Process Control and AI Application
CHL4020

Department of Chemical Engineering

Project Report

Implementation Of Temperature Regulation of CSTR with PID and Fuzzy Controllers

By Group 2:

Devang Dobliyal (B21CH007)

Harsh Raj (B21CH008)

Jahnavi Pandey (B21CH010)

Kachhawa Goutham (B21CH011)

Implementation Of Temperature Regulation of CSTR with PID and Fuzzy Controllers

In this project we discuss the control strategy to control the temperature of CSTR reactor using the PID controller and Fuzzy logic controller. The control strategy is implemented to control the temperature of a CSTR in a closed loop.

Motivation:

In the chemical industry, the lack of ability to control the occurrence of disruptions (In case of not proper temperature control) may lead to accidents under certain circumstances. Therefore, safety measures to prevent the issues in the chemical plants are important in ensuring the plant's normal operation. CSTRs serve as workhorses for continuous production of various chemical compounds, enabling reactions such as polymerization, esterification, and neutralization on a large scale. Moreover, in wastewater treatment plants, CSTRs are instrumental in biological processes like activated sludge treatment and anaerobic digestion, effectively removing organic pollutants and producing biogas for energy generation. The food and beverage industry heavily relies on CSTRs for fermentation processes, yielding products like beer, yogurt, and bioethanol with consistent quality and yield. In pharmaceutical manufacturing, CSTRs are necessary for synthesizing pharmaceutical compounds and intermediates, ensuring high product purity and yield. Advanced control is one of the most important directions in which the production situation can be improved. Conventional ways of dealing with constraints have a lot of problems. This is why academics are looking for more flexible and powerful algorithms like bio-inspired intelligent computing, which includes the Genetic Algorithm and neural networks and Fuzzy logic. Fuzzy logic is becoming more popular around the world because it works better in real life than more rigid models of thinking. Fuzzy logic is a very open way to solve complex nonlinear problems because it doesn't cost much to compute.

The efficient process control approach will enhance the following factors:

- Operational efficiency.
- Process safety.
- Profit in material production.

PID (Proportional- Integral- Derivative) was initially popular in its deployment both in academics and industries for their transit benefits in feedback loop mechanism and easy mechanism. PID controller minimizes the void in desired and measured outputs to satisfy transient and steady state responses.

The fuzzy controllers in their industrial implementation are employed as intelligent controllers in real controlling applications. With low computational cost, fuzzy logic is highly flexible to deal with complex nonlinear problems.

The old PID controllers can only control linear systems, so they need to be updated to work with nonlinear processes in the chemical business. The fuzzy logic controller (FLC), on the other hand, has been used in a system that is not linear.

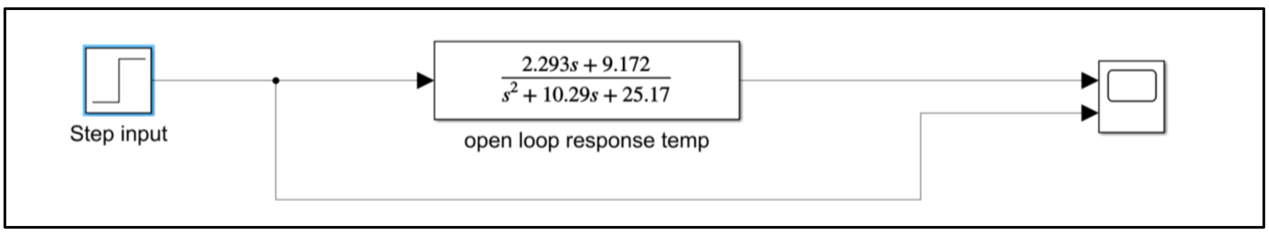
In this project we have simulated the response of PID and Fuzzy controller using MATLAB-SIMULINK for a step change in input. The model equations have been borrowed from the cited literature along with the transfer function of CSTR.

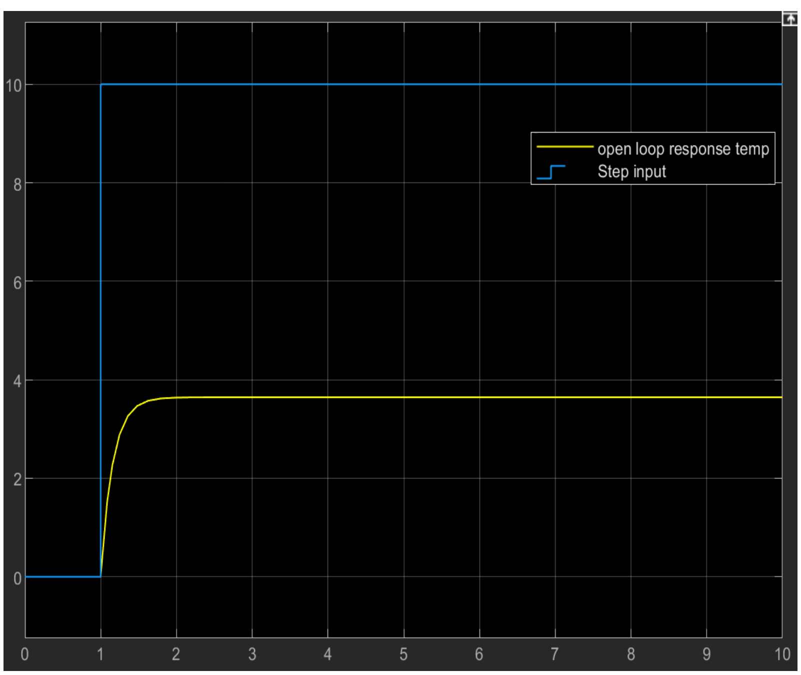
In the given process the controller and reactor are connected in a closed loop with a feedback stream from the CSTR output entering the comparator. This comparator calculates the error signal and then tunes the parameters for the controller. The response is displayed using an oscilloscope.

CSTR Transfer Function:

$$CSTR_{tf} = \frac{2.293s+9.172}{s^2+10.29s+25.17}$$

Open Loop Response:

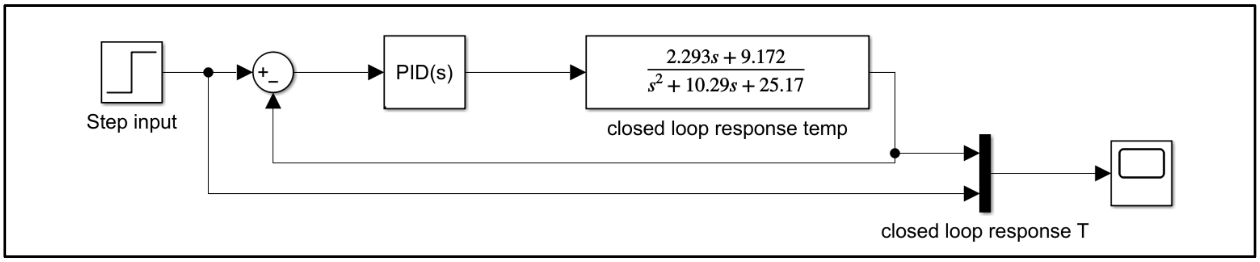


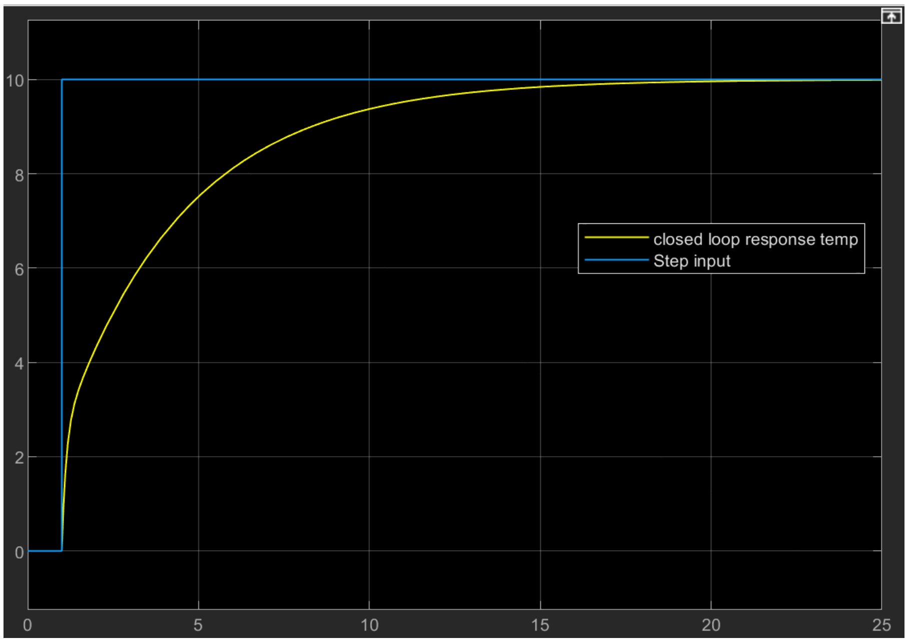



```
    RiseTime: 0.3479
TransientTime: 1.6229
 SettlingTime: 1.6229
  SettlingMin: 3.4700
  SettlingMax: 3.6440
    Overshoot: 1.6391e-06
   Undershoot: 0
         Peak: 3.6440
     PeakTime: 3.7947
```

Closed Loop Model:

Parameters = P=1; I=1; D=0

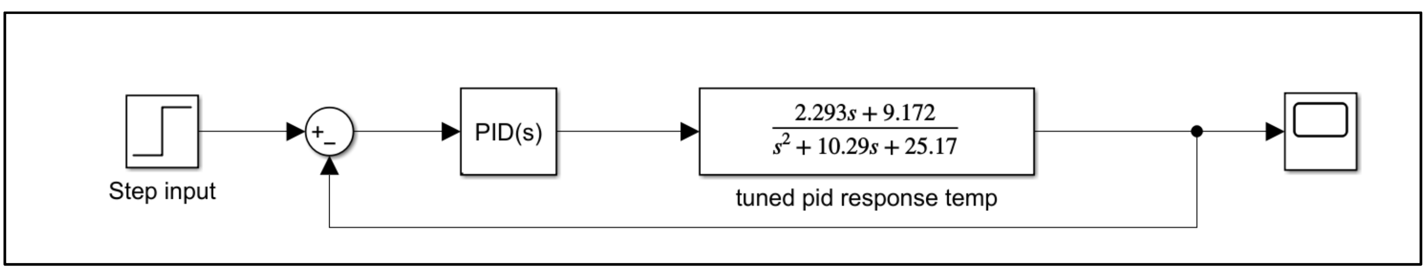


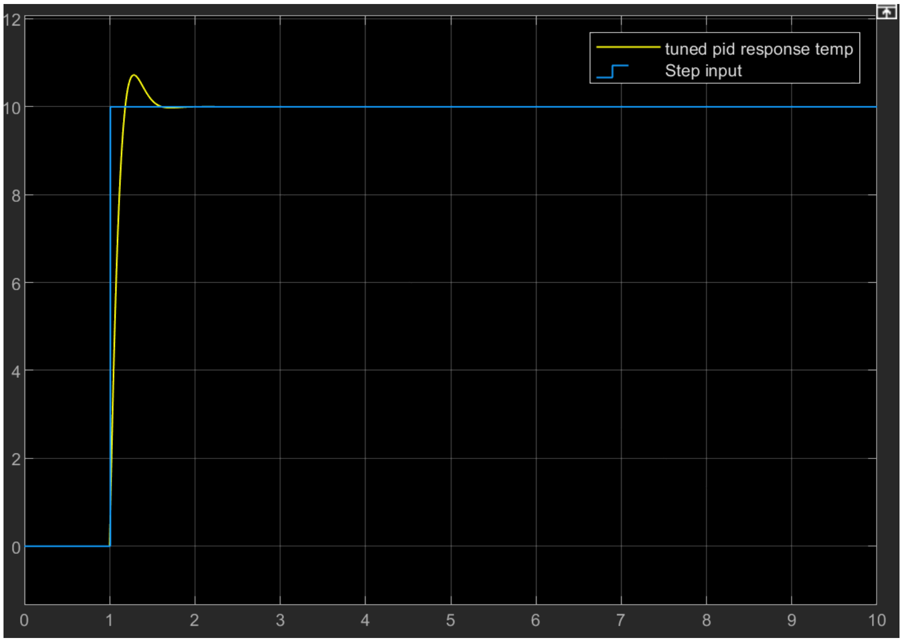



```
      RiseTime: 7.2444
 TransientTime: 14.1337
  SettlingTime: 14.1337
   SettlingMin: 9.0040
   SettlingMax: 9.9999
     Overshoot: 2.0746e-04
    Undershoot: 0
          Peak: 9.9999
      PeakTime: 48.2877
```

Auto tuned PID Parameters:

P = 4.17169602708866,
I = 50.9712560094397,
D = -0.00301394977236811

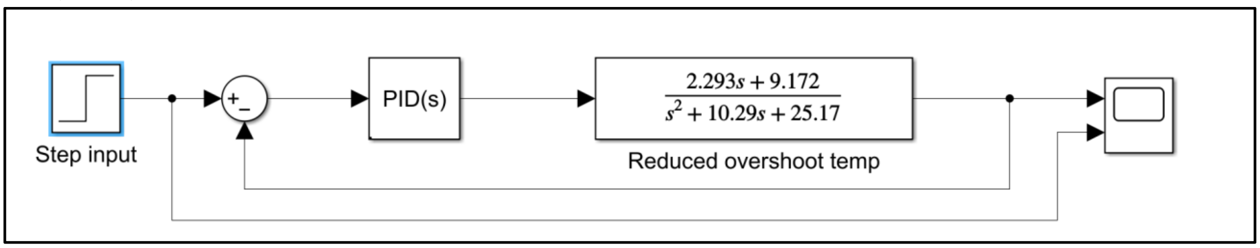


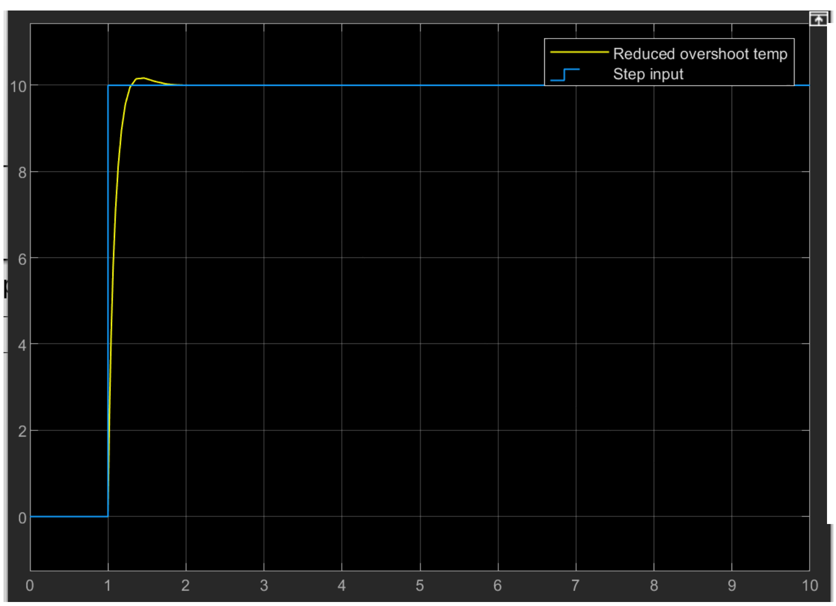


RiseTime: 0.1303
TransientTime: 1.4731
SettlingTime: 1.4731
SettlingMin: 9.0298
SettlingMax: 10.7231
Overshoot: 7.2307
Undershoot: 0
Peak: 10.7231
PeakTime: 1.2851

Slightly more response time, less overshoot:

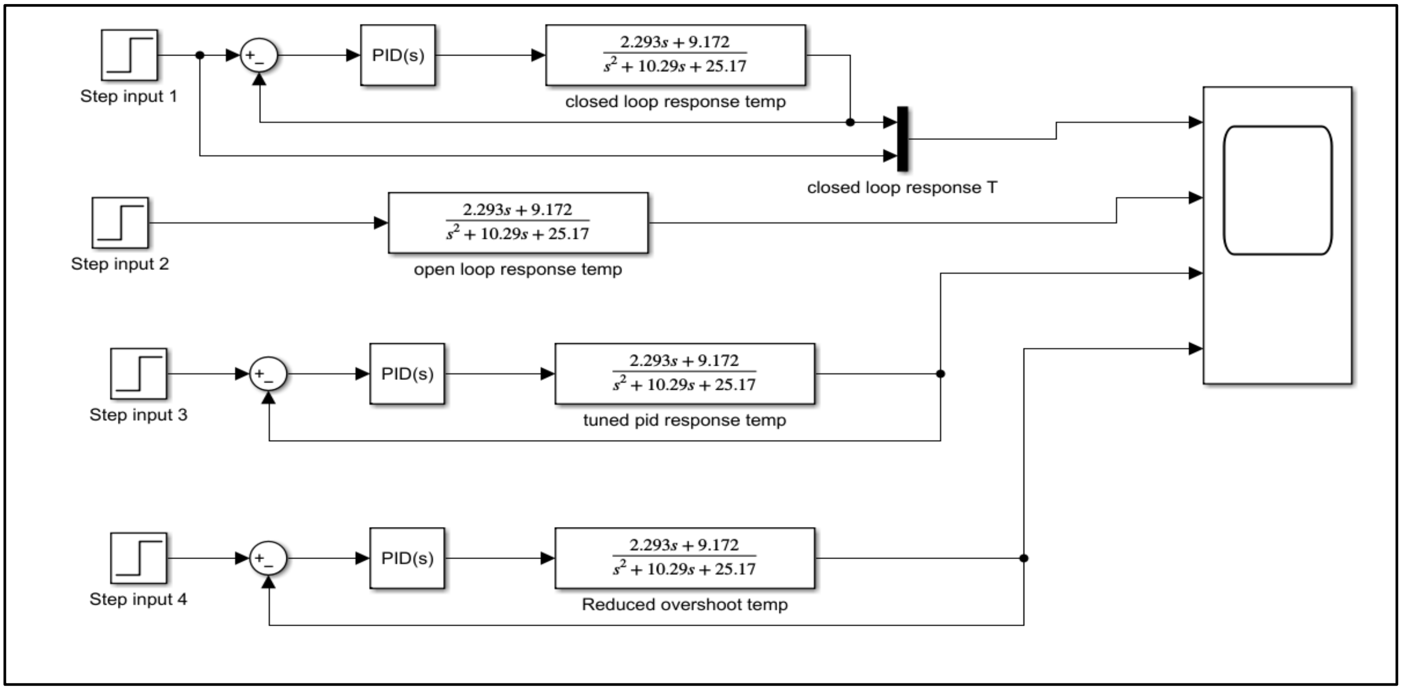
P= 5; I=40; D=0.06

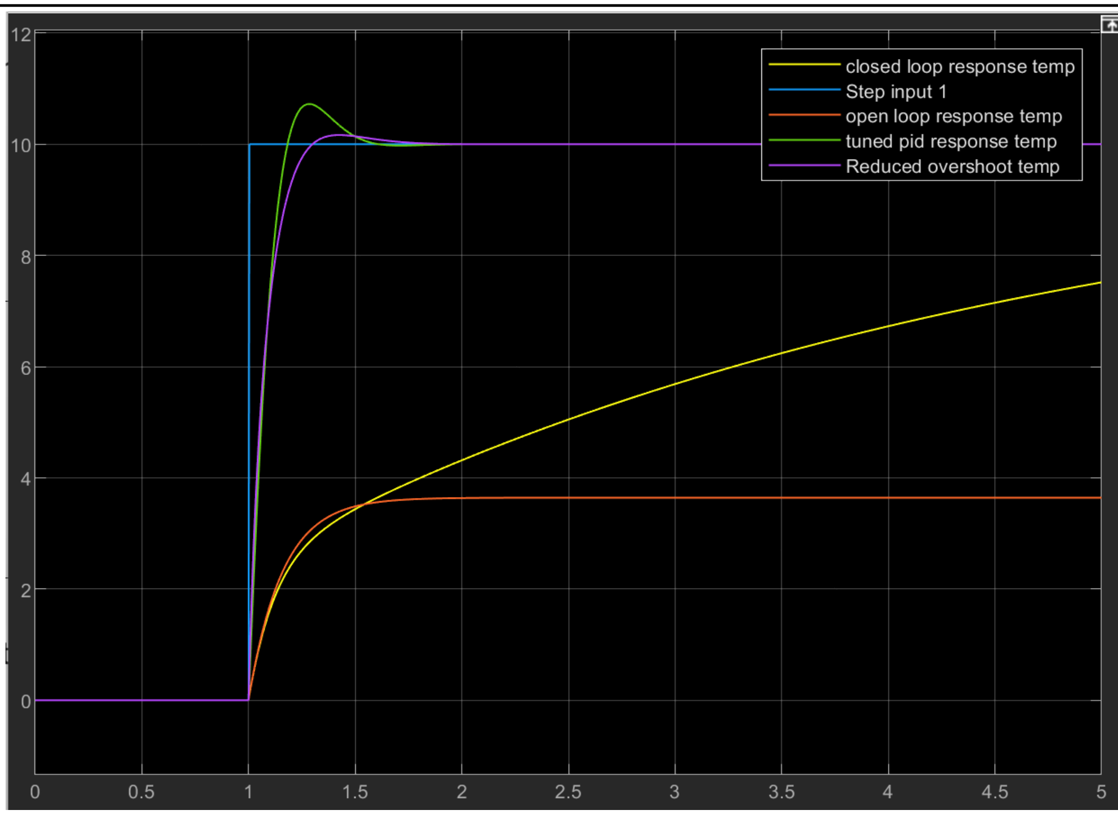




RiseTime: 0.1688
TransientTime: 1.2594
SettlingTime: 1.2594
SettlingMin: 9.5590
SettlingMax: 10.1673
Overshoot: 1.6739
Undershoot: 0
Peak: 10.1673
PeakTime: 1.4619

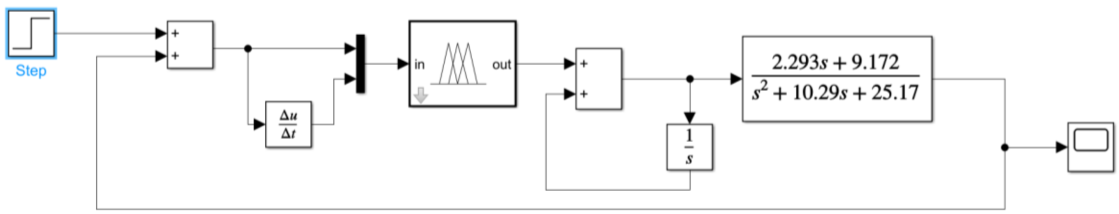
Comparison of All PID responses:



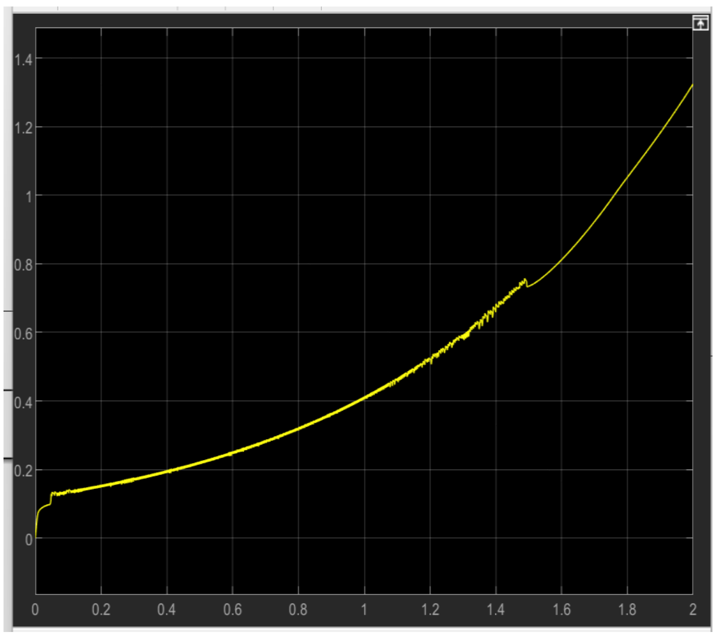


From the results we can observe that the reduced overshot temperature has the least overshot whereas the automatic tuned response has the fastest response time. We can select the design parameters for the controller based on the requirement of the process and the capacity of the equipment.

Fuzzy Controller using MATLAB:



Output:



Excepted Output:

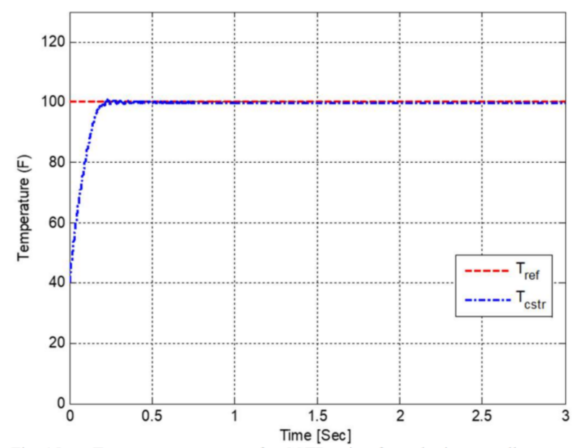
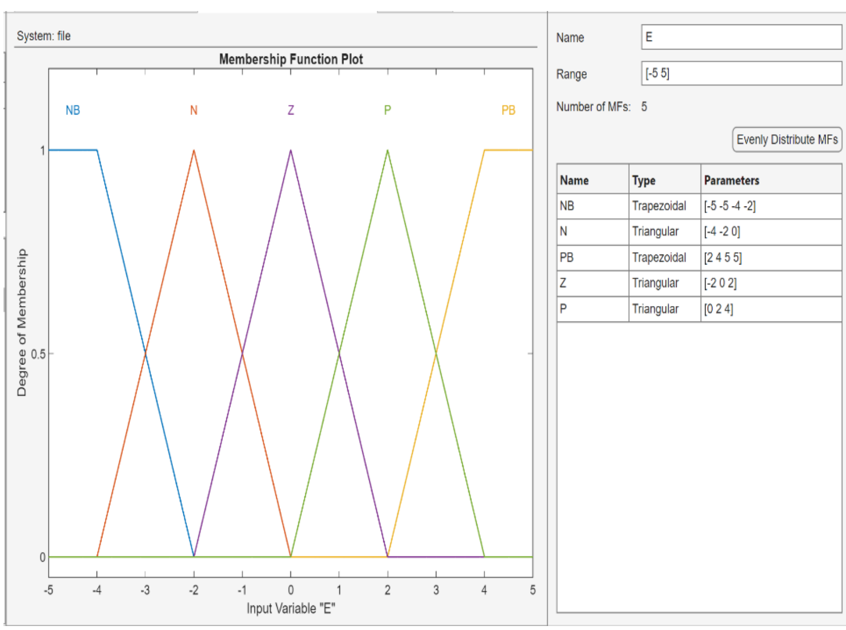
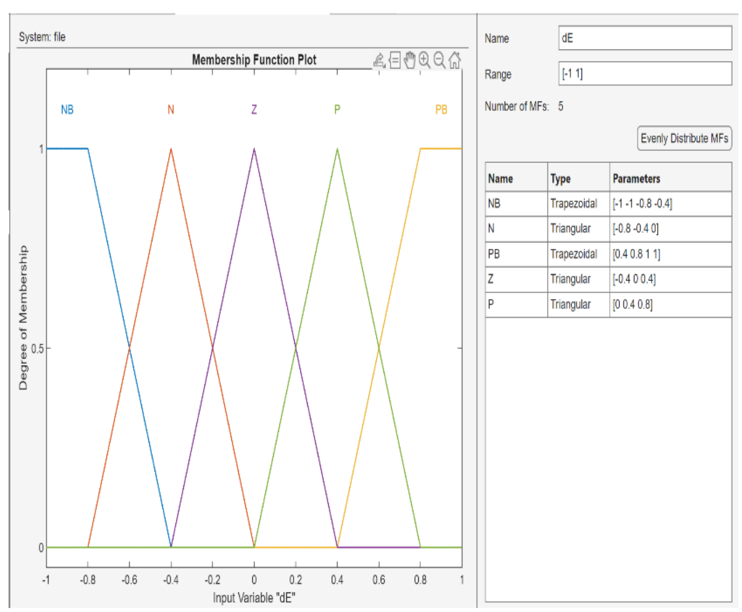


Fig. 15. Temperature response for CSTR using fuzzy logic controller

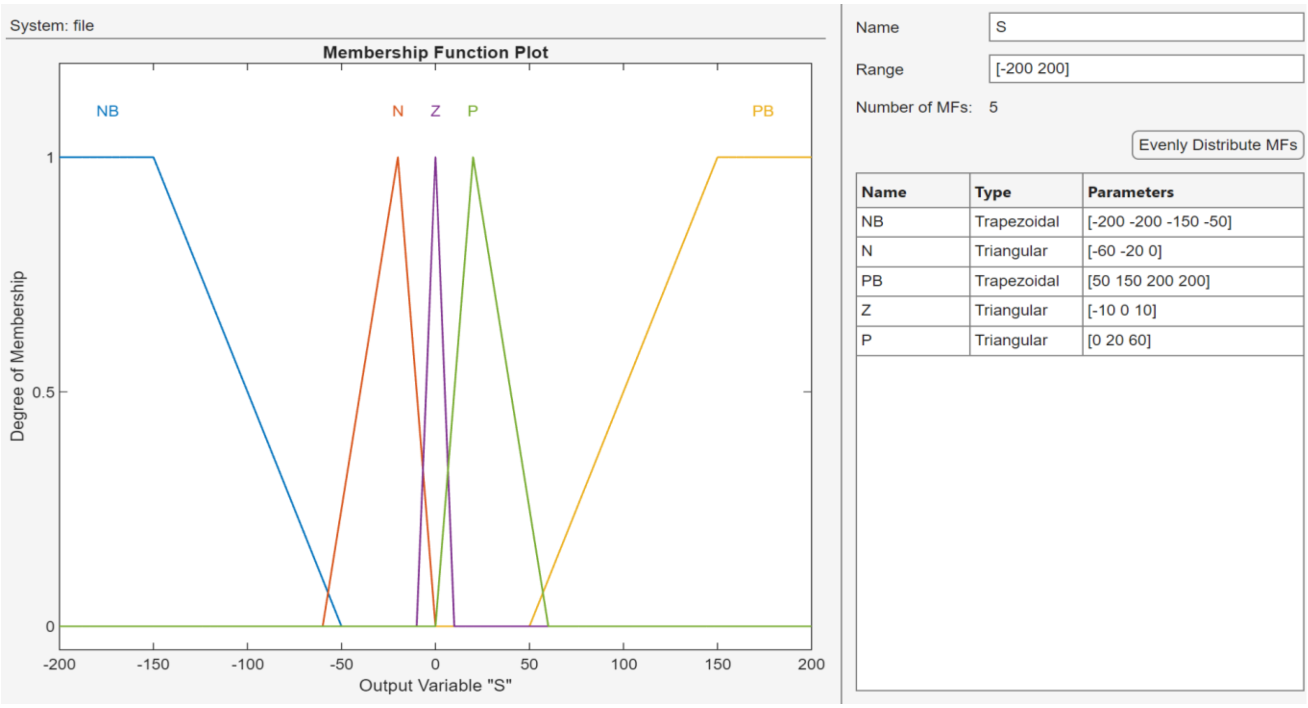
Membership function: E



Membership function:



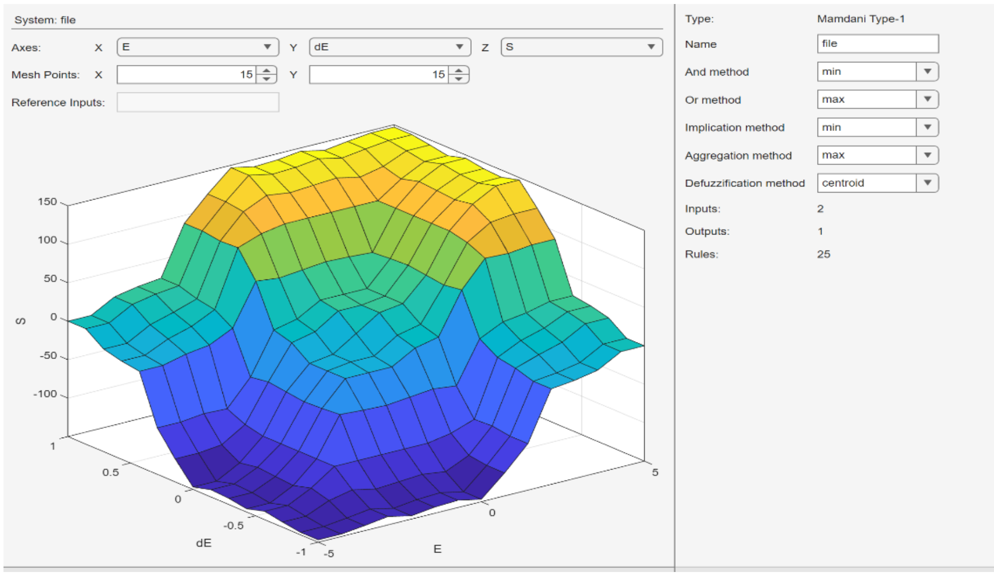
S:



Fuzzy Rules:

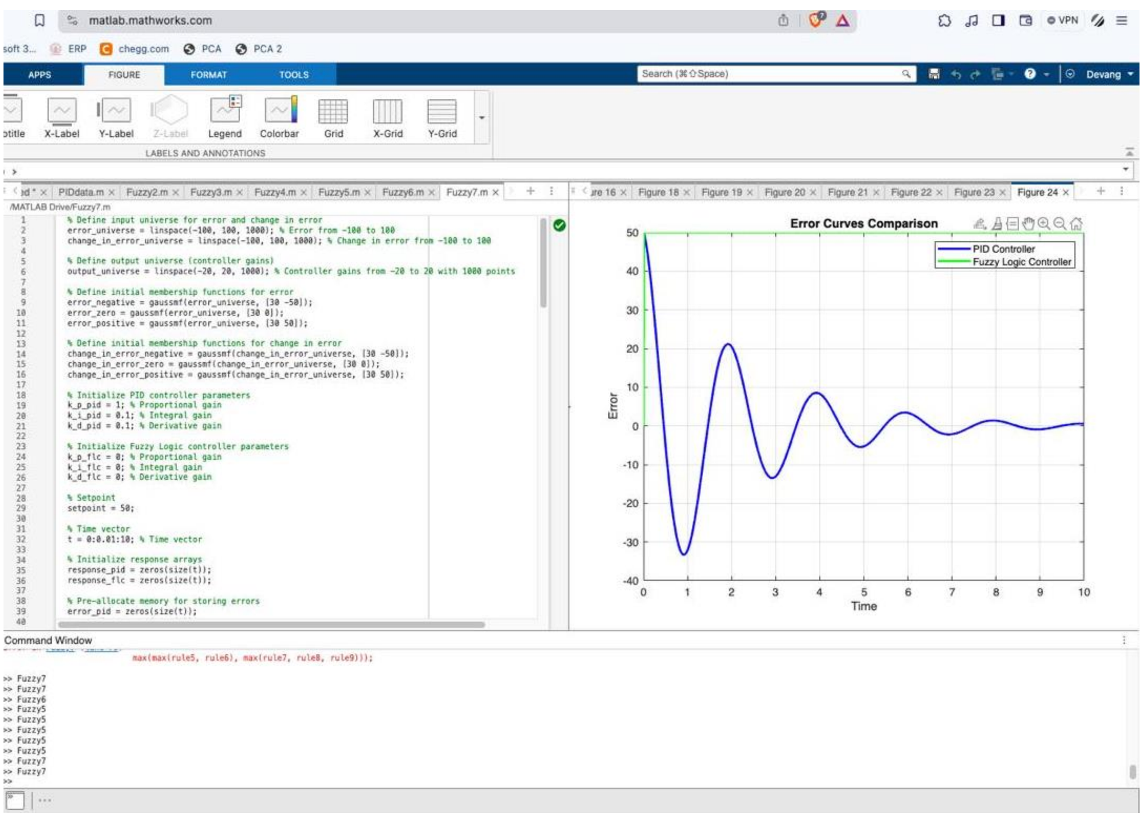
	Rule	Weight	Name
1	If E is NB and dE is NB then S is NB	1	rule1
2	If E is NB and dE is N then S is NB	1	rule2
3	If E is NB and dE is Z then S is NB	1	rule3
4	If E is NB and dE is P then S is N	1	rule4
5	If E is NB and dE is PB then S is Z	1	rule5
6	If E is N and dE is NB then S is NB	1	rule6
7	If E is N and dE is N then S is N	1	rule7
8	If E is N and dE is Z then S is N	1	rule8
9	If E is N and dE is P then S is Z	1	rule9
10	If E is N and dE is PB then S is P	1	rule10
11	If E is Z and dE is NB then S is NB	1	rule11
12	If E is Z and dE is N then S is N	1	rule12
13	If E is Z and dE is Z then S is Z	1	rule13
14	If E is Z and dE is P then S is P	1	rule14
15	If E is Z and dE is PB then S is PB	1	rule15
16	If E is P and dE is NB then S is N	1	rule16
17	If E is P and dE is N then S is Z	1	rule17
18	If E is P and dE is Z then S is P	1	rule18
19	If E is P and dE is P then S is P	1	rule19
20	If E is P and dE is PB then S is PB	1	rule20
21	If E is PB and dE is NB then S is Z	1	rule21
22	If E is PB and dE is N then S is P	1	rule22
23	If E is PB and dE is Z then S is PB	1	rule23
24	If E is PB and dE is P then S is PB	1	rule24

Fuzzy control surface:



Fuzzy Logic using MATLAB CODE:

We have used code to define the fuzzy controller and the results are:

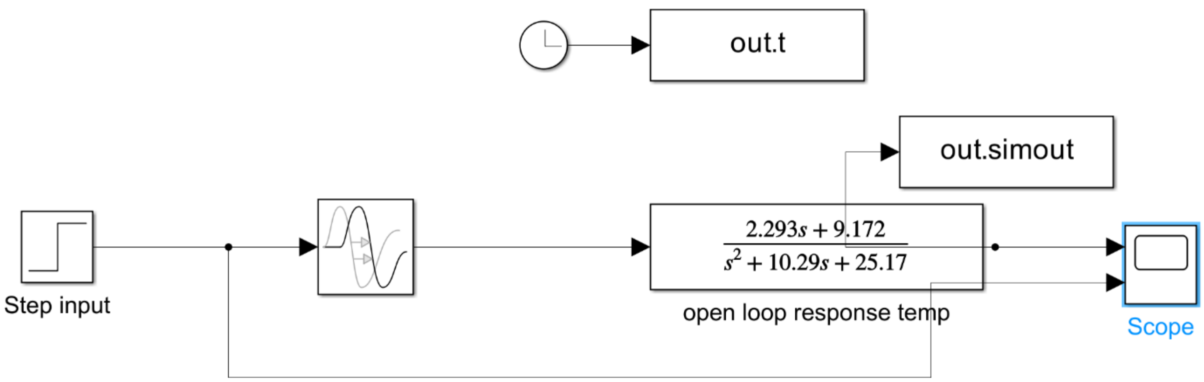


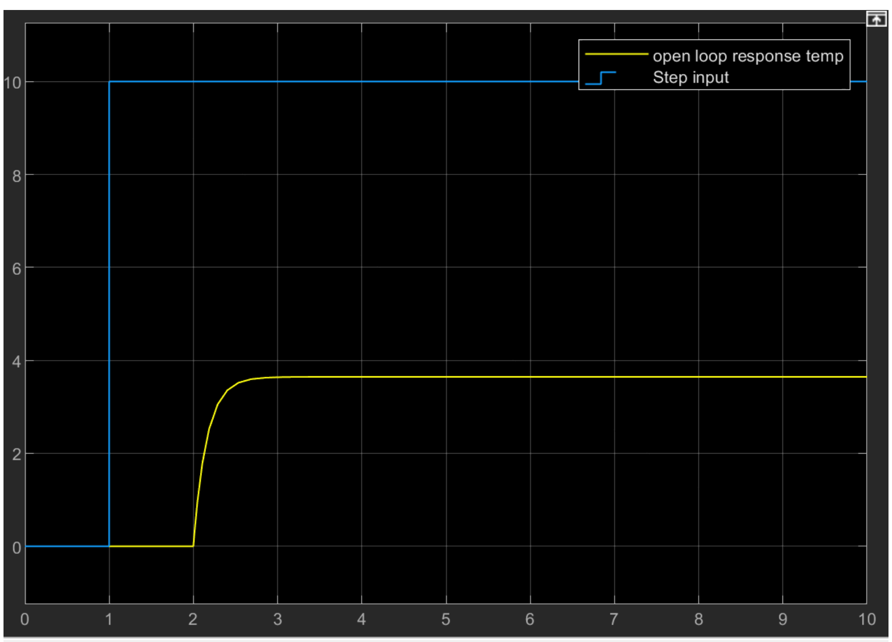
The fuzzy logic required a lot and well-defined Membership Functions which were not clearly given in the paper. So, we defined the membership after reference from some paper. Usually, the membership functions definition comes with experience.

Time delay in the system:

Introducing time in the system and overserving the response of the system and tuning the PID according to the time delay system.

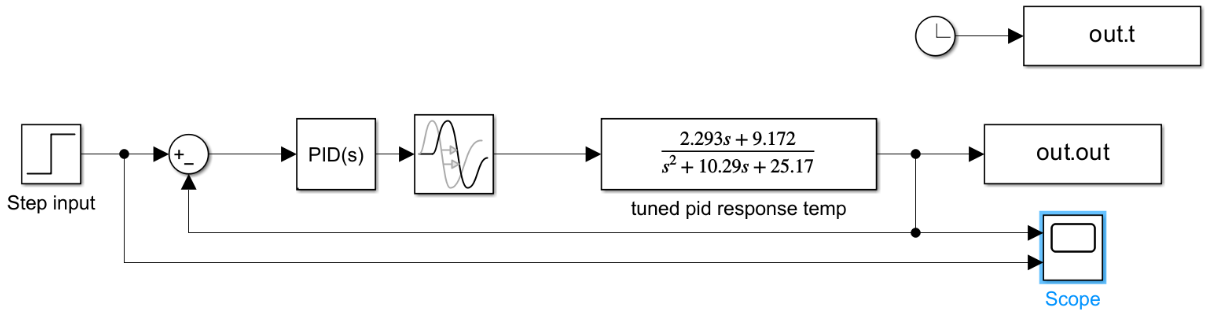
Closed loop time delay:

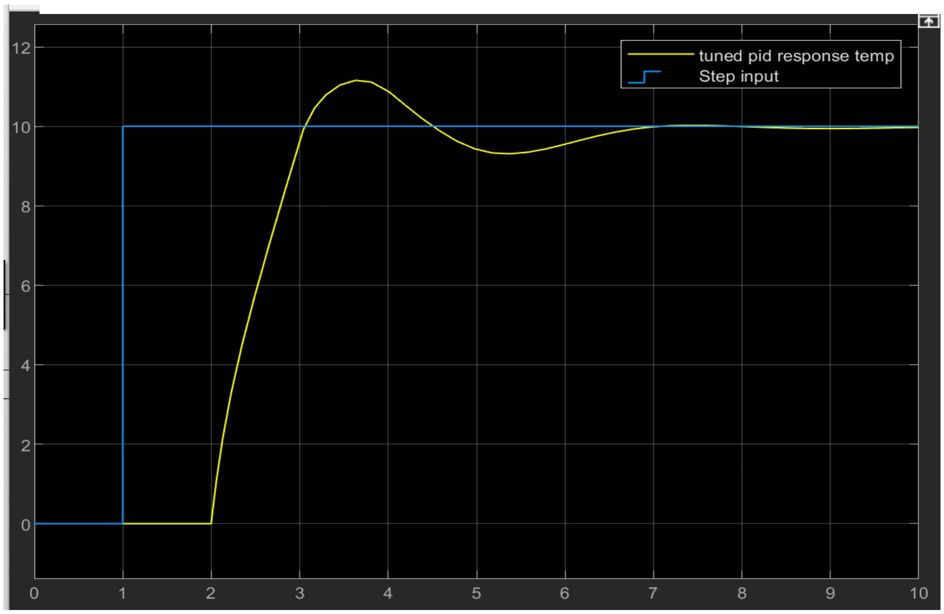





```
        RiseTime: 0.3572
   TransientTime: 2.6384
    SettlingTime: 2.6384
     SettlingMin: 3.3580
     SettlingMax: 3.6440
       Overshoot: 1.5054e-06
      Undershoot: 0
            Peak: 3.6440
        PeakTime: 4.8698
```

Time Delay closed loop tuned PID:





RiseTime: 0.8604
TransientTime: 6.3988
SettlingTime: 6.3988
SettlingMin: 9.3091
SettlingMax: 11.1573
Overshoot: 11.8778
Undershoot: 0
Peak: 11.1573
PeakTime: 3.6326

Errors faced:

The fuzzy logic controller did not produce the desired response since the system is non-linear and has to be linearized, besides the rigorous trial and error required in tuning it.

We have implemented fuzzy by using the MATLAB Fuzzy logic designer and by coding the fuzzy logic in MATLAB code, but the data was big and for this big data we should define a good membership function for better control. Which was an error in the Fuzzy control.

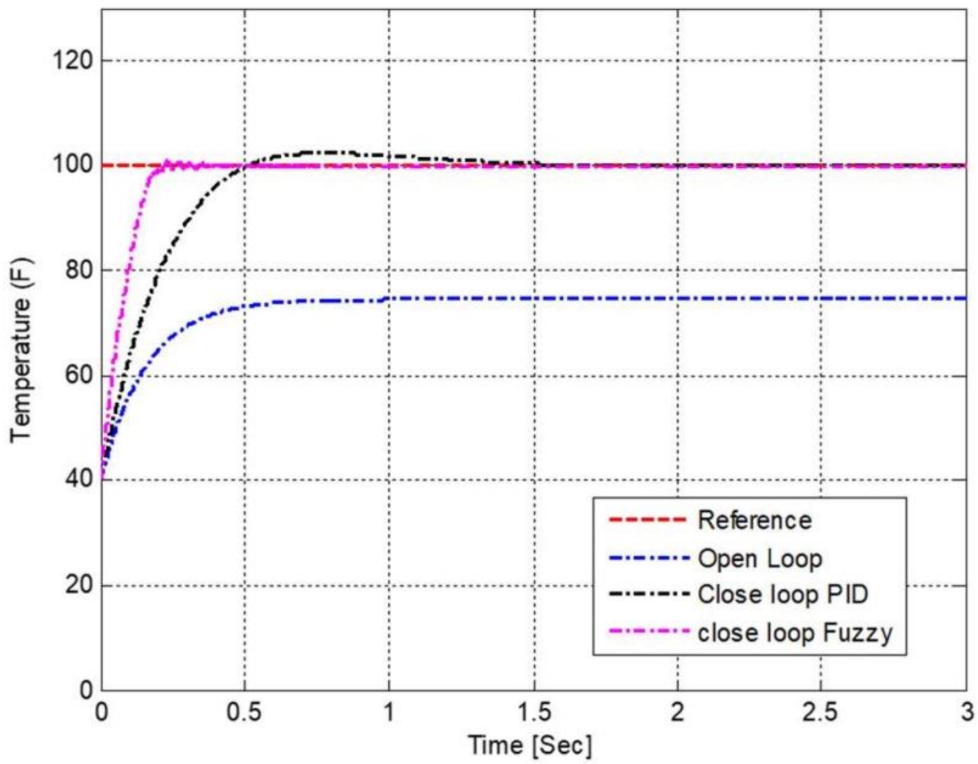
Comparison:

The closed loop fuzzy control provides the most accurate control with no overshoot and fast settling time.

The closed loop PID control is faster than an open loop but has a slight overshoot before settling at the reference temperature.

The open loop control system is the slowest and does not reach the reference temperature.

The least overshot is seen in the lowered overshoot temperature (tuned further after auto tuned), while the fastest response time is seen in the automatic tuned response. The design settings for the controller can be chosen based on what the process needs and how much the equipment can handle.



Conclusion:

This study presents a new way of designing and applying an intelligent controller that will improve the CSTR system's performance. It was shown through simulations that the suggested method is a smarter way to find the response parameters of the PID and fuzzy controllers for the CSTR system, which leads to better dynamic performance. The performance of this process was much better than the usual PID and fuzzy methods because it chose the right fuzzy rules.

It is very important that the temperature does not rise too high. When compared to the fuzzy logic controller, the PID controller responds much faster (less rise time) and has the same setting time. However, it overshoots. This doesn't happen with a fuzzy logic processor, so it's better in this case.

References:

Anirudh Damodar Kakule, Palhavi Kerkar, “Implementation of Temperature Regulation and Concentration Tracking of CSTR with Fuzzy Controllers”.

Pankaj Mohindru, “Review on PID, fuzzy and hybrid fuzzy PID controllers for controlling non-linear dynamic behaviour of chemical plants”. DOI : https://doi.org/10.1007/s10462-024-10743-0