ISSN: 2181-4341, IMPACT FACTOR (RESEARCH BIB) – 7,245, SJIF – 5,431

Sadullayev Azizbek Nasillo o‘g`li
O`qituvchi-stajyor, Buxoro muhandislik texnologiya instituti

NEFTNI ISITISH TEXNOLOGIK TIZIMIDAGI ISSIQLIK ALMSHINISH QURILMASIDA HARORATNI ROSTLASH VA BOSHQARUV DASTURINI ISHLAB CHIQISH

Annotatsiya: Ushbu maqolada neftni qayta ishlash texnologik tizimida qo‘llaniladigan issiqlik almashinish qurilmasidagi haroratni rostlash va boshqarish dasturi Python dasturlash tilidagi noaniq mantiq elementlaridan foydalanilgan holatda tuzildi. Dastur komputerda intepretatsiya qilinib natijalar olindi va tahlil qilindi.

Kalit so‘zlar: Fuzzylogic, Python, Rasberry Pi 3 Model B, matplotlib kutubxonasi, Domain, Set, Uchburchak funksiyasi.

Kirish. Neftni qayta ishlash texnologik tizimidagi issiqlik almashinish qurilmasidagi haroratni rostlash va boshqarish dasturini Python dasturlash tilida tuziladi. Python dasturlash tizimida juda ko‘plab tayyor kutubxonalari mavjud. Shu jumladan, haroratni rostlash dasturi uchun *fuzzylogic* kutubxonasidan foydalanish maqsadga muvofiq bo‘ladi. Bu kutubxona yordamida noaniq sharoitlarda optimal qaror qabul qilish mumkin bo‘ladi. Ushbu kutubxona orqali yozilgan kodni shu tilda ishlash imkoniyatiga ega bo‘lgan Rasberry Pi 3 Model B mikrokontrollerning xotirasiga yoziladi. Dasturni joriy etish quyidagi bosqichlarni o‘z ichiga oladi.

Mikrokontroller uchun boshqaruvchi dasturni kompyuterda bajariladi va quyidagi bosqichlar amalga oshiriladi:

1. dasturning matnini tayyorlash
2. matnni mashina kodiga translyatsiyalash va sintaktik xatolarni tuzatish
3. dasturni sozlash (mantiqiy xatoliklarni tuzatish),
4. mikrokontrollerni yakuniy dasturlashtirish.

Nazariy qism. Jarayonni boshqarish algoritmi bizga unga asoslangan holda boshqarish dasturini ishlab chiqishga imkon beradi. Algoritmni ishlab chiqishda texnologik jarayonda ishtirok etadigan issiqlik almashinish qurilmasida kechadigan qonuniyatlarni inobatga olishimiz kerak: Tabiiyki issiqlik almashinish jarayonida qurilmaga kiritiladigan issiq va sovuq agentlarning fizik ko‘rsatgichlari batafsil o‘rganilib chiqqan holda, issiq agent yuqori haroratli benzin bug‘lari bo‘lib harorati 120 ^{0}C atrofida bo‘ladi, Sovuq agent neftgazkondensat bo‘lib, harorati 10÷30 ^{0}C gacha va sarfi esa 1÷3 kg/s gacha o‘zaradi. Neftgazkondensatning chiqish harorati shartli ravishda 30 ^{0}C deb qabul qilgan holda mikrokontroller uchun dastur tuziladi.

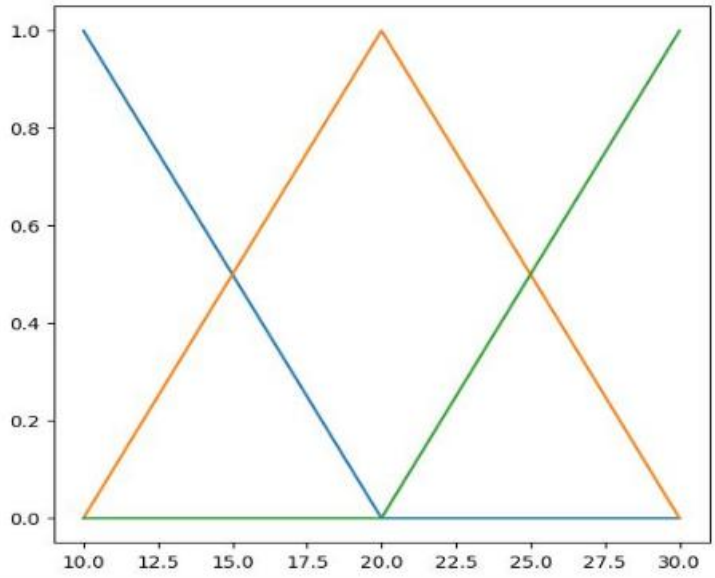
Avvaliga dasturning kod qismini yozishdan oldin bazi kerak bo‘ladigan kutubxonalari yuklab olish zarur. Eng birinchi navbatda *matplotlib* kutubxonasi yuklab olinadi.

```
from matplotlib import pyplot

from fuzzylogic.classes import Domain, Set

from fuzzylogic.functions import (triangular, S,R)
```

Yuqorida birinchi qatorda *matplotlib* kutubxonasidan *pyplot* funksiyasini chaqirib olinadi. Ikkinchi qatorda esa *fuzzylogic.classes* kutubxonasidan *Domain* va *Set* klaslarini chaqirib oladi. Uchinchi qatorda esa *fuzzylogic.functions* funksiyasidan *triangular* ya'ni uchburchak funksiyasini chaqirib oladi.

```
temp = Domain("Temperature", 10, 30)

temp.cold = triangular(0,20, c=10)

temp.normal = triangular(10,30, c=20)

temp.hot = triangular(20,40, c=30)

pyplot.rc("figure", figsize=(6, 6))

temp.cold.plot()

temp.normal.plot()

temp.hot.plot()
```

Birinchi qatorda *Domain* klasida *Temperature* paramerti kiritildi. Oraliq diapozonida sifatida 10 va 30 qiymati berildi. Keyingi qatorlarda esa aynan shu klass ichida *cold, normal* va *hot* funksiyalari tanlanib, ularga uchburchak funksiya ichida mos ravishda 0 20, 10 30 va 20 40 oraliq beriladi. Shundan so‘ng *plot* funksiyasi yordamida har birini bitta chizmada berilgan qiymatlar bo‘yicha yaratadi (1.1-rasm).

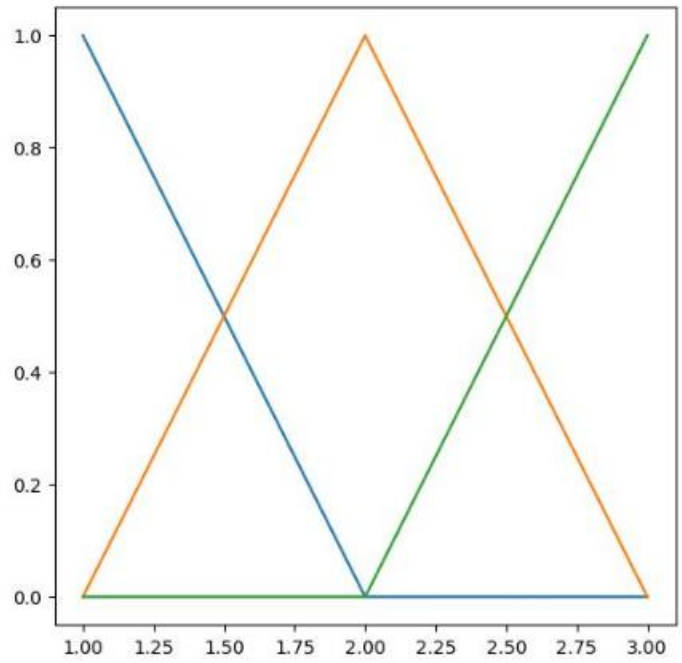


1.1-rasm. Pythonda dasturida horarat qiymatlarini noaniqlikka o‘tkazish grafigi.

```
from matplotlib import pyplot

from fuzzylogic.classes import Domain, Set

from fuzzylogic.functions import (triangular, S,R)
```

```
sarf = Domain("Sarf", 1, 3)

sarf.few = triangular(0,2, c=1, c_m=1, no_m=0)

sarf.normal = triangular(1,3, c=2, c_m=1, no_m=0)

sarf.many = triangular(2,4, c=3, c_m=1, no_m=0)

sarf.few.plot()

sarf.normal.plot()

sarf.many.plot()
```

Domain klasida *Sarf* paramerti kiritildi. Oraliq diapozonida sifatida 1 va 3 qiymati berildi. Keyingi qatorlarda esa aynan shu klass ichida *few, normal* va *many* funksiyalari tanlanib, ularga uchburchak funksiya ichida mos ravishda 0 2, 1 3 va 2 4 oraliq beriladi. Shundan so‘ng *plot* funksiyasi yordamida har birini bitta chizmada berilgan qiymatlar bo‘yicha yaratadi (1.2-rasm).

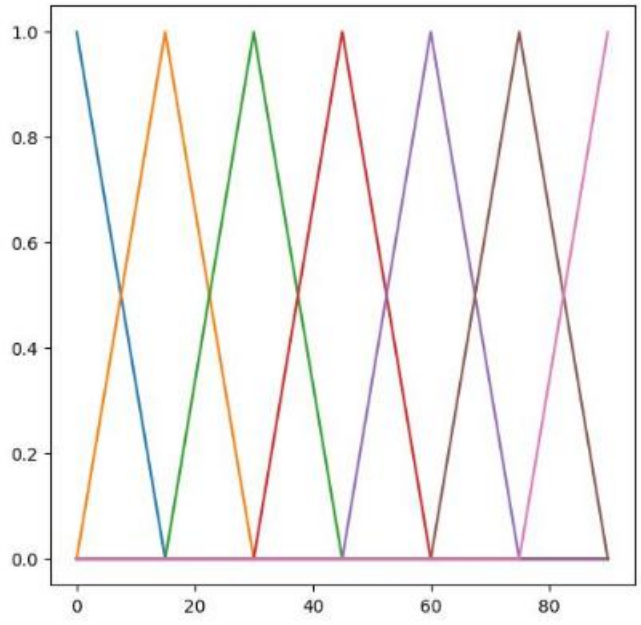


1.2-rasm. Pythonda dasturida sarf qiymatlarini noaniqlikka o‘tkazish grafigi.

```
from fuzzylogic.classes import Domain, Set

from fuzzylogic.functions import (triangular, S,R)

burchak = Domain("Buralish burchagi", 0, 90)

burchak.minimal = S(0,15)

burchak.lesser = triangular(0,30, c=15)

burchak.less = triangular(15,45, c=30)
```

```
burchak.normal = triangular(30,60, c=45)

burchak.many = triangular(45,75, c=60)

burchak.more_many = triangular(60,90, c=75)

burchak.maximal = R(75,90)

pyplot.rc("figure", figsize=(6, 6))

burchak.minimal.plot()

burchak.lesser.plot()

burchak.less.plot()

burchak.normal.plot()

burchak.many.plot()

burchak.more_many.plot()

burchak.maximal.plot()
```

Dastlab *Domain* klasida *Buralish burchagi* paramerti kiritildi. Oraliq diapozonida sifatida 0 va 90 qiymati berildi. Keyingi qatorlarda esa aynan shu klass ichida *minimal, lesser, less, normal, many, more many* va *maximal* funksiya tanlanib, ularga uchburchak funksiya ichida mos ravishda 0 30, 15 45, 30 60, 45 75, 60 90 va 75 90 oraliq beriladi. Shundan so‘ng *plot* funksiyasi yordamida har birini bitta chizmada berilgan qiymatlar bo‘yicha yaratadi (1.3-rasm).



1.3-rasm. Pythonda dasturida burchak o‘zgarishi qiymatlarini noaniqlikka o‘tkazish grafigi.

```
from fuzzylogic.classes import Domain, Set, Rule
```

```
from fuzzylogic.hedges import very

from fuzzylogic.functions import triangular, S,R

def gradirovka_temp(x):

    return 5+1.25*x
```

Bu yerda *gradirovka temp* deb saqlangan funksiya yaratilgan. Ushbu funksiya harorat datchigidan kelgan signalni *Domain* klassi oraliq diapozoniga tegishli qiymatga aylantiradi.

```
def gradirovka_sarf(y):

    return 1/2+1/8*y
```

Yuqorida *gradirovka sarf* deb saqlangan funksiya yaratilgan. Ushbu funksiya sarf datchigidan kelgan signalni *Domain* klassi oraliq diapozoniga tegishli qiymatga aylantiradi.

```
def gradirovka_tok(z):

    return (z+22.5)/5.625
```

Gradirovka tok deb saqlangan funksiya yaratilgan. Ushbu funksiya *fuzzylogic* metodi yordamida qayta ishlangan noaniq qiymatni 4-20 mA qiymat diapozoniga keltiradi.

```
temp = Domain("Temperature", 10, 30)

sarf = Domain("Sarf", 1, 3)

burchak = Domain("Buralish burchagi", 0, 90)

temp.cold = triangular(0,20, c=10)

temp.normal = triangular(10,30, c=20)

temp.hot = triangular(20,40, c=30)

sarf.few = triangular(0,2, c=1, c_m=1, no_m=0)

sarf.normal = triangular(1,3, c=2, c_m=1, no_m=0)

sarf.many = triangular(2,4, c=3, c_m=1, no_m=0)

burchak.minimal = S(0,15)

burchak.lesser = triangular(0,30, c=15)

burchak.less = triangular(15,45, c=30)

burchak.normal = triangular(30,60, c=45)
```

```
burchak.many = triangular(45,75, c=60)

burchak.more_many = triangular(60,90, c=75)

burchak.maximal = R(75,90)
```

Yuqorida qatorlarda harorat, sarf va buralish burchagi *Domain* klassi qayta kiritilgan.

Barcha ketma-ketlik dastur tili o'girilgach, qoidalar jadvali shakllantiriladi (1-jadval). Bunda har bir qoida jarayonni chuqur tahlil qilgan holda kritik nuqtalari qoida sifatida qaraladi.

1-jadval

Xom-ashyo sarfi (X_2)	Harorat qiymatlari (X_1)		
	past	o'rtacha	yuqori
kam	ko'p	kamroq	juda kam
o'rtacha	ko'proq	kam	kam
ko'p	juda ko'p	o'rtacha	kamroq

```
R1 = Rule({(temp.cold, sarf.few): burchak.many})

R2 = Rule({(temp.cold, sarf.normal): burchak.more_many})

R3 = Rule({(temp.cold, sarf.many): burchak.maximal})

R4 = Rule({(temp.normal, sarf.few): burchak.lesser})

R5 = Rule({(temp.normal, sarf.normal): burchak.less})

R6 = Rule({(temp.normal, sarf.many): burchak.normal})

R7 = Rule({(temp.hot, sarf.many): burchak.minimal})

R8 = Rule({(temp.hot, sarf.normal): burchak.lesser})

R9 = Rule({(temp.hot, sarf.few): burchak.less})
```

1-jadvalga muvofiq qoidalar ketma-ketligi Python tilida yoziladi. Demak *R1* qoida bo'yicha harorat past sarf esa kam bo'lganda ijrochi mexanizmning buralish burchagi ko'p bo'lsin degan shart berilgan. Shu tarzda qolgan qoidalar birin-ketin dasturga kiritiladi.

```
rules = Rule({(temp.cold, sarf.few): burchak.many,
              (temp.cold, sarf.normal): burchak.more_many,
              (temp.cold, sarf.many): burchak.maximal,
```

```
    (temp.normal, sarf.few): burchak.lesser,
    (temp.normal, sarf.normal): burchak.less,
    (temp.normal, sarf.many): burchak.normal,
    (temp.hot, sarf.few): burchak.less,
    (temp.hot, sarf.normal): burchak.lesser,
    (temp.hot, sarf.many): burchak.minimal
})
```

Yuqoridagi qoidalar umumiy qilib olingan ya’ni bitta qoida sifatida jamlangan. Pastda esa qoidalarni umumlashtirish umumiy qiymatini chiqaradi. Keyingi qatorlarda kod qismi orqali qiymati umumlashtiridi yoki har bir qiymatni yangi qatordan yozish kabi buyruqlar kiritilgan. Oxirgi qatorda esa chiqqan oxirgi natijani 4-20 mA qiymatda gradirovka qilib ko‘rsatdi.

```
rules == R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | R9 == sum([R1, R2, R3, R4, R5, R6, R7, R8, R9])

values = {sarf: gradirovka_sarf(6), temp: gradirovka_temp(17)}

print(R1(values), R2(values), R3(values), R4(values), R5(values), R6(values), R7(values), R8(values), R9(values), "=>", rules(values))

print(f"{round(rules(values))}\u00b0")

tok = rules(values)

print(f"{round(gradirovka_tok(tok))} mA")
```

Yuqoridagi barcha yozilgan kod Python dasturiga kiritildi va intepretatsiya qilindi. Natijalrni quyidagi jadval ustunlarida aks ettirildi (2-jadval). Haroratdan kelgan signal qiymati 20 mA va sarfdan kelgan signal qiymati 20 mA bo‘lganda Ijrochi mexanizm buralish burchagi 5 ^{0} ga buralishini ko‘rsatdi va ijrochi mezanizmga boradigan signal kattaligi esa 5 mA ga teng ekan.

2-jadval

Haroratdan kelgan signal qiymati, mA	Sarfdan kelgan signal qiymati, mA	Ijrochi mexanizmni buralish burchagi, graduslarda ^{0}	Ijrochi mexanizmga boradian signal qiymati, mA
20	20	5	5
10	5	31	10
20	10	19	7
20	5	28	9
5	20	79	18
10	15	50	13

10	12	41	11
4	19	83	19
5	11	62	15
14	11	25	8
18	10	20	8
15	6	21	8
4	20	84	19
5	10	61	15
5	16	71	17
8	16	58	14
8	11	49	13
20	4	30	9
4	4	59	15
13	8	22	8
9	9	42	12
11	9	33	10
10	10	40	11
7	7	21	8

Xulosa. Python dasturlash tilining imkoniyatlari juda ko‘p xususan, fuzzy logic kutubxonasi (fuzzylogic.classes) dan foydalangan holatda noaniq mantiq elementlarini qo‘llash orqali turli xil ishlab chiqarish jarayonlardagi texnologik reglamentga mos ravishda texnologik paramertni boshqarish va nazorat qilish imkoni mavjud. Bu orqali texnologik jarayonni boshqarishda optimal qaror qabul qilsa bo‘ladi. Maqolada yozilgan dastur kodini qo‘llab quvvatlovchi mikrokontrollerlarni tanlash muhim hisoblanadi.

Foydalanilgan adabiyotlar

1. N.R.Yusupbekov, H.S.Nurmuhammedov va S.G.Zokirov, “Kimyoviy texnologiya asosiy jarayon va qurilmalari”, “Sharq” nashriyoti Toshkent 2003.
2. N.R.Yusupbekov, O.A.Jumayev, G‘.B.Mahmudov, M,T,Ismoilov “Noaniq mantiq asosida intelektual boshqarish tizimlarini ishlab chiqish” Journal of advanced in engineering technology Vol.2(2) Dec 2020 y.
3. Д. Рутковская, М. Пилиньский, Л. Рутковский. Нейронные сети, генетические алгоритмы и нечеткие системы. М. Горячая линия – Телеком, 2004 г.
4. A.N.sadullayev “Issiqlik almashinish qurilmasidagi haroratni intelektual tizimlar asosida boshqarish” “Axborot oqimlari va intellektual tizimlarning ijtimoiy, iqtisodiy va texnik-texnologik tarmoqlardagi o‘rni” mavzusida xalqaro ilmiy-amaliy anjuman, Andijon-2023 10-noyabr.
5. https://en.wikipedia.org/wiki/Python_(programming_language)
6. https://revolutionpi.com/revolution-pi-series/
7. https://revolutionpi.com/analog-io-module/
8. A.N.sadullayev, O.R.Abduraxmonov “Mathematical modeling of the process of heat exchange in the technological system of oil refining” "Science and Education" Scientific Journal / ISSN 2181-0842, April 2022 / Volume 3 Issue 4,
9. A.N.Sadullayev, O.R.Abduraxmonov “Mathematical Modeling of Oil Heating Process” Middle European Scientific Bulletin ISSN 2694-9970 Volume 30 | nov-2022.