

**Wydział Elektroniki i Technik Informacyjnych  
Politechnika Warszawska**

**Projektowanie układów sterowania  
(projekt grupowy)**

**Sprawozdanie z laboratorium nr 4**

**Paweł Bugyi, Marcin Michalski, Krzysztof Pierczyk**

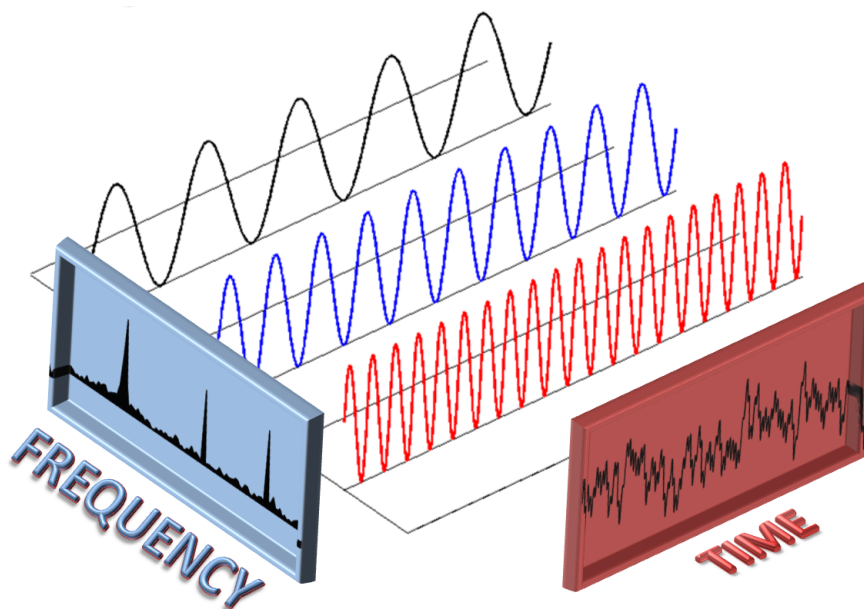
**Warszawa, 2020**

# Spis treści

|   |    |
|---|----|
| <b>1. Wstęp</b>                               | 2  |
| <b>2. Stanowisko grzejąco-chłodzące</b>       | 3  |
| 2.1. Określanie punktu pracy                  | 3  |
| 2.2. Przeciwdziałanie awarii czujników        | 4  |
| 2.3. Zbieranie odpowiedzi skokowych           | 5  |
| 2.4. Implementacja regulatora DMC             | 6  |
| 2.5. Strojenie parametrów regulatora          | 8  |
| 2.6. Panel HMI i kontrola trajektorii zadanej | 10 |
| <b>3. Stanowisko Inteco</b>                   | 13 |
| 3.1. Charakterystyka statyczna                | 13 |
| 3.2. Własna implementacja PID                 | 15 |
| 3.3. Wbudowany regulator PID                  | 19 |
| 3.4. Trajektorie zadane                       | 19 |
| 3.5. Badanie odporności wybranego regulatora  | 20 |
| 3.6. Wizualizacja procesu                     | 22 |
| <b>4. Podsumowanie</b>                        | 24 |

# 1. Wstęp

Teoria sterowania oraz szeroko pojęte modelowanie układów dynamicznych są bardzo ciekawymi zagadnieniami z punktu widzenia matematyki. Wiele narzędzi wykorzystywanych na tym polu, jak choćby transformacja Laplace'a, pokazuje właściwości naszego świata, które potrafią być nieoczywiste, a niejednokrotnie nawet niemożliwe do wychwycenia za pomocą ludzkiej percepcji. Dziedziny te są jednak nie tylko obszernym polem teoretycznym, ale stanowią również podwaliny pod nieustający proces automatyzacji.



Rys. 1.1: Transformacja Fouriera - jedno z najważniejszych narzędzi współczesnej automatyki

Do tej pory, w ramach zajęć zajmowaliśmy się głównie implementacją i badaniem algorytmów sterowania opartych o modele programowe. Jednak to, co funkcjonuje w idealnym, matematycznym świecie nie zawsze przekłada się na rzeczywistość. Modele stanowią jedynie przybliżenie realnych obiektów. Ze względu na ograniczoną moc obliczeniową lub braki teoretyczne nie jesteśmy w stanie przedstawić wielu z nich z wystarczającą dokładnością. Ponadto świat pełny jest zjawisk losowych, które zakłócają przebiegi wartości fizycznych i prowadzą do jeszcze większych odstępstw od tworzonych modeli. Aby skonfrontować zdobytą w trakcie semestru wiedzę teoretyczną z realiami współczesnej automatyki zajęliśmy się implementacją wielowymiarowych regulatorów z wykorzystaniem przemysłowego sterownika PLC (ang. *programmable logic controller*). Urządzenia tego typu, pomimo swoich wad, są wciąż powszechnie stosowane we współczesnym przemyśle co sprawia, że stanowią one dobre narzędzie do zdobywania praktycznych umiejętności.

## 2. Stanowisko grzejąco-chłodzące

W ramach laboratorium otrzymaliśmy do dyspozycji dwa obiekty regulacji<sup>1</sup>. Pierwszym z nich było, dobrze znane wszystkim studentom Automatyki i Robotyki z naszego wydziału, stanowisko grzejąco-chłodzące (G-C). Jego unikalne własności w postaci choćby stabilnych wartości parametrów statycznych czy niemal idealnie wyważonych relacji między torami wejścia i wyjścia sprawiają, że stanowi ono doskonałe narzędzie do badań laboratoryjnych, szczególnie prowadzonych przez studentów. Jedną z najważniejszych cech stanowiska jest jego okres pomiaru równy jednej sekundzie. Nie da się zaprzeczyć, że praca z modelami matematycznymi ma tę wadę, że wszystkie symulacje przeprowadzane są w ułamkach sekund. Sprawia to, że osoba zajmująca się badaniem nie ma często czasu zastanowić się nad ich głębszymi aspektami. Nie ma czasu przystanąć i pomyśleć, co skutkuje niejednokrotnie wieloma niedociągnięciami lub po prostu pominięciem ciekawszych zagadnień.

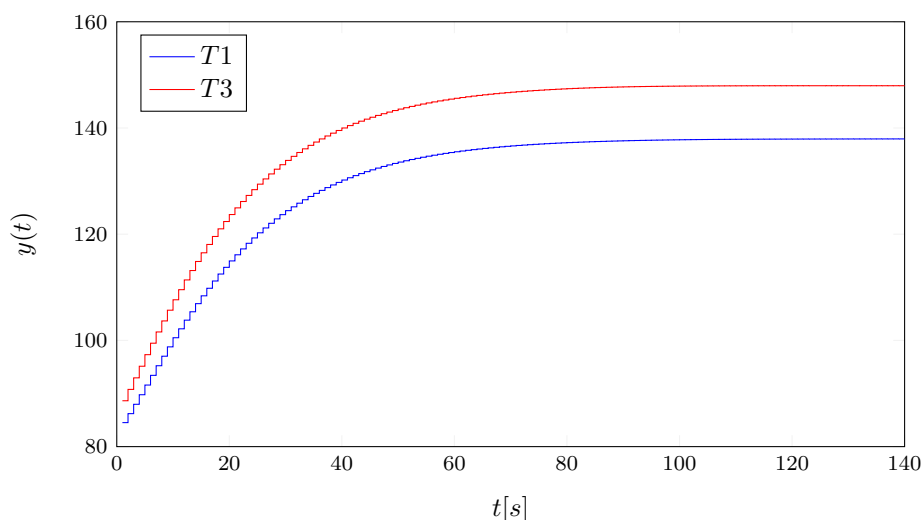
Na szczęście sytuacja taka nie ma miejsca w przypadku stanowiska G-C. Wielogodzinne zbieranie danych z obiektu celem pozyskania odpowiedniego modelu, długie sesje dostrajania regulatorów oraz następujące po nich kilkudniowe maratony akwizycji danych porównujących działanie kilku wariantów to tylko niektóre z przyjemności, których dostarcza ono studentom naszego wydziału. Taki stan rzeczy ma oczywiście szereg pozytywnych konsekwencji żeby wymienić chociaż wspomniany czas na refleksje. Refleksje nie tylko na temat algorytmów regulacji, ale również na temat samego siebie, na temat miejsca i czasu, w którym się znaleźliśmy. Na temat potoku życiowych możliwości, które pojawiają się przed nami z każdą osiągniętą przez stanowisko wartością zadaną. Okresy obserwacji tworzących się na naszych oczach przebiegów zmiennych procesowych to także czas na relaks i oczyszczenie umysłu, których próżno szukać wśród innych projektów laboratoryjnych. Nocne koszmary z udziałem niedających się nastroić regulatorów PID oraz wadliwych czujników temperatury stanowią jedynie symboliczną opłatę za tak obfity zbiór niekwestionowanych przyjemności. Z nieprzejechanym zapalem i niepohamowaną euforią budzącą się w nas na myśl o możliwości pracy z tak kunsztownym obiektem zabraliśmy się więc do pracy...

### 2.1. Określanie punktu pracy

W przypadku modeli matematycznych eksperymentalne określanie punktów pracy mogło wydawać się bezcelowe, zbędne. Mogły być one przecież wyliczone w przeciągu kilku chwil z użyciem równań opisujących model. Zadanie to przybiera jednak na wadze, gdy do czynienia mamy z rzeczywistym procesem. Eksperyment bywa wówczas jedyną sensowną metodą pozyskania informacji o obiekcie. Wyznaczenie dokładnych punktów pracy może być jednym z kluczowych punktów w procesie projektowania regulatora, ponieważ stanowią one będą o jego ostatecznej postaci. Mając na uwadze taki stan rzeczy przeprowadziliśmy kilka eksperymentów mających na celu ustalenie wartości temperatur wyjściowych przy podanym w zadaniu sterowaniu  $G1 = 34$ ,  $G2 = 39$ .

---

<sup>1</sup> Niestety ze względu na zaistniałą pandemię ostatecznie musieliśmy ograniczyć się jedynie do badań opartych o modele tychże obiektów



Rys. 2.1: Przebieg temperatur wyjściowych po ustaleniu wartości sterowań w punkcie pracy

Na rys. 2.1 przedstawione zostały przebiegi temperatur T1 i T3 po zaaplikowaniu w.w. wartości sterowania. Jak widać, zbiegają one do stabilnych wartości równych w przybliżeniu  $T1 = \dots$ ,  $T3 = \dots$

## 2.2. Przeciwdziałanie awarii czujników

Rozważania teoretyczne, z przyczyn czysto praktycznych, zakładają zazwyczaj stabilną, niezakłóconą pracę sensorów oraz elementów wykonawczych. Taki stan rzeczy nie odzwierciedla jednak sytuacji rzeczywistych. Każdy układ fizyczny narażony jest na czynniki zewnętrzne, które mogą spowodować jego całkowitą awarię lub przynajmniej częściowe upośledzenie funkcjonalności. Szczególnie niebezpieczne mogą być awarie torów pomiarowych. Jeżeli algorytm sterujący (lub nawet operator) nie jest świadomy rzeczywistej wartości regulowanej zmiennej, to może on doprowadzić ją poza bezpieczny zakres pracy. To z kolei skutkuje często dalszą ekspansją uszkodzeń lub nawet zagrożeniem dla życia ludzkiego.

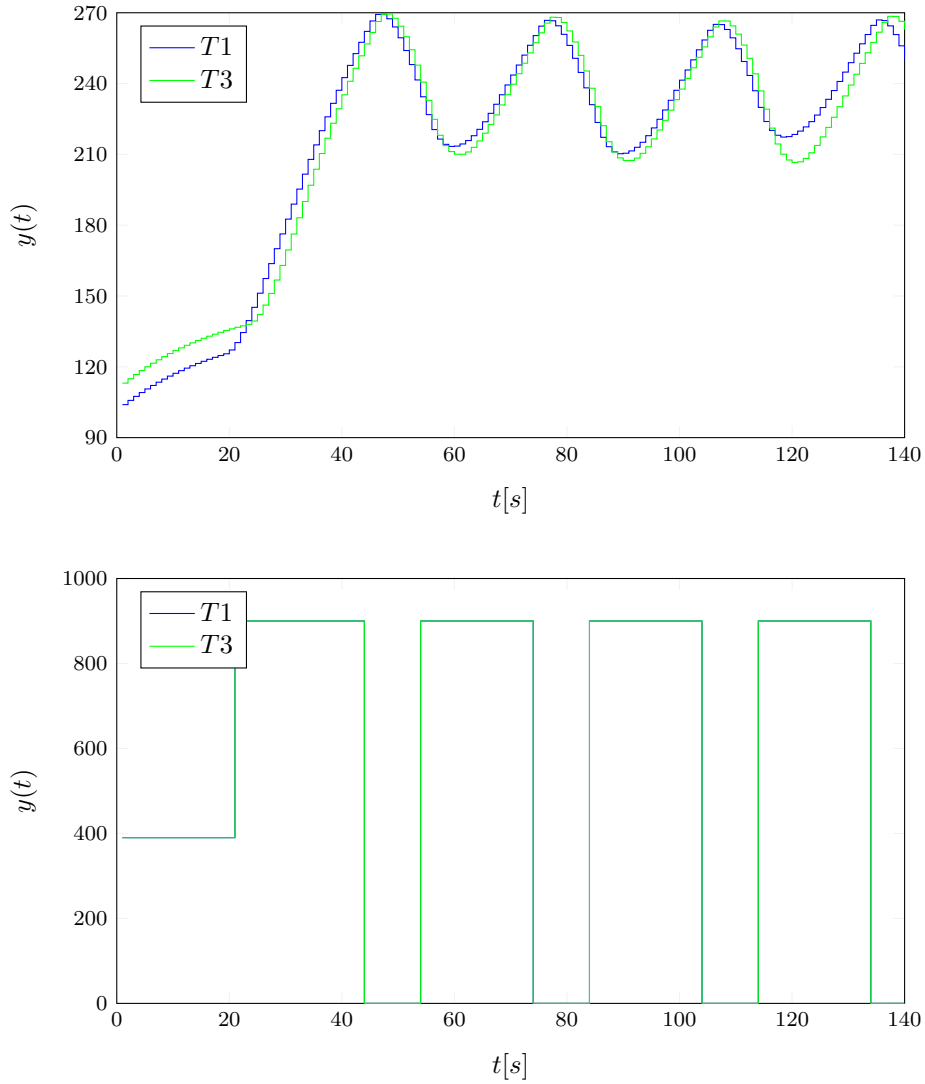
Jedną z najprostszych metod przeciwdziałania takim awariom jest zgrubne oszacowanie dopuszczalnych wartości mierzonych. Gdy pomiar opuści przyjęty zakres, wówczas czujnik uznawany jest za uszkodzony i jego odczyty nie są dane pod uwagę przy dalszym wyznaczniku sterowań. Podjęta może być wówczas procedura mająca na celu utrzymanie bezpiecznych wartości wyjści torów wyjściowych w bezpiecznym zakresie (nawet kosztem pogorszenia jakości regulacji) do czasu naprawy awarii. Właśnie taki mechanizm został przez nas zaimplementowany w przypadku stanowiska G-C.

Gdy pomiar wartości wyjściowej przekracza 250°C korespondująca z nią grzałka zostaje wyłączona. Sytuacja taka została przedstawiona na rys. 2.2. Mechanizm ten realizowany jest poprzez proste wyrażenie warunkowe widoczne na poniższym listingu.

```

IF HC_T1 > HC_UPPER_BOUND_PV THEN
    HC_G1 := HC_LOWER_BOUND_PV;
END_IF;
IF HC_T3 > HC_UPPER_BOUND_PV THEN
    HC_G2 := HC_LOWER_BOUND_PV;
END_IF;

```



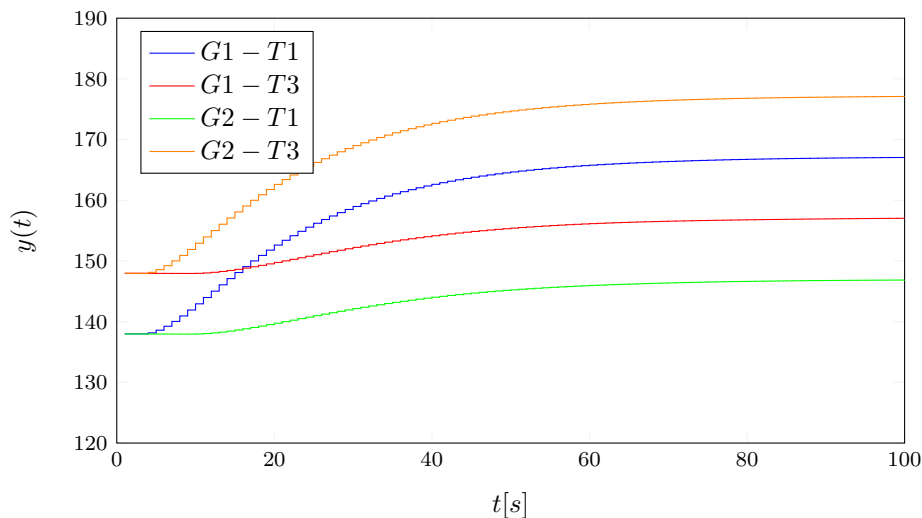
Rys. 2.2: Przebieg sygnałów sterujących  $G1$ ,  $G2$  po przekroczeniu maksymalnych dopuszczalnych odczytów z czujników  $T1$ ,  $T3$

### 2.3. Zbieranie odpowiedzi skokowych

W ramach projektu laboratoryjnego 3.5 przygotowaliśmy podstawowy regulator DMC w wersji analitycznej o wymiarze  $2 \times 2$ . Miał on za zadanie regulować pracę modelu stanowiska grzejąco-chłodzącego przygotowanego w środowisku **Matlab**. Kolejne laboratorium wymogło na nas przeniesienie tej implementacji na sterownik PLC. Podstawy teoretyczne opisujące działanie regulatora zostały opisane szerzej w poprzednim sprawozdaniu. W tym miejscu przypomnimy jedynie kilka podstawowych wzorów. Wartość sterowania obliczana jest w procesie optymalizacji uchybu na pewnym horyzoncie predykcji. Jeżeli znany jest model obiektu sterowania w postaci odpowiedzi skokowych poszczególnych torów wejścia-wyjścia, to można skonstruować pewne macierze  $M$  oraz  $M^P$ . Dla macierzy parametrów regulatora  $\Psi$ ,  $\Lambda$  prawo regulacji przybiera wówczas postać 2.1.

$$\begin{cases} \Delta U(k) = K[Y^{zad}(k) - Y(k) - M^P \Delta U^P(k)] \\ K = (M^T \Psi M + \Lambda)^{-1} M^T \Psi \end{cases} \quad (2.1)$$

Projektowanie rozpoczęliśmy od zebrania czterech odpowiedzi skokowych, po jednej dla każdego toru. W tym celu napisany został prosty program wykonujący skoki wartości sterujących. Po ustabilizowaniu układu w punkcie pracy następował skok o (...) na grzałce  $G1$ , a następnie zbierane były odpowiedzi skokowe na obu wyjściach. Po ustabilizowaniu temperatur  $T1$  i  $T3$ , rozumianym jako spadek wariancji pomiarów w pewnym oknie czasowym poniżej wartości zadanej, wartość sterowania wracała do punktu pracy. Następnie wykonywany był skok na drugiej grzałce i ponownie zbierane były odpowiedzi skokowe. Zbieranie przebiegów odbyło się za pomocą aplikacji *GX LogViewer*, która umożliwia monitorowanie i zapisywanie do plików wartości wskazanych komórek pamięci sterownika w regularnych odstępach czasu. Zgodnie ze specyfikacją okres próbkowania wynosił 1s. Zebrane odpowiedzi zostały zaprezentowane na 2.3,



Rys. 2.3: Nieznormalizowane odpowiedzi skokowe zebrane z obiektu

## 2.4. Implementacja regulatora DMC

Pliki tekstowe zawierające surowe dane z obiektu zostały przez nas zaimportowane do środowiska *Matlab*. Po uporządkowaniu do postaci wektorów o jednolitej długości poddaliśmy je obróbce z wykorzystaniem klasy **DMC**, którą zaimplementowaliśmy w ramach projektu trzeciego. Klasa ta dokonuje automatycznej normalizacji odpowiedzi oraz oblicza macierze obecne w prawie regulacji na bazie zadanych parametrów regulatora. Generacja macierzy odbywała się oczywiście kilkakrotnie w czasie dobierania nastaw algorytmu. Kolejny prosty skrypt umożliwił nam zapisanie obliczonych wartości do plików w postaci, w której mogły być one bezpośrednio przekopiowane do programu pisanego w środowisku *GX Works*. Zapisywana była oczywiście jedynie część danych z obliczonych macierzy, ponieważ **algorytm DMC został zaimplementowany w wersji "oszczędnej"**

Między implementacją DMC w *Matlabie*, a implementacją na rzeczywistym sterowniku PLC jest kilka zasadniczych różnic. Po pierwsze, należy wziąć pod uwagę zasoby sprzętowe sterownika. Gdy tworzony regulator wchodził będzie w interakcję jedynie z modelem obiektu, to zazwyczaj czas wykonania obliczeń nie gra roli. Model obiektu funkcjonuje pod dyktando programu nadrzędnego, a czas upływający w symulacji jest jedynie pozorny <sup>1</sup>. W rzeczywistości obiekt

<sup>1</sup> Oczywiście możliwe jest stworzenie bardziej realistycznych modeli, o czym przekonaaliśmy się w trakcie laboratorium 3.5

regulacji funkcjonuje niezależnie od regulatora. Utrzymanie stałego okresu próbkowania oraz aplikacji sterowania jest fundamentem poprawnej regulacji, dlatego projektując rzeczywisty regulator należy bezwzględnie wziąć pod uwagę dostępną moc obliczeniową i na jej podstawie oszacować maksymalny czas obliczeń w pojedynczej iteracji. Szacowanie tego typu na podstawie samej dokumentacji sterownika może być bardzo uciążliwe, ponieważ czasy wykonywania poszczególnych instrukcji nie muszą być jednakowe. Z pomocą przychodzi nam środowisko *GX Works*. Umożliwia ono przeprowadzenie symulacji napisanego programu i obserwację parametrów jego wykonywania, w tym parametrów czasowych. Kryterium czasu nie było w naszym przypadku krytyczne. Rząd dynamiki stanowiska G-C został ustalony na 139. Konsekwencją tego jest bardzo krótki czas wykonywania programu niezależnie od przyjętych horyzontów regulacji i sterowania. Wahał się on od  $81\mu s$  do  $7200\mu s$  (przy inicjalizacji sterownika) na iterację. Średnia uplasowała się na poziomie  $200\mu s$ . Czasy te są dalece niższe niż okres próbkowania obiektu.

Kolejnym zagadnieniem jest zajętość pamięci. Gdy obiekt stabilizuje się długo wydłuża się również rząd dynamiki. To z kolei pociąga za sobą zwiększenie wymiarów macierzy i wektorów, na których operuje DMC. Jako, że obiekty te, są zazwyczaj przechowywane bezpośrednio w pamięci sterownika należy uwzględnić je w procesie projektowania. Wymiar macierzy  $M^P$  to  $(N \cdot n_y) \times ((D - 1) \cdot n_u)$ , a macierzy  $K$   $(N_u \cdot n_u) \times (N \cdot n_y)$ . Jak widać, złożoność pamięciowa jest liniowo zależna od horyzontu predykcji  $N$ . Jeżeli korzystamy z "oszczędnej" wersji DMC, to przechowujemy jedynie pierwsze  $n_u$  wierszy macierzy  $K$ , zatem złożoność pamięciowa **nie zależy od horyzontu sterowania**  $N_u$ . Przepiszmy prawo regulacji 2.1 w wersji "oszczędnej".

$$\Delta U(k) = \overline{K_1}[Y^{zad}(k) - Y(k) - M^P \Delta U^P(k)] \quad (2.2)$$

Wektory  $Y^{zad}(k)$  oraz  $Y(k)$  są długości  $(N \cdot n_y)$ . Możemy jednak zauważyć, że ilość unikalnych elementów w każdym z nich wynosi jedynie  $n_y$ . Następnie zauważmy, że macierze  $K$  i  $M^P$  nie zmieniają się w trakcie pracy sterownika. Pogrupujmy teraz wyrażenia obecne w równaniu 2.2.

$$\Delta U(k) = \overline{K_1}[Y^{zad}(k) - Y(k) - M^P \Delta U^P(k)] \quad (2.3)$$

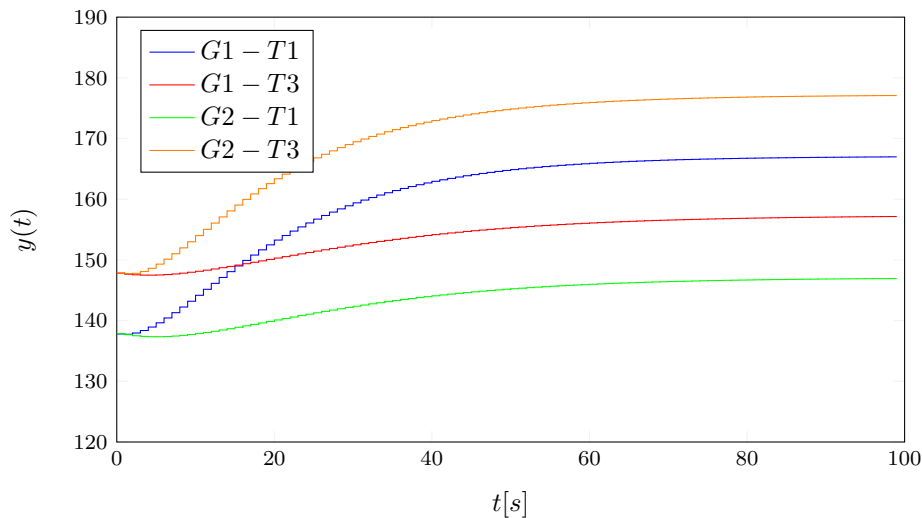
Wykorzystując pierwsze z powyższych spostrzeżeń możemy dojść do wniosku, że reprezentację macierzy  $\overline{K_1}$  dla pierwszego składnika można efektywnie ograniczyć do rozmiaru  $n_y \cdot n_y$  sumując odpowiednie jej elementy. Ponadto, zamiast przechowywać oddzielnie macierze  $\overline{K_1}$  i  $M^P$  możemy a priori obliczyć ich iloczyn otrzymując macierz wymiaru  $n_u \times ((D - 1) \cdot n_u)$ . W ten sposób oszczędzamy znaczną ilość pamięci. Ponadto niwelujemy problem przechowywania dwuwymiarowych tablic w pamięci sterownika. Właśnie takie podejście zastosowaliśmy w projekcie.

Ostatnią z widocznych różnic, chociaż nie wpływającą na pracę samego sterownika, jest tryb pracy w dwóch odrębnych środowiskach w trakcie projektowania układu regulacji. W przypadku *Matlaba* obliczenia off-line i on-line były wykonywane w tym samym środowisku. Oprogramowanie *GX Works* nie posiada modułu umożliwiającego wykonywanie tego typu operacji, co wymusiło repetetywne przenoszenie danych między środowiskami. Warto podkreślić ten fakt szczególnie, gdy możliwe jest wykonanie matematycznego modelu obiektu regulacji i przeprowadzenie wstępnego strojenia algorytmu w środowisku symulowanym. Może to zaoszczędzić znaczną ilość czasu projektanta.



## 2.5. Strojenie parametrów regulatora

Dobór parametrów regulatora przebiegał dwuetapowo. Kierując się wyżej poczynionymi spostrzeżeniami postanowiliśmy rozpocząć od **przybliżenia odpowiedzi skokowych modelami matematycznymi** w postaci inercji drugiego rzędu. W tym celu wykorzystaliśmy mechanizm zaimplementowany w ramach laboratorium 3.5. Stworzyliśmy funkcję celu, która równa była błędowi średniokwadratowemu popełnianemu przez model na zadanej trajektorii sygnałów wejściowych<sup>2</sup>. Funkcja ta została następnie zminimalizowana przez algorytm genetyczny dla wszystkich czterech modeli<sup>3</sup>. Wyjścia modeli takich samych przebiegów wejść zostały przedstawione na rys. 2.4



Rys. 2.4: Odpowiedzi skokowe stworzonych modeli

Po dokonaniu aproksymacji przystąpiliśmy do strojenia wirtualnego regulatora. Ponownie zostały wykorzystane mechanizmy z laboratorium 3.5. Modele z postaci wektora parametrów zostały przekształcone do postaci funkcji, które po podaniu zadanej trajektorii wejściowej zwracają przebieg wyjścia. W takiej formie zostały one przekazane do konstruktora zaimplementowanej przez nas wcześniej klasy DMC<sup>4</sup>. Konstruktor ten dokonuje zebrania i normalizacji odpowiedzi skokowej, a następnie obliczenia optymalnych macierzy DMC dla zadanych parametrów. Udobnie też wygodny interfejs, który automatycznie zarządza wektorami zapisanymi w pamięci regulatora, co umożliwia zwięzłą implementację pętli sterowania. Dzięki temu udało nam się w stosunkowo krótkim czasie odnaleźć parametry zapewniające zadowalającą jakość sterowania.

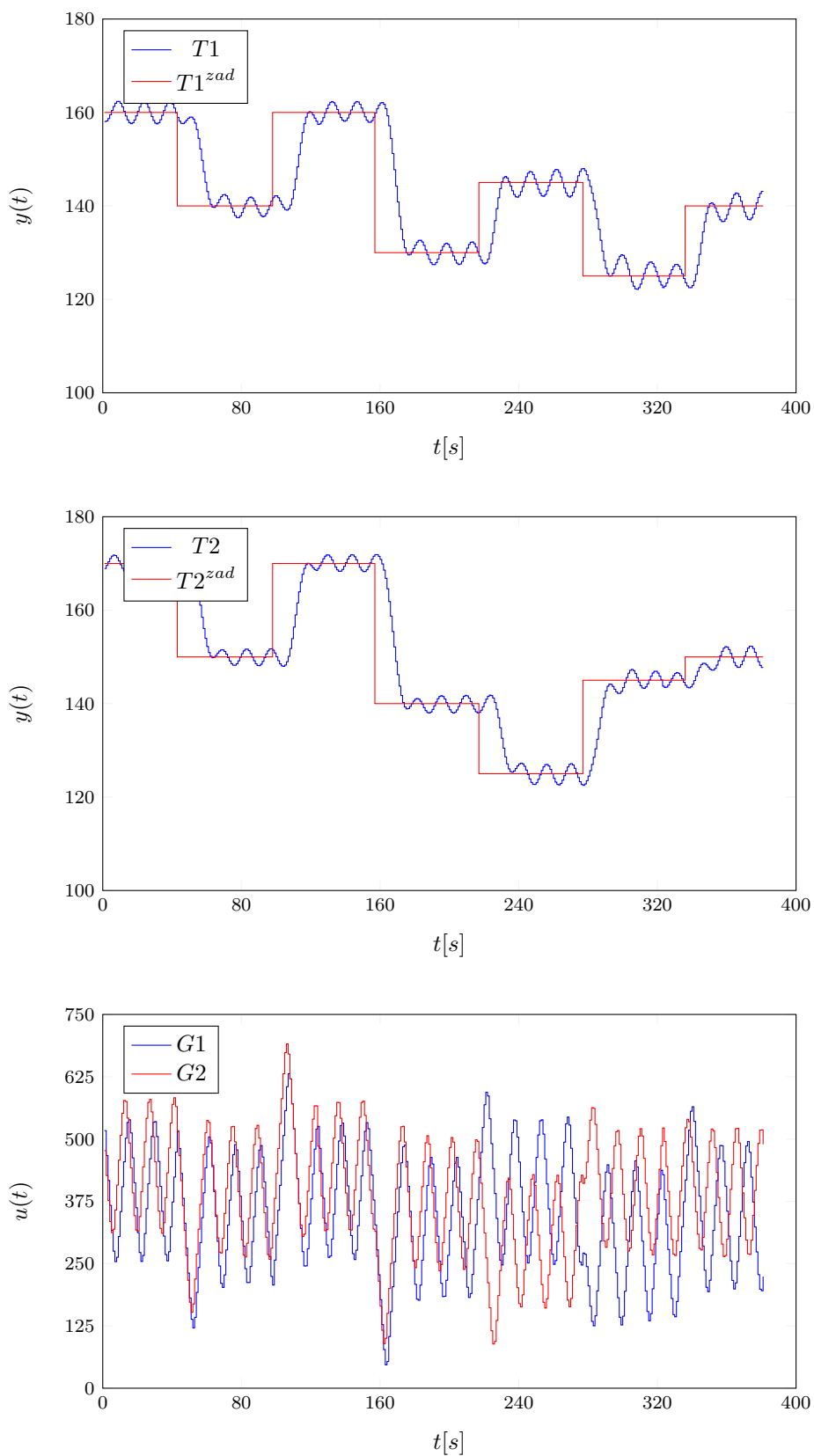
Drugim krokiem było przeniesienie znalezionych parametrów na sterownik. Zgodnie z naszymi oczekiwaniami nie była ona idealna, chociaż o wiele lepsza niż w przypadku losowo wybranych parametrów. Dalsze poprawki zostały nałożone metodą inżynierską. Manipulacja wartościami  $N$ ,  $N_u$  i  $\lambda$  w niewielkich otoczeniach punktów znalezionych w symulacji pozwoliło poprawić zarówno parametry sterowania, tj. przeregulowania i czas regulacji, a także złagodzić przebiegi wartości

<sup>2</sup> Wykorzystane zostały trajektorie użyte do zebrania odpowiedzi skokowych z obiektu

<sup>3</sup> Optymalizacja przeprowadzana na pojedynczej trajektorii może dawać niezadowalające efekty, gdy obiekt jest nieliniowy. Ze względu na obszerność projektu zdecydowaliśmy się jednak pozostać przy takim scenariuszu, aby nie zwiększać czasu poświęconego na zbieranie danych z obiektu. Jak wynika z naszych dotychczasowych doświadczeń, stanowisko G-C ma charakter liniowy przy pracy w niewielkim zakresie wokół punktu pracy. Biorąc pod uwagę, że modele miały posłużyć jedynie do wstępnej kalibracji regulatora, podejście to powinno umożliwić osiągnięcie zamierzonych rezultatów.

<sup>4</sup> Klasa DMC implementuje regulator DMC dla ogólnego przypadku  $n \times m$ -wymiarowego

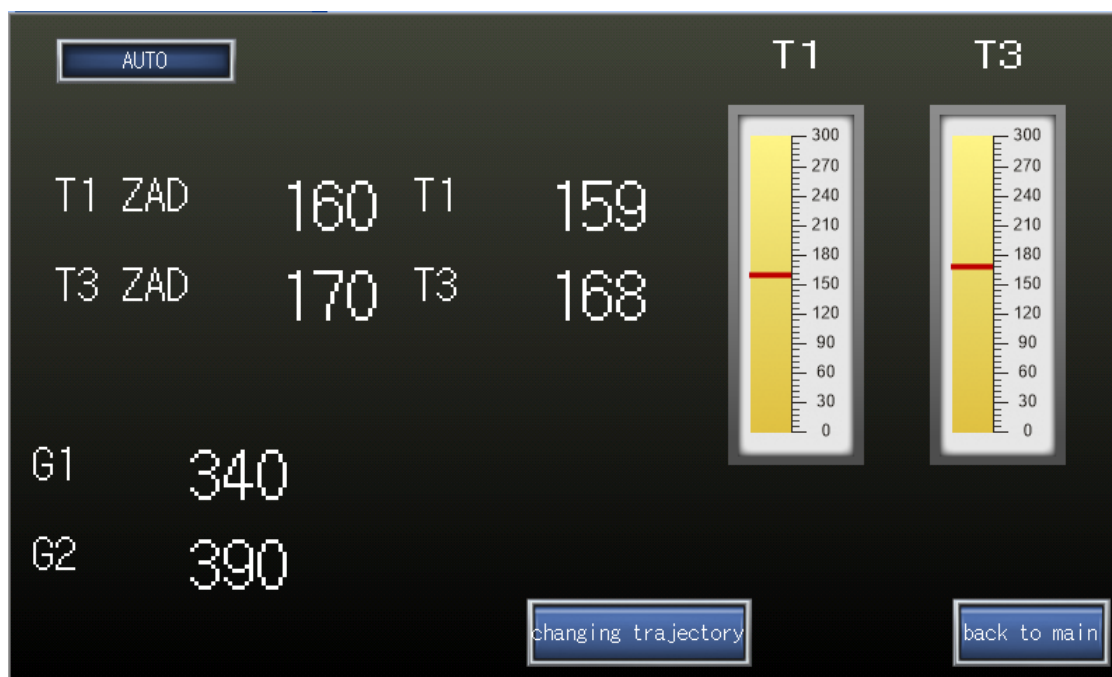
sterujących. Praca ostatecznej wersji regulatora dla przykładowej trajektorii zadanej została przedstawiona na rys. 2.5



Rys. 2.5: Przebiegi sygnałów sterujących i wyjściowych przy ostatecznej wersji reulatora DMC

Jak widać przebiegi nie są idealne. Szczególnie uderzające są oscylacje wartości wyjściowej w stanie ustalony o amplitudzie rzędu  $3 - 4^{\circ}\text{C}$ . Niestety mimo długiego czasu strojenia nie udało się ich wyeliminować. Biorąc pod uwagę, że stanowisko operuje na temperaturach w zakresie  $25 - 200^{\circ}\text{C}$  oscylacje te nie wydają się tak krytyczne. Obserwując pozostałe cechy regulacji można zauważyć, że wartości zadane są osiąmane stosunkowo szybko. Ponadto praktycznie nie występuje przeregulowanie. Miejscem wartym poprawy jest natomiast przebieg wartości sterującej. Oscylacje, które występują na grzałkach  $G1$  i  $G2$  mają stosunkowo wysokie amplitudy i dość krótki okres. Zwiększenie parametru  $\lambda$  mogło by poprawić tę sytuację, jednak przy zmianie tej bez wprowadzanie poprawek także do innych parametrów jakość regulacji ulegała pogorszeniu. Kolejnym pomysłem na poprawę algorytmu byłoby, implementowane już przez nas w poprzednich projektach, "przewidywanie" wartości zadanej. Jako, że w wielu praktycznych przypadkach sterowania jest ona odgórnie znana, warto zawrzeć tę informację w wektorze  $Y^{zad}$ , na którym operuje algorytm. Sprawia to, że regulator podejmuje akcję jeszcze przed faktyczną zmianą aktualnej wartości zadanej, co może zwiększyć stabilność układu.

## 2.6. Panel HMI i kontrola trajektorii zadanej



Rys. 2.6: Część główna panelu operatora

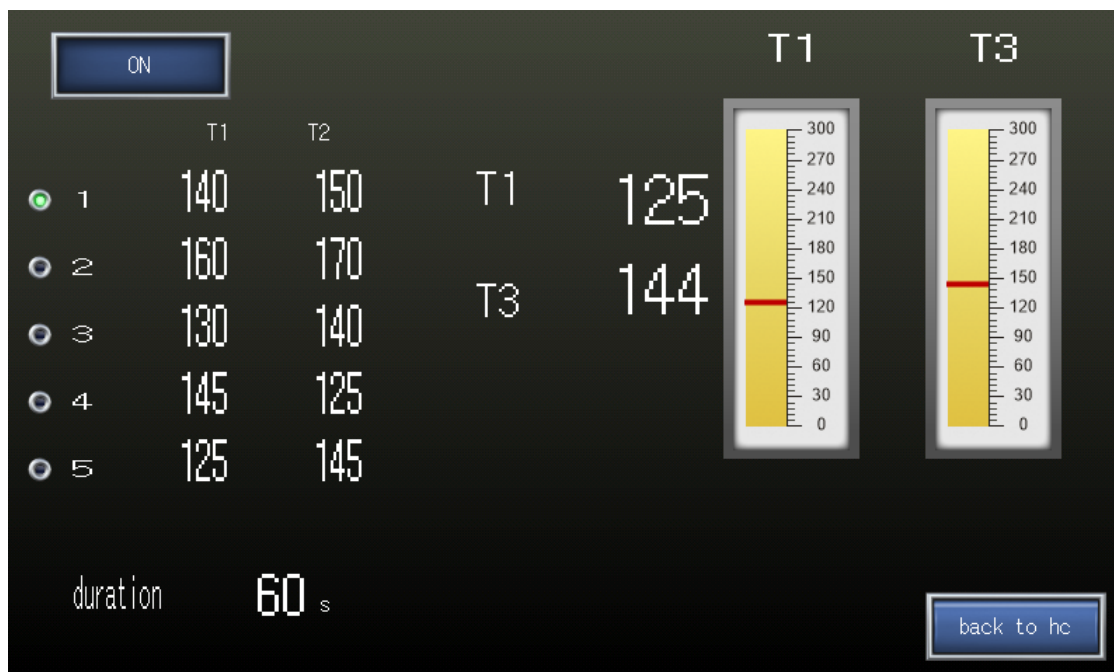
Stworzony panel operatorski posiada trzy poziomy. Pierwszy z nich pozwala po prostu na wybór obiektu sterowania. Po wejściu w sekcję poświęconom stanowisku grzejąco-chłodzącemu ukazuje nam się obraz widoczny na rys. 2.6. Wartości temperatur  $T1$  i  $T3$  zostały zwizualizowane w postaci słupków z podziałką i poruszających się po nich czerwonych poprzeczek. Dokładna wartość liczbową wyświetlana jest na lewo od wskaźników. Na tej samej wysokości, po lewej stronie panelu znajdują się pola pozwalające ustawić wartość zadaną<sup>5</sup>. Powyżej został z kolei umiejscowiony przycisk, który umożliwia przełączanie regulatora między trybem manualnym

<sup>5</sup> W przypadku sterowania automatycznego

a automatycznym. W przypadku pierwszego z nich wartości sterowań mogą zostać ustawione przy pomocy pól w lewej dolnej części ekranu.

Po naciśnięciu przycisku "changing trajectory" użytkownik przechodzi do panelu, w którym możliwe jest ustawienie zmienianej skokowo trajektorii zadanej (rys. 2.7). Trajektoria składa się z pięciu przedziałów czasowych, każdy o długości równej wartości ustawionej w polu "duration". Dla każdego z przedziałów możliwe jest ustawienie innej wartości zadanej. Po zakończeniu ostatniego segmentu trajektoria wartości zadanej jest zapętlana. Po ustawieniu porządkanej trajektorii użytkownik może uruchomić jej realizację poprzez przełączenie przycisku w lewym górnym rogu ekranu. Implementacja ta rozwiązuje problem postawiony w zadaniu 5. Realizacja programowa odbywa się poprzez prostą sekwencję instrukcji warunkowych widocznych na poniższym listingu. Wykonywana jest ona przy każdej iteracji programu.

```
IF HC_TRAJ_ON THEN
  IF traj = 0 THEN
    HC_DMC.HC_T1_ZAD := HC_T1_TRAJ1;
    HC_DMC.HC_T2_ZAD := HC_T2_TRAJ1;
  ELSIF traj = 1*HC_DURATION THEN
    HC_DMC.HC_T1_ZAD := HC_T1_TRAJ2;
    HC_DMC.HC_T2_ZAD := HC_T2_TRAJ2;
  ELSIF traj = 2*HC_DURATION THEN
    HC_DMC.HC_T1_ZAD := HC_T1_TRAJ3;
    HC_DMC.HC_T2_ZAD := HC_T2_TRAJ3;
  ELSIF traj = 3*HC_DURATION THEN
    HC_DMC.HC_T1_ZAD := HC_T1_TRAJ4;
    HC_DMC.HC_T2_ZAD := HC_T2_TRAJ4;
  ELSIF traj = 4*HC_DURATION THEN
    HC_DMC.HC_T1_ZAD := HC_T1_TRAJ5;
    HC_DMC.HC_T2_ZAD := HC_T2_TRAJ5;
  ELSIF traj = 5*HC_DURATION THEN
    traj :=0;
  ELSE;
  END_IF;
  traj := traj + 1;
END_IF;
```



Rys. 2.7: Panel umożliwiający konfigurację trajektorii zadanej dla stanowiska G-C

### 3. Stanowisko Inteco

Po zrealizowaniu ostatniego projektu z Projektowania Układów Sterowania każdy student Automatyki i Robotyki może stwierdzić, że stanowisko grzejąco chłodzące nie ma przed nim żadnych tajemnic. Jeżeli kiedykolwiek zostanie przed kimś z nas postawiony obiekt, to regulator PID lub DMC zaprojektuje i nastroi on intuicyjnie, niemal bezwiednie nawet jeżeli nie będzie to potrzebne. Świadomi takiego stanu rzeczy przechodzimy do konfrontacji z kolejnym obiektem — stanowiskiem Inteco.

Jak już opisano w przypadku stanowiska G-C awaria czujników jest jedną z potencjalnie najniebezpieczniejszych awarii w systemie. Pamiętając o tym także w przypadku stanowiska Inteco zaimplementowaliśmy prosty mechanizm zabezpieczający analogiczny do opisanego wyżej. Po wykryciu wartości krytycznej ( $\geq 19.95$ ) na którymkolwiek z czujników poziomu zapalana jest odpowiednia flaga. Sprawdzana jest ona z kolei w momencie aktualizacji sterowania. Jeżeli jest zapalona, to korespondujący z nią zawór zostaje w pełni otwarty. W przeciwnym wypadku aplikowane jest sterowanie wynikające z regulatora. Blok instrukcji warunkowych odpowiedzialnych za realizację mechanizmu zabezpieczającego został przedstawiony na listingu poniżej.

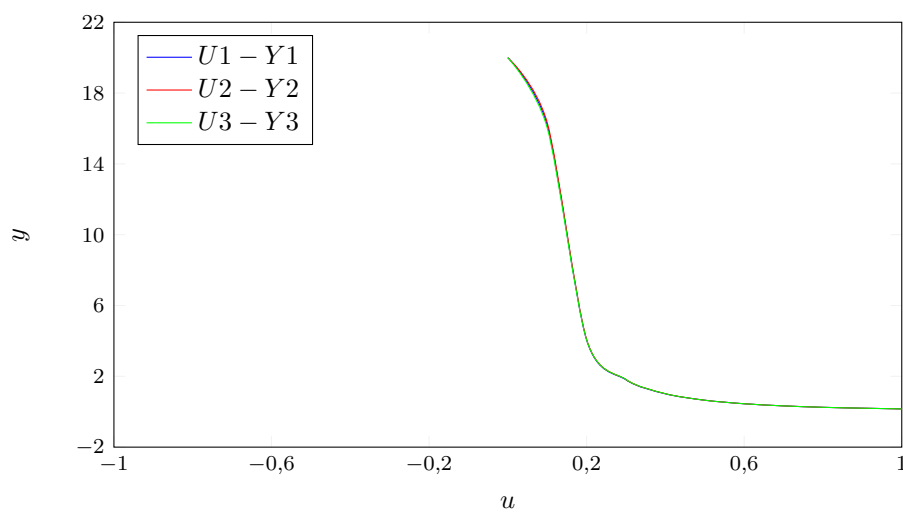
```
IF WATER_Y1 > 19.95 THEN
    WATER_Y1_OVERFLOW := TRUE;
ELSE
    WATER_Y1_OVERFLOW := FALSE;
END_IF;
IF WATER_Y2 > 19.95 THEN
    WATER_Y2_OVERFLOW := TRUE;
ELSE
    WATER_Y2_OVERFLOW := FALSE;
END_IF;
IF WATER_Y3 > 19.95 THEN
    WATER_Y3_OVERFLOW := TRUE;
ELSE
    WATER_Y3_OVERFLOW := FALSE;
END_IF;
```

#### 3.1. Charakterystyka statyczna

Na pierwszy rzut oka stanowisko Inteco można uznać za obiekt  $3 \times 3$ . Chwila refleksji uzmysławia jednak, że nie wszystkie wejścia mają wpływ na wszystkie wyjścia. Na przykład rozszczelnienie zaworu poniżej zbiornika dolnego nie wpływa na poziom wody w zbiorniku górnym. Zadanie 7 postawiło przed nami problem zbadania charakterystyki statycznej obiektu. W przypadku ogólnego obiektu  $3 \times 3$  byłoby to niebagatelne zadanie. Wymagało by ono najpewniej zbadania stanów ustalonych wyjść w pewnym sześciennym zbiorze stanów ustalonych sygnałów wejściowych. Dla krawędzi takiego sześciennego zbioru ekstrapolowanej z  $n$  punktów ( $n$  pomiarów) oznaczałoby to konieczność wykonania  $n^3$  eksperymentów! Innym podejściem, przy założeniu, że obiekt jest, przynajmniej w przybliżeniu, liniowy byłoby zebranie odpowiedzi skokowych dla

każdego toru wejścia-wyjścia i identyfikacja na ich podstawie macierzy transmitancji obiektu. Z macierzy tej możnaby z kolei w sposób analityczny wyznaczyć charakterystykę statyczną.

Problem ten można jednak w znakomitym stopniu uprościć, jeżeli weźmie się pod uwagę pewne charakterystyczne cechy obiektu. Załóżmy, że obiekt znajduje się w stanie ustalonym<sup>1</sup>. Jeżeli zmienimy stopień otwarcia zaworu pod zbiornikiem górnym, to poziom wody zmieni się oczywiście chwilowo we wszystkich zbiornikach<sup>2</sup>. Zauważmy jednak, że po pewnym czasie poziom wody w zbiorniku górnym ustali się na pewnym poziomie. Jeżeli poziom będzie utrzymywany, a dopływ do zbiornika górnego stały to będzie to znaczyć, że wyływa z niego ta sama ilość wody, co przed zmianą pozycji zaworu. Co za tym idzie, poziomy w pozostałych zbiornikach również wróćą do stanu pierwotnego. Rozumowanie to można uogólnić na wszystkie zbiorniki.



Rys. 3.1: Charakterystyki statyczne poszczególnych zbiorników stanowiska Inteco

Spostrzeżenie to nasuwa bardzo praktyczny wniosek: patrząc jedynie z punktu widzenia stanów ustalonych, stanowisko Inteco można traktować jako trzy obiekty wymiaru  $1 \times 1$ . Co za tym idzie, do określenia charakterystyki statycznej obiektu wystarczające jest przeprowadzenie  $n$  eksperymentów dla każdej pary wejście-wyjście. Eksperymenty takie zostały przeprowadzone, a ich wyniki przedstawiono rys. 3.1. Pomiary na każdym torze wykonywane były w przedziale sterowań  $< 0; 1 >$  z krokiem 0.1. Jak widać, tory są silnie nieliniowe. Charakterystyki mogą być przybliżone relacją  $\bar{y}(\bar{u}) \propto \sqrt{\bar{u}}$ , co zgadza się z przewidywaniami teoretycznymi. Zgodnie z treścią zadan projektowych sterowanie obiektem powinno być ograniczone do wartości zadanych z przedziału  $< 10; 15 >$ . W tym zakresie można z powodzeniem przybliżyć charakterystyki funkcjami liniowymi, co uzasadnia użycie klasycznego regulatora PID do sterowania obiektem.

<sup>1</sup> Tak jak w treści zadania zakładamy, że dopływ do górnego zbiornika jest stały, a manipulujemy jedynie stopniem rozszczelnienia zaworów

<sup>2</sup> Ze zbiornika górnego zacznie wypływać więcej wody, zatem więcej wody wpadnąć będzie do zbiornika środkowego, przez co zwiększy się w nim ciśnienie hydrostatyczne. Poskutkuje to zmianą wypływu z tego zbiornika i zająściami podobnych zmian w zbiorniku dolnym.

### 3.2. Własna implementacja PID

Implementacja klasycznego regulatora PID omawiana była na łamach naszych sprawozdań już niejednokrotnie. Zasadniczo i w tym przypadku nie różni się ona niczym prócz wywołań funkcji dostępnych w środowisku *GX works*. Regulator został napisany w wariantcie inkrementalnym, w którym w pamięci przechowywane są dwie poprzednie wartości uchybu oraz ostatnia wartość sterowania. Na bazie równań ?? - ?? wyznaczone są przyrosty sterowań w kolejnych iteracjach. Generyczny regulator PID został zaimplementowany jako osobny, parametryzowany blok funkcyjny. Ograniczenia wartości sterowań są w nim uwzględniane w sposób progowy poprzez ich ucinanie po opuszczeniu przedziału  $< 0, 1 >$ .

$$u(k) = r_0 * e(k) + r_1 * e(k - 1) + r_2 * e(k - 2) \quad (3.1)$$

$$r_0 = K * (1 + \frac{T_s}{2 * T_i} + \frac{T_d}{T_s}) \quad (3.2)$$

$$r_1 = K * (\frac{T_s}{2 * T_i} - \frac{2 * T_d}{T_s} - 1) \quad (3.3)$$

$$r_2 = K * \frac{T_d}{T_s} \quad (3.4)$$

Tym razem zdecydowaliśmy się spróbować nastroić regulator bezpośrendio na obiekcie <sup>3</sup>. Gdyby podejście takie zawiodło lub okazało się zbyt czasochłonne mogliśmy w każdym momencie pozyskać odpowiedzi skokowe z liniowego obszaru charakterystyki i na ich podstawie zidentyfikować macierz transmitancji, która posłużyłaby nam do nastrojenia regulatora w trybie off-line. Działanie takie okazało się jednak niepotrzebne, gdyż z powodzeniem udało nam się nastroić regulatory metodą inżynierską.

Rozpoczęliśmy od ustawienia stałych wartości sterowań  $U2$  i  $U3$  <sup>4</sup> równych 1. Następnym krokiem było uruchomienie regulatora PID sprzężonego z sygnałami  $U1$ ,  $Y1$ . Po wyłączeniu członów całkującego i różniczkującego dobraliśmy wartość członu proporcjonalnego tak, aby zapewnić jak najmniejszy uchyb ustalony przy jednoczesnym niewprowadzaniu nadmiernych oscylacji do układu. Kolejnym krokiem było dobranie członu całkującego niwelującego uchyb ustalony bez nadmiernego przeregulowania. Ostatecznie włączyliśmy człon różniczkujący i postaraliśmy się o poprawienie dynamiki całego układu. Po kilku drobnych korektach mających ograniczyć przeregulowania związane z włączeniem różniczki oraz złagodzeniem oscylacji sterowań cały proces został powtórzony kolejno dla środkowego i dolnego zbiornika <sup>5</sup> udało nam się uzyskać regulator, którego przebiegi dla przykładowej trajektorii zadanej widoczne są na rys 3.2.

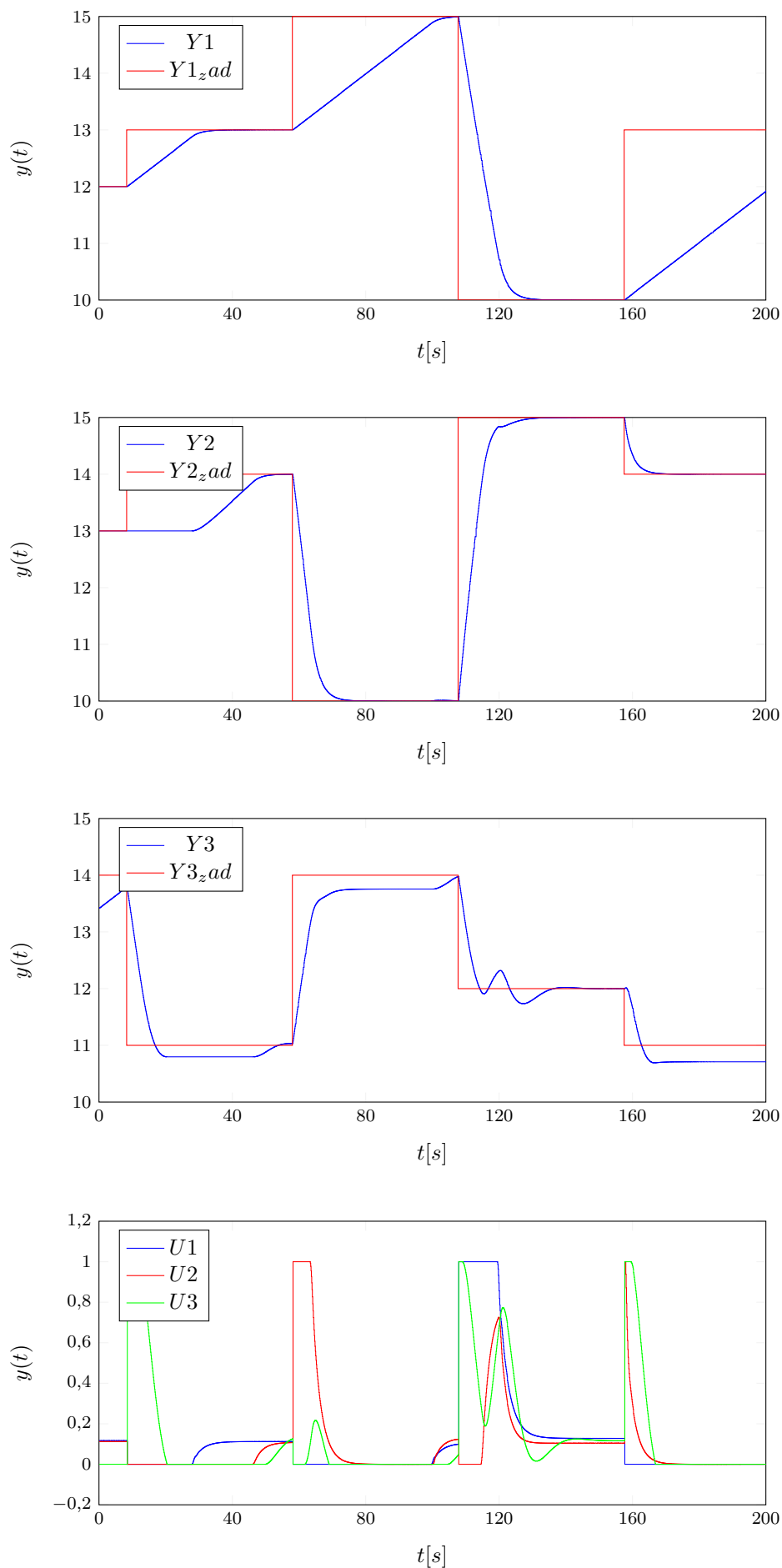
Regulator działa poprawnie. Wszystkie wartości zadane są osiągalne, przeregulowanie praktycznie nie występuje. Przebiegi sygnałów sterujących nie oscylują. Jediną wadą naszego rozwiązania jest czas regulacji. W przypadku spadku wartości zadanej jest on stosunkowo krótki, jednak przy wzroście napływ wody do zbiornika ma charakter co najwyżej liniowy. Wynika to wprost z charakteru obiektu sterowania a także z braku interferencji między regulatorami. Jako, że możliwe jest sterowanie jedynie odpływem, a nie dopływem, nie da się tego stanu rzeczy zmienić. Gdyby zastosowany został wielowymiarowy regulator DMC sytuacja mogła by ulec poprawie **w niektórych przypadkach**.

<sup>3</sup> Należy pamiętać, że podejście takie nie zawsze jest możliwe. Jeżeli sterowalibyśmy reaktorem jądrowym, to z pewnością bardziej racjonalnym podejściem byłoby stworzenie modelu obiektu, który posłużyłby do strojenia algorytmu.

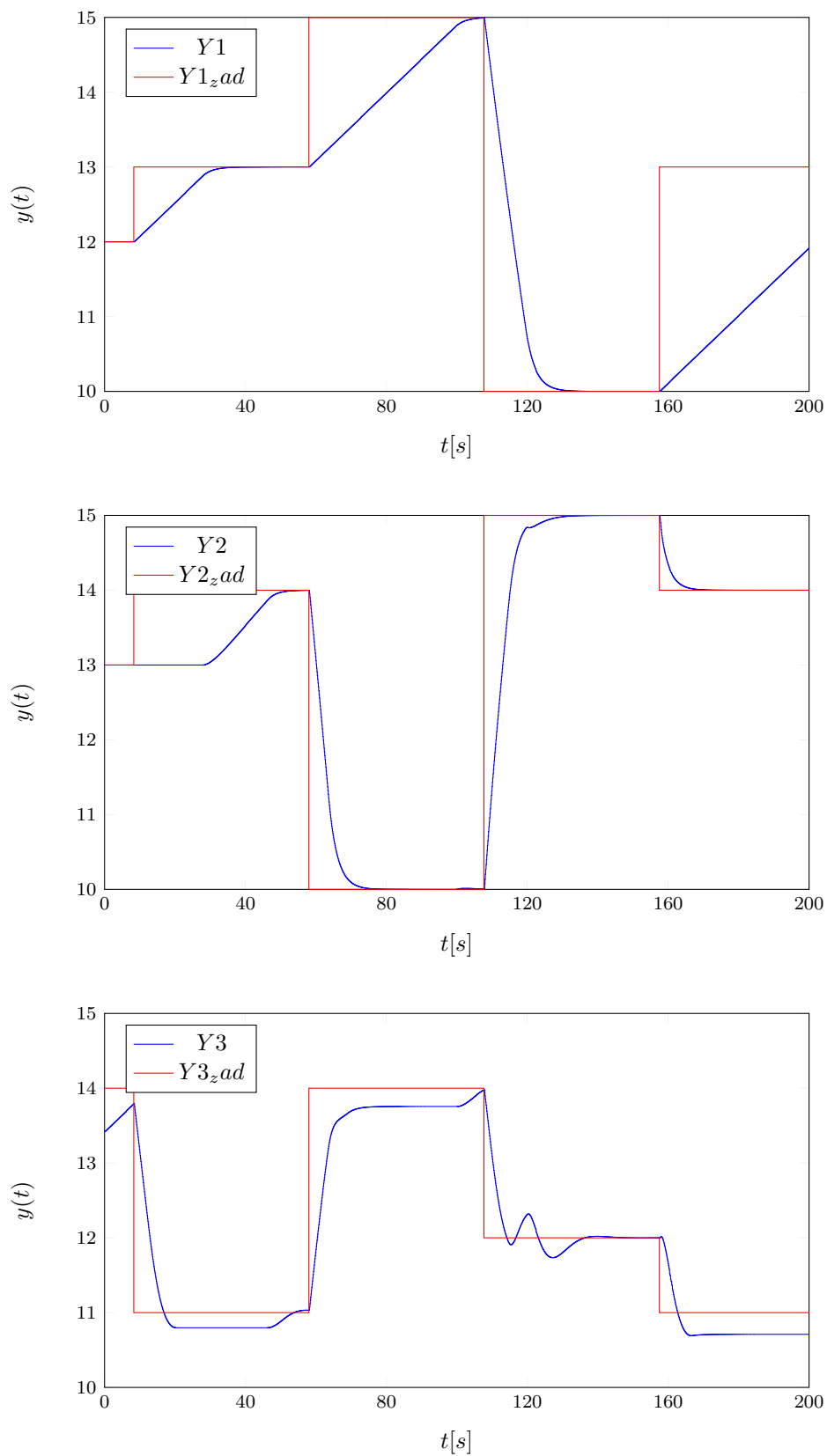
<sup>4</sup> zawory pod zbiornikami środkowym i dolnym

<sup>5</sup> Przy strojeniu natępnych regulatorów poprzednie pozostawały włączone

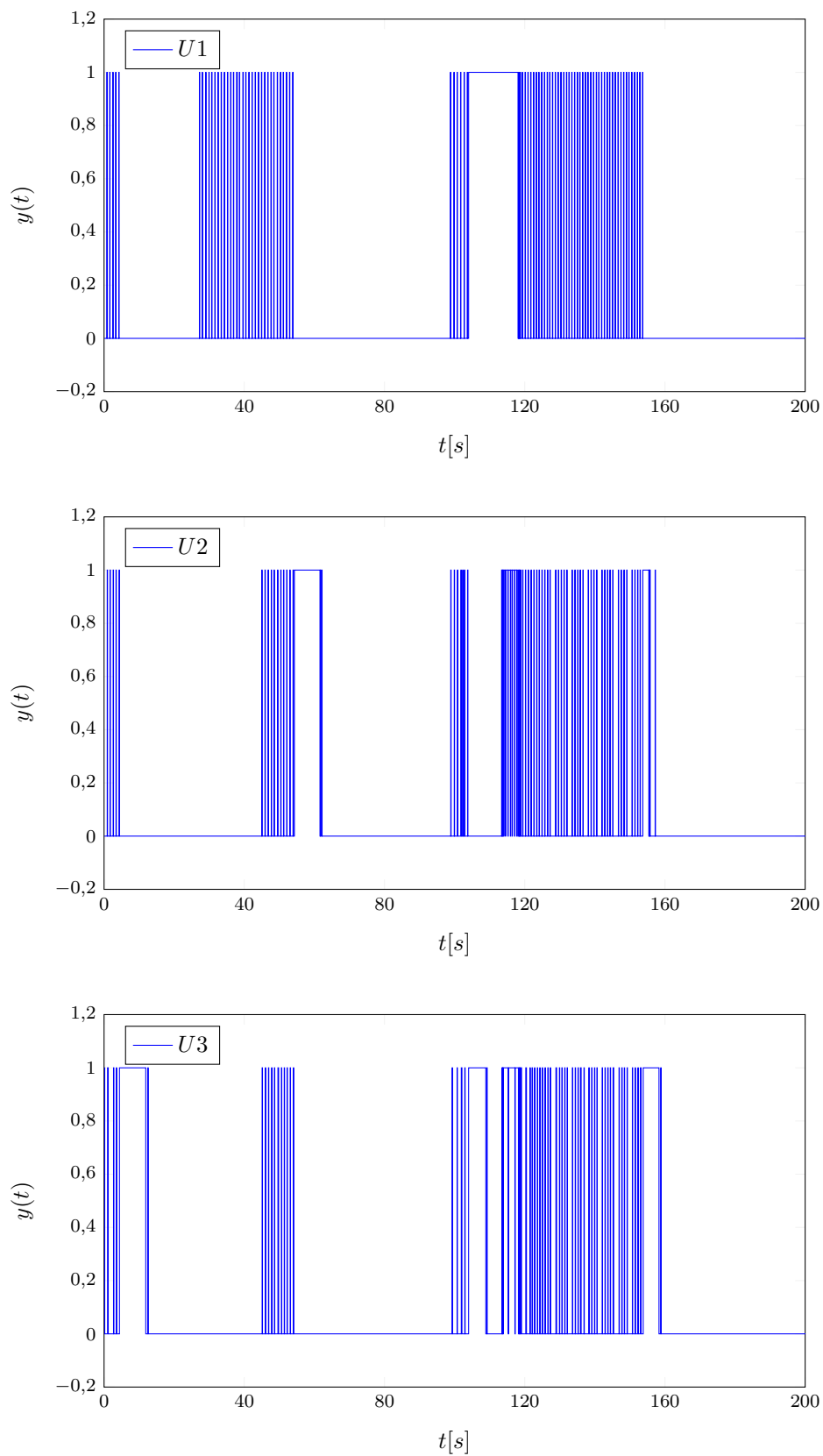




Rys. 3.2: Przebiegi sygnałów sterujących i wyjściowych przy ostatecznej wersji autorskiego reulatora PID



Rys. 3.3: Przebiegi sygnałów wyjściowych przy ostatecznej wersji wbudowanego reulatora PID



Rys. 3.4: Przebiegi sygnałów sterujących przy ostatecznej wersji wbudowanego reulatora PID

### 3.3. Wbudowany regulator PID

Środowisko *GX Works* posiada wbudowany blok funkcyjny realizujący regulator PID. Kosztem zupełnie nieintuicyjnego interfejsu niweluje ona potrzebę budowania jego własnej implementacji. W ramach projektu przetestowaliśmy jego działanie w tych samych warunkach, w których testowany był nasz autorski regulator. Procedura strojenia była identyczna z tą opisaną w poprzedniej sekcji. Wyniki pracy nastrojonego regulatora przedstawia rys. ?? i ??.

Nastawy znalezione dla regulatora wbudowanego są identyczne jak w przypadku wersji autorskiej. Przy takim ustawieniu przebiegi wartości wyjściowych są również identyczne, przez co wszystkie wnioski z poprzedniego paragrafu pozostają w mocy. Jedyną, potencjalnie znaczącą, różnicą między regulatorami jest sposób zarządzania sygnałami sterującymi. Jak widać na rys. ?? wbudowany regulator steruje za pomocą **gęstości sygnałów binarnych**. Podejście takie jest jak najbardziej do przyjęcia w przypadku elementów wykonawczych o dużej bezwładności. W przypadku elementów "szybszych" może to doprowadzić do ich uszkodzenia.

### 3.4. Trajektorie zadane

W ramach panelu operatorskiego zimplementowana została zakładka pozwalając zmieniać trajektorię zadaną wszystkich trzech wyjść procesu w trzech przedziałach czasowych równej długości <sup>6</sup>. Tak jak w przypadku stanowiska G-C implementacja sprowadza się do blok instrukcji warunkowych śledzących wartość czasu *traj*. Gdy znajduje się on w pwnych ramach wartości sterowania ustawiane są na te widoczne w paneli HMI. Fragment kodu odpowiedzialny za automat stanów przedstawiony został na poniższym listingu.

```
IF WATER_TRAJ_ON THEN

  IF WATER_DURATION <> WATER_DURATION_HMI THEN
    WATER_DURATION := WATER_DURATION_HMI;
    traj := 0;
  END_IF;

  IF traj <= 1*WATER_DURATION_HMI-1 THEN
    PID1.SV := WATER_Y1_TRAJ1;
    PID2.SV := WATER_Y2_TRAJ1;
    PID3.SV := WATER_Y3_TRAJ1;
    WATER_CURRENT_TRAJ1 := TRUE;
    WATER_CURRENT_TRAJ2 := FALSE;
    WATER_CURRENT_TRAJ3 := FALSE;
  ELSIF traj <= 2*WATER_DURATION_HMI-1 THEN
    PID1.SV := WATER_Y1_TRAJ2;
    PID2.SV := WATER_Y2_TRAJ2;
    PID3.SV := WATER_Y3_TRAJ2;
    WATER_CURRENT_TRAJ1 := FALSE;
    WATER_CURRENT_TRAJ2 := TRUE;
    WATER_CURRENT_TRAJ3 := FALSE;
  ELSIF traj <= 3*WATER_DURATION_HMI-1 THEN
    PID1.SV := WATER_Y1_TRAJ3;
    PID2.SV := WATER_Y2_TRAJ3;
    PID3.SV := WATER_Y3_TRAJ3;
    WATER_CURRENT_TRAJ1 := FALSE;
    WATER_CURRENT_TRAJ2 := FALSE;
    WATER_CURRENT_TRAJ3 := TRUE;
  ELSIF traj >= 3*WATER_DURATION_HMI-1 THEN
```

<sup>6</sup> Opis z perspektywy użytkownika zamieszczony został w sekcji 7

```

        PID1.SV := WATER_Y1_TRAJ1;
        PID2.SV := WATER_Y2_TRAJ1;
        PID3.SV := WATER_Y3_TRAJ1;
        WATER_CURRENT_TRAJ1 := TRUE;
        WATER_CURRENT_TRAJ2 := FALSE;
        WATER_CURRENT_TRAJ3 := FALSE;
        traj := 0;
    ELSE;
    END_IF;
    traj := traj + 1;
ELSE
    WATER_CURRENT_TRAJ1 := FALSE;
    WATER_CURRENT_TRAJ2 := FALSE;
    WATER_CURRENT_TRAJ3 := FALSE;
    traj := 0;
END_IF;

```

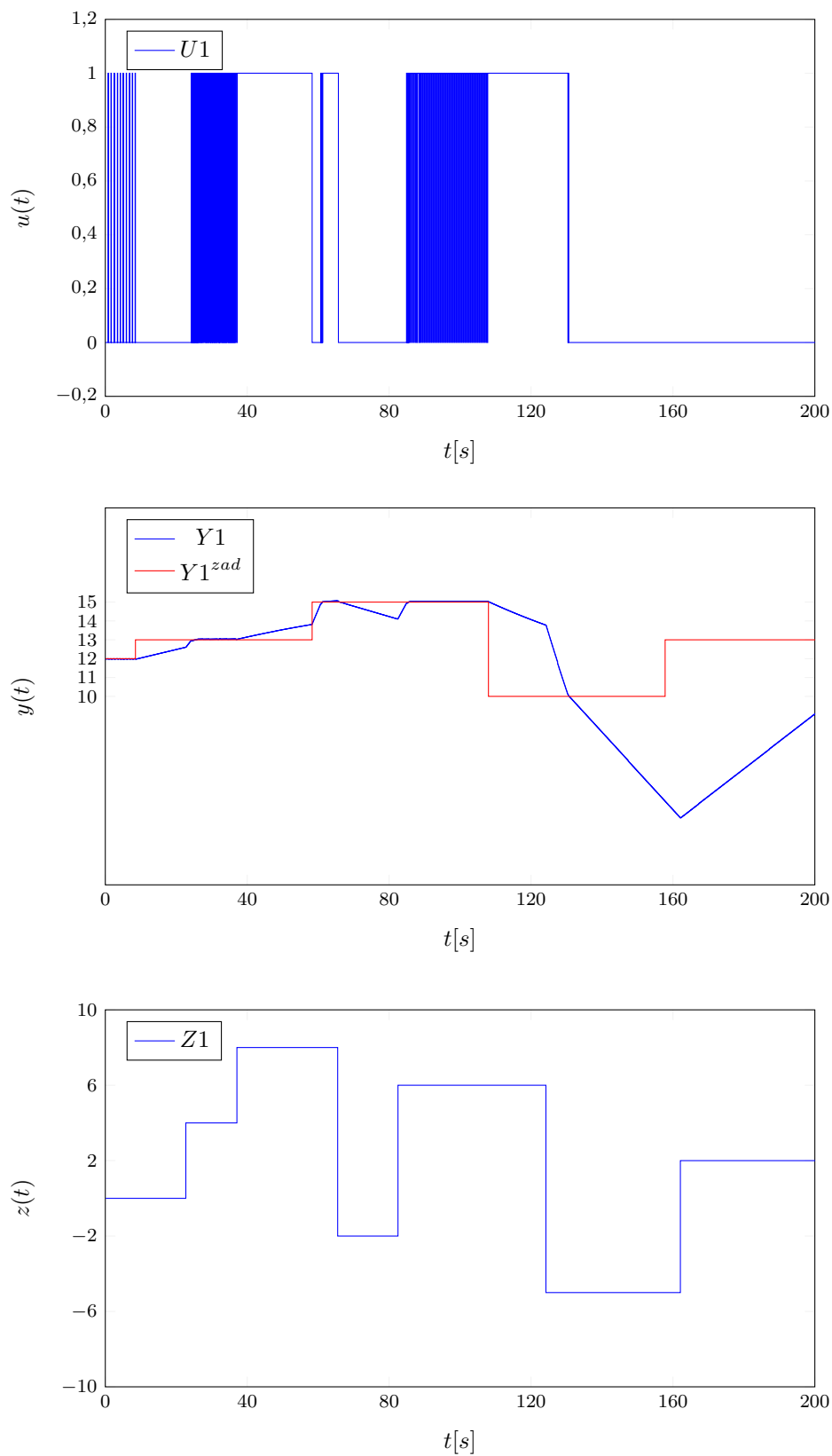
Zmienne WATER\_CURRENT\_TRAJx odpowiedzialne są za wizualizację aktualnego stanu układu. Po przejściu automatu przez wszystkie stany jest on resetowany, a cała sekwencja powtarza się.

### 3.5. Badanie odporności wybranego regulatora

Badania odporności przeprowadzono na regulatorze sprzężonym z najwyższym zbiornikiem. Testy zostały przeprowadzone na tej samej trajektorii co testy w zadaniach 8 i 9. Rezultaty zostały przedstawione na rys 3.5. Jak widać, regulator nie radzi sobie dobrze w szerszym zakresie zakłóceń. Analizę takiego stanu rzeczy warto rozbić na dwie części. Po pierwsze, jeżeli zakłócenia się dodatnie (woda jest dopływa do zbiornika), to w pewnym zakresie ich wartości regulator jest na nie niemal w pełni odporny. Wynika to z faktu, że regulator jest zazwyczaj w stanie otworzyć sterowany zawór bardziej niż wynikałoby to z niezakłóconej pracy <sup>7</sup>. Na rysunkach zostały przedstawione jedynie przebiegi związane ze zbiornikiem górnym. Należy jednak pamiętać, że taki zabieg ma wpływ na zbiorniki znajdujące się poniżej. Drugi przypadek, to zakłócenia ujemne (woda wyciekająca ze zbiornika). W takiej sytuacji regulator radzi sobie znacznie gorzej. Wynika to z faktu podkreślonego już w poprzednich paragrafach - sterownik nie może zapewnić dodatkowego dopływu do zbiorników, a jedynie zmniejszyć stopień otwarcia zworu. Tu też ponownie objawia się wada niezależnie działających regulatorów. Nieświadome sytuacji w innych zbiornikach nie są one w stanie skoordynować swoich działań, aby zapewnić najmniejszy **średni uchyb** na wszystkich zbiornikach <sup>8</sup>. Omówionych problemów nie da się zniwelować poprzez zmianę dobranych parametrów regulacji. Poprawa odporności wymagała by całkowitej zmiany konstrukcji zestawu regulatorów.

<sup>7</sup> Obiekt jest regulowany w zakresie wyjściowym < 10; 15 > zatem zawór w niewielu przypadkach jest otwarty w całości

<sup>8</sup> Uwaga ta dotyczy w większym stopniu zbiorników znajdujących się niżej



Rys. 3.5: Przebiegi sygnałów procesowych w trakcie testowania odporności górnego regulatora na zewnętrzne zakłócenie

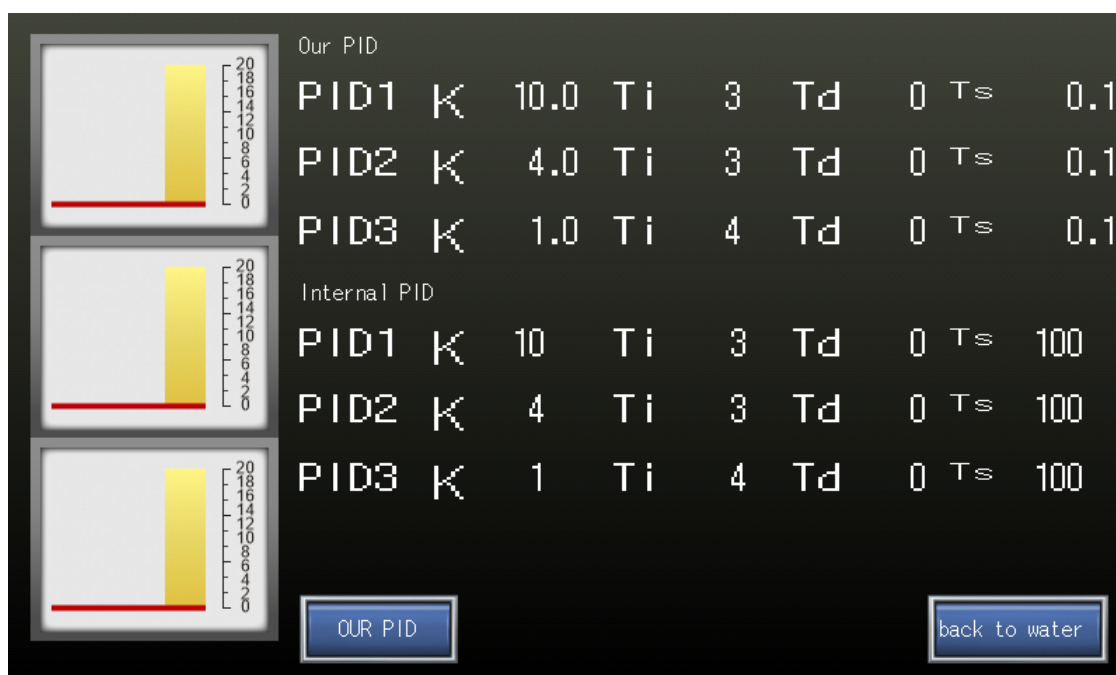
### 3.6. Wizualizacja procesu

HMI przeznaczony do obsługi stanowiska Inteco składa się z trzech zakładek. Pierwsza z nich, widoczna na rys. 3.6, to karta główna. Widoczna jest na niej wizualizacja wartości zmiennych wyjściowych oraz dokładne wartości liczbowe zarówno wielkości zadanych jak i aktualnych. Pola z podpisem SV (ang. *set value*) są edytowalne, a ich wartość wyznacza aktualną wartość zadaną. Kolejna zakładka, ukazana na rys. 3.7, jest przeznaczona do kontrolowania nastaw regulatora. Widoczne po lewej stronie słupki pozwalają ponownie monitorować stan poziomów wody w zbiornikach. Centralną część ekranu zajmuje zbiór edytowalnych pól umożliwiających zmianę nastaw regulatora w trybie on-line. Przycisk widoczny po lewej stronie umożliwia natomiast przełączanie pomiędzy autorskim, a wbudowanym regulatorem PID.

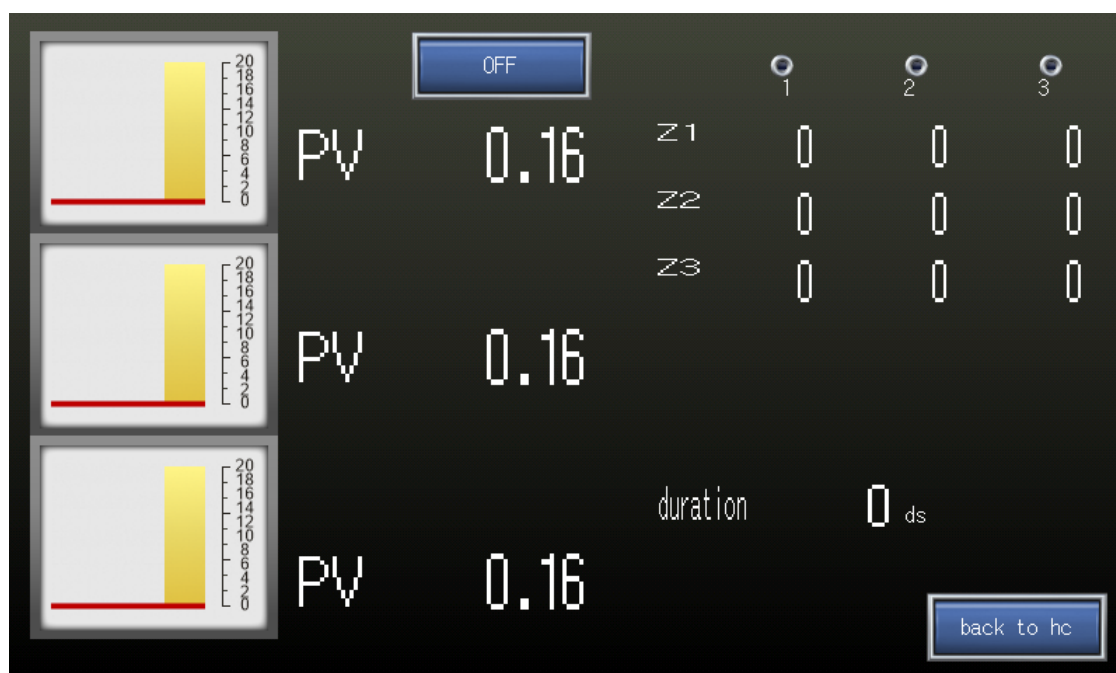


Rys. 3.6: Panel główny dla stanowiska Inteco

Ostatnia karta przeznaczona jest to zadawania trajektorii zmian wartości zadanej. Jak wspomniano w poprzednich sekcjach, zaimplementowany został automat stanów umożliwiający zaplanowanie wartości zadanych w trzech równych przedziałach czasowych. Jest to możliwe poprzez wpisanie odpowiednich wartości w pola widoczne w prawym górnym rogu rys. ???. Przełączenie między wartościami zadanymi wynikającym z automatu stanu, a tymi podanymi na panelu głównym następuje na skutek przycisku znajdującego się w górnej części rysunku. Gdy podążanie za trajektorią zadaną jest włączone, aktualny stan automatu wizualizowany jest w postaci jednej z lampek zapalanych w prawym górnym rogu ekranu. Okres przedziałów czasowych konfiguruje się poprzez zmianę wartości pola `duration`. Wykresy obecne w lewej części ekranu oraz przystające do nich pola liczbowe pozwalają monitorować poziom rozszczelnienia poszczególnych zaworów.



Rys. 3.7: Panel konfiguracyjny regulatorów PID



Rys. 3.8: Panel konfiguracji trajektorii zadanych.



## 4. Podsumowanie

Tak duży, a przy tym ostatni projekt z przedmiotu wymaga pewnej formy podsumowania. Autorska implementacja wielowymiarowych reuglatorów PID oraz DMC są z pewnością czymś, co każdy student Automatyki i Robotyki naszego wydziału powinien chociaż raz w życiu wykonać. Praktyka taka pozwala uświadomić sobie wiele faktów niewynikających wprost z teorii sterowania, żeby wymienić chociaż wpływ braku sprzężenia zwrotnego między niezależnymi reulatorami PID lub praktyczne aspekty wykorzystania modelu w procesie projektowania układu sterowania. Również poznanie metod uwzględniania znanych sygnałów zakłóceń w procesie regulacji jest bardzo korzystnym aspektem wykonanych projektów. W ogólności można powiedzieć, że koncepcja stojąca za projektami realizowanymi w ramach Projektowania Układów Sterowania stanowi dobre podsumowanie wszystkich poprzednich przedmiotów oscylujących wokół tematu automatyki. Mielśmy przy tej okazji możliwość powtórzenia zdobytych do tej pory informacji na temat regulatorów PID oraz DMC, przeciećwiczenia obsługi sterownika PLC oraz wykorzystania szerokiego zakresu narzędzi z dziedziny identyfikacji i modelowania poznanych w toku edukacji. Te aspekty stanowią mocną stronę przedmiotu i w uznaniu autorów pozwalają określić go mianem obowiązkowego dla studentów naszego kierunku.

W ten sposób kształtują się ogólne wnioski wysnute z perspektywy studentów kończących ten przedmiot. Rzeczywistość, z którą mierzyliśmy się w trakcie trwania semestru miała jednak odcień dalece bardziej wpadający w czerń rozczarowań, przygnębienia i poczucia marnowanego czasu. Praca automatyka nieodczownie łączy się z potrzebą akwizycji i analizy dużej ilości danych z obiektów regulacji oraz długimi godzinami spędzonymi na kalibracji regulatorów, z czego w pełni zdajemy sobie sprawę. Jeżeli jednak doświadczenia takie czekają nas w życiu zawodowym, to czas laboratoriów oraz projektów wydaje się tym bardziej odpowiedni by od takich zagadnień abstrahować w nieco większym stopniu, a uzyskany w ten sposób czas przeznaczyć na poszerzenie świadomości studentów w obrębie omawianej dziedziny.

Każdy z nas słyszał o regulatorach PID czy DMC niejednokrotnie. Każdy niejednokrotnie miał okazję je implementować. Powtarzanie tych samych informacji na przestrzeni trzech lub czterech przedmiotów, które przypomnijmy są obowiązkowe dla studentów naszego kierunku, wydaje się być bezcelowe. Bezcelowość takich działań wydaje się tym bardziej dosadna jeżeli uświadomić sobie, że poza dwoma wymienionymi regulatorami nie mieliśmy okazji zapoznać się z innymi podejściami do tematu. Poszerzenie wiedzy o regulacji predykcyjnej poprzez wprowadzenie algorytmu GPC czy ukazanie aspektu stosowalności sztucznych sieci neuronowych w sterowaniu to tylko niektóre z pomysłów, jakie mogłyby poszerzyć horyzonty studentów, a przy tym zwiększyć ich zaangażowanie w przedmiot.

Również wspomniane już godziny spędzone na zbieraniu danych z obiektów oraz kolejne godziny przeznaczone na formatowanie ich tak, aby mogły być zamieszczone w obszernych sprawozdaniach realizowanych po każdym z bloków wydają się w odczuciach autorów zbędne. Zadania takie nie wnoszą niemal nic do zestawu umiejętności studenta, a jedynie budzą frustrację wynikającą z niemocy zajęcia się tematami bardziej wartościowymi. W naszej opinii fakt powtarzania takiego schematu na niemal wszystkich przedmiotach z omawianej grupy stanowi silną przesłankę do podjęcia znacznych zmian w materii zarządzania czasem studenta. Trzymamy kciuki zarówno za studentów przyszłych lat jak i prowadzących, aby wspólną pracą wypracowali konsensus, który łączyłby w sobie zarówno dotychczasowy nakład pracy wymagany od studentów jak i satysfakcję wynikającą z poszerzania przez nich wiedzy i umiejętności.