



Politechnika Łódzka

ZESZYTY NAUKOWE

ELEKTRYKA

Nr 123

ŁÓDŹ 2011

ZESZYTY NAUKOWE POLITECHNIKI ŁÓDZKIEJ
SCIENTIFIC BULLETIN OF THE TECHNICAL UNIVERSITY OF LODZ
BULLETIN SCIENTIFIQUE
DE L'UNIVERSITÉ POLYTECHNIQUE DE LODZ
НАУЧНЫЕ ЗАПИСКИ
ЛОДЗИНСКОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА
WISSENSCHAFTLICHE HEFTE
DER TECHNISCHEN UNIVERSITÄT IN LODZ

Redaktor Działu: **prof. dr hab. inż. Piotr Ostalczyk**
Sekretarz Techniczny: **dr inż. Janusz Kacerka**

© Copyright by Politechnika Łódzka 2011

Adres Redakcji – Адрес Редакции – Editor's Office
Adresse de Redaction – Schriftleitungsadresse:

WYDAWNICTWO POLITECHNIKI ŁÓDZKIEJ
90-924 Łódź, ul. Wólczańska 223
tel./fax 42-684-07-93
e-mail: a-row-1@adm.p.lodz.pl
www.wydawnictwa.p.lodz.pl

ISSN 0374-4817

Nakład 80 egz. Ark. druk. 6,0. Papier offset. 80 g, 70 x 100
Druk ukończono w marcu 2012 r.
Wykonano w Drukarni Offsetowej „Quick-Druk” s.c., 90-652 Łódź, ul. Łąkowa 11

CONTENTS

Paweł Adameczyk – Optimal Sliding Mode Control of an Electrohydraulic Servomechanism.....	5
Optymalne ślizgowe sterowanie elektrohydraulicznym serwomechanizmem pozycyjnym	
Wojciech Błaśiński, Zbigniew Nowacki – Układ hamowania elektrycznego do badania napędów.....	17
Electric Brake System for Drive Testing	
Krzysztof Kuźmiński, Maciej Krawiecki – Dyskusja warunków przesuwalności biegunów układów liniowych za pomocą statycznych sprzężeń zwrotnych od wyjścia	27
Discussion of Pole Assignability Conditions for Linear Systems Using Static Output Feedback	
Zbigniew A. Nowacki – File Duplicator for the Microsoft Windows Operating System.....	39
Powielacz plików: przykład usług systemu Windows	
Grzegorz Wasiak – Algorytm przekształcania schematów blokowych w programie AutoCAD	51
Algorithm for the Transform of Block Diagrams in AutoCAD	

Global Summaries of PhD Theses Autoreferaty prac doktorskich

Kamil Grabowski – Person Identification Based on Iris Image Analysis with Special Emphasis On Hardware Implementation – Statistical Analysis	67
Identyfikacja osób na podstawie analizy obrazu tęczówki oka ze szczególnym uwzględnieniem możliwości implementacji sprzętowej	
Cezary Maj – Modelling and Optimization of Electrostatic Membrane-Based Actuators	77
Modelowanie i optymalizacja mikrosystemów sterowanych siłą elektrostatyczną opartych na membranie	

PAWEŁ ADAMCZYK

Technical University of Łódź
Institute of Automatic Control

OPTIMAL SLIDING MODE CONTROL OF AN ELECTROHYDRAULIC SERVOMECHANISM

Reviewer: Prof. dr hab. Zbigniew Nowacki

Manuscript received: 11.12.2011

In this paper sliding mode control with acceleration constraint is proposed to achieve robustness and optimal performance of electrohydraulic servomechanism. The main feature of the proposed control is application of a time-varying switching plane. Initially, the plane passes through the system representative point in the error state space and then it moves with a constant velocity to the origin of the space. Having reached the origin the plane stops moving and remains fixed. The plane parameters (determining angles of inclination and the velocity of its motion) are selected to ensure the minimum integral absolute error without violating acceleration constraints. The simulation results show, that the proposed control ensures accuracy, and robust performance.

1. INTRODUCTION

In recent years much of the research in the area of control systems theory focused on the design of a discontinuous feedback which switches the structure of the system according to the evolution of its state vector. This technique, usually called sliding mode control, provides an effective and robust means of controlling nonlinear plants [7, 8, 12, 19, 22]. The main advantage of this technique is that once the system state reaches a sliding surface, the system dynamics remain insensitive to a class of parameter variations and disturbances.

However, robust tracking is assured only after the system state hits the sliding surface, i.e. the robustness is not guaranteed during the reaching phase. In order to overcome these problems the idea of the time-varying switching lines applied for the sliding mode control of the second order systems was introduced in [1, 2, 5, 10, 11, 14, 17, 18, 20]. The sliding mode control of the third order, nonlinear and time-varying system was considered in [3, 4, 5, 6, 15, 16]. On the other hand, sliding mode controllers for electrohydraulic servomechanisms were considered in [9, 13].

In this paper the sliding mode control with time-varying switching plane for the same plant is proposed. The switching plane has constant angles of inclination and originally passes through the system representative point in the error state space. Initially the plane moves with a constant velocity to the origin of the space. Once the plane reaches the origin, it stops moving and remains fixed (time-invariant). The switching plane parameters are selected to ensure the minimization of the integral absolute error in the controlled system subject to the acceleration constraint. The proposed approach ensures favourable dynamic properties of the proposed control system and – more importantly – its robustness with respect to the external load torque from the very beginning of the control process.

2. SYTEM DESCRIPTION

The plant considered in this paper is an electrohydraulic position servo system. It usually consists of two components: hydraulic actuator and servo valve. The actuator may be linear or rotary. Linear actuator is the cylinder with moving piston, which slides inside the cylinder and divides the actuator into two chambers. Piston movement is caused by the pressure difference between the two chambers. Typical applications of linear actuators include hydraulic presses and injection moulding machines. However, since rotary actuators are probably more popular than the linear ones, in this paper an electrohydraulic position servo system with rotary actuator is considered. The rotary actuators can be classified as gear, piston, vane or deri sine type. Further in this paper gear and semi rotary piston actuators will be presented and gear motor will be considered in detail.

In gear motors, hydraulic fluid follows from inlet to outlet around the outside of the gears in the space between the teeth. One of the two gears is coupled to the output shaft. This is schematically shown in Fig. 1.

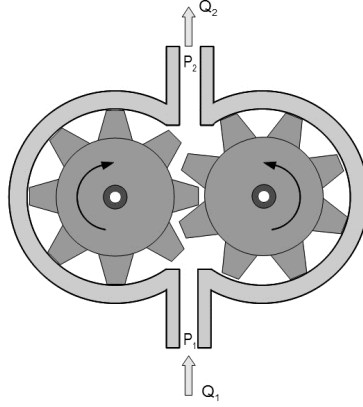


Fig. 1. Schematic diagram of gear hydraulic motor

Semi rotary piston actuator produces very large torque at low speed. The fluid enters into the chamber between fixed and moving vanes. Moving vane is coupled with the output shaft. This type of actuators can have one or more sets of vanes. In Fig. 2 actuators with two sets (a) and one set of vanes (b) are shown. They provide approximately 180° and less than 300° of rotation range respectively.

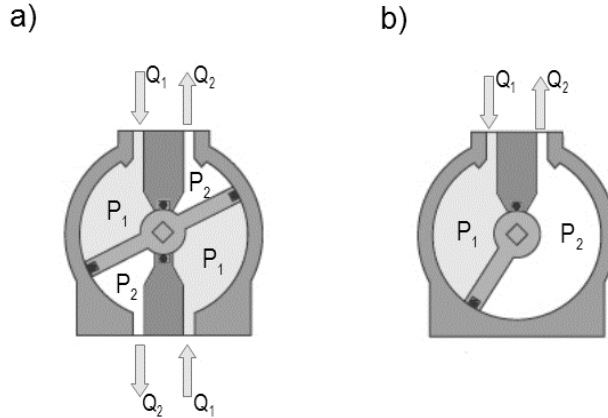


Fig. 2. Schematic diagram of semi rotary piston actuators

Equations, describing the considered hydraulic motor can be expressed as follows. The motor chamber equation can be derived using flow continuity property

$$Q_l = D_m \cdot \dot{\theta}_m + C_{tm} \cdot P_l + (V_t/4 \cdot \beta_e) \cdot \dot{P}_l \quad (1)$$

Furthermore, the torque balance equation

$$P_l \cdot D_m = J_t \cdot \ddot{\theta}_m + B_m \cdot \dot{\theta}_m + G \cdot \theta_m + T_l \quad (2)$$

where:

D_m – volumetric displacement of motor,	Q_l – load flow,
C_{tm} – total leakage coefficient,	θ_m – angular position of motor shaft,
β_e – effective bulk modulus,	T_l – load torque,
V_t – total compressed volume	J_t – total motor and load inertia,
B_m – viscous damping coefficient	G – torsional spring gradient.
$P_l = P_1 - P_2$ – load pressure,	

Another significant component of the considered system is the electrohydraulic servo valve. Its role is to control pressurized fluid flow in order to move the actuator towards the desired position. The input of the servo valve is an electric current, supplied by the amplifier. The current powers an electromagnetic torque motor which in turn moves the spool valve. The spool valve allows fluid to pass from supply via actuator to the return with a controlled flow rate. Shifting the spool direction enables the fluid flow direction change. Since the relation between spool valve displacement and the control voltage is approximately linear the control voltage is usually taken as the input signal. The electrohydraulic servo valve is schematically presented in Fig. 3. Symbols in this figure correspond to the symbols used earlier in this chapter in Figs. 1 and 2.

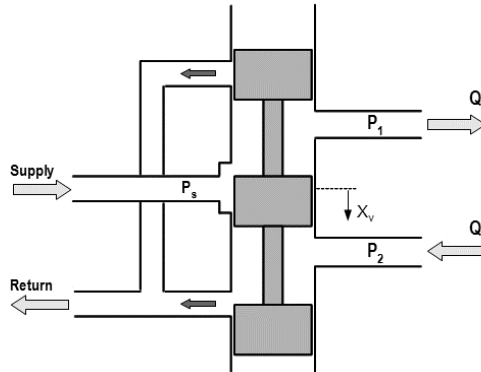


Fig. 3. Schematic representation of electrohydraulic servo valve

The following equations describe electrohydraulic servo valve. The relation between spool displacement and the load flow rate may be represented as

$$Q_l = K_q \cdot X_v - K_c \cdot P_l \quad (3)$$

The valve flow gain, depends on operating point

$$K_q = C_d \cdot W \cdot \sqrt{[P_s - \text{sgn}(X_v) \cdot P_l] / \rho} \quad (4)$$

where:

K_c – flow – pressure coefficient,

K_q – servovalve gain,

W – area gradient,

X_v – valve displacement,

P_s – supply pressure,

ρ – fluid mass density,

C_d – discharge coefficient.

Combining equations 1-4, describing both actuator and servo valve dynamics, we get the following state equations of the system

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= x_3(t) \\ \dot{x}_3(t) &= -a_1 \cdot x_1(t) - a_2 \cdot x_2(t) - a_3 \cdot x_3(t) + b(\mathbf{x}, t) \cdot u(t) - d(t) \end{aligned} \quad (5)$$

where:

$$\begin{aligned} \mathbf{x}(t) &= [x_1(t) \quad x_2(t) \quad x_3(t)]^T = [\theta_m(t) \quad \dot{\theta}_m(t) \quad \ddot{\theta}_m(t)]^T \\ a_1 &= (4 \cdot \beta_e / V_t) \cdot (K_{ce} / J_t) \cdot G \\ a_2 &= G / J_t + (4 \cdot \beta_e / V_t) \cdot (D_m^2 / J_t) + (4 \cdot \beta_e / V_t) \cdot (K_{ce} / J_t) \cdot B_m \\ a_3 &= B_m / J_t + (4 \cdot \beta_e / V_t) \cdot K_{ce} \\ b(\mathbf{x}, t) &= (4 \cdot \beta_e / V_t) \cdot (D_m / J_t) \cdot K_q \cdot K_v \\ d(t) &= (4 \cdot \beta_e / V_t) \cdot (K_{ce} / J_t) \cdot T_l + (1 / J_t) \cdot \dot{T}_l \end{aligned} \quad (6)$$

In these equations $K_{ce} = K_c + C_{tm}$ is the total flow–pressure coefficient and K_v represents the servovalve amplifier gain. From the above equations it can be seen, that the considered system is nonlinear and subject to the external

disturbance so it requires a robust and possibly nonlinear control scheme. In this paper the sliding mode control scheme with time varying switching plane is proposed.

3. CONTROL SYSTEM

The trajectory tracking error is defined by the following vector $\mathbf{e}(t) = [e_1(t) \ e_2(t) \ e_3(t)]^T = \mathbf{x}(t) - \mathbf{x}_d(t)$. Hence, we have $e_1(t) = x_1(t) - x_{1d}(t)$, $e_2(t) = x_2(t) - x_{2d}(t)$, $e_3(t) = x_3(t) - x_{3d}(t)$. In this paper it is assumed that at the initial time $t = t_0$, the tracking error and the error derivatives $e_1(t_0) = e_0$, $e_2(t_0) = 0$, $e_3(t_0) = 0$. Let us consider a time varying switching plane with the constant angle of inclination. Originally the plane moves uniformly (i.e. with a constant velocity) in the state space and then it stops at the time instant t_f . Consequently, for any $t \leq t_f$ the switching plane is described by the following equation

$$s(\mathbf{e}, t) = 0, \text{ where } s(\mathbf{e}, t) = e_3(t) + c_2 \cdot e_2(t) + c_1 \cdot e_1(t) + A + B \cdot t \quad (8)$$

where c_1, c_2, A and B are some constants. The selection of these constants will be considered further in this section. Since the plane stops at the time t_f , for any $t \geq t_f$

$$s(\mathbf{e}, t) = 0, \text{ where } s(\mathbf{e}, t) = e_3(t) + c_2 \cdot e_2(t) + c_1 \cdot e_1(t) \quad (9)$$

First, the constants c_1, c_2, A and B should be chosen in such a way that the representative point of the system at the initial time $t = t_0$ belongs to the switching plane. For that purpose, the following condition must be satisfied

$$s(\mathbf{e}(t_0), t_0) = e_3(t_0) + c_2 \cdot e_2(t_0) + c_1 \cdot e_1(t_0) + A + B \cdot t_0 \quad (10)$$

Notice that the input signal

$$u = \frac{-f(\mathbf{x}, t) - c_2 \cdot e_3(t) - c_1 \cdot e_2(t) + \dot{x}_{3d}(t) - B - \gamma \cdot \text{sgn}[s(\mathbf{e}, t)]}{b(\mathbf{x}, t)} \quad (11)$$

where $\gamma = \eta + \mu$ and η is a strictly positive constant, ensures the stability of the sliding motion on the switching plane (8). We demand tracking error to be an aperiodic function of time, i.e. we wish to avoid undesirable overshoots and oscillations. Furthermore, since we require the system to exhibit good dynamic properties, our objective is to minimize the control quality criterion

$$J = \int_{t_0}^{\infty} |e_1(t)| dt \quad (12)$$

subject to acceleration constraint.

According to [4] the optimal switching plane parameters subject to acceleration constraint are given as

$$B = [2(a_{\max})^{3/2} \cdot \text{sgn}(e_0)] / \sqrt{|e_0|} \quad (13)$$

$$c_1 = (4 \cdot B / e_0)^{2/3} \quad (14)$$

$$A = -c_1 \cdot e_0 \quad (15)$$

$$c_2 = 2 \cdot \sqrt{c_1} \quad (16)$$

$$t_f = (e_0 \cdot c_1) / B \quad (17)$$

where $a_{\max}$ is the maximum admissible acceleration value.

4. SIMULATION RESULTS

We simulated dynamic behaviour of the electrohydraulic servomechanism, controlled by the presented algorithm. Parameters of the system are given in the following table [13].

PARAMETER	VALUE	UNIT
P_s	13.79	MPa
K_q	$6.45 \cdot \sqrt{[P_s - \text{sgn}(X_v) \cdot P_l]} / \rho \cdot 10^{-6}$	$\text{m}^2 \cdot \text{s}^{-1}$
β_e	344.74	MPa
V_t	1.64	m^3
K_{ce}	$1.13 \cdot 10^{-6}$	$\text{m}^3 \cdot \text{s} \cdot \text{Pa}^{-1}$
D_m	$8.19 \cdot 10^{-6}$	$\text{m}^3 \cdot \text{rad}^{-1}$
J_t	$5.65 \cdot 10^{-2}$	$\text{kg} \cdot \text{m}^2$
B_m	8.47	$\text{N} \cdot \text{m} \cdot \text{s}$
K_v	0.51	$\text{m} \cdot \text{V}^{-1}$
G	$1.13 \cdot 10^{-3}$	$\text{N} \cdot \text{m} \cdot \text{rad}^{-1}$

The maximum admissible actuator acceleration value $a_{\max} = 10 \text{ rad/s}^2$. Initial actuator position $x_1(t_0) = 10 \text{ rad}$, and the main goal to achieve is to set the final position to zero. It makes tracking error e_1 equal to x_1 . From equations (13) – (17) we get the following switching plane parameters: $c_1 \approx 10.87 \text{ 1/s}^2$, $c_2 \approx 6.594 \text{ 1/s}$, $B \approx 89.63 \text{ rad/s}^3$, $A \approx -108.7 \text{ rad/s}^2$ and the stop time $t_f \approx 1.213 \text{ s}$. The load torque is the sine like function with amplitude $2.825 \text{ N}\cdot\text{m}$. According to this load we choose the coefficient $\gamma = 10^4 \text{ rad/s}^3$. In order to avoid the chattering, we replace sign function in equation (11) with saturation function with the threshold value equal to 0.2 rad/s^2 . Simulation results are shown in figures 4 – 6.

Figure 4 shows actuator position and velocity. Figure 5 presents its acceleration and figure 6 demonstrates the control voltage. It can be seen from these figures that the acceleration constraint $a_{\max} = 10 \text{ rad/s}^2$ is satisfied and the actuator position converges to zero without oscillations. The system is also insensitive with respect to external disturbance (load torque) from the very beginning of the control process.

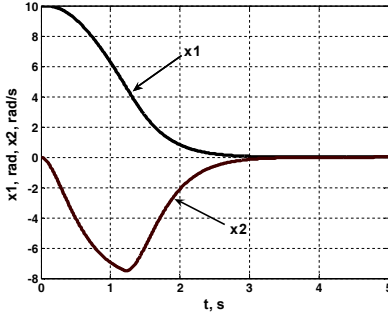


Fig. 4. Actuator position (x_1) and velocity (x_2)

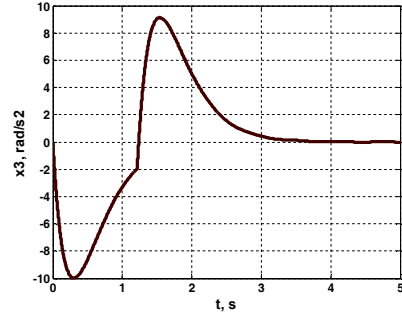


Fig. 5. Actuator acceleration

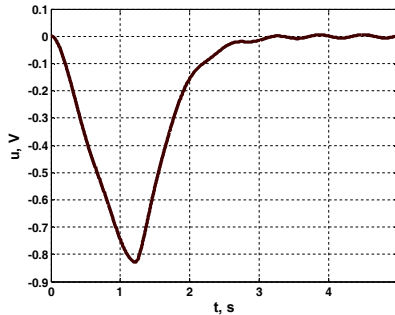


Fig. 6. Control voltage

5. CONCLUSIONS

In this paper sliding mode control with time varying switching plane for electrohydraulic servomechanism has been considered. The switching plane parameters are selected to ensure the minimum integral absolute error without violating acceleration constraint. It has been demonstrated that applying time varying switching plane makes the system insensitive to external disturbance from the very beginning of the control action. Furthermore, the proposed control scheme assures good system dynamics and its non-oscillatory response.

6. ACKNOWLEDGEMENT

This work has been performed in the framework of a project "Optimal sliding mode control of time delay systems" financed by the National Science Centre of Poland – decision number DEC 2011/01/B/ST7/02582.

REFERENCES

- [1] **Bartoszewicz A.:** A comment on 'A time-varying sliding surface for fast and robust tracking control of second-order uncertain systems'. *Automatica*, Vol. 31, 1995, pp. 1893-1895.
- [2] **Bartoszewicz A.:** Time-varying sliding modes for second-order systems. *IEE Proceedings on Control Theory and Applications*, Vol. 143, 1996, pp. 455-462.
- [3] **Bartoszewicz A., Nowacka A.:** Switching plane design for the sliding mode control of systems with elastic input constraint. *Proceedings of IMechE*, Vol. 219, 2005, pp. 393-403.
- [4] **Bartoszewicz A., Nowacka A.:** Reaching phase elimination in variable structure control of the third order system with state constraints. *Kybernetika*, Vol. 42, 2006, pp. 111-126.
- [5] **Bartoszewicz A., Nowacka-Leverton A.:** Time-varying sliding modes for second and third order systems, Springer-Verlag, Berlin Heidelberg, 2009 (ISBN 978-3-540-92216-2).
- [6] **Bartoszewicz A., Nowacka-Leverton A.:** ITAE optimal sliding modes for third order systems with input signal and state constraints. *IEEE Transactions on Automatic Control*, Vol. 55, 2010, pp. 1928-1932.
- [7] **Brock S., Deskur J., Zawirski K.:** Modified sliding – mode speed controller for servo drives. *Proceedings of the IEEE International Symposium on Industrial Electronics*, Vol. 2, 1999, pp. 635-640.
- [8] **Brock S., Deskur J., Zawirski K.:** Robust speed and position control of PMSM. *Proceedings of the IEEE International Symposium on Industrial Electronics*, Vol. 2, 1999, pp. 667-672.
- [9] **Chen H.-M., Renn J.-C., Su J.-P.:** Sliding mode control with varying boundary

- layers for an electro-hydraulic position servo system. *International Journal of Advanced Manufacturing Technology*, Vol. 26, 2005, pp. 117-123.
- [10] **Choi S.B., Park D. W.:** Moving sliding surfaces for fast tracking control of second-order dynamic systems. *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 116, 1994, pp. 154-158.
- [11] **Choi S.B., Park D. W., Jayasuriya S.:** A time-varying sliding surface for fast and robust tracking control of second-order uncertain systems. *Automatica*, Vol. 30, 1994, pp. 899-904.
- [12] **DeCarlo R.S., Žak S., Mathews G.:** Variable structure control of nonlinear multivariable systems: a tutorial. *Proceedings of IEEE*, Vol. 76, 1988, pp. 212-232.
- [13] **Hwang C.L., Lan C.H.:** The position control of electrohydraulic servomechanism via a novel variable structure control. *Mechatronics*, Vol. 4, 1994, pp. 369-389.
- [14] **Nowacka-Leverton A., Bartoszewicz A.:** ITAE optimal variable structure control of second order systems with input signal and velocity constraints. *Kybernetes*, vol. 38, 2009, pp. 1093-1105.
- [15] **Nowacka-Leverton A., Bartoszewicz A.:** ITAE optimal switching plane design for VSC of continuous-time third order systems with state constraints. *European Control Conference*, Budapest, Hungary, 2009, pp. 2116-2121.
- [16] **Nowacka-Leverton A., Bartoszewicz A.:** ITAE optimal transient performance in SMC of third order systems with state and input constraints. *48th IEEE Conference on Decision and Control*, Shanghai, China, 2009, pp. 6720-6725.
- [17] **Nowacka-Leverton A., Michalek M., Pazderski D., Bartoszewicz A.:** Experimental application of time-varying sliding modes for hoisting crane position control with constraints. *49th IEEE Conference on Decision and Control*, Atlanta, USA, 2010, pp. 90-95.
- [18] **Nowacka-Leverton A., Pazderski D., Michalek M., Bartoszewicz A.:** Experimental verification of optimal sliding mode controllers for hoisting cranes. *IFAC World Congress*, Milan, Italy, 2011, pp. 11006-11011.
- [19] **Nowacki Z.:** Sterowanie optymalne układem napędowym, pracującym przy zmiennym obciążeniu i momencie bezwładności. *Podstawy Sterowania*, Vol. 2, 1972, pp. 119-135.
- [20] **Tokat S., Eksin I., Güzelkaya M.:** A new design method for sliding mode controllers using a linear time-varying sliding surface. *IMechE*, Vol. 216, 2002, pp. 455-466.
- [21] **Tokat S., Eksin I., Güzelkaya M., Söylemez M.T.:** Design of a sliding mode controller with a nonlinear time-varying sliding surface. *Transactions of the Institute of Measurement and Control*, Vol. 25, 2003, pp. 145-162.
- [22] **Utkin V.:** Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, Vol. 22, 1977, pp. 212-222.

OPTYMALNE ŚLIZGOWE STEROWANIE ELEKTROHYDRAULICZNYM SERWOMECHANIZMEM POZYCYJNYM

Streszczenie

W artykule przedstawiono ślizgowe sterowanie elektrohydraulicznym serwomechanizmem pozycyjnym, który został zamodelowany jako nieliniowy i niestacjonarny obiekt dynamiczny trzeciego rzędu. Zaproponowane sterowanie wykorzystuje ruchomą płaszczyznę przełączeń o stałym nachyleniu. Płaszczyzna przemieszcza się ruchem jednostajnym (ze stałą prędkością) ze swojego pierwotnego położenia określonego przez wartość uchybu regulacji w chwili rozpoczęcia procesu regulacji do początku układu współrzędnych w przestrzeni stanu. Parametry płaszczyzny zostały wyznaczone tak, aby zapewnić minimalizację całki modułu uchybu regulacji, przy uwzględnieniu ograniczenia przyśpieszenia elementu wykonawczego. Badania symulacyjne wykazały, że zastosowane sterowanie zapewnia pożądaną dokładność oraz niewrażliwość na zewnętrzny moment oporowy od początku trwania procesu regulacji.

Paweł Adamczyk
Politechnika Łódzka, Instytut Automatyki

WOJCIECH BŁASIŃSKI, ZBIGNIEW NOWACKI

Politechnika Łódzka
Instytut Automatyki

UKŁAD HAMOWANIA ELEKTRYCZNEGO DO BADANIA NAPĘDÓW

Recenzent: **dr hab. inż. Andrzej Dębowski, prof. PŁ**

Maszynopis dostarczono: 15.12.2011

Do obciążania badanych napędów zwykle stosuje się generatory prądu stałego. Energia pobierana z badanego napędu jest tracona w dołączonych do generatora rezystorach. Oprócz strat energii wadą takiego układu jest trudność z uzyskaniem odpowiednich momentów obciążenia przy małych prędkościach obrotowych. W pracy przedstawiono zastosowanie zautomatyzowanego układu Leonarda do obciążania badanego napędu. Układ Leonarda pozwala na zwrot pobieranej energii do sieci zasilającej oraz umożliwia dowolne kształtowanie momentu obciążenia, także dla małych prędkości obrotowych. Odpowiednie właściwości układu uzyskuje się przez sterowanie wzbudzenia maszyn prądu stałego.

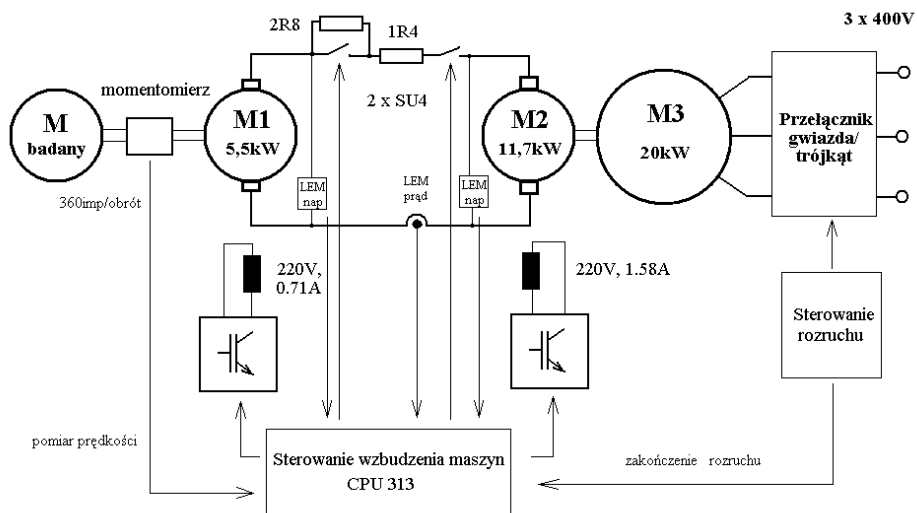
1. WSTĘP

Nowe koncepcje związane ze sterowaniem układów napędowych z silnikami elektrycznymi oprócz badań symulacyjnych, które potwierdzają zasadność ich stosowania wymagają także weryfikacji w układach rzeczywistych. Ważną próbą każdego napędu są próby obciążeniowe, których wyniki ostatecznie potwierdzają właściwości układu napędowego.

W pracy przedstawiono opis maszynowego układu do realizacji prób obciążeniowych badanych napędów. Dzięki odpowiedniemu sterowaniu możliwe jest wykonywanie prób obciążenia w szerokim zakresie stanów pracy badanego napędu z możliwością zwrotu pobieranej energii do sieci zasilającej.

Układ umożliwia na kształtowanie charakterystyk obciążenia w funkcji czasu oraz w funkcji prędkości. Możliwa jest realizacja reakcyjnego lub potencjalnego momentu obciążenia.

2. OPIS UKŁADU OBCIĄŻANIA



Rys. 1. Ogólny schemat blokowy układu obciążania silnika

Ogólny schemat blokowy układu obciążania silnika przedstawiono na rys. 1. Układ składa się z dwóch zestawów maszyn: silnika badanego M i maszyny prądu stałego M1 oraz maszyny prądu stałego M2 i silnika indukcyjnego M3. Zasadniczym celem układu jest możliwość obciążania badanego silnika i odpowiedniego kształtowania charakterystyk obciążenia. W tym stanie pracy maszyna M1 jest prądnicą, maszyna M2 pracuje jako silnik, maszyna M3 pracuje jako generator nadsynchroniczny i przesyła energię do sieci zasilającej. Mostkowe układy sterowania wzbudzenia maszyn prądu stałego umożliwiają pracę czterokwadrantową, co sprawia, że kierunek przesyłania energii może być zmieniany w zależności od potrzeb. Zatem maszyna M1 może pracować jako silnik, maszyna M2 jako prądnica, a maszyna M3 jako silnik np. przy realizacji potencjalnego momentu obciążenia. Różne stany pracy układu są możliwe dzięki czterokwadrantowemu sterowaniu wzbudzenia maszyn prądu stałego. Mostkowe układy zasilania obwodów wzbudzenia maszyn (4 tranzystory IGBT) pracują z modulacją MSI i są sterowane sygnałami napięciowymi z układu sterowania.

Sterowanie wzbudzenia maszyny M1 jest realizowane w układzie otwartym i w trakcie próby zwykle nie jest zmieniane. Wartość prądu wzbudzenia tej maszyny stanowi współczynnik proporcjonalności w zależności przeliczającej prąd tworników maszyn prądu stałego na moment obciążenia.

Sterowanie wzbudzenia maszyny M2 jest zależne od wyboru trybu pracy układu i od stanu łączników w obwodzie tworników maszyny M1 i M2.

Zasadniczą częścią układu sterowania jest sterownik PLC, CPU313. Generuje sygnały do mostkowych układów zasilania obwodów wzbudzenia maszyn oraz umożliwia zamykanie i otwieranie obwodu DC maszyn M1 i M2. Do zamykania i otwierania obwodu tworników maszyn prądu stałego wykorzystano łączniki prądu stałego SU4.

Przełącznik gwiazda/trójkąt wraz z układem sterowania jest w zasadzie niezależną częścią układu. Realizuje sterowaną czasowo procedurę rozruchu maszyny M3 wraz ze wszystkimi funkcjami kontrolnymi. Sygnał zakończenia rozruchu jest przesyłany do układu sterowania wzbudzenia i stanowi warunek konieczny umożliwiający realizację procedury obciążania.

Obwód mocy wyposażony jest w czujniki pomiarowe: napięcie (SEM) tworników maszyn prądu stałego, prądu tworników, prędkości obrotowej badanego napędu i momentu na wale badanego silnika. Sygnał prędkości obrotowej może być wykorzystany do kształtowania charakterystyki obciążenia, sygnał momentu stanowi informację o stanie pracy badanego silnika i nie jest wykorzystywany w układzie sterowania.

3. OPIS UKŁADU STEROWANIA

Sterowanie wzbudzenia maszyny M2 jest zależne od stanu łączników w obwodzie DC. Jeżeli obwód DC jest otwarty, to wzbudzenie maszyny M2 jest ustawiane tak, aby napięcia tworników maszyny M1 i M2 były równe. Sterowanie to jest realizowane w układzie zamkniętym, z regulatorem napięcia. Wartością zadaną jest napięcie twornika maszyny M1, a sygnałem sprzężenia zwrotnego napięcie twornika maszyny M2. Regulator jest proporcjonalny i zapewnia dobre właściwości dynamiczne i niewielki uchyb ustalony. Wybrano regulator proporcjonalny ze względu na to, że nadrzędnym wymaganiem jest stabilność układu zamkniętego w szerokim zakresie zmian punktu pracy układu oraz zapewnienie możliwie krótkich czasów trwania stanów dynamicznych. Zastosowanie regulatora PI powodowałoby dodatkowe opóźnienia w torze przetwarzania uchybu regulacji. Dobór nastaw regulatora PI utrudnia także histereza w obwodzie wzbudzenia maszyn prądu stałego, dlatego wobec braku wymagań dokładności statycznej zdecydowano się na regulator proporcjonalny.

Po zamknięciu obwodu DC oraz zwarcie jednego z rezystorów w obwodzie, sterowanie wzbudzenia maszyny M2 zostaje przełączone na sterowanie wartości prądu w obwodzie. Regulacja napięcia zostaje zastąpiona regulacją prądu w obwodzie tworników maszyn DC. Wartość zadana prądu wynika z realizacji określonej wartości momentu oporowego dla badanego silnika. Algorytm sterowania zależy od wybranego trybu pracy układu. Poniżej przedstawiono możliwe tryby pracy:

- tryb ręczny – ustawianie wzbudzenia maszyny M2 jest realizowane ręcznie, w układzie zamkniętym z regulatorem proporcjonalnym, z zachowaniem kontroli wartości maksymalnej prądu;
- tryb automatyczny ze skokowym momentem obciążenia – moment reakcyjny obciążenia. W tym trybie prąd w obwodzie DC jest utrzymywany na stałej, nastawionej uprzednio wartości, niezależnie od prędkości maszyny M1. Jedynie dla prędkości w zakresie 0-100 obr/min prąd jest liniowo zależny od prędkości, co zapewnia przybliżoną charakterystykę reakcyjną momentu obciążenia. Ten tryb pracy umożliwia badanie rozruchu oraz stanów statycznych ze stałym momentem;
- tryb automatyczny ze skokowym momentem obciążenia – moment potencjalny obciążenia. Ten tryb pracy pozwala na sprawdzanie właściwości badanego napędu, w sytuacji gdy energia jest przekazywana od obciążenia do badanego silnika;
- tryb automatyczny z momentem liniowo zależnym od prędkości. Prąd w obwodzie DC jest proporcjonalny do prędkości obrotowej maszyny M1, z ograniczeniem do wartości maksymalnej prądu. Ten tryb pracy umożliwia badanie napędów z momentem obciążenia liniowo zależnym od prędkości obrotowej;
- tryb automatyczny z momentem liniowo zależnym od czasu.

Modyfikacja programu sterownika pozwala na realizację także innych zależności funkcyjnych momentu obciążenia maszyny badanej.

Sterowanie łącznikami prądu stałego w obwodzie DC jest realizowane ręcznie, przy czym sprawdzane są warunki dla zamknięcia obwodu i dla otwarcia obwodu. Warunkiem zamknięcia obwodu DC jest zmniejszenie różnicy napięć tworników maszyn M1 i M2 poniżej wartości przyjętej za dopuszczalną (przyjęto wartość 15 V), warunkiem otwarcia obwodu jest zmniejszenie prądu DC poniżej wartości dopuszczalnej (przyjęto 10 A). Kontrolą podlega wartość maksymalna prądu DC, po przekroczeniu wartości maksymalnej (przyjęto 80 A) następuje awaryjne otwarcie łączników w obwodzie DC. Wyłączenie awaryjne jest także możliwe ręcznie, wyłącznikiem awaryjnym.

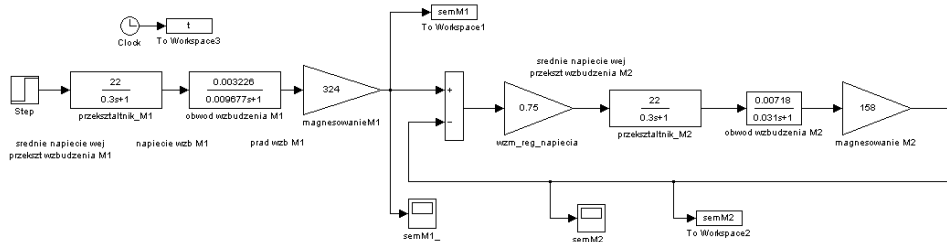
Zastosowanie dwóch łączników w obwodzie tworników ze stopniowym załączaniem i wyłączaniem wraz z odpowiednio połączonymi rezystorami umożliwia ograniczenie skutków przerywania prądu stałego w obwodach o indukcyjnym charakterze. Przez odpowiedni dobór wartości rezystorów zmniejsza się wrażliwość prądu twornika na zmiany warunków wzbudzenia maszyn (zmniejszenie wzmocnienia obiektu dla pętli prądowej).

Ze względu na możliwość pracy z prędkością obrotową przekraczającą wartość znamionową dodatkowo zrealizowane jest interwencyjne ograniczenie wzrostu napięcia twornika maszyny powyżej wartości dopuszczalnej. Regulator interwencyjny jest regulatorem proporcjonalnym, powoduje odwzbudzenie maszyny M1 i działa zawsze, niezależnie od stanu łączników w obwodzie DC maszyn M1 i M2.

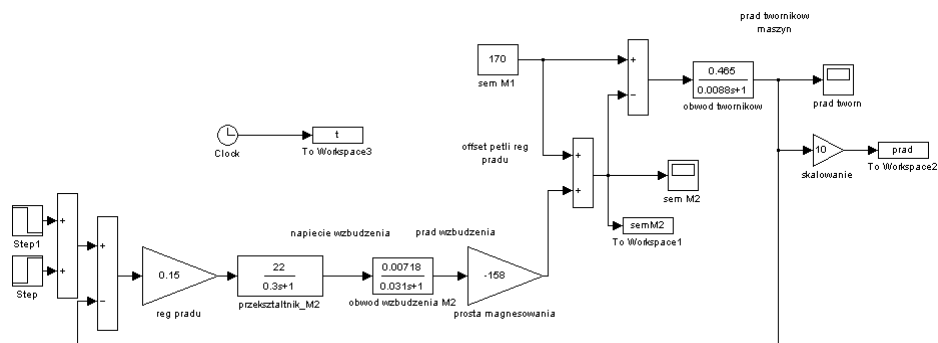
4. BADANIA SYMULACYJNE UKŁADU

Zastosowanie zamkniętych pętli regulacji napięcia i prądu oraz przyjęcie regulacji proporcjonalnej powoduje konieczność ustalenia wzmocnienia regulatorów. Do tego celu zostały zrealizowane modele układu regulacji napięcia i regulacji prądu. Istotne jest przyjęcie uproszczenia nieliniowych układów przekształtników zasilających obwody wzbudzenia. Układy wzbudzenia są sterowane sygnałami wyjściowymi sterownika (a więc sygnałami cyfrowymi) i w układzie modulatora, zewnętrznym w stosunku do sterownika są przekształcone na współczynnik wypełnienia sterujący pracą tranzystorów przekształtników mostkowych. Uwzględniając krótkie czasy przetwarzania sygnałów w sterowniku oraz odpowiednio wysokie częstotliwości pracy przekształtników założono, że mogą one być zastąpione odpowiednimi układami ciągłymi. Zatem na schematach modelu cyfrowy regulator proporcjonalny zastąpiono regulatorem ciągłym, zaś modulator z przekształtnikiem zastąpiono członem inercyjnym I rzędu. Stała czasowa w inercyjnym modelu przekształtnika wynika głównie z funkcji ograniczenia szybkości zmian sygnału wyjściowego modulatora. Taką funkcję realizują scalone modulatory zastosowane w układzie. Powyższe uproszczenia pozwoliły na uzyskanie liniowych modeli układów regulacji napięcia i prądu. Do obliczenia wzmocnienia regulatorów zastosowano kryterium modułowe Kesslera, a następnie sprawdzono wyniki w modelach symulacyjnych. Ostateczną weryfikację przyjętych założeń upraszczających i wyznaczonych nastaw przeprowadzono w układzie rzeczywistym.

Na rys. 2 przedstawiono schemat blokowy układu regulacji napięcia, regulacja napięcia jest włączona, gdy obwody tworników maszyn DC są rozłączone. Na rys. 3 pokazano schemat blokowy układu regulacji prądu, regulacja prądu jest włączana po zamknięciu obwodu tworników maszyn DC.

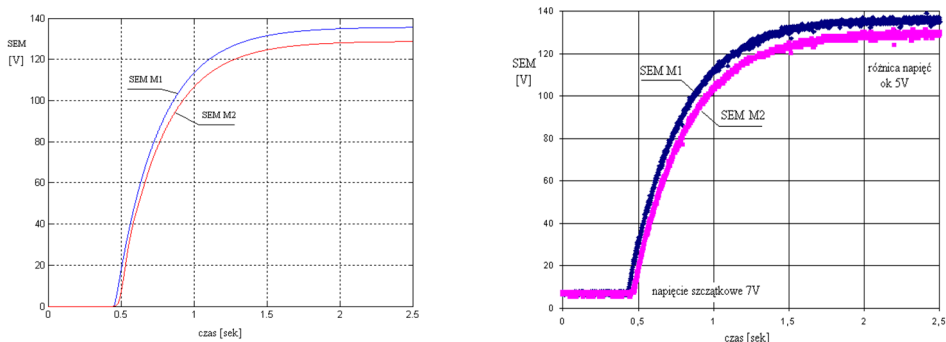


Rys. 2. Schemat modelu symulacyjnego układu regulacji napięcia maszyny M2, wartością zadaną jest napięcie maszyny M1

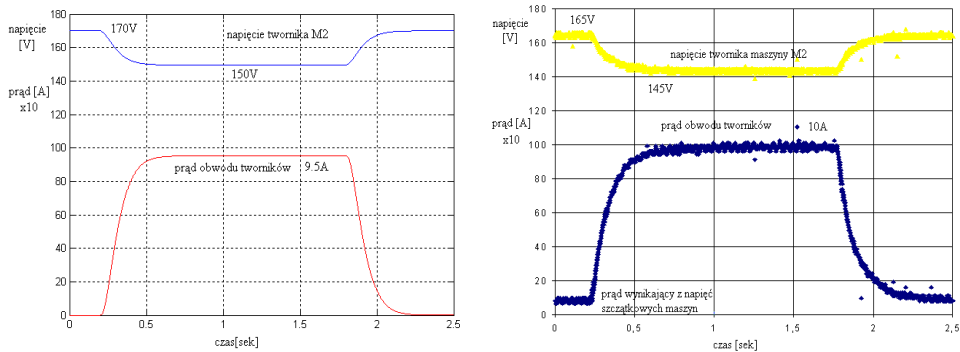


Rys.3. Schemat modelu symulacyjnego układu regulacji prądu w obwodzie tworników maszyn M1 i M2

Na rys. 4 przedstawiono porównanie przebiegów czasowych w modelu symulacyjnym układu regulacji napięcia i w układzie rzeczywistym. Przebiegi zarejestrowano po skokowym załączeniu napięcia wzbudzenia maszyny M1. Regulator napięcia ustawia wzbudzenie maszyny M2, tak aby różnica SEM obu maszyn nie przekraczała wartości przyjętej za dopuszczalną.



Rys. 4. Przebiegi czasowe sił elektromotorycznych maszyn M1 i M2 w modelu symulacyjnym i układzie rzeczywistym dla skokowego załączenia wzbudzenia maszyny M1



Rys. 5. Przebiegi czasowe napięcia twornika maszyny M2 i prądu tworników maszyn w modelu symulacyjnym i układzie rzeczywistym po skokowym włączeniu i wyłączeniu prądu zadanego

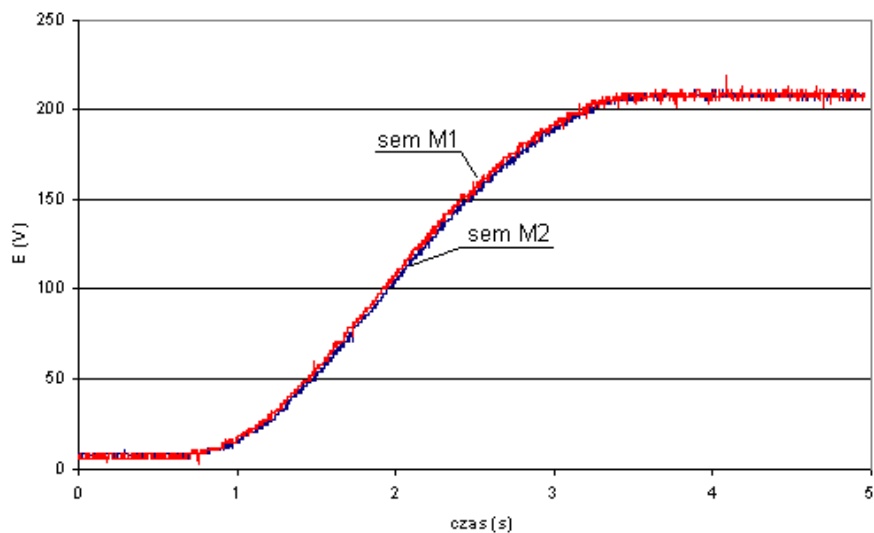
Z kolei na rys. 5 przedstawiono porównanie przebiegów czasowych w modelu symulacyjnym układu regulacji prądu i w układzie rzeczywistym. Przebiegi zarejestrowano po skokowym włączeniu i wyłączeniu zadanego prądu w obwodzie tworników. Wyniki przedstawione na rys. 4 i rys. 5 stanowią potwierdzenie uproszczeń przyjętych przy realizacji modelu układu. Dla nastaw regulatorów przyjętych na podstawie modelu uzyskano zbliżone rezultaty w symulacjach i układzie rzeczywistym.

Kolejne oscylogramy przedstawiają wybrane przebiegi czasowe ilustrujące właściwości układu hamowania dla różnych sytuacji dynamicznych.

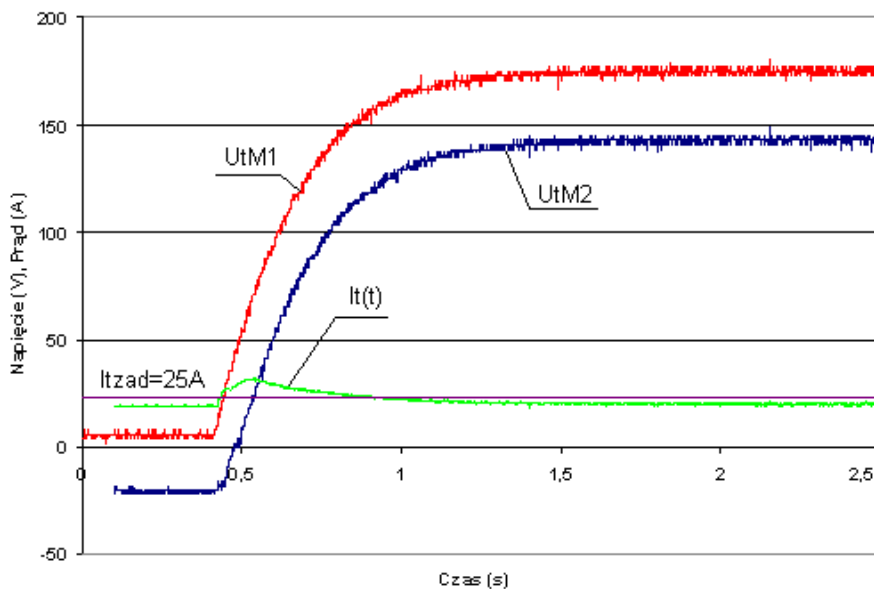
Na rys. 6 przedstawiono działanie układu regulacji napięcia maszyn dla przypadku liniowej zmiany wzbudzenia maszyny M1. SEM maszyny M1 jest wartością zadaną w układzie regulacji SEM maszyny M2 (przypadek skokowej zmiany wzbudzenia przedstawiono na rys. 4).

Na rys. 7 przedstawiono stabilizację prądu w obwodzie tworników maszyn DC. Początek przebiegu to stan, gdy maszyna M1 nie jest wzbudzona, wzbudzenie maszyny M2 zapewnia odpowiednią wartość prądu tworników. Skokowe załączenie wzbudzenia maszyny M1 powoduje pojawienie się stanu dynamicznego, a regulator prądu modyfikuje wzbudzenie maszyny M2, tak aby prąd tworników nie uległ zmianie.

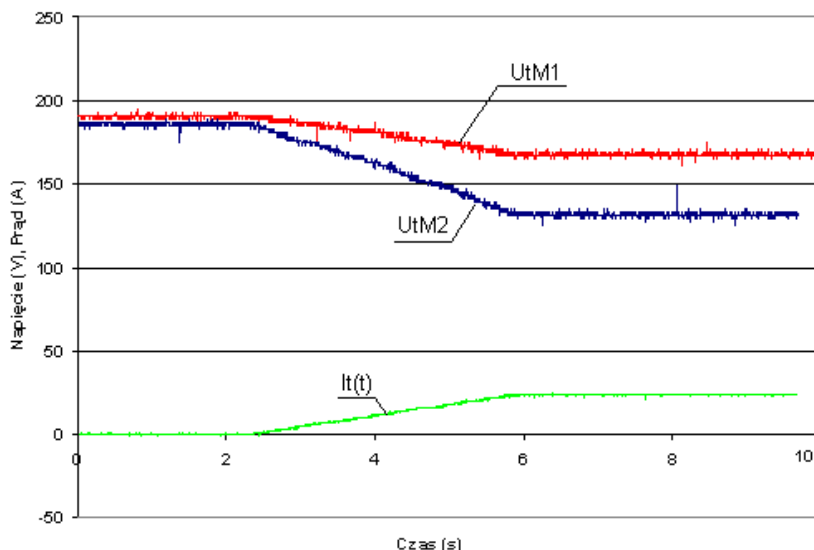
Na rys. 8 przedstawiono przebiegi czasowe dla przypadku liniowej w czasie zmiany prądu zadanego tworników maszyn DC.



Rys. 6. Przebiegi czasowe SEM maszyny M1 i M2 przy liniowej zmianie wzbudzenia maszyny M1



Rys. 7. Przebiegi czasowe napięć tworników maszyn i prądu tworników po skokowym załączeniu wzbudzenia maszyny M1



Rys. 8. Przebiegi czasowe napięć i prądu tworników maszyn DC po włączeniu liniowej w czasie zmiany prądu zadanego tworników

5. WNIOSKI

Przedstawiony układ został zastosowany w badaniach porównawczych napędów z silnikiem indukcyjnym i silnikiem z magnesami trwałymi. Wykorzystano zarówno możliwość stabilizacji momentu obciążenia, jak również możliwość dowolnego kształtowania momentu obciążenia w zależności od prędkości obrotowej wału badanego silnika.

Skuteczny w całym zakresie punktu pracy układ regulacji prądu stwarza możliwość eliminacji regulatora napięcia i pracy układu z połączonymi obwodami tworników maszyn wyłącznie z regulatorem prądu. Wniosek ten potwierdzają symulacje, ale zastosowanie takiego układu sterowania wymaga dodatkowych badań w układzie rzeczywistym. Jak wspomniano, duże znaczenie w regulacji prądu ma dobór wartości rezystora łączącego obwody tworników maszyn DC. Dodatkowa rezystancja w obwodzie tworników maszyn ułatwia uzyskanie stabilnej pracy układu regulacji prądu, ale zwiększa straty w układzie.

LITERATURA

- [1] **Blasiński W.:** Modelling and Simulation of the Electric Brake System Applied to the Test Drive. 8-th International Symposium on System – Modelling – Control, Zakopane, Maj 1995, ss. 114-118.

- [2] **Bąkiewicz K.:** Zastosowanie przekształtnika mostkowego do sterowania wzbudzenia maszyn prądu stałego. Praca dyplomowa. Instytut Automatyki Politechniki Łódzkiej, Łódź, 2011.

ELECTRIC BRAKE SYSTEM FOR DRIVE TESTING

Abstract

The paper presents the Ward – Leonard system used as the electric brake to test drives. The investigations concerning new electrical drives usually require the different experimental tests. The most important are the load tests. System consists of two sets of machines, tested motor coupled to DC generator and the DC machine coupled to induction machine connected to the line. Proposed system enables modelling the load torque as the function of time or the function of angular velocity of motor. The required value of the load torque is set by respective adjust the excitation current of DC machine. The dynamic properties of the system are investigated. The simulation model as well as the experimental system with voltage and current proportional controllers are presented.

Wojciech Błasiński, Zbigniew Nowacki
Technical University of Lodz, Institute of Automatic Control

KRZYSZTOF KUŹMIŃSKI, MACIEJ KRAWIECKI

Politechnika Łódzka

Instytut Automatyki

DYSKUSJA WARUNKÓW PRZESUWALNOŚCI BIEGUNÓW UKŁADÓW LINIOWYCH ZA POMOCĄ STATYCZNYCH SPRĘŻEŃ ZWROTNYCH OD WYJŚCIA

Recenzent: dr hab. Jacek Kabziński, prof. PŁ

Maszynopis dostarczono: 21.03.2011

W artykule przeprowadzono dyskusję typowych warunków przesuwalności biegunów układów liniowych za pomocą statycznych sprzężeń zwrotnych od wyjścia. Jako typowe warunki wystarczające przesuwalności biegunów rozważono warunki Kimury [12] i Wanga [20]. Dla ilustracji problemu rozważono dyskretne zbiory punktów, którym odpowiada określona liczba wejść i wyjść układu. Dla różnych wartości n (4, 6, 7, 8, 9, 10, 12, 16) zaznaczono granice obszarów spełniających warunki Kimury oraz Wanga. Przeanalizowano właściwości poszczególnych obszarów oraz obliczono liczbę Schuberta [3] dla wybranych punktów szczególnych, badając tym samym, czy układy o tych wymiarach mają generycznie właściwość pełnej przesuwalności biegunów.

1. WPROWADZENIE

Istotnymi problemami z zakresu teorii sterowania w dalszym ciągu są zagadnienia związane z przesuwaniem biegunów macierzy transmitancji w określone położenia na płaszczyźnie zmiennej zespolonej [17], [18]. W ten sposób można uzyskać pożądane właściwości dynamiczne układu, które odpowiadają wybranym położeniom biegunów.

W niniejszej pracy skoncentrowano się na zastosowaniu statycznych sprzężeń zwrotnych od wyjścia układu w celu przesunięcia biegunów macierzy transmitancji. W szczególności dotyczy to dyskusji warunków kompletnej przesuwalności, to jest możliwości przemieszczenia wszystkich biegunów układu w dowolnie wybrane

położenie za pomocą sprzężeń, określonych przez rzeczywistą macierz $\mathbf{K}$. W literaturze [17], [18] rozpatrywane są także sprzężenia dynamiczne oraz przesuwalność nie wszystkich biegunów (częściowa). Ze względów ekonomicznych sprzężenia statyczne wydają się lepsze. Natomiast przesuwalność częściowa jest nie do przyjęcia ze względu na to, że bieguny nieprzesuwane podlegają niekontrolowanemu przemieszczeniu, na przykład mogą przesunąć się do prawej półpłaszczyzny zespolonej. Czasem w literaturze są również prowadzone rozważania czysto teoretyczne, dopuszczające zespoloną macierz sprzężeń zwrotnych.

W pracy wzięto pod uwagę wyłącznie metody bezpośrednie, tzn. nieiteracyjne przesuwanie biegunów za pomocą sprzężeń zwrotnych od wyjścia. W ostatnich trzydziestu pięciu latach opracowano wiele takich metod, których pewien przegląd można znaleźć między innymi w pracach [17], [18], [2], [4] i innych pozycjach podanych w bibliografii. Często problem przesuwania biegunów wiąże się bezpośrednio z bardziej ogólnym zagadnieniem projektowania struktury własnej obiektu. Stosowane są przy tym różne podejścia do rozpatrywanego problemu, między innymi podejście parametryczne [15], geometryczne [5], wykorzystujące dwa równania Sylvester'a sprzężone przez warunek ortogonalności [18] i inne. Warunek ortogonalności jest wykorzystywany w większości algorytmów obliczeń macierzy sprzężeń zwrotnych, co prowadzi do dwuetapowej procedury obliczeń.

Autorzy tej publikacji opracowali metodę podaną w [14], która nie wymaga konstrukcji struktury własnej układu i wykorzystania warunku ortogonalności. Prowadzi to do algorytmu jednoetapowego (w dwóch wersjach), w wyniku zastosowania macierzy pseudoodwrotnej Moore'a-Penrose'a i operacji iloczynu Kroneckera.

2. ŚCISŁE SFORMUŁOWANIE PROBLEMU

Przedmiotem rozważań jest liniowy układ stacjonarny $\mathbf{S} = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ opisany przez równania:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (1)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \quad (2)$$

gdzie $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{u} \in \mathbf{R}^m$, $\mathbf{y} \in \mathbf{R}^p$ oraz $\mathbf{A} \in \mathbf{R}^{n \times n}$, $\mathbf{B} \in \mathbf{R}^{n \times m}$, $\mathbf{C} \in \mathbf{R}^{p \times n}$. Zakłada się przy tym, że

$$\text{rank } \mathbf{B} = m \leq n \quad \text{ i } \quad \text{rank } \mathbf{C} = p \leq n, \quad (3)$$

co jest równoznaczne z faktem przekazywania przez sensory i aktuatory wyłącznie znaczących informacji.

Jak wiadomo równaniom (1) i (2) odpowiada macierz transmitancji

$$\mathbf{G}_o(s) = \mathbf{C}(s\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}, \quad (4)$$

której bieguny $\mathcal{A}_o = \{s_1^o, s_2^o, \dots, s_n^o\}$ spełniają równanie charakterystyczne

$$M_o(s) = \det(s\mathbf{I}_n - \mathbf{A}) = 0. \quad (5)$$

W celu narzucenia w rozważanym układzie pożądanego zbioru biegunów macierzy transmitancji $\mathcal{A}_z = \{s_1, s_2, \dots, s_n\}$, wprowadza się statyczne sprzężenie zwrotne od wyjścia określające prawo sterowania

$$\mathbf{u}(t) = \mathbf{K}\mathbf{y}(t), \quad \text{gdzie } \mathbf{K} \in \mathbf{R}^{m \times p}. \quad (6)$$

Zatem układ zamknięty będzie opisany równaniem:

$$\dot{\mathbf{x}}(t) = (\mathbf{A} + \mathbf{B}\mathbf{K}\mathbf{C})\mathbf{x}(t), \quad (7)$$

któremu odpowiada równanie charakterystyczne

$$M(s) = \det(s\mathbf{I}_n - \mathbf{A} - \mathbf{B}\mathbf{K}\mathbf{C}) = 0, \quad (8)$$

spełnione przez elementy zbioru $\mathcal{A}_z$.

Dla określenia niektórych warunków przesuwalności biegunów wygodnie jest wprowadzić pewne pojęcia topologii i geometrii algebraicznej. W związku z powyższym podano dalej ujęcie zaczerpnięte z [7], [8].

Niech $\text{Mat}_R(m \times p)$ będzie zbiorem wszystkich rzeczywistych macierzy o wymiarze $m \times p$, a $\text{Poly}_R(n)$ zbiorem wszystkich monicznych wielomianów stopnia n ze współczynnikami z $\mathbf{R}$, który można utożsamiać z $\mathbf{R}^n$ (użycie współczynników wielomianów – z wyłączeniem wiodącej jedynki – jako współrzędnych ustala bijekcję między nimi). Wtedy dla określonych $\mathcal{S} = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ o elementach rzeczywistych, można zdefiniować odwzorowanie $\chi_{\mathcal{S}}$ jako

$$\chi_{\mathcal{S}} : \text{Mat}_R(m \times p) \rightarrow \text{Poly}_R(n) : \chi_{\mathcal{S}}(\mathbf{K}) = M(s), \quad (9)$$

gdzie $M(s)$ jest określone wzorem (8).

Możliwość narzucenia układowi $\mathcal{S} = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ dowolnego symetrycznego względem osi rzeczywistej zbioru biegunów $\mathcal{A}_z$ przez rzeczywistą macierz $\mathbf{K}$ jest równoważna ze stwierdzeniem, że odwzorowanie $\chi_{\mathcal{S}}$ jest surjekcją. Należy zwrócić uwagę na to, że dla każdych m, n, p istnieją układy $\mathcal{S}$, dla których $\chi_{\mathcal{S}}$ nie jest surjekcją. Przykładem mogą być tutaj układy, dla których $\mathbf{B} = \mathbf{0}$ lub $\mathbf{C} = \mathbf{0}$. Dlatego wprowadza się pojęcie generycznej cechy (właściwości) układów liniowych, którą mają prawie wszystkie systemy w sensie pewnej topologii. Dla danych m, n, p odwzorowanie $\chi_{\mathcal{S}}$ jest generycznie surjektywne, jeżeli istnieje

otwarty, gęsty zbiór V w przestrzeni $\text{Mat}_R(n \times n) \times \text{Mat}_R(n \times m) \times \text{Mat}_R(p \times n)$ taki, że dla $(A, B, C) = S \in V$, odwzorowanie χ_S jest surjekcją. Zbiór otwarty i gęsty rozumieć można w sensie topologii indukowanej metryką euklidesową w przestrzeni $\mathbf{R}^{n^2+mn+mp}$, z którą można utożsamiać przestrzeń systemów liniowych określonego rzędu i wymiarów, ale w literaturze dotyczącej geometrii algebraicznej zwykle używa się dużo słabszej topologii Zariskiego [10].

3. DYSKUSJA CAŁKOWITEJ PRZESUWALNOŚCI BIEGUNÓW

Wspomniana wyżej równoważność całkowitej przesuwalności biegunów i surjektywności odwzorowania χ_S , pozwala stwierdzić, że warunkiem koniecznym całkowitej przesuwalności biegunów za pomocą rzeczywistej macierzy $\mathbf{K}$ jest sterowalność i obserwowalność S . Wynika to z udowodnionego na przykład w [16] warunku koniecznego surjektywności χ_S , który wymaga, aby układ S był sterowalny i obserwowalny. Jednocześnie jest to równoważne ze stwierdzeniem, że stopień MacMillana macierzy transmitancji tego układu jest równy n , to znaczy wymiarowi przestrzeni stanów (a zatem realizacja S musi być minimalna). Sterowalność i obserwowalność są generycznymi właściwościami układów liniowych, więc nie wnosi to nic do kwestii generycznej surjektywności odwzorowania χ_S . Co więcej, sterowalność i obserwowalność nie są wystarczające dla surjektywności χ_S , co pokazuje przykład podany w [13].

Znacznie bardziej skomplikowany jest problem warunków wystarczających całkowitej przesuwalności biegunów, które w różny sposób były formułowane przez wielu autorów.

Najbardziej ogólnymi i najczęściej wykorzystywanymi są:

1. Warunek Kimury [12]

$$m + p > n \quad (10)$$

$$(\text{lub w wersji nieostrej} \quad m + p \geq n) \quad (10')$$

2. Warunek Wanga [20]

$$mp > n \quad (11)$$

Nierówność (10) określa warunek przesuwalności spektrum generycznego zbioru rzeczywistych realizacji S . Warunek ten jest ostrzejszy od (11) dla $n > 5$, ale jednocześnie metody obliczeniowe opracowane przy założeniu jego prawdziwości są łatwiejsze w zastosowaniu praktycznym. Natomiast warunek (11) gwarantuje generyczną przesuwalność spektrum S (biegunów układu) przez macierz rzeczywistą $\mathbf{K}$ [6].

Jednocześnie należy stwierdzić, że jak wykazano w [13], warunek (11) nie jest wystarczający dla osiągnięcia pełnej (nie generycznej) arbitralnej przesuwalności przez statyczne sprzężenie od wyjścia $\mathbf{K}$. Szczególnie skomplikowany jest przypadek krytyczny, gdy

$$mp = n. \quad (12)$$

W pracy [3] uzasadniono, że zagadnienie przesuwalności biegunów z wykorzystaniem macierzy $\mathbf{K}$ jest równoważne z klasycznym problemem Schuberta. Jednocześnie wykazano tam, że jeżeli $mp = n$ oraz liczba Schuberta

$$d(m, p) = \frac{1!2!\dots(p-1)!(mp)!}{m!(m+1)!\dots(m+p-1)!} \quad (13)$$

jest nieparzysta, to odwzorowanie χ_S jest generycznie surjektywne.

Wykazano ponadto [4], że liczba $d(m, p)$ jest nieparzysta wtedy i tylko wtedy, jeżeli spełniony jest jeden z następujących warunków

$$\min\{m, p\} = 1 \quad (14)$$

lub

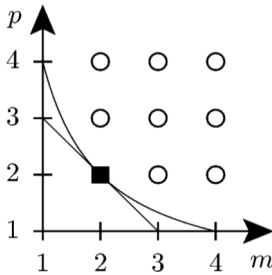
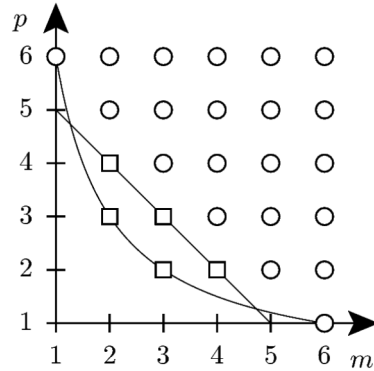
$$\min\{m, p\} = 2 \quad \text{i} \quad \max\{m, p\} = 2^k - 1 \quad \text{dla} \quad k = 2, 3, \dots, \quad (15)$$

Co więcej, udowodniono [7], że jeżeli spełniony jest warunek (12) oraz m i p są liczbami parzystymi, to odwzorowanie χ_S nie jest generycznie surjektywne. Pozwoliło to sformułować wniosek, że układy liniowe $\mathcal{S}$ są generycznie stabilizowalne, jeżeli spełniony jest warunek (11) lub też zachodzi (12) i $m + p$ jest liczbą nieparzystą.

W niniejszej pracy dla ilustracji problemu warunków wystarczających przesuwalności biegunów, zilustrowano dyskretne zbiory punktów, którym odpowiada określona liczba wejść i wyjść układu, dla różnych wartości n z zaznaczeniem granic obszarów spełniających warunki (10') i (12). Pokazano to na rysunkach 1÷8, gdzie dla wybranych punktów krytycznych leżących na hiperboli Wanga (12) obliczono wartości $d(m, p)$. Ponadto na podstawie literatury [2] zaznaczono przez zaciemnienie punkty przetestowane z wynikiem pozytywnym (w sensie znalezienia rzeczywistej macierzy $\mathbf{K}$). Na wymienionych wyżej rysunkach można wyodrębnić trzy obszary, a mianowicie:

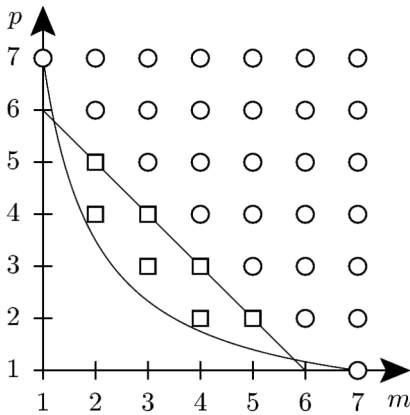
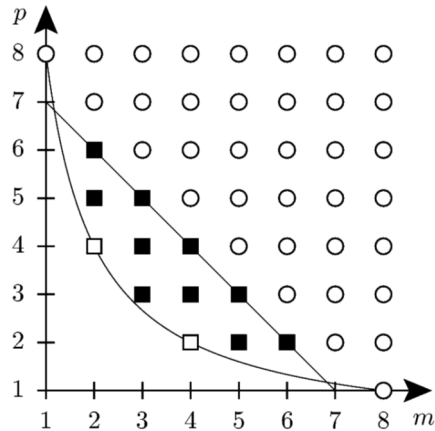
- I. Obszar spełniający warunek Kimury, tj. $m + p > n$. Jest to zbiór punktów, w których dla każdego z nich mamy całkowitą przesuwalność biegunów. W skład tego zbioru wchodzi dwa szczególne punkty, które nie spełniają ostrego warunku Wanga (11). Są to punkty, którym odpowiadają $m = 1 \wedge p = n$ oraz $m = n \wedge p = 1$. Dla tych punktów jednak $d(m, p) = 1$, co zgodnie z wcześniejszymi stwierdzeniami (14) jest wystarczające do tego, by odwzorowanie χ_S było generycznie surjektywne.

- II. Obszar spełniający warunek $m + p \leq n \leq mp$. Jest to obszar, który sprawia największe trudności obliczeniowe i którego punkty nie gwarantują pełnej przesuwalności (w dowolne położenia biegunów). Należy jednak stwierdzić, że w literaturze podano już wiele przykładów odpowiadających poszczególnym punktom tego zbioru, dla których znaleziono pozytywne rozwiązanie, między innymi w [1], [2], [6], [20]. W tej ostatniej pozycji podano to dla $n = 8$, $m = 3$ i $p = 3$ przy $d(3,3) = 42$.
- III. Zbiór spełniający warunek $mp < n$. Jest to obszar dla którego punktów nie można uzyskać całkowitej przesuwalności biegunów.

Rys. 1. Zbiór punktów (m,p) dla $n = 4$ Rys. 2. Zbiór punktów (m,p) dla $n = 6$

Rys. 1 przedstawia zbiór punktów dla $n = 4$. Ten przypadek obejmuje jeden szczególny punkt $(2,2)$, w którym hiperbola Wanga jest styczna do prostej Kimury. Zgodnie z podanym wcześniej stwierdzeniem, w tym punkcie odwzorowanie χ_S nie jest generycznie surjektywne (m i p parzyste), co potwierdza obliczona liczba Schuberta $d(2,2) = 2$ (parzysta). W pracy [21] wykazano, że w tym przypadku problem przesuwania biegunów przez statyczne sprzężenie zwrotne od wyjścia nie jest generycznie rozwiązalny nad ciałem $\mathbf{R}$. Wielu autorów znalazło jednak przykłady, w których wyznaczono macierze $\mathbf{K}$. Oczywiście to nie przeczy wyżej podanym stwierdzeniom. Tym niemniej przez dłuższy czas ten przypadek uważany był za wyjątkowy. Szczególnie ciekawe są tutaj wyniki losowych badań symulacyjnych wykonywanych metodą Monte Carlo, podanych w pracy [6]. W tych badaniach losowo wybierano elementy macierzy definiujących układ $\mathbf{S}$ oraz elementy spektrum $\mathcal{A}_\pm$, przy określonych obszarach zmienności tych elementów. Na podstawie kilkuset tysięcy eksperymentów oszacowano prawdopodobieństwo wystąpienia takich zestawów elementów, które zapewniają pełną przesuwalność biegunów. Uzyskano wynik pozytywny w 85% przypadków.

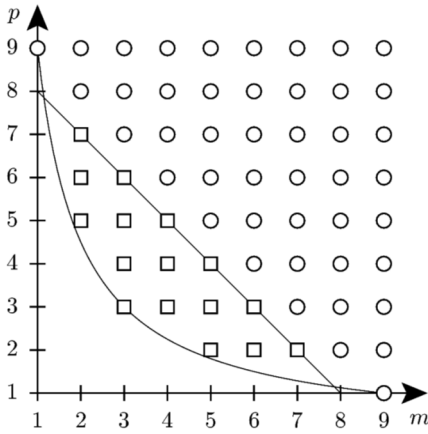
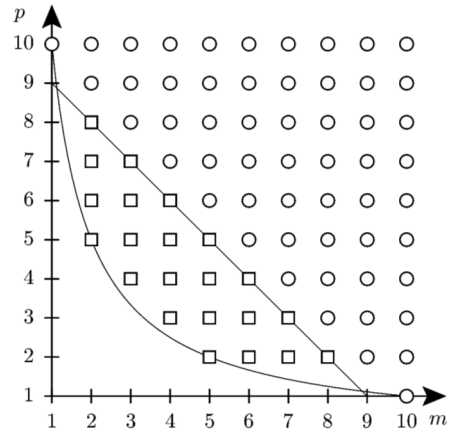
Rys. 2 dotyczy przypadku $n = 6$. W tym przypadku występują dwa punkty szczególne $(2,3)$ i $(3,2)$, które leżą na hiperboli Wanga, przecinającej dwukrotnie prostą Kimury (co jest typowe dla $n \geq 5$). Dla obu punktów szczególnych $d(2,3) = d(3,2) = 5$. Potwierdza to spełnienie warunku (15) dla $k = 2$, co oznacza, że w tych punktach odwzorowanie χ_s jest generycznie surjektywne.

Rys. 3. Zbiór punktów (m,p) dla $n = 7$ Rys. 4. Zbiór punktów (m,p) dla $n = 8$

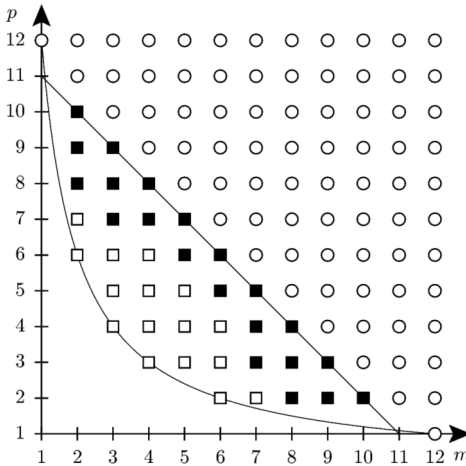
Rys. 3 dotyczy przypadku $n = 7$. W tym przypadku nie występują punkty szczególne. W obszarze II poza punktami na prostej Kimury znajdują się jeszcze 3 punkty, w których odwzorowanie χ_s jest generycznie surjektywne.

Z kolei rys. 4 przedstawia zbiór punktów dla $n = 8$. W tym przypadku ma się do czynienia z dwoma punktami szczególnymi, leżącymi na hiperboli Wanga, tj. $(2,4)$ i $(4,2)$. Dla tych punktów zarówno liczba Schuberta $d(2,4) = d(4,2) = 14$, jak m i p są parzyste, więc odwzorowanie χ_s nie jest dla nich generycznie surjektywne. Wszystkie pozostałe punkty II obszaru zostały przetestowane z wynikiem pozytywnym w literaturze [20], [2].

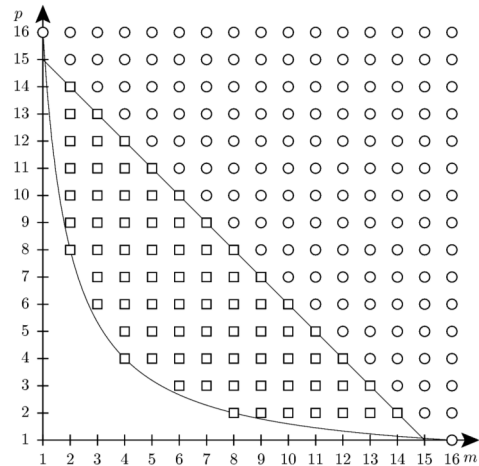
Na rys. 5 i rys. 6, które odpowiadają $n = 9$ i $n = 10$, występują na hiperboli Wanga punkty szczególne odpowiednio jeden punkt $(3,3)$ i dwa punkty $(2,5)$ i $(5,2)$. Wartości liczby Schuberta obliczone dla tych punktów wynoszą $d(3,3) = 42$ oraz $d(2,5) = d(5,2) = 42$. Powoduje to, że nie można na podstawie przytoczonych twierdzeń wywnioskować, czy odwzorowanie χ_s dla tych punktów jest generycznie surjektywne.

Rys. 5. Zbiór punktów (m, p) dla $n = 9$ Rys. 6. Zbiór punktów (m, p) dla $n = 10$

Rys. 7 przedstawia zbiór punktów dla $n = 12$. Na hiperboli Wanga występują w tym przypadku 4 punkty szczególne, dla których obliczono $d(2,6) = d(6,2) = 132$ i $d(3,4) = d(4,3) = 432$. Powoduje to, że dla pierwszej pary punktów odwzorowanie χ_s nie jest generycznie surjektywne, a dla drugiej problem pozostaje otwarty. Dla $n = 12$ zostało przetestowanych w literaturze [2] wiele punktów z rozpatrywanych zbiorów. Wynik testów tych punktów (zaczernionych na rysunku) był pozytywny.



Rys. 7. Zbiór punktów (m, p) dla $n = 12$.
Zaczerniono punkty, dla których w [2]
z powodzeniem wyznaczono nieiteracyjnie
macierz sprzężeń zwrotnych

Rys. 8. Zbiór punktów (m, p) dla $n = 16$

Na ostatnim rys. 8 przedstawiono zbiór punktów dla $n = 16$. W tym przypadku występują trzy punkty szczególne na hiperboli Wanga. Odpowiadają im $d(2,8) = d(8,2) = 1430$ i $d(4,4) = 24024$, i liczby m oraz p są parzyste, co powoduje, że odwzorowanie χ_S w tych przypadkach nie jest generycznie surjektywne.

4. ZAKOŃCZENIE

Na podstawie analizy zbiorów, w których odpowiednie punkty określone są przez liczbę wejść i liczbę wyjść, oraz znanych z literatury twierdzeń zilustrowano pewne aspekty warunków Kimury i Wanga kompletnej przesuwalności biegunów układów liniowych za pomocą statycznych sprzężeń zwrotnych. Wiadomo jest, że w literaturze są podawane różne warunki wystarczające przesuwalności biegunów przy przyjętych założeniach generalnych. Najczęściej jednak dotyczą one przypadków szczególnych i z reguły wynikają z warunku Kimury lub warunku Wanga. Przy wykorzystaniu wymienionych warunków zostało opracowanych wiele algorytmów obliczeń macierzy statycznych sprzężeń zwrotnych $\mathbf{K}$, których twórcami było wielu autorów.

Na marginesie można tutaj wspomnieć o jednoetapowych algorytmach obliczeń opracowanych przez autorów tej pracy [14]. Te algorytmy obejmują cały obszar I zbiorów przedstawionych na rysunkach, to jest punktów spełniających warunek Kimury (z jednym ograniczeniem $A_o \cap A_z = \emptyset$). Umożliwiają one obliczenia zarówno dla biegunów jednokrotnych, jak i dla wielokrotnych oraz rzeczywistych i zespolonych (bez potrzeby prowadzenia obliczeń w dziedzinie liczb zespolonych).

W pewnych sytuacjach można także wykorzystać tzw. lokalną przesuwalność biegunów (local assignability), które to pojęcie wprowadzono w [19]. Polega to na przesunięciu wszystkich biegunów do odpowiednio małego sąsiedztwa każdego z nich w stosunku do położenia pierwotnego. Zazwyczaj sąsiedztwo ograniczone jest okręgiem o środku w położeniu pierwotnym danego bieguna i promieniu, który można obliczyć z określonego warunku tego typu przesuwalności. Zilustrowano to na przykładzie układu z $m = 2$, $p = 2$ i $n = 4$ w pracy [11], gdzie wyprowadzono także warunek przesuwalności i obliczono promień wspomnianego wyżej okręgu ($r_{max} = 1,0013$). Jak stąd wynika obszary, do których można przesunąć bieguny są stosunkowo niewielkie.

Na zakończenie można stwierdzić, że generalnie trudności związane z rozwiązaniem problemu przesuwania biegunów za pomocą statycznych sprzężeń zwrotnych wynikają stąd, że jak wykazał Fu [9], problem należy do klasy „NP-trudnych” (tzn. może nie być rozwiązany w czasie wielomianowym) [18].

LITERATURA

- [1] **Bachelier O., Bosche J., Mechdi D.:** On Pole Placement via Eigenstructure Assignment Approach. *IEEE Trans. On Autom. Contr.*, Vol. 51, No. 9, 2006, pp. 1554-1558.
- [2] **Bachelier O., Mehdi D.:** Non-iterative pole placement technique: A step further. *Journal of the Franklin Institute*, No. 345, 2008, pp. 267-281.
- [3] **Brockett R.W., Byrnes C.J.:** Multivariable Nyquist criteria, root loci and pole placement: a geometric viewpoint. *IEEE Trans. Autom. Contr.* AC-26, 1981, pp. 271-284.
- [4] **Byrnes C.J.:** Pole assignment by output feedback. in *Three Decades of Mathematical System Theory. Lecture Notes in Control and Inform. Sci.* 135, H. Nijmeijer and J.M.Schumacher eds., Springer-Verlag, New York. 1989, pp. 31-78.
- [5] **Champetier C. and Magni J.F.:** On eigenstructure assignm by gain output feedback. *SIAM J. Contr. Optim.* Vol. 29, No. 4, 1991, pp. 848-865.
- [6] **Carotenuto L., Franze G., Muraca P.:** Some results on the genericity of the pole placement problem. *Systems and Control Letters*, Vol. 42, 2001, pp. 291-298.
- [7] **Eremenko A, Gabrielov A.:** Pole placement by static output feedback for generic linear systems. *SIAM J. Contr. Optim.*, Vol. 41, No. 1, 2002, pp. 303-312.
- [8] **Eremenko A, Gabrielov A.:** Counterexamples to pole placement by static output feedback. *Linear Algebra and Its Applications*, 351-352, 2002, pp. 211-218.
- [9] **Fu M.** Pole placement via static output feedback is NP-hard. *IEEE Trans. Autom. Contr.*, Vol. 49, No 5, 2004, pp. 855-857.
- [10] **Herman R., Martin C.F.:** Applications of Algebraic Geometry to Systems Theory – Part I. *IEEE Trans. Aut. Contr.*, Vol. AC-22, No. 1, 1977, pp. 19-25.
- [11] **Jezierski E.:** Local assignability of linear time-invariant systems. *SAMS*, Vol. 16, 1994, pp. 9-15.
- [12] **Kimura H.:** Pole Assignment by Gain Output Feedback. *IEEE Trans. Autom. Contr.* Vol. 20, 1975, pp. 509-516.
- [13] **Kiritsis K.H.:** A necessary condition for pole assignment by constant output feedback. *Systems and Control Letters*. Vol. 45, 2002, pp. 317-320.
- [14] **Kuźmiński K., Krawiecki M.:** Single stage algorithms for pole placement using static output feedback. *Control and Cybernetics*, Vol. 36, No. 4, 2007, pp. 939-965.
- [15] **Roppenecker G., O'Reilly:** Parametric output feedback controller design. *Automatica* 25 (2) 1989, pp. 259-265.
- [16] **Rosenthal J., Sottile F.:** Some remarks on real and complex output feedback. *Syst. Contr. Lett.* 33, 1998, pp. 73-80.
- [17] **Rosenthal J., Willems J.C.:** Open problems in area of pole placement. In V.D. Blondel, E.D. Sontag, M. Vidyasagar, J.C. Willems (eds.), *Open Problems in Mathematical Systems and Control Theory*. Springer Verlag. Chapter 37, 1998, pp. 181-191,
- [18] **Syrmos V.L., Abdallah C.T., Dorato P., Grigoriadis K.:** Static Output Feedback – A Survey. *Automatica* Vol. 33, No. 2, 1997, pp. 125-137.
- [19] **Tarokh M.:** Approach to pole assignment by centralised and decentralised output feedback. *IEE Proc.* 136-D, 1989, pp. 89-97.

- [20] **Wang X.A.:** Grassmanian, Central Projection and Output Feedback Pole Assignment of Linear Systems. IEEE Trans Autom. Contr., Vol. 41, No. 6, 1996, pp. 786-794.
- [21] **Willems J., Hesselink W.:** Generic properties of the pole placement problem. In Proc 7th IFAC World Congress, Helsinki, 1978, pp. 1725-1729.

DISCUSSION OF POLE ASSIGNABILITY CONDITIONS FOR LINEAR SYSTEMS USING STATIC OUTPUT FEEDBACK

Summary

The paper discusses typical pole assignability conditions for linear time invariant systems, using static output feedback. As typical sufficient assignability conditions the theorems of Kimura [12] and Wang [2] are taken under consideration. For illustration of the problem, discrete sets of points with number of inputs and outputs as coordinates are considered. This is shown for various system degree of 4, 6, 7, 8, 9, 10, 12, 16 with marking of domains in which Kimura and Wang conditions are satisfied. Properties of subsequent domains are analyzed and for specific critical cases the Schubert number [3] is calculated for deciding if real pole placement map is generically surjective for those system dimensions.

Krzysztof Kuźmiński, Maciej Krawiecki
Technical University of Łódź, Institute of Automatic Control

ZBIGNIEW ANDRZEJ NOWACKI

**Technical University of Łódź
Computer Engineering Department**

FILE DUPLICATOR FOR THE MICROSOFT WINDOWS OPERATING SYSTEM

Reviewer: **dr hab. inż. Włodzimierz Mosorow, prof. PŁ**

Manuscript received: 10.12.2011

The article presents a tool utility, designed by the author and termed File Duplicator, running under the Microsoft Windows operating system. We have demonstrated that in many cases, especially related to the automatic back-up services, this program can be found useful. One should pay attention to the fact that File Duplicator is an example of memory-resident applications called Windows Services. We have pointed out some differences between such residents and normal programs. A universal and reliable method for creating and installing Windows Services has been specified.

1. INTRODUCTION

The principal motivation for the creation of File Duplicator was the widely understood safety of data being preserved in disk files. In the case of malfunction or malicious attack [1], it is good to have a copy of a lost file. The most popular method of achieving this goal is to archive the disk storage periodically [2]. The disadvantages of this approach are that the process may be time-consuming, and there is a possibility of loss of data that have been updated after the last backup.

File Duplicator removes these drawbacks, for the backup process takes place in a manner transparent to the user, and there is always a practically up-to-date copy of a selected disk file. However, this method can be used solely for the most important files; their maximum number depends on the size of the computer memory.

It is worth emphasizing that, in contrast to traditional archiving systems, File Duplicator does not create backup copies at regular intervals, but it makes them

only if the original file has been changed. This novel approach can help to reduce the consumption of computing resources such as disk storage and CPU time.

Older computer users remember the text editors or word processors [3] that after the update of a document wrote its previous contents to the file with the extension ".BAK". This was convenient, but most of today's programs do not offer this facility. Therefore, it is worth noting that applying File Duplicator you may back up the previous contents of a chosen file as well. This option is of importance also because if, e.g., a virus modifies the file while retaining its name, then the current copy will be corrupted, but the prior copy will be correct.

Using the program is easy. Its executable can be downloaded from the author's web page [4], and run with command-line arguments discussed in Section 2. File Duplicator is able to be started with any method known under the Microsoft Windows operating system [5], but it is best to run it automatically at startup. This can be done by initiating the utility via the *Startup* folder (see Section 3.) or, better, by installing it as a Windows Service [6, 7]. Such a solution is a preferred technique to build the equivalent of a UNIX daemon [8].

Windows Services (cf. Section 4.) work without revealing its presence and usually remain unnoticed by the user. Thus they are able to be treated as a component of the operating system [9]. Their use and implementation is somewhat mysterious, mainly because of the ambiguities and errors occurring in the descriptions that are available on the Internet. That is why Section 5. presents a universal and reliable method of installing Windows Services, and Section 6. contains the full source code of File Duplicator. The reader can easily adapt it to their goals and write their own Windows Service.

2. THE COMMAND-LINE ARGUMENTS OF FILE DUPLICATOR

Savefile.exe (the executable version of File Duplicator) requires at least one command-line argument specifying a disk file *file0*. The program resides in memory [8, 10] and ensures that anytime:

- There is a disk file *file1* containing the exact copy of *file0*.
- There is a disk file *file2* containing the prior copy of *file0*.

Precisely speaking, after starting *Savefile.exe* checks whether *file0* and *file1* exist. If the former exists, but the latter does not, *file0* is copied to *file1*. Next, the program checks every 2000 ms (the number may be changed in the fourth command-line argument) whether *file0* has appeared or it has been modified. If so, then:

- *file2* (if it exists) is deleted,
- *file1* (if it exists) is renamed to *file2*,
- *file0* is copied to *file1*.

The name of *file1* may be specified in the second command-line parameter. Otherwise, it will have the same base name as *file0* with the extension ".\$\$\$". Similarly, *file2* may be specified in the third argument or it equals the base name of *file1* plus the extension ".BAK". The first three parameters should be full paths unless you want to change solely the disk (typing a single letter followed by, maybe, a colon). For instance,

Savefile d:\docs\myfile.doc e:

specifies that *file1* is equal to *e:\docs\myfile.\$\$\$*, and *file2* – to *e:\docs\myfile.bak*, while

Savefile d:\docs\myfile.doc d d 1000

does not alter default names. If you do not need the previous copy of *file0*, set the third parameter to a digit, e.g.,

Savefile d:\docs\myfile.doc d 0

Note that a command-line argument cannot contain spaces. Therefore, if you have a long name [11] with spaces, use either its short counterpart [12] or the following option. If the first character of the first parameter is not a letter, in the first three parameters it is replaced by a space. For example,

Savefile &d:\program&files\my&file.doc d: e:\prior&file.doc

specifies that *file0* is equal to *d:\program files\my file.doc*, and *file2* – to *e:\prior file.doc*.

As follows from the above description, *Savefile.exe* backs up a single disk file. However, it is possible to run a number of its copies with different first parameters. The only limitation is the computer memory capacity (a single copy of the program takes about 1 MB of memory). Note also that *file0* of a copy can be equal to *file2* of another. In this fashion you may obtain a chain of consecutive copies of a very important file.

3. INSTALLING FILE DUPLICATOR IN THE STARTUP FOLDER

As *Savefile.exe* requires command-line arguments, it cannot be directly located in the *Startup* folder. However, you may compile an auxiliary C program [13, 14] containing [15]

```
#include <process.h>
#ifndef P_NOWAIT
#define P_NOWAIT _P_NOWAIT
#endif
void main()
{
    spawnl (P_NOWAIT, "full path\\savefile.exe",
            "full path\\savefile.exe",
            "first parameter for savefile.exe",
            "second parameter",...,NULL);
    spawnl (P_NOWAIT, "full path\\savefile.exe",           // optional
            "full path\\savefile.exe",
            "another parameter",NULL);
}
```

and its executable can be moved to the *Startup* folder.

4. WINDOWS SERVICES

Microsoft Windows Services [6, 7], earlier called NT Services, are memory-resident (i.e., working in the background) programs running in their own window stations that are different from the interactive station of the logged-on user. The reader familiar with Unix-type systems [16] may want to compare them with Unix daemons [8]. In fact, they also set up a mechanism for being called up either periodically or by an application at a later time, and otherwise remain idle in the background until explicitly stopped.

A window station [6] is a secure object containing a local clipboard and a group of local desktop objects. This implies that a Windows Service cannot show any user interface (with the exceptions described in [17]). Precisely speaking, a utility of this type is not able to input or output anything (unless it is located in the disk storage or a method of interprocess communication [18] is applied). Even error messages should not be raised in the user interface, since dialog boxes will not be visible and can cause the program to stop responding.

After installing, Windows Services are able to be automatically started when the system boots (earlier than the user logs into the computer). They run under the system account that has more privileges and permissions than a user

(even administrator) account. At any time services can be stopped and restarted via Windows Task Manager. From the application level this may be done with the aid of, e.g., global semaphores [19-21].

5. INSTALLING FILE DUPLICATOR AS A WINDOWS SERVICE

To install *Savefile.exe* as a Windows Service you will need utilities called *Instsrv.exe* and *Srvany.exe*. They were designed for Windows XP, but under Windows 7 and Windows Vista they work fine as well. The programs are contained in the Windows NT Resource Kit that can be downloaded from [22]. In this way you will obtain the *Rktools.exe* package. After running it, *Instsrv.exe*, *Srvany.exe* and other files will be written to the *c:\Program Files\Windows Resource Kits\Tools* folder, where *c:* might be replaced by another designation of the system disk.

Now perform the following steps:

1. At a command prompt (running as administrator under Windows 7 and Vista), type the following command:

```
<path>\Instsrv.exe SAVEFILE <path>\Srvany.exe
```

where *<path>* is the drive and directory of the utilities (i.e., most frequently the above folder). This creates a service with the name *SAVEFILE* (it may be changed).

2. Run Registry Editor (*Regedt32.exe* of the *System32* folder) and locate the following subkey:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SAVEFILE
```

3. From the *Edit* menu select *New Key* and type:

```
Parameters<Enter>
```

4. From the *Edit* menu choose *New String Value* and type:

```
Application<Enter>
```

5. From the *Edit* menu (or clicking the right mouse button at *Application*) select *Modify...* and type:

```
<path>\savefile.exe command-line parameters <Enter>
```

where *<path>* is a full path to *Savefile.exe*, e.g.,

```
d:\programs\savefile.exe c:\myfile.doc d:\copies\copy1.doc <Enter>
```

6. Close Registry Editor.

Savefile.exe will run automatically with these command-line parameters when the system is restarted. Many such services can be configured if necessary.

They have to have different names, but they might use the same full paths of *Savefile.exe* and *Srvany.exe*.

Using *Instsrv.exe* it is also possible to delete services, e.g.,

Instsrv.exe SAVEFILE remove

Remember that earlier it may be necessary to stop the service with the aid of Task Manager.

When replacing *Savefile.exe* by another application, the foregoing method enables us to run it as a service. However, this makes sense only for some programs that precisely for this reason are able to be called Windows Services. Similar installation instructions can be found in [23], but they contain errors and say nothing about command-line parameters. Their use is vital here because of the absence of the normal user interface.

6. THE FULL SOURCE CODE OF FILE DUPLICATOR

The program has been written in the C programming language [13, 14]. Its full source code is presented below:

```
#include <windows.h>
#include <time.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <io.h>
#include <stdio.h>
#include <fcntl.h>
#include <share.h>
#include <ctype.h>
// CONSTANTS
#define BUFSIZE 50000 // the buffer size
#ifndef MAX_PATH
#define MAX_PATH _MAX_PATH // the maximum path size
#endif
// EXTERNAL VARIABLES
char buf[BUFSIZE];
char file[3][MAX_PATH+4];

#ifdef __BORLANDC__
    struct stati64 state;
#else
    struct _stati64 state;
#endif
time_t prior,current;
__int64 psize,csize;
```

```
int espace;
```

```
// FUNCTION PROTOTYPES
```

```
void createlong(char *file, char *arg);           // create long path
```

```
void createpath(int n, const char *ext, char *arg); // create copy path
```

```
int copyfile(const char *from, const char *to);    // copy files
```

```
time_t modtime(__int64 *size);                    // time of last modification of file[0]
```

```
void CALLBACK ResidentProc(HWND, UINT, UINT, DWORD);
```

```
// MAIN FUNCTION
```

```
int main(int argc, char *argv[])
```

```
{
```

```
if(argc<2) return 1;
```

```
createlong(file[0],argv[1]+(isalpha(espace=*argv[1]) ? espace=0 : 1));
```

```
createpath(1, "$$$", argc>=3 ? argv[2] : NULL); // create path of copy
```

```
createpath(2, "BAK", argc>=4 ? argv[3] : NULL); // path of prior copy
```

```
// TIMER DEFINITION
```

```
SetTimer(NULL,1,argc>=5 ? atoi(argv[4]) : 2000,ResidentProc);
```

```
prior=modtime(&psize);
```

```
// MESSAGE LOOP
```

```
MSG event;
```

```
while(GetMessage(&event, NULL, 0, 0))
```

```
{
```

```
DispatchMessage(&event);
```

```
}
```

```
return 0;
```

```
}
```

```
// CREATE LONG PATH
```

```
void createlong(char *file, char *arg)
```

```
{
```

```
if(espace)
```

```
{
```

```
char *p=arg=strdup(arg);
```

```
for(;;)
```

```
{
```

```
p=strchr(p,espace);
```

```
if(!p) break;
```

```
*p++=' ';
```

```
}
```

```
}
```

```
if(!GetLongPathNameA(arg,file,MAX_PATH)) strcpy(file,arg);
```

```
if(!strchr(file,'.')) strcat(file,".");
```

```
strcat(file," ");
```

```
}
```

```
// CREATE COPY PATH
```

```
void createpath(int n, const char *ext, char *arg)
```

```
{
```

```

memcpy(file[n],file[n-1],MAX_PATH+3);
if(arg)
{
    if(strlen(arg)<=2) file[n][0]=arg[0];           // only disk
    else
    {
        createlong(file[n],arg);                   // path
        return ;
    }
}
strcpy(strrchr(file[n],'.')+1,ext);
}
// COPYING FILES
int copyfile(const char *from, const char *to)
{
    int i;
    int infile=sopen(from, O_RDONLY | O_BINARY, SH_DENYNO);
    if (infile<0) return 0;
    int outfile=sopen(to, O_CREAT | O_RDWR | O_BINARY | O_TRUNC,
        SH_DENYWR, S_IRREAD | S_IWRITE);
    if (outfile<0) return 0;
    for(;;)
    {
        i=read(infile,buf,BUFSIZE);
        if(!i) break;
        write(outfile,buf,i);
    }
    close(outfile);
    close(infile);
    return 1;
}
// TIME OF LAST MODIFICATION OF FILE[0]
time_t modtime(__int64 *size)
{
    {
        if(!_stati64(file[0], &state ))
        {
            if(size) *size=state.st_size;
            return state.st_mtime;
        }
    }
    if(size) *size=-1;
    return 0;
}
// TIMER PROCEDURE
void CALLBACK ResidentProc(HWND, UINT, UINT, DWORD)
{

```

```

static int work;
if(work) return ;
work=1;
if(access(file[1],0)<0) copyfile(file[0],file[1]);
else
{
    current=modtime(&csize);
    if(current && (current != prior || csize != psize))
    {
        if(isalpha(*file[2]))
        {
            unlink(file[2]);
            rename(file[1],file[2]);
        }
        copyfile(file[0],file[1]);
        prior=current;
        psize=csize;
    }
}
work=0;
}

```

Let us discuss some aspects of this code. as windows services are designed to work without a graphical user interface, they can be console applications [24]. therefore, although file duplicator includes the *windows.h* file, it contains the *main* function instead of *winmain* [25, 26].

Some tasks of services require a periodic testing of certain conditions and undertaking an action only when they are satisfied. For instance, File Duplicator has to examine if the file specified in the first parameter has appeared or has been modified. To this end, the *SetTimer* function [27] in *main* defines a so-called timer. As a result, the *ResidentProc* function will be invoked every 2000 ms (by default). Note that the procedure of copying may be long-term, so the use of the *work* variable in the timer procedure is essential.

At the end of the main function one can find the loop testing events (manifested in coming messages) [28, 29]. In contrast to a normal loop, this sequence of instructions does not overburden the system even if it is being performed for a long time and by many programs simultaneously. Note that the message loop differs slightly from the analogous one in Windows applications [30-32]. For in the latter [26, 33] the call of *DispatchMessage* [34] is preceded by that of *TranslateMessage* [35] responsible for providing ASCII codes of characters entered from the keyboard. But services do not allocate the entry of this device, so *TranslateMessage* can be omitted here.

7. CONCLUSIONS

In the paper we present an easy to use and helpful tool utility, called File Duplicator, for creating up-to-date copies of important disk files. Its purpose, command-line arguments and installation methods have been discussed in detail. The full source code of File Duplicator has been also included.

It should be emphasized that File Duplicator is an example of applications called Windows Services. Hence the instructions incorporated in the work may be applied to run any program as a Windows Service. And the whole article can be regarded as a guide to writing such memory-resident utilities by those who know the C/C++ language [13, 14, 36] even if they have no experience in building so-called Windows applications [30-32].

REFERENCES

- [1] **Zimoch M.:** Privacy and Security on the Internet. Master's thesis supervised by Z. A. Nowacki. Technical University of Łódź, 2009.
- [2] Back up – Guide to backing up the important files on your hard drive. <http://www.helpwithpcs.com/maintenance/back-up-backing-up-guide.htm>.
- [3] **Ignaczak M.:** The Comparative Analysis of Computer Text Processing Systems. Master's thesis supervised by Z.A. Nowacki. Technical University of Łódź, 2008.
- [4] Modern TSR programs by Z.A. Nowacki. <http://www.nova.pc.pl/software.htm>.
- [5] **Kominiak L.:** The Genesis, Evolution and Structure of Microsoft Windows Operating Systems. Master's thesis supervised by Z.A. Nowacki. Technical University of Łódź, 2007.
- [6] Introduction to Windows Service Applications. [http://msdn.microsoft.com/en-us/library/d56de412\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/d56de412(VS.80).aspx).
- [7] Services. <http://msdn.microsoft.com/en-us/library/ms685141.aspx>.
- [8] Writing resident programs under Linux. http://rudy.mif.pg.gda.pl/~bogdro/linux/tsr_tut_linux_en.html.
- [9] **Stallings W.:** Operating Systems: Internals and Design Principles. Prentice Hall, New Jersey, 2009.
- [10] **Tomczyk M.:** The Methods of Creating Resident Programs in the DOS and Windows Systems. Master's thesis supervised by Z.A. Nowacki. Technical University of Łódź, 2001.
- [11] Naming Files, Paths, and Namespaces. [http://msdn.microsoft.com/en-us/library/aa365247\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa365247(v=vs.85).aspx).
- [12] GetShortPathName Function. [http://msdn.microsoft.com/en-us/library/aa364989\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa364989(v=vs.85).aspx).
- [13] **Kernighan B. W., Ritchie D. M.:** The C Programming Language. Second Edition. Prentice Hall, New Jersey, 1988.
- [14] C Language Tutorial. http://einstein.drexel.edu/courses/Comp_Phys/General/C_basics/.

- [15] spawn functions. <http://www.users.pjwstk.edu.pl/~jms/qnx/help/watcom/clibref/src/spawn.html>.
- [16] **Bukowiecki M.:** The Genesis, Development and Practical Applications of the Linux System. Master's thesis supervised by Z.A. Nowacki. Technical University of Łódź, 2007.
- [17] Interactive Services. [http://msdn.microsoft.com/en-us/library/ms683502\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms683502(v=VS.85).aspx).
- [18] Interprocess Communications. [http://msdn.microsoft.com/en-us/library/windows/desktop/aa365574\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa365574(v=vs.85).aspx).
- [19] **Szurgot R.:** Parallel Programming in Windows. Master's thesis supervised by Z.A. Nowacki. Technical University of Łódź, 2009.
- [20] Multiple Threads. [http://msdn.microsoft.com/en-us/library/ms684254\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms684254(v=vs.85).aspx).
- [21] Concurrent Processes: Basic Issues. <http://cnx.org/content/m12312/latest/>.
- [22] Windows Server 2003 Resource Kit Tools. <http://www.microsoft.com/download/en/confirmation.aspx?id=17657>
- [23] How To Create a User-Defined Service. <http://support.microsoft.com/default.aspx?scid=kb;en-us;137890>.
- [24] Running Console Applications from Windows. <http://www.hermetic.ch/rundos.htm>.
- [25] WinMain Entry Point. [http://msdn.microsoft.com/en-us/library/ms633559\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms633559(v=vs.85).aspx).
- [26] WinMain – application entry point. <http://www.toymaker.info/Games/html/winmain.html>.
- [27] SetTimer Function. [http://msdn.microsoft.com/en-us/library/ms644906\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms644906(v=vs.85).aspx).
- [28] MSG Structure. [http://msdn.microsoft.com/en-us/library/ms644958\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms644958(v=vs.85).aspx).
- [29] GetMessage Function. [http://msdn.microsoft.com/en-us/library/ms644936\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms644936(v=vs.85).aspx).
- [30] **Petzold C.:** Programming Windows. Microsoft Press, 1998.
- [31] **Richter J.:** Programming Applications for Microsoft Window. Microsoft Press, 1999.
- [32] **Prosiś J.:** Programming Windows with MFC. Microsoft Press, 1999.
- [33] Creating a Message Loop. [http://msdn.microsoft.com/en-us/library/ms644928\(v=vs.85\).aspx#creating_loop](http://msdn.microsoft.com/en-us/library/ms644928(v=vs.85).aspx#creating_loop).
- [34] DispatchMessage Function. [http://msdn.microsoft.com/en-us/library/ms644934\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms644934(v=vs.85).aspx).
- [35] TranslateMessage Function. [http://msdn.microsoft.com/en-us/library/ms644955\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms644955(v=vs.85).aspx).
- [36] **Stroustrup B.:** The C++ Programming Language. Third Edition. Addison Wesley, New Jersey, 1997.

POWIELACZ PLIKÓW: PRZYKŁAD USŁUG SYSTEMU WINDOWS

Streszczenie

Artykuł prezentuje program narzędziowy, zaprojektowany przez autora i nazwany Powielaczem Plików, pracujący pod systemem operacyjnym Microsoft Windows. Zostało zademonstrowane, że w wielu przypadkach, zwłaszcza związanych z automatyczną archiwizacją plików, program może być użyteczny. Zwrócono uwagę na fakt, że Powielacz Plików jest przykładem rezydujących w pamięci aplikacji zwanych Usługami Windows. Wskazano na pewne różnice między takimi rezydentami a zwykłymi programami. Została podana uniwersalna i niezawodna metoda projektowania i instalowania Usług Windows.

Zbigniew Andrzej Nowacki
Politechnika Łódzka, Katedra Informatyki Stosowanej

GRZEGORZ WASIAK

**Politechnika Łódzka
Instytut Automatyki**

ALGORYTM PRZEKSZTAŁCANIA SCHEMATÓW BLOKOWYCH W PROGRAMIE AUTOCAD

Recenzent: **dr hab. inż. Jacek Kabziński, prof. PŁ**

Maszynopis dostarczono: 15.12.2011

Jako przykład komputerowego wspomaganie projektowania w zakresie szeroko rozumianej automatyki przedstawiono algorytm aplikacji wspomagającej przekształcanie schematów blokowych w środowisku programu AutoCAD. W omawianym algorytmie proponuje się uzupełnienie graficznego odwzorowania poszczególnych elementów używanych w schematach blokowych o atrybuty związane z poszczególnymi wejściami i wyjściami rozpatrywanego elementu, które zawierają nazwy elementów dołączonych do tych wejść i wyjść. Daje to możliwość stworzenia biblioteki elementów wykorzystywanych do rysowania schematów blokowych oraz zbudowania swoistej tablicy połączeń poszczególnych elementów w schemacie.

1. WSTĘP

Komputerowe wspomaganie projektowania, CAD (ang. *Computer Aided Design*) – jest pojęciem określającym zastosowanie sprzętu i oprogramowania komputerowego w projektowaniu technicznym. Popularnym przykładem programu typu CAD jest AutoCAD – produkt firmy Autodesk. Znajduje on zastosowanie między innymi w inżynierii mechanicznej, elektrycznej, budowlanej. We wszystkich tych dziedzinach AutoCAD traktowany jest jako program graficzny uzupełniany co najwyżej bibliotekami elementów ułatwiającymi sam proces rysowania. Podobne zastosowania znaleźć można w zakresie szeroko rozumianej automatyki. Brak jest natomiast aplikacji wykorzystujących połączenie możliwości programu AutoCAD w jego warstwie graficznej z możliwościami obliczeniowymi

współpracującego z nim języka AutoLISP. Niniejszy artykuł pokazuje przykład takich możliwości. Jest nim algorytm aplikacji wspomagającej rysowanie schematu blokowego, a następnie przekształcanie go w kolejnych krokach (zgodnie z algebrą schematów blokowych). Każde z przekształceń zostaje odwzorowane graficznie, a w bazie rysunkowej zapisywane są dodatkowo kolejne wypadkowe transmitancje tych przekształceń. Według autora takie rozwiązanie ma nowatorski charakter w zakresie inżynierskiej praktyki projektowania układów automatyki. Zakłada się u czytelnika podstawową znajomość programu AutoCAD, języka AutoLISP oraz zasad ich współdziałania.

2. NARZĘDZIA OMAWIANEGO ALGORYTMU

AutoCAD jest programem, którego elementy rysunkowe typu blok mogą zawierać dodatkowe informacje, wykorzystywane również poza samym programem. Ogólnie blok (wyróżniany unikalną nazwą i traktowany przez polecenia edycyjne jako jeden obiekt) jest złożony z powiązanych ze sobą elementów składowych, przy czym oprócz typowych elementów rysunkowych (również tekstów) w jego skład mogą wchodzić tzw. atrybuty – specyficzne „pojemniki” przechowujące zapisane w nich informacje (tekstowe lub numeryczne). Raz zdefiniowany blok może być wstawiany wielokrotnie w różnych miejscach rysunku, m.in. przy różnych kątach obrotu oraz wartościach związanych z nim atrybutów. Wartości atrybutów dla każdego wystąpienia bloku w rysunku można m.in. eksportować poza program i wykorzystywać w plikach tekstowych czy bazach danych, a ich widoczność na ekranie może być sterowana przez użytkownika.

W omawianym algorytmie proponuje się m.in. uzupełnienie graficznego odwzorowania poszczególnych elementów używanych w schematach blokowych (węzeł rozgałęźny i sumacyjny, blok transmitacyjny oraz gałąź przepływu sygnału) o atrybuty związane z poszczególnymi wejściami i wyjściami rozpatrywanego elementu, które zawierają nazwy elementów dołączonych do tych wejść i wyjść. Daje to możliwość stworzenia biblioteki elementów wykorzystywanych do rysowania schematów blokowych oraz zbudowania swoistej tablicy połączeń poszczególnych elementów w schemacie, pozwalając na ciągłą kontrolę i zapis wykonywanych przekształceń.

Kolejną własnością programu AutoCAD, która zostanie wykorzystana w omawianym algorytmie jest możliwość zapisu rysunku w formacie DXF (ang. *Drawing eXchange Format*). Jest to format pozwalający na wymianę rysunków pomiędzy różnymi programami graficznymi w postaci pliku tekstowego. Plik DXF z formalnego punktu widzenia jest pełną listą opisu elementów bazy rysunkowej programu AutoCAD. Elementami listy DXF są kolejne podlisty zawierające wielkości charakteryzujące poszczególne obiekty rysunkowe, przy czym dla każdego obiektu pierwszą podlistą jest zawsze tzw. para kropkowa – lista złożona z dwóch elementów rozdzielonych kropką – np. (-1 . <Entity name:

600000cc>). Pierwszy element tej pary kropkowej jest zawsze liczbą całkowitą określającą rodzaj danej zgodnie z kodem DXF. Natomiast drugim elementem tej pary kropkowej jest wielkość charakteryzująca dany obiekt rysunkowy zgodnie z podanym kodem. W przedstawionym przykładzie **-1** oznacza kod wywołujący nazwę identyfikacyjną obiektu, a **<Entity name: 600000cc>** jest tą właśnie nazwą. Oprócz par kropkowych na liście DXF opisu obiektu występują także listy zwykle zawierające dwa lub więcej elementów, przy czym pierwszy z nich zawsze oznacza kod DXF – np. **(10 10.0 150.0 0.0)**. W tym przykładzie **10** jest kodem tzw. punktu głównego obiektu (jest nim punkt początkowy dla linii, punkt wstawienia dla bloku itp), a **10.0 150.0 0.0** to wartości współrzędnych prostokątnych tego punktu. Dla złożonych obiektów rysunkowych (np. blok z atrybutami czy polilinia z kolejnymi wierzchołkami) lista DXF elementu głównego (np. bloku) zawiera w sobie m.in. podlisty określające własności elementów podrzędnych (składowych) elementu głównego (np. wartości atrybutów).

Ogólnie więc pliki DXF określające aktualny stan rysunku pozwalają na modyfikacje wybranych własności na listach związanych z poszczególnymi obiektami rysunkowymi, np. zmianę współrzędnych punktu wstawienia bloku lub łańcucha tekstowego atrybutu określającego nazwę elementu dołączonego do węzła sumacyjnego. Możliwe jest także wymazanie z pliku niepotrzebnego już obiektu. Tak zmieniony plik DXF może zostać uaktualniony w programie, uwidaczniając dokonane modyfikacje.

Najistotniejszym jednak elementem pozwalającym zbudować i zastosować przedstawiany algorytm jest język programowania AutoLISP, w którego interpreter jest wyposażona każda kolejna wersja programu AutoCAD. AutoLISP jest językiem wysokiego poziomu, dobrze nadającym się do zastosowań graficznych i pozwalającym użytkownikowi na tworzenie własnych funkcji i programów. Przykładem może być funkcja wstawiająca do rysunku blok o określonej postaci zależnej od wskazywanych punktów orientujących kierunek przepływu sygnału. Istotną cechą języka AutoLISP jest m.in. operowanie nad danymi typu **nazwa obiektu rysunkowego** (typ **ENAME**) oraz typu **lista** (typ **LIST**).

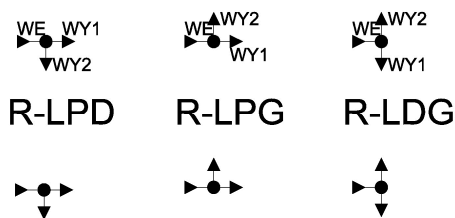
Do obsługi obiektów wykorzystywane mogą być m.in. funkcje: **(entsel)** – pozwala wybrać obiekt rysunkowy, **(entnext)** – odnajduje kolejny (główny lub podrzędny) obiekt w rysunku, **(entget)** – odczytuje listę DXF wybranego obiektu, **(entmod)** – modyfikuje listę DXF wybranego obiektu, **(entdel)** – usuwa obiekt z rysunku, **(entupd)** – aktualizuje wygląd obiektu na rysunku. Natomiast do bezpośrednich działań na listach DXF można zastosować m.in. funkcje: **(car)** – zwraca pierwszy element listy, **(cdr)** – zwraca listę bez pierwszego elementu, **(cons)** – zwraca listę z nowym elementem dodanym na początku listy, **(assoc)** – zwraca element skojarzony z podanym kodem, **(subst)** – zwraca listę po zamianie wybranego elementu. Podane powyżej funkcje występując w programie zapisanym w języku AutoLISP, muszą oczywiście być użyte wraz z niezbędnymi dla nich argumentami, zgodnie z odpowiednią konwencją języka.

3. BUDOWA BIBLIOTEKI ELEMENTÓW

Tworzona biblioteka musi być dostępna dla użytkownika poprzez fragment menu programu AutoCAD (np. menu rozwijane, piktogramowe, paskowe lub paletowe), które musi składać się z 4 podstawowych pozycji pozwalających wstawić do rysunku obiekty wykorzystywane przy budowie schematu blokowego. Są nimi w kolejności węzeł rozgałęźny, węzeł sumacyjny, blok transmitancyjny oraz gałąź przepływu sygnału. Każdy z tych obiektów musi zawierać – oprócz elementów rysunkowych składających się na jego graficzne odwzorowanie w programie AutoCAD – także niewidoczne atrybuty pozwalające zapisywać informacje związane m.in. z połączeniami danego obiektu z innymi obiektami w schemacie. Jednocześnie ze względu na możliwą różną orientację danego obiektu na rysunku, należy zaprogramować w języku AutoLISP procedurę wyboru konkretnej postaci obiektu w zależności od wzajemnego położenia punktów, którymi użytkownik odwzorowuje kierunki sygnałów wejściowych i wyjściowych dopływających i wypływających ze wstawianego aktualnie obiektu. Poniżej przedstawione zostaną propozycje budowy bloków dla poszczególnych obiektów i pokazane propozycje procedury wyboru ich określonej postaci.

3.1. Węzeł rozgałęźny

Graficzne możliwości odwzorowania węzła rozgałęźnego zaproponowano na rys. 1, na którym przedstawiono w trzech wariantach definicję bloku, jego nazwę i wygląd po wstawieniu do rysunku. Pokazane warianty (razem z możliwością obrotu wstawianego bloku o kąty $\pm 90^\circ$ oraz 180°) wyczerpują możliwości rozgałęziania sygnałów dla przyjętego przepływu tylko w kierunku poziomym oraz pionowym. W każdym z bloków jego punkt wstawienia został umieszczony na początku strzałki oznaczonej atrybutem WE. Atrybut ten (podobnie jak pozostałe atrybuty WY1 i WY2) jest atrybutem niewidocznym, a jego domyślna wartość powinna być ustawiona na 0 – co oznacza brak elementów przyłączonych do danego wejścia/wyjścia. Wartość ta będzie ulegała zmianie po uzupełnieniu rysunku o gałąź przepływu sygnałów.

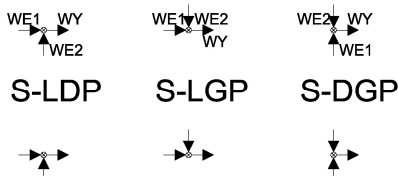


Rys. 1. Bloki odwzorowujące węzeł rozgałęźny

Stworzona w języku AutoLISP procedura wyboru jednego z przedstawionych bloków (wraz z odpowiednim kątem obrotu) powinna być oparta na wskazaniu przez użytkownika – przy włączonym trybie ORTO, co wymusi wskazywanie tylko kierunku poziomego lub pionowego – trzech punktów określających kolejno punkt wstawienia bloku (pokrywający się z kierunkiem sygnału wejściowego WE) oraz kierunki sygnałów WY1 oraz WY2. Dla każdego z sygnałów wyjściowych procedura, opierając się na wartościach współrzędnych zapamiętanych punktów, powinna określić ich położenie względem punktu WE. Przykładowo dla WY1 leżącego poniżej oraz dla WY2 leżącego na lewo od punktu WE, procedura powinna wybrać blok R-LPD z kątem obrotu -90° .

3.2. Węzeł sumacyjny

Graficzne możliwości odwzorowania węzła sumacyjnego zaproponowano na rys. 2, na którym przedstawiono w trzech wariantach definicję bloku, jego nazwę i wygląd po wstawieniu do rysunku. Podobnie jak dla węzła rozgałęźnego pokazane warianty (razem z możliwością obrotu wstawianego bloku o kąty $\pm 90^\circ$ oraz 180°) wyczerpują możliwości sumowania sygnałów dla przyjętego przepływu tylko w kierunku poziomym oraz pionowym. W każdym z bloków jego punkt wstawienia został umieszczony na początku strzałki oznaczonej atrybutem WE1. Atrybut ten (podobnie jak pozostałe atrybuty WE2 i WY) jest atrybutem niewidocznym, a jego domyślna wartość powinna być ustawiona na 0 – co oznacza brak elementów przyłączonych do danego wejścia/wyjścia. Wartość ta będzie ulegała zmianie po uzupełnieniu rysunku o gałęzie przepływu sygnałów.

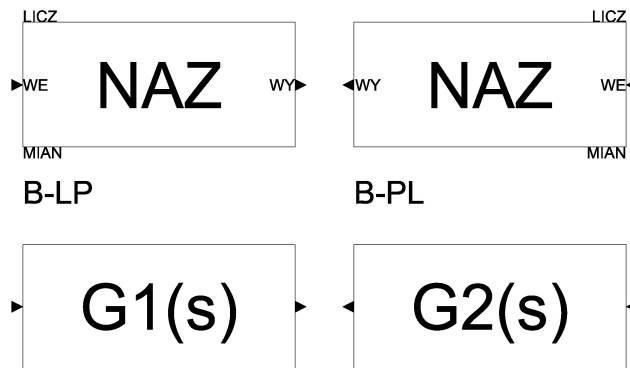


Rys. 2. Bloki odwzorowujące węzeł sumacyjny

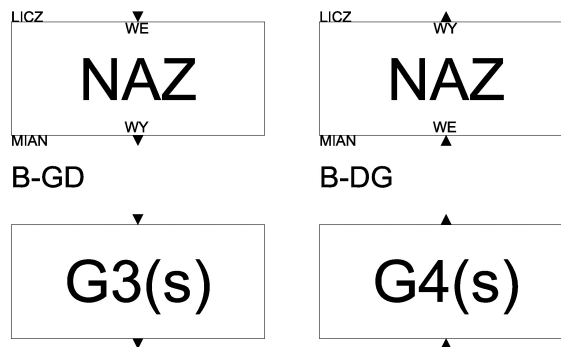
Stworzona w języku AutoLISP procedura wyboru jednego z przedstawionych bloków (wraz z odpowiednim kątem obrotu) powinna być oparta na wskazaniu przez użytkownika – przy włączonym trybie ORTO, co wymusi wskazywanie tylko kierunku poziomego lub pionowego – trzech punktów określających kolejno punkt wstawienia bloku (pokrywający się z kierunkiem sygnału wejściowego WE1) oraz kierunki sygnałów WE2 oraz WY. Dla każdego z sygnałów procedura, opierając się na wartościach współrzędnych zapamiętanych punktów, powinna określić ich położenie względem punktu WE1. Przykładowo dla WE2 leżącego na lewo oraz dla WY leżącego powyżej punktu WE1, procedura powinna wybrać blok S-LGD z kątem obrotu $+90^\circ$.

3.3. Blok transmitancyjny

Graficzne możliwości odwzorowania bloku transmitancyjnego zaproponowano na rys. 3 oraz rys. 4, na których przedstawiono w czterech wariantach definicję bloku, jego nazwę i wygląd po wstawieniu do rysunku. W odróżnieniu od wcześniej opisanych bloków odwzorowujących węzły schematu blokowego, w definicji bloku transmitancyjnego występuje atrybut NAZ określający nazwę bloku, przy czym (jak widać na rys. 3 i rys. 4) jest on oczywiście atrybutem widocznym po wstawieniu do rysunku. Uniemożliwia to stosowanie do wstawianego bloku różnego od 0 kąta obrotu i wymusza zdefiniowanie dwóch par bloków, których postacie zależą od kierunku sygnałów wejściowego i wyjściowego. Upraszcza to procedurę wyboru bloku, gdyż w tym przypadku należy tylko zbadać wzajemne położenie punktu określającego kierunek sygnału WY względem punktu wstawienia bloku (pokrywającego się z kierunkiem sygnału WE).



Rys. 3. Bloki odwzorowujące bloki transmitancyjne dla poziomego przepływu sygnału



Rys. 4. Bloki odwzorowujące bloki transmitancyjne dla pionowego przepływu sygnału

Dodatkowo definicja każdego bloku zawiera dwa niewidoczne atrybuty LICZ oraz MIAN. Przy wstawianiu bloku zapisuje się w nich współczynniki wielomianów licznika i mianownika opisywanej transmitancji operatorowej. Proponuje się zapis tych współczynników w postaci listy języka AutoLISP.

3.4. Gałąź przepływu sygnału

Gałąź przepływu sygnału powinna być wstawiana przy pomocy elementu rysunkowego typu polilinia, charakteryzującego się m.in. możliwością łączenia w jednym obiekcie kilku segmentów liniowych odwzorowujących połączenia pomiędzy wcześniej wstawionymi węzłami i blokami transmitancyjnymi. Procedura rysowania takiej gałęzi powinna umożliwiać (poza standardowym wskazywaniem kolejnych wierzchołków polilinii) zapamiętanie początkowego i końcowego punktu polilinii, przy czym punkt początkowy musi pokrywać się z punktem wyjściowym, a punkt końcowy – z punktem wejściowym odpowiedniego węzła lub bloku transmitancyjnego. Każdy z tych punktów jako argument występującego w języku AutoLISP polecenia (**entsel**) pozwala określić występującą w bazie rysunkowej nazwę obiektu, z którym gałąź jest połączona. Pozwala to zmienić standardowe wartości ustawione na 0 dla odpowiednich atrybutów (wyjściowego dla obiektu przyłączonego do początku i wejściowego dla obiektu przyłączonego do końca gałęzi) na nazwy obiektów przyłączonych do drugiego z krańców gałęzi.

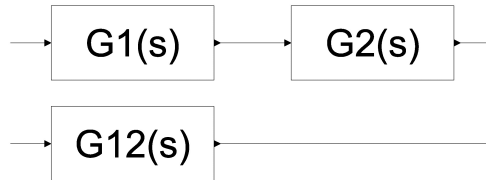
Należy także pamiętać, że wymazywanie z rysunku gałęzi przepływu sygnału musi działać wraz z procedurą odczytującą dla danej gałęzi (poprzez jej punkty krańcowe) nazwy przyłączonych obiektów i zmieniającą wartości ich odpowiednich atrybutów na 0. Takie postępowanie zapewni użytkownikowi, że spośród wszystkich atrybutów typu WE i WY w całej bazie rysunkowej tylko te, które opisują sygnały wejściowe i wyjściowe całego układu będą miały wartości równe 0.

4. PRZEKSZTAŁCENIA SCHEMATÓW BLOKOWYCH

Przygotowany w sposób podany powyżej schemat blokowy może zostać poddany przekształceniom wynikającym z algebry schematów. Tworzone procedury, podobnie jak biblioteka elementów, muszą być dostępne dla użytkownika w określonym typie menu. Poniżej zostaną zaproponowane algorytmy procedur wykonujących poszczególne przekształcenia. Należy jednak pamiętać o konieczności modyfikacji każdej z tych procedur w zależności od tego, do jakiej konfiguracji gałęzi przepływu sygnałów jest ona stosowana. Opisane procedury dotyczą konfiguracji przedstawionej na odpowiadających im rysunkach.

4.1. Połączenie szeregowe

Procedura przekształcająca połączenie szeregowe powinna rozpocząć się od wskazania przez użytkownika dwóch bloków transmitancyjnych podlegających przekształceniu w kolejności zgodnej z kierunkiem przepływu sygnału (na rys. 5 oznacza to kolejność $G1(s)$ – $G2(s)$).

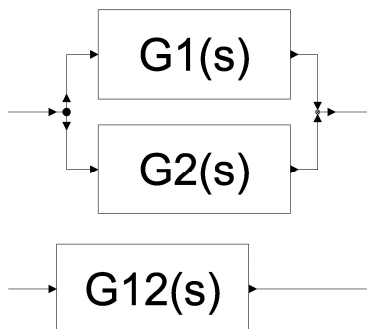


Rys. 5. Połączenie szeregowe – procedura zwijania schematu

Kolejny krok to zmiana nazwy pierwszego z bloków na nazwę podawaną przez użytkownika, wraz z uwzględnieniem tej zmiany w odpowiednich atrybutach obiektów znajdujących się na przeciwległych krańcach dwóch gałęzi – doprowadzającej sygnał do bloku $G1(s)$ oraz wyprowadzającej sygnał z bloku $G2(s)$. Jednocześnie przesuwa się punkt początkowy gałęzi wychodzącej z $G2(s)$ do wyjścia bloku $G12(s)$. Oblicza się także nowe wartości współczynników licznika i mianownika transmitancji wypadkowej, modyfikując w ten sposób wartości atrybutów LICZ oraz MIAN bloku $G12(s)$. Na zakończenie usuwane są z rysunku blok $G2(s)$ oraz gałąź łącząca przekształcane bloki.

4.2. Połączenie równoległe

Procedura przekształcająca połączenie równoległe powinna rozpocząć się od wskazania przez użytkownika dwóch bloków transmitancyjnych podlegających przekształceniu w dowolnej kolejności (na rys. 6 zaproponowano kolejność $G1(s)$ – $G2(s)$).

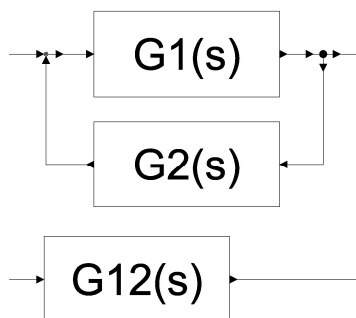


Rys. 6. Połączenie równoległe – procedura zwijania schematu

Kolejny krok to zmiana nazwy pierwszego z bloków na nazwę podawaną przez użytkownika, wraz z uwzględnieniem tej zmiany w odpowiednich atrybutach obiektów znajdujących się na przeciwległych krańcach dwóch gałęzi – doprowadzającej sygnał do węzła rozgałęźnego oraz wyprowadzającej sygnał z węzła sumacyjnego. Jednocześnie przesuwa się punkt wstawienia bloku $G_{12}(s)$ do punktu wstawienia węzła rozgałęźnego oraz punkt początkowy gałęzi wychodzącej z węzła sumacyjnego do wyjścia bloku $G_{12}(s)$. Oblicza się także nowe wartości współczynników licznika i mianownika transmitancji wypadkowej, modyfikując w ten sposób wartości atrybutów LICZ oraz MIAN bloku $G_{12}(s)$. Na zakończenie usuwane są z rysunku blok $G_2(s)$, węzły rozgałęźny i sumacyjny oraz gałęzie łączące te węzły z przekształcanymi blokami.

4.3. Połączenie ze sprzężeniem zwrotnym

Procedura przekształcająca połączenie ze sprzężeniem zwrotnym powinna rozpocząć się od wskazania przez użytkownika dwóch bloków transmitancyjnych podlegających przekształceniu w kolejności blok toru głównego oraz blok sprzężenia zwrotnego (na rys. 7 oznacza to kolejność $G_1(s)$ – $G_2(s)$).



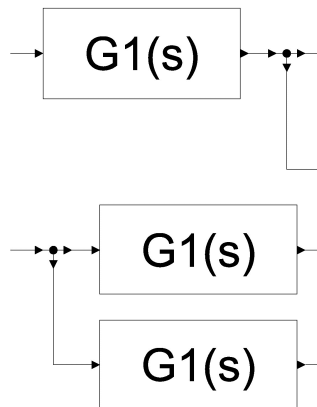
Rys. 7. Połączenie ze sprzężeniem zwrotnym – procedura zwijania schematu

Kolejny krok to zmiana nazwy pierwszego z bloków na nazwę podawaną przez użytkownika, wraz z uwzględnieniem tej zmiany w odpowiednich atrybutach obiektów znajdujących się na przeciwległych krańcach dwóch gałęzi – doprowadzającej z zewnątrz sygnał do węzła sumacyjnego oraz wyprowadzającej na zewnątrz sygnał z węzła rozgałęźnego. Jednocześnie przesuwa się punkt wstawienia bloku $G_{12}(s)$ do punktu wstawienia węzła sumacyjnego oraz punkt początkowy gałęzi wychodzącej na zewnątrz z węzła rozgałęźnego do wyjścia bloku $G_{12}(s)$. Oblicza się także nowe wartości współczynników licznika i mianownika transmitancji wypadkowej, modyfikując w ten sposób wartości atrybutów LICZ oraz MIAN bloku $G_{12}(s)$. Na zakończenie usuwane są z rysunku blok $G_2(s)$, węzły rozgałęźny i sumacyjny oraz gałęzie łączące te węzły z przekształcanymi blokami.

4.4. Przesunięcie węzła rozgałęźnego z wyjścia bloku na jego wejście

Procedura przesuwająca węzeł rozgałęźny z wyjścia bloku na jego wejście powinna rozpocząć się od wskazania przez użytkownika obiektów biorących udział w przekształceniu w kolejności blok – węzeł (jak na rys. 8).

Kolejny krok to usunięcie gałęzi łączącej te obiekty, zamiana punktów wstawienia węzła i bloku oraz połączenie ich nową gałęzią (wraz z ewentualnym zastosowaniem w rysunku polecenia ROZCIĄGNIJ w celu zmiany odległości pomiędzy węzłem i blokiem). Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z węzła (na rys. 8 jest to wyjście poziome) do wyjścia bloku, a końcowy punkt gałęzi doprowadzonej do wejścia bloku – do wejścia węzła. Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

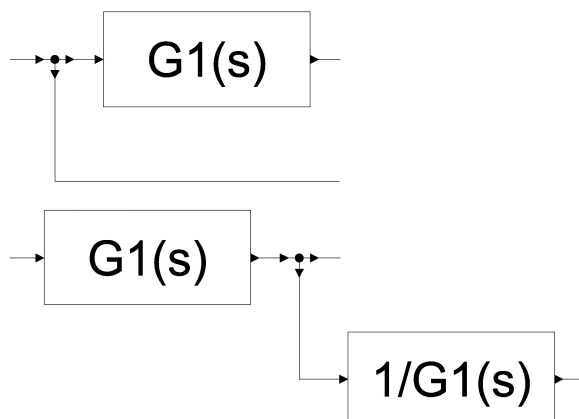


Rys. 8. Przesunięcie węzła rozgałęźnego z wyjścia bloku na jego wejście

W dalszej części procedura umieszcza w rysunku kopię transmitancyjnego bloku $G1(s)$, łącząc jego punkt wstawienia nową gałęzią z drugim wyjściem węzła rozgałęźnego oraz przesuując do wyjścia tego bloku punkt początkowy gałęzi wychodzącej przed przekształceniem z drugiego wyjścia węzła (na rys. 8 jest to wyjście pionowe). Również te modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

4.5. Przesunięcie węzła rozgałęźnego z wejścia bloku na jego wyjście

Procedura przesuwająca węzeł rozgałęźny z wejścia bloku na jego wyjście powinna rozpocząć się od wskazania przez użytkownika obiektów biorących udział w przekształceniu w kolejności węzeł – blok (jak na rys. 9).



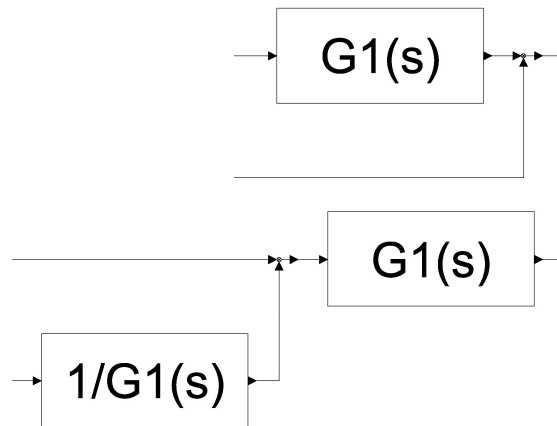
Rys. 9. Przesunięcie węzła rozgałęźnego z wejścia bloku na jego wyjście

Kolejny krok to usunięcie gałęzi łączącej te obiekty, zamiana punktów wstawienia węzła i bloku oraz połączenie ich nową gałęzią (wraz z ewentualnym zastosowaniem w rysunku polecenia ROZCIĄGNIJ w celu zmiany odległości pomiędzy węzłem i blokiem). Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z bloku do wyjścia węzła (na rys. 9 jest to wyjście poziome), a końcowy punkt gałęzi doprowadzonej do wejścia bloku – do wejścia węzła. Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

W dalszej części procedura umieszcza w rysunku kopię transmitancyjnego bloku (o zmienionej nazwie $1/G1(s)$ i zamienionych wartościach atrybutów LICZ i MIAN), łącząc jego punkt wstawienia nową gałęzią z drugim wyjściem węzła rozgałęźnego oraz przesuując do wyjścia tego bloku punkt początkowy gałęzi wychodzącej przed przekształceniem z drugiego wyjścia węzła (na rys. 9 jest to wyjście pionowe). Również te modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

4.6. Przesunięcie węzła sumacyjnego z wyjścia bloku na jego wejście

Procedura przesuująca węzeł sumacyjny z wyjścia bloku na jego wejście powinna rozpocząć się od wskazania przez użytkownika obiektów biorących udział w przekształceniu w kolejności blok – węzeł (jak na rys. 10).



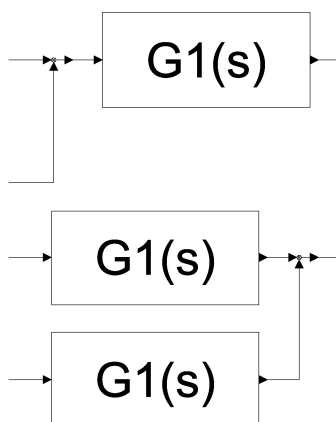
Rys. 10. Przesunięcie węzła sumacyjnego z wyjścia bloku na jego wejście

Kolejny krok to usunięcie gałęzi łączącej te obiekty, zamiana punktów wstawienia węzła i bloku oraz połączenie ich nową gałęzią (wraz z ewentualnym zastosowaniem w rysunku polecenia ROZCIĄGNIJ w celu zmiany odległości pomiędzy węzłem i blokiem). Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z węzła do wyjścia bloku, a końcowy punkt gałęzi doprowadzonej do wejścia bloku – do wejścia węzła (na rys. 10 jest to wejście poziome). Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

W dalszej części procedura umieszcza w rysunku kopię transmitancyjnego bloku (o zmienionej nazwie $1/G1(s)$ i zamienionych wartościach atrybutów LICZ i MIAN), łącząc jego punkt wyjściowy nową gałęzią z drugim wejściem węzła sumacyjnego oraz przesuując do wejścia tego bloku punkt końcowy gałęzi dochodzącej przed przekształceniem do drugiego wejścia węzła (na rys. 10 jest to wejście pionowe). Również te modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

4.7. Przesunięcie węzła sumacyjnego z wejścia bloku na jego wyjście

Procedura przesuująca węzeł sumacyjny z wyjścia bloku na jego wejście powinna rozpocząć się od wskazania przez użytkownika obiektów biorących udział w przekształceniu w kolejności węzeł – blok (jak na rys. 11).



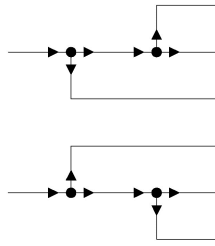
Rys. 11. Przesunięcie węzła sumacyjnego z wejścia bloku na jego wyjście

Kolejny krok to usunięcie gałęzi łączącej te obiekty, zamiana punktów wstawienia węzła i bloku oraz połączenie ich nową gałęzią (wraz z ewentualnym zastosowaniem w rysunku polecenia ROZCIĄGNIJ w celu zmiany odległości pomiędzy węzłem i blokiem). Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z bloku do wyjścia węzła, a końcowy punkt gałęzi doprowadzonej do wejścia węzła (na rys. 10 jest to wejście poziome) – do wejścia bloku. Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

W dalszej części procedura umieszcza w rysunku kopię transmitancyjnego bloku $G1(s)$, łącząc jego punkt wyjściowy nową gałęzią z drugim wejściem węzła sumacyjnego oraz przesuując do wejścia tego bloku punkt końcowy gałęzi dochodzącej przed przekształceniem do drugiego wejścia węzła (na rys. 10 jest to wejście pionowe). Również te modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwległych ich krańcach.

4.8. Zamiana węzłów rozgałęźnych

Procedura zamieniająca położenie węzłów rozgałęźnych powinna rozpocząć się od wskazania przez użytkownika dwóch węzłów w kolejności zgodnej z kierunkiem przepływu sygnału.

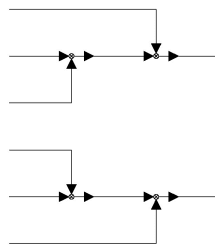


Rys. 12. Zamiana węzłów rozgałęźnych

Kolejny krok to usunięcie gałęzi łączącej te węzły, zamiana punktów wstawienia oraz połączenie ich nową gałęzią. Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z węzła drugiego do wyjścia węzła pierwszego (na rys. 12 jest to wyjście poziome), a końcowy punkt gałęzi doprowadzonej do wejścia węzła pierwszego – do wejścia węzła drugiego. Dodatkowo (dla obu węzłów) uwzględnia się przeniesienie punktu początkowego gałęzi przyłączonej do drugiego wyjścia do nowego położenia. Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwnych ich krańcach.

4.9. Zamiana węzłów sumacyjnych

Procedura zamieniająca położenie węzłów rozgałęźnych powinna rozpocząć się od wskazania przez użytkownika dwóch węzłów w kolejności zgodnej z kierunkiem przepływu sygnału.



Rys. 13. Zamiana węzłów sumacyjnych

Kolejny krok to usunięcie gałęzi łączącej te węzły, zamiana punktów wstawienia oraz połączenie ich nową gałęzią. Jednocześnie przesuwa się początkowy punkt gałęzi wychodzącej z węzła drugiego do wyjścia węzła pierwszego, a końcowy punkt gałęzi doprowadzonej do wejścia węzła pierwszego – do

wejścia węzła drugiego (na rys. 13 jest to wejście poziome). Dodatkowo (dla obu węzłów) uwzględnia się przeniesienie punktu końcowego gałęzi przyłączonej do drugiego wejścia do nowego położenia. Wszystkie powyższe modyfikacje gałęzi wymagają zmian w odpowiednich atrybutach obiektów znajdujących się na przeciwnych ich krańcach.

5. PODSUMOWANIE

Przedstawiona koncepcja algorytmu jest ogólnym przykładem ilustrującym możliwości wykorzystania programu AutoCAD (rozszerzonego przez aplikację zapisane w języku AutoLISP) w zakresie szeroko rozumianej automatyki. Zbudowanie na tej podstawie aplikacji, która wspomaga przekształcanie schematów blokowych niezależnie od kierunków przepływu sygnałów, wymaga oczywiście dokładnego rozważenia i uwzględnienia wszystkich możliwości konfiguracyjnych schematu. Tym niemniej zaproponowane podejście może być bazą do zaprojektowania całościowego programu.

LITERATURA

- [1] **Dudek M.:** AutoLISP. Praktyczny kurs. Gliwice. Helion 2003.
- [2] **Pikoń A.:** AutoCAD 2007 PL. Gliwice. Helion 2007.
- [3] Pomoc programu AutoCAD 2009 PL.
- [4] www.autodesk.pl

ALGORITHM FOR THE TRANSFORM OF BLOCK DIAGRAMS IN AutoCAD

Summary

In the paper, an algorithm is presented for supporting the transformation of block diagrams in the AutoCAD environment, as an example of computer-aided design in the wide range of automatics. The algorithm proposes an extension of a graphical representation of components used in the block diagrams with attributes associated with various inputs and outputs of these components. The attributes contain the names of elements connected to these inputs and outputs. This gives the possibility of creating a library of elements used for drawing block diagrams, and building the specific array of elements' connections in the diagram.

Grzegorz Wasiak
Instytut Automatyki Politechnika Łódzka

KAMIL GRABOWSKI

Technical University of Łódź

Department of Microelectronics and Computer Science

PERSON IDENTIFICATION BASED ON IRIS IMAGE ANALYSIS WITH SPECIAL EMPHASIS ON HARDWARE IMPLEMENTATION – STATISTICAL ANALYSIS

Reviewer: **prof. dr hab. inż. Andrzej Napieralski**

Manuscript received: 5.07.2010

The paper presents main results of PhD dissertation concerning authentication systems based on the analysis of iris pattern. Two main threads of the work are presented: iris image segmentation and its influence on the feature extraction algorithm and methods of analysis of biometric system efficiency.

1. INTRODUCTION

Biometrics is a field of study including person identification based on their physiological or behavioral features. Among different biometric features suitable for person identification one of the most promising is a human iris due to its relatively low error rates. However, current state of the art in the iris recognition field indicates that existing methods have limited reliability when large populations are considered.

To identify a person based on his/her iris pattern, first a photo of the eye must be taken. Next, the photo is processed. In all iris recognition systems this processing consists of three major stages. Firstly, the iris region is segmented, secondly, its features are extracted. Finally the features are compared with the signatures stored previously in the database of registered users. On the basis of comparison result, the system recognizes the person standing in front of the camera.

2. IRIS SEGMENTATION

In the analysis process of biometric sample B, quality and quantity of information contained in its digital representation play a crucial role. In case of the iris biometric these factors vary along the lighting conditions and affect the whole authentication process. The absorption spectrum of melanin – the substance whose concentration makes the iris tissue colored, shows interesting characteristics for various wavelengths of light [1]. Although in current iris recognition systems near infrared light is used for illumination (due to overall ease of implementation), the meaningful increase in information content, measured as image entropy, occurs for iris images taken under visible light – figure 1. However, in this type of images correct iris segmentation is much more difficult, as it is shown in [2].

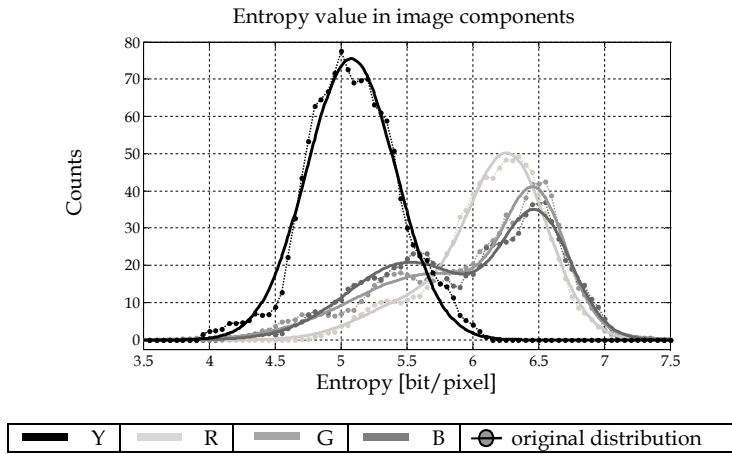


Fig. 1. Entropy distribution inside iris area obtained for two databases: ISEP (visible light in RGB components) and BATH (near infrared light in Y component)

In the iris segmentation algorithm optimized for images taken under near infrared light, a clear-cut iris-pupil boundary is used for the algorithm accuracy improvement. Similarly, the author suggested using the red (R) component of RGB iris image for segmentation algorithm optimization in images taken under visible light. Figure 2 shows the histogram of gradients recorded during the test on an iris-pupil border in each image in a group of 921 pictures taken in the visible light, contained in the ISEP database¹.

¹ Institut Supérieur d'Electronique de Paris iris database.

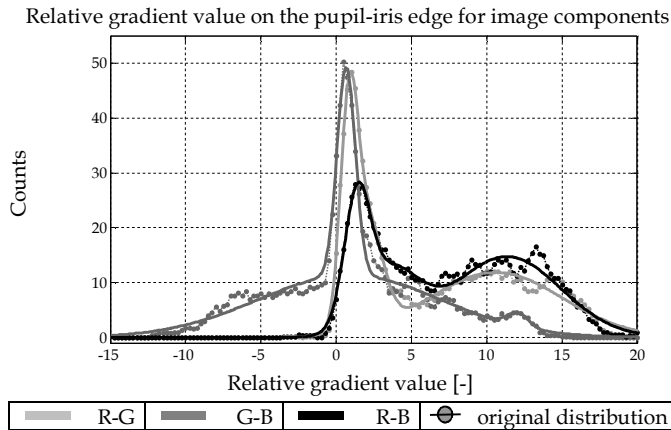


Fig. 2. Distribution of relative gradient of luminance on iris-pupil border

It may be noted that the greatest average gradient of the relative clarity of the iris-pupil boundary is observed for the red band of visible light spectrum. A small standard deviation of the distribution suggests that this feature is relatively stable for a large group of photos from the test set – table 1.

Table 1. Distribution mean values and confidence bounds comparison

	Gradient differences in image components	
	Sclera-Iris	Iris-Pupil
R-G component difference	$-0.46^{+2.32}_{-2.01}$	$6.32^{+10.07}_{-6.04}$
G-B component difference	$-0.46^{+1.67}_{-1.20}$	$1.05^{+10.87}_{-9.47}$
R-B component difference	$-1.35^{+3.76}_{-2.80}$	$7.37^{+8.76}_{-6.96}$

Therefore, a selective use of the red component on the pupil segmentation stage should result in an increased efficiency and reliability of the whole iris segmentation process for the images taken under visible light. The proposed approach was used in the software for segmentation of iris images taken in visible light, developed by W. Sankowski. The solution increased the number of correct segmentations among UBIRIS Database iris images [3] – table 2.

Table 2. Segmentation efficiency comparison [3]

Database	Segmentation efficiency	
	Grayscale	Red component
DMCS	100.00 %	100.00 %
UBIRIS (session I)	92.20 %	99.67 %

The presented results show that the selective use of the red component in the process of iris image segmentation increases the efficiency of this phase.

3. BIOMETRICAL SYSTEM EFFICIENCY

The performance of a positive biometric system is determined by error rates of false acceptance – FAR and false rejection – FRR. In general their analysis allows to compare different biometric systems and to choose the most optimal solutions depending on the assumed security strategy.

In the biometric system, the estimation of the mentioned error rates is based on the results obtained in the authentication protocol after similarity score calculation is done by the biometric comparator (1). Based on the score distributions for authentic and impostor comparisons the impact of specific factors on the authentication process can be evaluated. These can be: quality of acquisition, segmentation precision and feature extraction algorithm.

$$s = s(\hat{B}_1, \hat{B}_2) \quad (1)$$

The values of FAR and FRR are functions of the comparator threshold t (2) used to decide whether biometrics match each other. Selection of the threshold t (called also the operating point of the biometric system) depends on the assumption of safety and convenience tradeoff of the biometric system.

$$\begin{aligned} s \leq t &\Rightarrow \text{biometrics match} \\ s > t &\Rightarrow \text{biometrics do not match} \end{aligned} \quad (2)$$

Appropriate functions FAR(t) and FRR(t) describing the error rate as a function of the threshold t can be determined by the distribution functions for the null hypothesis H_0 (biometrics match) and the alternative hypothesis H_a (biometrics do not match) (3).

$$\begin{aligned} FRR(t) &= P(s > t | H_0) \\ FAR(t) &= P(s \leq t | H_a) \end{aligned} \quad (3)$$

The value of False Rejection Rate (FRR) is the probability of scores being greater than a certain threshold t , assuming that the null hypothesis H_0 is true. Accordingly, the value of False Acceptance Rate is the probability of scores being less than a certain threshold t , assuming that the alternative hypothesis H_a is true. In both cases it is assumed that a low scoring value calculated by the biometric comparator is equivalent to strong evidence that the compared biometrics are conformable.

Recognized criteria for assessing the interaction between error rates FAR and FRR, defined for any value of the decision threshold t , are Receiver Operating Characteristics, called also ROC curves (4) [4].

$$ROC(t) = (FAR(t), FRR(t)) \quad (4)$$

The obtained estimates of FAR and FRR curves cannot yet definitively decide about the general performance of a biometric system. Keeping in mind that the data source is a random process, one would know certainty of the obtained results for a given t . For this purpose, an additional element of statistical analysis of biometric data is the estimation of confidence bounds for FAR and FRR curves. In other words it is necessary to find intervals of true average error rate values for false rejection and false acceptance functions, on a certain confidence level $(1-\alpha)$ for many possible decision threshold values t (5).

$$\begin{aligned} E_{FRR}(t) &\in (\overline{FRR}(t) - \Delta FRR_1(t, \alpha), \overline{FRR}(t) + \Delta FRR_2(t, \alpha)) \\ E_{FAR}(t) &\in (\overline{FAR}(t) - \Delta FAR_1(t, \alpha), \overline{FAR}(t) + \Delta FAR_2(t, \alpha)) \end{aligned} \quad (5)$$

In this particular case the best solution for such estimation is to use the bootstrap method proposed by Efron modified by the author [5].

The main aim of the mentioned analysis was the investigation of segmentation algorithm accuracy influence on the final authentication result. The precision of the algorithm is graded as follows:

- pupil segmentation by b-spline curve and iris outer boundary segmentation by ellipse with occlusions masking – Precise variant;
- pupil and iris outer boundary segmentation by non-concentric circles with occlusions masking – Simple variant;
- pupil segmentation by circle and iris outer boundary segmentation by ellipse with occlusions masking – mixed variant 1 (CircleEllipse);
- pupil segmentation by b-spline curve and iris outer boundary segmentation by circle with occlusions masking – mixed variant 2 (SplineCircle).

The source for scoring are comparisons among images contained in two databases: ISEP (images in visible light) and BATH (images in the near-infrared light).

On the basis of preliminary observations of ROC curves, due to the large confidence intervals at 90% level, it was difficult to identify the ultimate impact of segmentation accuracy to the performance of the biometric system. Therefore, for each individual comparison an additional differential scoring was calculated in order to detect changes in score value caused by accuracy increase of the segmentation algorithm. As a reference, scores generated for a simple segmentation variant were assumed (6).

$$\begin{aligned} \Delta s_{Simple-\{Precise, SplineCircle, CircleEllipse\}} &= s_{Simple} - s_{\{Precise, SplineCircle, CircleEllipse\}} && \text{for } H_0 \\ \Delta s_{Simple-\{Precise, SplineCircle, CircleEllipse\}} &= s_{\{Precise, SplineCircle, CircleEllipse\}} - s_{Simple} && \text{for } H_a \end{aligned} \quad (6)$$

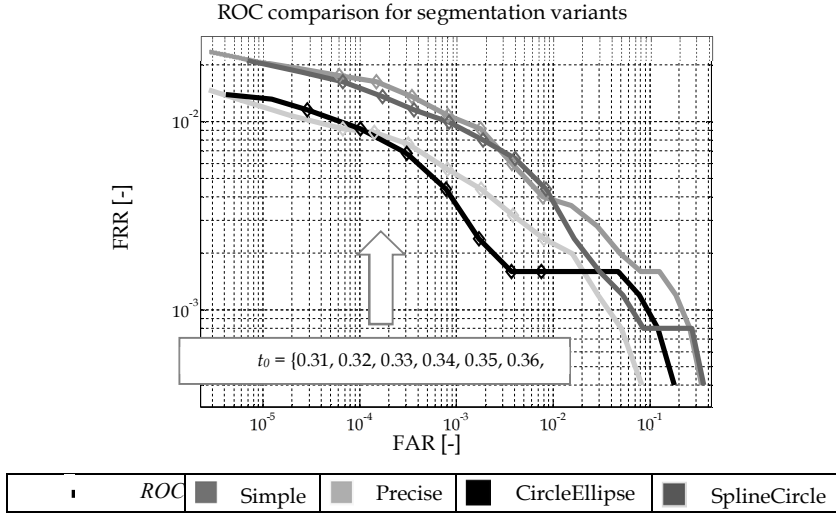


Fig. 3. ROC Curves for four variants of segmentation algorithm applied in ISEP database. Selected operating points are indicated

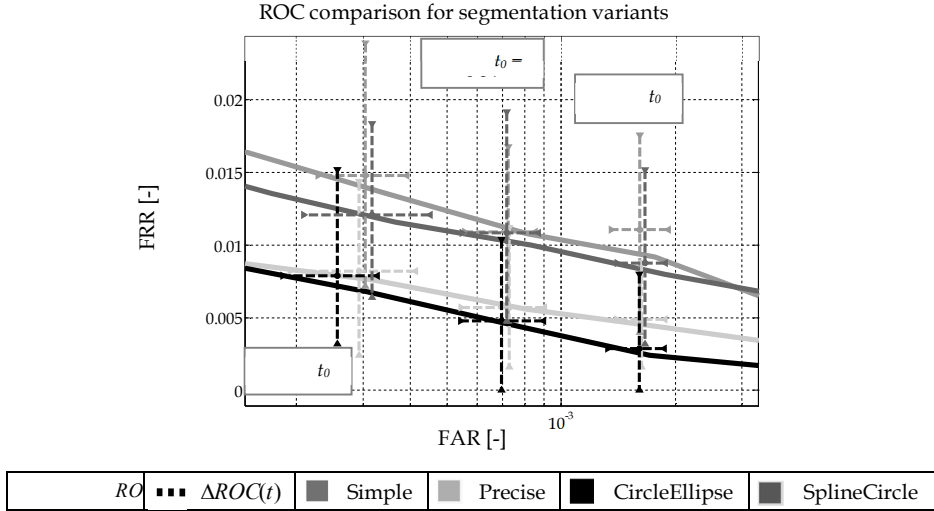


Fig. 4. ROC curves fragments for four variants of segmentation algorithm applied in ISEP. Confidence bounds at 90% level are indicated for selected operating points of the system

An example of the relative scores presented in figure 5, shows almost normal distributions. The average values of these distributions are positive, but very close to zero. For the ISEP database a slight predominance of comparisons, where an increase of the segmentation precision results in a reduction in the scoring for H_0 hypothesis, is present. Furthermore, for the alternative hypothesis H_a the dominance of comparisons, where an increase of the segmentation precision results in a deterioration of the results, is obtained (shifting the distribution toward smaller score values).

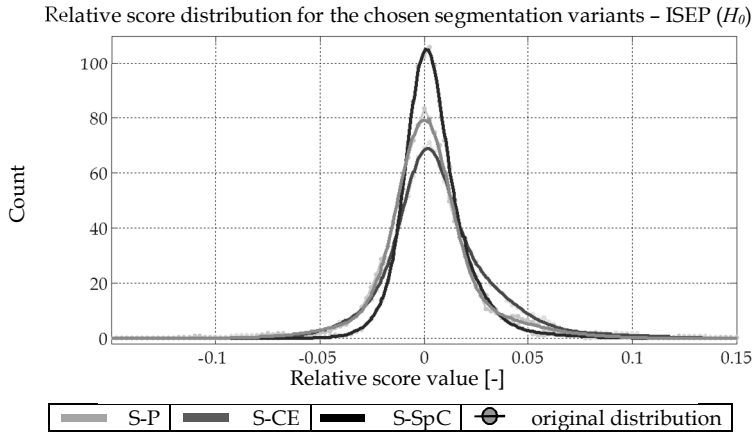


Fig. 5. Relative score distribution for the chosen segmentation variants for null hypothesis and ISEP database

It should be noted that in each of these cases, the distribution of the relative score values extends to both positive and negative, which means almost equal number of improvements and deteriorations in identification as a result of the segmentation accuracy increase.

Table 3. Comparison of the distributions parameters of the relative scores for ISEP database (for a confidence level at 95%)

	Mean relative score value (ISEP database)	
	$s H_0$ comparisons	$s H_a$ comparisons
$\Delta s_{Simple-Precise}$	$(8.8 \cdot 10^{-3})^{+69.9}_{-48.7}$	$(-1.0 \cdot 10^{-3})^{+59.5}_{-60.9}$
$\Delta s_{Simple-CircleEllipse}$	$(4.2 \cdot 10^{-3})^{+60.0}_{-42.2}$	$(0.8 \cdot 10^{-3})^{+54.1}_{-55.4}$
$\Delta s_{Simple-SplineCircle}$	$(4.7 \cdot 10^{-3})^{+37.6}_{-27.1}$	$(-1.8 \cdot 10^{-3})^{+41.0}_{-41.3}$

4. CONCLUSIONS

All stages of the iris image processing are the possible sources of errors resulting in an incorrect person identification. Firstly, the analysis related to true reasons of the above mentioned errors was done. Several improvements were proposed to help typical iris recognition systems become more reliable. The above analysis leads to the conclusion that increasing the segmentation accuracy by applying a precise modeling of iris borders curves, does not cause a significant increase in the number of comparisons for which the authentication process result gives additional evidence to support any of the hypotheses. Wide confidence intervals of the FAR and FRR estimates show a large dispersion in biometric scoring, which means that the feature extraction methods based on a texture analysis are not very stable in the sense of the results of authentication.

REFERENCES

- [1] **Wistrand P., Stjernschantz J., Olsson K.:** The incidence and time-course of latanoprost-induced iridial pigmentation as a function of eye color, *Survey of Ophthalmology* 41(S2), 1997, pp. S129-S138.
- [2] **Sankowski W.:** Algorytmy segmentacji obrazu oka, dla potrzeb systemu identyfikacji osób w oparciu o wzór tęczówki, *Rozprawa doktorska*, Politechnika Łódzka, 2009.
- [3] **Sankowski W., Grabowski K., Napieralska M., Zubert M.:** Eyelids Localization Method Designed for Iris Recognition System, Proceedings of the 14th International Conference of Integrated Circuit and Systems (MIXDES '07), Ciechocinek, Poland 21-23 June 2007, pp. 622-627.
- [4] **Bolle R. M., Connel J. H., Pankanti S., Ratha N. K., Senior A. W.:** Guide to Biometrics, Springer Science&Buisness Media Inc., New York NY, USA, 2004Omega. The temperature Handbook 2000. Vol. 29.
- [5] **Grabowski K.:** Analiza wydajności systemów biometrycznych na przykładzie baz tęczówek ISEP oraz BATH, Raport wewnętrzny, Politechnika Łódzka, 2009.

IDENTYFIKACJA OSÓB NA PODSTAWIE ANALIZY OBRAZU TĘCZÓWKI OKA ZE SZCZEGÓLNYM UWZGLĘDNIENIEM MOŻLIWOŚCI IMPLEMENTACJI SPRZĘTOWEJ

Streszczenie

W artykule przedstawiono główne rezultaty badań zawartych w rozprawie autora dotyczącej systemów uwierzytelniania osób na podstawie obrazu tęczówki

oka. Zaprezentowano dwa główne wątki pracy doktorskiej: segmentacji obrazu tęczówki i jej wpływu na proces ekstrakcji cech oraz metod analizy wydajności biometrycznej systemów uwierzytelniania.

Promotor prof. dr hab. Andrzej Napieralski

Recenzenci pracy doktorskiej:

1. prof. dr hab. inż. Kazimierz Wiatr, AGH
2. dr hab. inż. Krzysztof Ślot, prof. PŁ

CEZARY MAJ

Technical University of Łódź

Department of Microelectronics and Computer Science

MODELLING AND OPTIMIZATION OF ELECTROSTATIC MEMBRANE-BASED ACTUATORS

Reviewers: prof. **dr hab. inż. Andrzej Napieralski**

Manuscript received: 8.07.2009

The paper presents methods of electrostatic membrane-based actuators modelling. Based on a combination of the classical and reduced model, a simplified approach fully describing the membrane behaviour under hydrostatic and electrostatic load for various conditions, is proposed. The obtained results have been verified experimentally for various shapes of membranes. Furthermore, a new complex tool for simulation and optimization has been developed.

1. INTRODUCTION

The use of MEMS devices has grown rapidly in last decades. They are used in various markets e.g. automotive, industrial, consumer, military. The most promising domain concerns the biomedical application, in which a variety of sensor types and actuators are commonly used. Due to these facts, a proper design method is very important for the project. It affects crucial factors such as project duration, cost and even its success. Thus, the main step performed in the project, i.e. simulations, should be fast and precise enough. In case of electrostatic actuator, the calculations are complicated due to nonlinear nature of electrostatic attraction. Therefore, the use of an analytical method of modelling instead of commonly used FEM simulations seems to be a good alternative. Consequently, the paper presents a new analytical approach to modelling and optimization of electrostatic membrane-based actuators.

2. MODELLING OF ELECTROSTATIC MEMBRANE-BASED ACTUATOR

Mechanical behaviour of a thin plate ($10 < a/h < 80$, named later ‘a membrane’) is given by Newton-Lagrange equation [1]:

$$D\Delta\Delta(w(x,y) - w_0(x,y)) + \sigma_0 h \Delta w(x,y) = P \quad (1)$$

where a is a membrane length, h is a membrane thickness, w is a total membrane deflection, w_0 is an initial membrane deflection, P is an applied pressure, σ is a residual stress, α is a coefficient of anisotropy, ν is a Poisson ratio, E is a Young’s modulus, D_0 is a membrane rigidity, D is a flexural rigidity given by:

$$D = D_0 h^3, \quad D_0 = \frac{E}{12(1 - \nu^2)} \quad (2)$$

Δ is a Laplace operator and $\Delta\Delta$ is an operator as follows:

$$\Delta\Delta = \frac{\partial^4}{\partial x^4} + 2\alpha \frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4}$$

Equation (1) is solved using Galerkin [2] method taking the following membrane form [3]:

$$w(u,v) = (1 - u^2)^2 (1 - v^2)^2 \sum_{i,j=0}^n K_{ij} u^{2i} v^{2j} \quad (3)$$

where u and v are coordinates normalized to membrane length a and width b . The dependence between the applied pressure and a membrane deflection is linear. Thus, the model can be reduced into linear form:

$$k_{mem} w_{max} = P + P_{w0_max}, \quad k_{mem} = C_1 \frac{D}{b^4} + C_2 \frac{\sigma_0 h}{b^2} \quad (4)$$

where C_1 and C_2 are constants dependent on ratio $R=a/b$ and C_2 on coefficient of anisotropy also. Since this model operates on maximal membrane deflection w_{max} only, it does not give any information about the membrane form. Moreover, the model is valid for a specific range of residual stress.

In case of electrostatic actuation of the membrane [4], equation (1) takes the following form:

$$D\Delta\Delta(w(x,y) - w_0(x,y)) + \sigma_0 h \Delta w(x,y) = P + \varepsilon \frac{V^2}{2(d - w(x,y))^2} \quad (5)$$

where V is an applied voltage and d is a distance between the membrane and the bottom electrode. Equation (5) is solved in the same way as equation (1). However, it needs an iterative procedure (electrostatic force depends on membrane position), in which the following integral has to be calculated numerically [5]:

$$\iint_{\Omega} \frac{1}{(d-w)^2} \varphi_{i,j} dudv \quad (6)$$

Therefore, the use of the reduced model is desirable in order to shorten calculation time:

$$k_{mem} w_{max} = P_h + \varepsilon \frac{V^2}{2(d-w_{max})^2} \quad (7)$$

Simulations showed that this model is not correct in whole range of applied voltage (Fig. 1a). To correct this model, two additional coefficients A and B were introduced into equation (7), which depends on membrane ratio R and material properties:

$$k_{mem} w_{max} = P_h + B\varepsilon \frac{V^2}{2(d-Aw_{max})^2} \quad (8)$$

Moreover, the form of characteristic $w = f(V)$ does not fit the real one and has to be corrected by a function showed in (Fig. 1).

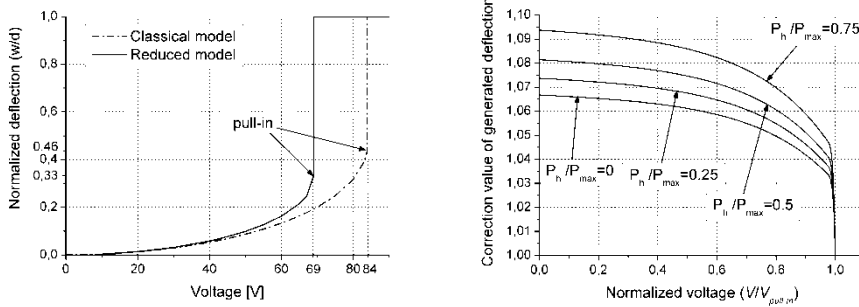


Fig. 1. a) comparison of reduced and classical model, b) function correcting the form of characteristic $w = f(V)$

FEM simulations showed that the corrected model gives accurate results and that calculation time is much shorter than for a FEM simulator (approximately 50 ms instead of 6 minutes needed to trace a full characteristic of the actuator).

3. OPTIMIZATION AND STATISTICAL APPROACH

An optimization phase is very important because it affects the project success (cost, duration, etc.). This step should contain precise simulations, which would not be time consuming. Therefore, the use of the corrected reduced model is a good alternative. The optimization loop contains then only fast calculations and only one FEM simulation is needed to verify the results (Fig. 2).

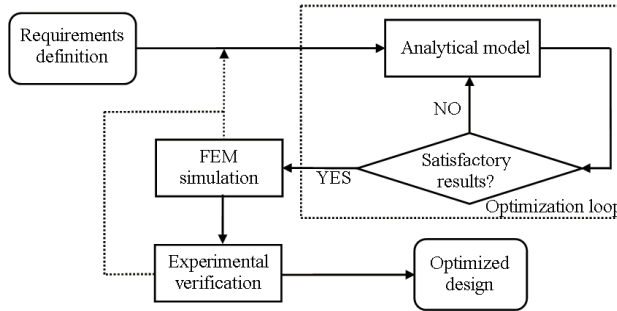


Fig. 2. Optimization phase

The reduced model may be used also in a statistical simulation in order to characterize a technological process. The optimization phase gives us the information about optimal design. However, it is not possible to fabricate such device as the technological process has some dispersion due to inaccuracy of machines and human errors. It may affect the device performance significantly and result in a decrease of yield production. Therefore, a fast statistical simulation giving the information about the device performance before the fabrication, may be useful in finding project weaknesses (Fig. 3).

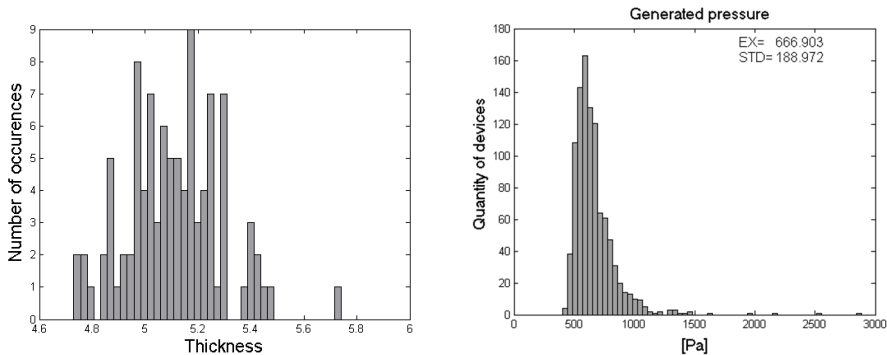


Fig. 3. a) Dispersion in membrane thickness (measurements of SOI thickness made by a vendor) and b) its influence on generated pressure

4. FABRICATION AND CHARACTERIZATION

The fabrication of test structures was performed using classical technological process [6]. The main steps were: oxidation of Si wafers and cavity etching, bonding with SOI wafers and creating contacts on topside. The use of Si wafers as a substrate eliminates the necessity of electrode deposition on cavity bottom. The SOI wafers simplify the membrane fabrication and define the proper thickness of the membrane (SOI layer forms the membrane).

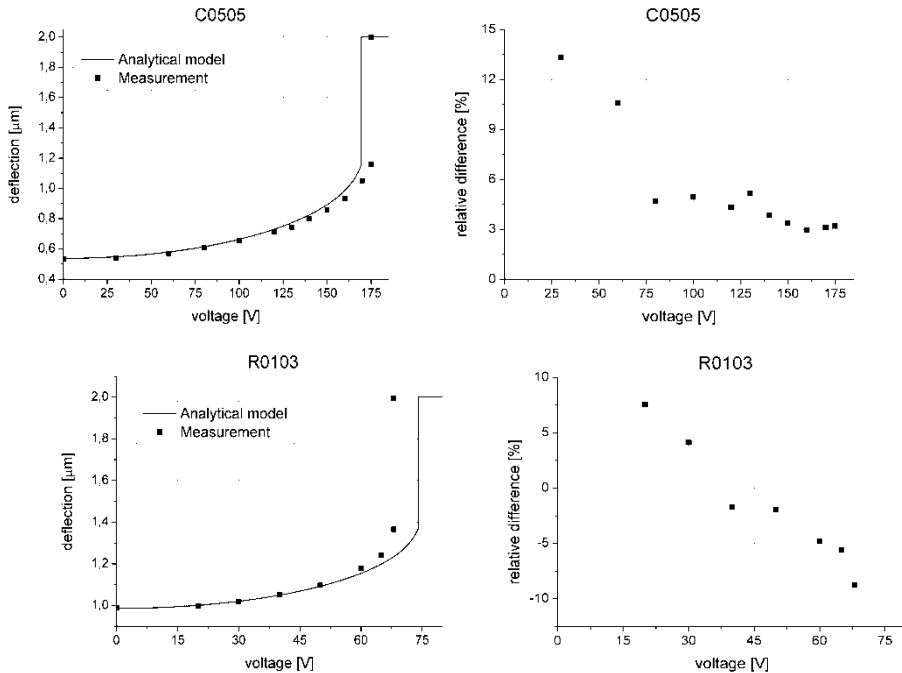


Fig. 4. Exemplary membrane deflection in function of applied voltage (left) and corresponding relative difference between analytical model and measurements (right) for square (top) and rectangular (bottom) membranes

The characterization required the following steps: measurement of membrane initial deflection in vacuum (vacuum inside the cavity), measurement of membrane response to the hydrostatic pressure in order to estimate the value of residual stress, measurement of membrane response to applied voltage. In our experiment square and rectangular membranes, fabricated in different conditions, were measured. A membrane initial deflection varied in range of -310 nm to 250 nm showing that fabrication process was not very reproducible (bonding especially) and that membrane parameters depended on its position on the wafer. Residual stress was quite reproducible and was about -25 MPa and

-50 Mpa for square and rectangular membrane, respectively. The measurements under applied voltage verified successfully the proposed analytical model. The difference did not exceed 20%, even though the analytical model has some limitations (the most important is ideal clamping).

5. CONCLUSIONS

In this paper:

1. Analytical models of electrostatic membrane-based actuators have been studied in terms of accuracy and calculation time.
2. The existing reduced model has been corrected to be fully applicable for electrostatic load.
3. The corrected model has been used in optimization of an actuator and in statistical simulation of technological process. Dedicated tools have been written in MATLAB® environment.
4. The developed model has been successfully verified by the conducted experiment.

REFERENCES

- [1] **Timoschenko S., Gere J.:** Theory of elastic stability, McGraw-Hill, 1963.
- [2] **Timoschenko S., Gere J.:** Theory of plates and shells, 2nd ed., McGraw-Hill, 1959.
- [3] **Blasquez G., Naciri Y.:** Static response of capacitive pressure sensor with square or rectangular silicon diaphragm, *Revue de Physique Appliquée*, Vol. 22/7 (1987).
- [4] **Marquès A. F., Castelló R. C.:** A. M. Shkel, Modelling the electrostatic actuation of MEMS: state of the art 2005, Universitat Politècnica de Catalunya.
- [5] **Francais O., Dufour I.:** Normalized abacus for the global behavior of diaphragms: pneumatic, electrostatic, piezoelectric or electromagnetic actuation, *Journal of Modeling and Simulation of Microsystems*, 1999, Vol. 1, No. 2.
- [6] **Huang Y.:** Fabricating Capacitive Micromachined Ultrasonic Transducers with Wafer-Bonding Technology, *Journal of Microelectromechanical Systems*, 2003, Vol. 12, No.2.

MODELOWANIE I OPTYMALIZACJA MIKROSYSTEMÓW STEROWANYCH SIŁĄ ELEKTROSTATYCZNĄ OPARTYCH NA MEMBRANIE

Streszczenie

W niniejszym artykule zostały przedstawione metody modelowania aktuatorów sterowanych elektrostatycznie, opartych na membranie. Zaproponowane

podejście, które w pełni opisuje zachowanie membrany pod wymuszeniem hydrostatycznym i elektrostatycznym dla wielu założeń, oparte jest na połączeniu modelu klasycznego i zredukowanego. Wyniki zostały potwierdzone eksperymentalnie dla różnych kształtów membran. Dodatkowo zostały stworzone narzędzia do kompletnej symulacji i optymalizacji.

Promotorzy: prof. dr hab. inż. Andrzej Napieralski, dr inż. Patrick Pons
Recenzenci pracy doktorskiej:

1. dr hab. inż. Sylvain Ballandras
2. prof. Robert Puers
3. prof. Jan Dziuban
4. prof. Zygmunt Ciota



ISSN 0374-4817