

Prototyp kaskadowego autopilota okrętowego zaimplementowany w środowisku CPDev

Zbigniew Świder

Politechnika Rzeszowska, Katedra Informatyki i Automatyki, al. Powstańców Warszawy 12, 35-959 Rzeszów

Streszczenie: Autopiloty okrętowe można podzielić na konwencjonalne, zdolne jedynie do utrzymywania zadanego kursu, oraz zaawansowane, które dodatkowo mogą utrzymywać statek na „ścieżce” łączącej zadane punkty nawigacyjne na trasie). W artykule przedstawiono strukturę prototypowego autopilota statku zaimplementowanego w środowisku CPDev oraz wzory pozwalające wyznaczyć nastawy regulatora kursu (PID_H) oraz ścieżki (PID_T) w regulacji kaskadowej. Dla każdego z nich przyjęto pojedyncze parametry projektowe określające dynamikę zamkniętej pętli regulacji. Reguły te zastosowano w oprogramowaniu prototypu autopilota, stworzonego we współpracy z holenderską firmą projektującą systemy sterowania i wizualizacji dla statków.

Słowa kluczowe: układ kaskadowy, regulator PID, autopilot statku, nastawy, środowisko inżynierskie

1. Wprowadzenie

Sterowanie kursem statku jest realizowane zwykle przez autopiloty, które automatycznie korygują kurs wykorzystując wbudowane urządzenia sterowe. Taki adaptacyjny autopilot dostosowuje ustawienia regulatora kursu w zależności od prędkości statku, stanu morza oraz innych czynników. Dodatkowo, oprócz utrzymywania zadanego przez operatora stałego kursu, może być w stanie prowadzić statek po „ścieżce” łączącej kolejne punkty nawigacyjne, i to mimo zakłóceń np. od prądów morskich czy wpływu wiatru. Taki system wymaga zainstalowania modułu nawigacji obszarowej, który oblicza odchylenie od zaprogramowanej trasy, a następnie automatycznie wprowadza konieczne poprawki.

Klasycznym algorytmem sterowania kursem statku (ang. *heading*) jest PID z nastawami wyznaczonymi dla uproszczonego opisu dynamiki, znanego jako model Nomoto [1–3], mającego postać integratora ze stałą czasową. Śledzenie prostoliniowego fragmentu trasy, inaczej ścieżki (ang. *track*), można realizować w układzie kaskadowym [4, 5] z regulatorem nadrzędnym, który to wyznacza kurs dla regulatora podrzędnego. Oba regulatory mogą być typu PID, lub w nowszych rozwiązaniach, zrealizowane jako regulatory rozmyte (ang. *fuzzy*) [6].

Komercyjne autopiloty statków można podzielić na trzy podstawowe grupy – pierwszą grupę stanowią proste regulatory kursu, których nastawy dobierane są ręcznie [12]. Druga obejmuje

urządzenia wyposażone w GPS [13], które mogą kierować statek do wybranego punktu (ang. *Goto Waypoint*) lub przechwytywać ścieżkę (ang. *Goto Track*). Trzecią grupę stanowią wielomodułowe rozwiązania z wbudowanym trybem śledzenia ścieżki oraz możliwością planowania trasy podróży, np. na urządzeniu ECDIS (ang. *Electronic Chart Display Information System*) [14].

W artykule podano zasady określania nastaw, zarówno dla regulatora kursu (PID_H), jak i regulatora „ścieżki” (PID_T) oraz przedstawiono środowisko inżynierskie CPDev, w którym to zaimplementowano prezentowany prototyp autopilota. Został on przetestowany na pełnym (nieliniowym) modelu kontenerowca 4-DOF, zaimplementowanym również w pakiecie CPDev, a wybrane wyniki z badań przedstawiono w dalszej części artykułu.

2. Struktura regulatora kursu

2.1. Model statku

Pełny (nieliniowy) model statku 4-DOF jest opisany równaniami w przestrzeni stanów, jednak jest on zbyt złożony do określenia analitycznych wyrażeń dla obliczenia nastaw poszczególnych regulatorów. Dlatego więc do takiego projektowania zwykle stosuje się uproszczony model statku, zwany modelem Nomoto [1–3], w postaci połączenia członu całkującego z członem inercyjnym (1). Określa on, jak wychylenie steru δ wpływa na kąt ψ (kurs), pod jakim statek płyne.

Wzmocnienie k w modelu zależy wprost proporcjonalnie, a stała czasowa T odwrotnie proporcjonalnie od prędkości statku V , czyli

$$\frac{k}{s(Ts + 1)} = \frac{\psi(s)}{\delta(s)}, \quad \text{gdzie} \quad k = k_0 \frac{V}{V_0}, \quad T = T_0 \frac{V_0}{V}. \quad (1)$$

Przyjmuje się, że wartości referencyjne $\{k_0, T_0, V_0\}$ występujące we wzorach (1) są znane z wcześniejszych manewrów

Autor korespondujący:

Zbigniew Świder, swiderzb@prz.edu.pl

Artykuł recenzowany

nadesłany 10.10.2022 r., przyjęty do druku 17.01.2023 r.



Zezwala się na korzystanie z artykułu na warunkach licencji Creative Commons Uznanie autorstwa 3.0

wykonanych podczas prób na otwartym morzu lub też określone bezpośrednio po wyjściu statku z portu (przed wyjściem w morze). Tutaj wartości k_0 , T_0 obejmują również dynamikę maszyny sterowej i samego steru.

2.2. Regulator kursu

Strukturę prototypowego regulatora kursu statku, będącego modyfikacją rozwiązań przedstawianych w [1–2, 8], pokazano na rys. 1. Wielkości ψ_{ref} oraz ψ to odpowiednio wartość zadana i kurs rzeczywisty statku, natomiast δ oznacza tu kąt wychylenia steru. Wielkości ψ_L , ψ_H są odpowiednio składowymi niski- i wysokoczęstotliwościowymi. Moduł obserwatora stanu generuje dwie estymaty $\hat{\psi}_L$ i $\hat{\psi}_H$, z których to tylko $\hat{\psi}_L$ (niskoczęstotliwościowa) jest wykorzystywana do sterowania kursem, co pozwala na znaczną redukcję reakcji steru na wpływy zakłóceń środowiskowych. Dodatkowy blok ARMAX estymuje nieznaną częstotliwość fal ω_n , natomiast ogranicznik prędkości wielkości zadanej eliminuje ewentualne przeregulowania po skokowej zmianie ψ_{ref} . Wylczone nastawy dla regulatora PID_H są automatycznie adaptowane do aktualnej prędkości statku V .

2.3. Dobór nastaw

Sterowanie kursem statku odbywa się za pomocą klasycznego regulatora PID:

$$R(s) = k_p \left(1 + \frac{1}{T_i s} + T_d s \right) = k_p \frac{(T_1 s + 1)(T_2 s + 1)}{(T_1 + T_2) s}, \quad (2a)$$

$$T_i = T_1 + T_2, \quad T_d = \frac{T_1 T_2}{T_1 + T_2}. \quad (2b)$$

Dla obliczeń nastaw regulatora przyjęto brak oscylacji w odpowiedziach przejściowych oraz przebiegi aperiodyczne krytyczne dla skokowych zmian wielkości zadanej ψ_{ref} . Dodatkowo, stała czasowa zamkniętej pętli regulacji jest określona jako $T_{cl} = T/r$, $r > 1$. Wymaga to, aby stała czasowa T_{cl} była r razy mniejsza

niż stała czasowa T w modelu statku (1). Tak więc, współczynnik r będzie tu głównym parametrem projektowym. Zwiększając r można sprawić, że pętla będzie „szybsza”. Przyjmijmy $T_2 = T$ i wtedy transmitancje układu otwartego i zamkniętego, oraz wzmocnienie K , przyjmują postać:

$$G_{op}(s) = K \frac{T_1 s + 1}{s^2}, \quad G_{cl}(s) = \frac{K(T_1 s + 1)}{s^2 + K T_1 s + K}, \quad K = \frac{k_p k}{T_i}. \quad (3)$$

Z przyjętych założeń wynika, że wyznacznik mianownika transmitancji G_{cl} musi być równy zero, co daje nam warunek na K oraz określa wartości podwójnego bieguna $s_{1,2}$, jak również stała czasowa $T_{cl} = 1/|s_{1,2}| = T_1/2$ ma być równa T/r , tak więc T_1 będzie wynosić

$$K = \frac{4}{T_1^2}, \quad s_{1,2} = -\frac{K T_1}{2} = -\frac{2}{T_1}, \quad T_1 = \frac{2T}{r}. \quad (4)$$

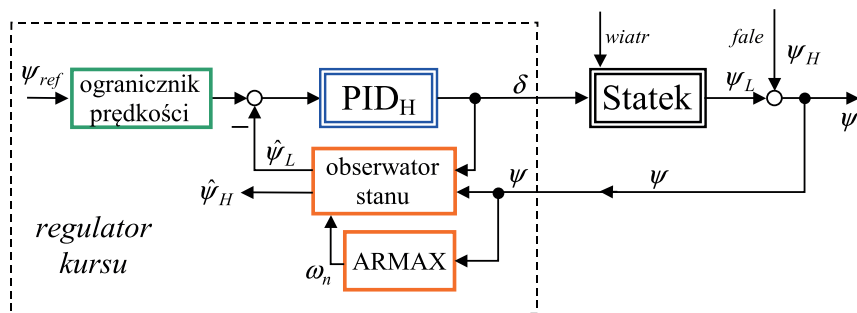
Ostatecznie, nastawy regulatora PID, otrzymane na podstawie powyższych równań, wynoszą

$$k_p = \frac{1}{kT} r(r+2), \quad T_i = T \frac{r+2}{r}, \quad T_d = T \frac{2}{r+2}. \quad (5)$$

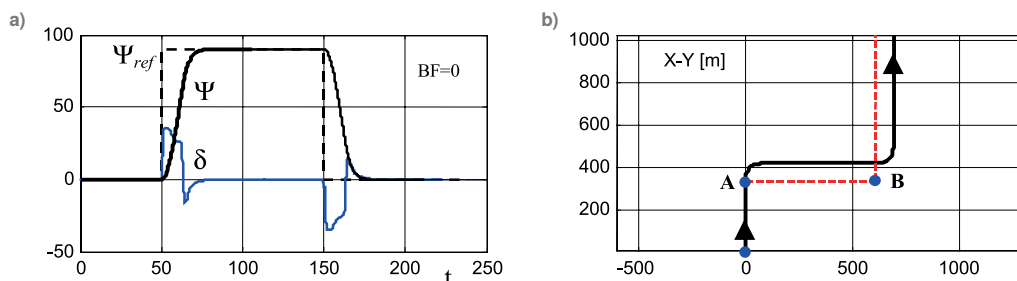
Pamiętajmy, że nastawy regulatora (5) zależą od prędkości statku V . Symulacje wykazały, że ustawienia należy zmodyfikować, jeśli zmiana prędkości przekroczy 20 %.

System zawierający tylko klasyczny regulator PID w odpowiedzi skokowej może wykazywać pewne przeregulowanie ze względu na człon $T_1 s + 1$ w liczniku transmitancji G_{cl} . Aby to ograniczyć, można zastosować filtr wstępny dla wartości zadanej. Transmitancja układu zamkniętego wyniesie wtedy

$$G_{cl}(s) = \frac{1}{(T_{cl} s + 1)^2} = \frac{\psi(s)}{\psi'_{ref}(s)}. \quad (6)$$



Rys. 1. Struktura regulatora kursu w autopilocie
Fig. 1. Structure of heading controller in autopilot



Rys. 2. (a) Położenie steru i kurs statku, (b) Trajektoria X-Y statku (dwie zmiany kursu o 90 stopni, spokojne morze)
Fig. 2. (a) Rudder position and ship heading; (b) X-Y trajectory of the ship (two 90 degree heading changes, calm sea)

2.4. Symulacje

Na rys. 2 pokazano aktualny kurs statku oraz wychylenie steru przy skokowej zmianie wielkości zadanej (kursu) przy aktywnym tylko regulatorze PID_H (ang. *heading control*) oraz spokojnym morzu. W chwili $t = 50$ (punkt A) zmieniono zadany kurs na 90° , a w chwili $t = 150$ (punkt B) powrócono do pierwotnego kursu 0° . Jak widać na rys. 2a, założony kurs został osiągnięty bez przeregulowania (takie było założenie projektowe). Na rys. 2b pokazano trajektorię statku we współrzędnych X-Y (kurs 0° jest pionowo do góry, *North*). W punkcie A zmieniono zadany kurs i statek zakreślił na wschód, a następnie płynął kursem 90° , ale równoległe do ścieżki (odcinka) AB. Podobna sytuacja wystąpiła w punkcie B, gdzie zadany kurs zmieniono z powrotem na 0° (na północ).

Na rys. 3 pokazano wyniki podobnej próby, ale przeprowadzonej przy wzburzonym morzu ($BF = 5$). Początkowo regulator próbuje kompensować ruchami steru wpływ falowania morza (zakłóceń środowiskowych), ale po chwili obserwator stanu (rys. 1) dostroi się do zakłóceń i sterowanie już w mniejszym stopniu kompensuje te zakłócenia.

3. Struktura regulatora trasy

3.1. Regulator trasy

Strukturę autopilota dla kaskadowej regulacji trasy („ścieżki”) pokazano na rys. 4. Jeśli zostanie wykryte odchylenie y pozycji statku od zadanej „ścieżki”, to regulator trasy (główny) koryguje zadany kurs ψ_{ref} o wartość $\Delta\psi_{ref}$, tak więc suma $\psi_{ref} + \Delta\psi_{ref}$ staje się nową wielkością zadaną dla regulatora kursu (pomocniczego) [8, 9]. Równania obejmujące aktualną pozycję statku (np. na podstawie sygnału GPS), prędkość V i kurs ψ (lub odchylenie $\Delta\psi = \psi - \psi_{ref}$) określają obie składowe ruchu (przemieszczenie do przodu x oraz odchylenie prostopadłe do kierunku ruchu y).

PID_T jest regulatorem śledzenia (ang. *track*), PID_H regulatorem kursu, zaś blok V/s przekształca odchyłkę kursu $\psi - \psi_{ref}$ na odchylenie od ścieżki XTE (ang. *Cross Track Error*), zależne od prędkości V [1, 2]. Faktyczne odchylenie XTE wyznaczane jest przez moduł nawigacyjny statku.

3.2. Nastawy

Ponieważ dynamika obiektu sterowanego przez PID_T odpowiada transmitancji integratora o stałej czasowej T_T , tzn.

$$\frac{XTE(s)}{\Delta\psi_T(s)} = \frac{V}{s(T_T s + 1)} \quad (7)$$

i jest tego samego typu, co w (1), więc po zmianie oznaczeń można zastosować wcześniejsze wyniki projektowania dla regulatora kursu PID_H .

Można przyjąć

$$T_{cl,T} = \frac{T_T}{r_T} \quad (8)$$

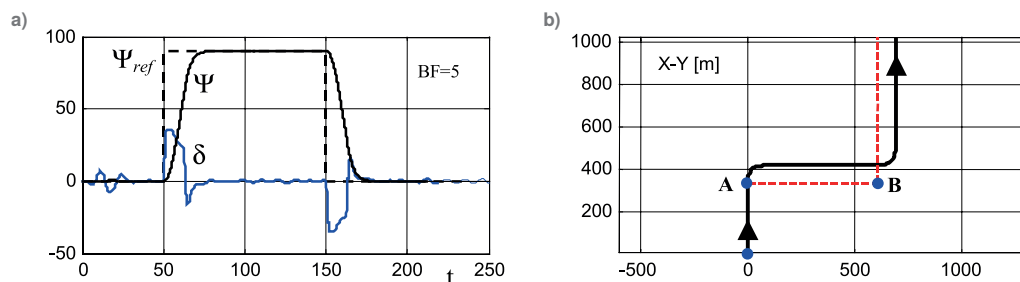
jako stałą czasową kaskadowego układu zamkniętego z odpowiednio dobranym dzielnikiem r_T (jest to kolejny parametr projektowy). Nastawy regulatora PID_T będą wynosić (por. (5)):

$$k_{p,T} = \frac{1}{VT_T} r_T(r_T + 2), \quad T_{i,T} = T_T \frac{r_T + 2}{r_T}, \quad T_{d,T} = T_T \frac{2}{r_T + 2}. \quad (9)$$

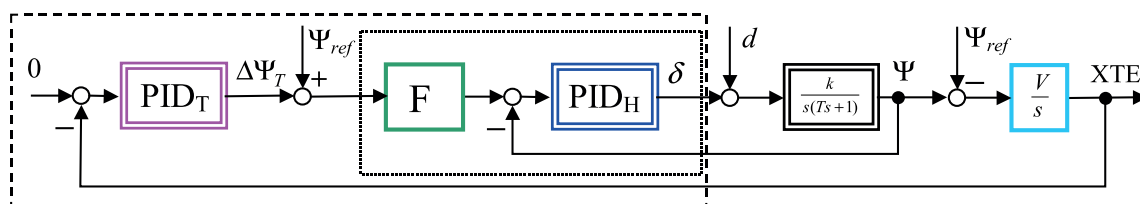
3.3. Symulacje

Na rys. 5 pokazano ponownie aktualny kurs statku oraz wychylenie steru przy skokowej zmianie trasy o 90° , ale teraz aktywny jest cały regulator kaskadowy (tryb *track control*, por. rys. 4). Regulator PID_T odpowiednio modyfikuje wielkość zadaną dla regulatora kursu PID_H tak, aby utrzymać statek dokładnie na ścieżce (a nie równoległe do niej, jak poprzednio).

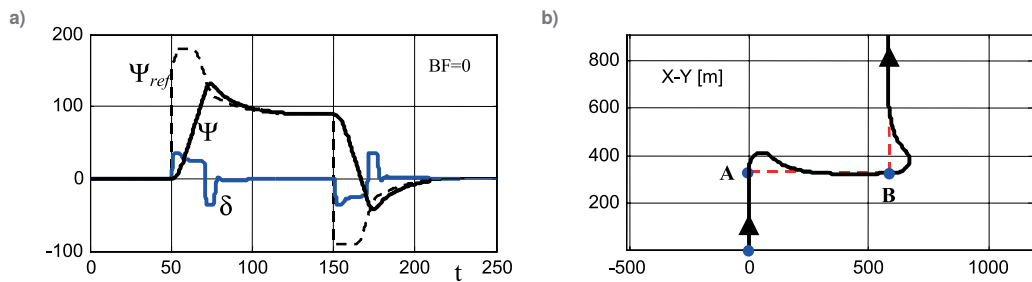
Próba na wzburzonym morzu (rys. 6) wygląda podobnie i dzięki zastosowanej strukturze sterowanie (wychylenie steru) w małym stopniu stara się kompensować szybkozmienne zakłócenia od falowania morza, w celu zmniejszenia zużycia paliwa i mechanizmu sterowego.



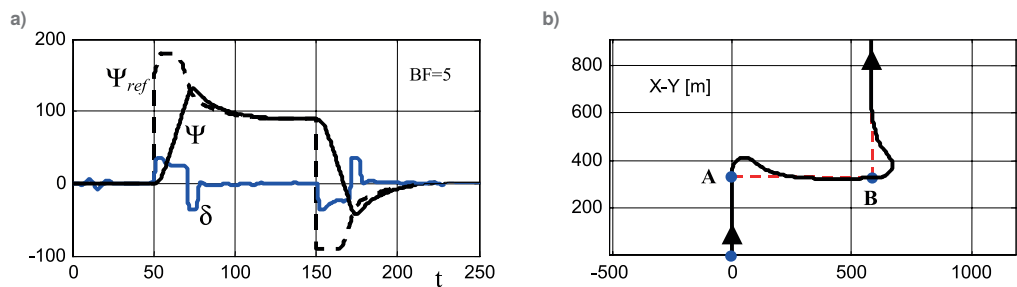
Rys. 3. (a) Położenie steru i kurs statku, (b) Trajektorja X-Y statku (dwie zmiany kursu o 90° , wzburzone morze)
Fig. 3. (a) Rudder position and ship heading; (b) X-Y trajectory of the ship (two changes of heading by 90° , rough sea)



Rys. 4. Struktura regulatora trasy („ścieżki”) w autopilocie
Fig. 4. Structure of the track-keeping controller in autopilot



Rys. 5. (a) Położenie steru i kurs statku, (b) Trajektoria X-Y statku (dwie zmiany kąta ścieżki o 90°, spokojne morze)
 Fig. 5. (a) Rudder position and ship heading; (b) X-Y trajectory of the ship (two changes of path angle by 90°, calm sea)



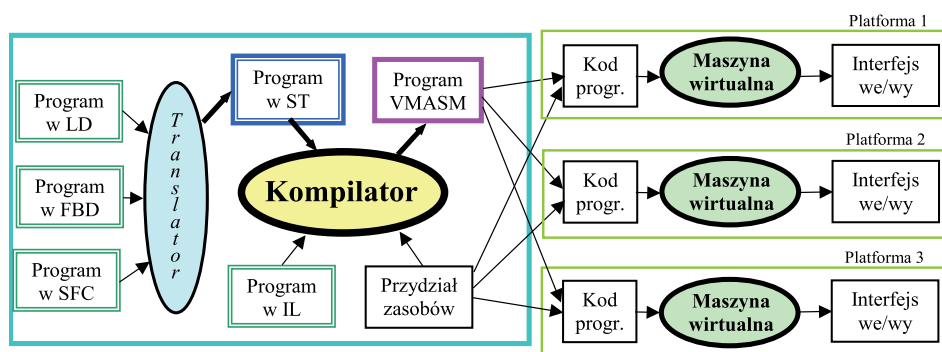
Rys. 6. (a) Położenie steru i kurs statku, (b) Trajektoria X-Y statku (dwie zmiany kąta ścieżki o 90°, wzburzone morze)
 Fig. 6. (a) Rudder position and ship heading; (b) X-Y trajectory of the ship (two changes of path angle by 90°, rough sea)

4. Środowisko inżynierskie CPDev

Prace nad nowoczesnym i uniwersalnym środowiskiem inżynierskim, zgodnym z normą IEC 61131-3, rozpoczęły się już ponad 10 lat temu [10, 11]. Opracowane wtedy środowisko nazwane zostało CPDev (ang. *Control Program Developer*) i od początku wspierało możliwość samodzielnej implementacji na docelowym sprzęcie, zarówno przez inżynierów w przemyśle, jak i studentów na uczelniach. Początkowo wspierało

FBD i SFC są najpierw konwertowane na język ST, a następnie kompilowane do VMASM. Wykonywalny kod binarny programu jest generowany z mnemoników VMASM i ładowany do maszyny wirtualnej. Źródła maszyny wirtualnej napisano w całości w języku C, co znacząco ułatwia przenośność na różne platformy sprzętowe.

Kompilator wykorzystuje zbiór reguł języka ST, biblioteki bloków funkcjonalnych oraz wykaz instrukcji elementarnych realizowanych przez maszynę wirtualną. Oprócz kodu wykonywalnego,



Rys. 7. Etapy tworzenia oprogramowania dla różnych platform sprzętowych
 Fig. 7. Stages of software development for various hardware platforms

jedynie język ST, a w kolejnych latach zostało rozszerzone o obsługę pozostałych języków normy IEC 61131-3, jak również o graficzne projektowanie interfejsu HMI, automatyczny generator dokumentacji oraz moduły o zastosowaniu badawczym. Na rys. 7 przedstawiono kolejne etapy przetwarzania programów sterowania napisanych w różnych językach normy. Języki tekstowe (ST, IL) są tu kompilowane do unikalnych kodów mnemonicznych opracowanych na potrzeby środowiska specjalizowanego języka VMASM. Diagramy graficzne LD,

tworzone są informacje służące do śledzenia, uruchamiania i symulacji programu. Konfiguracja zasobów sprzętowych dotyczy pamięci, interfejsu wejścia/wyjścia oraz interfejsu komunikacyjnego, a specyfikacja obejmuje typy i obszary pamięci, liczby oraz typy wejść i wyjść oraz kanały komunikacyjne. Maszyna wirtualna działa jako interpreter wykonując kolejne linie kodu programu. Interfejs dla docelowej platformy składa się z niskopoziomowych funkcji zależnych od sprzętu i systemu operacyjnego, jak aktualizacja wejść i wyjść czy też zarządzanie czasem cyklu.

4.1. Kod źródłowy

Jak wspomniano, cały kod autopilota został napisany w języku ST i obejmuje łącznie blisko 900 linii programu, z czego prawie 140 przypada na blok PID_H. Fragment kodu dotyczący regulatora kursu przedstawiono poniżej.

```

r_uP := Kp*r_e; (* proportional part P *)
if AM then (*----- AUTO mode ----- *)
  r_uI := r_uI_ + Kp*Tc/Ti*r_e_; (* integral part I *)
  r_uD := (1.0-r_D*Tc/Td)*r_uD+Kp*r_D*(r_e-r_e_); (* part D *)
  r_u:=r_uP+r_uD+r_uI; (* output: P+I+D *)

  (*----- anti-windup limiter ----- *)
  if r_u>1.05*r_uMax then
    r_uI:=r_uI_; (* stop of integration (+) *)
  end_if
  if r_u<-1.05*r_uMax then
    r_uI:=r_uI_; (* stop of integration (-) *)
  end_if

  (*----- output limiter ----- *)
  if r_u>r_uMax then
    r_u:=r_uMax; (* output limit (+) *)
  end_if
  if r_u<-r_uMax then
    r_u:=-r_uMax; (* output limit (-) *)
  end_if

else (*----- MANUAL mode ----- *)
  r_u:=CV_M; (* manual output *)
  r_uI:=CV_M-r_uP; r_uD:=0.0;
end_if

```

Cały kod autopilota jest w pełni przenaszalny, gdyż wszystkie funkcje pomocnicze (jak np. odwracanie macierzy 3×3) zostały napisane od nowa. Nie wykorzystuje on więc żadnych funkcji specyficznych dla danej platformy sprzętowej.

5. Podsumowanie

W artykule przedstawiono strukturę kaskadowego autopilota statku zaimplementowanego w środowisku CPDev oraz zasady analitycznego doboru nastaw, zarówno dla regulatora kursu, jak i trasy (ścieżki). Regulator kursu jest projektowany dla statku opisanego modelem Nomoto, czyli integratora ze stałą czasową. Również w odniesieniu do regulatora śledzenia w układzie kaskadowym, model odchyłań XTE od ścieżki wygląda analogicznie. Dla każdego z nich wymagane są jedynie pojedyncze parametry projektowe do określenia dynamiki w pętli zamkniętej, co znacząco upraszcza proces strojenia i obsługi autopilota statku. Dodatkowo jego zachowanie można dość łatwo dostosować do różnej wielkości statku, stopnia załadunku oraz stanu morza (wiatr, fale). Nastawy są automatycznie dopasowywane do prędkości statku, a wbudowany filtr wstępny i ogranicznik prędkości zadanej dodatkowo eliminuje ewentualne przeregulowania.

Oprogramowanie to jest obecnie testowane w prototypowym autopilocie rozwijanym we współpracy z firmą Praxis Automation Technology B.V. z Holandii [15], który to będzie wdrażany na małych i średnich jednostkach pływających.

Podziękowania

Autor dziękuje współtwórcom oprogramowania autopilota oraz całego środowiska CPDev, a w szczególności Leszkowi Trybusowi, Bartoszowi Trybusowi, Janowi Sadolewskiemu, Dariuszowi Rzońca, Andrzejowi Stecowi oraz Marcinowi Jamro. Projekt finansowany w ramach programu Ministra Edukacji i Nauki pod nazwą „Regionalna Inicjatywa Doskonałości” w latach 2019–2023, nr projektu 027/RID/2018/19, kwota finansowania 11 999 900 zł.

Bibliografia

1. Fossen T.I., *Guidance and Control of Ocean Vehicles* (4th ed.), Wiley, Chichester 1999.
2. Fossen T.I., *Marine Control Systems*, Marine Cybernetics, Trondheim 2002.
3. Lisowski J., *Statek jako obiekt sterowania automatycznego*, Wydawnictwo Morskie, Gdańsk 1981.
4. Morawski L., Pomirski J., Rak A., *A cascade control system for trajectory tracking*, 4th International Congress on Maritime Technological Innovations and Research, 2004, 265–276.
5. Chen C.Y., Ruiz M.T., Delefortrie G., Vantorre M., Lataire E., *An Energy-Efficient Adaptive Course Control System for Ocean Surface Ships*, 11th International Workshop on Ship and Marine Hydrodynamics, Hamburg, 2019, DOI: 10.15480/882.3349.
6. Tomera M., *Fuzzy Self-tuning PID Controller for a Ship Autopilot*, Proceedings of the 12th International Conference on Marine Navigation and Safety of sea transportation, TransNav, Gdynia, 2017, DOI: 10.1201/9781315099132-15.
7. van Amerongen J., van Nauta Lemke H.R., van der Veen J.C.T., *An autopilot for ships designed with fuzzy sets*, “IFAC Proceedings Volumes”, Vol. 10, No. 16, 1977, 479–487, DOI: 10.1016/S1474-6670(17)69559-4.
8. Świder Z., *Implementation of the ship’s autopilot in the CPDev environment*, [In:] Szewczyk R., Zieliński C., Kaliczyńska M. (eds): *Automation 2022: New Solutions and Technologies for Automation, Robotics and Measurement techniques*, Vol. 1427, 2022, 133–142, DOI: 10.1007/978-3-031-03502-9_14.
9. Trybus L., Świder Z., Stec A., *Tuning Rules of Conventional and Advanced Ship Autopilot Controllers*, [In:] *Automation 2015*, 303–311, DOI: 10.1007/978-3-319-15796-2_31.

10. Jamro M., Rzońca D., Sadolewki J., Stec A., Świder Z., Trybus B., Trybus L., *CPDev Engineering Environment for Modeling, Implementation, Testing, and Visualization of Control Software*. [In:] Szewczyk R., Zieliński C., Kaliczyńska M. (eds): Recent Advances in Automation, Robotics and Measuring Techniques. Vol. 267, 2014, 81–90, DOI: 10.1007/978-3-319-05353-0_9.
11. Rzońca D., Sadolewski J., Stec A., Świder Z., Trybus B., Trybus L., *Mini-DCS system programming in IEC 61131-3 Structured Text*, “Journal of Automation, Mobile Robotics and Intelligent Systems”, Vol. 2, No. 3, 2008, 48–54.

Inne źródła

12. *ALPHASEAPILOT MFC Autopilot Operating Manual*, Alpatron, <https://www.alpatronmarine.com>
13. *FAP-2000 Autopilot Operator Manual*, Furuno, <https://www.furuno.com>
14. *Simrad AP70/80 Operator Manual*, Simrad, <https://rowlandsmarine.co.uk>
15. Praxis Automation Technology B.V., <http://www.praxis-automation.nl>

A Prototype of a Cascade Ship Autopilot Implemented in the CPDev Environment

Abstract: Ship autopilots can be divided into conventional, only capable of maintaining a given heading, and advanced, which can additionally follow a „track” connecting the given navigation points along the ship’s route. The article presents the structure of the prototype autopilot of the ship implemented in the CPDev environment and the formulas allowing to determine the settings of the course controller (PID) and track controller (PI) in the cascade control. For each of them, individual design parameters were adopted to define the dynamics of the closed control loop. These rules were applied in the software of the autopilot prototype, created in cooperation with a Dutch company designing control and visualization systems for ships.

Keywords: cascade system, PID controller, ship autopilot, settings, engineering environment

dr hab. inż. Zbigniew Świder

swiderzb@prz.edu.pl

ORCID: 0000-0003-3504-5340

Absolwent Wydziału Elektrotechniki i Informatyki Politechniki Rzeszowskiej (1984). Na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej w Gliwicach uzyskał stopień doktora nauk technicznych (1992) oraz doktora habilitowanego nauk technicznych (2004). Od początku pracy zawodowej zajmuje się sterownikami mikroprocesorowymi, a ostatnio metodami samostrojzenia i adaptacji nastaw regulatorów, komputerowymi systemami automatyki oraz środowiskami inżynierskimi do programowania przemysłowych układów sterowania.

