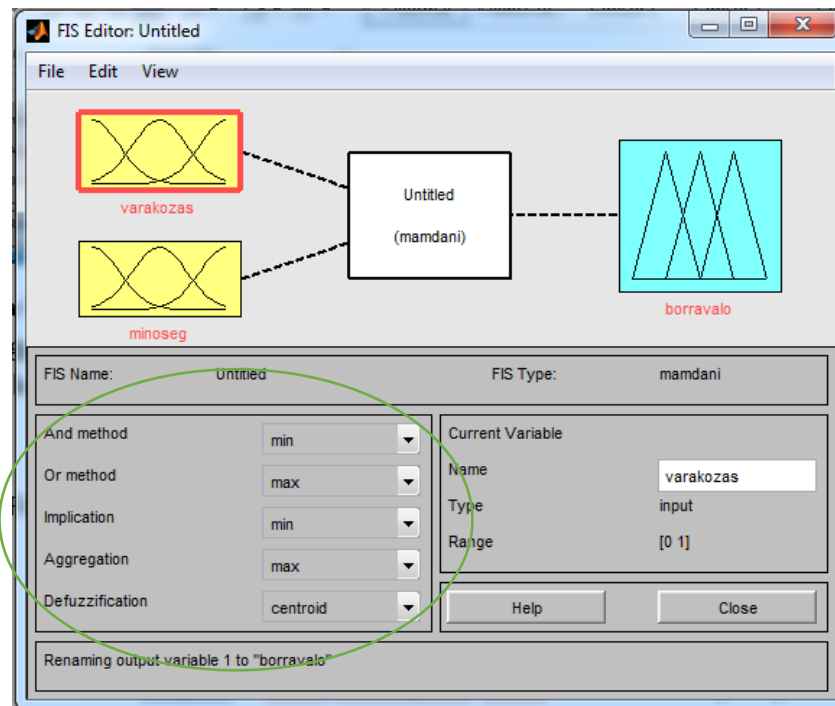
A Mamdani és a Sugeno szabályzó közötti különbség nem nagy, de annál fontosabb. Igazából csak a kimenetük különbözik.

A Mamdani típusú szabályzó kimenete halmazhoz tartozási függvényekből áll, addig a Sugeno típusú Fuzzy szabályzó kimentére nem halmazhoz tartozási függvényt kell létrehozni, hanem nagyon fontos hogy polinomot. A kimenetre tehát polinomot kell definiálnunk. A polinomnak lehet különböző fokszáma. Lehet:

- 0-ad rendű polinom azaz konstans
- 1 rendű polinom azaz $ax+b$. Ez a kimeneti polinom a bemenet függvénye lesz. Több bemenet esetén a bemenetek lineáris kombinációi lehetnek az egyes kimenetek
- 2 rendű polinom

Ami még fontos a két szabályzó esetében az a használati mód. A Mamdani Fuzzy szabályzó esetében nagyon egyértelmű hogy hogyan kell felépíteni. Nagyon intuitív, könnyű elképzelni, logikus. A probléma az, hogy ahhoz hogy precízen számolható legyen olyan defuzzifikáló eljárás (centroid, center of gravity) kell, ami nagyon számításigényes. Tehát túlságosan számításigényes egy jól működő Mamdani Fuzzy szabályzó működtetése. Ennek egyik megoldása, hogy a felhasználó megtervezi a Mamdani Fuzzy szabályozót, amit ért és tudja karbantartani, majd következő lépésben csinál olyan Sugeno típusú Fuzzy szabályozót, ami ugyanazt csinálja (legenerálható a megtervezett Mamdani szabályzóból a Sugeno) mint a Mamdani típusú azzal a különbséggel, hogy ez a Fuzzy szabályzó már tud futni nagyon hatékonyan számolva hadveren. Mivel egy polinom van, a kimenetén nem kell hozzá trükkös defuzzifikáló eljárás hisz a polinomot könnyű kiszámolni.

Most nézzük meg a megvalósításhoz kötődő beállításokat



Ezek az alap Fuzzy műveletek beállítása. Ezek nélkülözhetetlenek ahhoz, hogy matematikailag végigszámolható legyen egy Fuzzy szabályozó a bemenetétől a kimenetéig.

And method: Itt meg lehet mondani, hogy az ÉS metódus, azaz a logikai ÉS (Fuzzy logikai ÉS) matematikai nyelven mi legyen, azaz hogy ezt matematikailag hogyan kell kiszámolni.

Or method: Itt meg lehet mondani, hogy az VAGY metódus, azaz a logikai VAGY (Fuzzy logikai VAGY) matematikai nyelven mi legyen, azaz hogy ezt matematikailag hogyan kell kiszámolni.

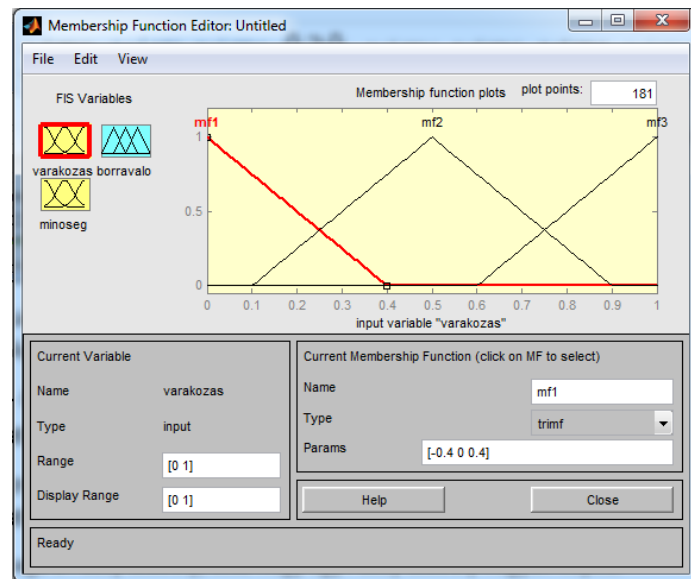
Implication: Az implikáció művelete az maga a Fuzzy szabály, azaz hogy egy Fuzzy szabály hogyan képezi le a feltételét a következményre, azaz az Antecedens ágat a Konzekvensre.

Aggregation: Mivel nem csak egy szabály lehet egy Fuzzy szabályzóban, így ha már 2 van, akkor a kimenetet a két szabály két akár különböző képen próbálja meg befolyásolni. Ez a művelet fogja azt megmondani, hogy ezek az egyesével kiértékelt szabály konzekvenssek együttesen hogyan aggregálódnak rá a kimenetre.

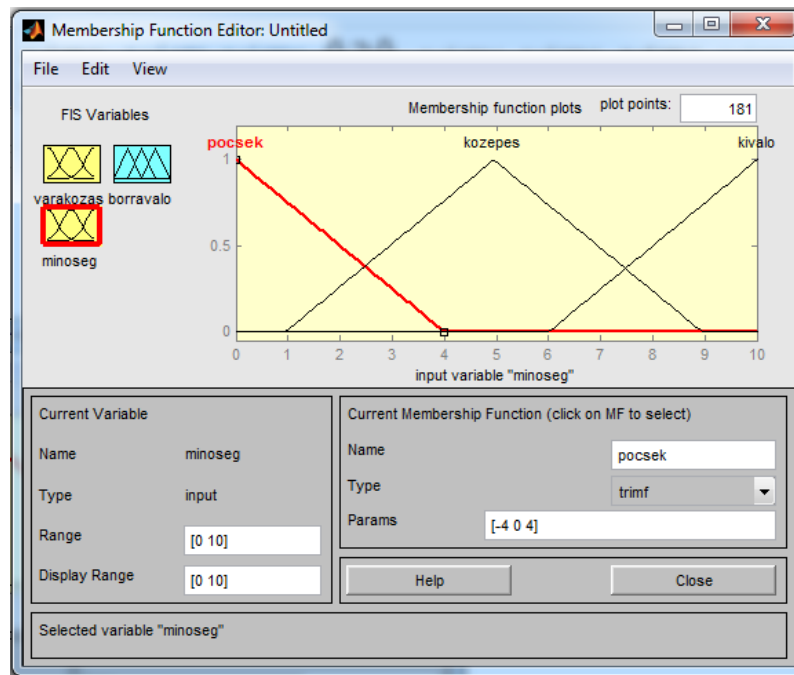
Defuzzification: A kimeneten megjelenő eredmény egy Fuzzy halmazhoz tartozási függvény ráadásul rettentő komplex és összetett. Semmiképpen nem alkalmas arra hogy ezt a világ fele továbbítsuk. A világ egy konkrét számot vár (például a motor esetében 55%-os kitöltési tényező). Ezt állítja elő a centroid vagy a center of gravity eljárás.

Nagyon egyszerű matematikai alpműveletek közül lehet választani. Ezek az alpműveletek a Zadehi alpműveletek.

Indítsuk el a Membership Function Editor-t és állítsuk be a példának megfelelően:



Alaphelyzet

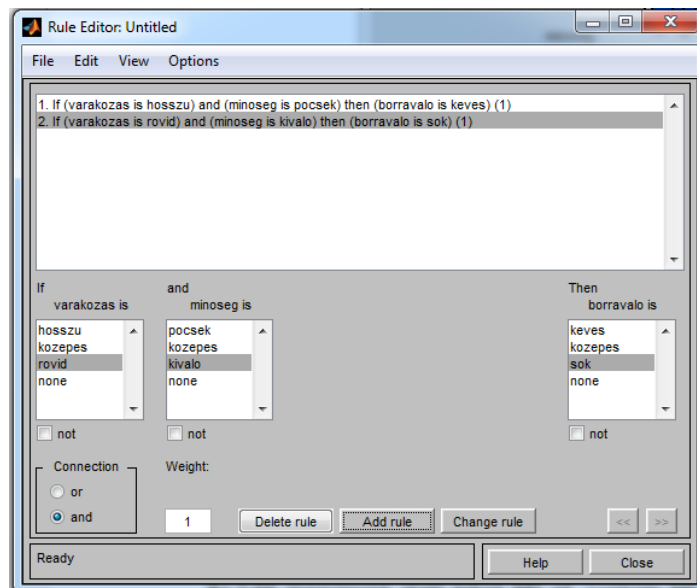


Kitöltve (range és a halmazhoz tartozási függvények átnevezése)

A display range segítségével csak nagyítani vagy kicsinyíteni lehet. Ki lehet választani a típusát is a tagsági függvénynek, de most ezeket ne változtassuk.

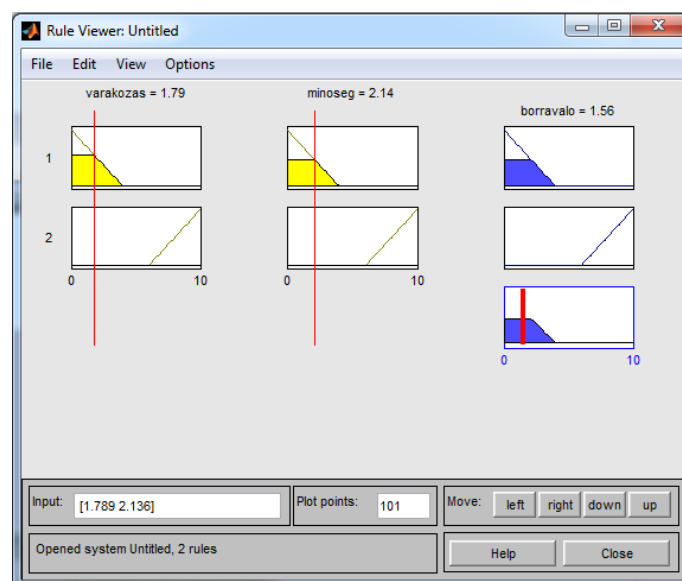
Az Edit menüpont alatt lehet MF-et, custom MF-et létrehozni (egyenlő szárú háromszög) vagy egy, vagy több MF-et törölni.

Ezután nyissuk meg a szabályszerkesztőt és adjunk a rendszerhez két szabályt:



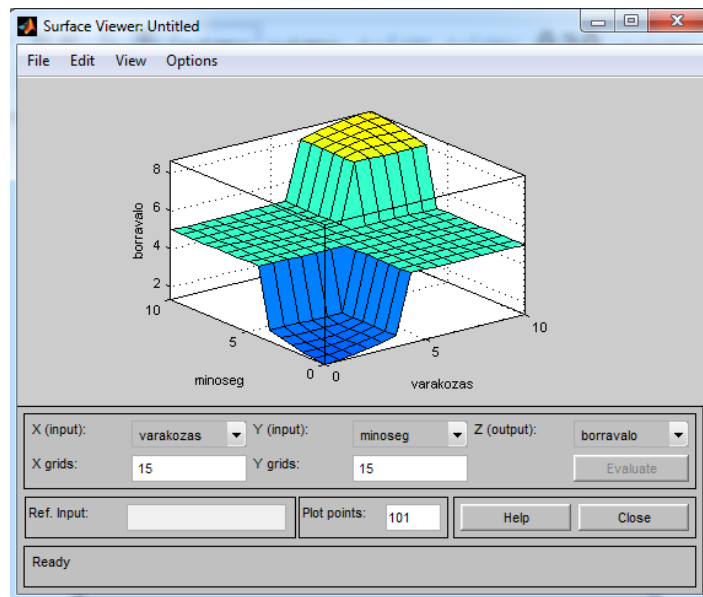
2 szabály

Mindig and-et használjunk, de használjunk súlytényezőket és ne használjuk a kimeneten a negálást, mert matematikailag van értelme ugyan, de a valóságban nincs (motor fordulát például ne 300 legyen). A none segítségével nem vesszük figyelembe az adott változót.



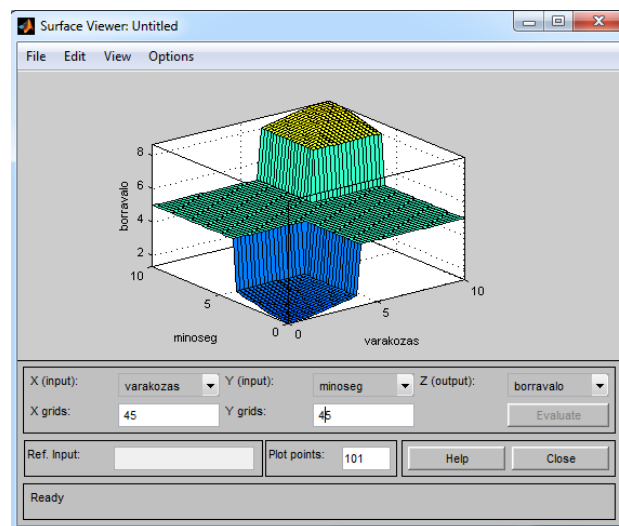
Rule Viewer

A fenti ablak egy klasszikus debug ablak. A piros csíkkal meg lehet adni hogy milyen bemenetünk, van és megnézhetjük hogy ennek hatására milyen kimenet keletkezik. Az alsó részen ;-el elválasztva pontosan be lehet írni.



Felület 2 szabály esetében

Ennek segítségével is lehet debugolni. Van pár beállítási lehetőség. Az X grid és az Y grid-et érdemes nagyra állítani, így sokkal finomabb lesz a beosztása. Legyen ez most 45, 45.



Grid hatására finomabb lett a beosztás

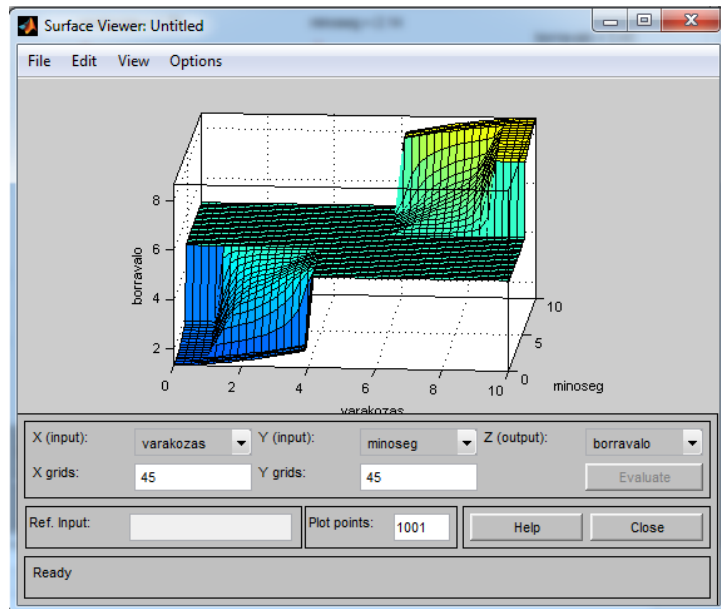
A plot points beállítás határozza meg a belső numerikus integrálás felbontását. A legkisebb a 100, ennél kisebbet is megadhatunk, de belül valójában nem lesz kisebb 100-nál. Ha viszont nagy értéket adunk meg akkor sokkal pontosabban közelíti numerikusan az adott feladatot.

A fenti surface az lényegében egy átviteli függvény. A Fuzzy szabályzónak nincs belső állapota, mindig egy adott bemenet hatására állít elő egy kimenetet. Ebből következik, hogy nagyon egyszerűen lehet az esetében átviteli függvényt vizsgálni. Leolvasható hogy melyik input kombinációban milyen kimenetet fog adni.

Látható hogy itt még vannak érdekességek. A függvényen vannak nagy platók, és vannak is benne elég nagy lépések. Ha a Fuzzy szabályzó az adott bemeneti vektorra nem értelmezett akkor még muszáj a MATLAB Fuzzy szabályzónak a kimenetére kiadni valamit. Ez a valami, amit kiad az nem 0 lesz. Ha nem

tudjuk, hogy mit kell csinálni akkor azt kell csinálni, amivel a lehető legkisebb kárt tudjuk okozni. Ez pedig akkor van ha a kimeneti range-ünknek a közepét állítjuk be kimenetnek.

Látható hogy nem minden bemeneti kombináció van lefedve. A szabály szerkesztőbe vigyük fel a közepes-közepes-közepes szabályt is:

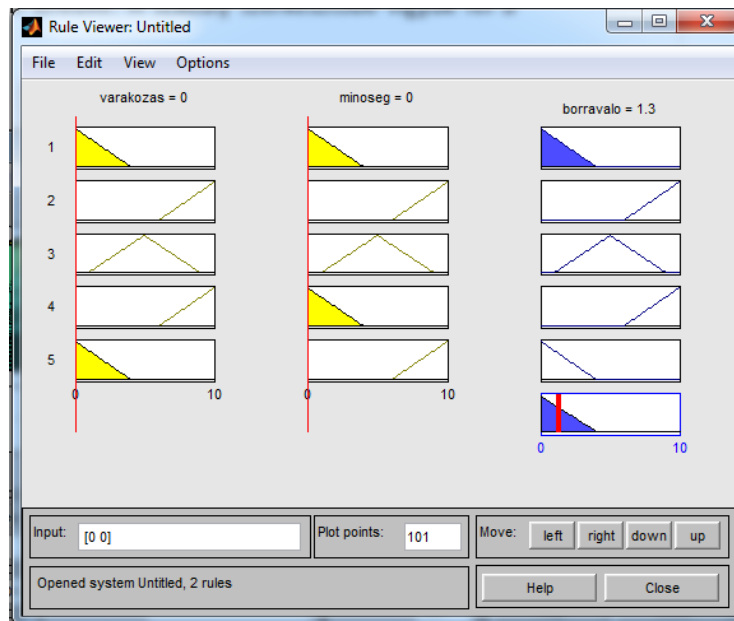


Meglapul az átviteli függvény

Az átviteli függvény akkor jó, mindentől függetlenül, ha az mindenhol folytonos. Ha nem akkor az a probléma hogy ha a rendszerünkben zaj van akkor a kimenet ugrálni fog. Például ha a minőség és a várakozás is 4.9 körül van akkor 2% borravalót adunk. De ha kicsit is változik az egyik bement, akkor már rögtön 7.5% lesz a borravaló. Tehát ha kicsit változik, a bemenet akkor nagyot változhat a kimenet. Ez az instabil rendszer tipikus ismérve.

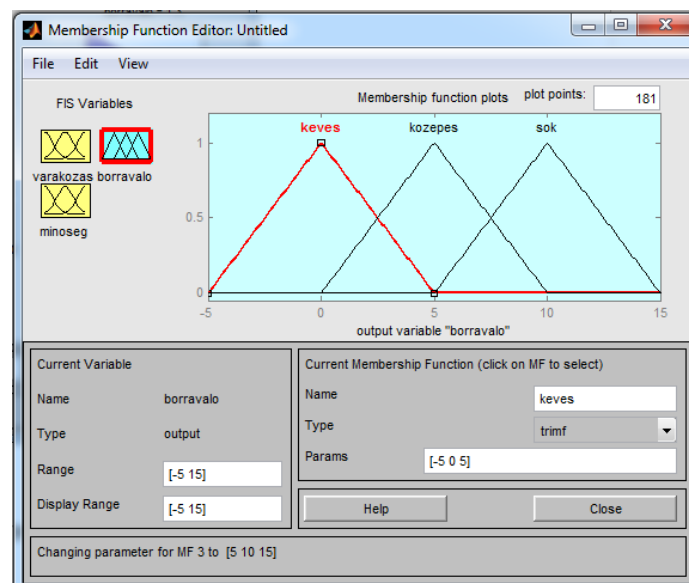
A bemenetet pedig próbáljuk lefedni. Ebben is segít a surface viewer.

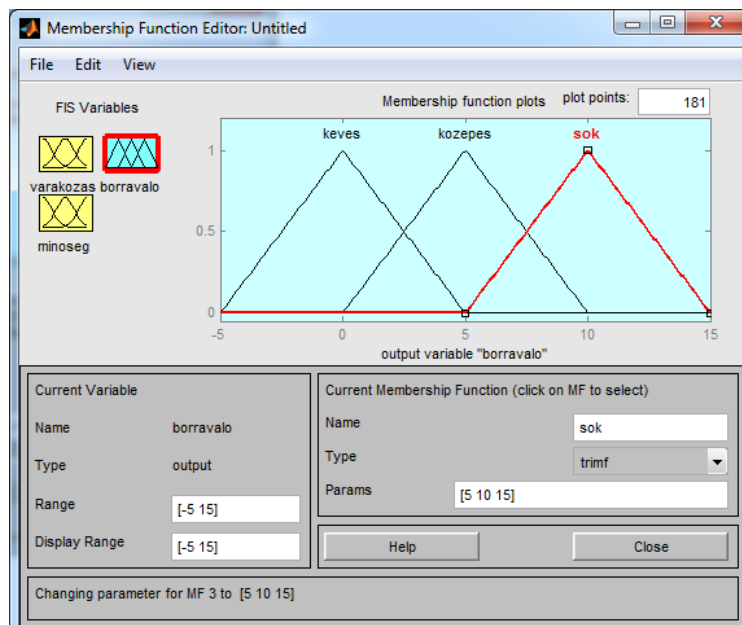
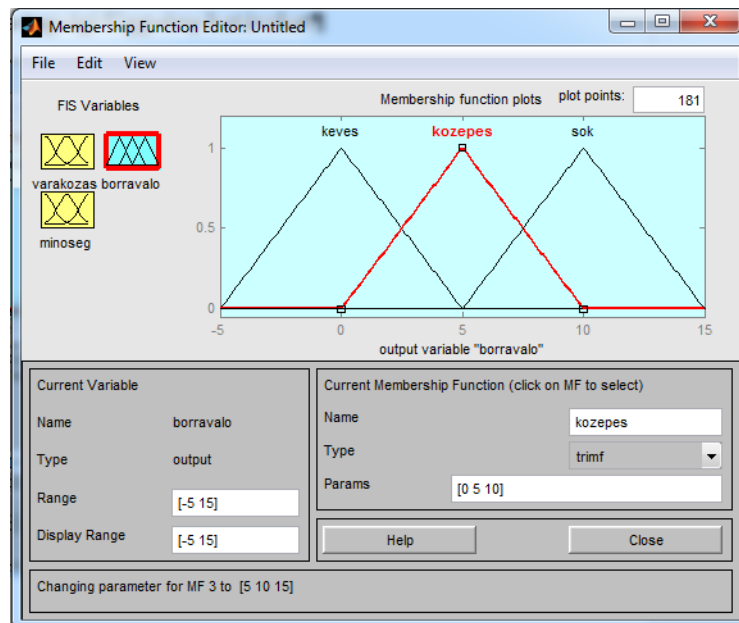
Nem lehet 0 borravalót adni:



Hogy lehet ezt kijavítani? Van erre is egy ökölszabály hogy a kimeneti halmaz függvényeknek a range-be bele kell férnie.

Állítsuk át a kimeneten a rang-et [-5 15]-re és állítsuk be a tagsági függvény határokat





Extra szabályokat is fel lehet vinni.