

Projekt

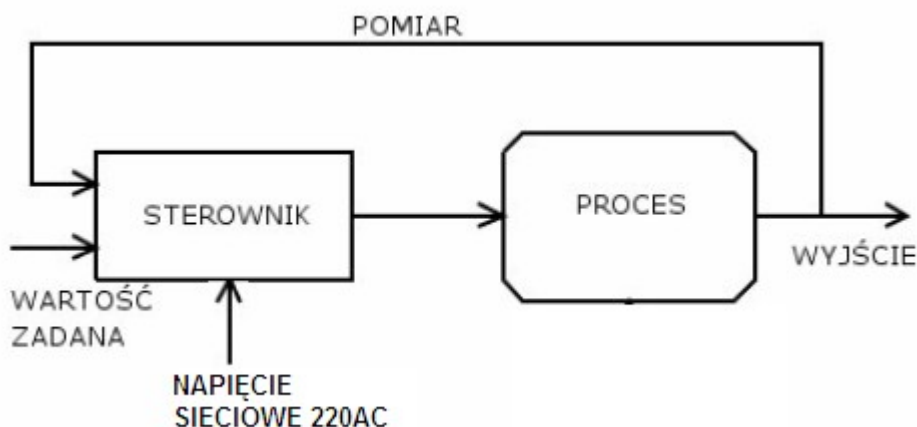
Regulator temperatury PID

Przemysław Ciołek 131138

Tomek Więckowski 137357

1. Cel i założenia projektowe

Wykonanie projektu regulatora PID (proporcjonalno-całkująco-różniczkującego) miało na celu uświadomienie nam w jaki sposób działają, oraz jak i dlaczego są projektowane w taki a nie inny sposób. Jako obiektu użyliśmy czajnika elektrycznego (o mocy nie większej niż 2kW), zaś sam algorytm regulatora realizowany jest na procesorze Atmega8 firmy Atmel, dzięki czemu koszty projektu zostały na akceptowalnym dla nas poziomie. Regulator nie wykorzystuje jakże popularnych współcześnie algorytmów samostrojenia. Naszym celem było poznanie i zastosowanie metod strojenia takich jak metoda Zieglera-Nicholsa dla zadanych konkretnie temperatur, także aby skupić się na jakości regulacji musieliśmy zrezygnować z regulacji w pełnym zakresie, powiedzmy 0-90°C.



Powyższy schemat ilustruje ideę sterownika. W naszym przypadku procesem jest grzanie wody w obiekcie, którym jest czajnik, zaś wyjściem jest po prostu oddawanie energii elektrycznej na energię cieplną, zarówno wody jak i otoczenia.

2. Regulacja PID

Na samym początku jednak trzeba wyjaśnić czym jest regulator PID. Otóż jest on złożeniem kilku innych regulatorów: P, I, PI, PD wraz ich zaletami oraz wadami! Regulacja w naszym przypadku dokonuje się poprzez pętlę sprzężenia zwrotnego, którą jest odczyt temperatury. Wartością wejściową moc regulowana wypełnieniem impulsu PWM. Omówmy zatem wady i zalety:

– regulator proporcjonalny P

Uchyb $e(t)$ jest to różnica między wartością zadaną, a wartością aktualnie otrzymywaną na wyjściu regulowanego obiektu. Zależność między sygnałem wyjściowym, a uchybem regulacji regulatora proporcjonalnego opisana jest zależnością:

$$u(t) = K_p \cdot e(t)$$

gdzie K_p jest wzmocnieniem członu proporcjonalnego. Zaletą jest prostota, wadą to, że nigdy tak naprawdę nie zlikwidujemy uchybu, zawsze zostaje pewien minimalny uchyb w stanie ustalonym

– regulacja całkująca I i proporcjonalno-całkująca PI

W regulatorze całkującym mamy współczynnik K_i który wyznacza proporcje między przyrostem wartości wyjściowej a uchybem sygnału.

$$\frac{du(t)}{dt} = K_i \cdot e(t)$$

Zaletą: ze względu na fakt iż regulator ten może mieć niezerową wartość sygnału wyjściowego przy zerowej wartości uchybu – można zlikwidować uchyb w stanie ustalonym. Najczęściej jednak stosuje się go razem z członem proporcjonalnym (i/lub różniczkowym)

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(t) dt$$

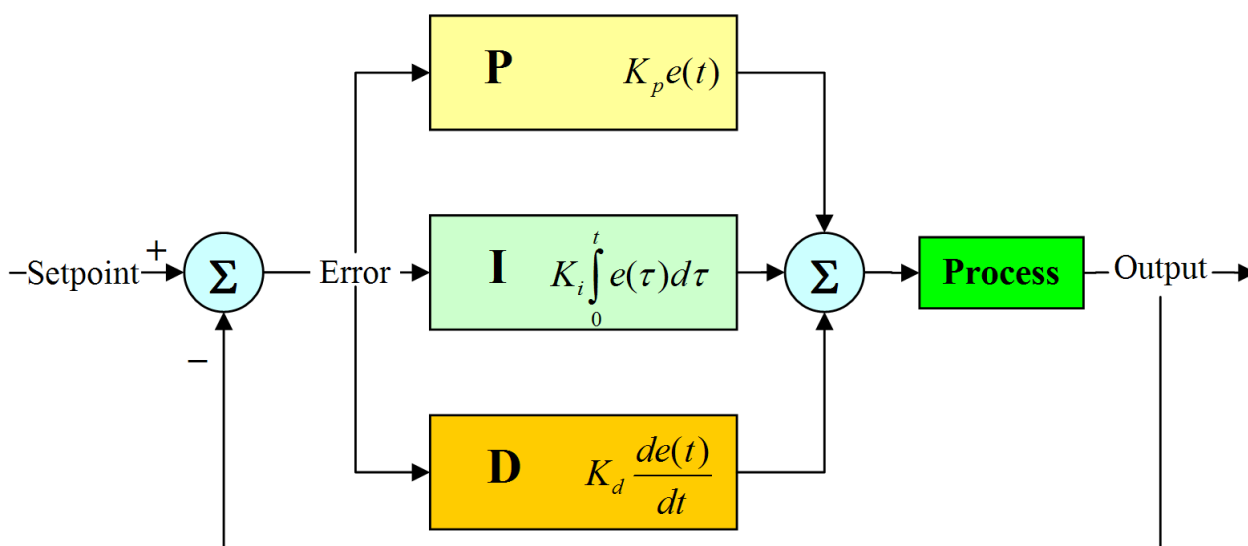
Wadą regulatora PI jest mała dynamika (mała szybkość działania lub inaczej długi czas regulacji).

- regulator proporcjonalno-różniczkujący PD

$$u(t) = K_p \cdot e(t) + K_d \frac{de(t)}{dt}$$

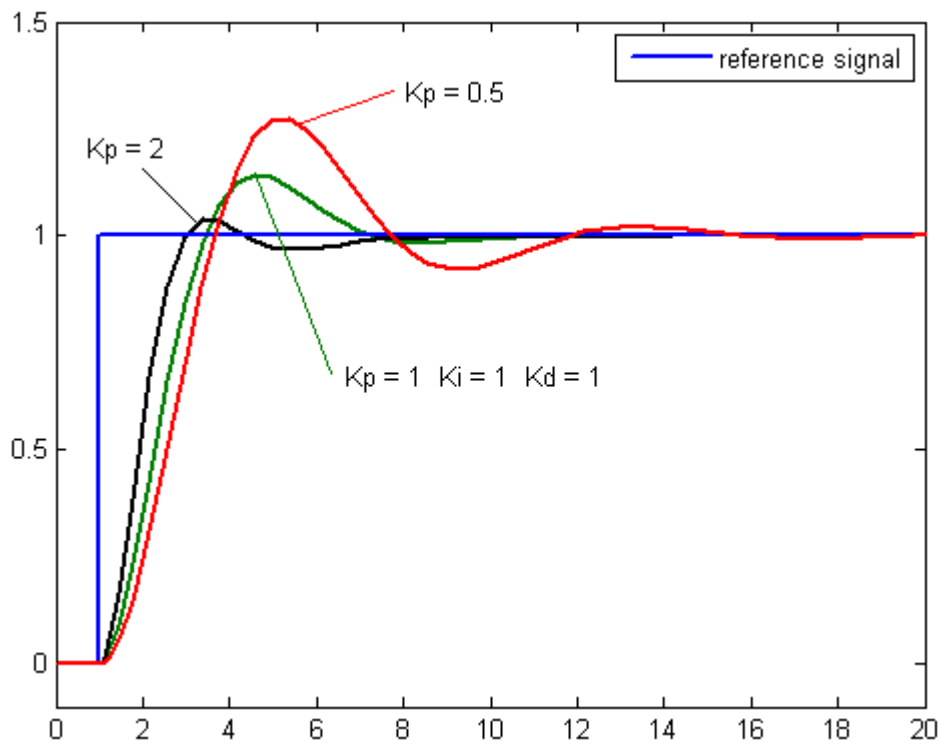
Regulator PD składa się z dwóch członów: proporcjonalnego i różniczkującego. Dodanie członu różniczkującego do sterowania proporcjonalnego powoduje wzrost wrażliwości regulatora. Jego istotną zaletą jest reakcja na przyrost zmian uchybu, na podstawie której wyznacza korektę sterowania. W ten sposób zabezpiecza przed powstaniem zbyt dużej amplitudy sygnału wykonawczego i przyczynia się do wzrostu stabilności układu. Wadą tego typu regulacji, w porównaniu do sterowania z całkowaniem, jest brak likwidacji uchybu w stanie ustalonym. Istotną cechą członu różniczkującego jest brak reakcji na uchyb regulacji a jedynie na jego zmianę, co powoduje, że człon ten nigdy nie może występować samodzielnie.

- regulator proporcjonalno-całkująco-różniczkujący PID

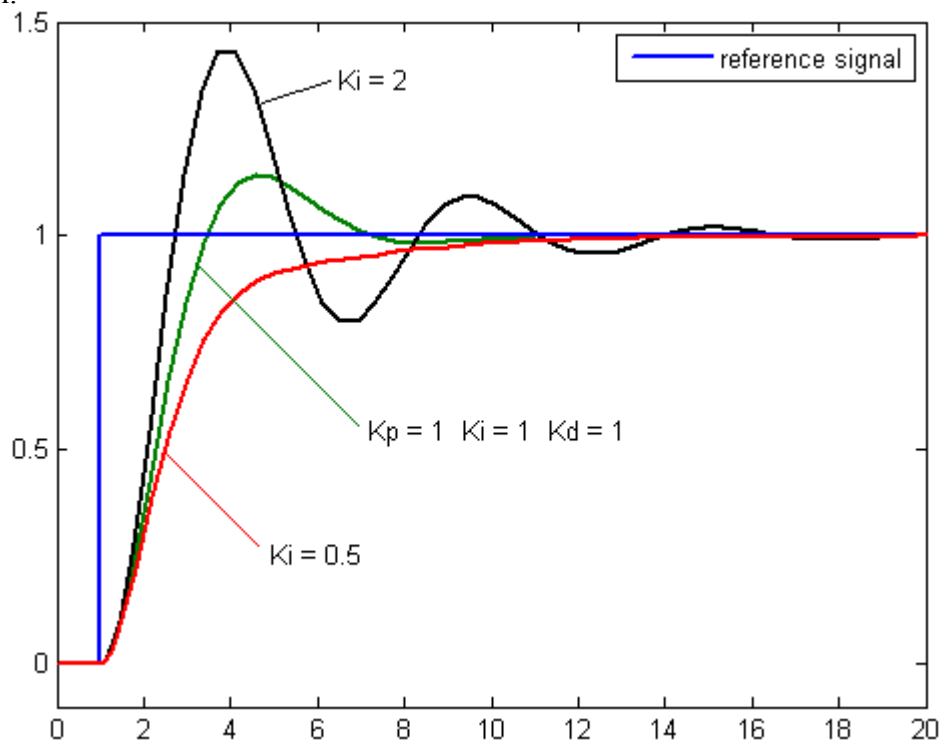


Algorytm PID łączy wszystkie trzy aspekty działania uwzględniając stan aktualny(P), historię (I) i obecną tendencję zmian (D). Jednocześnie posiada zalety i wady wszystkich części składowych. Wzmocnienie członu proporcjonalnego ma wpływ na czas narastania i wartość uchybu regulacji w stanie ustalonym. Człon całkujący likwiduje uchyb regulacji kosztem pogorszenia odpowiedzi w stanie przejściowym. Natomiast różniczkowanie zwiększa stabilność i przeregulowanie a także wpływa na poprawę odpowiedzi przejściowej.

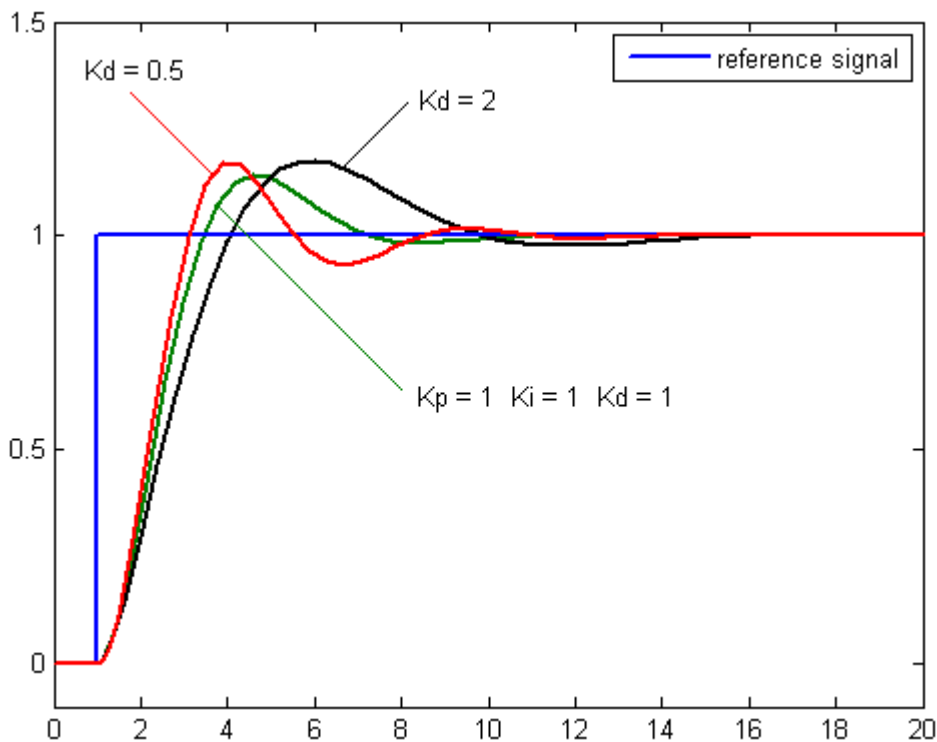
Połączenie tych wszystkich członów daje sumę ich zalet i wad co można zobaczyć na wykresie wyjściowym. Przy okazji zaobserwujemy jaki wpływ na charakterystykę regulacji mają poszczególne człony:



K_p – czyli człon proporcjonalny. W tym przypadku im większy tym lepiej – małe oscylacje, krótki czas regulacji.



K_i – człon całkujący, im większy tym szybciej osiągniemy zadaną wartość, lecz jak widać przypisanie zbyt dużej wartości temu współczynnikowi nie jest bezpieczne.



Kd – człon różniczkujący odpowiada za łagodzenie „pofalowania” początkowego kształtu wykresu. Zaś dobieranie tych współczynników najlepiej omówić wraz z samym algorytmem...

3. Omówienie algorytmu

Regulatory PID są powszechnie stosowane zarówno w przemyśle jak i urządzeniach z którymi zwykły człowiek styka się na co dzień jak np. sterownik pieca centralnego ogrzewania. Tak jak napisaliśmy w poprzednim akapicie – wyjście sterownika jest obliczane za pomocą wzoru łączącego trzy elementy P-I-D, każdy z nich jest w naszym programie realizowany za pomocą odpowiednich parametrów. Spójrzmy najpierw na kod algorytmu, dostarczonego przez firmę Atmel, który zaimplementowaliśmy w naszym sterowniku. W komentarzach wyjaśniliśmy jak działa.

//Inicjalizacja algorytmu

```
void pid_Init(int16_t p_factor, int16_t i_factor, int16_t d_factor, struct PID_DATA *pid)
{
    // Wartości startowe
    pid->sumError = 0;
    pid->lastProcessValue = 0;
    // Stałe współczynniki dla pętli regulacji
    pid->P_Factor = p_factor;
    pid->I_Factor = i_factor;
    pid->D_Factor = d_factor;
    // Maksymalne wartości (zabezpieczenie przed przeładowaniem pamięci)
    pid->maxError = MAX_INT / (pid->P_Factor + 1);
    pid->maxSumError = MAX_I_TERM / (pid->I_Factor + 1);
}
```

/*Algorytm PID

```
* Zwraca wartość liczbową, która jest wykorzystywana jako procentowa wartość wypełnienia PWM
* setPoint – zadana wartość (temperatura)
* processValue - mierzona wartość (temperatura)
* \param pid_st - aktualny stan regulatora
*/
```

```
int16_t pid_Controller(int16_t setPoint, int16_t processValue, struct PID_DATA *pid_st)
```

```

{
int16_t error, p_term, d_term;
int32_t i_term, ret, temp;

error = setPoint - processValue; //obliczanie uchybu e(t)

// obliczanie Pterm (proporcji) i zabezpieczenie przed przeładowaniem,
if (error > pid_st->maxError){
    p_term = MAX_INT;
}
else if (error < -pid_st->maxError){
    p_term = -MAX_INT;
}
else{
    p_term = pid_st->P_Factor * error;
}

// obliczanie Iterm(całkowanie) i granic całkowania, całkowanie metodą kwadratów
temp = pid_st->sumError + error;
if(temp > pid_st->maxSumError){
    i_term = MAX_I_TERM;
    pid_st->sumError = pid_st->maxSumError;
}
else if(temp < -pid_st->maxSumError){
    i_term = -MAX_I_TERM;
    pid_st->sumError = -pid_st->maxSumError;
}
else{
    pid_st->sumError = temp;
    i_term = pid_st->I_Factor * pid_st->sumError;
}

// obliczanie Dterm (różniczkowanie)
d_term = pid_st->D_Factor * (pid_st->lastProcessValue - processValue);

pid_st->lastProcessValue = processValue;

//ret – wartość sygnału sterującego, procentowe wypełnienie impulsu generowanego dalej przez układ z
triakiem. P_term – wartość członu proporcjonalnego, I_term – całkującego, D_term - różniczkującego
ret = (p_term + i_term + d_term) / SCALING_FACTOR;
if(ret > MAX_INT){
    ret = MAX_INT;
}
else if(ret < -MAX_INT){
    ret = -MAX_INT;
}

return((int16_t)ret);
}

```

Tak jak wspomnieliśmy na wstępie: chcemy dobrać parametry aby jak najbardziej optymalnie osiągnąć zadaną temperaturę. Optymalnie znaczy w jak najkrótszym czasie i z jak najmniejszymi oscylacjami.

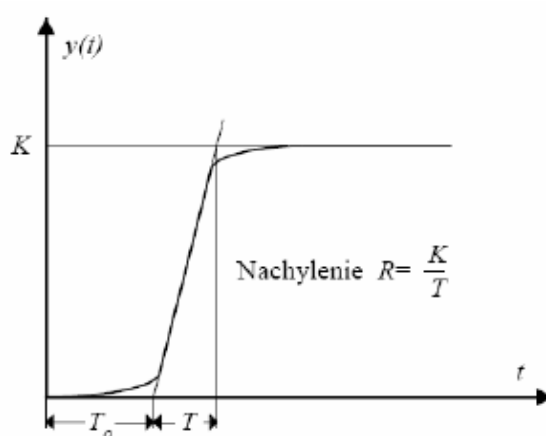
4. Dobór parametrów K_p , K_i , K_d , czyli strojenie regulatora

Po przejrzaniu pewnej ilości literatury na temat doboru współczynników regulacji zdecydowaliśmy się wybrać najbardziej intuicyjną metodę Zeiglera-Nicholsa, która bazuje na interpretacji charakterystyki odpowiedzi na zadany sygnał jednostkowy. Posiłkując się pracą magisterską kolegi Piotra Pomorskiego próbowaliśmy zastosować ją do potrzeb i możliwości naszego projektu, gdzie obiektem sterowanym jest czajnik elektryczny.

Otóż większość obiektów jest typu statycznego, tak jak nasz czajnik. Panowie Zeigler i Nichols spostrzegli, że na podstawie odpowiedzi tego sygnału można dobrać potrzebne nam parametry.

$$G(s) = \frac{K \cdot e^{-sT_0}}{1 + T \cdot s}$$

Transmitancja układu statycznego aproksymuje układy wyższego rzędu prostym równaniem rzędu pierwszego z dodatkową zwłoką T_0 .



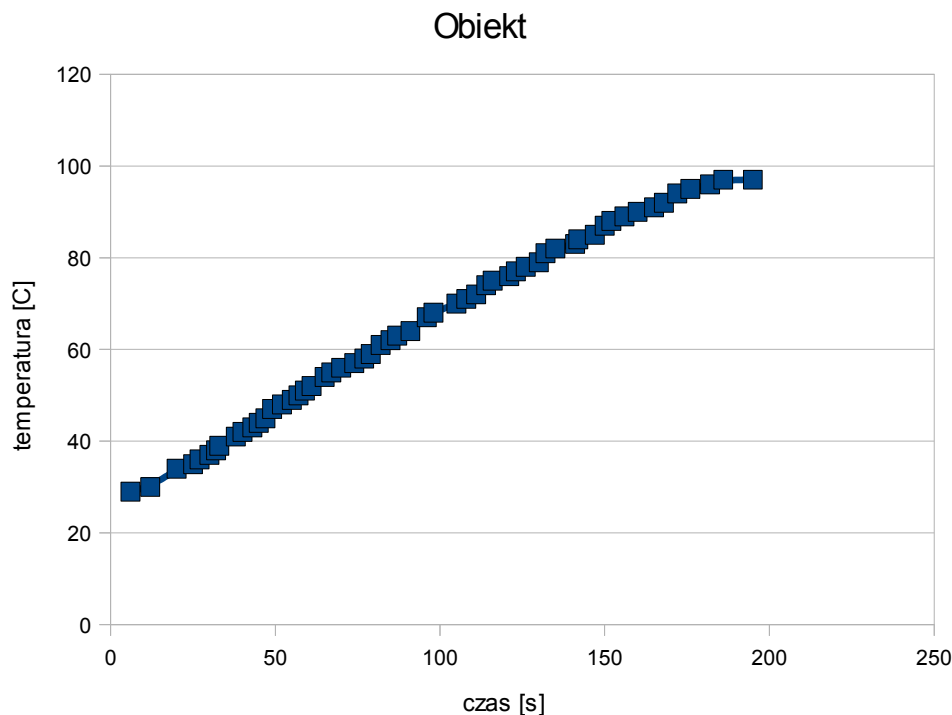
(a)

Dane z powyższej charakterystyki wykorzystujemy wraz z tabelką poniżej:

Typ regulatora	K_p	T_i	T_d
P	$1/a$	---	---
PI	$0,9/a$	$3T_0$	---
PID	$1,2/a$	$2T_0$	$0,5T_0$
$a = RT_0$			

Wybór parametrów w metodzie Zeiglera-Nicholsa opiera się na współczynniku zanikania równym 0,25. Oznacza to, że składowa przejściowa zaniknie do 0,25 początkowej wartości po jednym okresie oscylacji.

Jak zrobiliśmy to my? Po pierwsze włączyliśmy czajnik, a na wyświetlaczu LCD odliczaliśmy sekundy i temperaturę. Wyniki zanotowane w arkuszu kalkulacyjnym pozwoliły na sporządzenie dość precyzyjnej charakterystyki.



Jak widzimy grzałka jest elementem rezystancyjnym, więc charakterystyka jest liniowa. Opóźnienie dość spore $T_o = 8$ sekund. $K = T_{max} - T_{min} = 70$ °C. Dalej odczytujemy czas narastania odpowiedzi $T = 174$ s. Nachylenie $R = K/T = 0,4$.

Korzystając z danych w tabelce obliczamy:

$$K_p = 0,37$$

$$T_i = 16$$

$$T_d = 4$$

Tutaj przeliczamy T_i i T_d na współczynniki K_i i K_d :

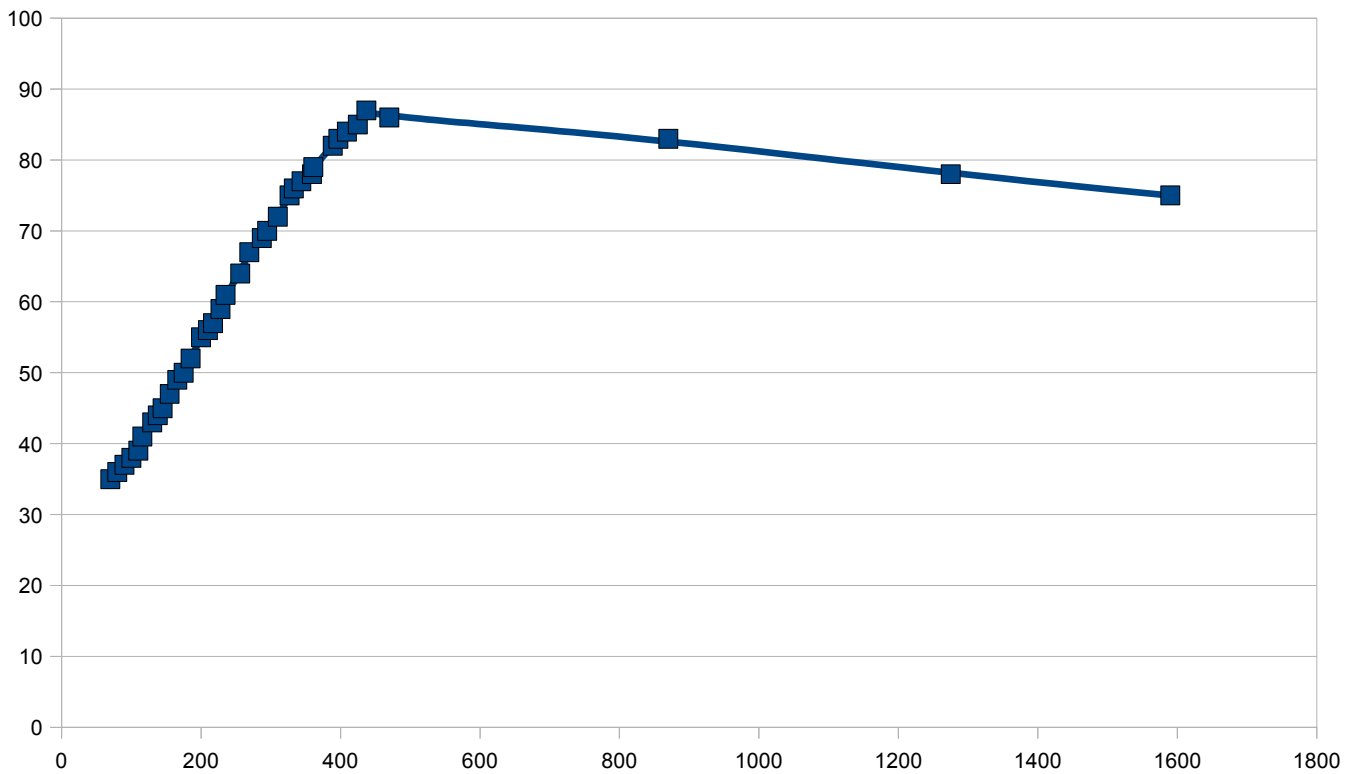
$$K_i = K_p * T / T_i$$

$$K_d = K_p * T_d / T$$

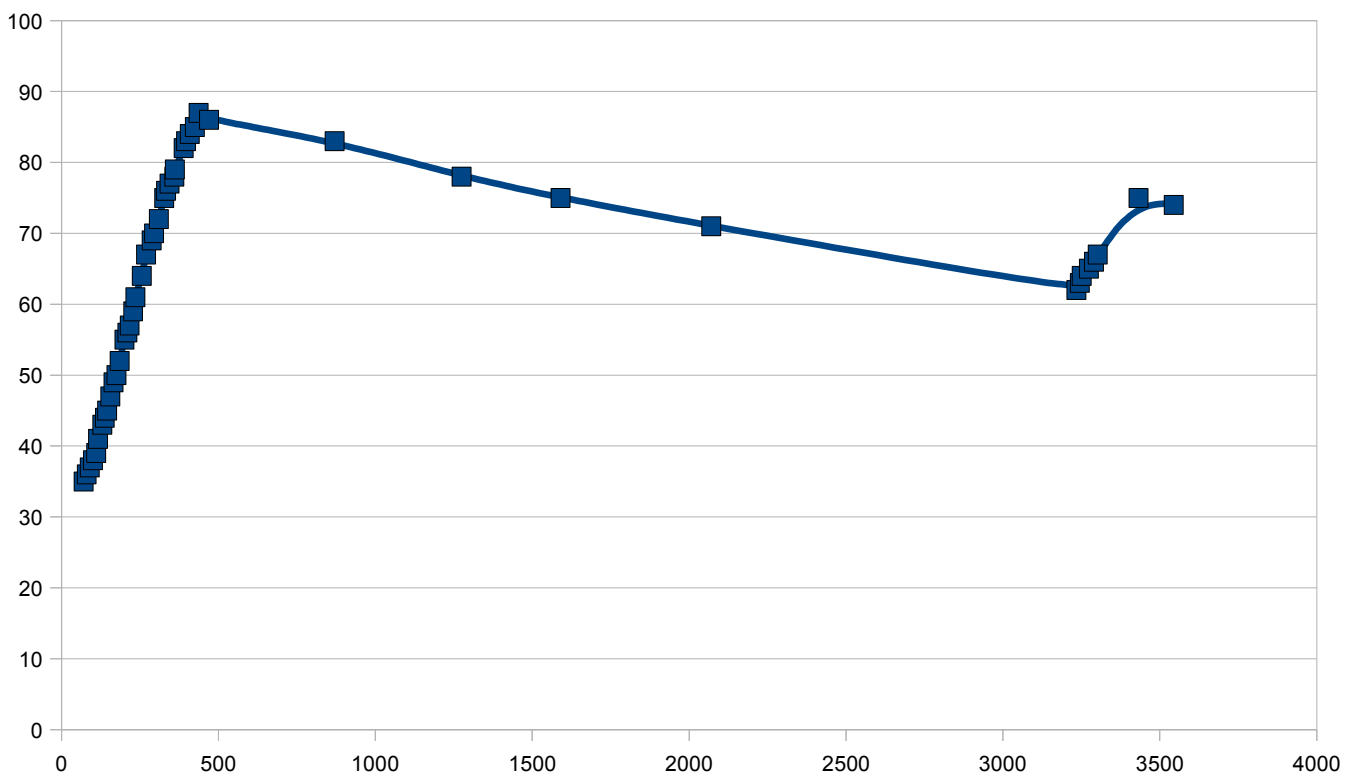
gdzie T jest czasem próbkowania. Jako, że nasz program wykonuje się jako jedna pętla z częstotliwością ok 1Hz zatem przyjmujemy $T=1$

5. Wynik działania regulatora

Regulator nasz reguluje temperaturę 1 litra wody w czajniku o mocy nie większej niż 2kW (nie dysponujemy sprzętem pomiarowym, ani notą katalogową, ale przypuszczamy, że jest to standardowa grzałka). Temperaturą zadaną jest $T = 65$ °C. Na podstawie odczytów sporządziliśmy wykres Temp(Czas)



Na powyższym wykresie widzimy czoło wraz z charakterystycznym uchybem regulatora PID wynoszącym około 30% sygnału zadanego.



Powyżej widzimy zafalowanie i kolejny uchyb. Ciekawy jest fakt, że chłodzenie odbywa się w sposób liniowy.

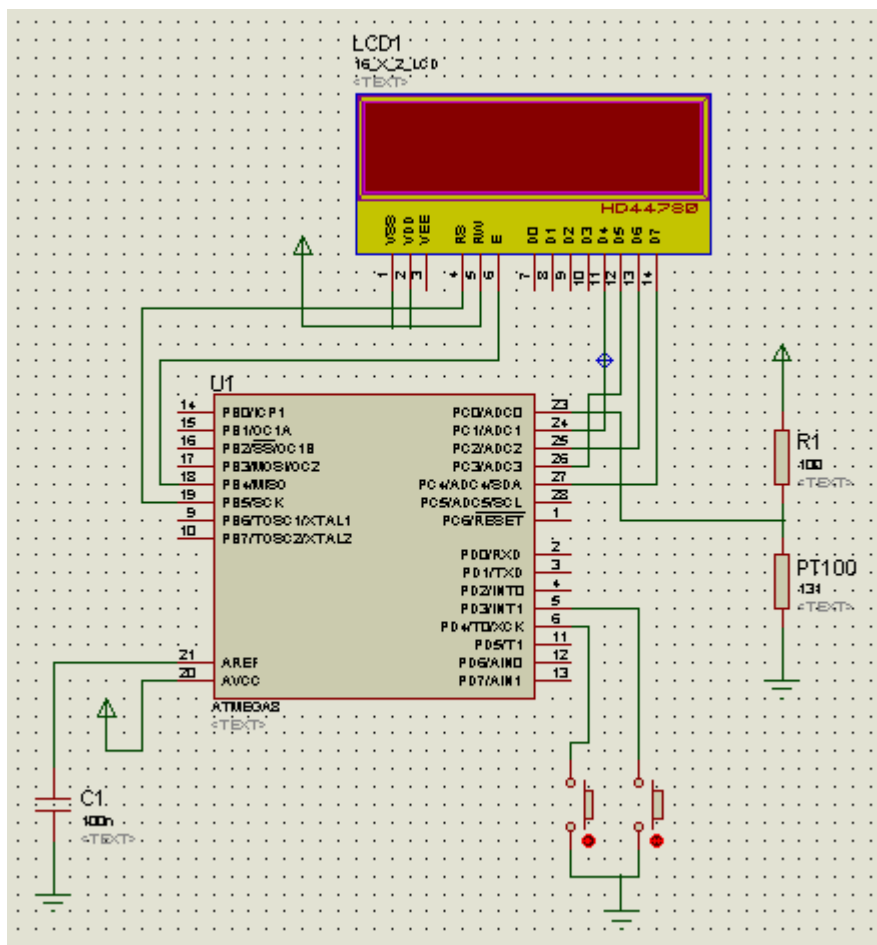
Wnioski

Dobór naszych paramterów metodą Zeiglera-Nicholsa był kompromisem między sensownym czasem podgrzewania i jak najmniejszym uchybem. Uchyb można zminimalizować do kilku stopni, lecz wtedy osiągnięcie zadanej temperatury „za pierwszym razem” będzie trwać dość długo. Problemem jest jednak obliczenie modelu bezwładnościowego grzałki. Opierając się na literaturze zawartej w pracy magisterskiej kolegi Piotra Pomorskiego na temat sterowników PLC nie znaleźliśmy sposobów doboru innych niż empiryczne, które pasowałyby do naszego modelu. Obserwując zaś wynik naszego projektu można pokusić się o myśl, że problemem jest to, że zadane współczynniki nie są całkiem kompatybilne z bezwładnością grzałki.

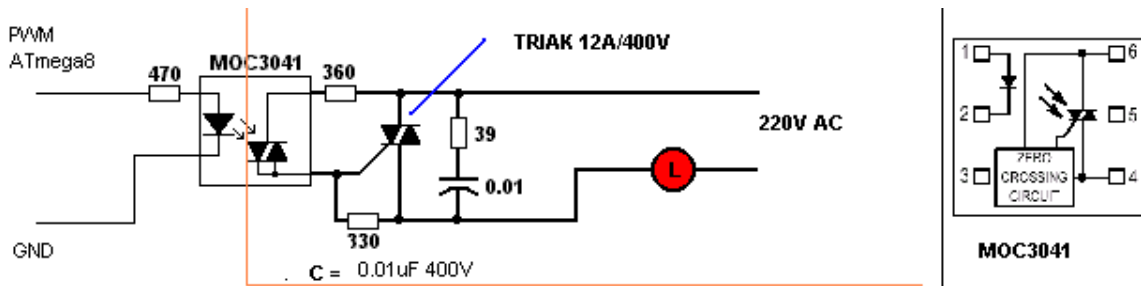
Podsumowując projekt pozwolił nam zapoznać się z kilkoma rodzajami regulatorów, co było niezbędne do zrozumienia regulatora PID, sposobami ich strojenia i problemami mogącymi nękać gotowe aplikacje. Pomimo iż rozumiemy zasadność docelowego doboru parametrów PID dla zadanego obiektu, obserwując współczesny rynek bardzo dużą popularnością cieszą się układy z algorytmami autotuningu, które są łatwiejsze w aplikacji i wykluczają pomyłkę projektanta.

6. Schemat układu regulatora

a) mikroprocesor i peryferia



b) układ sterowania z optotriakiem MOC3041



Jak widać na powyższych rysunkach nasz projekt składał się z procesora Atmega8, który za pomocą przetwornika ADC (analogowo-cyfrowego) zczytuje napięcie na termistorze PT100 i na podstawie charakterystyki dostarczonej przez producenta odczytuje temperaturę. Termistor ten jako pierwszy wpadł nam w ręcę i bardzo łatwo można było wykonać z niego czujnik temperatury wody. Jednak trzeba przyznać, że w następnym projekcie wykorzystamy układ LM35, gdyż w przeciwieństwie do termistora ma charakterystykę liniową. Termistor zaś pozwala uzyskać dużą dokładność ale tylko w pewnym zakresie temperatur, chyba że użyje się kompensujących, ale wydaje nam się, że w ramach tego projektu mimo wszystko wystarczy nam ten oto przyrząd.

Wyświetlacz LCD



Tz – temperatura zadana

Ta – temperatura aktualna

czas – czas podany w sekundach