## 1 Model budynku

### 1.1 Opis modelu

Model budynku opierał się na dwóch równaniach różniczkowych, które opisują kaloryfery wewnątrz (1) (rozumiane jako jeden, "uśredniony") oraz sam budynek wraz z zawartym wewnątrz niego powietrzem (2) (wartości parametrów były identyczne jak w dokumentacji):

$$m_h c_h \frac{dT_{PCO}}{dt} = F_{COB} \zeta c_w \left( T_{ZCO} - T_{PCO} \right) - k_h \left( T_{PCO} - T_r \right) \tag{1}$$

$$m_b c_b \frac{dT_r}{dt} = k_h \left( T_{PCO} - T_r \right) - k_{ext} \left( T_r - T_O \right) \tag{2}$$

Budynek posiadał swój własny regulator, którego zadaniem było utrzymanie stałej, ustalonej temperatury wewnątrz pomieszczeń. Wybrany został regulator PID w postaci dyskretnej opisany równaniem:

$$u(t) = K_P \cdot e(t) + K_I \cdot h \cdot \sum_{n=0}^{t} e(n) + K_D \cdot \frac{e(t) - e(t-1)}{h}$$
 (3)

gdzie  $K_P,\,K_I$  oraz  $K_D$  oznaczają wzmocnienia poszczególnych członów, zaś h jest przyrostem czasu.

Regulator sterował przeływem wody przez budynek, lecz przepływ ten nie mógł być nieograniczony i mógł się zawierać w przedziale od 0 do  $40 \left[ \frac{m^3}{h} \right]$ . Konieczne było zatem zastosowanie saturacji sygnału sterującego:

$$u_{sat}(t) = \begin{cases} 0, & \text{gdy } u(t) \leq 0 \\ u(t), & \text{gdy } u(t) \in (0; 1) \\ 1, & \text{gdy } u(t) \geqslant 1 \end{cases}$$
 (4)

Wartość  $u_{sat}$  rozumieć można jako poziom otwarcia zaworu sterującego przepływem wody, przez co równanie (1) zostało zmodyfikowane do postaci:

$$m_h c_h \frac{dT_{PCO}}{dt} = u_{sat} \cdot F_{COB} \zeta c_w \left( T_{ZCO} - T_{PCO} \right) - k_h \left( T_{PCO} - T_r \right)$$
 (5)

# 1.2 Opis implementacji

Modele przedstawione w poprzednim rozdziałe zostały wykorzystane do napisania odpowiednich klas w języku C++. Do rozwiązywania równań różniczkowych wykorzystano metodę Runge-Kutta.

Komunikacja w systemie wykorzystuje sockety TCP, dzięki czemu kod nie jest niepotrzebnie rozbudowany, a jednocześnie umożliwia dwustronną komunikację sieciową. Transport danych został zrealizowany z wykorzystaniem biblioteki *rapidjson*, natomiast poszczególne pola składowe tych JSONów były zgodne z ogólnie przyjętą dla całego projektu specyfikacją.

#### 1.3 Platforma docelowa

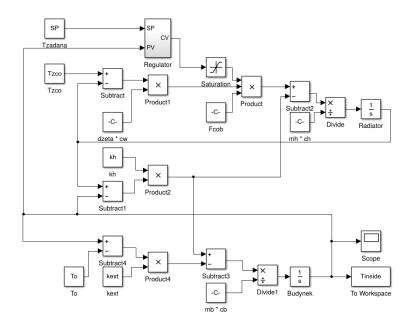
Jako platforma docelowa wybrana została płytka  $Arduino\ Yun$ , na pokładzie której dostępne są dwa mikrokontrolery. Pierwszy, z rdzeniem megaAVR, jest w przypadku tego projektu niewykorzystywany. Drugi, z rdzeniem MIPS, jest wykorzystywany przez system OpenWRT. System ten, oparty na jądrze Linux, występuje powszechnie w urządzeniach sieciowych takich jak routery, switche czy access pointy. Na tej płytce połączony jest z gniazdem Ethernet oraz modułem WiFi.

Dzięki dostępnymi w systemie OpenWRT podstawowymi narzędziami programistycznymi, takimi jak gcc czy też make, możliwe było tworzenie aplikacji docelowej na komputerze PC (w trakcie testów), aby następnie móc przenieść ją na platformę docelową.

Płytka  $Arduino\ Yún$  stworzona została z myślą o internecie rzeczy  $(Io\ T)$ , dzięki czemu jednym z możliwych scenariuszy rozwojowych tego projektu byłoby wykorzystanie tego układu do sterowania rzeczywistym procesem (a także gromadzeniem danych na jego temat), zamiast modelowania go, a następnie łączenie tych rozproszonych procesów w zorganizowany system.

### 1.4 Dobór nastaw regulatora

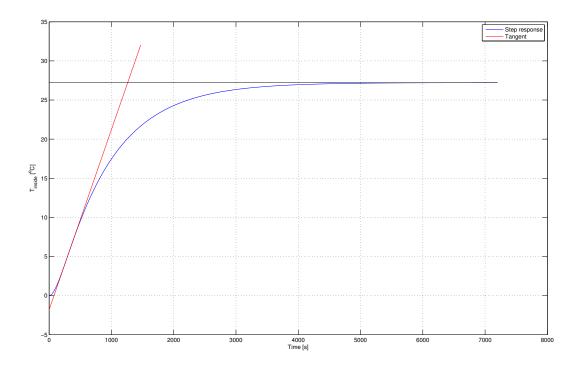
W celu doboru nastaw regulatora temperatury w budynku, stworzony został, zgodnie z równaniami przedstawionymi w rozdziale 1.1, model budynku w programie SIMULINK.



Rysunek 1: Model budynku stworzony w programie SIMULINK

Do doboru nastaw wykorzystano pierwszą metodę Zieglera-Nicholsa. W pierwszym kroku konieczne było otrzymanie odpowiedzi skokowej obiektu, która przedstawiona jest na rysunku 2. Odpowiedź ta uzyskana została przy założeniu stałej temperatury wody dopływającej do kaloryferów równej 70 [ ${}^{o}C$ ] oraz stałej temperatury otoczenia równej 0 [ ${}^{o}C$ ]. Następnie wyzna-

czono punkt, w którym pochodna odpowiedzi skokowej jest największa i wyrysowano styczną do odpowiedzi w tym punkcie. Kolejnym etapem było wyznaczenie parametrów obiektu – stałej czasowej T oraz opóźnienia  $\tau$ .



Rysunek 2: Odpowiedź skokowa obiektu

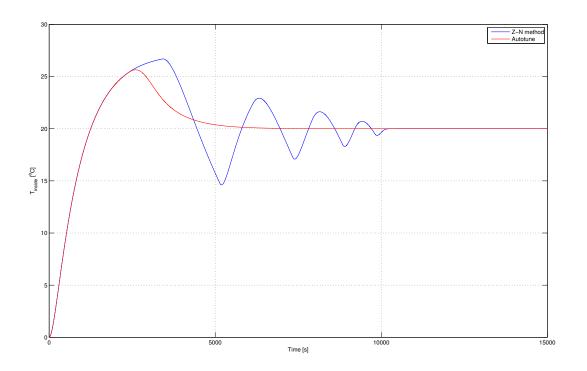
Na podstawie wyznaczonych parametrów, zgodnie z tabelą dla pierwszej metody Z-N, dobrane zostały nastawy regulatora, które wynosiły:

- $K_P = 5.2$ ,
- $K_I = 0.028$ ,
- $K_D = 203$ .

Podjęta została również próba doboru nastaw przy wykorzystaniu opcji *Tune* w programie MATLAB. Nastawy otrzymane w tym przypadku (po zaokrągleniu) wyniosły:

- $K_P = 0.5$ ,
- $K_I = 0.0005$ ,
- $K_D = 50$ .

Przebiegi temperatury wewnątrz budynku dla obu zestawów przedstawione są na rysunku 3. Widać wyraźnie, że pierwszy zestaw nastaw powoduje trwające długi czas oscylacje, co jest spowodowane ewidentnie zbyt duża wartością wzmocnienia części całkującej, która powoduje, w połączeniu z blokiem saturacji, że powstaje tzw. "windup". Funkcja *Tune* uwzględnia saturację (pracuje na modelu, który posiada ten blok), tak więc pozwala dobrać nastawy lepiej pasujące do konkretnego obiektu.



Rysunek 3: Przebiegi temperatury dla różnych nastaw regulatora

Przedstawione powyżej nastawy otrzymane zostały przy pewnych ograniczeniach związanych z wartościami parametrów, które w rzeczywistym systemie ulegają częstym zmianom, a mianowicie wartość temperatury wody zasilającej oraz temperatury otoczenia. Mimo wszystko stanowią one dobry punkt wyjściowy do doboru nastaw regulatora w całym systemie.