I. Podstawy teorii modelowania systemów rozmytych

I.1. Historia postępu i stan aktualny

Ponad trzydzieści lat minęło od sformułowania podstaw teorii zbiorów rozmytych przez Lotofil Zadeha z Uniwersytetu Kalifornijskiego[1]. W trakcie trwania ostatniego ćwierć wiecza teoria ta się bardzo rozwinęła.Wstępując zarówno do wielu działów matematyki, techniki, i co jest najbardziej zadziwiające również do ekonomii i wielu innych dziedzin. To, że teoria zbiorów rozmytych jest przydatna jest widoczne patrząc na dowody w postaci zastosowania ich w technice sterowania i w systemach ekspertowych o romaitym przeznaczeniu, np. w technologii maszyn. Najbardziej spektaktularnym osiągnięciem ostatnich lat w dziedzinie sterowania rozmytego jest zaprojktowanie oraz zrobienie modelu helikoptera, który był bezzałogowy. Reagował on na rozmyte rozkazy, które były przekazywane drogą radiową. Poza tym do dnia dzisiejszego w przemyśle samochodowym również zastosowano sterowanie rozmyte, dotyczących automatycznych przekładani biegów.

I.2. Podstawowe pojęcia i teoria zbirów rozmytych

Obserwując cały rozwój logiki, od czasów starożytnych do pierwszych dziesięcioleci obecnego wieku, można tą podstawową gałąź matematyki jaką jest logika, utożsamić z log iką dwuwartościową, mimo tego, że zgadnienia logiki wielowartościowej były wówczas znany[2]. Miało to związek z tendencją do ścisłego i precyzyjnie sformalizowanego opisu wszelkich obiektów jak i pewnej niechęci dla niejednoznaczności.

Aby poprawnie opisać zbiór rozmyty można stosować różne aspekty. Przyjmuje się, że w zbiorze rozmytym granicy ostrej nie ma pomiędzy elementami, które należą do zbioru, a tymi, które należą do dopełnienia. W takim razie określa się stopień przynależności elementów do zbioru. Poto aby móc scharakteryzować zbiór rozmyty poprzez funkcje przynależności, która przyjmuje wartości [0,1].

Za pomocą zbiorów rozmytych możemy formalnie określić pojęcia nieprecyzyjne i wieloznaczne, takie jak „wysoka temperatura”,”młody człowiek”, „średni wzrost” lub „duże miasto”. [3]Przed podaniem definicji zbioru rozmytego musimy ustalić tzw. obszar rozważań,w przypadku pojęcia wieloznacznego „dużo pieniędzy” inna suma będzie uważana za dużą, jeżeli ograniczymy się do obszaru rozważań [0.1000 zł], a inna – jeżeli przyjmiemy przedział [0;1000 000 zł]. [4]Obszar rozważań nazywany jest w dalszym ciągu zbiorem lub przestrzenią, oznaczymy go jako X, pamiętając że X jest zbiorem nierozmytym.[5]

Definicja [6]

Zbiorem rozmytym A w pewnej (niepustej) przestrzeni X definiowany przez pary

F={u, $\mu_F(u) \mid u \in U$ }[7]

w którym μa:X->[0.1] jest funkcją przynależności zbioru rozmytego A[8]. W odróżnieniu od

---

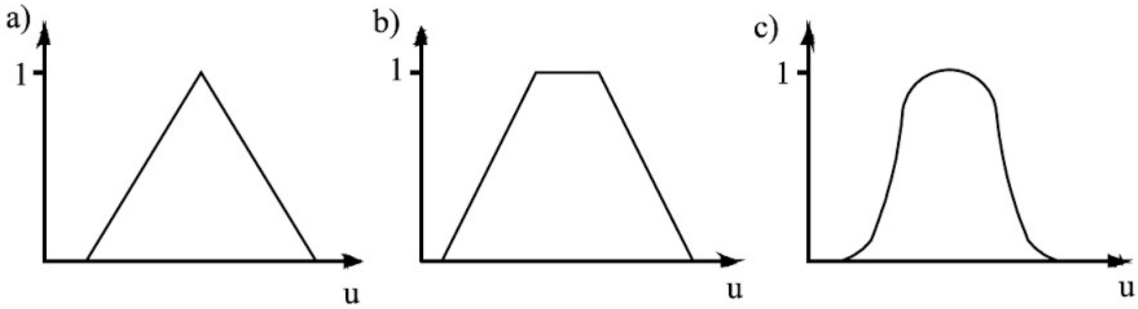
1) Zadehl A. Fuzzy sets;Informationand Control, Vol.8,pp.338- 53 1965
2) ReschеrN. : Many-ValuedLogic; New York, McGraw-Hill, 1969
3) Rutkowski L.: Metody i techniki sztucznej inteligencji
4) Rutkowski L.: Metody i techniki sztucznej inteligencji
5) Rutkowski L.: Metody i techniki sztucznej inteligencji
6) Rutkowski L.: Metody i techniki sztucznej inteligencji
7) *www.mimuw.edu.pl/~son/sysdec/materials/Sterowanie%20**rozmyte**.pdf*
8) *kolegia.sgh.waw.pl/pl/KZiF/publikacje/koik/.../2013_NR_5(158).pdf*

klasycznego podejścia teori zbiorów, która mówi o funkcji przypisującej dwie wartości {0,1}, w zbiorach rozmytych wyróżniamy trzy przypadki:

μa(x)=1 – pełną przynależność do zbioru rozmytego A[9]

μa(x)=0 – brak przynależności elementu x do zbioru rozmytego A[10]

0<μa(x)<1 – częściowa przynależność elementu x do zbioru rozmytego A[11]



rys. I.2./1Różne formy funkcji przynależności: a) trójkątna, b)trapezoidalna, c) kształt dzwonu[12]

I.3. Operacje na zbiorach rozmytych

Poniżej zostaną opisane najważniejsze operacje na zbiorach rozmytych. Wśród wielu liczby operacji można wyróżnić operacje mnogościowe oraz algerbaiczne.

Z operacji mnogościowych wymienić trzeba następujące operacje:

- jednoargumentowa operacja dopełnienia zbioru rozmytego

- dwuargumentowa operacja sumy mnogościowej dwóch zbiorów

$A, B, C \in X$

$C = A \vee B$

$\forall x \in X:$

μc(x) = max[$\mu_A(x), \mu_B(x)$]

- przecięcie

$A, B, D \in X$

D= $a \wedge b$

$\forall x \in X:$

$\mu_D(x) = min \ [\ \mu_A(x), \mu_B(x) \ ]$

Definicje, które są zawarte wyżej nadają zbiorowi podzbiorów rozmytych zbioru X strukturę, podobną do algebry Boolea, jednak jest ona inna , poprzez brak obowiązywania fundamentalnych zasad teorii klasycznych:

- zasada sprzeczności, bo: $A \cap \bar{A} \neq \emptyset$ [13]

9) *www.mimuw.edu.pl/~son/sysdec/materials/Sterowanie%20**rozmyte**.pdf*
10) *www.mimuw.edu.pl/~son/sysdec/materials/Sterowanie%20**rozmyte**.pdf*
11) *www.mimuw.edu.pl/~son/sysdec/materials/Sterowanie%20**rozmyte**.pdf*
12) home.agh.edu.pl/~ligeza/wiki/_media/ke:fuzzy-control-tecza.pdf?...

13) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*

- zasada łączonego ośrodka, ponieważ: $A \cup \bar{A} \neq \emptyset$ [14]

Oprócz operacji mnogościowych, trzeba wymienić jeszcze szereg operacji algebraicznych zbiorach rozmytych.[15] Należoną do nich operacje mnożenia i potęgowania funkcji przynależności zbiorów rozmytych. [16]Do operacji potęgowania należą: operacja koncetracji zaostrzająca zbiór rozmyty, operacja rozcieńczania spłaszczająca zbiór rozmyty oraz intesyfikacja kontrastu zbioru rozmytego.[17]

Istnieje jednak jeszcze bardzo duża ilość operacji stosowanych, niektóre własności , operacji, które zajmują ważne miejsce w zbiorach rozmytych są szczególnie interesujące:

-przemienność:

$A \cup B = B \cup A$ [18]

$A \cap B = B \cap A$ [19]

-idempotentność

$A \cup B = A$ [20]

$A \cap B = A$ [21]

-łączność

$A \cup (B \cup C) = (A \cup B) \cup C = A \cup B \cup C$ [22]

$A \cap (B \cap C) = (A \cap B) \cap C = A \cap B \cap C$ [23]

- rodzielność

$A \cap (B \cap C)(A \cap B) \cup (A \cap C)$ [24]

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ [25]

- prawa de Morgana

$\overline{A \cup B} = \bar{A} \cap \bar{B}$ [26]

$\overline{A \cap B} = \bar{A} \cup \bar{B}$ [27]

Najważniejsze pojęcia teorii zbiorów rozmytych, które docydują o przeznaczeniu w praktyce, zostaną przedstawione poniżej.

Zmienna lingwistyczna jest to wielkość, która może przyjmować wartości lingwistyczne (wartości to zdania wyrażone w języku naturalnym). Na przykład szybkość, zmienna ta może przyjmować różne wartości (inaczej zwane termami). Termy mogą być opisane numerycznie. **Przetwarzanie danych** przez człowieka oparte jest na zmiennych lingwistycznych. Zdolność przetwarzania zmiennych oraz rozumowania na ich podstawie, takie niezwykłe zdolności ma ludzki umysł. Żeby wykorzystać to trzeba jeszcze opisać zależności między zmiennymi. Można posłużyć się do tego

14) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*
15) home.agh.edu.pl/~ligeza/wiki/_media/ke:fuzzy-control-tecza.pdf?...
16) home.agh.edu.pl/~ligeza/wiki/_media/ke:fuzzy-control-tecza.pdf?.
17) home.agh.edu.pl/~ligeza/wiki/_media/ke:fuzzy-control-tecza.pdf?.
18) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*
19) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*
20) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*
21) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*
22) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*
23) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*
24) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*
25) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*
26) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*
27) *www.cs.put.poznan.pl/mhapke/OEiG1.pdf*

zdaniami warunkowymi, które opisują zależności tzw. przyczyna-skutek. Nazywane są one **regułami rozmytymi** wnioskowania. Są one typu: Jeżeli A i B to C, Ai B i C to twierdzenia rozmyte np. (a jest małe). Można tworzyć je na podstawie prób numerycznych lub na podstawie doświadczenia specjalisty. Jeżeli dana sytuacja jest opisana zestawem reguł, to tworzą one bazę reguł. Stanowi taka baza punkt wyjściowy do problemu wnioskowania, kiedy informacje nie są precyzyjnie podane lub wnioskowania przybliżonego. Wnioskowanie takie jest zautomatyzowane w systemach wnioskowania rozmytego.

I.4. Reguły sterownika rozmytego

Reguły którymi opisuje się sterownik rozmyty tworzone są następująco :

```
if <przesłanka> then <konkluzja>
```

Przesłankę tworzy się za pomocą serii predykatów połączonych spójnikami logicznymi, Konkluzję tworzą kolejne predykaty. Struktura ta przypomina swą budową bazy wiedzy znane z systemów ekspertowych, jednak wewnątrz każdej reguły "dzieje się" tu dużo więcej.

W zależności od zastosowanego modelu rozmytego możemy spotkać się z różnymi formami predykatów. Aplikacja, która jest przedmiotem tej pracy, bazuje na modelu Mamdaniego [3]. W tym modelu predykaty są postaci :

```
A is B
```

gdzie :
A - nazwa danej zmiennej lingwistycznej
B - jeden z termów opisujących tą zmienną.

Teoria zbiorów rozmytych zakłada, że termy określające zmienną lingwistyczną mają postać zbiorów rozmytych. Zmienna jest więc zbiorem nazwanych zbiorów rozmytych, które mają nazwy wzięte z języka naturalnego. Predykaty w tej postaci może wyglądać tak :

```
temperatura is niska
```

w miejsce niska możemy wstawić też np. wysoka, śerdnia. Definiujemy w ten sposób kolejne termy ze zbioru termów zmiennej temperatura. Termom tym przypisane są pewne zbiory rozmyte które opisują małą, średnią i wysoką temperaturę.

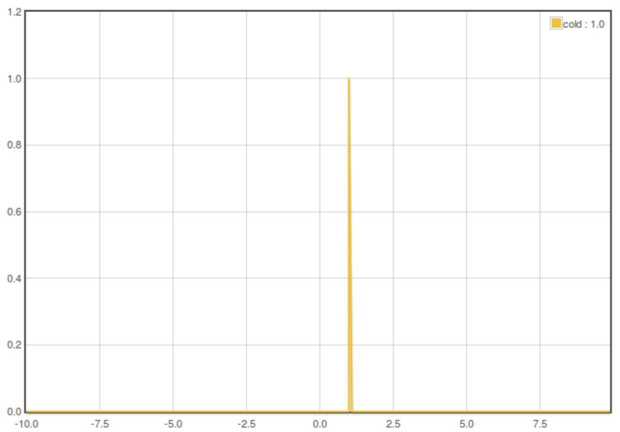
Zmienne lingwistyczne które występują po lewej stronie reguły nazywamy zmiennymi wejściowymi sterownika. Natomiast zmienne znajdujące się w konkluzji to zmienne w-wyjściowe. Do wnioskowania rozmytego jako podstawę użyto uogólnioną na teorię zbiorów rozmytych regułę *modus ponens*. Zachowano element, który określał, że prawdziwość przesłanki implikacji pozwala wnioskować o prawdziwości konkluzji. Dodano pojęcie stopnia prawdziwości albo stopnia spełnienia zarówno przesłanki jak i konkluzji.

I.5. Bloki i proces wnioskowania rozmytego

I.5.1. Blok rozmywania

Bazując cały czas na modelu Mamdaniego wyróżniamy kolejne etapy(bloki) wnioskowania rozmytego. Pierwszym z nich jest blok rozmywania. Blok ten przekształca wartość zmiennej wejściowej (zmiennej lingwistycznej) na stopień spełnienia predykatów - przesłanek reguł, które angażują tę zmienną.

Najczęściej spotykaną i najprostszą obliczeniowo metodą, która realizuje te cele jest metoda typu singleton. Polega ona na utworzeniu funkcji - zbioru rozmytego X następującej postaci :



rys. I.5.1./1. Singleton

Który zdefiniowany jest funkcją :

$$f(x) = \left\{ \begin{matrix} 1 : x = C \\ 0 : x \neq C \end{matrix} \right\}$$

Następnie jako wartość spełnieniaa predykatu przesłanki A is B uznaje się zbiór powstały w wyniku przecięcia zbioru X ze ze zbiorem skojarzonym z termem B. Jest to prosta metoda w wyniku której zbiorem wynikowym predykatu przesłanki jest liczba mówiąca o stopniu aktywacji tego predykatu. Istnieje wiele innych, bardziej skomplikowanych metod rozmywania [4].

I.5.2. Blok i metody inferencji

Inferencja jest kolejnym etapem działania sterownika rozmytego. Jest ona jednocześnie najbardziej złożona obliczeniowo. Możemy wyróżnić trzy etapy prac tego bloku :

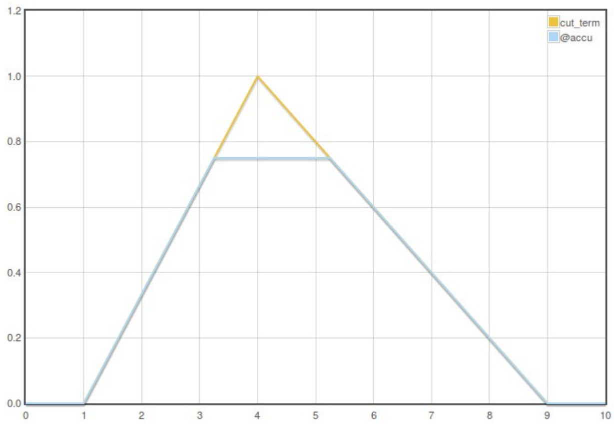
- agregacja (aggregation)
- wnioskowanie
- kumulacja (accumulation)

Wszystkie te etapy wykonywane są w oparciu o przygotowaną przez eksperta baze reguł. W poprzednim bloku dane zmiennych wejściowych zamienione zostały na stopnie spełnienia odpowiednich predykatów w przesłankach reguł. W tym bloku następuje uruchomienie wszystkich

reguł, których przesłanki są spełnione, wyliczenie zbioru rozmytego który jest wynikiem tych reguł oraz kumulacja wyników w każdym bloku reguł.
Na etapie agregacji stopień spełnienia każdej z reguł obliczany jest na podstawie stopnia spełnienia ich przesłanek. W tym celu używane są logiczne operatory rozmyte znane z logiki Boole'a : AND, OR oraz NOT. W zastosowanej przeze mnie implementacji języka Fuzzy Control Language dostępne są następujące funkcje (t-normy) dla tych spójników :

	AND	OR
MIN	$\mu_{A,B}(x,y) = min(\mu_A(x),\mu_B(y))$	$\mu_{A,B}(x,y) = max(\mu_A(x),\mu_B(y))$
PROD	$\mu_{A,B}(x,y) = \mu_A(x)\cdot\mu_B(y)$	$\mu_{A,B}(x,y) = \mu_A(x)+\mu_B(y)-\mu_A(x)\cdot\mu_B(y)$

Te reguły dla których stopień spełnienia zagregowanych przesłanek jest niezerowy zostają aktywowane. W wyniku wnioskowania obliczany jest zbiór rozmyty stanowiący konkluzję danej reguły. Operatorem implikacji jest operator MIN, a polega to obrazowo na "obcięciu" zbioru termu wyjściowego na wysokości stopnia spełnienia zagregowanych przesłanek, na przykład :



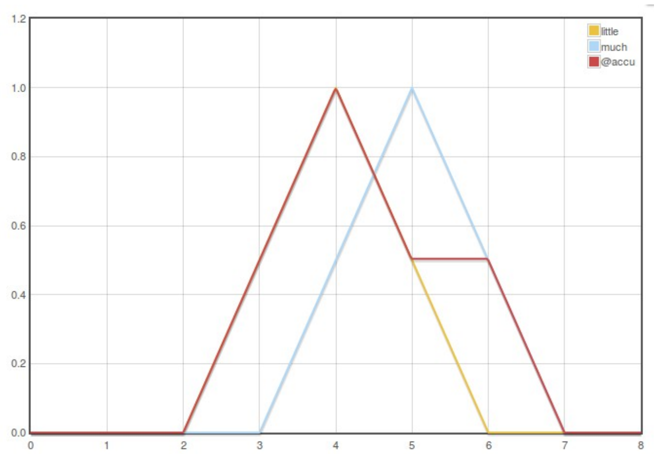
rys I.5.2./1. Przykład obciętego termu

Dodatkowo w konkluzji każdej reguły możemy podać kilka predykatów dotyczących różnych lub nawet tej samej zmiennej, rozdzielając je operatorem AND. Użycie operatora o tej nazwie podyktowane było zachowaniem czytelności reguł, nie ma on żadnego znaczenia pod względem matematycznym.

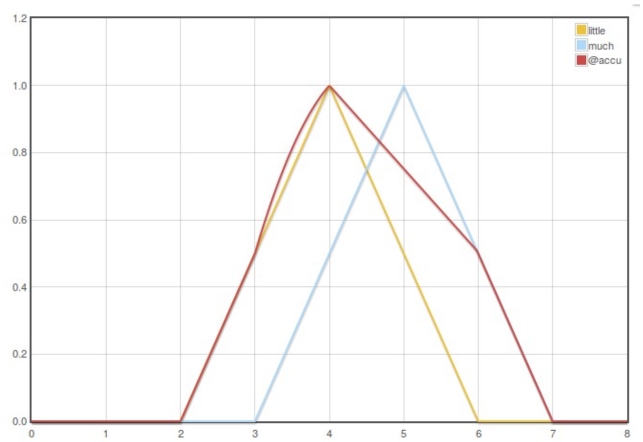
Wynikowe zbiory rozmyte kumulują się w ramach danego bloku funkcji sterownika w zbiór rozmyty, który poddawany jest następnie procesowi wyostrzania. Kumulacja polega na zebraniu wszystkich wynikowych zbiorów rozmytych danej zmiennej w jeden zbiór za pomocą funkcji kumulacji. Funkcje kumulacji dostępne w implementowanym języku FCL :

	ACCU
MAX	$\mu_{A,B}(x,y) = max(\mu_A(x),\mu_B(y))$
PROD	$\mu_{A,B}(x,y) = \mu_A(x)+\mu_B(y)-\mu_A(x)\cdot\mu_B(y)$

Obie metody nieco się różnią kształtami zbiorów wynikowych np.:



rys. I.5.2./1 akumulacja metodą MAX



rys I.5.2./2 Akumulacja metodą PROD

Różnice te nieznacznie wpływają na rezultat działania całego systemu.

I.5.3. Blok wyostrzania.

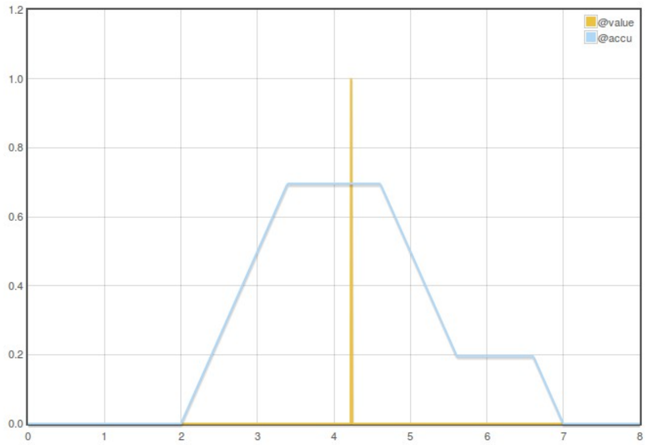
Wyostrzanie ma na celu przekształcenie wynikowego zbioru rozmytego na określoną wartość rzeczywistą stanowiącą wartość wyjścia modelu.

W tworzonym systemie możliwymi operatorami są :

- Środek ciężkości (CoG -center of gravity).
- Środek ciężkości dla singletonów (CoS - center of singleton)

	WZÓR

COG	$$v = \frac{\int_{v_{min}}^{v_{max}} v\mu_v(v)\,dv}{\int_{v_{min}}^{v_{max}} \mu_v(v)\,dv}$$
COS	$$v = \frac{\sum_{t\in S} \mu_t(t)\,t}{\sum_{t\in S} \mu_t(t)}, \text{gdzie } S - \text{zbiór zagregowanych singletonów}$$



Rys. I.5.3./1 Wyostrzanie metodą COG

I.6. Przykład wnioskowania rozmytego – zraszanie trawnika

I.6.1. Zbieranie danych

Załóżmy, że mamy trawnik o sporej powierzchni, który codziennie w miarę potrzeb podlewa sztab ogrodników. W celu zmniejszenia kosztów zatrudnienia chcemy utworzyć system, który będzie automatycznie zraszał nasz trawnik i będzie to robił w zależności od aktualnych warunków pogodowych – wilgotności gruntu i temperatury. W tym celu rozmieszczamy czujniki temperatury, wilgotności i nasłonecznienia w strategicznych punktach trawnika, tworzymy prosty system filtrujący błędne dane (powstałe w przypadku awarii jednego z czujników) i stajemy przed problemem implementacji właściwej aplikacji sterującej zraszaniem. Przy projektowaniu takiej aplikacji, w której zastosujemy logikę rozmytą, możemy skorzystać z pomocy jednego z ogrodników, który do tej pory pracował w naszym gospodarstwie. W tym celu prosimy go o spisanie zasad, którymi kierował się podlewając trawnik. Otrzymujemy od niego mniej więcej taki dokument:

Jeżeli temperatura jest bardzo niska to nie podlewamy trawnika
Jeżeli wilgotność jest duża i temperatura jest niska to nie podlewamy trawnika
Jeżeli wilgotność jest duża i temperatura jest średnia to nie podlewamy trawnika
Jeżeli wilgotność jest duża i temperatura jest wysoka to ustawiamy słabe zraszanie
Jeżeli wilgotność jest średnia i temperatura jest niska to nie podlewamy trawnika
Jeżeli wilgotność jest średnia i temperatura jest średnia to ustawiamy słabe zraszanie
Jeżeli wilgotność jest średnia i temperatura jest wysoka to ustawiamy średnie zraszanie

Jeżeli wilgotność jest niska i temperatura jest niska to ustawiamy słabe zraszanie
Jeżeli wilgotność jest niska i temperatura jest średnia to ustawiamy średnie zraszanie
Jeżeli wilgotność jest niska i temperatura jest wysoka to ustawiamy silne zraszanie

Z racji, że pewne określenia zawarte w tym dokumencie nie do końca są jasne zbieramy kolejne informacje:
temperatura bardzo niska jest poniżej 5°C
temperatura niska to około 10°C
temperatura średnia to około 18°C
temperatura wysoka to 25 °C i więcej
dowiadujemy się też, że około to +- 8°C

wilgotność niska to poniżej 10%
wilgotność średnia to około 15%
wilgotność duża to to powyżej 20%
około to +- 4%

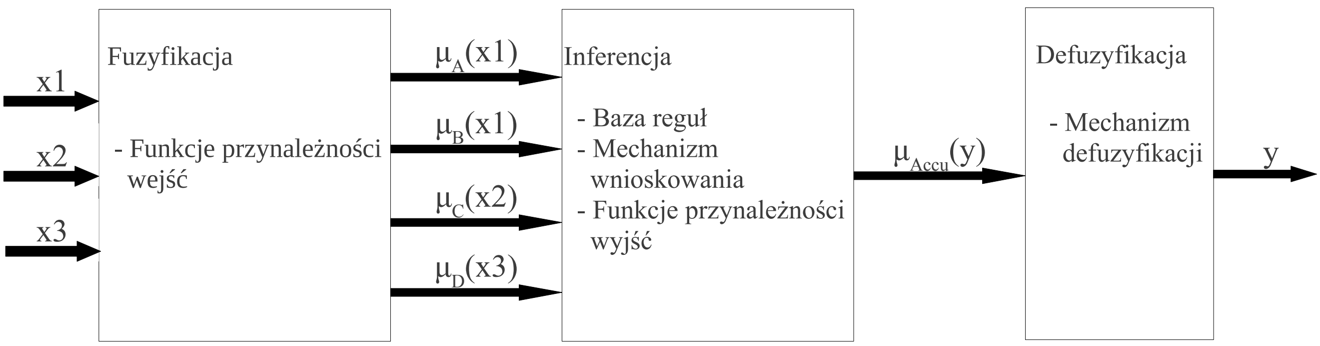
zraszanie słabe to 20%
zraszanie średnie to 40%
zraszanie silne to 60%

Większego zraszania nigdy nie ustawiali, ponieważ podmywało glebę. Z rozmowy wynika również, że sprawdzali warunki co pół godziny i dostrajali cały system, można jednak dostrajać go w trybie ciągłym, cały czas próbkując dane w małych odstępach czasu – wpłynie to pozytywnie na zużycie wody.

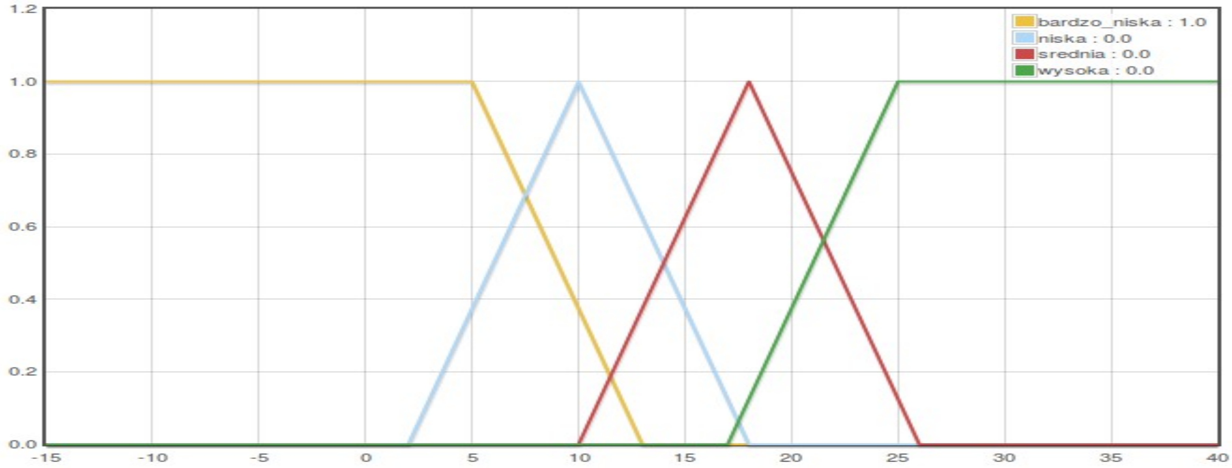
Z tak zebranymi danymi możemy przystąpić do zaprojektowania naszego systemu rozmytego.

I.6.2. Modelowanie rozmyte

Do implementacji naszej aplikacji zraszającej trawnik użyjemy modelu Mamdaniego. W ogólnym zarysie model ten prezentuje się następująco:



Z naszych danych tworzymy funkcje przynależności zmiennych wejściowych:

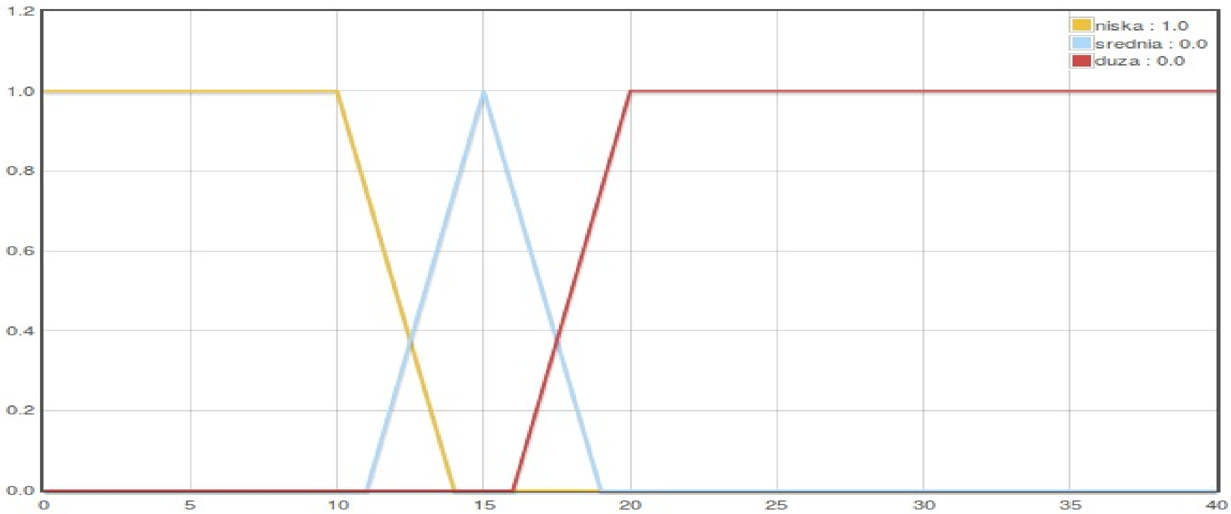


rys. I.6.2./1 Zraszanie trawnika, funkcje przynależności termów zmiennej temperatura

Przy tworzeniu zbiorów korzystamy z punktów które przedstawił nam ogrodnik opisując własne podejście do temperatury tworząc z nich kolejne termy:

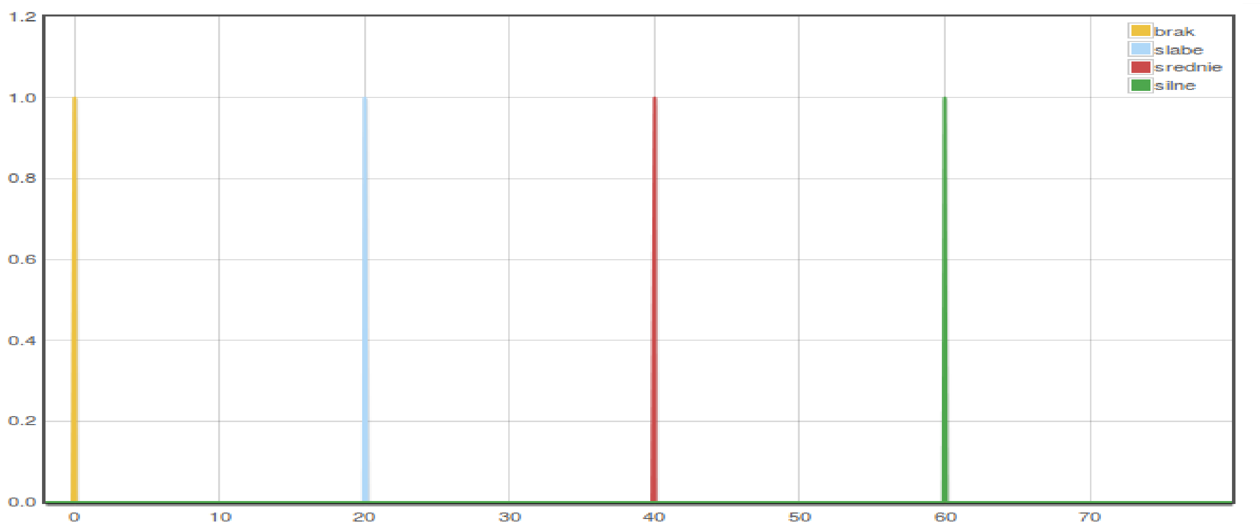
- TERM bardzo_niska := (-15,1) (5,1) (13,0);
- TERM niska := (2, 0) (10, 1) (18, 0);
- TERM srednia := (10, 0) (18, 1) (26,0);
- TERM wysoka := (17,0) (25, 1) (40, 1);

Kolejną zmienną wejściową jest wilgotność gruntu dla uproszczenia nazwana dalej wilgotnością:



rys. I.6.2./2 Zraszanie trawnika, funkcje przynależności termów zmiennej wilgotność

Pozostaje jeszcze zmienna wyjściowa zraszanie, którą opiszemy specjalnym rodzajem termów – singletonami :



rys. I.6.2./3 Zraszanie trawnika, funkcje przynależności termów zmiennej wyjściowej zraszanie

Do tego potrzebujemy jeszcze zestawu reguł, które opisał nam ogrodnik. Dla ułatwienia zapiszę je w notacji używanej w FCL:

- `RULE 1: if temperatura is bardzo_niska then zraszanie is brak;`
- `RULE 2: if wilgotnosc is duza and temperatura is niska then zraszanie is brak;`
- `RULE 3: if wilgotnosc is duza and temperatura is srednia then zraszanie is brak;`
- `RULE 4: if wilgotnosc is duza and temperatura is wysoka then zraszanie is slabe;`
- `RULE 5: if wilgotnosc is srednia and temperatura is niska then zraszanie is brak;`
- `RULE 6: if wilgotnosc is srednia and temperatura is srednia then zraszanie is slabe;`
- `RULE 7: if wilgotnosc is srednia and temperatura is wysoka then zraszanie is srednie;`
- `RULE 8: if wilgotnosc is niska and temperatura is niska then zraszanie is slabe;`
- `RULE 9: if wilgotnosc is niska and temperatura is srednia then zraszanie is srednie;`
- `RULE 10: if wilgotnosc is niska and temperatura is wysoka then zraszanie is silne;`

Tak zdefiniowane reguły i funkcje przynależności termów zmiennych wejściowych i wyjściowych opisują model Mamdaniego. Przeprowadzimy teraz przykładowy proces wnioskowania od początku do końca.

I.6.3. Rozmywanie (fuzyfikacja)

Załóżmy stan zmiennych wejściowych :

- temperatura : 20°C
- wilgotność : 16%

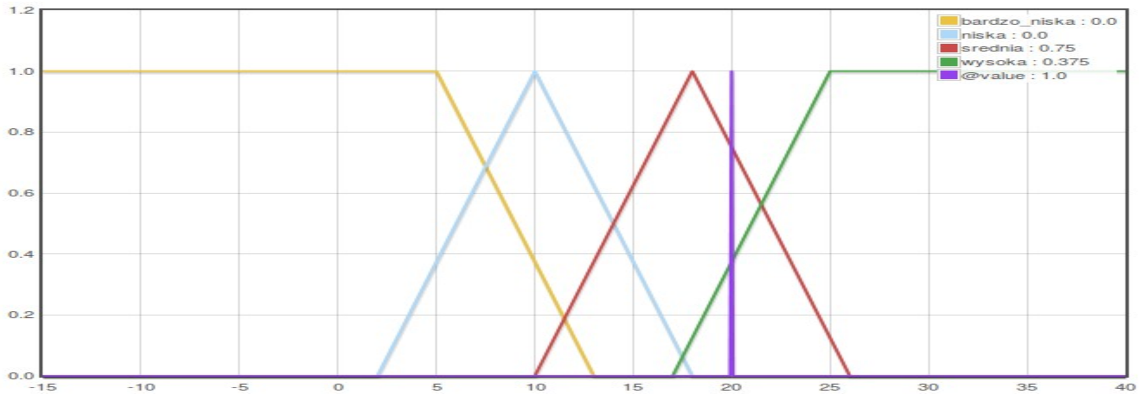
Obliczamy wartości funkcji przynależności szczególnych predykatów reguł. Otrzymujemy:

$$\mu_{TEMPERATURA\ IS\ SREDNIA}(20) = 0{,}75$$

$$\mu_{TEMPERATURA\ IS\ WYSOKA}(20) = 0{,}375$$

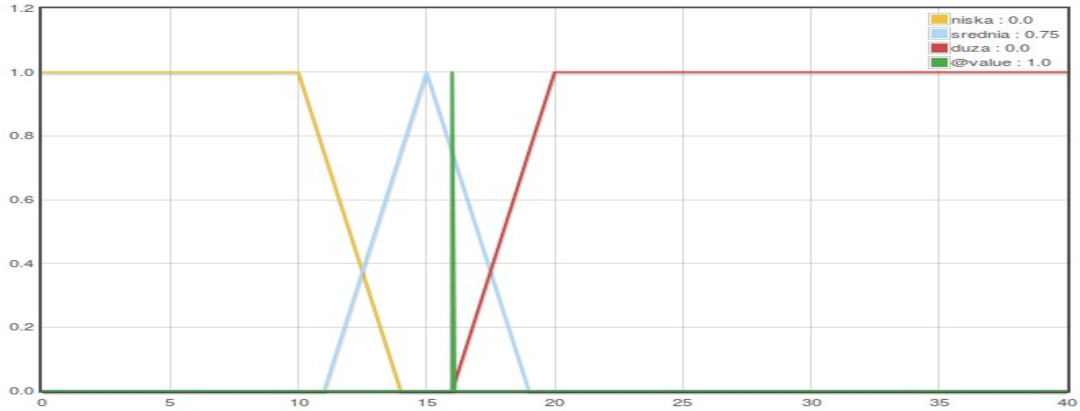
Dla pozostałych predykatów występujących w regułach wartość funkcji przynależności wynosi 0. Widać to na poniższym wykresie – przecięcie wykresów termów z fioletowym singletonem

@value:



rys I.6.3./1 Funkcje aktywacji termów zmiennej wejściowej temperatura

Fuzyfikujemy również zmienną wilgotność :



rys. I.6.3./2 Funkcje aktywacji termów zmiennej wejściowej wilgotność

$\mu_{WILGOTNOSC\ IS\ SREDNIA}(16) = 0{,}75$

Dla pozostałych termów wartość funkcji przynależności wynosi 0.

I.6.4. Wnioskowanie (inferencja)

Wnioskowanie rozpoczyna się od wyszukania tych reguł, których predykaty przesłanek mają niezerowe wartości funkcji aktywacji:

- `RULE 1: if temperatura is bardzo_niska then zraszanie is brak;`
- `RULE 2: if wilgotnosc is duza and temperatura is niska then zraszanie is brak;`
- `RULE 3: if wilgotnosc is duza and` **`temperatura is srednia`** `then zraszanie is brak;`
- `RULE 4: if wilgotnosc is duza and` **`temperatura is wysoka`** `then zraszanie is slabe;`
- `RULE 5: if` **`wilgotnosc is srednia`** `and temperatura is niska then zraszanie is brak;`
- `RULE 6: if` **`wilgotnosc is srednia`** `and` **`temperatura is srednia`** `then zraszanie is slabe;`
- `RULE 7: if` **`wilgotnosc is srednia`** `and` **`temperatura is wysoka`** `then zraszanie is srednie;`

- `RULE 8: if wilgotnosc is niska and temperatura is niska then zraszanie is slabe;`
- `RULE 9: if wilgotnosc is niska and` **`temperatura is srednia`** `then zraszanie is srednie;`
- `RULE 10: if wilgotnosc is niska and` **`temperatura is wysoka`** `then zraszanie is silne;`

Następnie obliczamy wartość funkcji aktywacji całych przesłanek reguł, w których występują niezerowe funkcje aktywacji predykatów. Zakładamy, że spójnik AND wyraża się t-normą MIN

- `RULE 3: if wilgotnosc is duza and` **`temperatura is srednia`** `then zraszanie is brak;`

$$\mu_{RULE\,3}(16{,}20) = min(0.0\ ,\ 0.75)$$

$$\mu_{RULE3}(16{,}20) = 0$$

Jak widzimy wartość funkcji aktywacji tej reguły wynosi 0. Możemy to odnieść do pozostałych reguł, w których przesłankach występują predykaty o zerowej wartości funkcji przynależności. Pozostają nam następujące reguły :

- `RULE 6: if` **`wilgotnosc is srednia`** `and` **`temperatura is srednia`** `then zraszanie is slabe;`
- `RULE 7: if` **`wilgotnosc is srednia`** `and` **`temperatura is wysoka`** `then zraszanie is srednie;`

Obliczamy wartości funkcji aktywacji tych reguł :

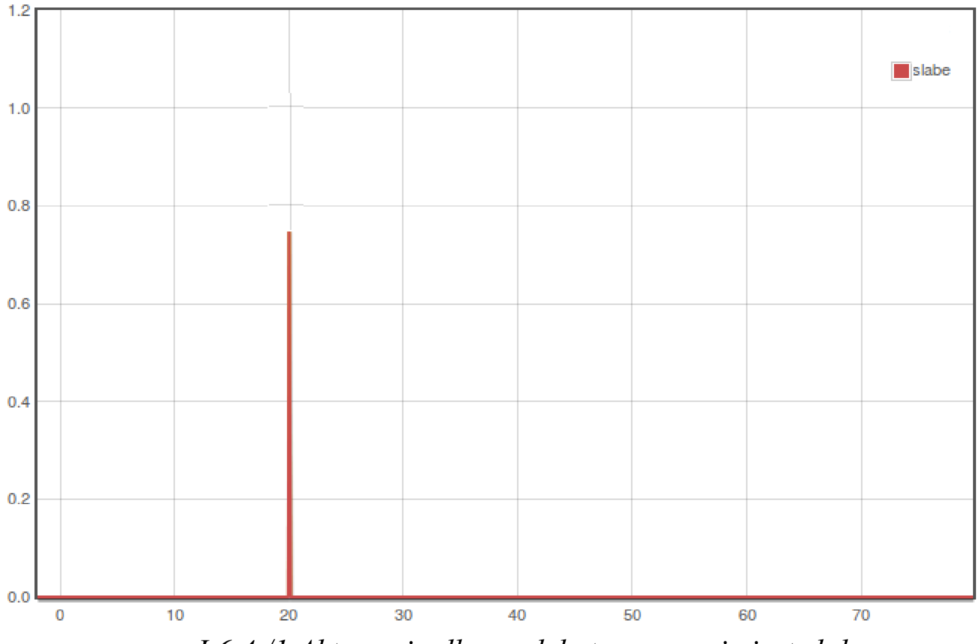
$$\mu_{RULE\,6}(16{,}20) = min(0.75\ ,\ 0.75)$$

$$\mu_{RULE6}(16{,}20) = 0.75$$

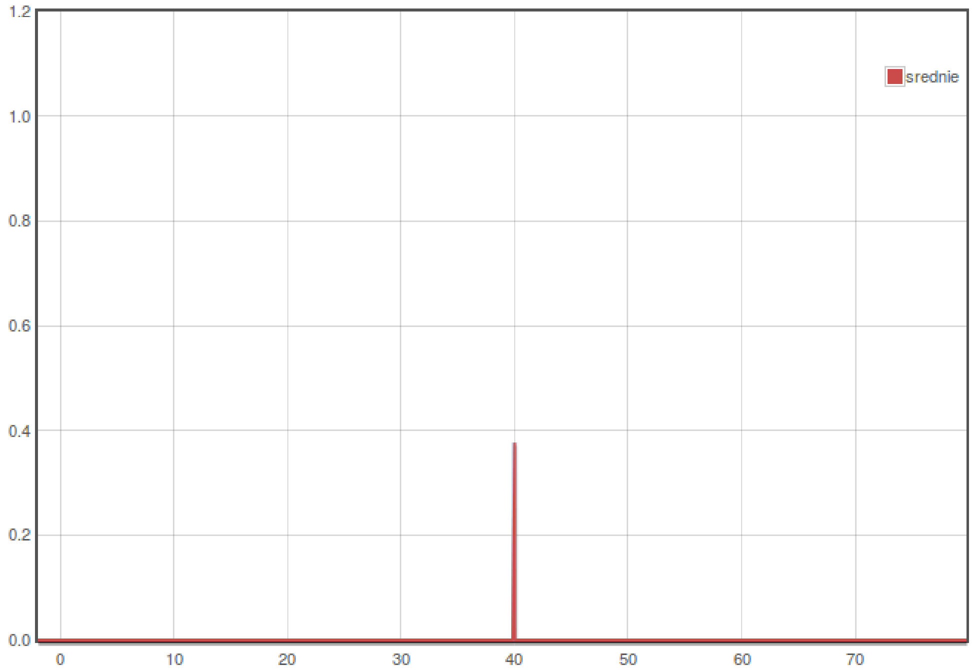
$$\mu_{RULE\,7}\,(16{,}20) = min(0.75\ ,\ 0.375)$$

$$\mu_{RULE\,7}\,(16{,}20) = 0.375$$

Zgodnie z konkluzją naszych reguł otrzymujemy zraszanie słabe na poziomie 0,75 i średnie na poziomie 0,375. Przeprowadzamy implikację na tych regułach (operator MIN) co prowadzi do powstania następujących termów :

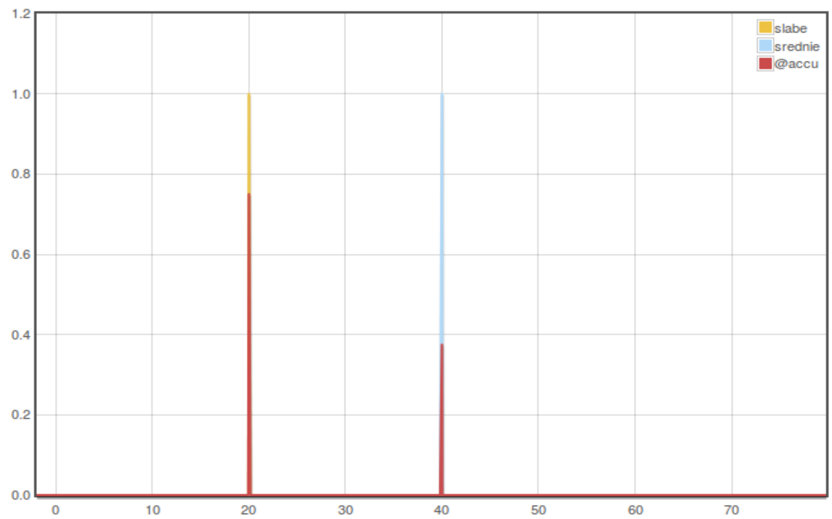


rys. I.6.4./1 Aktywacja dla predykatu zraszanie jest słabe



rys. I.6.4./2 Aktywacja dla predykatu zraszanie jest średnie

Na tak zebranych funkcjach przynależności przeprowadza się proces akumulacji zgodnie ze zdefiniowaną metodą (tutaj MAX). Prowadzi on do powstania następującego zbioru rozmytego :



rys. I.6.4./3 Akumulacja zebranych zbiorów rozmytych (kolor czerwony)

Na tym wykresie czerwoną linią zaznaczono poziom obcięcia singletonów termów SLABE I SREDNIE. Jednocześnie czerwony wykres jest akumulacją zbiorów tych dwóch singletonów. Użyty operator akumulacji to MAX.

I.6.5. Defuzyfikacja

Do defuzyfikacji użyjemy metody CoS (Centre of Singleton) zwaną również CoGS (Centre of Gravity for Singletons). Nie musimy obliczać całek oznaczonych jak w przypadku metody COG, ponieważ znamy z góry kształt funkcji przynależności zakumulowanych termów.

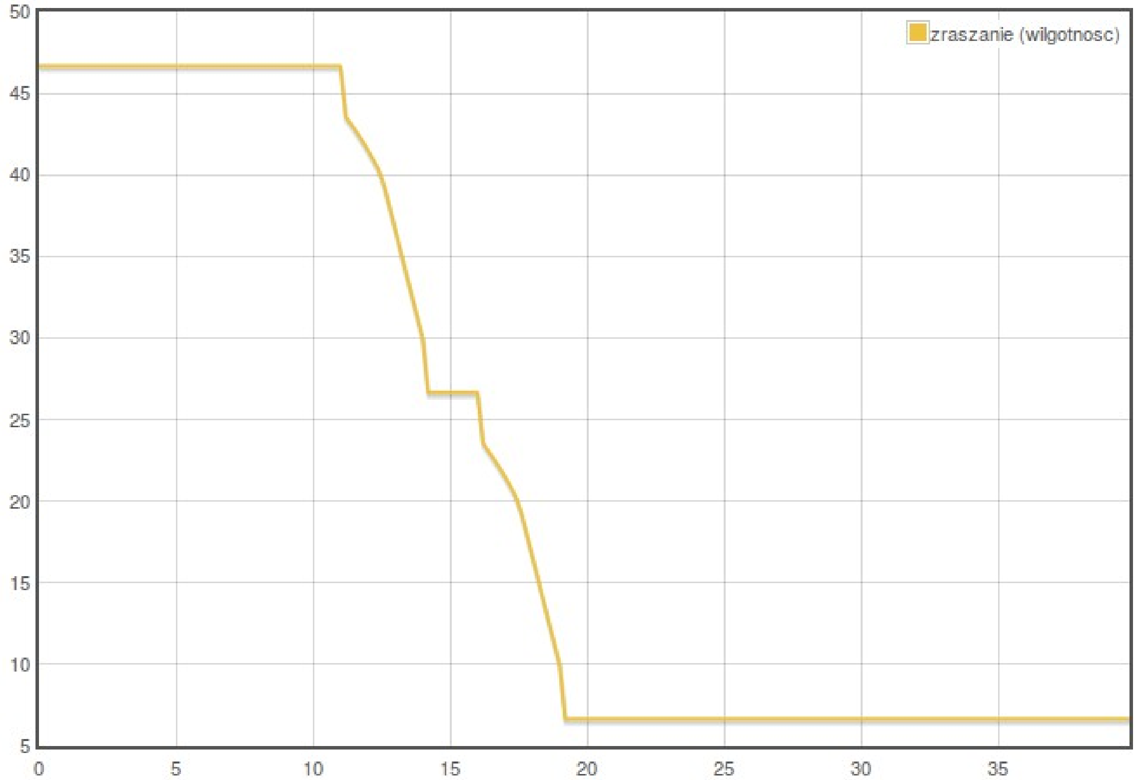
Wzór służący do obliczania tą metodą to :

$$v=\frac{\sum_{t\in S}\mu_t(t)\,t}{\sum_{t\in S}\mu_t(t)},\text{gdzie } S-\text{zbiór zagregowanych singletonów}$$

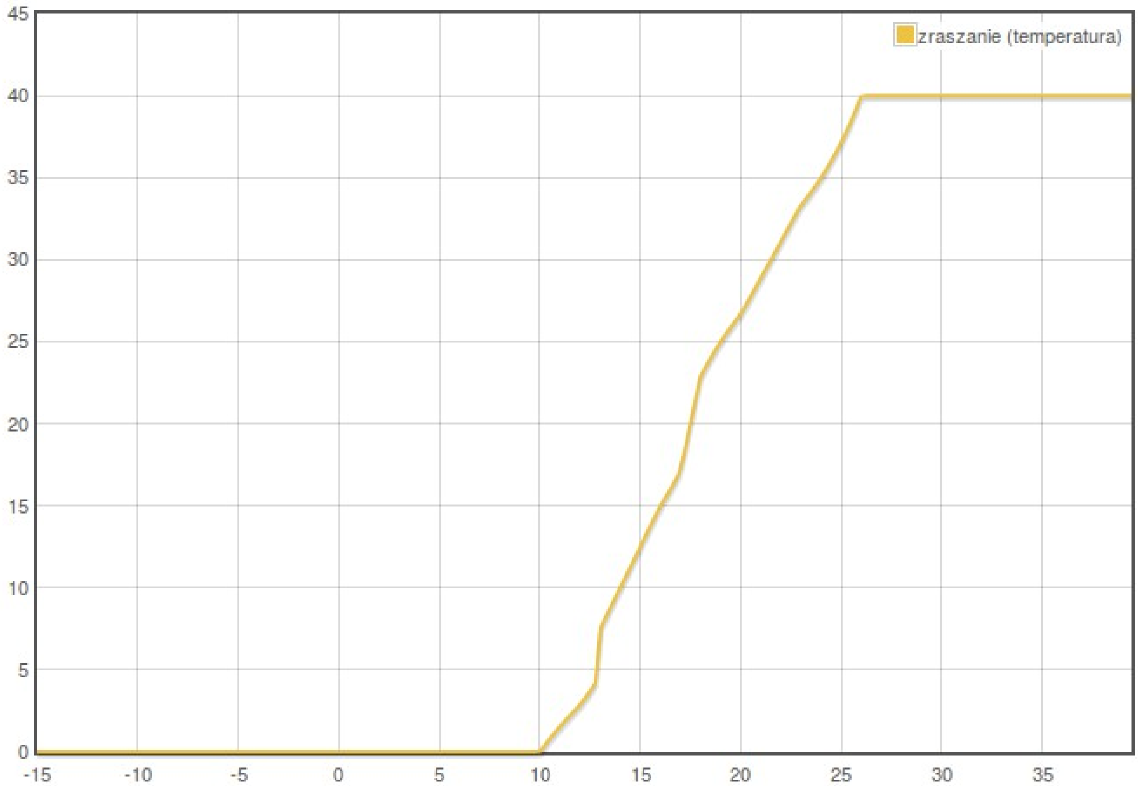
Obliczamy dla naszych danych:

$$v=\frac{0,75\cdot 20+0,375\cdot 40}{0,75+0,375}=\frac{30}{1,125}=26,(6)$$

Warto zauważyć, że model Mamdaniego daje nam możliwość tworzenia złożonych funkcji na podstawie prostych reguł opisanych przez osobę, która może być laikiem w kwestii projektowania aplikacji. Z racji, że mamy do czynienia z płaszczyzną w przestrzeni następne wykresy będą dotyczyły stanu wyjściowego z poprzedniego wnioskowania :

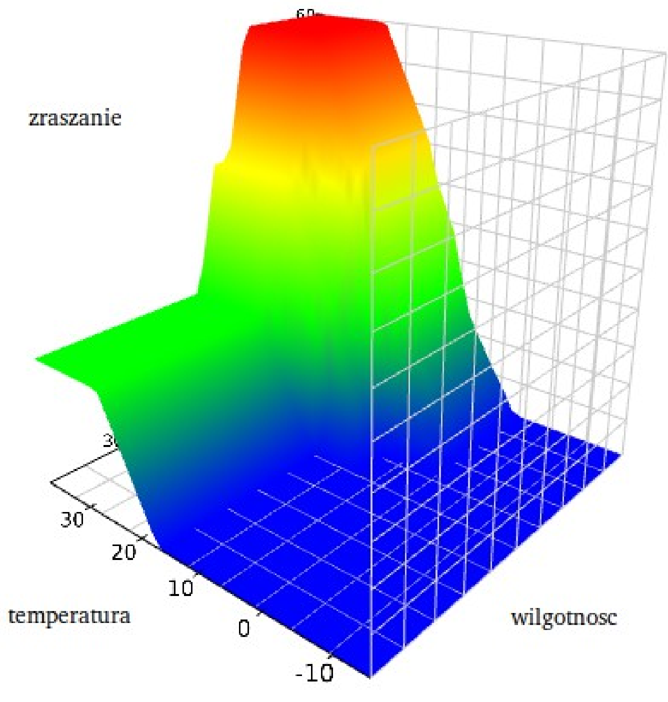


rys. I.6.5./1 Zraszanie w funkcji wilgotności dla temperatury 20°C

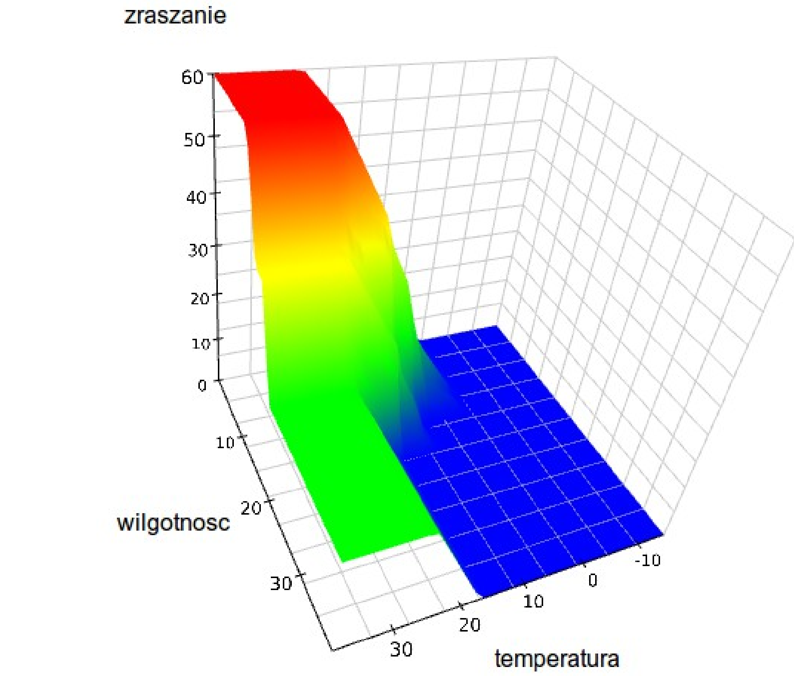


rys. I.6.5./2 Zraszanie w funkcji temperatury dla wilgotności 16%

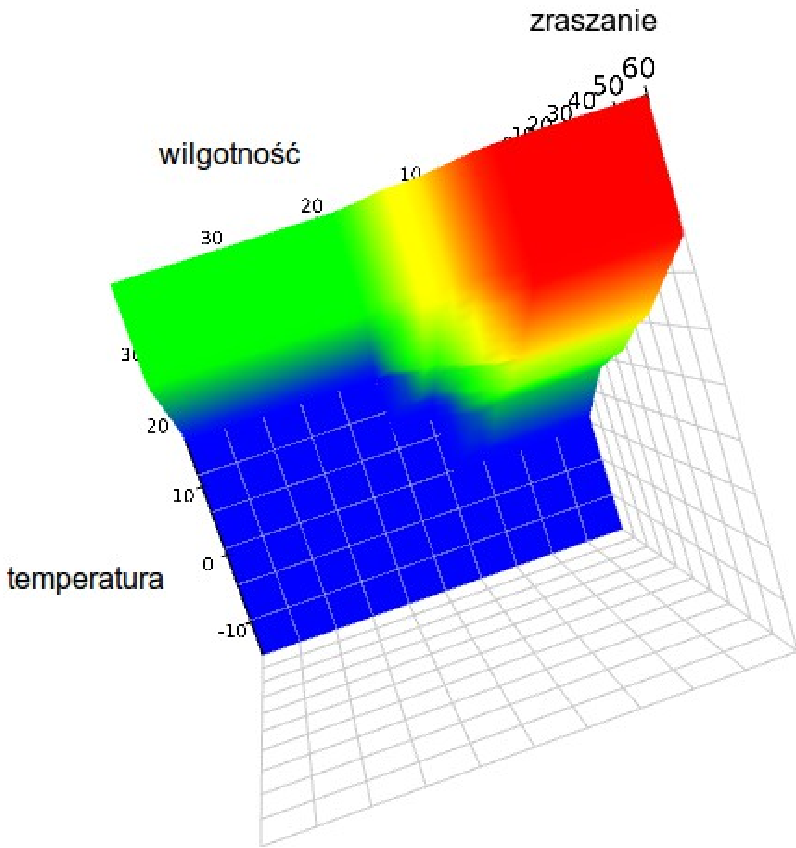
Całą złożoność opisanego modelu widać jednak dopiero na wykresie 3D przedstawiającym przestrzeń rozwiązań (model) opracowanego systemu rozmytego $zraszanie(temperatura, wilgotnosc)$:



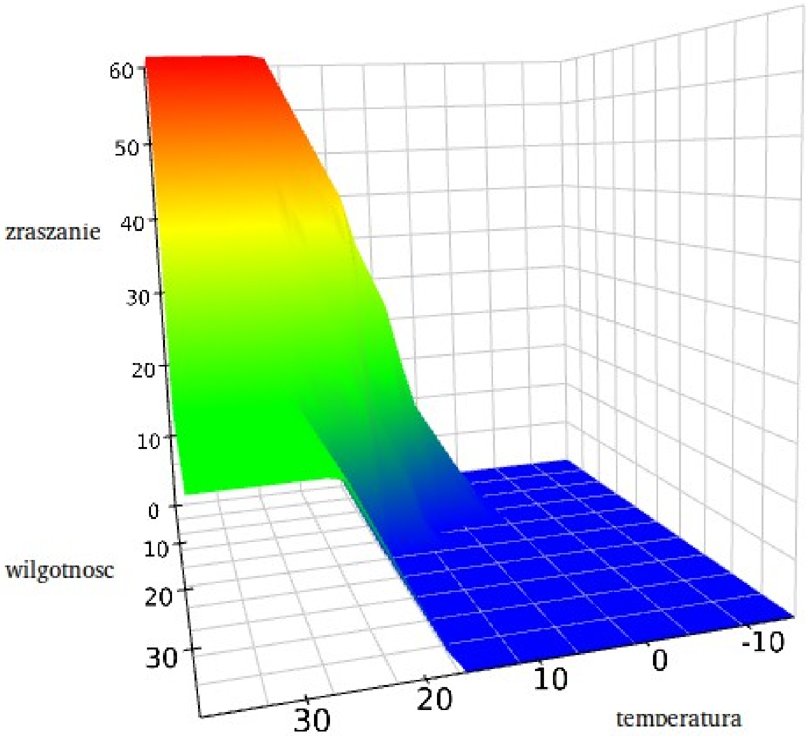
rys. I.6.5./3a Zraszanie w funkcji temperatury i wilgotności



rys. I.6.5./3b Zraszanie w funkcji temperatury i wilgotności



rys. I.6.5./3c Zraszanie w funkcji temperatury i wilgotności



rys. I.6.5./3d Zraszanie w funkcji temperatury i wilgotności