

Sławomir GRYŚ, Waldemar MINKINA

POLITECHNIKA CZĘSTOCHOWSKA, INSTYTUT ELEKTRONIKI I SYSTEMÓW STEROWANIA

Identyfikacja dynamiki metodą dwóch czujników w środowisku programowo-sprzętowym Matlab/Simulink-TMS320C6713 DSK

Dr inż. Sławomir GRYŚ

Absolwent Wydziału Elektrycznego Politechniki Częstochowskiej. Stopień naukowy doktora nauk technicznych uzyskał w 2003 r. Pracuje na stanowisku adiunkta w Instytucie Elektroniki i Systemów Sterowania Politechniki Częstochowskiej. Jest autorem podręcznika, współautorem monografii i kilkudziesięciu artykułów publikowanych w czasopiśmie krajowych, zagranicznych oraz materiałach konferencyjnych. Jego zainteresowania naukowe obejmują zagadnienia metrologiczne, elektronikę i nauki informatyczne.



e-mail: gryś@el.pcz.czest.pl

Prof. dr hab. inż. Waldemar MINKINA

Ukończył w roku 1977 Wydział Elektryczny Politechniki Częstochowskiej, doktorat w 1983 r. w Instytucie Metrologii Elektrycznej Politechniki Wrocławskiej, habilitacja w 1995 r. na Wydziale Automatyki Politechniki Lwowskiej. W roku 2006 uzyskał tytuł profesora nauk technicznych. Jest autorem i współautorem 5 monografii z metrologii, ponad 100 publikacji, 5 patentów oraz 2 zgłoszeń patentowych. Specjalizuje się w problematyce termometrii, termografii komputerowej, pomiarów cieplnych oraz teorii i techniki wymiany ciepła.



e-mail: minkina@el.pcz.czest.pl

Streszczenie

W artykule przedstawiono sposób implementacji algorytmu adaptacyjnej korekcji błędów dynamicznych metodą dwóch czujników. Zadanie zrealizowano w środowisku Matlab/Simulink z wykorzystaniem filtrów adaptacyjnych FIR ze strojeniem parametrów wg algorytmu RLS. Zastosowanie pakietu Matlab R14 pozwoliło na symulację działania elementów układu pomiarowego, a przede wszystkim, za pośrednictwem odpowiednich narzędzi programistycznych generację kodu maszynowego dla procesora sygnałowego TMS320C6713. W rezultacie korektor adaptacyjny został zrealizowany sprzętowo przez rzeczywisty procesor DSP firmy Texas Instruments umieszczony w systemie uruchomieniowym DSK.

Słowa kluczowe: korekcja błędów dynamicznych czujnika, cyfrowy filtr adaptacyjny, algorytm RLS, procesor sygnałowy, Matlab.

Two-sensor-method in blind identification of sensor's dynamics using Matlab/Simulink and TMS320C6713 DSK

Abstract

In many cases dynamical error due to the sensor's dynamics is a considerable part of measurement error. Therefore one of measuring system objectives is to minimize this error. If the sensor's dynamics is time-varying due to dependence on the measured quantity then continuous identification of the dynamics' model and on-line error correction must be carried out. This task can be realized for example with use of a second sensor with different dynamical properties. In the paper there is presented an implementation of a two-sensor method in Matlab R14 and Simulink environment. The TMS320C6713 DSK system including Texas Instruments DSP processor, cooperating with Matlab/Simulink, is applied to correct the dynamical error due to sensor's dynamics.

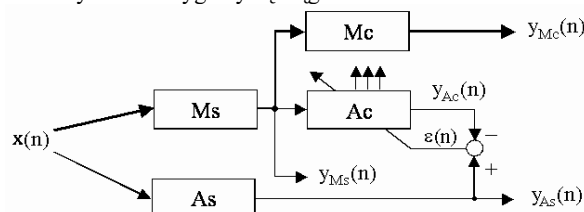
Keywords: sensor's dynamical error correction, digital adaptive filter, RLS algorithm, digital signal processor, Matlab.

1. Algorytm identyfikacji

W wielu przypadkach nie do pominięcia jest składowa błędu pomiaru związana z właściwościami dynamicznymi czujnika, nazywanymi w niniejszym artykule w skrócie „dynamiką czujnika”. Minimalizacja jej wpływu na dokładność pomiaru jest zadaniem stosunkowo prostym w sytuacji, gdy dynamika jest niezmienna, niezależna od warunków pomiaru. Rzeczywiste właściwości dynamiczne próbuje się wówczas przybliżyć modelem matematycznym o parametrach podawanych przez producenta czujnika lub zidentyfikowanych w warunkach laboratoryjnych. Jeśli dynamika czujnika zależy od innych, często

niemierzonych lub niemierzalnych czynników, wówczas niezbędna jest ciągła lub okresowa identyfikacja parametrów modelu w celu nadążnej korekcji błędów dynamicznych. Przykładowo, w pomiarach temperatury dynamikę czujnika kształtują takie czynniki jak: konstrukcja i wymiary czujnika, właściwości materiałów, z których został wykonany, rodzaj i prędkość medium, którego temperatura jest określana, zakres zmian temperatury i wartość chwilowa temperatury [1-3]. Zadanie ciągłej identyfikacji jest realizowalne np. dzięki zastosowaniu dodatkowego, pomocniczego czujnika o innych właściwościach dynamicznych niż czujnik główny. Takie podejście jest nazywane metodą dwóch czujników lub niekiedy korekcją „w ciemno”. Metoda zakłada, że oba czujniki pracują w identycznych warunkach pomiaru, podlegają temu samemu wymuszeniu i wzajemnie na siebie nie oddziałują. Pierwsze implementacje to elektromechaniczne lub elektroniczne rozwiązania analogowe, obecnie sukcesywnie wypierane przez rozwiązania dyskretne, realizowane w systemach z procesorem klasycznym, systemach wbudowanych z mikrokontrolerem lub specjalizowanym procesorem DSP [4-11].

Schemat blokowy układu działającego wg metody dwóch czujników przedstawia rys. 1, gdzie: $x(n)$ – sygnał wejściowy czujnika (wymuszenie), $y(n)$ – sygnał wyjściowy czujnika, M_s , M_c – czujnik główny i korektor główny, A_s , A_c – czujnik pomocniczy i korektor pomocniczy. W artykule założono, że dysponuje się wartościami sygnałów w chwili dyskretnej n , co pozwoliło na stosowanie opisu w dziedzinie zmiennej „ z ”, mimo iż w rzeczywistości sygnały są ciągłe w czasie.



Rys. 1. Adaptacyjna korekcja błędów dynamicznych metodą dwóch czujników
Fig. 1. Adaptive dynamical error correction using two-sensor-method

Pomijając inne składowe błędu pomiaru oraz zakładając, że $\varepsilon(n)=0$ i przyjęty model dynamiki czujnika dokładnie odzwierciedla dynamikę rzeczywistego czujnika, wobec tego sygnał $y_{M_c}(n)$ odtwarza wartość $x(n)$. W praktyce, efekt korekcji błędów dynamicznych będzie ograniczony, ze względu m.in. na niedoskonałość przyjętego modelu, nieliniowość zjawisk, obecność zakłóceń i konieczność ich filtracji. Tor przetwarzania: czujnik główny M_s – korektor pomocniczy A_c jest dostrojony z

pewnym błędem $\varepsilon(n) \approx 0$ do dynamiki czujnika pomocniczego A_s . Jeśli dynamikę czujników opisać modelami liniowymi:

$$G_{M_s}(z) = \frac{B_{M_s}(z)}{A_{M_s}(z)}, \quad G_{A_s}(z) = \frac{B_{A_s}(z)}{A_{A_s}(z)}, \quad (1)$$

to w stanie dostrojenia układu, gdy $\varepsilon(n) \approx 0$:

$$y_{A_s}(n) \approx y_{A_c}(n) \quad (2)$$

zachodzi:

$$G_{A_c}(z) = \frac{B_{A_c}(z)}{A_{A_c}(z)} \approx \frac{G_{A_s}(z)}{G_{M_s}(z)} = \frac{B_{A_s}(z)}{B_{M_s}(z)} \cdot \frac{A_{M_s}(z)}{A_{A_s}(z)}. \quad (3)$$

Dla następującej postaci wielomianów transmittancji dyskretnych czujników:

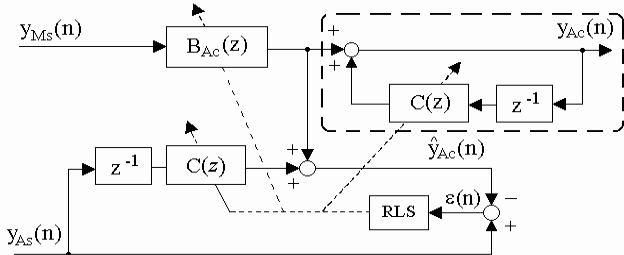
$$A_{M_s}(z) = a_0^{M_s} + a_1^{M_s}z^{-1} + \dots + a_N^{M_s}z^{-N}, \quad B_{M_s}(z) = 1, \quad (4)$$

$$A_{A_s}(z) = a_0^{A_s} + a_1^{A_s}z^{-1} + \dots + a_N^{A_s}z^{-N}, \quad B_{A_s}(z) = 1,$$

otrzymuje się jednoznaczne rozwiązanie problemu identyfikacji. Transmittancja korektora głównego $G_{M_c}(z)$ jest równa wówczas licznikowi transmittancji korektora pomocniczego. Przyjmując dodatkowe założenie o jednostkowym wzmocnieniu statycznym korektora głównego, otrzymuje się końcową zależność na współczynniki korektora głównego:

$$G_{M_c}(z) = B_{M_c}(z) = b_0^{M_c} + b_1^{M_c}z^{-1} + \dots + b_N^{M_c}z^{-N} = \frac{B_{A_c}(z)}{\sum_{i=0}^N b_i^{A_c}}. \quad (5)$$

Z równań (3) i (5) wynika, że korektor główny może być realizowany przez filtr cyfrowy N -tego rzędu o skończonej odpowiedzi impulsowej (w skrócie SOI), zaś korektor pomocniczy przez filtr o nieskończonej odpowiedzi impulsowej (w skrócie NOI). Filtr NOI można zbudować z dwóch filtrów SOI, tj. $B_{A_c}(z)$ i $C(z)$ połączonych jak na rys. 2 (przepływ sygnału od $y_{M_s}(n)$ do $y_{A_c}(n)$).



Rys. 2. Korektor pomocniczy realizowany przez filtr NOI złożony z filtrów SOI
Fig. 2. Auxiliary compensator realized by an IIR filter consisting of FIR filters

Filtr $B_{A_c}(z)$ realizuje licznik transmittancji $G_{A_c}(z)$, a filtr $C(z)$, umieszczony w pętli sprzężenia zwrotnego, jej mianownik. Konieczność zastosowania opóźnienia z^{-1} w ww. pętli, pociąga za sobą obniżenie rzędu filtru $C(z)$ do $N-1$:

$$C(z) = c_1 + c_2z^{-1} + \dots + c_Nz^{-(N-1)}. \quad (6)$$

Transmittancja powstałego filtru NOI może być zapisana w postaci:

$$G_{A_c}(z) = \frac{B_{A_c}(z)}{1 - C(z) \cdot z^{-1}} \equiv \frac{B_{A_c}(z)}{A_{A_c}(z)} \Rightarrow a_i^{A_c} = -c_i \quad i = 1, 2, \dots, N. \quad (7)$$

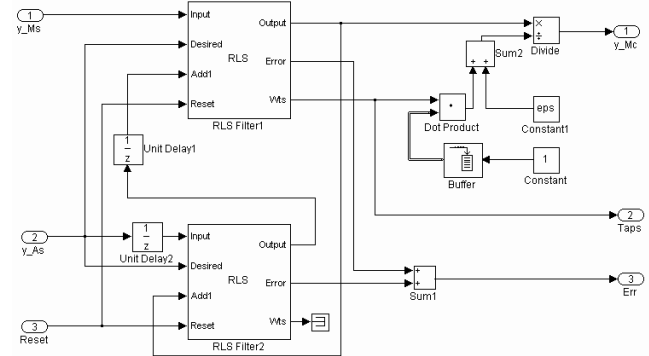
Próba zastosowania algorytmu RLS (*ang. Recursive Least Squares*) do aktualizacji współczynników filtru NOI (korektora pomocniczego) pociąga za sobą uciążliwe wyliczanie pochodnej cząstkowej sygnału wyjściowego filtru [12]. Można tego uniknąć wprowadzając drugi, identyczny jak w pętli sprzężenia zwrotnego, filtr $C(z)$. Uzyskuje się filtr, którego współczynniki są uaktualniane na bazie tzw. „błędów równania”:

$$\varepsilon(n) = y_{A_s}(n) - \hat{y}_{A_c}(n). \quad (8)$$

Sygnał $\hat{y}_{A_c}(n)$ jest wyznaczany z równania:

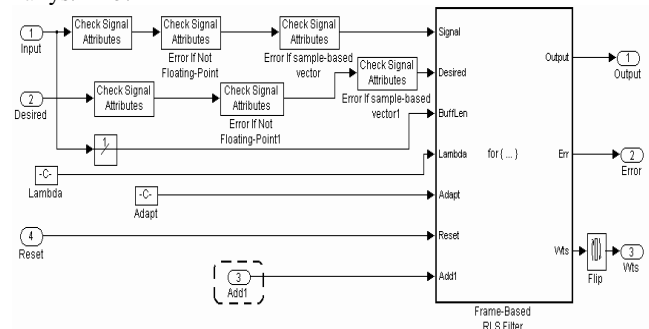
$$\begin{aligned} \hat{y}_{A_c}(n) &= \sum_{i=0}^N b_i^{A_c} \cdot y_{M_s}(n-i) + \sum_{i=1}^N c_i \cdot y_{A_s}(n-i) = \\ &= \sum_{i=0}^N b_i^{A_c} \cdot y_{M_s}(n-i) - \sum_{i=1}^N a_i^{A_c} \cdot y_{A_s}(n-i), \end{aligned} \quad (9)$$

w którym nie występuje rekursja, dzięki czemu można użyć algorytm RLS w klasycznej postaci jak dla filtru SOI [11-13]. Sygnał $y_{A_c}(n)$ nie jest używany w algorytmie identyfikacji, stąd elementy objęte linią przerywaną na rys. 2. mogą być usunięte. Korektor adaptacyjny zaimplementowany w środowisku Matlab/Simulink R14 pokazano na rys. 3. Bloki „RLS Filter1”, „RLS Filter2”, „Unit Delay2” oraz „Sum1” tworzą korektor pomocniczy. Blok „Unit Delay1” został dodany, aby usunąć powstającą pętlę algebraiczną. Pozostałe elementy z tego rysunku realizują korektor główny. Jak można zauważyć z rys. 2, wartość błędów $\varepsilon(n)$ jest wyznaczana jako różnica sygnału $y_{A_s}(n)$ i sumy sygnałów wyjściowych filtrów $B_{A_c}(z)$ oraz $C(z)$.

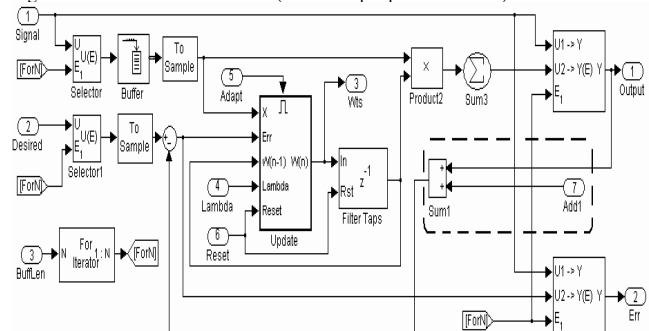


Rys. 3. Implementacja cyfrowego korektora dynamiki czujników
Fig. 3. Implementation of discrete time compensator of the sensors' dynamics

Aby zaimplementować tę sumę należało wprowadzić drobne modyfikacje do oryginalnego bloku „RLS Filter1”, co pokazano na rys. 4 i 5.



Rys. 4. Zmodyfikowany „RLS Filter1” (dodano port wyjściowy „Add1”)
Fig. 4. Modified “RLS Filter1” (“Add1” input port was added)



Rys. 5. Zmodyfikowany blok „Frame-Based RLS Filter1” (dodano „Add1” i „Sum1”)
Fig. 5. Modified “Frame-Based RLS Filter1” (“Add1” input port and “Sum1” were added)

Identyczne zmiany wprowadzono w drugim filtrze RLS o nazwie „RLS Filter2” występującym na rys. 3.

2. Model czujnika

Przykładowo dynamika czujnika może być modelowana nieliniowym równaniem różniczkowym 1-go rzędu, w którym parametr dynamiki $T(y(t))$ jest zależny od wartości chwilowej sygnału wyjściowego $y(t)$ [2,3]:

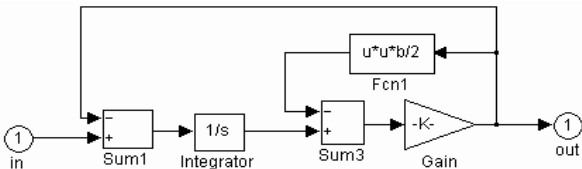
$$T(y(t)) \cdot \frac{dy(t)}{dt} + y(t) = x(t) \quad (10)$$

W równaniu tym przyjęto: t – czas, $x(t)$ – sygnał wejściowy czujnika (wymuszenie), $y(t)$ – sygnał wyjściowy czujnika. Zależność $T=f(y(t))$ może być dowolna. Dla czujników temperatury w ustalonych warunkach wymiany ciepła zazwyczaj wystarczające jest przybliżenie liniowe:

$$T_{Ms}(y_{Ms}) = T_p + b \cdot y_{Ms} \quad (11a)$$

$$T_{As}(y_{As}) = c \cdot T_p + c \cdot b \cdot y_{As} \quad (11b)$$

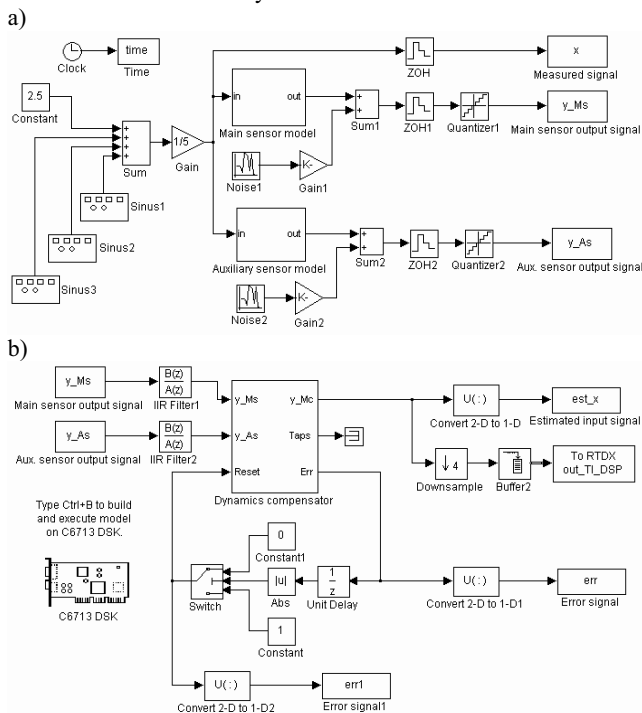
gdzie: T_p – składowa niezależna parametru dynamiki, b – współczynnik nieliniowości. Parametr c określa ile razy wolniejszy jest czujnik pomocniczy w porównaniu do czujnika głównego. Takie podejście nie uwzględnia wpływu zmiany dynamiki czujników wywołane zmianami stanu środowiska, np. prędkości przepływu medium, którego temperatura jest mierzona. Warunek $c>1$ musi być spełniony, aby zadość uczynić założeniom metody dwóch czujników. Rysunek 6 pokazuje implementację wzorów (10) i (11a) w środowisku Matlab/Simulink. Dla przypadku $b=0$ otrzymuje się klasyczny model inercji pierwszego rzędu ze stałą czasową T_p .



Rys. 6. Nieliniowy model dynamiki czujnika
Fig. 6. Model of non-linear sensor's dynamics

3. Wyniki symulacji

Rysunek 7a przedstawia część 1 układu pomiarowego. Wymuszeniem jest suma trzech sinusoid (bloki "Sinus1", "Sinus2", "Sinus3") z jednostkową amplitudą i częstotliwościami odpowiednio 30 Hz, 60 Hz i 90 Hz. Amplituda wymuszenia została znormalizowana do zakresu $\langle 0,1 \rangle$ (bloki "Constant" i "Gain"). Modele czujników umieszczono w podsystemach "Main sensor model" i "Auxiliary sensor model".



Rys. 7. Model układu pomiarowego: a) – część 1: wymuszenie, czujniki, źródło szumu, przetwornik A/C; b) – część 2: filtry odszumiające, korektor dynamiki, system resetu, odczyt danych z pamięci DSP do zmiennych Matlaba
Fig. 7. Model of measurement path: a) – part 1: input signal, sensors, noise sources, ADC S&H and quantization; b) – part 2: noise filtering, dynamics compensator, reset subsystem, data reading from the DSP memory to the Matlab

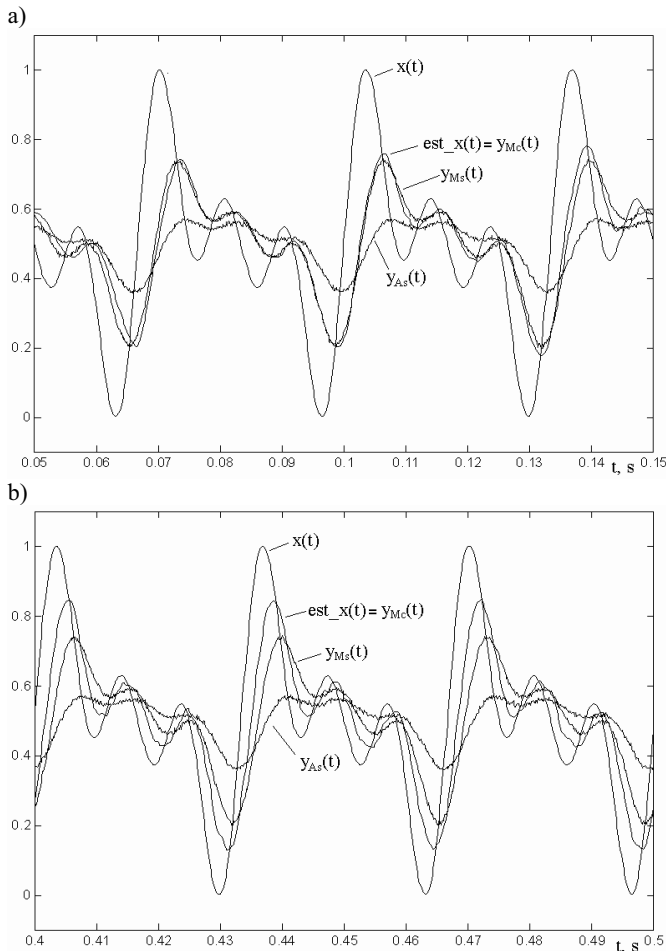
Użyto bloków „Noise” do zamodelowania błędów występujących w rzeczywistych pomiarach. Przyjęto rozkład normalny błędu (symulowanego przez zmienną losową) o zerowej średniej $\mu=0$ i odchyleniu standardowym $\sigma=(1/3)\cdot 0.01$. Zerowa wartość średniej symuluje brak błędu systematycznego, a dobrana wartość σ , zapewnia występowanie błędu przypadkowego o amplitudzie przekraczającej 1% wartości maksymalnej amplitudy wymuszenia, tu równej 1, z prawdopodobieństwem mniejszym niż 1%, ponieważ dla rozkładu normalnego przedział $\mu\pm 3\sigma$ obejmuje nieco ponad 99% rozkładu. Sygnał losowy nałożono na sygnały z czujników. Przyjęto następujące wartości parametrów modelu czujników opisanych zależnością (11): $T_p=5$ ms, $c=3$, $b/T_p=0.3$. Również zasymulowano działanie przetwornika A/C, tj. 16 bitową kwantyzację oraz funkcję S&H. Sygnały w postaci dyskretyzowanych próbek zostały zapisane jako zmienne w Matlabie.

W drugiej części układu (rys. 7b) sygnały z obu czujników zostały poddane odsumowaniu za pomocą dolnoprzepustowego filtra Butterwortha III-go rzędu (bloki „IIR Filter1” i „IIR Filter2”) o stosunku, częstotliwości -3 dB do połowy częstotliwości próbkowania równym $f_{-3dB}/(1/2\cdot f_s)=0,1$. Szerszą dyskusję wpływu typu i rzędu filtra odszumiającego na efekt korekcji dynamiki oraz propagację zakłóceń podjęto w pracy [11]. Częstotliwość próbkowania f_s przyjęto 8 kHz, zgodną, co do wartości z jedną z standardowych częstotliwości przetwornika A/C systemu TMS320C6713 DSK. Pozostałe elementy z rys. 7b to: korektor dynamiki z cyfrowymi filtrami adaptacyjnymi RLS rzędu $N=1$ i współczynnikiem zapomnienia $\lambda=0.98$, system resetu i odczyt danych z pamięci procesora DSP do zmiennej Matlaba poprzez kanał czasu rzeczywistego, tzw. RTDX.

Oprócz standardowej symulacji działania układu pomiarowego w środowisku Matlab za pomocą przyborka Embedded Target for TI C6000 DSP można wygenerować i przesłać wykonywalny kod maszynowy dla procesora DSP firmy Texas Instruments. Dostarczone narzędzia programistyczne umożliwiają dwustronną komunikację procesora DSP ze środowiskiem Matlaba za pośrednictwem interfejsu USB. Kolejną bardzo użyteczną funkcją przyborka jest odczyt wartości zmiennych, co umożliwia analizę sygnałów w czasie rzeczywistym. Należy zaznaczyć, że do generacji kodu niezbędny jest zewnętrzny kompilator C, zawarty np. w pakiecie Microsoft Visual Studio oraz narzędzie firmowe TI Code Composer Studio, które w ograniczonej wersji, dostarczane jest wraz z systemem uruchomieniowym TMS320C6713 DSK. Szczegółowe informacje na temat mechanizmów tworzenia kodu i uruchamiania aplikacji na wspomnianym procesorze można znaleźć w dokumentacji Matlaba [14-17].

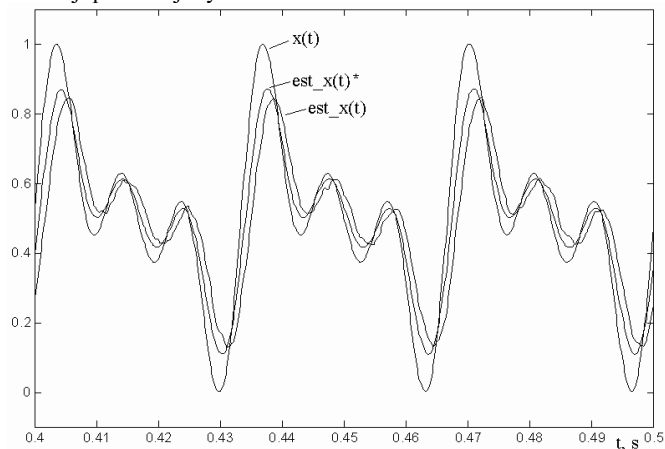
Użyte środowisko programistyczne umożliwia uruchomienie, wstrzymanie wykonania kodu w procesorze DSP oraz wydobywanie danych z pamięci systemu DSK za pośrednictwem kanału czasu rzeczywistego, skojarzonego na rys. 7b ze zmienną „out_TI_DSP”. Dane te można następnie przetwarzać w dowolnych celach, np. przedstawić przebiegi sygnałów na monitorze komputera PC, wyznaczyć wartości błędów przed i po korekcji. Zmienna „out_TI_DSP” zawiera decymowane (co czwarta próbka) wartości „est_x” przesłane z systemu DSK.

Otrzymane wyniki korekcji przedstawia rys. 8. Sygnał wyjściowy czujników ma słabiej amplitudę i przesuniętą fazę w stosunku do sygnału wymuszenia. Oznacza to występowanie błędów pomiaru wartości chwilowej wymuszenia związanego z dynamiką czujników. Kilka początkowych okresów sygnałów (rys. 8a) obrazuje proces strojenia filtrów RLS zawartych w strukturze korektora dynamiki. Po pewnym czasie (rys. 8b), od momentu uruchomienia korektora, sygnał oznaczony jako $est_x(t)$ (tożsamy sygnałowi wyjściowemu korektora głównego z rys. 1) jest bliższy sygnałowi wymuszenia $x(t)$ niż sygnały z czujników $y_{Ms}(t)$ i $y_{As}(t)$, co oznacza częściową, ale znaczną redukcję błędów dynamicznego.



Rys. 8. Przebieg sygnałów w dwóch przedziałach czasowych, gdzie: $x(t)$ – wymuszenie, $y_{Me}(t)$ – sygnał wyjściowy czujnika głównego, $y_{Aa}(t)$ – sygnał wyjściowy czujnika pomocniczego, $est_x(t)$ – estymowany sygnał wymuszenia po korekcji błędów dynamicznych czujnika głównego
 Fig. 8. Signals in two time intervals, where: $x(t)$ – input signal, $y_{Me}(t)$ – main sensor output signal, $y_{Aa}(t)$ – auxiliary sensors output signal, $est_x(t)$ – estimated input signal after dynamics' compensation

Idealna korekcja błędów dynamicznych jest trudna do osiągnięcia z powodu szumu, konieczności stosowania filtru odszumiającego o własnej dynamice, liniowości korektorów próbujących przybliżyć nieliniową dynamikę czujników. Wpływ filtracji odszumiającej i obecności błędów przypadkowych na efekt korekcji prezentuje rys. 9.



Rys. 9. Porównanie sygnału wymuszenia $x(t)$ z sygnałem po korekcji błędów dynamicznych dla dwóch przypadków: $est_x(t)$ – identycznie jak na rys. 8b i $est_x(t)^*$ – brak błędów przypadkowych i filtracji odszumiającej
 Fig. 9. Comparison of signals in two cases, where: $x(t)$ – input signal, $est_x(t)$ – estimated input signal after dynamics' compensation presented in Fig. 8b, $est_x(t)^*$ – without noises and de-noising filters

Na rys. 9 można zauważyć, że sygnał $est_x(t)^*$ w przypadku braku zakłóceń i filtrów odszumiających w porównaniu do $est_x(t)$, czyli sygnału dla warunków symulacji jak z rys 8b, jest bliższy sygnałowi wymuszenia $x(t)$. Pozostałe parametry symulacji jak: model czujnika, nastawy korektora, itp. zachowano bez zmian.

4. Podsumowanie

W artykule przedstawiono szczegóły implementacji algorytmu identyfikacji adaptacyjnej dynamiki czujników metodą dwóch czujników w środowisku Matlab współpracującym z systemem uruchomieniowym TMS320C6713 DSK. Uzyskano pozytywne rezultaty badań modelowych korekcji błędu dynamicznego dla nieliniowych modeli czujników temperatury przybliżanych na bieżąco modelami liniowymi. Kolejnym etapem badań będzie weryfikacja skuteczności algorytmu na sygnałach z rzeczywistych czujników, podłączonych do wejścia A/C systemu DSK.

5. Literatura

- [1] Bernhard F.: Technische Temperaturmessung-physikalische und meßtechnische Grundlagen, Sensoren und Meßverfahren, Meßfehler und Kalibrierung. Springer-Verlag Berlin, Heidelberg, New York, 2003.
- [2] Minkina W.: Non-linear models of temperature sensor dynamics, Sensors and Actuators A: Physical Vol. A 30, 1992, p. 209-214.
- [3] Minkina W.: Theoretical and experimental identification of the temperature sensor unit step response non-linearity during air temperature measurement, Sensors and Actuators A: Physical Vol. A 78, 1999, p. 81-87.
- [4] Gryś S., Minkina W.: Fast temperature determination using two thermometers with different dynamical properties, Sensors & Actuators A: Physical Vol. A 100, 2002, p. 192-198.
- [5] Minkina W., Gryś S.: Dynamics of contact thermometric sensors with electric output and methods of its improvement, Metrology and Measurement Systems PAN, Vol. XII, 4/2005, p. 371-391.
- [6] Bisztyga B., Nabielec J.: Zastosowanie metod rekurencyjnych do korekcji „w ciemno” w pomiarach sygnałów okresowych, PAK nr 9bis/2006, str. 80-83.
- [7] Nalepa J.: Próba porównania metod optymalizacji parametrycznej stosowanych w badaniach symulacyjnych korekcji błędu dynamicznego metodą „w ciemno”, PAK nr 9bis/2006, str. 155-157.
- [8] Nabielec J., Jurkiewicz J.: Rozszerzenie korekcji dynamicznej niestacjonarnego systemu pomiarowego I-rzędu metodą „w ciemno” o autoidentyfikację parametrów statycznych, PAK nr 9bis/2006, str. 183-185.
- [9] O'Reilly P., Kee R., Fleck R., McEntee P.: Two-wire thermocouples: A nonlinear state estimation approach to temperature reconstruction, Review of Scientific Instruments 72(8), 2001, p. 3449-3457.
- [10] Kar K., Roberts S., et al.: Instantaneous exhaust temperature measurements using thermocouple compensation techniques, SAE World Congress, March 8-11, Detroit, 2004.
- [11] Minkina W., Gryś S.: Korekcja charakterystyk dynamicznych czujników termometrycznych – metody, układy, algorytmy, Wyd. Politechniki Częstochowskiej, Częstochowa, 2004.
- [12] Rutkowski L.: Filtry adaptacyjne i adaptacyjne przetwarzanie sygnałów, WNT, Warszawa, 1994.
- [13] Zieliński T. Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań, Wyd. WKiŁ, Warszawa 2005.
- [14] Signal Processing Blockset version 6.0. User's Guide, MathWorks Inc., 2004.
- [15] Link for Code Composer Studio® Development Tools version 1.3.1. User's Guide, MathWorks Inc., 2004.
- [16] Real Time Workshop version 6.0. User's Guide, MathWorks Inc., 2004.
- [17] Embedded Target for TI C6000 DSP version 2.1, User's Guide, MathWorks Inc., 2004.