Process System Engineering #2

#03150796 Amane Suzuki

1 Theory

Given reaction rate constant k as a function of temperature. Constant temperature condition therefore k is common in all reactor.

mole balance on reactor 1, with flow rate v, reactor size V, and monomar concentration C_n,

$$C_0 v - kC_1 V = C_1 v, \tag{1}$$

$$\frac{C_1}{C_0} = \frac{v}{v + kV}, \tag{2}$$

mole balance on reactor 2,

$$C_1 v - kC_2 V = C_2 v, \tag{3}$$

$$\frac{C_2}{C_0} = \frac{C_2}{C_1}\frac{C_1}{C_0} = \left(\frac{v}{v + kV}\right)^2, \tag{4}$$

therefore,

$$\frac{C_n}{C_0} = \left(\frac{v}{v + kV}\right)^n. \tag{5}$$

In terms of weight fraction of monomar γ_n,

$$\frac{\gamma_n}{\gamma_0} = \left(\frac{v}{v + kV}\right)^n, \tag{6}$$

moreover,

$$1 - \zeta = \frac{\gamma_N}{\gamma_0} = \left(\frac{v}{v + kV}\right)^N, \tag{7}$$

$$\frac{v}{v + kV} = (1 - \zeta)^{\frac{1}{N}}, \tag{8}$$

$$\tag{9}$$

therefore,

$$\gamma_n = (1 - \zeta)^{\frac{n}{N}} \gamma_0. \tag{10}$$

2 Algorithm

Power consumption P of each reactor is defined by γ_{in}, γ_{out}, and invariable conditions (e.g. ρ, ΔH, etc.)

therefore the structure of the source program is,

```
class Plant
  initialize(N, gamma_0, T2)
  calc()
  reactor(gamma_in, gamma_out)
end

plant = Plant.new(N, gamma_0, T2)
plant.calc()
```

In `initialize()` function, we setup

- flow rate v
- reactor size V
- diameter of reactor D
- height of reactor H

based on the method of Home Task #1.
Given reactor size V is calculated by,

$$V = \frac{v\tau}{N}, \tag{11}$$

In `reactor()` function, we calculate

- viscosity μ
- heat transfer rate h
- dimentionless numbers Pr, Nu, Re
- revolution number n
- power consumption P

In `calc()` function,

1. calculate γ_n by equation (8)
2. call `reactor(`γ_{n-1}`,`γ_n`)` from $n = 1$ to $n = N$
3. output total power consumption P_{tot}

3 Result

(1) $N = 1, \gamma_0 = 0.05$

Use listing 1,

```
$ ruby plant.rb
input n[-], gamma_0[wt%], T2[K]
1
0.05
258
Conditions:
(N, gamma_0, T2)=(1, 0.05, 258)
Reactor Size
V = 514.706[m3]
D = 7.959[m]
H = 10.346[m]
Result:
#1
Re = 6893.441
n = 0.316[rps]
P = 32745.985[W]
Total:
Ptot = 32745.985[W]
```

volume	514.706 m^3
diameter	7.959 m
height	10.346 m
total power consumption	32,745.985 W

Table 1: Result in $(N, \gamma_0, T_2) = (1, 0.05, 258)$

(2) $N = 3, \gamma_0 = 0.04$

Use listing 1,

```
$ ruby plant.rb
input n[-], gamma_0[wt%], T2[K]
3
0.04
258
Conditions:
(N, gamma_0, T2)=(3, 0.04, 258)
Reactor Size:
V = 214.461[m3]
D = 5.944[m]
H = 7.728[m]
Results:
#1
Re = 9653.252
n = 0.082[rps]
P = 120.778[W]
#2
Re = 3062.065
n = 0.103[rps]
P = 334.018[W]
#3
Re = 1378.443
n = 0.092[rps]
P = 293.009[W]
Total:
Ptot = 747.805[W]
```

volume	$214.461\,\mathrm{m}^3$
diameter	$5.944\,\mathrm{m}$
height	$7.728\,\mathrm{m}$
total power consumption	$747.805\,\mathrm{W}$

Table 2: Result in $(N, \gamma_0, T_2) = (3, 0.04, 258)$

(3) $N = 1 \mathrm{to} 5, \gamma_0 = 0.02 \mathrm{to} 0.10$

Use listing 2,

```
$ ruby plant-advanced.rb
2.47 0.31 0.12 0.06 0.04
6.92 0.87 0.33 0.18 0.11
15.9 2.01 0.75 0.41 0.26
32.75 4.13 1.54 0.84 0.54
62.92 7.94 2.96 1.61 1.04
115.28 14.55 5.42 2.95 1.91
203.95 25.75 9.59 5.22 3.38
351.32 44.35 16.52 8.99 5.82
592.59 74.81 27.87 15.16 9.81
```

	N				
γ_0	1	2	3	4	5
0.02	2.47	0.31	0.12	0.06	0.04
0.03	6.92	0.87	0.33	0.18	0.11
0.04	15.9	2.01	0.75	0.41	0.26
0.05	32.75	4.13	1.54	0.84	0.54
0.06	62.92	7.94	2.96	1.61	1.04
0.07	115.28	14.55	5.42	2.95	1.91
0.08	203.95	25.75	9.59	5.22	3.38
0.09	351.32	44.35	16.52	8.99	5.82
0.10	592.59	74.81	27.87	15.16	9.81

Table 3: Power Consumption P[kW] in each condition

4 Source Program

Listing 1: plant.rb

```ruby
include Math

GP = (63.0*1000/24) # product flow rate [kg h-1]
ML = 50.0 # polymer length
TAU = 5.0 # residence time
RHO = 850.0 # density [kg m-3]
ALPHA = 1.3 # height/diameter of reactor
HP = (72.8*1000) # heat of polymerization [J mol-1]
CV = 0.6 # conversion
T1 = (273+50) # [K]
TC = 0.128 # thermal conductivity [W m-1 K-1]
CP = (1.68*1000) # specific heat of toluene [J kg-1 K-1]
M = 54.0 # molecular weight of butadiene [g mol-1]

class Plant
  def initialize(n, feed, coolant)
    @n = n # number of reactors [-]
    @feed = feed # feed [wt%]
    @rate = GP/(RHO*CV*feed) # feed speed [m3 h-1]
    @volume = @rate*TAU/n # [m3]
    @diameter = (@volume/(ALPHA*2*PI))**(1/3.0)*2 # [m]
    @height = @diameter*ALPHA # [m]
    @coolant = coolant # T2 [K]
  end

  def calc()
    tpc = 0 # total power consumption [W]
    prop = (1-CV)**(1.0/@n) # proportional constant [-]

    # show conditions
    puts "Conditions:"
    puts "(N, gamma_0, T2)="+
      "("+@n.to_s+", "+@feed.to_s+", "+@coolant.to_s+")"

    # show reactor size
    puts "Reactor Size:"
    puts "V = "+@volume.round(3).to_s+"[m3]"
    puts "D = "+@diameter.round(3).to_s+"[m]"
    puts "H = "+@height.round(3).to_s+"[m]"

    # show result of each reactor
    puts "Results:"
    for n in 1..@n
      puts "#"+n.to_s
      tpc += reactor(@feed*(prop**(n-1)),@feed*(prop**n))
    end

    # show total power consumption
    puts "Total:"
    puts "Ptot = "+tpc.round(3).to_s+"[W]"
```

```
  end

  def reactor(gamma_in, gamma_out)
    # viscosity [Pa s]
    visc = ((ML)**1.7)*((1-(gamma_out/@feed))**2.5)*exp(21.0*@feed)*1e-3

    # heat transfer rate
    h = HP*(@rate*RHO*(gamma_in-gamma_out)*1000/3600)/M/(@diameter*PI*@height)/(T1-@coolant)

    # dimensionless numbers
    pr = visc*CP/TC
    nu = h*@diameter/TC
    re = (2*nu/pr**(1/3.0))**1.5
    puts "Re = "+re.round(3).to_s

    # revolution number
    revnum = re*visc/RHO/(@diameter/2)**2
    puts "n = "+revnum.round(3).to_s+"[rps]"

    # power consumption
    np = 14.6*re**(-0.28)
    p = np*RHO*(revnum**3)*(@diameter/2)**5
    puts "P = "+p.round(3).to_s+"[W]"

    return p
  end
end

puts "input n[-], gamma_0[wt%], T2[K]"
n = gets.to_i
feed = gets.to_f
coolant = gets.to_i

plant = Plant.new(n, feed, coolant)
plant.calc()
```

Listing 2: plant-advanced.rb

```ruby
include Math

GP = (63.0*1000/24) # product flow rate [kg h-1]
ML = 50.0 # polymer length
TAU = 5.0 # residence time
RHO = 850.0 # density [kg m-3]
ALPHA = 1.3 # height/diameter of reactor
HP = (72.8*1000) # heat of polymerization [J mol-1]
CV = 0.6 # conversion
T1 = (273+50) # [K]
TC = 0.128 # thermal conductivity [W m-1 K-1]
CP = (1.68*1000) # specific heat of toluene [J kg-1 K-1]
M = 54.0 # molecular weight of butadiene [g mol-1]

class Plant
  def initialize(n, feed, coolant)
    @n = n # number of reactors [-]
    @feed = feed # feed [wt%]
    @rate = GP/(RHO*CV*feed) # flow rate [m3 h-1]
    @volume = @rate*TAU/n # [m3]
    @diameter = (@volume/(ALPHA*2*PI))**(1/3.0)*2 # [m]
    @height = @diameter*ALPHA # [m]
    @coolant = coolant # T2 [K]
  end

  def calc()
    tpc = 0 # total power consumption [W]
    prop = (1-CV)**(1.0/@n) # proportional constant [-]

    for n in 1..@n
      tpc += reactor(@feed*(prop**(n-1)),@feed*(prop**n))
    end

    return ((tpc)/1000).round(2)
  end

  def reactor(gamma_in, gamma_out)
    # viscosity [Pa s]
    visc = ((ML)**1.7)*((1-(gamma_out/@feed))**2.5)*exp(21.0*@feed)
        *1e-3

    # heat transfer rate
    h = HP*(@rate*RHO*(gamma_in-gamma_out)*1000/3600)/M/(@diameter*
        PI*@height)/(T1-@coolant)

    # dimensionless numbers
    pr = visc*CP/TC
    nu = h*@diameter/TC
    re = (2*nu/pr**(1/3.0))**1.5

    # revolution number
    revnum = re*visc/RHO/(@diameter/2)**2

```

```
    # power consumption
    np = 14.6*re**(-0.28)
    p = np*RHO*(revnum**3)*(@diameter/2)**5

    return p
  end
end

coolant = 258
for feed in 2..10
  f = feed*0.01
  for n in 1..5
    plant = Plant.new(n, f, coolant)
    print plant.calc().to_s + "\t"
  end
  print "\n"
end
```