

**Radosław MACIASZCZYK, Sebastian KRUSZKO**  
ZACHODNIOPOMORSKI UNIwersYTET TECHNOLOGICZNY WYDZIAŁ INFORMATYKI,  
ul. Żołnierska 49, Szczecin

## Optimalizacja obliczeń rozszerzonego filtra Kalmana dla mikrokontrolera z rodziny ARM

Dr inż. Radosław MACIASZCZYK

Ukończył studia na Wydziale Informatyki w roku 2000, w roku 2006 obronił pracę doktorską. Obecnie jest adiunktem w Katedrze Architektury Komputerów i Telekomunikacji na Wydziale Informatyki Zachodniopomorskiego Uniwersytetu Technologicznego w Szczecinie. Jego zainteresowania naukowe to techniki cyfrowego przetwarzania sygnałów oraz systemy mobilne.



e-mail: rmaciaszczyk@wi.zut.edu.pl

Inż. Sebastian KRUSZKO

Ukończył studia inżynierskie na Wydziale Informatyki Zachodniopomorskiego Uniwersytetu Technologicznego w 2011 r. Obecnie jest studentem I roku studiów II stopnia na Wydziale Informatyki Zachodniopomorskiego Uniwersytetu Technologicznego w Szczecinie. Jego zainteresowania naukowe obejmują zagadnienia związane z robotyką, informatyką, oraz elektroniką.



e-mail: skruszko@wi.zut.edu.pl

### Streszczenie

W artykule została zaprezentowana procedura optymalizująca liczbę operacji arytmetycznych dla rozszerzonego filtra Kalmana integrującego odczyty z czujników inercyjnych w systemie nawigacji bezwładnościowej. Orientacja pojazdu jest określana z wykorzystaniem reprezentacji kwaternionowej, dzięki czemu przy obrotach nie występuje osobliwość zwana „Gimbal lock”. Dzięki znacznemu zmniejszeniu liczby operacji, możliwe stało się zaimplementowanie filtra EKF w mikrokontrolerze ARM AT91SAM7S256 uzyskując 30% obciążenie mikrokontrolera, przy częstotliwości pracy 55 MHz.

**Słowa kluczowe:** fuzja danych, IMU, rozszerzony filtr Kalmana, kwaterniony.

### Optimization calculations of an extended Kalman filter for the ARM microcontroller family

#### Abstract

One of the most important subsystems of an unmanned vehicle is the inertial measurement unit (IMU) which provides obtainment of data about orientation in space. Its work is conditioned by measurement equipment and a filtering algorithm for the input data. This paper describes implementation of an extended Kalman filter which is an important element in the IMU calculation procedures. As a result of optimization calculations, the quantity of arithmetic operations performed has significantly been reduced, i.e. the number of additions has decreased by 84.57%, subtractions by 41.84%, multiplications by 78.3%, divides by 61.54%. Exponentiation operation and iteration variables have been completely eliminated. Moreover, the fourquadrant inverse tangent function has been replaced by polynomial approximation, calculated with use of Horner's scheme, so that the number of function calls from the library math.h decreased by 77.78%. This significant simplification of the algorithm has been obtained at the cost of execution an additional two square root operations and increasing number of variables used by 16.54%. Reduction in the number of operations has made it possible to implement an EKF filter on a microcontroller ARM AT91SAM7S256 obtaining the microcontroller load of 30% for the operating frequency equal to 55 MHz.

**Keywords:** data fusion, IMU, extended Kalman filter, quaternion representation.

## 1. Wprowadzenie

System nawigacji bezwładnościowej (IMU – ang. inertial measurement unit) jest systemem pozwalającym uzyskać wartości określonych co do kierunku przyspieszeń liniowych bądź prędkości kątowych. Umożliwia to określenie ruchu obrotowego obiektu, jego aktualnej orientacji w przestrzeni, a także – jeśli zachodzi taka potrzeba, ruchu postępowego mierzonego względem położenia początkowego. W zależności od implementacji w celu pozyskania danych pomiarowych używane są różnorodne sensory.

Niedoskonałości budowy poszczególnych czujników powodują powstawanie niedokładności wymuszających synergiczne połączenie oraz filtrację wykonywanych pomiarów.

Oprócz problemów związanych z niedokładnością dokonanych pomiarów należy zwrócić uwagę na konieczność reprezentacji danych pozyskanych z czujników inercyjnych, w sposób umożliwiający ich jednoznaczną interpretację, oraz wprowadzając możliwość jak najmniejszą redundancję danych. Usytuowanie bryły sztywnej [1] względem pewnego układu odniesienia opisuje się za pomocą położenia oraz orientacji.

Można spotkać się z kilkoma różnymi sposobami przedstawienia orientacji bryły sztywnej:

- macierz rotacji,
- reprezentacja „oś/kąt”,
- kąty Eulera,
- kwaterniony jednostkowe.

Każda z wymienione reprezentacji pozwala na jednoznaczne określenie orientacji, jednak jedynie kwaterniony pozbawione są pewnej osobliwości znanej pod nazwą „Gimbal lock”. Jednak postać kwaternionowa jest nieliniowa oraz do obliczeń wymaga stosowania tzw. algebry kwaternionów zdefiniowanej przez William'a Rowan'a Hamilton'a [2]. Powoduje to, iż obliczenia z wykorzystaniem kwaternionów wymagają większej liczby operacji. W układach IMU można uzyskać orientację z wykorzystaniem różnych sensorów np. akcelerometru i żyroskopu. Uzyskane wyniki z pojedynczego czujnika pozwalają na określenie orientacji, jednak obarczone jest znacznym błędem. Wykorzystanie fuzji danych pozwala na zwiększenie dokładności wyznaczenia orientacji.

Fuzji danych często dokonuje się z wykorzystaniem filtra Kalmana [3], przy czym najczęściej wykonuje się obliczenia dla poszczególnych osi korzystając np. z reprezentacji orientacji za pomocą kątów Eulera. Tym samym w trakcie obliczeń może wystąpić osobliwość „Gimbal lock”. Zastosowanie kwaternionów w połączeniu z filtrem Kalmana pozwala na uzyskanie satysfakcjonujących wyników przy jednoczesnym pozbyciu się tej osobliwości. Jednak nieliniowość kwaternionów pociąga za sobą konieczność zastosowanie algorytmu rozszerzonego filtra Kalmana co znacznie zwiększa liczbę operacji arytmetycznych.

## 2. Rozszerzony filtr Kalmana

Filtr Kalmana (w skrócie FK) jest typem filtracji optymalnej rekursywnej, tzn. informacje przetwarzane są w każdym kroku a filtr nie przechowuje stanów z poprzednich cykli pomiarów [4]. Optymalność filtra jest uzyskiwana poprzez uwzględnienie wszystkich dostępnych pomiarów danej wielkości fizycznej niezależnie od ich dokładności. W wyniku filtracji dostajemy estymację stanu mierzonego układu.

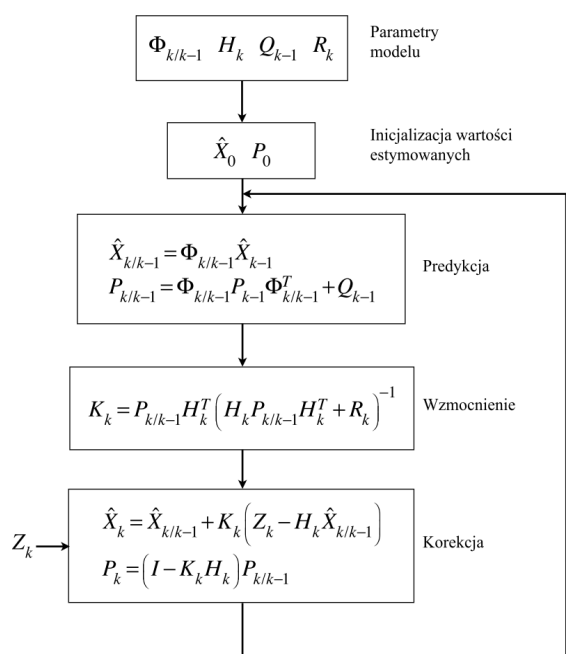
Dużą zaletą filtru Kalmana jest możliwość wyznaczenia wartości niezmiernych na podstawie danych wejściowych. Filtr przeznaczony on jest dla procesów liniowych [4]. Jednak w przypadku procesów nieliniowych istnieją pewne rozszerzenia filtru Kalmana – EKF czy UKF [5].

Proces obliczeń filtru Kalmana jest procesem ciągłym i polega na cyklicznym obliczeniu wektora stanu oraz jego kowariancji. FK składa się dwóch etapów (faz):

- predykcji,
- korekcji.

W etapie predykcji wyznaczane są wartości a priori modelu stanu jak i kowariancji i nazywany jest często aktualizacją czasową [2], natomiast w etapie korekcji obliczane są wartości a posteriori stanu procesu i kowariancji i nazywany jest aktualizacją pomiarową. Faza predykcji korzysta z danych obliczonych w poprzednim kroku w fazie korekcji. Jak wspomniano jeżeli istnieją nieliniowe równania opisu procesu lub pomiaru to konieczne jest zastosowanie pewnej modyfikacji. W poniższym rozwiązaniu wykorzystano algorytm rozszerzonego filtru Kalmana, który wymaga aby dokonać linearyzacji równań nieliniowych. Szczegółowe wyjaśnienie EKF można znaleźć w następujących pozycjach [6, 7].

Proces rozszerzonego filtru Kalmana został przedstawiony za pomocą diagramu na rysunku nr 1 [6].



Rys. 1. Diagram procesu obliczeniowego rozszerzonego filtru Kalmana [6]  
Fig. 1. Diagram of Extended Kalman filter [6]

W prezentowanym przykładzie fuzja danych będzie dokonana z wykorzystaniem żyroskopu i akcelerometru. Na etapie predykcji do obliczeń wykorzystanie są odczyty z żyroskopu, natomiast na etapie aktualizacji odczyty z akcelerometru. Orientacja obiektu będzie opisywana z wykorzystaniem reprezentacji kwaternionowej.

Równanie procesu w procedurze obliczeniowej EFK dla tak zdefiniowanego problemu jest następujące (1-2):

$$\dot{q} = [\dot{q}_0 \quad \dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]^T = \frac{1}{2} \cdot \Omega \cdot q, \quad (1)$$

$$\Omega = \begin{pmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{pmatrix}, \quad (2)$$

gdzie:

$q = [q_0, q_1, q_2, q_3]^T$  - współczynniki kwaternionu oznaczającego aktualną orientację pojazdu,  
 $\omega_x, \omega_y, \omega_z$  - zmiana kąta położenia pojazdu dla poszczególnych osi.

Natomiast równania pomiaru przyjmuje postać (3-4):

$$Z_k = H_k \cdot q^T, \quad (3)$$

$$H = \text{jacobian}([\phi, \theta, \psi], [q_0, q_1, q_2, q_3]), \quad (4)$$

gdzie:

$$\phi = \text{atan2}\left(2 \cdot (q_0 \cdot q_1 + q_2 \cdot q_3), \left(1 - 2 \cdot (q_1^2 + q_2^2)\right)\right), \quad (5)$$

$$\theta = \text{asin}\left(2 \cdot (q_0 \cdot q_2 - q_1 \cdot q_3)\right) \quad (6)$$

$$\psi = \text{atan2}\left(2 \cdot (q_0 \cdot q_3 + q_1 \cdot q_2), \left(1 - 2 \cdot (q_1^2 + q_2^2)\right)\right) \quad (7)$$

$\phi, \theta, \psi$  - poszczególne kąty Eulera oznaczające orientacje w przestrzeni.

Uwzględniając powyższe równania a także znane parametry modelu poszczególne macierze z diagramu obliczeniowego EKF przyjmują postać (8-11):

$$\hat{X}_{k-1} = [q_0, q_1, q_2, q_3, \omega_{x0}, \omega_{y0}, \omega_{z0}]^T, \quad (8)$$

$$Q_{k-1} = \text{diag}(Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7), \quad (9)$$

$$R_k = \text{diag}(R_1, R_2, R_3), \quad (10)$$

$$\Phi_{k/k-1} = I_7 + \frac{1}{2} \cdot \Omega' \cdot \Delta t, \quad (11)$$

$$\Omega' = \begin{pmatrix} 0 & \omega_z & -\omega_y & \omega_x & -q_3 & q_2 & -q_1 \\ -\omega_z & 0 & \omega_x & \omega_y & -q_2 & -q_3 & q_0 \\ \omega_y & -\omega_x & 0 & \omega_z & -q_1 & -q_0 & q_3 \\ -\omega_x & -\omega_y & -\omega_z & 0 & q_0 & q_1 & q_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (12)$$

gdzie:

$\omega_{x0}, \omega_{y0}, \omega_{z0}$  - wyliczany dryft żyroskopów

$Q_i$  - wariancja procesu

$R_i$  - wariancja pomiaru

Natomiast wartości początkowe estymowanych wartości przyjmują postać:

$$\hat{X}_0 = [1, 0, 0, 0, 0, 0, 0]^T, \quad (13)$$

$$P_0 = I_7. \quad (14)$$

### 3. Optymalizacja liczby operacji arytmetycznych

Przedstawiony w poprzednim rozdziale algorytm wymaga znacznej liczby operacji arytmetycznych. Implementując go w sposób bezpośredni wymaga on 10091 pojedynczych operacji arytmetycznych dla każdej iteracji obliczeń. Szczegółowe przedstawienie liczby operacji przedstawione jest w tabeli nr 1 w kolumnie oznaczonej jako algorytm referencyjny.

Tab. 1. Rezultat optymalizacji algorytmu  
Tab. 1. The result of the optimization algorithm

Rodzaj operacji.	Algorytm referencyjny	Algorytm zoptymalizowany	Różnica w %
dodawanie	3947	609	84,57
odejmowanie	533	310	41,84
mnożenie	5447	1182	78,3
dzielenie	65	25	61,54
potęgowanie	72	0	100
pierwiastkowanie	0	2	-
funkcje trygonometryczne	27	6	77,78
ilość użytych zmiennych	260	303	-16,54

Uwzględniając 100 Hz częstotliwość obliczeń daje to ponad milion operacji arytmetycznych na liczbach zmiennoprzecinkowych. Biorąc pod uwagę że to jest tylko jedna z wielu procedur jakie trzeba zaimplementować w układzie IMU, to praktyczna implementacja w mikrokontrolerze wybranym do projektu czyli ARM AT91SAM7S256 okazała się niemożliwa.

W celu optymalizacji liczby operacji zastosowano poniższe czynności:

- zamieniono operacje macierzowe na instrukcje sekwencyjne, przy czym wyeliminowano niepotrzebne operacje arytmetyczne,
- wyznaczono wartości pośrednie i użyto ich wielokrotnie,
- zastąpiono funkcję czteroćwiartkowego arcusa tangensa przez szereg Taylora.

Wyeliminowanie zbędnych operacji odbyło się poprzez analizę współczynników macierzy stałych, a tym samym usunięcie wszystkich operacji mnożenia przez element zerowy. Kolejnym krokiem analizy było wyszukanie wielokrotnie występujących takich samych operacji arytmetycznych. Tym samym możliwe stało się wyznaczenie wartości pośrednich. Kosztem tych operacji było zwiększenie liczby zmiennych, a także zwiększenie kodu programu.

Obliczanie wartości  $\arctan 2$  przy użyciu funkcji zaimplementowanej w bibliotece `math.h` wiąże się ze znaczącym spowolnieniem wykonywanych obliczeń. W związku z tym funkcja ta została zastąpiona przez jej wielomianowe przybliżenie, tj. szereg Taylora i obliczana jest przy użyciu schematu Hornera, redukującego liczbę operacji arytmetycznych.

O ile dwie pierwsze czynności nie wpływają na dokładność obliczeń, to wykorzystanie szeregu Taylora do obliczania wartości arcusa Tangensa wymagało przeprowadzenia dodatkowych badań dotyczących minimalnej liczby elementów w szeregu Taylora. Wyniki ich przedstawione są w tabeli nr 2. Jak można zauważyć wartości poszczególnych średnich różnic dla poszczególnych kątów w większości przypadków nie ulegają zmianie od szeregu Taylora złożonego z trzech elementów, w związku z czym został on użyty w docelowej implementacji. Porównania dokonywano z funkcjami bibliotecznymi ze standardowej biblioteki języka C.

Tab. 2. Średnia różnica błędów w zależności od liczby elementów w szeregu Taylora

Tab. 2. The mean difference of error depending on the number of elements in the Taylor series

Liczba elementów w szeregu Taylora	Średnia różnica błędów dla osi Roll (°)	Średnia różnica błędów dla osi Pitch (°)	Średnia różnica błędów dla osi Yaw (°)
i=1	1,095	8,924	4,362
i=2	0,249	0,778	1,088
i=3	0,091	0,778	0,329
i=4	0,091	0,778	0,329
i=5	0,091	0,778	0,329
i=6	0,091	0,778	0,329

Tak znaczne zmniejszenie liczby operacji arytmetycznych pozwoliło na implementację algorytmu w mikrokontrolerze. Przeprowadzone badania pokazały, że wykonanie zoptymalizowanej procedury z częstotliwością 100 Hz, obciąża procesor w około 30%. Tym samym możliwe stało się zastosowanie tej procedury w układzie IMU. Zaproponowana implementacja algorytmu została napisana w 413 liniach kodu.

### 4. Podsumowanie

W pracy omówiony został algorytm rozszerzonego filtru Kalmana integrujący odczyty sensorów (akcelerometru oraz żyroskopu) zainstalowanych w IMU. Do opisu orientacji wykorzystano reprezentację kwaternionową, co pozwoliło wyeliminować problem „Gimbal lock”. Zastosowanie kwaternionów znacznie zwiększyło złożoność obliczeniową, dlatego konieczne stało się zoptymalizowanie liczby operacji arytmetycznych. Dzięki zastosowanym zabiegom udało się zredukować łączną liczbę operacji arytmetycznych o ponad 77%. Ponadto należy wspomnieć iż implementacji dokonano na liczbach zmiennoprzecinkowych typu float. Przy wymaganej częstotliwości obliczeń (100 Hz) na mikrokontrolerze ARM AT91SAM7S256 uzyskano około 30% zajętości czasu procesora.

Dalsze prace będą prowadzone w kierunku stworzenia dedykowanego układu obliczeniowego realizującego tylko tę funkcję, a także poszukiwania innych metod fuzji danych o mniejszej złożoności obliczeniowej.

### 5. Literatura

- [1] Kozłowski K., Dutkiewicz P., Wróblewski W.: Modelowanie i sterowanie robotów. Wydawnictwo Naukowe PWN, Warszawa, 2003.
- [2] Sierpiński W.: Arytmetyka teoretyczna. PWN, 1968.
- [3] Zieliński T.: Cyfrowe przetwarzanie sygnałów. Wydawnictwo Komunikacji i Łączności, Warszawa, 2007.
- [4] Lyons Richard G.: Wprowadzenie do cyfrowego przetwarzania sygnałów, Wydawnictwa Komunikacji i Łączności WKŁ, 2006.
- [5] Julier S.J., Uhlmann J.K.: Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*:204 pp401–422.
- [6] Tongyue Gao, Zhenbang Gong, Jun Luo, Wei Ding and Wei Feng: An Attitude Determination System For A Small Unmanned Helicopter Using Low-Cost Sensors, *Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics*, pp 1203 – 1208.
- [7] Welch G. and Bishop G.: An introduction to the kalman filter. Technical Report 95-041, University of North Carolina, Department of Computer Science, 1995.