Centrum Rzeczoznawstwa Budowlanego

Open Access Journal
Modern Engineering 1 (2017) 74-80
journal homepage: http://mengineering.eu

Modern Engineering

Projekt architektury programowej autonomicznego biomimetycznego pojazdu podwodnego przeznaczonego do zadań rozpoznania podwodnego
The project architecture of the autonomous biomimetic underwater vehicle designed for underwater reconnaissance tasks

Tomasz PRACZYK* [1] and Piotr SZYMAK[1]

[1]Akademia Marynarki Wojennej, Gdynia, Polska

Streszczenie

Artykuł przedstawia projekt architektury programowej autonomicznego biomimetycznego pojazdu podwodnego (ABPP) przeznaczonego do zadań rozpoznania podwodnego. Pojazd ten jest przedmiotem badań w ramach projektu finansowanego przez Narodowe Centrum Badań i Rozwoju. Dwie cechy czynią ten pojazd unikatowym w warunkach polskich, są nimi, po pierwsze – autonomiczność, a po drugie – biomimetyczność czyli podobieństwo do żywych przedstawicieli środowiska morskiego. Całkowita nowość technologii opracowywanej w ramach projektu czyni konstrukcję pojazdu w tym jego oprogramowania dużym wyzwaniem. Proponowana w artykule architektura oprogramowania zakłada jego elastyczność oraz skalowalność.

Słowa kluczowe: muzeum, Kraków, oryginalne rozwiązania

Abstract

The paper presents the project of software architecture for autonomous biomimetic underwater vehicle dedicated for underwater reconnaissance tasks. The vehicle is build within the framework of the national project funded by National Center for Research and Development. Two features make this vehicle unique in Poland, that is, autonomy, and resemblance to living organisms of sea environment. The project presented in the paper provides for easy expansion and modification of the vehicle control system with new sensors and autonomous behaviors.

Keywords: underwater vehicle, recognition

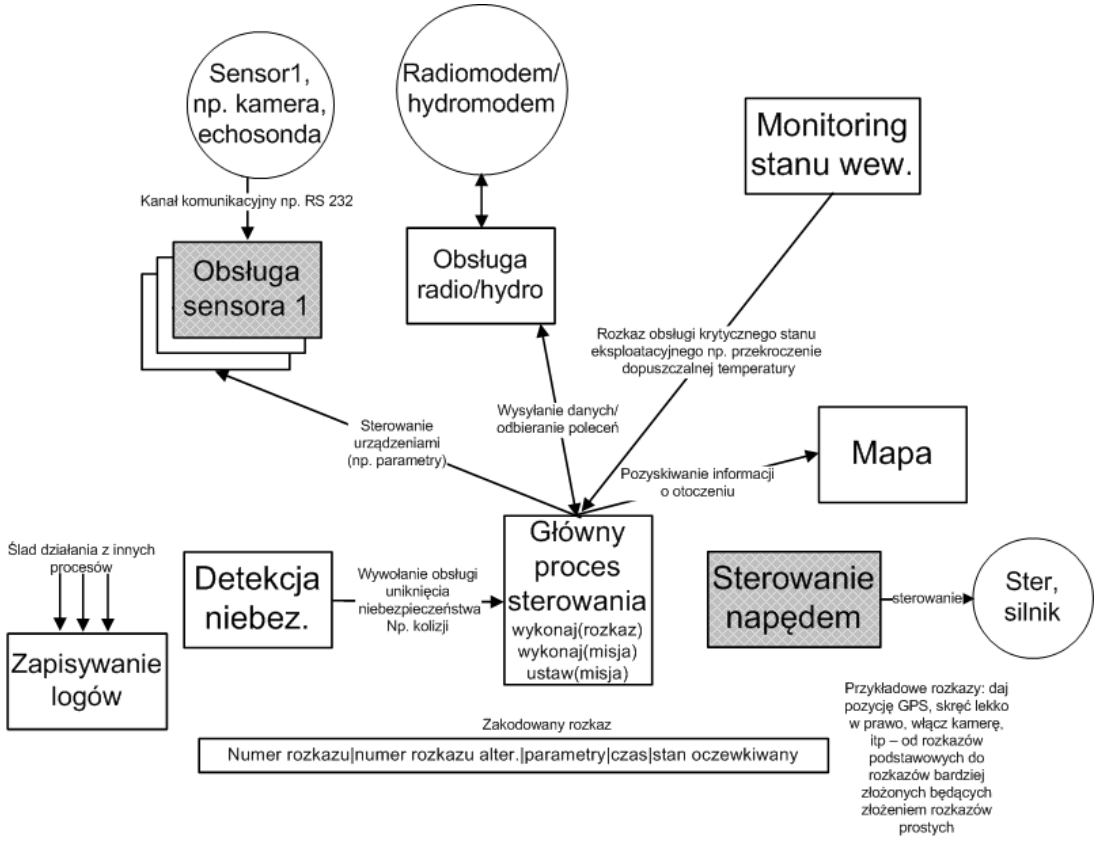
1 WPROWADZENIE

W Polsce powstały do tej pory różne konstrukcje robotów podwodnych jednak były to zazwyczaj konstrukcje typu ROV czyli Remotely Operated Vehicle (pojazd zdalnie sterowany). Pojazdy autonomiczne zgodnie z wiedzą autorów nie były dotąd budowane w naszym kraju. Pierwszym krokiem w tym zakresie jest projekt finansowany przez Narodowe Centrum Badań i Rozwoju którego celem jest budowa autonomicznego biomimetycznego pojazdu podwodnego do celów rozpoznania podwodnego.

Biomimetyczne pojazdy podwodne [2–4] to pojazdy imitujące swoją konstrukcją i zachowaniem żywe organizmy zamieszkujące środowisko wodne. Pojazdy te stanowią na świecie młodą, mało dojrzałą jednak bardzo obiecującą gałąź robotyki podwodnej. Zadania które mogą być realizowane przez takie pojazdy najczęściej związane są z neutralnym monitoringiem środowiska morskiego oraz badaniami naukowymi zakładającymi minimalizację ingerencji w życie mieszkańców tego środowiska. Jednym z zastosowań pojazdów biomimetycznych jest również rozpoznanie podwodne. Z uwagi na fakt iż konstrukcja pojazdów biomimetycznych upodabnia je do żywych organizmów morskich, pojazdy te są idealnym narzędziem rozpoznania podwodnego na terenie uniemożliwiającym swobodne poruszanie się i działanie.

*Corresponding author: E-mail address: t.praczyk@amw.gdynia.pl (Tomasz PRACZYK)

Received 11 September 2017
Available online 15 December 2017
ISSN 2450-5501
Published by Centrum Rzeczoznawstwa Budowlanego



Rysunek 1. Diagram przepływu komunikatów sterujących pomiędzy procesami w systemie informatycznym ABPP (formą zbliżony do diagramów komunikacji (kolaboracji) UML)

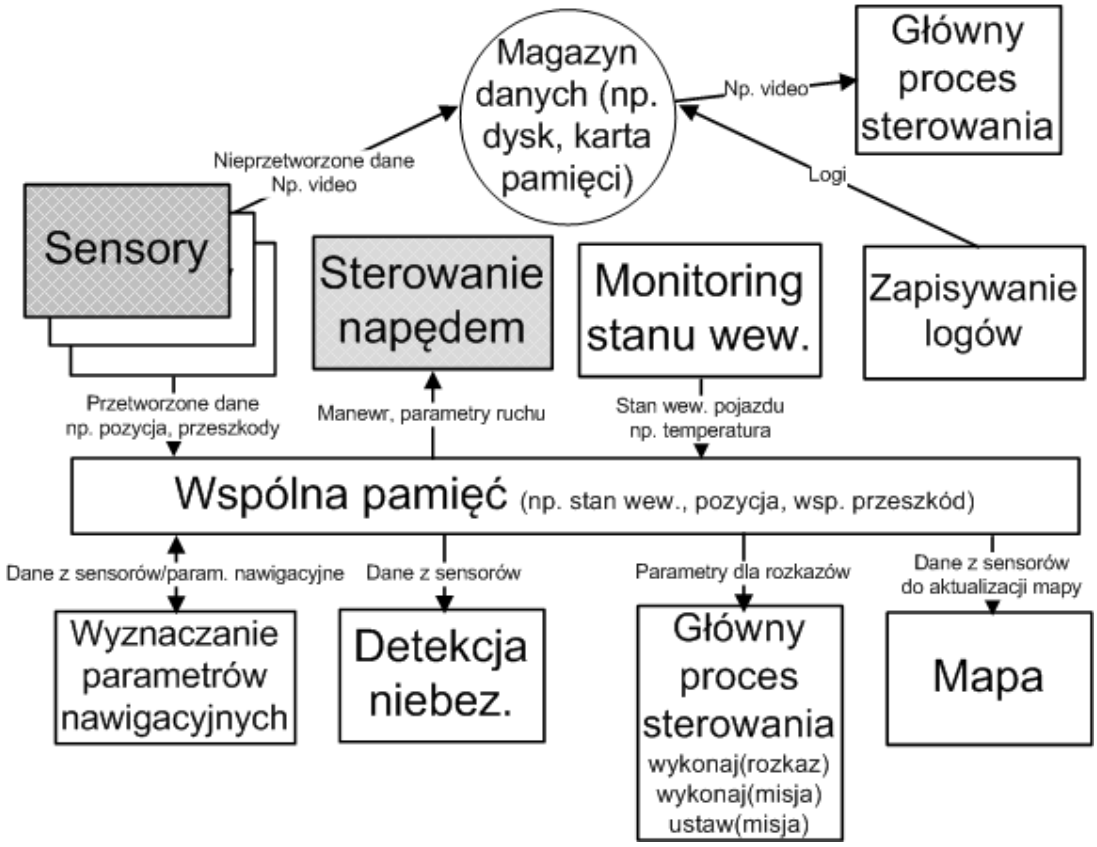
Z uwagi na fakt iż pojazdy autonomiczne a także pojazdy biomimetyczne nie były dotąd budowane w naszym kraju i nie istnieje krajowa technologia w tym zakresie, co więcej, nawet jeśli istnieje technologia budowy tych pojazdów poza krajem to albo stanowi ona ścisłą tajemnicę lub też nie jest na tyle dojrzała aby ją wprost wykorzystać, konstrukcja tych pojazdów zarówno pod względem ukształtowania kadłuba, materiałów, sensorów, innych urządzeń ale również oprogramowania stanowi ogromne wyzwanie.

W przypadku oprogramowania ogromnym problemem jest zapewnienie skalowalności czyli zdolności do szybkiej i łatwej rozbudowy o nowe funkcje i możliwości. Znaczenie tej właściwości oprogramowania dla projektowanego pojazdu wynika z tego iż system sterowania pojazdem będzie musiał podlegać wielu modyfikacjom, ponieważ jest on budowany od samego początku, nie są znane problemy z którymi zespół projektowy będzie miał do czynienia w trakcie realizacji projektu, zakłada się że rozwój systemu będzie stopniowy, wymagał będzie wielu korekt a jego architektura musi być tak przemyślana aby każdy nowy element systemu nie burzył tego co zostało wcześniej zaimplementowane.

W artykule zaprezentowano projekt elastycznej architektury systemu sterowania pojazdem umożliwiający łatwe dodawanie nowych zachowań autonomicznych oraz sensorów które jak założono będą podlegały najczęstszym modyfikacjom a także hierarchizację zachowań pozwalającą na konstrukcję złożonych zachowań poprzez kombinację zachowań prostych [1].

2 GŁÓWNE ZAŁOŻENIA DOTYCZĄCE ARCHITEKTURY OPROGRAMOWANIA

Proces sterowania ABPP nie dotyczy tylko i wyłącznie sposobu jego poruszania się pod wodą ale także innych czynności. To w jaki sposób ABPP będzie się zachowywał w czasie nie zależy tylko i wyłącznie od danych nawigacyjnych ale również od innych czynników takich jak na przykład stan wewnętrzny pojazdu (stan różnych jego mechanizmów i komponentów), sytuacja taktyczna, zauważony nieznany obiekt itd. Z uwagi na powyższe, oddzielenie w całej architekturze programowej ABPP komponentu (modułu, podsystemu) nawigacyjnego od komponentu autonomicznego



Rysunek 2. Diagram przepływu danych pomiędzy procesami w systemie informatycznym ABPP (formą zbliżony do diagramu przepływu danych DFD. Na diagramie uwzględniono wyłącznie przepływy danych do/z magazynu danych (np. dysku) oraz pamięci wspólnej)

sterowania, co jest obecnie często stosowanym rozwiązaniem w przypadku pojazdów podwodnych, jest błędem. W proponowanej architekturze komponent nawigacyjny jest elementem składowym całego systemu sterowania który tak jak inne komponenty dostarcza właściwych dla siebie informacji oraz realizuje pewne funkcje umożliwiające podejmowanie właściwych decyzji przez system sterowania.

Architektura systemu powinna zakładać możliwość jego rozbudowy – skalowalność systemu. Dodawanie nowych elementów nie może burzyć systemu, nie może wymuszać na projektantach, programistach systemu zmiany w dotychczasowej architekturze, np. poprzez dodawanie nowych funkcji do istniejących już komponentów, modułów, klas, podsystemów. Rozbudowa całego systemu powinna być pomyślana w taki sposób aby nowe elementy dopasowywały się do tego co już istnieje. To co zostało już zbudowane i stanowi element systemu nie powinno podlegać zmianom.

Proponowana architektura została tak skonstruowana aby umożliwić bezproblemowe dodawanie następujących elementów: (i) nowych sensorów, (ii) nowych zachowań pojazdu (autonomicznych i w trybie zdalnego sterowania).

3 OGÓLNY OPIS ARCHITEKTURY

W architekturze przedstawionej na rys. 1 i 2 wyróżniono trzy grupy komponentów programowych: (i) komponenty dostarczające informacje (producenci informacji), np. komponenty obsługujące sensory, mapa, komponent nawigacyjny (Wyznaczanie parametrów nawigacyjnych), informacja ta może być w formie nieprzetworzonej np. informacja o odległości od obiektu z echosondy lub przetworzonej np. położenie obiektu (namiar i odległość) zaobserwowanego na obrazie z kamery, za przetwarzanie informacji odpowiedzialny jest komponent obsługujący dane źródło informacji, np. system obsługi kamery zajmuje się wyłącznie przetwarzaniem danych wizyjnych (ii) komponenty wykonawcze czyli takie które wykonują polecenia i sterują urządzeniami, mechanizmami pojazdu, np. kamerą, sterem, silnikiem, światłami, część z komponentów może być zarówno producentem informacji jak również komponentem wykonawczym, np. komponent obsługi kamery, (iii) główny system sterowania który na podstawie informacji dostarczonej przez

producentów informacji wydaje polecenia komponentom wykonawczym.

Każdy z producentów informacji w większości działa niezależnie tzn. pozyskuje informacje z sensora (np. komponent obsługi echosondy, kamery) lub bazy danych (mapa), przetwarza ją (opcjonalnie) i wprowadza w ustalone, ściśle określone wcześniej miejsce (np. do wspólnej pamięci). Niezależność oznacza tutaj ciągłe produkowanie informacji (w pętli) – główny system sterowania nie wymusza na producentach pracy, działają one w sposób ciągły niezależnie od reszty systemu. Jest również możliwość działania na żądanie, tzn. główny system sterowania może potrzebować co jakiś czas pewnych informacji. W takiej sytuacji zmienia parametry działania danego producenta, tzn. zamiast prosić go o wytworzenie pewnej informacji i oczekiwanie na jej zwrot, zmienia konfigurację jego działania i oczekuje na informacje która powinna znaleźć się za jakiś czas w określonym miejscu.

Sterowanie pojazdem za pomocą komponentów wykonawczych odbywa się poprzez komendy dla tych komponentów, np. system starowania napędem otrzymuje komendę z numerem manewru który ma wykonać i parametrami tego manewru.

Sterowanie ABPP przez główny proces sterowania odbywa się za pomocą wcześniej zdefiniowanych prostych lub bardziej złożonych zachowań (operacji) pojazdu np. skręć w prawo, zapal światło, płyń prosto, płyń kursem 90 stopni, zaplanuj trajektorię, uniknij kolizji wykorzystując do tego celu sieć neuronową, uniknij kolizji używając systemu rozmytego. Sterowanie zdalne (wyłącznie blisko powierzchni) odbywa się poprzez przesłanie kanałem komunikacyjnym numeru operacji do wykonania i ewentualnie parametrów tej operacji. Sterowanie autonomiczne jest w takim przypadku uporządkowanym ciągiem operacji lub po prostu programem, który główny system sterowania wykonuje. Wykonanie różnych programów daje różne zachowania autonomiczne które możemy w sposób dowolny kształtować, zmieniać i dopasowywać do sytuacji. Takie rozwiązanie zapewnia iż sterowanie zdalne może polegać na wysłaniu do BPNN programu z ciągiem operacji do wykonania, ciąg ten będzie określany przez operatora konsoli – daje to możliwość sterowania semi-autonomicznego, zamiast sterować zdalnie każdym parametrem określającym zachowanie pojazdu dajemy mu możliwość pewnej samodzielności, stopień samodzielności określa operator konsoli.

Powyższe rozwiązanie umożliwia: (i) prostą synchronizację działań podejmowanych przez BPNN, nie ma konieczności synchronizacji w czasie wielu komponentów wchodzących w skład pojazdu, decyzja co do tego leży w gestii tylko jednego komponentu i jest uporządkowanym w czasie ciągiem operacji – samo-synchronizacja, jedna operacja po drugiej wykonywana w ramach jednego komponentu (w przypadku kiedy będzie konieczność równoległego wykonania pewnych czynności w ramach jednej operacji jest to realizowane poprzez uruchomienie wykonywania tej czynności w oddzielnym wątku, jest to jednak szczegół implementacyjny danej operacji który nie ma wpływu na całościową architekturę systemu), (ii) łatwą rozbudowę systemu o nowe zachowania zarówno autonomiczne jak i zdalnie sterowane, uzyskanie nowych zachowań odbywa się głównie poprzez różne uporządkowanie operacji pojazdu i różne parametry operacji, dodanie nowych operacji korzystających z udostępnionych przez ABPP usług i informacji dostarczanych przez producentów – w takim przypadku system uzupełniany jest o koncepcyjnie taki sam element jak inne operacje, nie dodajemy nowego systemu, podsystemu, komponentu czy modułu ale jeszcze jedną operację która ma dostęp do tych samych informacji oraz komponentów wykonawczych jak inne operacje (stopniowa rozbudowa systemu przy stabilnej architekturze).

4 SZCZEGÓŁOWA ARCHITEKTURA SYSTEMU

Szczegółowa architektura systemu przedstawiona jest na rys. 1 i 2. Zawiera ona: (i) komponenty programowe (kwadraty lub prostokąty) – każdy z komponentów jest wykonywany w oddzielnym procesie, (ii) urządzenia pojazdu (kółka), np. sonar, napęd, ster, światła, magazyn danych (baza danych, plik), (iii) sygnały sterujące oraz przepływy danych pomiędzy komponentami (procesami).

1) **Sensory** (obsługa sensorów, np. echosondy lub grupy echosond, kamery, magnetomatru) – procesy odbierające dane z sensorów, przetwarzające te dane do postaci określonej przez Główny proces sterujący (opcjonalnie) i umieszczające nieprzetworzone i/lub przetworzone dane w określonym wcześniej miejscu (pamięć wspólna, dysk), obsługa sensorów może znajdować się na oddzielnym komputerze w stosunku do głównego procesu sterowania tak aby wymagające dużych mocy obliczeniowych przetwarzanie surowych danych z sensorów nie spowalniało tego ostatniego.

2) **Obsługa radio/hydro** – proces odbierający polecenia z konsoli i przekazujący je do głównego procesu sterowania a także wysyłający różne dane do konsoli. Polecenia dostarczane są do procesu za pomocą radiomodemu (w położeniu nawodnym) lub hydromodemu (w położeniu podwodnym).

3) **Monitoring stanu wewnętrznego** – proces monitorujący stan poszczególnych elementów ABPP i umieszczający ten stan w określonym miejscu. Proces ten w przypadku sytuacji awaryjnej (np. przekroczonej temperatury

jakiegoś z urządzeń), informuje o zaistniałej sytuacji główny proces sterowania który uruchamia odpowiednią procedurę (operację lub ciąg operacji - program) obsługi tej sytuacji.

4) **Mapa** – proces obliczający dla danej pozycji ABPP (wyliczonej przez proces *Wyznaczanie parametrów nawigacyjnych* na podstawie informacji dostarczonej przez urządzenia nawigacyjne, np. system inercyjny INS, log, kompas, pozycja ABPP umieszczona jest w pamięci wspólnej) odległości dla różnych namiarów od najbliższych przeszkód nawigacyjnych i umieszczający tą informację w ściśle określonym miejscu, możliwe jest również rozwiązanie w którym proces ten odciążał będzie *Główny proces sterowania* od pewnych funkcji np. zamiast w trybie ciągłym generować odległości od niebezpieczeństw nawigacyjnych może poinformować główny proces sterujący poprzez odpowiedni komunikat (alarm) o sytuacji niebezpiecznej (nadmierne zbliżenie się do obiektu podwodnego, np. wraku), co więcej, informacje o odległościach od najbliższych obiektów o charakterze stałym mogą być przekazywane przez *Mapę* na wyraźną prośbę *Głównego procesu sterowania*. Dodatkową funkcją tego procesu może być planowanie trajektorii (w takim przypadku funkcja ta nie byłaby po stronie głównego procesu sterowania jako oddzielna operacja).

5) **Zapisywanie logów** – proces zapisu logów czyli całej historii działania systemu, przechwytywanie komunikatów przesyłanych przez inne procesy zawierających wybrane parametry pracy ważne z punktu widzenia eksploatacyjnego urządzeń, w tym generowanych przez nie alarmów i ostrzeżeń, archiwizowanie tych danych w formie logów w magazynie danych, np. zapisywanie pojedynczego logu zawierającego: datę i czas, parametry wektora stanu ABPP, alarmów i ostrzeżeń itp.

6) **Detekcja niebezpieczeństw** – proces którego zadaniem jest wykrywanie sytuacji niebezpiecznych dla pojazdu wymagających reakcji głównego procesu sterowania, jego zadanie jest podobne do zadania *Monitoringu stanu wewnętrznego*, różnica polega na tym iż ten pierwszy „patrzy w środek ABPP" i zgłasza alarm jeśli sytuacja tego wymaga, natomiast *Detekcja niebezpieczeństw* „patrzy na zewnątrz", tzn. analizuje dane dostarczane przez sensory i jeśli uzna że ABPP jest w niebezpieczeństwie zgłasza to *Głównemu procesowi sterowania* który z kolei uruchamia odpowiednią procedurę (operację lub program) obsługi tej sytuacji. Przykładem takiej sytuacji jest kolizja z innym obiektem.

7) **Sterowanie napędem** – proces którego zadaniem jest wydawanie konkretnych poleceń (w sposób ciągły – w pętli) na ster i napęd. Rodzaj manewru jak również jego parametry, wszystko określone przez *Główny proces sterujący* odczytuje z pamięci wspólnej lub też otrzymuje bezpośrednio od *Głównego procesu sterującego*. Proces ten może znajdować się na oddzielnym mikrokontrolerze.

8) **Wyznaczanie parametrów nawigacyjnych** (producent informacji) – proces którego zadaniem jest odczytywanie danych dostarczanych przez różne sensory (procesy obsługi sensorów, np. log, INS, GPS – w położeniu nawodnym), przetwarzanie tych danych, a następnie umieszczanie przetworzonych danych nawigacyjnych (np. pozycja, kurs, prędkość, kąty Eulera) w określonym miejscu systemu (np. pamięć wspólna)

9) **Główny proces sterowania** – w trybie zdalnego sterowania proces ten otrzymuje od procesu *Obsługi radiomodemu/hydromodemu* polecenia do wykonania (operacje z parametrami) które wykonuje wysyłając odpowiednie sygnały sterujące (komunikat do innego procesu lub też zapis odpowiednich danych w pamięci wspólnej) do procesów wykonawczych (np. napęd). W trybie autonomicznym proces ten wykonuje zapamiętany wcześniej lub przekazany z konsoli program (ciąg operacji).

Funkcje rozkazu (operacji):

Rozkazy(operacje) służą do sterowania ABPP (układ napędowy, balast), innymi procesami i w konsekwencji również urządzeni pokładowymi ABPP korzystając z danych zgromadzonych w pamięci wspólnej lub magazynie danych. Rozkazy mogą być typu prostego np. skręć w prawo, zanurzaj się szybko, a także bardziej skomplikowane typu złożonego - mogące korzystać z rozkazów typu prostego, np. unikaj kolizji, wykonaj przejście na pozycję 54.323243°N, 18.994339°E przy zanurzeniu 6 m, wynurz się i zarejestruj obraz video, zarejestruj pozycję GPS.

Takie rozwiązanie umożliwia tworzenie nowych rozkazów bez konieczności modyfikacji kodu w innych elementach systemu. Rozkazy wysyłane z konsoli mają postać zakodowaną (patrz Rys. 1). podobnie jak w przypadku kodu asemblerowego, gdzie pojedynczy rozkaz odpowiada pewnemu mikroprogramowi. Pełna implementacja rozkazów znajduje się w *Głównym Procesie Sterowania.*

Kodowanie rozkazu:

Zakodowany rozkaz (Rys. 1) zawiera następujące pola:

- numer rozkazu: kod czynności do wykonania, np. wynurz się, zarejestruj obraz video,

- numer rozkazu alternatywnego: kod czynności do wykonania w przypadku, kiedy pojazd nie osiągnie przez założony czas zadanego stanu np. zadanej głębokości określonej w rozkazie podstawowym,
- parametry: wartości parametrów rozkazu, np. zadana głębokość zanurzenia się,
- czas: przedział czasu oczekiwania na osiągnięcie stanu zadanego w rozkazie, tzn. oczekiwany czas wykonania się rozkazu, np. osiągnięcia zadanej głębokości zanurzenia się, dla wartości -1 tego parametru, kolejny rozkaz wykonywany będzie natychmiast (istotne w przypadku rozkazów prostych, np. wychyl płetwy ogonowe maksymalnie w prawo)
- stan zadany: stan zadany po zakończeniu poprawnego wykonania się rozkazu.

Przykład działania:

- Radiomodem odbiera zakodowany opis czynności realizowanych w czasie misji w postaci ciągu zakodowanych rozkazów,
- **Obsługa radio/hydro** odbiera ciąg rozkazów kodujących tą misję, (np. poruszanie się po zadanych punktach zwrotu – trajektorii z rejestracją obrazu video ,
- **Obsługa radio/hydro** wywołuje usługę (wysyła komunikat do) Głównego Procesu Sterowania, np. *ustawMisję(misja)* ,
- **Obsługa radio/hydro** po uzyskaniu odpowiedniej komendy z konsoli wysyła do *Głównego Procesu Sterowania* komunikat *start()* - zacznij wykonywać misję,
- **Główny Proces Sterowania** dekoduje pierwszy rozkaz (np. wynurz się – stan zadany=na powierzchni, czas=dowolny), przechodzi do jego implementacji i zaczyna go wykonywać, przykładowa implementacja: zainicjowanie wypompowywania wody ze zbiornika balastowego, rozkaz sprawdzania w pętli ciśnienia na zewnątrz kadłuba (wartość ciśnienia w pamięci i w momencie, kiedy to ciśnienie odpowiada ciśnieniu atmosferycznemu pompa jest zatrzymywana, wykonanie rozkazu uważa się za skończone i *Główny Proces Sterowania* przechodzi do wykonywania następnego rozkazu (np. włącz kamerę, czas pracy kamery=2 minuty, parametr=patrz na północ)
- Kolejny rozkaz inicjuje proces obsługi kamery nawodnej; kamera zaczyna rejestrować obraz i przesyłać go do magazynu danych, *Główny Proces Sterowania* realizuje sterowanie ogonem i płetwami aby ABPP ustawił się na północ, rozkaz w trakcie sterowania korzysta z wyników pomiarów kompasu umieszczanych w pamięci wspólnej i w momencie kiedy pojazd jest skierowany na północ steruje ogonem i płetwami, aby utrzymać kurs i płynąć z bardzo małą prędkością; W momencie kiedy upływa założony czas, *Główny Proces Sterowania* poprzez proces obsługi kamery (*Obsługa sensora*) wyłącza kamerę i w tym momencie kończy się wykonanie rozkazu; *Główny Proces Sterowania* przechodzi do wykonywania kolejnego rozkazu (rozkazy takie jak utrzymaj pojazd na kursie realizowane są za pomocą prostszych rozkazów)
- przez cały czas proces *Detekcja niebezpieczeństw* czyta dane zapisane przez echosondy i/lub sonar w pamięci wspólnej i kiedy uzna, że pojazd jest w niebezpieczeństwie wywołuje w *Głównym Procesie Sterowania* odpowiednią usługę (rozkaz uniknięcia niebezpieczeństwa, który również bazuje na danych z echosond), podobnie jest z *Monitoringiem stanu wewnętrznego* – w przypadku, kiedy jakiś stan zagraża pojazdowi (np. duży wzrost wilgotności w przedziale głównym kadłuba) uruchamiany jest odpowiedni rozkaz, np. wynurz się i przekaż swój stan do konsoli sterowania,
- przez cały czas wykonywania misji proces *Obsługa sensorów*, np. sonaru, rejestruje dane, dane w postaci nieprzetworzonej wysyłane są do magazynu danych, dane przetworzone, np. odległość od obiektów usytuowanych wkoło pojazdu wysyłane są do pamięci wspólnej, w trakcie wykonywania misji proces *Wyznaczanie parametrów nawigacyjnych* czyta przetworzone dane z pamięci wspólnej, np. dane z logu, INS-a, kompasu i wyznacza parametry nawigacyjne pojazdu, czyli prędkość, pozycję i parametry orientacji przestrzennej; Parametry te wysyłane są do pamięci wspólnej tak, aby inne procesy, szczególnie *Główny Proces Sterowania* i jego rozkazy mogły korzystać z tych parametrów.

5 PODSUMOWANIE

Artykuł prezentuje projekt oprogramowania systemu sterowania Autonomicznym Biomimetycznym Pojazdem Podwodnym. W projekcie założono iż system sterowania musi być maksymalnie odporny na wymianę sensorów, dodawanie nowych sensorów oraz dodawanie, aktualizację i modyfikację zachowań autonomicznych. Ponadto ważnym założeniem

jest również hierarchizacja ww. zachowań umożliwiająca stopniową budowę systemu od zachowań prostych do zachowań bardziej złożonych składających się z zachowań prostych. Wszystkie z zachowań mogą przybierać formę implementacji programowych pewnych algorytmów ale również sieci neuronowych, systemów rozmytych itp.

Informacje dodatkowe

Artykuł jest finansowany z granta Narodowego Centrum Badań i Rozwoju nr. DOBR-BIO4/033/13015/2013

Bibliografia

1. Colgate, J. E. & Lynch, K. M. Mechanics and Control of Swimming: A Review. *IEEE Journal of Oceanic Engineering* **29,** 660–673 (3 2004).
2. Low, K. H. Modelling and parametric study of modular undulating fin raysfor fish robots. *Mechanism and Machine Theory* **44,** 615–632 (2009).
3. Malec, M., Morawski, M., Szymak, P. & Trzmiel, A. Analysis of Parameters of Traveling Wave Impact on the Speed of Biomimetic Underwater Vehicle, *Trans Tech Publications, Solid State Phenomena* **210,** 273–279 (2014).
4. Szymak, P., Malec, M. & Morawski, M. Directions of Development of Underwater Vehicle with Undulating Propulsion, *HARD Publishing Company, Polish Journal of Environmental Studies* **19,** 107–110 (4A 2010).