

Prototyp zaawansowanego 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 śledzić „ścieżkę” łączącą zadane punkty nawigacyjne na trasie statku. W artykule przedstawiono strukturę prototypowego autopilota statku zaimplementowanego w środowisku CPDev oraz wzory pozwalające wyznaczyć nastawy regulatora kursu (PID) oraz trasy (PI) 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: autopilot statku, regulator PID, zasady strojenia, obserwator stanu, środowisko inżynierskie

1. Wprowadzenie

Sterowanie kursem statku jest realizowane zwykle przez konwencjonalne 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, a także, oprócz utrzymywania zadanego przez operatora kursu, jest w stanie śledzić „ścieżkę” łączącą kolejne punkty nawigacyjne na zaplanowanej trasie. Taki system wymaga modułu nawigacji obszarowej, który oblicza każde odchylenie od zaprogramowanej trasy, spowodowane np. oddziaływaniem wiatru i prądów morskich, a następnie automatycznie wprowadza konieczne poprawki korygujące kurs statku. Firmy o ugruntowanej pozycji, takie jak np. Sperry Marine, Raytheon, Kongsberg i wiele innych, oferują zarówno konwencjonalne, jak i bardzo zaawansowane autopiloty, w tym z funkcjami adaptacyjnymi. Literatura na temat autopilotów dla statków dotyczy w większości złożonych algorytmów opartych na modelu [1, 11]. Druga grupa obejmuje przykładowo algorytmy genetyczne [9], regulatory rozmyte [12] lub sieci neuronowe [13].

Prace nad uniwersalnym środowiskiem inżynierskim, zgodnym z normą IEC 61131-3, rozpoczęły się już ponad 10 lat temu [7]. Opracowane wtedy środowisko nazwane zostało CPDev (ang. *Control Program Developer*) i od początku wspierało możliwość samodzielnej implementacji na docelowy

wym sprzęcie, zarówno przez inżynierów w przemyśle, jak i studentów na uczelniach. W kolejnych latach do pakietu dodano rozszerzenia dla 32- i 64-bitowych procesorów, uzupełniono je o translatory języków graficznych LD, FBD, SFC, narzędzie do projektowania interfejsów HMI i wieloprojektowe środowisko uruchomieniowe. Środowisko CPDev zostało też wdrożone i jest nadal stosowane przez kilku znaczących producentów urządzeń automatyki w kraju oraz za granicą.

Politechnika Rzeszowska od kilku lat współpracuje z firmą Praxis Automation Technology B.V. z Holandii przy wdrożeniu pakietu CPDev [6] do tworzenia systemów monitoringu i nawigacji statków, rozpoczęto również wspólne prace nad prototypem autopilota [8], z Rzeszowem odpowiedzialnym za jego implementację. Jednym z istotnych problemów były zasady strojenia poszczególnych bloków PID autopilota, ponieważ klasyczne podręczniki [3–5] nie są w tym zakresie wystarczająco szczegółowe. Ze względów praktycznych poszukiwano takich reguł, które oprócz wymagań braku oscylacji i przeregulowania, byłyby w stanie określić dynamikę kursu za pomocą zrozumiałych dla operatora parametrów projektowych. Dynamika ta jednak mocno zależy od konstrukcji statku oraz stanu morza (wiatr, fale).

W artykule przedstawiono podstawy tworzenia takich reguł dla regulatora kursu (blok klasycznego regulatora PID) oraz regulatora „ścieżki” (blok PI lub regulator w przestrzeni stanów). Prototyp autopilota został przetestowany na pełnym (nieliniowym) modelu kontenerowca 4-DoF [14], zaimplementowanym również w pakiecie CPDev, a wybrane wyniki z badań symulacyjnych przedstawiono w dalszej części.

Autor korespondujący:

Zbigniew Świder, swiderzb@prz.edu.pl

Artykuł recenzowany

nadesłany 29.11.2020 r., przyjęty do druku 28.01.2020 r.



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

2. Struktura autopilota kursu

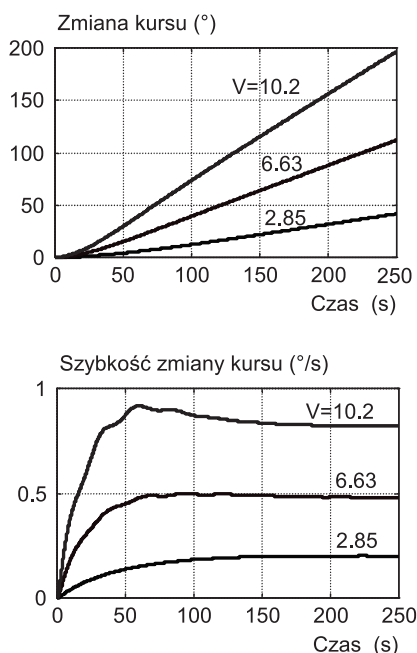
Model statku. Pełny (nieliniowy) model statku 4-DoF jest opisany równaniami w przestrzeni stanów [14], jednak jest on zbyt złożony do określenia analitycznych wyrażeń dla nastaw poszczególnych regulatorów. Zwykle do takiego projektowania stosuje się uproszczony model statku, zwany modelem Nomoto

[3–5], w postaci połączenia członu całkującego z członem inercyjnym (1), który pokazuje, jak wychylenie steru δ wpływa na kąt ψ (kurs), pod jakim statek płynie.

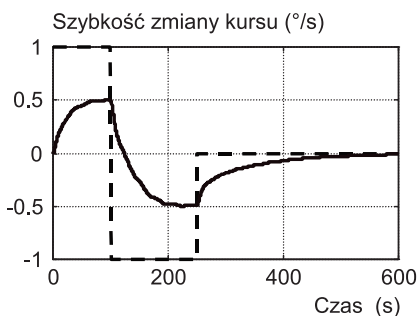
Wzmocnienie k zależy wprost proporcjonalnie, a stała czasowa T odwrotnie proporcjonalnie od prędkości statku V , czyli

$$\frac{k}{s(Ts + 1)} = \frac{\psi(s)}{\delta(s)} \quad (1)$$

gdzie $k = k_0 \frac{V}{V_0}$, $T = T_0 \frac{V_0}{V}$



Rys. 1. Zmiana kursu przy stałym wychyleniu steru
Fig. 1. Change course with constant rudder deflection



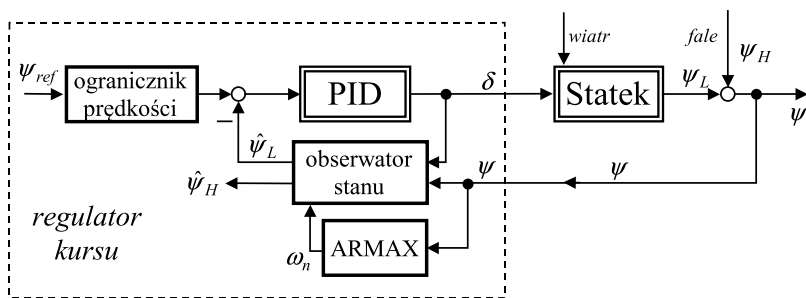
V_0 [m/s]	k_0 [1/s]	T_0 [s]
2.85 0	.020 4	1
6.63 0	.048 1	9
10.2	0.082 1	2

Rys. 2. Przykładowe wartości k_0 , T_0 dla różnych prędkości V_0
Fig. 2. Exemplary values of k_0 , T_0 for different speeds V_0

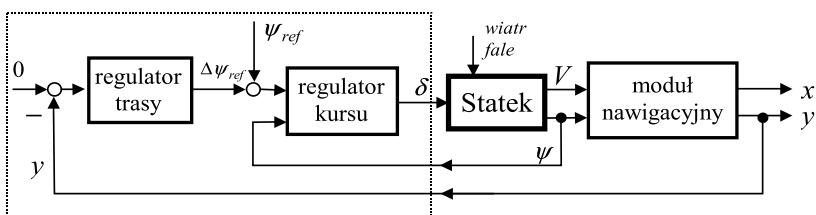
Przyjmuje się, że wartości referencyjne $\{k_0, T_0, V_0\}$ występujące we wzorach (1) są znane z wcześniejszych manewrów wykonanych podczas prób na otwartym morzu lub też bezpośrednio po wyjściu statku z portu. Tutaj wartości k_0, T_0 obejmują również dynamikę maszyny sterowej i samego steru (rys. 2).

Regulator kursu. Strukturę prototypowego regulatora kursu statku, będącego modyfikacją regulatorów przedstawianych w [3, 4], pokazano na rys. 3. Wielkości ψ_{ref} i ψ to odpowiednio wartość zadana i kurs rzeczywisty statku, natomiast δ oznacza kąt wychylenia steru. Wielkości ψ_L, ψ_H są odpowiednio składowymi nisko- i wysokoczęstotliwościowymi, przy czym te ostatnie są indukowane głównie przez falowanie morza. 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. Eliminacja składowych wysokoczęstotliwościowych redukuje znacznie reakcję steru na wpływ falowania morza. Obserwator stanu może być zaimplementowany jako filtr Kalmana lub też jako obserwator Luenbergera. Dodatkowy blok ARMAX estymuje nieznaną częstotliwość fal ω_n , która jest jednym ze współczynników w równaniach obserwatora. Należy pamiętać, że częstotliwość ta zależy od stanu morza oraz kierunku wiatru (w stosunku do kursu statku). Dodatkowo, ogranicznik prędkości eliminuje ewentualne przeregulowania po skokowej zmianie ψ_{ref} . Ustawienia regulatora PID są automatycznie dostosowane do prędkości statku V .

Regulator trasy. Strukturę części autopilota dla kaskadowej regulacji trasy („ścieżki”) przedstawiono na rysunku 4. Jeśli zostanie wykryte odchylenie y pozycji statku od zadanej „ścieżki”, to regulator trasy (główny) koryguje wartość zadaną kursu ψ_{ref} o $\Delta\psi_{ref}$ tak więc suma $\psi_{ref} + \Delta\psi_{ref}$ staje się nową wartością zadaną dla regulatora kursu (pomocniczego). Regulator trasy (główny) może być zaimplementowany jako regulator w przestrzeni stanów lub klasyczny regulator PI. Równania kinematyczne 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).



Rys. 3. Struktura regulatora kursu w autopilocie
Fig. 3. Structure of course controller in autopilot



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

3. Dobór nastaw dla regulatora kursu

Sterowanie kursem odbywa się za pomocą klasycznego regulatora PID (rys. 3)

$$k_p(1 + \frac{1}{T_i}s + T_d s) = k_p \frac{(T_1s+1)(T_2s+1)}{T_s} \quad (2a)$$

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

Założenia do projektowania. Dla obliczeń nastaw regulatora przyjęto:

- (a) brak oscylacji w odpowiedziach przejściowych
- (b) stała czasowa zamkniętej pętli jest określona jako $T_{cl} = T/r, r > 1$
- (c) brak przeregulowania dla skokowych zmian wartości zadanej ψ_{ref}

Warunek (b) wymaga, aby stała czasowa T_{cl} była r razy mniejsza niż stała czasowa T w modelu statku (1). Współczynnik r jest tu parametrem projektowym. Zwiększając r można sprawić, że pętla będzie „szybsza”. Aby określić nastawy k_p, T_i, T_d regulatora PID przyjmijmy $T_2 = T$. Wtedy transmitancje układu otwartego i zamkniętego, oraz wzmocnienie K , przyjmują postać:

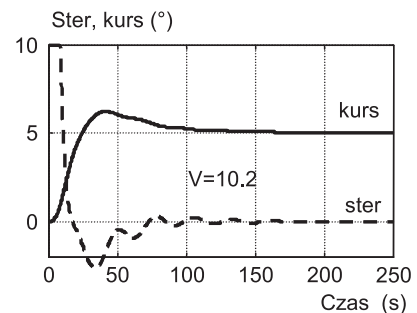
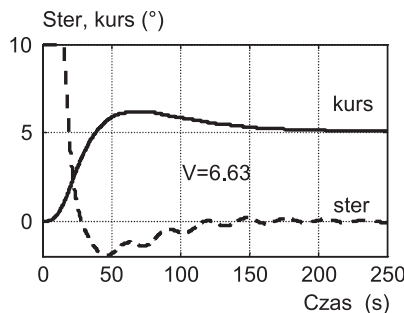
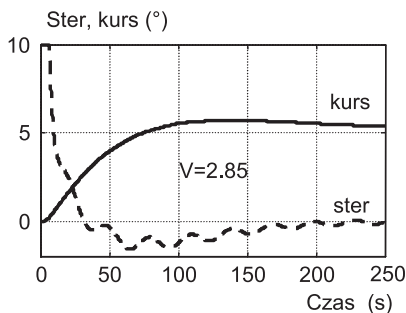
$$G_{open}(s) = K \frac{T_1s+1}{s^2},$$

$$G_{loop}(s) = \frac{K(T_1s+1)}{s^2 + K T_1s + K}, \quad K = \frac{k_p k}{T_i} \quad (3)$$

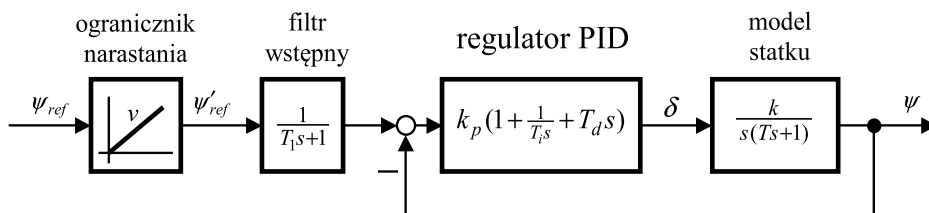
Założenie (a) będzie spełnione, jeśli wyznacznik mianownika transmitancji G_{loop} wyniesie zero, co daje warunek na K oraz określa wartości podwójnego bieguna $s_{1,2}$:

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

Założenie (b) wymaga, aby stała czasowa układu zamkniętego $T_{cl} = 1/|s_{1,2}| = T_1/2$ była równa T/r , tak więc stała czasowa T_1 będzie wynosić



Rys. 5. Odpowiedzi skokowe dla regulatora PID dla różnych prędkości V
Fig. 5. Step responses for PID controller at different speeds V



Rys. 6. Schemat regulatora kursu dla projektowania nastaw
Fig. 6. Course control system for controller design

$$T_1 = \frac{2T}{r} \quad (5)$$

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

$$k_p = \frac{1}{kT} r(r+2), \quad T_i = T \frac{r+2}{r}, \quad T_d = T \frac{2}{r+2} \quad (6)$$

Zwróćmy uwagę, iż ze względu na równanie (1), nastawy regulatora (6) zależą od prędkości statku V . Testy czułości wykazały, że ustawienia należy zmodyfikować, jeśli zmiana prędkości przekroczy 20 %. Odpowiedzi skokowe dla regulatora PID przy nastawach (6) dla różnych prędkości V pokazano na rys. 5.

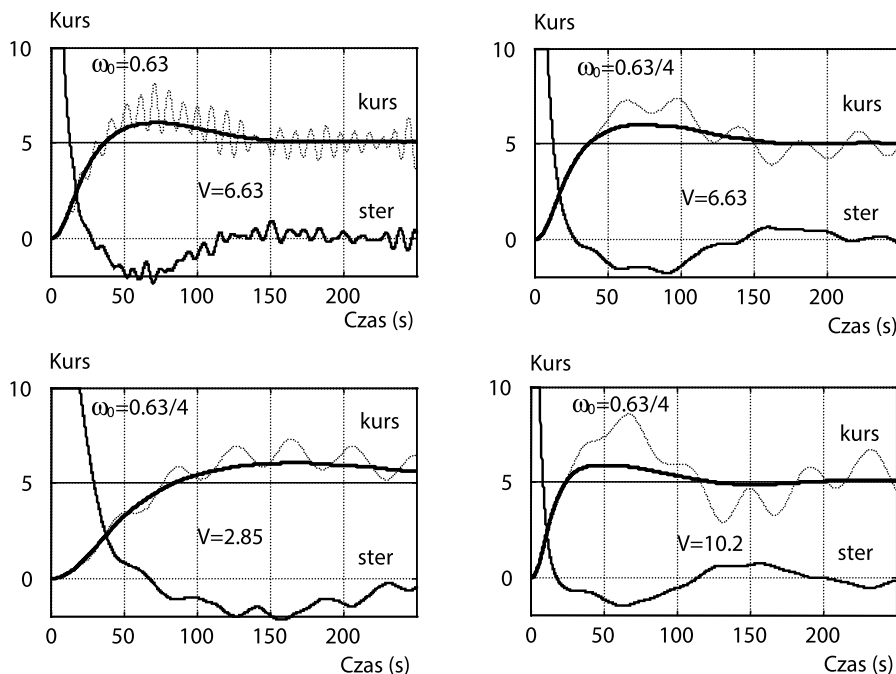
Filtr wstępny. System zawierający tylko klasyczny regulator PID będzie wykazywał pewne przeregulowanie ze względu na człon $T_1s + 1$ w liczniku transmitancji G_{loop} . Aby to ograniczyć lub wyeliminować, można zastosować filtr wstępny dla wartości zadanej (rys. 6). Transmitancja układu zamkniętego (bez ogranicznika prędkości narastania) wyniesie wtedy

$$G_{cl}(s) = \frac{1}{(T_{cl}s+1)^2} = \frac{\psi(s)}{\psi'_{ref}(s)} \quad (7)$$

Ogranicznik narastania. Ponieważ w praktyce wychylenie steru δ nie może przekroczyć pewnej maksymalnej granicy δ_M , czyli $|\delta| < \delta_M$, więc zwykle filtr wstępny $1/(T_1s + 1)$ nie wystarcza, aby uniknąć przeregulowania przy skokowych zmianach ψ'_{ref} (typowo $\max \delta_M = 35^\circ$). Dodatkowo, wartość δ_M może zależeć od prędkości statku V . Autopilot jest więc wyposażony w ogranicznik narastania wartości zadanej, który to ogranicza prędkość zmian ψ'_{ref} do założonej wartości v . Dla $\psi'_{ref} = v \cdot t$ wyrażenie określające kąt wychylenia steru przyjmie postać

$$\delta(s) = \frac{1}{k} \frac{s(Ts+1)}{(T_{cl}s+1)} \frac{v}{s^2} = \frac{1}{k} \frac{Ts+1}{(T_{cl}s+1)^2} \frac{v}{s}, \quad (8)$$

którego to postać czasowa ma maksimum dla $T_{cl} < T$ ($r > 1$).



Rys. 7. Odpowiedzi skokowe regulatora kursu przy falowaniu morza
Fig. 7. Step responses of course controller in the presence of sea waves

4. Dobór nastaw dla regulatora trasy

Załóżmy, że mamy zadaną trasę („ścieżkę” odniesienia) między dwoma punktami nawigacyjnymi, po której to ma podążać statek. Niech ψ_{ref} oznacza założony kurs na tej ścieżce. System nawigacji obszarowej jest w stanie określić prostopadłą składową y odchylenia od zadanej ścieżki (rys. 2). Jak pokazano w [3], dla małych odchyłek kursu odchylenie to jest całką ważoną $\Delta\psi$, czyli

$$y(s) = \frac{V}{s} \Delta\psi(s) \tag{9}$$

($\Delta\psi$ w radianach). Stosując standardowy regulator PI (jako główny w regulacji kaskadowej), rozważmy układ pokazany na rys. 8.

Specyfikacje są tu następujące:

- (a) brak oscylacji w stanach przejściowych dla odpowiedzi skokowej,
- (b) krótki czas narastania odpowiedzi skokowej (w miarę możliwości).

Problem doboru nastaw regulatora PI można rozwiązać projektując je metodą linii pierwiastkowych [2, 10]. Te dwa wymagania zostaną spełnione, jeśli znajdziemy takie zero $1/T_i$ sterownika, dla którego układ z zamkniętą pętlą ma potrójny biegun rzeczywisty. Wtedy wzmacnienie regulatora, stała całkowania oraz czas narastania odpowiedzi skokowej określone są wzorami

$$T_i = 15 T_{cl}, k_p = 0.2 \frac{1}{T_{cl}V}, t_{tr} \cong \frac{4}{|s_{tr}|} = 25 T_{cl} \tag{10}$$

Należy zwrócić uwagę, że taki czas ustalania jest ponad 6 razy dłuższy, niż czas ustalania dla regulatora kursu ($4T_{cl}$). Poza tym, dla typowych danych wzmacnienie k_p okazuje się małe, a stała czasowa całkowania T_i duża. Tak więc, jak można się spodziewać po [2], mamy regulator PI o stosunkowo niskim wzmacnieniu.

Jeśli powolna dynamika regulatora „ścieżki” z wykorzystaniem PI nie jest zadowalająca, rozwiązaniem może być regulator w przestrzeni stanów (ang. *state-space controller*). Odchylenie y , odchylenie kursu $\Delta\psi$ i jego pochodna $\Delta\dot{\psi}$ są zmiennymi stanu. Aby skompensować zakłócenia w stanie ustalonym, musimy dodać całkę z odchylenia y jako czwartą zmienną stanu. Zatem wektor stanu składa się z następujących składowych

$$x_1 = \int y dt, \quad x_1 = y, \quad x_2 = \psi, \quad x_3 = \Delta\dot{\psi}. \tag{11}$$

Wyjście regulatora stanu wynosi wtedy

$$\Delta\psi_{ref} = -(K_I x_1 + K_1 x_1 + K_2 x_2 + K_3 x_3). \tag{12}$$

Tworzenie opisu w przestrzeni stanów $\{A, b, c\}$ dla „obiektu” składającego się z $1/(T_{cl}s + 1)^2$ oraz V/s jest proste, dlatego tutaj je pominiemy. Aby szybko uzyskać wyniki analityczne, założymy, że warunek dla czasu ustalania ma postać poczwórnego bieguna dla zamkniętej pętli sprzężenia

$$p = \frac{\sigma}{T_{cl}} \tag{13}$$

w odniesieniu do stałej czasowej T_{cl} , z pewnym przyjętym parametrem projektowym σ . Wybierając $\sigma = 1$ można sprawić, że

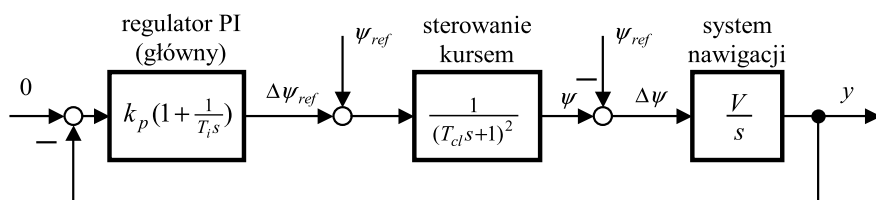


Fig. 8. System śledzenia trasy ze standardowym regulatorem PI
Fig. 8. Track-keeping system with standard PI controller

czas narastania odpowiedzi dla regulatora śledzenia będzie porównywalny z czasem narastania dla regulatora kursu ($4T_{cl}$). Może to być jednak zbyt ambitne, więc przy projektowaniu zalecane są wartości $\sigma \leq 0,5$. Mając opis w przestrzeni stanów $\{A, b, c\}$ i równanie charakterystyczne dla pętli zamkniętej $(s + p)^4$ z biegunem $p = \sigma/T_{cl}$ poszczególne wzmocnienia dla regulatora w przestrzeni stanów wynoszą

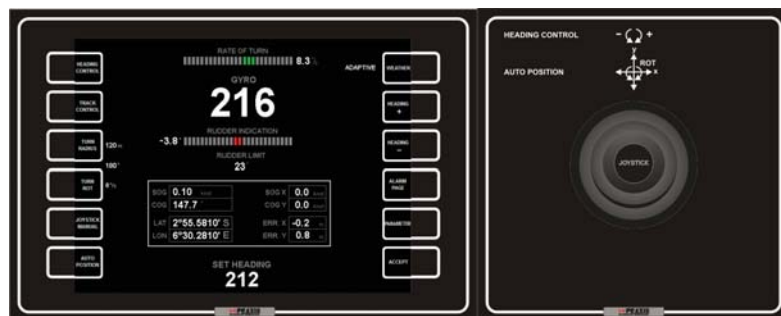
$$K_I = \frac{\sigma^4}{T_{cl}^2} \frac{1}{V}, \quad K_1 = 4 \frac{\sigma^3}{T_{cl}} \frac{1}{V},$$

$$K_2 = 6\sigma^2 - 1, \quad K_3 = 2T_{cl}(2\sigma - 1) \quad (14)$$

5. Podsumowanie

W artykule przedstawiono strukturę prototypowego autopilota statku oraz zasady analitycznego doboru nastaw dla regulatora kursu (PID) oraz trasy (PI lub SS). Dla każdego z nich wymagane są jedynie pojedyncze parametry projektowe do określenia dynamiki w pętli zamkniętej, co znacząco upraszcza proces strojenia autopilota. Dodatkowo zachowanie autopilota można dość łatwo dostosować do różnej konstrukcji statku oraz stanu morza (wiatr, fale). Nastawy są automatycznie dopasowywane do prędkości statku, a wbudowany filtr wstępny i ogranicznik prędkości zadanej dodatkowo eliminuje ewentualne przeregulowania.

Przedstawione w artykule reguły doboru nastaw zastosowano, z pewnymi modyfikacjami w celu zwiększenia ich odporności, w oprogramowaniu autopilota zaimplementowanym w pakiecie CPDev [8]. Ekran główny interfejsu (panel HCS) pokazano na rys. 9. Dostępne są różne tryby pracy autopilota, między innymi: Heading Control (stabilizacja kursu), Track Control (podążanie po „ścieżce”), Turn Radius (zakręt o stałym promieniu) czy też Turn ROT (zakręt o stałej szybkości).



Rys. 9. Ekran główny w autopilocie firmy Praxis (pakiet CPDev)
Fig. 9. Main screen of the Praxis ship autopilot (CPDev package)

Oprogramowanie autopilota będzie rozbudowywane o adaptacyjne dostrajanie nastaw (z użyciem regulatora rozmytego lub sieci neuronowej) oraz nowe algorytmy identyfikacji zakłóceń pochodzących od wiatru i falowania morza.

Podziękowania

Autor dziękuje współtwórcom oprogramowania autopilota oraz całego środowiska CPDev, a w szczególności Leszkowi Trybusowi, Bartoszowi Trybusowi, Janowi Sadolewskiemu, Dariuszowi Rzońca oraz Marcinowi Jamro.

Bibliografia

1. Cimen T., Banks S.P., *Nonlinear optimal tracking control with application to super-tankers for autopilot design*. "Automatica", Vol. 40, No. 11, 2004, 1845–1863, DOI: 10.1016/j.automatica.2004.05.015.
2. Dorf R.C., Bishop R.M., *Modern Control Systems* (11th ed.). Prentice Hall, Upper Saddle River, NY 2008.
3. Fossen T.I., *Guidance and Control of Ocean Vehicles* (4th ed.), Wiley, Chichester, 1999.
4. Fossen T.I., *Marine Control Systems. Marine Cybernetics*, Trondheim 2002.
5. Lisowski J., *Statek jako obiekt sterowania automatycznego*, Wydawnictwo Morskie, Gdańsk 1981.
6. Jamro M., Rzońca D., Sadolewski 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, sec. *Advances in Intelligent Systems and Computing*. Springer, 2014, 81–90.
7. 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.
8. Rzońca D., Sadolewski J., Stec A., Świder Z., Trybus B., Trybus L., *Ship Autopilot Software – A Case Study*, in: Bartoszewicz A., Kabziński J., Kacprzyk J. (eds.): *Advanced, Contemporary Control*. *Advances in Intelligent Systems and Computing*, Vol. 1196, Springer, Cham 2020, 1499–1506.
9. McCookin E.W., Murray-Smith D.J., Li Y., Fossen T.L., *Ship steering control system optimization using genetic algorithms*. "Control Engineering Practice", Vol. 8, 2000, 429–443.
10. Trybus L., *Teoria sterowania*. Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów 2007.
11. Tzeng C.Y., Lu G.H., *An Internal Model Control-based Neural Network ship autopilot design*. "Journal of the Society of Naval Architects and Marine Engineers of the ROC", 22(1), 2003, 13–23.
12. Vanec T.W., *Fuzzy guidance controller for an autonomous boat*. "IEEE Control Systems Magazine", Vol. 17, No. 2, 1997, 43–51, DOI: 10.1109/37.581294.
13. Zhang Y., Hearn G.E., Sen P., *A multivariable neural controller for automatic ship berthing*. "IEEE Control Systems Magazine", Vol. 17, No. 4, 1997, 31–45, DOI: 10.1109/37.608535.
14. <http://www.marinecontrol.org/Tutorial.html>.

A prototype of an advanced ship autopilot implemented in the CPDev environment

Abstract: Ship autopilots can be divided into conventional, only capable of maintaining a given course, 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: ship autopilot, PID controller, tuning rules, state observer, 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 samostrojenia i adaptacji nastaw regulatorów, komputerowymi systemami automatyki oraz środowiskami inżynierskimi do programowania przemysłowych układów sterowania.