Skierowane liczby rozmyte

Główne zagadnienia

1.Przyczyny zajęcia się tym tematem

2.Co rozumiemy pod pojęciem skierowanych liczb rozmytych (*ordered fuzzy numbers* – OFN)

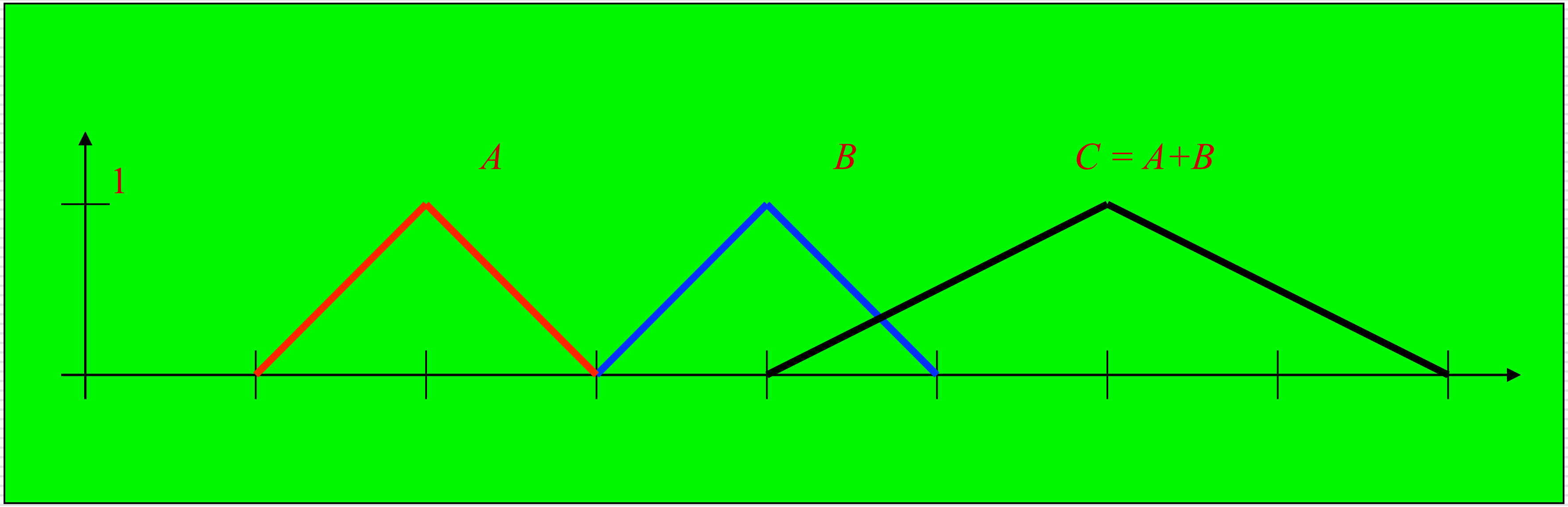
3.Działania na OFN

4.Co zyskujemy dzięki OFN

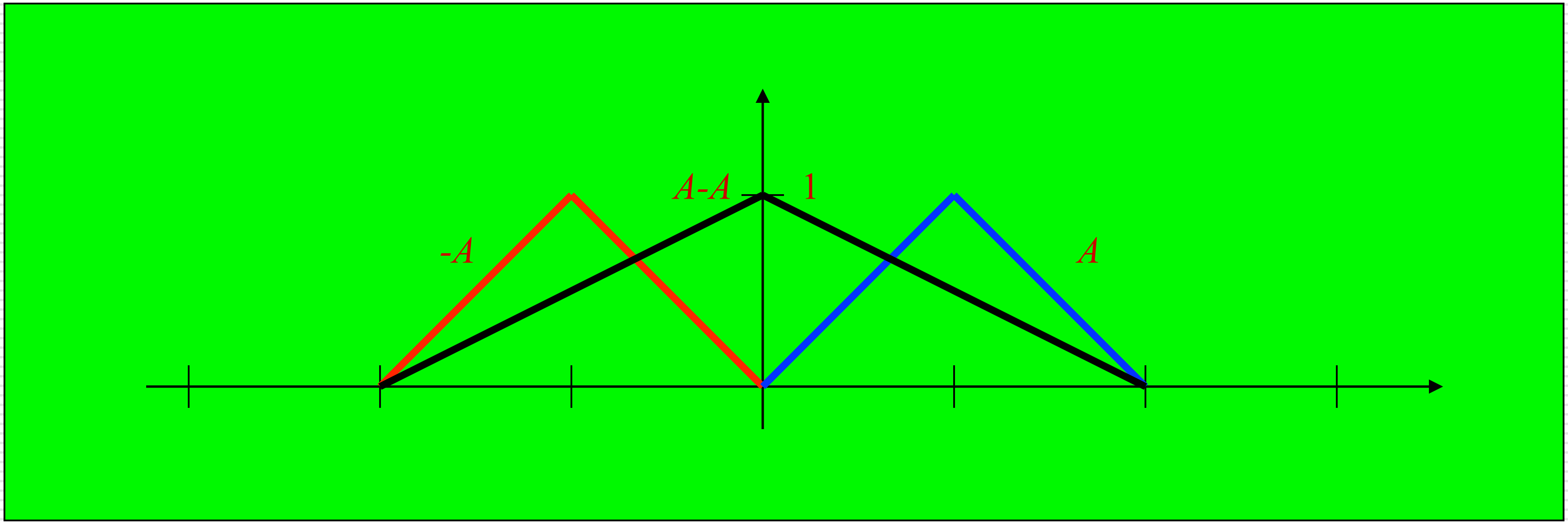
5.Omówienie tworzonego oprogramowania, korzystającego z idei OFN

Dlaczego szukamy innego modelu

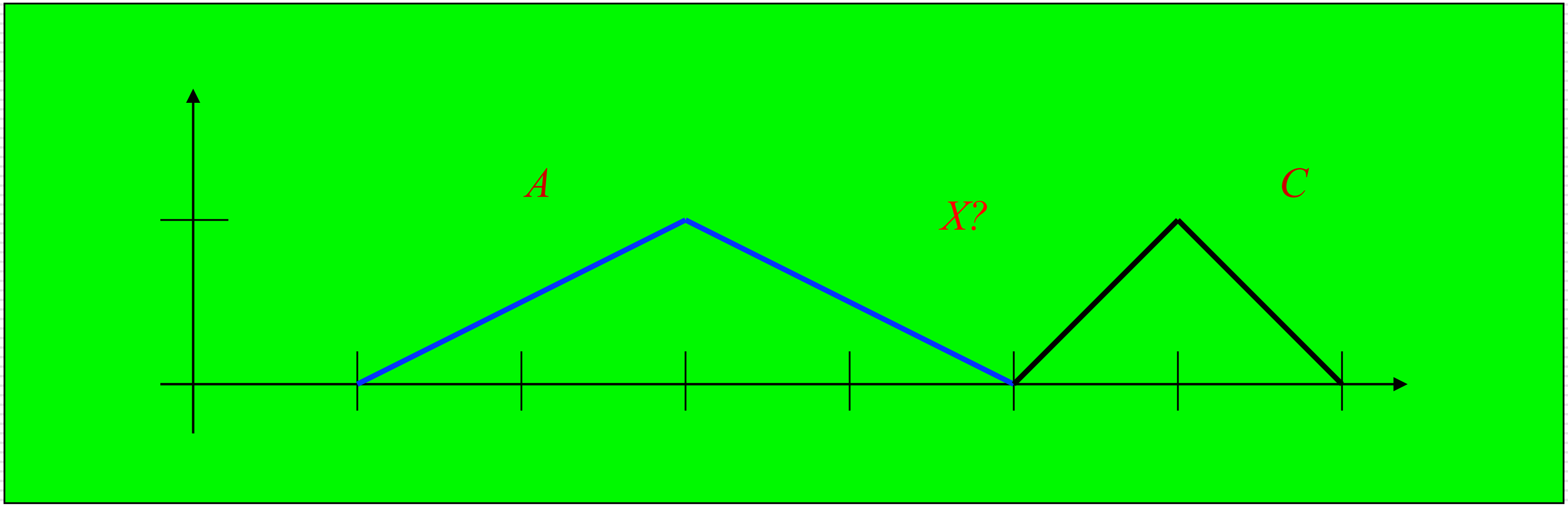
Podstawowe działania na liczbach rozmytych zwiększają ich „rozmytość” inaczej mówiąc nieprecyzyjność informacji po każdym działaniu zostaje powiększona, co w przypadku wielu działań może zmniejszyć ich użyteczność.



Konsekwencją powiększania nieprecyzyjności wraz z kolejnymi działaniami na liczbach rozmytych jest to, iż wynikiem działania *A+(-A)* nie jest element neutralny względem dodawania. Otrzymujemy zero rozmyte zamiast zera rzeczywistego.



Kolejne zagadnienie wynikające z powiększania nieprecyzyjności, to brak rozwiązania równania $A+X=C$ gdzie A i C są liczbami rozmytymi.



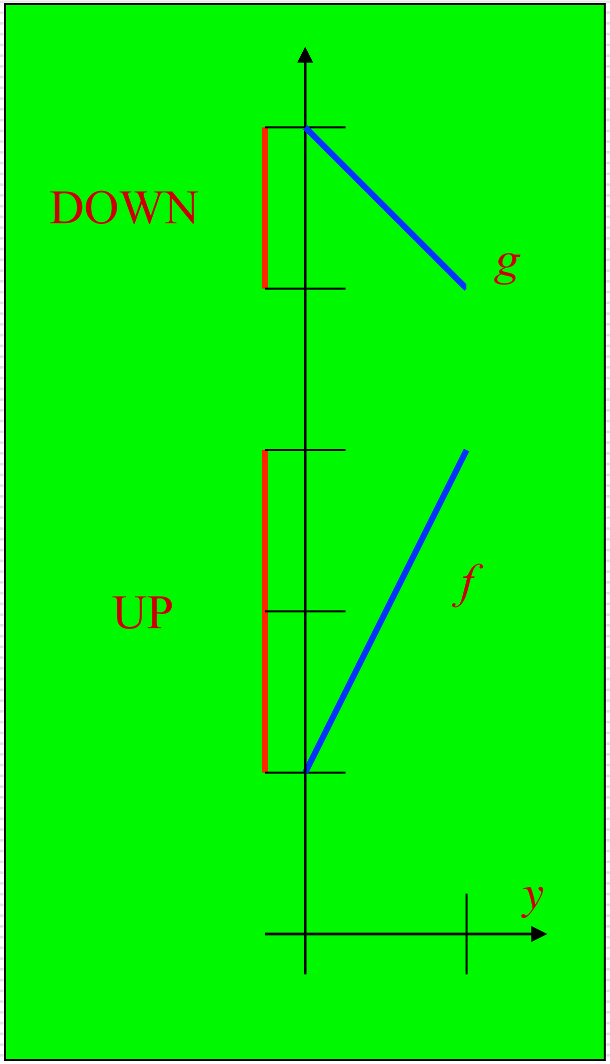
Skierowane liczby rozmyte

Przez skierowaną liczbę rozmytą *A* rozumiemy uporządkowaną parę ciągłych funkcji.

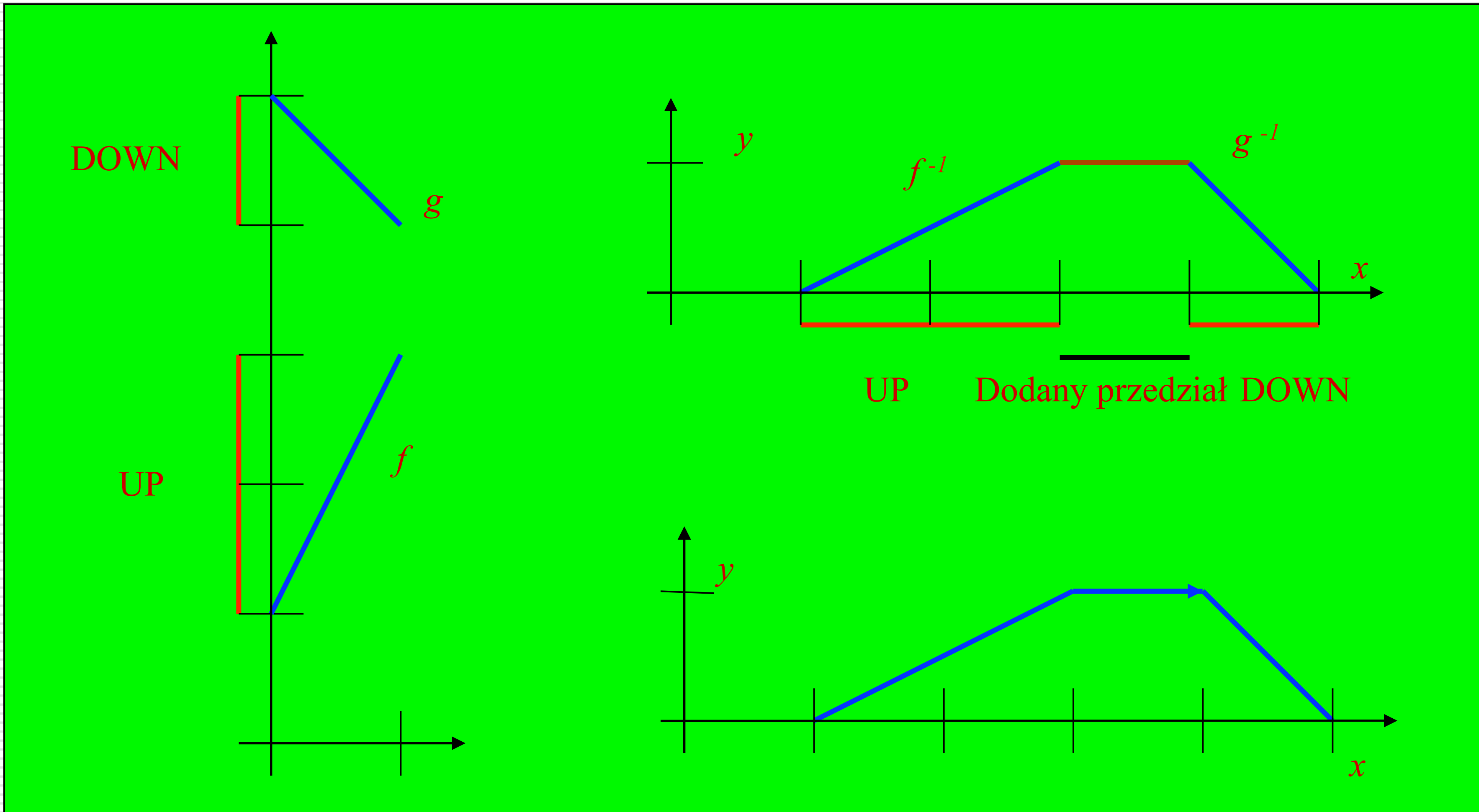
$$A = (f, g)$$

gdzie $f, g : [0, 1] \rightarrow \mathbf{R}$

Funkcję *f* nazywamy też częścią up, zaś funkcję *g* – częścią down skierowanej liczby rozmytej

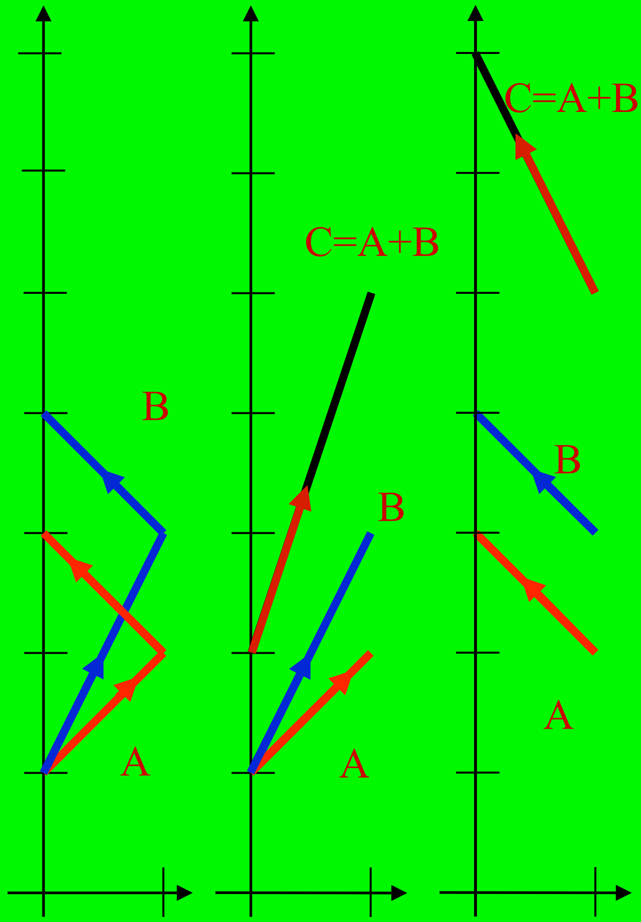


Sposób interpretacji skierowanej liczby rozmytej jako liczby rozmytej wypukłej. Jest to możliwe, gdy *f* i *g* są monotoniczne.

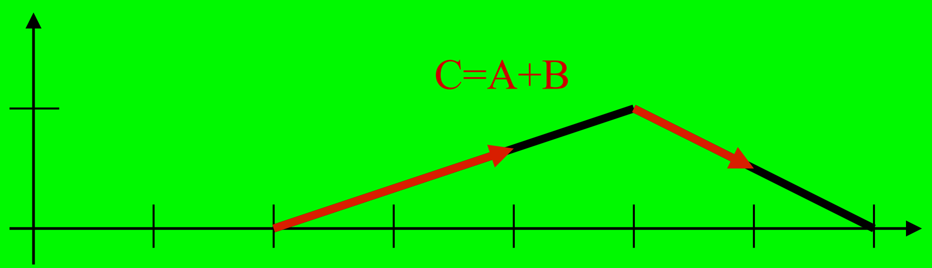


Dodawanie

Dodawanie: $C = A + B$ $f_A(y) + f_B(y) = f_C(y) \wedge g_A(y) + g_B(y) = g_C(y)$

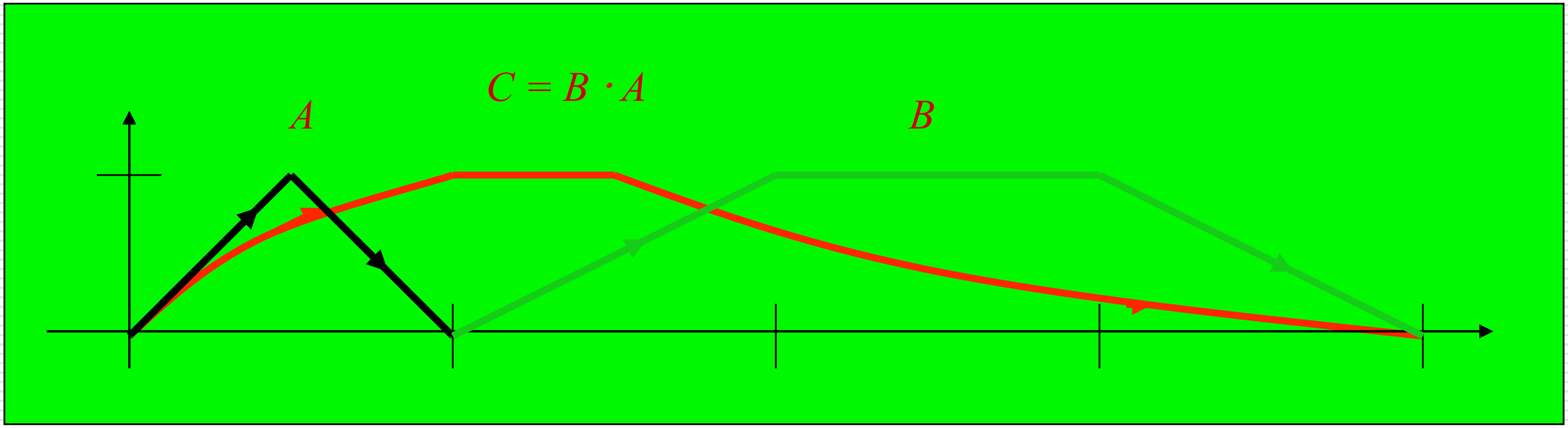


Przekształcenie do postaci porównywalnej z wypukłymi liczbami rozmytymi.



Mnożenie

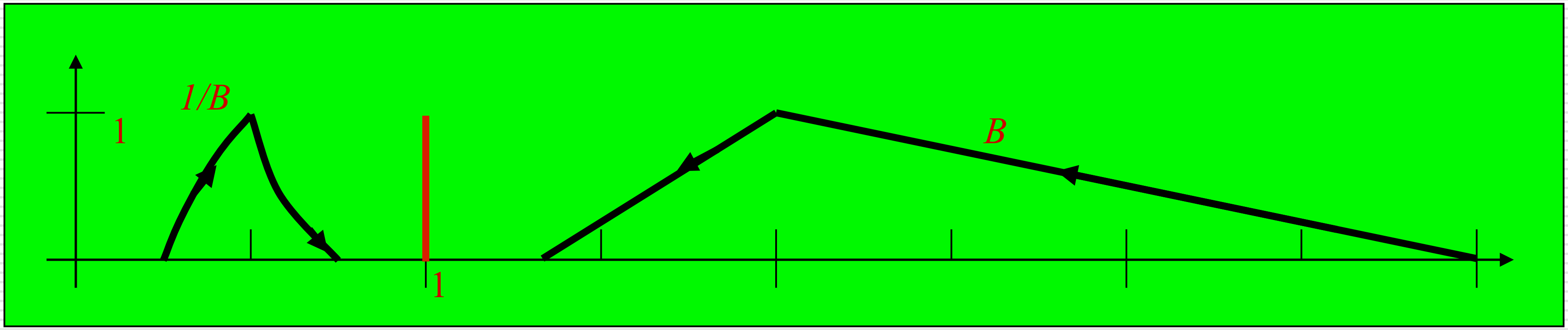
Mnożenie: $C = A \cdot B$ $\quad f_A(y) \cdot f_B(y) = f_C(y) \wedge g_A(y) \cdot g_B(y) = g_C(y)$

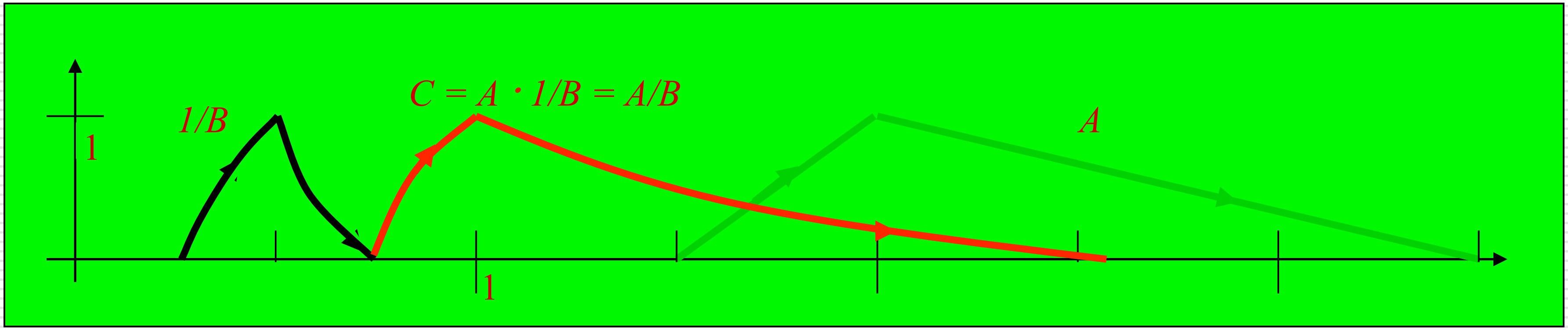


Dzielenie

Dzielenie: $C = A / B$ (o ile zero nie należy do nośnika B)

$$f_A(y) / f_B(y) = f_C(y) \wedge g_A(y) / g_B(y) = g_C(y)$$

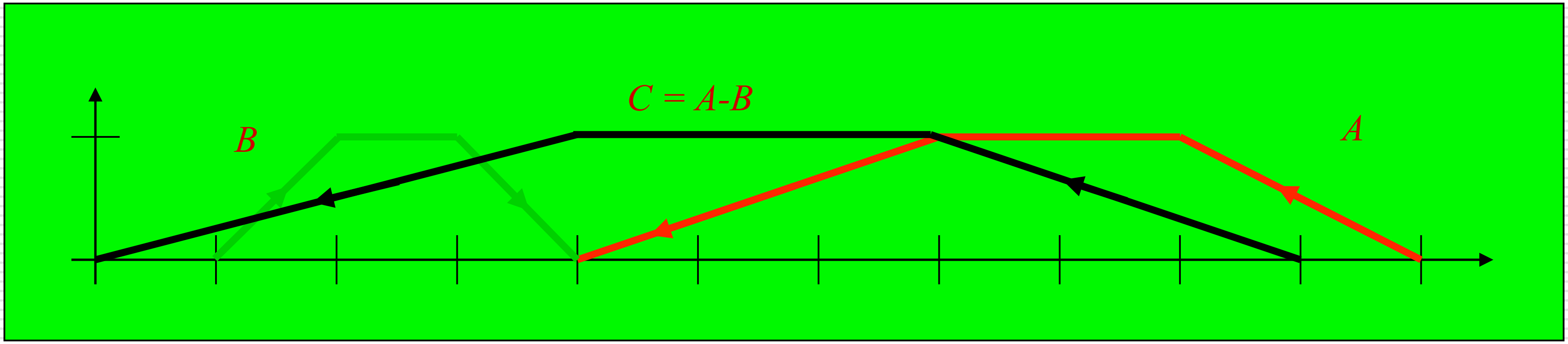


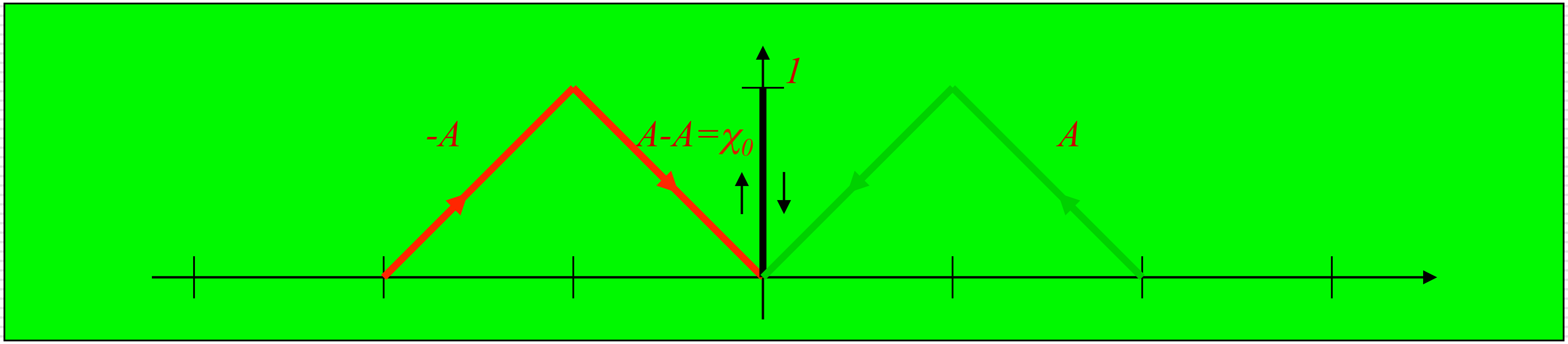


Odejmowanie

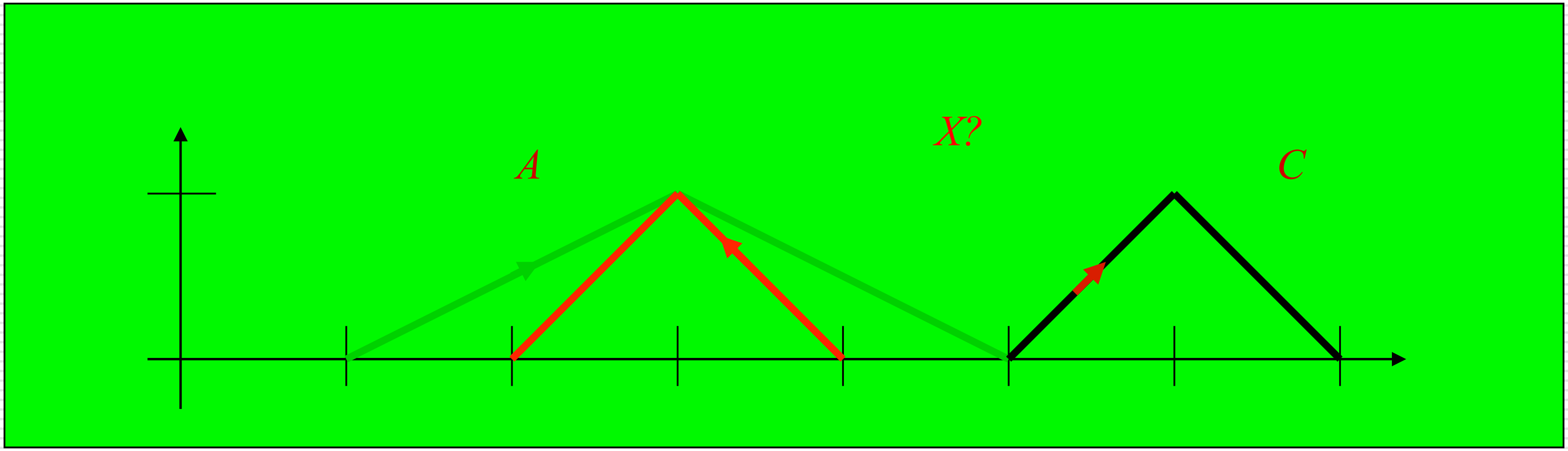
Liczba przeciwna: jeśli $A = (f_A, g_A)$, to $-A = (-f_A, -g_A)$

zatem odejmowanie: $C = A - B = A + (-B)$





Należy tu podkreślić, że jedynie część skierowanych liczb rozmytych można zinterpretować jako liczby rozmyte wypukłe. Pozostaje wiele przypadków, co do których interpretacja jest kłopotliwa, jak też wiele przypadków przydatnych w obliczeniach na liczbach rozmytych. Przykładem może być rozwiązanie równania $A+X=C$, gdzie już intuicyjnie widać, że wynik nie należy do zbioru liczb rozmytych wypukłych.



Korzyści z używania OFN

- Model OFN wraz z działaniami uwalnia nas od ciągłego zwiększania się nieprecyzyjności.

- Mamy możliwość swobodnej realizacji działań, bez rozgraniczania liczb rozmytych na trójkątne, trapezoidalne itd.

- Stosunkowo proste w realizacji działania (również w implementacji).

- Jeśli ograniczymy analizę działań na skierowanych liczbach rozmytych do ich podzbioru o orientacji pozytywnej (bądź tylko negatywnej), to w ogólności wyniki obliczeń będą zgodne z działaniami na trójkątnych liczbach rozmytych.

Implementacja idei OFN

Pakiet *Tlroz* został zaimplementowany jako podstawowe narzędzie do praktycznego korzystania z OFN. Został on napisany w języku Object Pascal w środowisku programistycznym Delphi 6.0. Pakiet ten zawiera definicję czterech klas oraz kilku struktur danych wykorzystywanych przez owe klasy.

Kolejny pakiet *Tsterownik*, wykorzystuje publiczne metody klas pakietu *Tlroz*, w celu przetwarzania informacji i umożliwienia procesu sterowania rozmytego. *Tsterownik,* zawiera definicję jednej klasy, kilku struktur oraz typów danych, a także paru stałych.

Funkcje *Tlroz*

Podstawowe funkcje realizowane przez pakiet *Tlroz* to:

- Tworzenie, definiowanie i przechowywanie OFN w postaci danych cyfrowych.
- Podstawowe operacje algebraiczne między dwiema OFN: dodawanie, odejmowanie, mnożenie, dzielenie.
- Obliczanie wartości OFN dla rzeczywistego argumentu.
- Możliwość korzystania z wielomianów rozmytych, czyli takich, w których współczynniki w wyrazach wielomianu oraz jego argument są liczbami rozmytymi (OFN).

Funkcje *Tsterownik*

Podstawowe funkcje realizowane przez pakiet *Tsterownik* to:

- Przechowywanie informacji o sterowanym procesie oraz o bazie reguł.
- Przetwarzanie danych zgodnie z ideą sterowania rozmytego, z wykorzystaniem OFN.
- Możliwość funkcjonowania sterownika zarówno jako sterownik typu Mamdaniego jak i Takagi-Sugeno-Kanga.
- Możliwość dobierania poszczególnych parametrów sterowania rozmytego. Dotyczy to: metod agregacji, implikacji oraz wyostrzania.

Wraz z pakietem *Tsterownik* powstawały dodatkowe narzędzia, niezbędne do testowania tego pakietu. Są to wszelkie interfejsy wspomagające wprowadzanie informacji (innych dla każdego rodzaju sterowania) o sterowanym procesie jak i bazie reguł. Interfejsy te są wersjami „mocno roboczymi” i nie są składnikami pakietu.

Co dalej?

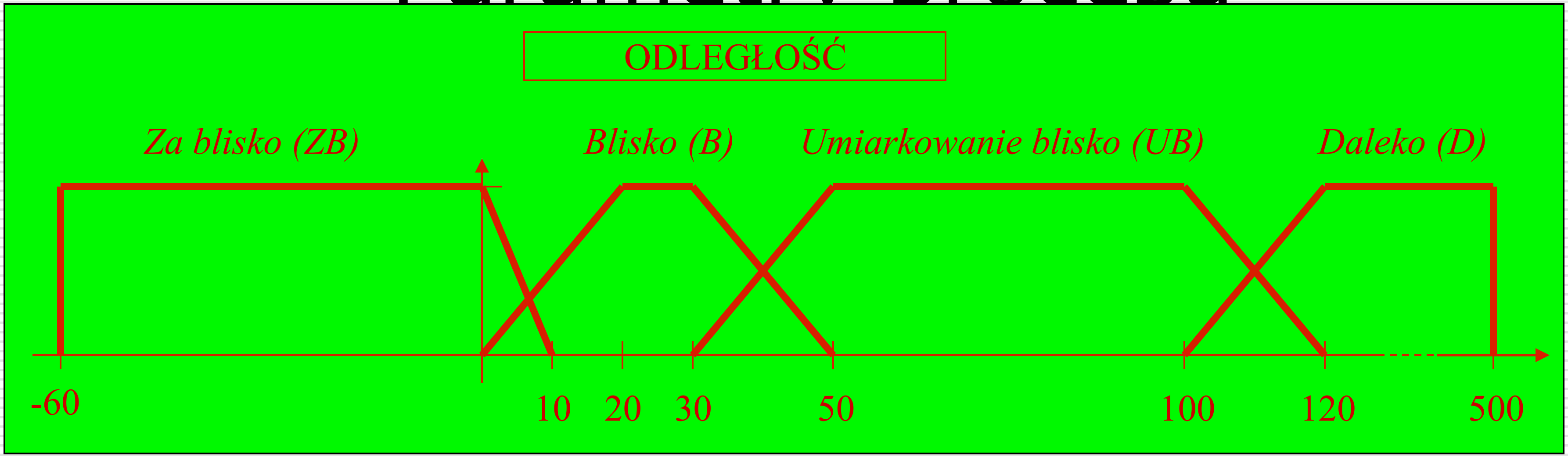
W obecnej chwili, głównym zagadnieniem związanym z dalszą pracą nad przedstawionymi tu pakietami jest wykorzystanie ich do sterowania robotem *Pioneer 2* (prod. *ActiveMedia ROBOTICS*).

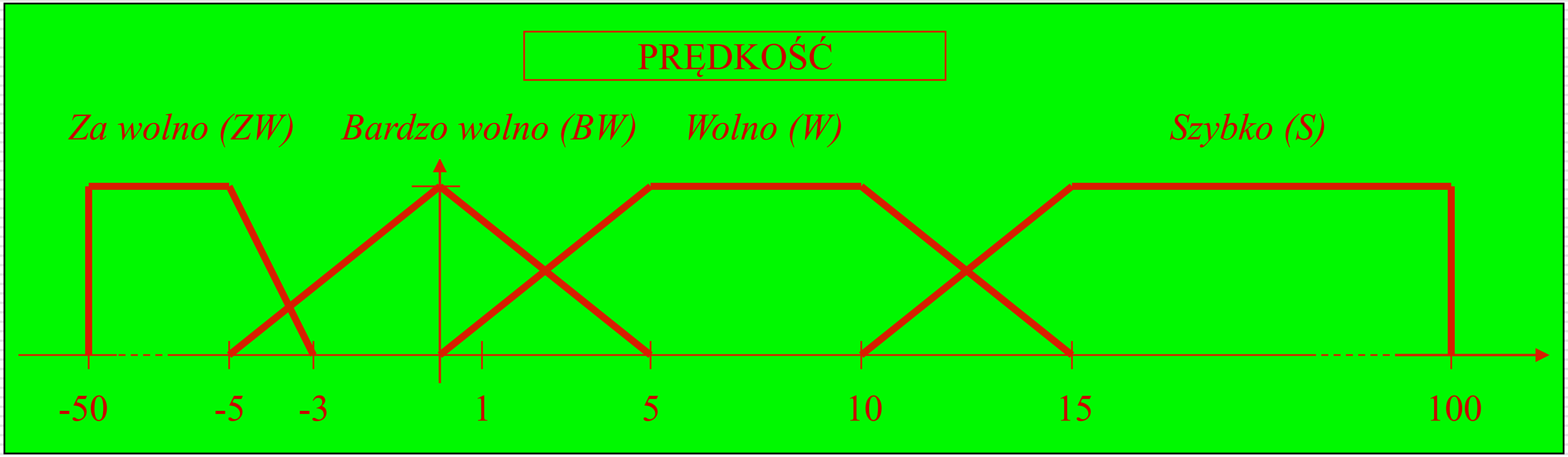
W dalszej części tej prezentacji pokazany zostanie pakiet *Tsterownik*, wraz z interfejsami umożliwiającymi dynamiczne przedstawienie wyników.

Omówienie przykładowego procesu

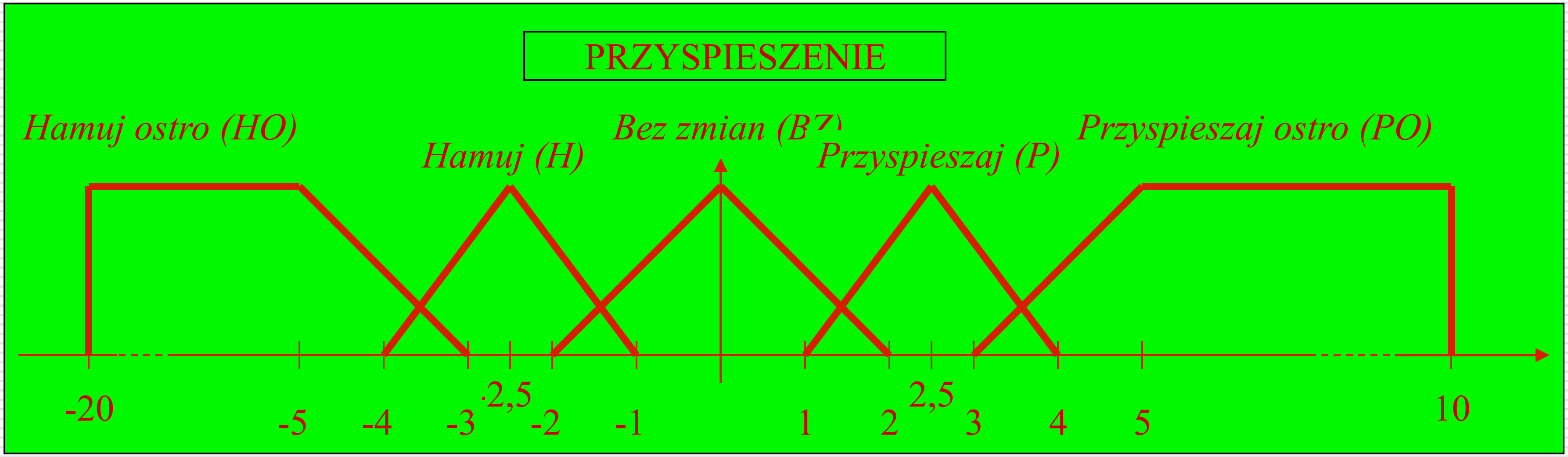
Zanim przejdziemy do bezpośredniej prezentacji działania pakietu *Tsterownik* należy się przyjrzeć przykładowemu procesowi. Będzie to klasyczny przykład sterowania poruszającym się punktem. Na podstawie prędkości i odległości od przeszkody wyliczana będzie przy pomocy sterownika rozmytego Mamdaniego wartość przyspieszenia.

Parametry procesu





Sterowanie procesem



Reguły

	Za blisko	Blisko	Umiarkowanie blisko	Daleko
Za wolno	**Przyspieszaj**	**Przyspieszaj ostro**	**Przyspieszaj ostro**	**Przyspieszaj ostro**
Bardzo wolno	**Hamuj**	**Bez zmian**	**Przyspieszaj**	**Przyspieszaj ostro**
Wolno	**Hamuj ostro**	**Hamuj**	**Bez zmian**	**Przyspieszaj**
Szybko	**Hamuj ostro**	**Hamuj Ostro**	**Hamuj**	**Bez zmian**

Elementy sterownika rozmytego specyficzne dla OFN

W klasie *Tsterownik* zaimplementowane zostały obok klasycznych metod agregacji (MIN, PROD), wnioskowania (Mamdani, Larsen) i wyostrzania (COG, COM) także metody specyficzne dla OFN. Wszelkie nowe propozycje wykorzystują swobodę liczenia na OFN. Przykładowo jeśli wykorzystamy metodę wnioskowania, która jako wyjście z reguły daje OFN, to metodę wyostrzania można określić jako średnią z wniosków poszczególnych reguł.

$$Y_{wyn} = \frac{\sum_{i=1}^{k} Y_{wynRi}}{k}$$

Y_{wynRi} – liczba OFN uzyskana w wyniku wnioskowania w *i*-tej regule

Y_{wyn} – wynikowa OFN z całej bazy reguł

k – ilość reguł

Do tak uzyskanej liczby Y_{wyn} możemy zastosować jedną z klasycznych metod wyostrzania do uzyskania wartości precyzyjnej będącej odpowiedzią całego regulatora rozmytego.

Wybrane publikacje

1. Zadeh L.A. Fuzzy sets, Information and Control **8** (1965) 338-353
2. Dubois D., Prade H., Operations on fuzzy numbers, *Int.J. System Sciencei* **9** (1978) 576-578
3. Kosiński W., Prokopowicz P. Modelling operations on fuzzy reals, in: *Materiały Konf. Modelowanie i symulacja komputerowa w technice.* Wyższa Szkoła Informatyki, Łódź, marzec 2002, pp. 97-101.
4. Kosiński W., Prokopowicz P., Ślęzak D., Drawback of fuzzy arthmetics – new intutions and propositions, in: *Proc. AI METH, Methods of Aritificial Intelligence* , T. Burczyński, W. Cholewa, W. Moczulski, (eds), Gliwice, Poland (2002), pp. 231-237
5. Kosiński W., P. Prokopowicz P., Ślęzak D., On algebraic operations on fuzzy numbers, *Intelligent Information Processing and Web Mining,* Proc. of the International IIS: IIPWM,03 Conference held in Zakopane, Poland, June *2-5,*2003, M. Kłopotek, S.T. Wierzchoń, K. Trojanowski (eds.),PhysicaVerlag, 2003, pp. 353-362.
6. Kosiński W., P. Prokopowicz P., Ślęzak D., Ordered fuzzy number, *Bulletin of the Polish Academy of Sciences, Ser. Sci. Math.,* 51 (3), 2003, 327-338.

Dziękuję za uwagę ...