GREENHOUSE IMPLEMENTATION

FUZZY LOGIC CONTROLLER OF A GREENHOUSE

Abstract: These seedlings should provide a controlled environment for the production of plants in adequate sunlight, temperature and humidity. Better growing conditions are found in nursery stores especially by maintaining a higher indoor surface compared to the outdoor temperature. Greenhouse heating requirements therefore depend on the amount of heat loss in the building. The proposed scheme calculates consecutive online temperature data from the greenhouse and the thermal energy is regenerated according to the low-temperature energy balance using intelligent controls. The simulation results of greenhouse dynamics show the effectiveness of the proposed system without a direct model of plant statistics.

Keywords: Greenhouse, unintelligible logic control, temperature loss, temperature control, temperature, PSO.

1. INTRODUCTION

Greenhouse farming aims to protect the fields from adverse weather conditions and in recent years has become a factor in achieving controlled agricultural production. Climate control in nursery has been widely acquired over the years. The main reasons for this growing interest are related to the following agronomic and financial objectives:

(a) to extend the period of growth and potential yield;

(b) climate management to achieve the highest standards of quality;

(c) developing low productivity programs, coupled with a lack of resources and low investment potential for farmers. The main purpose of the climate control problem is to maintain the temperature within the temperature range within the appropriate range. The difficulty lies in the complexity of the conditions that create the ideal environment, caused by the day / night cycle, the growing season, the local climate, and the nature of the culture. (Bakker J C et al., 1995).

2. HEAT

Multi-seedling storage areas should be heated with vegetable production all year round. A good temperature system is one of the most important steps in the successful production of plants. Any heating system that provides the same temperature control without removing harmful substances from plants is acceptable. Suitable energy sources include natural gas, LP gas, petroleum, wood and electricity. The cost and availability of these resources will vary somewhat from one place to another. Simple, investment and operating costs are considered alternatives. Savings in operations can allow a more expensive heating system with automatic controls. Greenhouse heat requirements depend on the amount of heat loss in the building. Heat loss from home heaters usually occurs in all three ways of heat transfer: conduction, mechanical alignment and radiation. Often many types of heat exchangers occur simultaneously.

2.1 Driving

Heat is carried out by an object or between objects by direct physical contact. The operating rate between the two items depends on the location, the length of the path, the temperature difference and the physical properties of the object (as a quantity). Condensed heat transfer is easily reduced by replacing the heat exchanger with an insulator or by placing an insulator in the heat flow path.

2.2 Conference

Convection heat transfer is the body's movement of warm gas or liquid in a cold environment. Heat loss by convection inside the greenhouse occurs with air inflow and inflow (fans and air leaks).

2.3 Radiation

Radiation heat transfer occurs between two bodies without direct contact or the need for an air-like environment. Like light, heat rays follow a straight line and are reflected, transmitted or pulled when they hit an object. Glowing energy must be applied to the heat. The degree of heat transfer of radiation varies with the location of the object, and the temperature and surface of the two bodies involved.

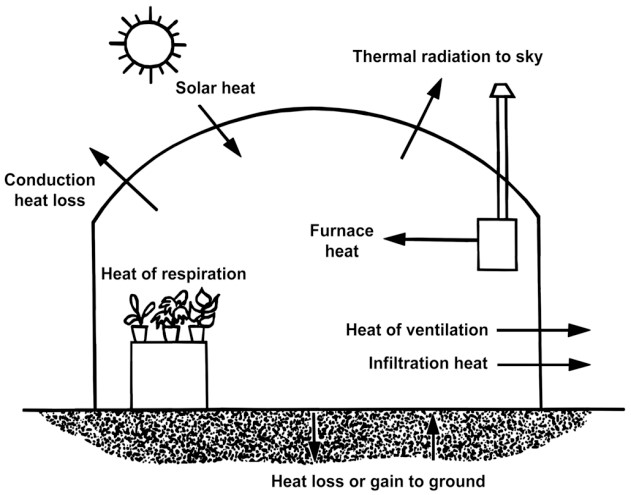
2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosted by Elsevier Ltd. All rights reserved. Peer review under the auspices of the International Federation of Automatic Control.

10.1016 / j.ifacol.2016.03.112

2.4 Factors Affecting Heat Loss

Temperature by air intake depends on age, condition and type of temperature. Old or nursery storage sheds often have cracks around the doors or holes to cover the entrance that can get a lot of cold air into them. Seedling houses covered with large sheets of molded material, large sheets of fiber glass or single or double layer of solid or flexible plastic have a small penetration (Figure 1). The sun's rays penetrate a house that absorbs heat and is absorbed by plants, soil, and heat. Warm things then re-release this energy out. The amount of heat loss dependent depends on the type of glazing, the ambient temperature and the number of cloud cover. (Seginer, et al.; 1992)

Figure 1. Loss of energy and gains in greenhouse



3. TALPERATURES ACTIVITY CALCULATION CALCULATION

Good external heat you can use in the calculation of the heater design

(Selecting heater size) can be obtained by subtracting 15 degrees F from the minimum daily temperature. Another requirement a heater must meet is to provide sufficient heat to prevent the plants from freezing during very low temperatures. (as stated at www.cps.gov.on.ca/english/plans/E6000/6701/M-6701L.pdf) For example consider table 1.

Table

|  |  |  |
| --- | --- | --- |
| **Location** | **Minimum Temperature**  **°F and (Year Occurring)** | **Average Daily Minimum January Temperatures**  **(°F)** |
| Atlanta | -8 (1985) | 33.6 |
| Athens | -4 (1985) | 33.2 |
| Augusta | -1 (1985) | 33.6 |
| Columbus | -2 (1985) | 36.4 |
| Macon | -6 (1985) | 35.8 |
| Rome | -9 (1985) | 30.5 |
| Savannah | 3 (1985) | 39.0 |
| Tifton | 0 (1985) | 38.0 |
| Valdosta | 9 (1981) | 38.6 |

In the Augusta area with an average January average temperature of 33.6 degrees F, the design temperature can be approximately 18.6 degrees F, so use 20 degrees F. This requires a 45-degree F rise above the temperature design; and, with glass, the value of R will be 0.91.

Temperature Loss Process, QC = Location x ΔT / R (1)

Ventilation Loss, QA = 0.02 x Volume x C x ΔT (2)

Perimeter Loss Heat, QP = P x L x (ΔT) (3) Total Heat Loss, QT = QC + QA + QP (4)

Where

Q = Heat loss, BTU / hr

A = Higher temperature, sq. Ft

R = Resistance to heat flow (information element) V = Greenhouse volume, cu. Ft

C = Air exchange rate per hour

P = Perimeter for equal heat loss, BTU / ft ° F hr L = Round perimeter The values ​​listed below in table 2.

Table 2: Estimates used for the estimated temperature

R (glass) 0.91

V 30,928 cu ft

C of new construction, glass or fiberglass 0.75 to 1

C classic

Construction glass, good storage 1 to 2

C classic

Construction glass, bad condition 2 to 4

The cycle of

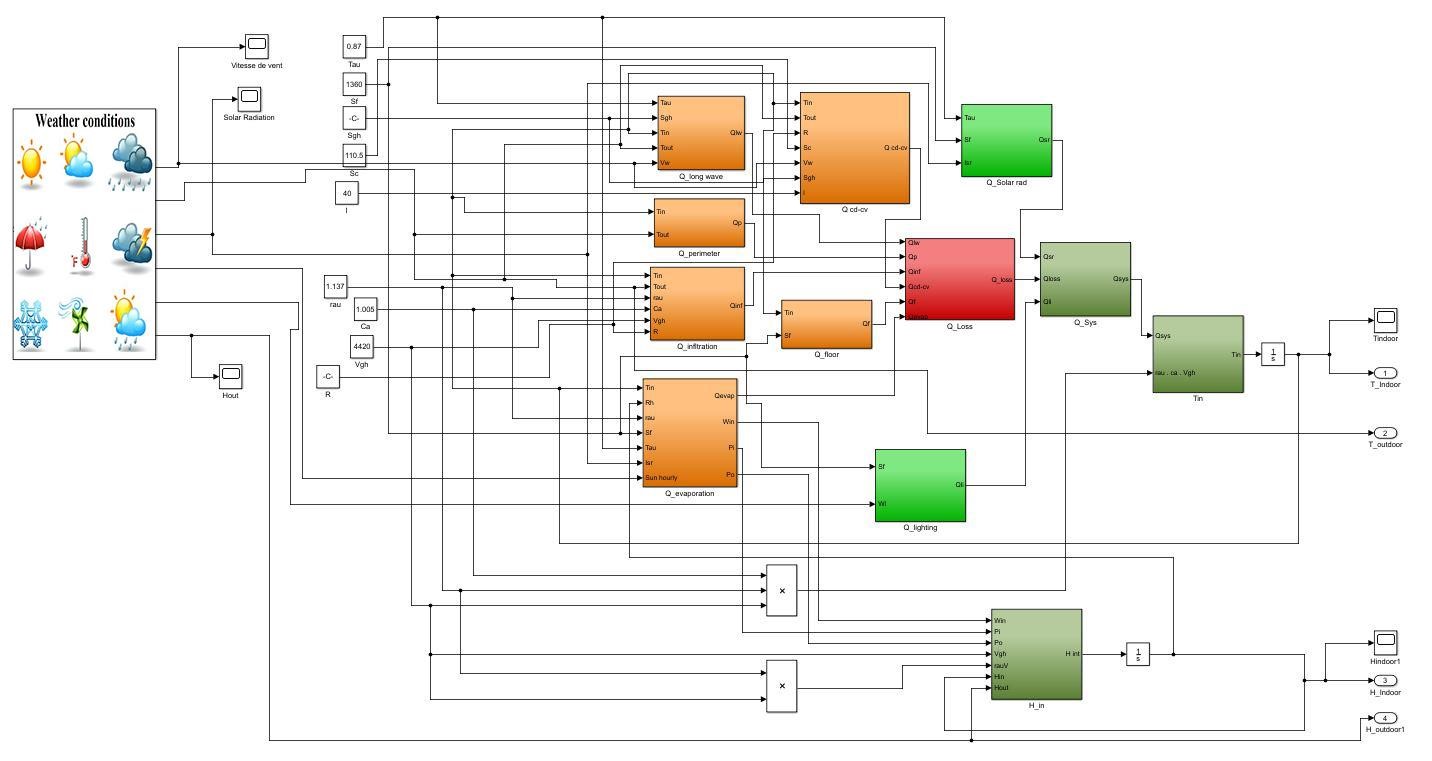
P 0.8 BTU / ft ° F not allowed

The cycle of

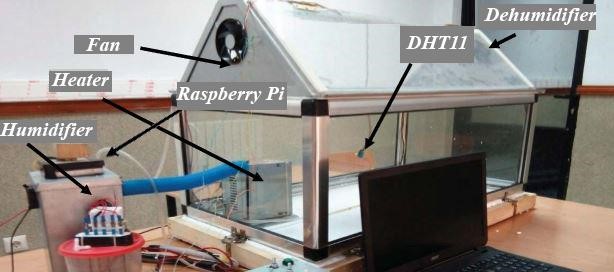
combined P 0.4 BTU / ft ° F hr

SYSTEM IMPLEMENTATION

SYSTEM SIMULATION WITH SIMULINK



COMPONENT DIAGRAM



The dimensions of the designed system are a small as spanning (w = 200 mm, L = 500 mm, H = 300 mm) covered by a 2 mm glass material.

The following are the components of the modelled system

- a Raspberry Pi 3,

- two DHT11 sensors to monitor the internal and external humidity and temperature.

-connected the required actuators in order to regulate the internal humidity and temperature

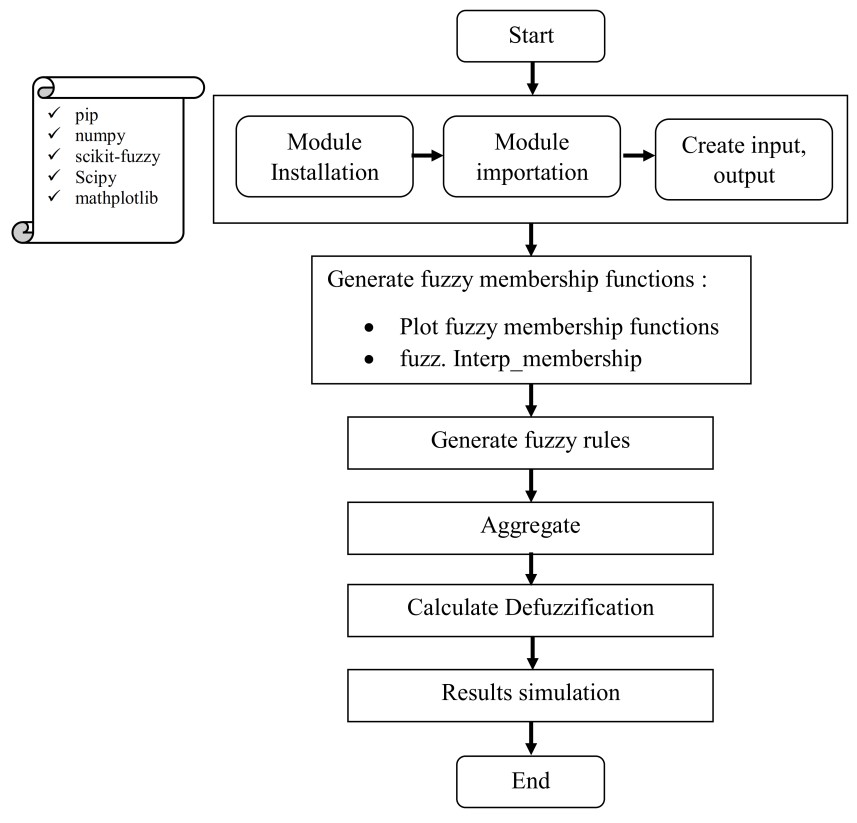
-heater

-fan

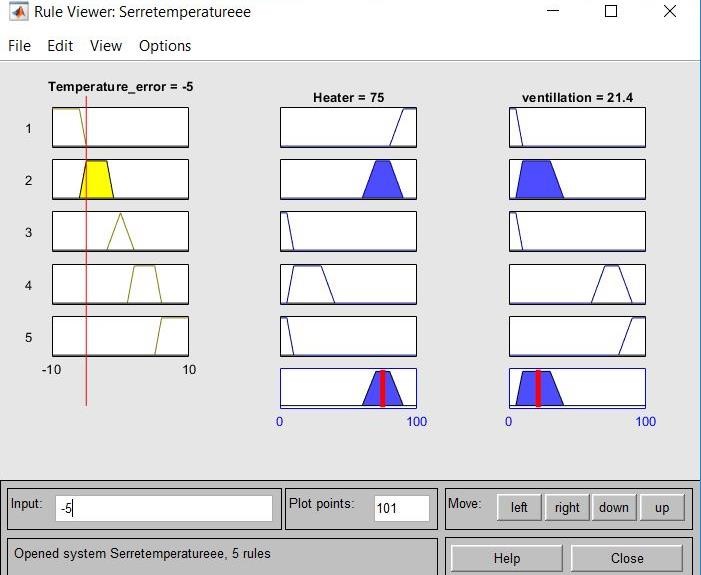
-humidifier

- dehumidifier

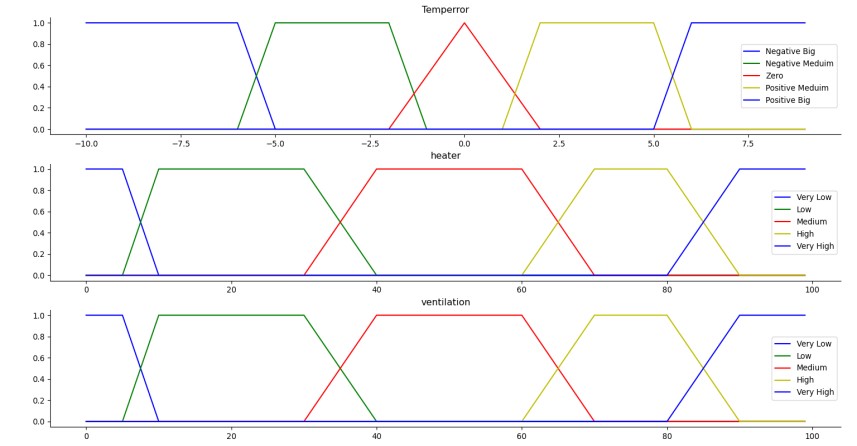
The processes and logic of the Fuzzy Logic Controller system are summarized in the following flowchart



The following are the FLC Simulink Results of the System



We conclude that for an input temperature error ΔT Figure 24 shows the input and the output of FLC of −5, i.e. for a set point temperature of 5 °C higher than implemented on the Raspberry Pi through the use of the the measured temperature, the output of the recorded control is 75% for heating and 21.4% for ventilation



The following figures indicates the output membership of the ventilation and heater systems.

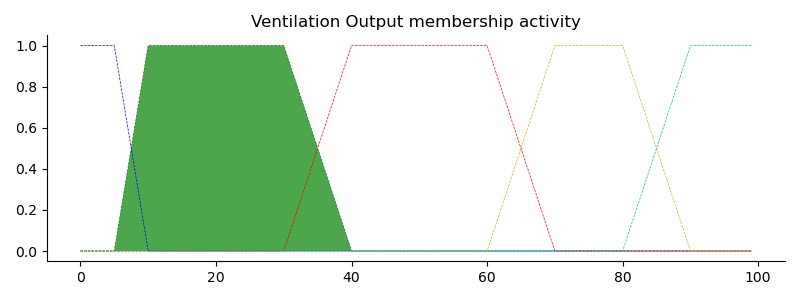


Figure 3.Output membership for ventilation activity for temperror=-5

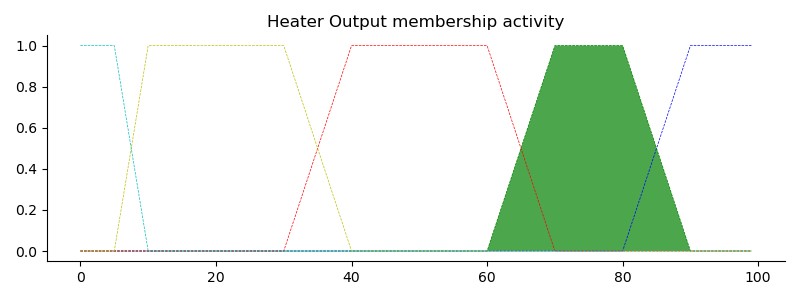


Figure 4.Aggregated membership and results for ventilation for temperror=-5

**Fuzzy Logic Controllers Implementation**

**Importing needed libraries**[**¶**](file:///E:\ONLINE%20WORKING\Modelling%20and%20Optimization%20Under%20Uncertainity\Fuzzy%20gain%20scheduling%20of%20PID%20controllers.html#Importing-needed-libraries)

In [1]:

**import** **skfuzzy** **as** **fuzz**

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

**import** **control.matlab** **as** **mt**

**Plant transfer function**

In [2]:

num = [27]

den = [1,1]

sys1 = mt.tf(num,den)

num = [1]

den = [1,3]

sys2 = mt.tf(num,den)

sys = sys1\*pow(sys2,4)

sys

Out[2]:

27

--------------------------------------------

s^5 + 13 s^4 + 66 s^3 + 162 s^2 + 189 s + 81

**Step response**

In [3]:

T = np.linspace(0,10,10000)

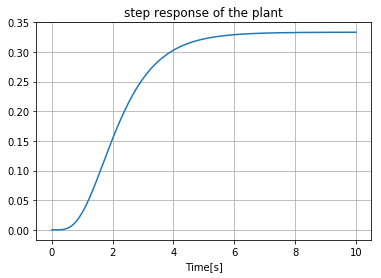
sim = mt.step(sys,T=T)

plt.plot(sim[1],sim[0])

plt.grid(**True**)

plt.title("step response of the plant")

plt.xlabel("Time[s]");



**Determination of**KuKu**,**TuTu**,**e˙maxe˙max**and**emaxemax

In [4]:

T = np.linspace(0,10,10000)

ku=10.6

*# ku=5.1*

clsys = ku\*sys/(1+ku\*sys)

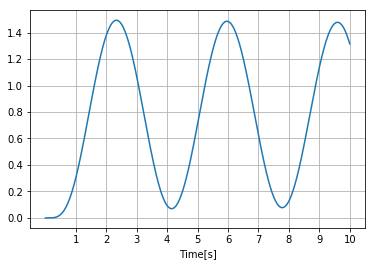
sim = mt.step(clsys,T=T)

plt.plot(sim[1], sim[0])

plt.grid(**True**)

plt.xticks(np.linspace(1,10,10));

plt.xlabel("Time[s]");



→Ku=10.6Tu=3.7→Ku=10.6Tu=3.7

In [5]:

Ku=10.6

Tu=3.7

*# Ku=5.1*

*# Tu=2.8*

kpmin=0.32\*Ku

kpmax=0.6\*Ku

kdmin=0.08\*Ku\*Tu

kdmax=0.15\*Ku\*Tu

print(kpmin,kpmax,kdmin,kdmax)

3.392 6.359999999999999 3.1376 5.883

In [6]:

np.diff(sim[0]).max()/(10/10000)

Out[6]:

1.2307967320753788

→e˙max=2emax=1→e˙max=2emax=1

In [7]:

demax=2

emax=1

**Data Base**

**Fuzzy sets for**e˙e˙**and**ee

In [8]:

**class** **triangularFuzzySet**:

**def** \_\_init\_\_(self, a, b, c):

self.a = a

self.b = b

self.c = c

A = [triangularFuzzySet(-4/3+i/3, -1+i/3, -2/3+i/3) **for** i **in** range(7)]

B = A

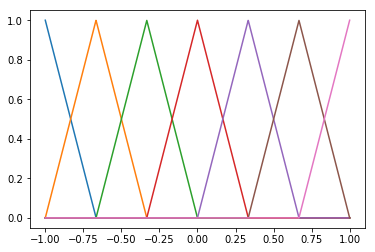
Visualization of e˙e˙ and ee

In [9]:

x = np.linspace(-1,1,2001)

**for** i **in** range(7):

plt.plot(x,fuzz.membership.trimf(x, [A[i].a, A[i].b, A[i].c]), label='A'+str(i))



**Fuzzy sets for**k′pkp′**and**k′dkd′

In [10]:

**class** **logarithmicFuzzySet**:

**def** \_\_init\_\_(self, a, b, c):

self.a = a

self.b = b

self.c = c

C = [logarithmicFuzzySet(-1/4, 0, 1), logarithmicFuzzySet(-1/4, 1, -1) ]

**Rules Base**

In [11]:

NB = 0

NM = 1

NS = 2

ZO = 3

PS = 4

PM = 5

PB = 6

S=0

B=1

Fuzzy tuning rules for k′pkp′

In [12]:

*# de(K)*

*#NB NM NS ZO PS PM PB*

rule\_base\_kp = [[B, B, B, B, B, B, B], *#NB*

[S, B, B, B, B, B, S], *#NM*

[S, S, B, B, B, S, S], *#NS*

[S, S, S, B, S, S, S], *#ZO e(K)*

[S, S, B, B, B, S, S], *#PS*

[S, B, B, B, B, B, S], *#PM*

[B, B, B, B, B, B, B]] *#PB*

Fuzzy tuning rules for k′dkd′

In [13]:

*# de(K)*

*#NB NM NS ZO PS PM PB*

rule\_base\_kd = [[S, S, S, S, S, S, S], *#NB*

[B, B, S, S, S, B, B], *#NM*

[B, B, B, S, B, B, B], *#NS*

[B, B, B, B, B, B, B], *#ZO e(K)*

[B, B, B, S, B, B, B], *#PS*

[B, B, S, S, S, B, B], *#PM*

[S, S, S, S, S, S, S]] *#PB*

Fuzzy tuning rules for αα

In [14]:

*# de(K)*

*#NB NM NS ZO PS PM PB*

rule\_base\_alpha = [[2, 2, 2, 2, 2, 2, 2], *#NB*

[3, 3, 2, 2, 2, 3, 3], *#NM*

[4, 3, 3, 2, 3, 3, 4], *#NS*

[5, 4, 3, 3, 3, 4, 5], *#ZO e(K)*

[4, 3, 3, 2, 3, 3, 4], *#PS*

[3, 3, 2, 2, 2, 3, 3], *#PM*

[2, 2, 2, 2, 2, 2, 2]] *#PB*

**Inference Mechanism**

**Degrees of compatibility**

In [15]:

*#Degrees of compatibility : MIN of MAX-MIN Inference*

**def** doc(fuzzy\_set, singleton\_input):

**if** (singleton\_input<=fuzzy\_set.a):

**return** 0

**if** (singleton\_input>fuzzy\_set.a **and** singleton\_input<=fuzzy\_set.b):

**return** (1/(fuzzy\_set.b-fuzzy\_set.a)\*(singleton\_input-fuzzy\_set.a))

**if** (singleton\_input>fuzzy\_set.b **and** singleton\_input<=fuzzy\_set.c):

**return** (1-1/(fuzzy\_set.c-fuzzy\_set.b)\*(singleton\_input-fuzzy\_set.b))

**if** (singleton\_input>fuzzy\_set.c):

**return** 0

**Inference of**k′pikpi′**and**k′dikdi′

In [16]:

**def** inference\_for\_logarithmic\_set(fuzzy\_set, value):

**return** ( (np.exp(value/fuzzy\_set.a)-fuzzy\_set.b)/fuzzy\_set.c )

**Defuzzification**

Deifned in "Fuzzy Gain Scheduling Module"

**Fuzzy Gain Scheduling Module**

In [17]:

**def** fuzzy\_gain\_scheduling(e, de):

*# Implication of u*

u = np.array([[doc(A[i], e)\*doc(A[j], de) **for** j **in** range(7)] **for** i **in** range(7)])

*# Inference of kpi and kdi*

kpi = np.array([[doc(C[rule\_base\_kp[i][j]],u[i][j]) **for** j **in** range(7)] **for** i **in** range(7)])

kdi = np.array([[doc(C[rule\_base\_kd[i][j]],u[i][j]) **for** j **in** range(7)] **for** i **in** range(7)])

*#Defuzzification*

kp = np.sum(np.multiply(kpi,u))

kd = np.sum(np.multiply(kdi,u))

alpha = np.sum(np.multiply(rule\_base\_alpha,u))

**return** kp, kd, alpha

**Simulation**

Converting transfer function of plant to state space model

In [18]:

ss=mt.tf2ss(sys)

**Fuzzy PID**

Simulation loop over 10 secconds with 0.001 second stem time

In [19]:

dt=0.001

Tf=14

time = np.linspace(0,Tf,int(Tf/dt))

kp\_mem =np.zeros((1,0))

kd\_mem =np.zeros((1,0))

ki\_mem =np.zeros((1,0))

r\_mem = np.zeros((1,0))

y\_mem = np.zeros((1,0))

x = np.zeros((5,1))

r = 1

error = 0

error\_previous = 0

d\_error = 0

y=0

i=0

**for** t **in** time:

error = r - y

d\_error = error - error\_previous

error\_previous = error

i += error

kpn, kdn, alpha = fuzzy\_gain\_scheduling(error/emax, d\_error/demax)

kp = kpmin+(kpmax-kpmin)\*kpn

kd = kdmin+(kdmax-kdmin)\*kdn

ki = kp\*kp/(alpha\*kd)

u = kp\*error + ki\*dt\*i + kd/dt\*d\_error

xd = ss.A\*x + ss.B\*u

x = x + xd\*dt

y = ss.C\*x

y\_mem = np.append(y\_mem,y)

r\_mem = np.append(r\_mem,r)

kp\_mem = np.append(kp\_mem,kp)

kd\_mem = np.append(kd\_mem,kd)

ki\_mem = np.append(ki\_mem,ki)

**PID**

In [20]:

time2 = np.linspace(0,Tf,int(Tf/dt))

r\_mem2 = np.zeros((1,0))

y\_mem2 = np.zeros((1,0))

x = np.zeros((5,1))

r = 1

error = 0

error\_previous = 0

d\_error = 0

y=0

i=0

**for** t **in** time2:

error = r - y

d\_error = error - error\_previous

error\_previous = error

i += error

kp = 8

kd = 1

ki = 1

u = kp\*error + ki\*dt\*i + kd/dt\*d\_error

xd = ss.A\*x + ss.B\*u

x = x + xd\*dt

y = ss.C\*x

y\_mem2 = np.append(y\_mem2,y)

r\_mem2 = np.append(r\_mem2,r)

In [21]:

plt.figure(figsize=(20, 5))

plt.subplot(121)

plt.title("Comparison of step response of the controlled process")

plt.plot(time,y\_mem, label='Fuzzy PID')

plt.plot(time,y\_mem2, label='PID')

plt.plot(time,r\_mem, label='Input')

plt.legend()

plt.subplot(122)

plt.title("PID parameters of the fuzzy gain scheduler")

plt.plot(time,kp\_mem, label='Kp')

plt.plot(time,ki\_mem, label='Ki')

plt.plot(time,kd\_mem, label='Kd')

plt.legend()

Out[21]:

<matplotlib.legend.Legend at 0x2b088fb1780>

