

SPRAWOZDANIE Z TRZECIEJ CZĘŚCI
PROJEKTU Z PRZEDMIOTU
„TECHNIKA AUTOMATYZACJI
PROCESÓW”

Numer zadania: 3

Wykonawcy:

Dawidiuk Marek

Giełdowski Daniel

Kłos Maciej

Taras Sylwia

Spis treści

1. Opis otrzymanego modelu	2
2. Implementacja obiektu	4
3. Regulator obiektu	8

1. Opis otrzymanego modelu

W ramach tego projektu korzystaliśmy z modelu obiektu znanego jako reaktor przepływowy. Obiekt składa się z pojemnika wypełnionego cieczą z rozpuszczoną nieokreśloną substancją. Do pojemnika wpływa strumieniem F_{in} ciecz o określonej temperaturze T_{in} oraz stężeniu rozpuszczonej substancji C_{Ain} . W pojemniku jest określona ilość cieczy V w określonej temperaturze T . Ciecz z pojemnika wypływa strumieniem F , zawierając stężenie C_A rozpuszczonej substancji. Dodatkowo przez pojemnik przeprowadzona jest rura odpowiedzialna za chłodzenie bądź podgrzewanie, którą strumieniem F_C płynie ciecz o temperaturze wejściowej T_{Cin} . Obiekt opisany jest następującymi równaniami:

$$\begin{cases} V \cdot \frac{dC_A}{dt} = F_{in} \cdot C_{Ain} - F \cdot C_A - V \cdot k_0 \cdot e^{-\frac{E}{R \cdot T}} \cdot C_A \\ V \cdot \rho \cdot c_p \cdot \frac{dT}{dt} = F_{in} \cdot \rho \cdot c_p \cdot T_{in} - F \cdot \rho \cdot c_p \cdot T + V \cdot h \cdot k_0 \cdot e^{-\frac{E}{R \cdot T}} \cdot C_A - \frac{a \cdot (F_C)^{b+1}}{F_C + \frac{a \cdot (F_C)^b}{2 \cdot \rho_c \cdot c_{pc}}} \cdot (T - T_{Cin}) \end{cases} \quad (1.1)$$

W równaniach występują stałe o podanych wartościach:

- $\rho = \rho_c = 10^6 \frac{g}{m^3}$
- $c_p = c_{pc} = 1 \frac{cal}{g \cdot K}$
- $k_0 = 10^{10} \frac{1}{min}$
- $\frac{E}{R} = 8330,1 \frac{1}{K}$
- $h = 130 \cdot 10^6 \frac{cal}{kmol}$
- $a = 1,678 \cdot 10^6 \frac{cal}{K \cdot m^3}$
- $b = 0,5$

Otrzymaliśmy także dane odnośnie wartości zmiennych modelu w zadanym punkcie pracy układu:

- $V = 1m^3$
- $F_{in} = F = 1 \frac{m^3}{min}$
- $C_{Ain} = 2 \frac{kmol}{m^3}$
- $F_C = 15 \frac{m^3}{min}$
- $T_{in} = 323K$
- $T_{Cin} = 365K$
- $C_A = 0,26 \frac{kmol}{m^3}$
- $T = 393,9K$

W ramach zadania zmienne te podzielone zostały na 4 grupy:

- stałe - V, F, F_{in}
- regulowane - C_A, T
- sterujące - C_{Ain}, F_C
- zakłócenia - T_{in}, T_{Cin}

Po zastąpieniu stałych w równaniach liczbami, otrzymywany jest następujący układ równań:

$$\begin{cases} \frac{dC_A}{dt} = C_{Ain} - C_A - 10^{10} \cdot e^{-\frac{8330,1}{T}} \cdot C_A \\ \frac{dT}{dt} = T_{in} - T + 130 \cdot 10^{10} \cdot e^{-\frac{8330,1}{T}} \cdot C_A - \frac{1,678 \cdot (F_C)^{1,5}}{F_C + 0,839 \cdot (F_C)^{0,5}} (T - T_{Cin}) \end{cases} \quad (1.2)$$

Ostatnią rzeczą jaka musiała zostać wzięta pod uwagę było dokładniejsze określenie wartości wyjść w punkcie pracy, ponieważ te podane w zadaniu były przybliżone. Proces ten został wykonany w pierwszej części projektu, a wartości wyjść wyniosły w przybliżeniu $C_A = 0,2646$ oraz $T = 393,9531$.

2. Implementacja obiektu

Pierwszym zadaniem do wykonania podczas tej części projektu była implementacja zadanego obiektu w środowisku OVATION. Zanim jednak przystąpiliśmy do implementacji należało przekształcić model z równań ciągłych na dyskretne. Zgodnie z instrukcją podaną przez prowadzącego zostało to wykonane poprzez proste podstawienie, w którym k to numer kroku modelu, a T_s oznacza czas próbkowania:

$$\begin{cases} \frac{dC_A}{dt} = \frac{C_{A_{k+1}} - C_{A_k}}{T_s} \\ \frac{dT}{dt} = \frac{T_{k+1} - T_k}{T_s} \end{cases} \quad (2.1)$$

Jako czas próbkowania przyjęliśmy 1 sekundę. Ponieważ jednak nasz obiekt posługuje się nominalnie minutami wartość T_s wyniosła 1/60. Ostatecznie więc do implementacji otrzymaliśmy następujące równania:

$$\begin{cases} C_{A_{k+1}} = (C_{A_{in_k}} - C_{A_k} - 10^{10} \cdot e^{-\frac{8330,1}{T_k}} \cdot C_{A_k})/60 + C_{A_k} \\ T_{k+1} = (T_{in_k} - T_k + 130 \cdot 10^{10} \cdot e^{-\frac{8330,1}{T_k}} \cdot C_{A_k} - \frac{1,678 \cdot (F_{C_k})^{1,5}}{F_{C_k} + 0,839 \cdot (F_{C_k})^{0,5}} (T_k - T_{C_{in_k}}))/60 + T_k \end{cases} \quad (2.2)$$

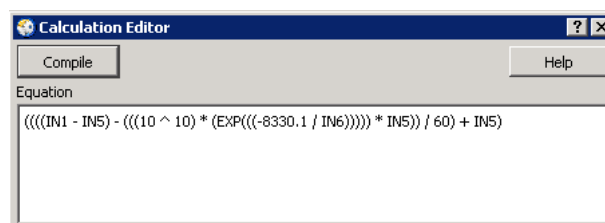
Mając tak przygotowane wzory przystąpiliśmy do implementacji, którą zaczęliśmy od utworzenia odpowiednich punktów wejściowych (4 punkty: 2 dla sterowania i 2 dla zakłóceń) i wyjściowych (2 punkty). Utworzone zostały punkty o następujących nazwach:

- 3.CAIN_IN
- 3.FC_IN
- 3.TIN_IN
- 3.TCIN_IN
- 3.CA_OUT
- 3.T_OUT

Punkty odpowiadające sterowaniom oraz wyjściom obiektu zostały następnie przypisane odpowiednio do bloków wejść i wyjść wewnątrz arkusza projektowego. Punkty zakłóceń zadane zostały poprzez użycie bloków AVALGEN. Dodatkowo sterowania podłączone zostały poprzez blok TRANSFER sterowany sygnałem binarnym. Blok ten określa czy w danej chwili regulator pracuje w trybie ręcznym czy też automatycznym.

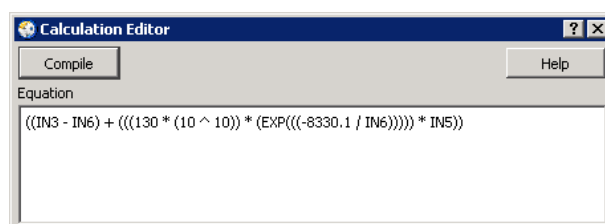
Po umieszczeniu wejść oraz wyjść modelu na arkuszu nadeszła pora na implementację samego obiektu. W tym celu wykorzystaliśmy bloki CALCBLOCK umożliwiające dokonywanie obliczeń z wykorzystaniem do 18 zmiennych wejściowych, z których to wejść nasz model wykorzystywał jedynie 6. Dla każdego użytego bloku, jako wejścia od IN1 do IN6 posłużyły nam kolejno: $C_{A_{in}}$, F_C , T_{in} , $T_{C_{in}}$, C_A oraz T . Implementacja pierwszego równania obyla się bez przeszkód. Jest ona przedstawiona na poniższym rysunku.

Niewielki problem wystąpił podczas implementacji równania drugiego wyjścia modelu. Ze względu na ograniczenie nałożone na ilość znaków mogących się znaleźć wewnątrz formuły bloku CALCBLOCK, działanie to musiało być przez nas podzielone na dwa bloki. W pierwszym z nich umieszczony został fragment: $T_{in_k} - T_k + 130 \cdot 10^{10} \cdot e^{-\frac{8330,1}{T_k}} \cdot C_{A_k}$. Pozostała część zmieściła się w drugim. W celu wyraźnego odgraniczenia wyjścia pierwszego bloku od zmiennych procesu, zostało one wprowadzone do drugiego bloku jako IN18. Obydwa bloki zostały także

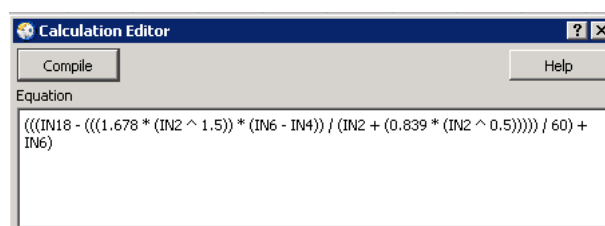


Rys. 2.1. Formuła pierwszego równania modelu wewnątrz CALCBLOCK

odpowiednio połączone, tak aby kalkulacje w bloku drugim zachodziły zawsze po kalkulacjach bloku pierwszego. Zawartość obydwu bloków przedstawiona została na rysunkach poniżej.

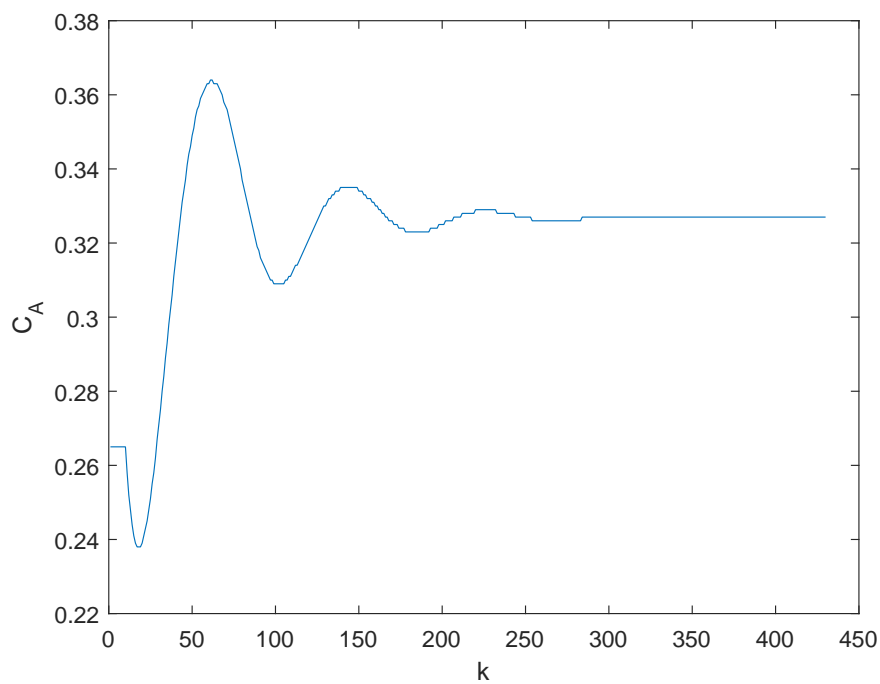


Rys. 2.2. Formuła wewnątrz pierwszego bloku CALCBLOCK dla drugiego równania modelu

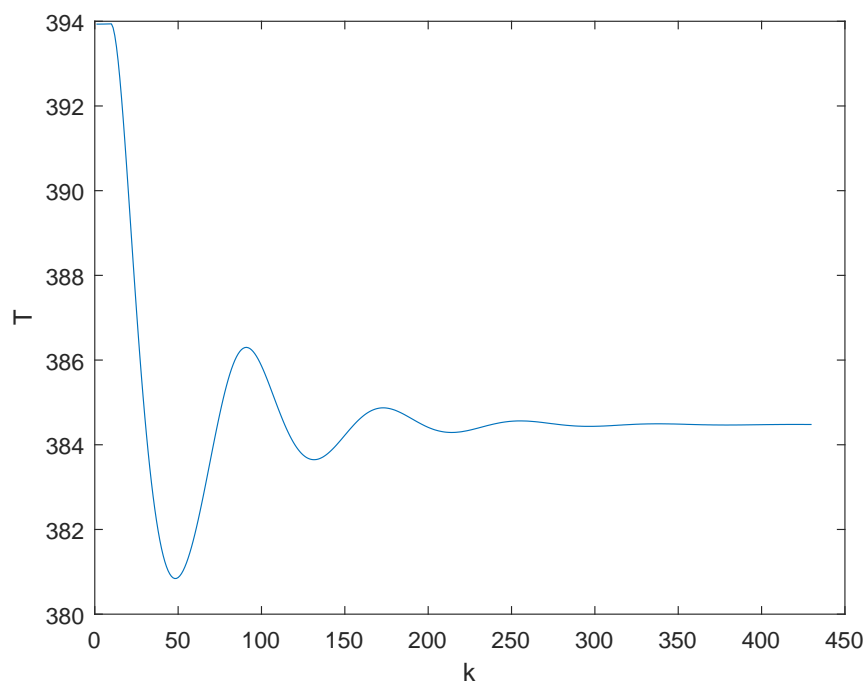


Rys. 2.3. Formuła wewnątrz drugiego bloku CALCBLOCK dla drugiego równania modelu

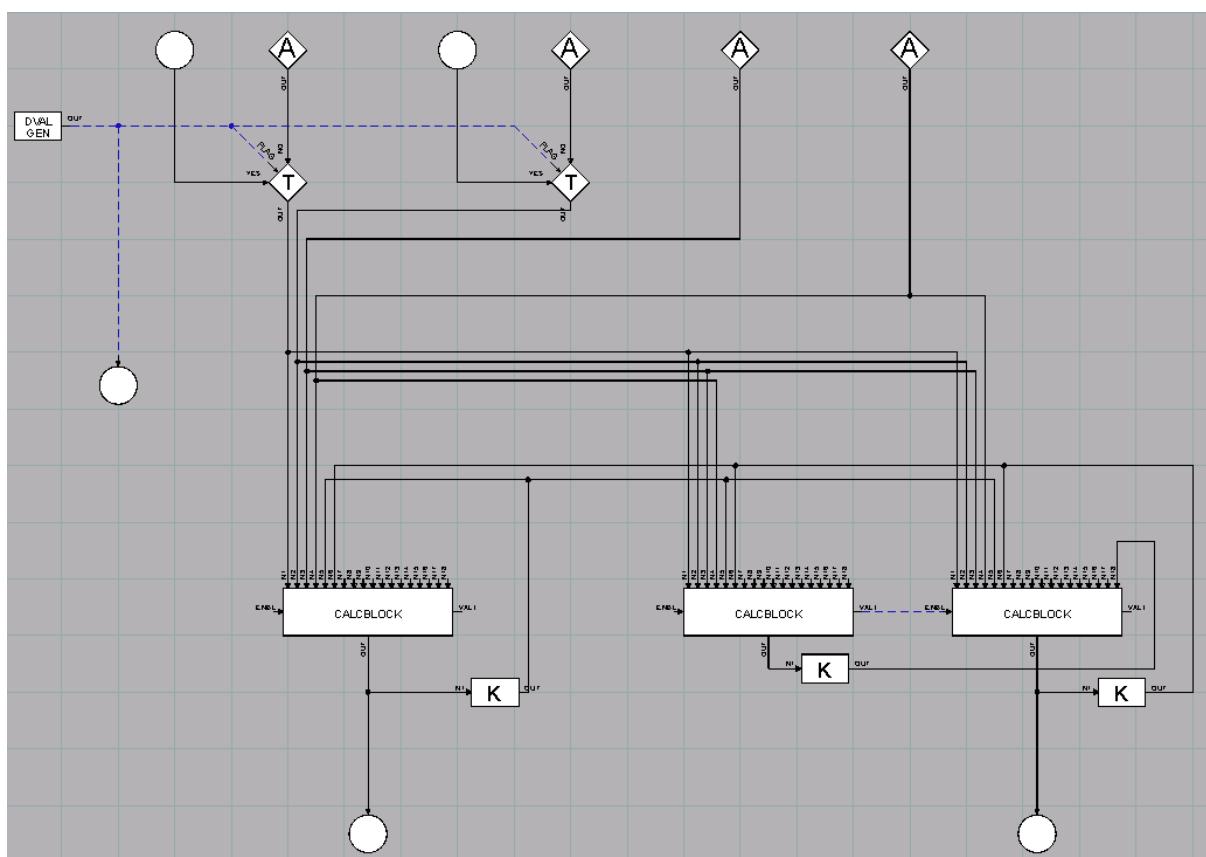
Po odpowiednim połączeniu wszystkich elementów otrzymaliśmy działający model zadanego dla nas obiektu. W celu jego przetestowania wykonaliśmy skok sterowania C_{Ain} do wartości 1,6. Obiekt zachował się zgodnie z oczekiwaniami, a wartości wyjścia zbiegły do oczekiwanych. Poniżej zamieszczamy wykresy otrzymane w wyniku tego testu.



Rys. 2.4. Wyjście C_A modelu po skoku sterowania C_{Ain} do 1,6



Rys. 2.5. Wyjście T modelu po skoku sterowania C_{Ain} do 1,6

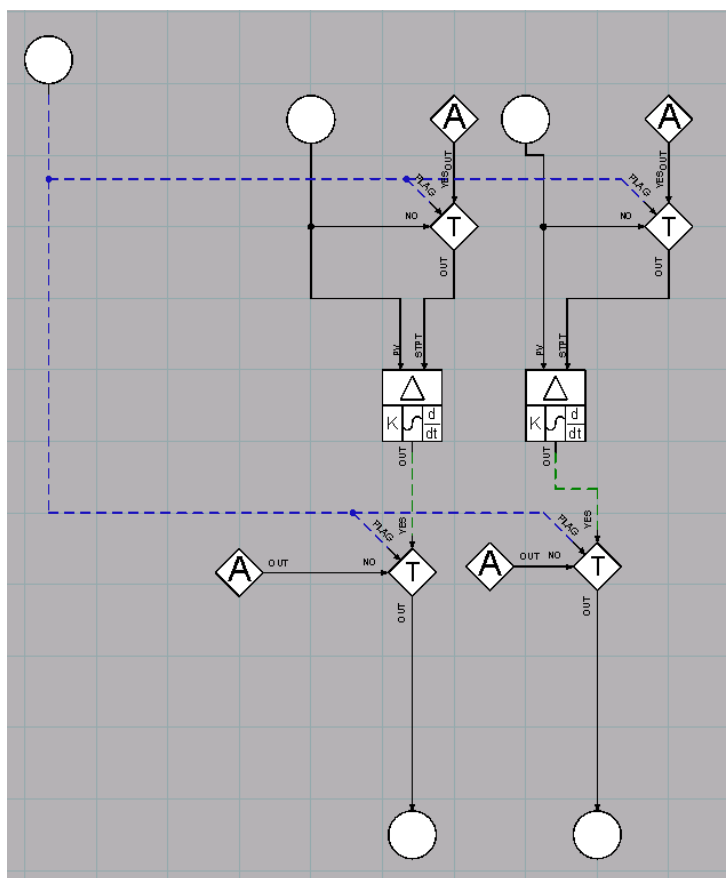


Rys. 2.6. Model obiektu wykonany w programie OVATION

3. Regulator obiektu

Drugim zadaniem wchodzącym w skład projektu była implementacja działającego regulatora PID. Miał być to regulator równoległy bez odsprzęgania, innymi słowy składał się on z dwóch regulatorów PID nie ingerujących we własne działanie. Każdy z nich otrzymywał na wejście uchyb jednego z wyjść i generował odpowiednie sterowanie. Wyjścia i wejścia zostały przez nas sparowane w następujący sposób: C_A z F_C oraz T z C_{Ain} . Decyzja ta została podyktowana charakterystyką działania obiektu określoną w pierwszej części projektu, według której zwiększenie sterowania C_{Ain} powodowało spadek wartości wyjścia C_A , a zwiększenie T . Reakcja dla drugiego wyjścia była odwrotna.

Mając określone wejścia i wyjścia regulatorów mogliśmy przystąpić do implementacji w programie OVATION. Zgodnie z instrukcją prowadzącego posłużyliśmy się w tym celu blokiem regulatora PID obecnym w systemie. Dodatkowo skorzystaliśmy z przełącznika binarnego używanego w arkuszu modelu obiektu służącego do przełączania trybów sterowania: manualnego i regulowanego. W trybie manualnym regulatory zostają odcięte, a podawany na nie uchyb jest zerowany.



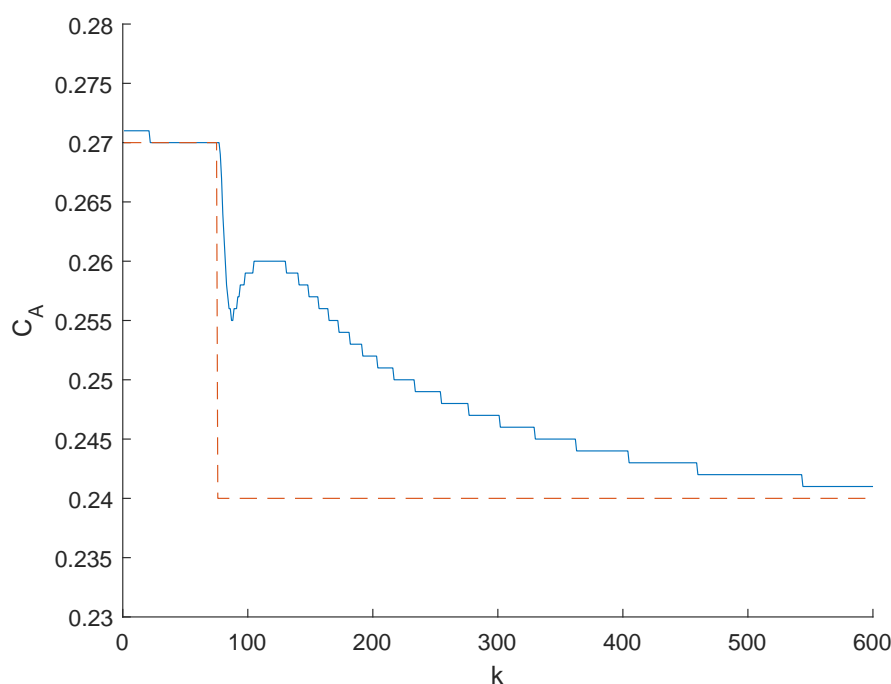
Rys. 3.1. Model regulatora wykonany w programie OVATION

Blok PID zawierał cztery parametry, które należało odpowiednio dobrać w celu osiągnięcia optymalnej regulacji. Ze względu na fakt, że strojenie regulatorów bezpośrednio na zaimplementowanym obiekcie mogłoby zająć więcej czasu niż mieliśmy, zdecydowaliśmy się dobrać nastawy z użyciem modelu wykonanego w programie Matlab i klasy regulatora, udostępnionej przez prowadzącego, bliźniaczo podobnej w działaniu do bloku obecnego w OVATION. Po uzyskaniu zadowalających wyników, nastawy regulatorów przeniesione zostały do symulatora OVATION. Ze względu na fakt, że Matlab oraz OVATION używają innych rozmiarów zmiennych liczbowych, działanie symulacji nie było całkowicie identyczne z danymi uzyskanymi w programie Matlab. Z tego powodu nastawy regulatorów zostały przez nas udoskonalone na podstawie wyników działania symulacji. Poniżej zamieszczamy tabelę z ostatecznymi parametrami obydwu regulatorów.

Zmienna regulowana	Wyjście	K	Ti	Kd	Td
C_A	F_C	10	0,55	80	10
T	C_{Ain}	0,002	600	0,2	1,8

Tab. 3.1. Nastawy regulatorów PID

Na koniec pozostało jeszcze zadanie udokumentowania działania zaimplementowanego przez nas układu regulacji. Poniżej przedstawiamy wyniki działania regulatorów w formie wykresów.



Rys. 3.2. Działanie regulatora PID regulującego wyjście C_A