Advances in Intelligent Systems and Computing 466

Radek Silhavy
Roman Senkerik
Zuzana Kominkova Oplatkova
Petr Silhavy
Zdenka Prokopova *Editors*

Automation Control Theory Perspectives in Intelligent Systems

Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Vol 3

Springer

Advances in Intelligent Systems and Computing

Volume 466

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Advisory Board

More information about this series at http://www.springer.com/series/11156

Radek Silhavy · Roman Senkerik
Zuzana Kominkova Oplatkova
Petr Silhavy · Zdenka Prokopova
Editors

Automation Control Theory Perspectives in Intelligent Systems

Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Vol 3

Springer

Editors
Radek Silhavy
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín
Czech Republic

Petr Silhavy
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín
Czech Republic

Roman Senkerik
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín
Czech Republic

Zdenka Prokopova
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín
Czech Republic

Zuzana Kominkova Oplatkova
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín
Czech Republic

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-33387-8 ISBN 978-3-319-33389-2 (eBook)
DOI 10.1007/978-3-319-33389-2

Library of Congress Control Number: 2016937381

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Preface

This book constitutes the refereed proceedings of the Automation Control Theory Perspectives in Intelligent Systems Section and of the Intelligent Information Technology, System Monitoring and Proactive Management of Complex Objects Section of the 5th Computer Science On-line Conference 2016 (CSOC 2016), held in April 2016.

The volume Automation Control Theory Perspectives in Intelligent Systems brings 47 of the accepted papers. Each of them presents new approaches and methods to real-world problems and exploratory research that describes novel approaches in the field of cybernetics, automation control theory and proactive management of complex objects.

CSOC 2016 has received (all sections) 254 submissions, 136 of them were accepted for publication. More than 60 % of all accepted submissions were received from Europe, 20 % from Asia, 16 % from America and 4 % from Africa. Researchers from 32 countries participated in CSOC 2016.

CSOC 2016 intends to provide an international forum for the discussion of the latest high-quality research results in all areas related to computer science. The addressed topics are theoretical aspects and applications of computer science, artificial intelligence, cybernetics, automation control theory and software engineering.

Computer Science On-line Conference is held online and broad usage of modern communication technology improves the traditional concept of scientific conferences. It brings equal opportunity to participate to all researchers around the world.

The editors believe that readers will find the proceedings interesting and useful for their own research work.

March 2016

Radek Silhavy
Roman Senkerik
Zuzana Kominkova Oplatkova
Petr Silhavy
Zdenka Prokopova

Program Committee

Program Committee Chairs

Zdenka Prokopova, Ph.D., Associate Professor, Tomas Bata University in Zlín, Faculty of Applied Informatics, email: prokopova@fai.utb.cz

Zuzana Kominkova Oplatkova, Ph.D., Associate Professor, Tomas Bata University in Zlín, Faculty of Applied Informatics, email: kominkovaoplatkova@fai.utb.cz

Roman Senkerik, Ph.D., Associate Professor, Tomas Bata University in Zlín, Faculty of Applied Informatics, email: senkerik@fai.utb.cz

Petr Silhavy, Ph.D., Senior Lecturer, Tomas Bata University in Zlín, Faculty of Applied Informatics, email: psilhavy@fai.utb.cz

Radek Silhavy, Ph.D., Senior Lecturer, Tomas Bata University in Zlín, Faculty of Applied Informatics, email: rsilhavy@fai.utb.cz

Roman Prokop, Ph.D., Professor, Tomas Bata University in Zlín, Faculty of Applied Informatics, email: prokop@fai.utb.cz

Program Committee Chairs for Special Sections

Intelligent Information Technology, System Monitoring and Proactive Management of Complex Objects

Prof. Viacheslav Zelentsov, Doctor of Engineering Sciences, Chief Researcher of St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (SPIIRAS)

Program Committee Members

Boguslaw Cyganek, Ph.D., D.Sc., Department of Computer Science, University of Science and Technology, Krakow, Poland

Krzysztof Okarma, Ph.D., D.Sc., Faculty of Electrical Engineering, West Pomeranian University of Technology, Szczecin, Poland

Monika Bakosova, Ph.D., Associate Professor, Institute of Information Engineering, Automation and Mathematics, Slovak University of Technology, Bratislava, Slovak Republic

Pavel Vaclavek, Ph.D., Associate Professor, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czech Republic

Miroslaw Ochodek, Ph.D., Faculty of Computing, Poznań University of Technology, Poznań, Poland

Olga Brovkina, Ph.D., Global Change Research Centre Academy of Science of the Czech Republic, Brno, Czech Republic

Elarbi Badidi, Ph.D., College of Information Technology, United Arab Emirates University, Al Ain, United Arab Emirates

Luis Alberto Morales Rosales, Head of the Master Program in Computer Science, Superior Technological Institute of Misantla, Mexico

Mariana Lobato Baes, M.Sc., Research-Professor, Superior Technological of Libres, Mexico

Abdessattar Chaâri, Professor, Laboratory of Sciences and Techniques of Automatic Control and Computer engineering, University of Sfax, Tunisian Republic

Gopal Sakarkar, Shri. Ramdeobaba College of Engineering and Management, Republic of India

V.V. Krishna Maddinala, Assistant Professor, GD Rungta College of Engineering and Technology, Republic of India

Anand N. Khobragade, Scientist, Maharashtra Remote Sensing Applications Centre, Republic of India

Abdallah Handoura, Assistant Prof., Computer and Communication Laboratory, Telecom Bretagne, France

Technical Program Committee Members

Ivo Bukovsky
Miroslaw Ochodek
Bronislav Chramcov
Eric Afful Dazie

Michal Bliznak
Donald Davendra
Radim Farana
Zuzana Kominkova Oplatkova
Martin Kotyrba
Erik Kral
David Malanik
Michal Pluhacek
Zdenka Prokopova
Martin Sysel
Roman Senkerik
Petr Silhavy
Radek Silhavy
Jiri Vojtesek
Eva Volna
Janez Brest
Ales Zamuda
Roman Prokop
Boguslaw Cyganek
Krzysztof Okarma
Monika Bakosova
Pavel Vaclavek
Olga Brovkina
Elarbi Badidi

Organizing Committee Chair

Radek Silhavy, Ph.D., Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: rsilhavy@fai.utb.cz

Conference Organizer (Production)

OpenPublish.eu s.r.o.
Web: http://www.openpublish.eu
e-mail: csoc@openpublish.eu

Conference Website, Call for Papers

http://www.openpublish.eu

Contents

Part I
Automation Control Theory Perspectives in Intelligent Systems

A Novel Color Image Encryption Algorithm Using Chaotic Map and Improved RC4

Cong Jin and Zhengwu Tu

Abstract In this paper, color image encryption algorithm based on improved RC4 and chaotic maps is proposed. In proposed algorithm, the classic RC4 algorithm in cryptography is improved, and then applied to proposed encryption algorithm. Firstly, the original image is divided into some sub-blocks. Then, the improved RC4 algorithm is applied to the operation between the adjacent two sub-blocks, thereby changes the value of pixels. Finally, the image is scrambled by logistic map. Experimental results show that the original image has a big change and a flat histogram after encrypted, and that the proposed algorithm has an enough large key space and a very high sensitivity to the key.

Keywords Chaotic maps · Image encryption · RC4 algorithm · Logistic map

1 Introduction

A good image encryption algorithm should have a high security, good encryption effect and spend a little time. Especially in the transmission of network, the encryption speed is an important indicator in the real-time requirements of image encryption. The common problem in traditional image encryption is that the process of image encryption need a long time and has a poor real-time in image transmission. In order to improve the speed of encryption, we will divide the image into many pixel blocks and consider pixel block as the minimum operation unit. Recently, a variety of image encryption based on blocks is proposed [1]. These encryption algorithms have a fast speed of encryption. The traditional encryption algorithm in modern cryptography has a high security for text data. In encryption theory, digital image can be encrypted by encryption algorithm of modern cryptography. But digital image is a kind of special data, which is featured with huge data capacity, two-dimensional data, and high redundancy and so on. Those

C. Jin (✉) · Z. Tu
School of Computer, Central China Normal University, Wuhan 430079, Hubei, China
e-mail: jincong@mail.ccnu.edu.cn

R. Silhavy et al. (eds.), *Automation Control Theory Perspectives in Intelligent Systems*, Advances in Intelligent Systems and Computing 466,
DOI 10.1007/978-3-319-33389-2_1

encryption algorithms of modern cryptography, such as Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA) and Advanced Encryption Standard (AES) [2], etc., are designed for text data which is one-dimension, and doesn't combine with the feature of image data. So it is very difficult to satisfy the image encryption. But we can modify the traditional cipher algorithms to be suitable for the image encryption. The proposed algorithm improves the classic RC4 algorithm in cryptography to encrypt the digital image.

The chaos system possesses of some characteristics, such as high sensitivity to initial values and system parameter, the statistical property of white noise and sequence ergodicity, and the characteristic of diffusion, permutation and randomness which conform to the requirements of cryptography. The chaotic maps can export the pseudo-random sequences whose structures are very complex and difficult to be analyzed and predicted. Researchers encrypt the digital image by the chaotic map, which reflects on three aspects: (1). Change the pixel value of original image by the pseudo-random sequences which are exported by the chaotic maps [3]; (2). Scramble the pixel position of original image by the chaotic maps [4]; (3). Combine two methods above. So far, many image encryption based on chaotic maps have been proposed [5]. Specific to these characteristics of the chaos system, the proposed algorithm applies the chaotic maps to change the pixel value and scramble the pixel position. Combined with the above points, color image encryption algorithm based on modified RC4 and chaotic maps is proposed. Proposed algorithm modifies the classical RC4 algorithm in cryptography and is applied to the encryption process of digital image. So this makes that the pixel value of the image changes largely, and conforms to the security and reliability in the process of transmission. Firstly, the original image is divided into blocks whose size is 8×8. Otherwise, a set of pseudo-random number are got by the iteration of logistic map and consist of a pseudo-random block whose size is 8×8. Secondly, the improved RC4 algorithm is operating between the adjacent blocks. Finally, the image is scrambled by the logistic map to generate the encrypted image.

2 Encryption Algorithm

The encryption algorithm is composed by five parts. These parts are divided into blocks, generating the pseudo-random block, operation between blocks, modified RC4 and scrambling the image. The architecture is shown in Fig. 1.

2.1 Divide Image into Several Blocks

As is assumed, the size of original image I is $M \times N$, so the pixel value of original image I is $I(i, j)$, $i = 0, 1, \ldots, M - 1$; $j = 0, 1, \ldots, N - 1$. The original image I is divided into blocks, whose size is 8×8. So the original image can be divided into

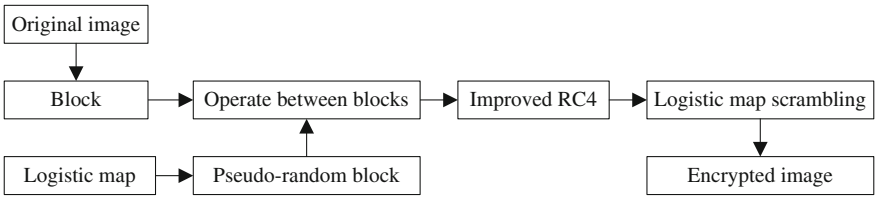

Fig. 1 Architecture of the encryption algorithm

$m \times n$ sub-blocks, which are represented as $I_{s,t,}$ $s = 0, 1, \ldots, m - 1$; $t = 0, 1, \ldots, n - 1$ and in which the pixel can be represented as $I_{s,t}$ (p, q), $p = 0, 1,\ldots,7$; $q = 0, 1,\ldots,7$. $\begin{cases} m = M/8 + \lceil M\%8 \rceil \\ n = N/8 + \lceil N\%8 \rceil \end{cases}$. Where $\lceil \rceil$ is the ceil operation and % is the mod operation. The formula is the conversion between the original image and sub-block $I_{s,t}(p,q) = I(i,j)$. Where $s = i/8, t = j/8, p = i\%8, q = j\%8$.

2.2 Generate Pseudo-Random Block

This pseudo-random block is composed of a set of pseudo-random numbers which is got by the iteration of the logistic map. The logistic map is described by $x_{n+1} = 4x_n(1 - x_n)$, $x_0 \in (0, 1), x_0 \neq 0.5$.

Step1 The initial value x_0 of the logistic map is got by the following formula $x_0 = key_1/10^8$. Where, the range of the key key_1 is 1-99999999.

Step2 Iterate the logistic map for N_0 times. In the experiment, N_0 is 100.

Step3 We continue to iterate the logistic map 8×8 times, and obtain a pseudo-random sequence $X = \{x_1, x_2,\ldots, x_{64}\}$ from the state value. We can get a new sequence $L = \{l_1, l_2,\ldots, l_{64}\}$ by the formula $l_i = x_i \times 2^{24}$.

Step4 The pseudo-random block J is composed of the sequence $L = \{l_1, l_2,\ldots, l_{64}\}$ by the formula $J(i, j) = l_{j+8\times i}$.

2.3 Operation Between Blocks

In the encryption algorithm, the operation between blocks uses a kind of improved RC4 algorithm. The operator of this operation is recorded as $\otimes$. In this paper, every sub-block of the image can be represented as $I_{s,t}$, which is same as I_u, $u = 0, 1,\ldots, mn-1$. $I_u = I_{t+8\times s} = I_{s,t}$ In the operation between blocks, the first sub-block I_0 operates with the pseudo-random block J, and generates a new first block, which is still recorded as I_0. Then I_1 operates with I_0, and generates a new second block,

which is still recorded as I_1. It goes on as so until the last sub-block. The calculating formula is as follows

$$\begin{cases} I_u = I_u \otimes J, when\, u = 0 \\ I_u = I_u \otimes I_{u-1}, when\, 0 < u < mn \end{cases}$$

2.4 Improved RC4 Algorithm

The operation of the improved RC4 algorithm which is recorded as $\otimes$ between adjacent sub-blocks $I_u(p, q)$, $I_{u\text{-}1}(p, q)$ is described as follows:

Step1 The two-dimensional sub-block $I_u(p, q)$, $I_{u\text{-}1}(p, q)$ is converted into one-dimension by $I_u(p,q) = I_u(q+8p) = I_u(k); I_{u-1}(p,q) = I_{u-1}(q+8p) = I_{u-1}(k)$.

Step2 We can define an array s[64] as S-box, then initialize S-box by the $s[i] = i, i = 0, 1, \ldots, 63$.

Step3 We randomly exchange two value of the S-box by the string key key_3 to disturb the S-box. The string key key_3 is converted into the byte array $k[i]$, then exchange s[i] and s[j] by $k[i] = key_3[i\%length], j = (j + s[i] + k[i])\%64$. Where *length* is length of the string key key_3 and i is the index of char in the string key key_3. The range of every char in the string key key_3 is 0-255, and this string key key_3 has at least 8 chars.

Step4 After finishing the exchange of S-box in Step2, we continue to exchange s [i] and s[j] of S-box, and get the index r which is the sum of i and j by the $j = (j + s[i])\%64, r = (s[i] + s[j])\%64$.

Step5 We can get a sequence $R = \{r_1, r_2, \ldots, r_{64}\}$, and XOR the r_ith pixel of the front sub-block with the ith pixel of the behind sub-block by the $I_u(i) = I_u(i) \oplus I_{u-1}(r_i)$.
Finally, we can merge all of the sub-blocks after XOR into the image E_1.

2.5 Scramble Image

We can scramble the pixel position of the image E_1 by the logistic map to get the encrypted image E_2. We can get a pseudo-random sequence by the logistic map, which is described as $x_{n+1} = 4x_n(1 - x_n); x_0 \in (0, 1), x_0 \neq 0.5$. In the scrambling process, the image is scrambled by the pseudo-random sequence, is described as follows:

Step1 We can get the initial value x_0 of the logistic map by $x_0 = key_2/10^8$. Where, the range of the key key_2 is 1-99999999.

Step2 Iterate the logistic map for N_0 times. In our experiments, N_0 is 100.

Step3 Traverse every pixel of the image E_1 in order, and iterate the logistic map for one time when accessing a pixel $E_1(i, j)$. So we can get a random number x_i, and a sequence number m_i by the following formula $m_i = \lfloor x_i \times (M \times N) \rfloor$. Where $M \times N$ is the size of the encrypted image E_2 and $\lfloor \rfloor$ is the floor operation.

Step4 If the pixel $E_2(p, q)$ is occupied, we will continue to carry out step3 until the $E_2(p, q)$ isn't occupied. So the pixel $E_2(p, q)$ is replaced $E_1(i, j)$, which is described as $E_2(p, q) = E_1(i, j)$. Where $p = m_i/M$, $q = m_i\%M$.
Finally, the image E_2 is the encrypted image which we get after the image E_1 is scrambled.

3 Decryption Algorithm

Compared with the encryption, the decryption is the inverse process of the encryption algorithm. The decryption algorithm is different from the encryption algorithm in only operation between blocks and scrambling the image. Firstly, the encrypted image E_2 is scrambled for the image E_1 by the logistic map, and the image E_1 is divided into 8×8 sub-blocks. Secondly, the pixels between adjacent sub-blocks have a XOR operation. Finally, the sub-blocks are merged into the image *I*.

3.1 Inversing Operation of Scrambling Image

In the process of decryption, the encrypted image E_2 is scrambled for the image E_1. In the process of scrambling the image, step1, step2 and step3 are same as the encryption algorithm, but step4 has a little difference. So we only show the process of step4, which is described as

Step4 If the pixel $E_2(p, q)$ is occupied, we will continue to carry out step3 until the $E_2(p, q)$ isn't occupied. So the pixel $E_1(i, j)$ is replaced $E_2(p, q)$, which is described as $E_1(i,j) = E_2(p, q)$.

3.2 Inversing Operation Between Blocks

As same as the encryption algorithm, the image E_1 is divided into sub-blocks, which is described as I_u, $u = 0, 1, \ldots, mn-1$. In the process of decryption, we begin

to operate from the last sub-block I_{mn-1}. The last sub-block I_{mn-1} operates with its front sub-block I_{mn-2} to get the last sub-block, which is still recorded as I_{mn-1}. The sub-block I_u operates with its front sub-block I_{u-1} until the first sub-block I_0 in order. Finally, the first sub-block I_0 operates with the pseudo-random block Logistic which is got by the logistic map. This is the opposite order of encryption. The operation is described as

$$\begin{cases} I_u = I_u \otimes J, when\, u = 0 \\ I_u = I_u \otimes I_{u-1}, when\, 0 < u < mn \end{cases}$$

4 Experiment Results

The experiment of the encryption algorithm is conducted under eclipse using java in a computer with an Intel Core i5 2.5 GHz and 4 GB RAM running Windows 8.0 operating system. In the experiment, we select *Airplane* with size 128 × 128, *Sailboat* with size 256 × 256 and *Pens* with size 512 × 512 as the original image. In encrypting the image we select a set of key with key_1 = 88888888, key_2 = 88888888, key_3 = abcdefgh. After the image is encrypted, we compare the original image with the encrypted image and the recovered image, which is shown as Fig. 2.

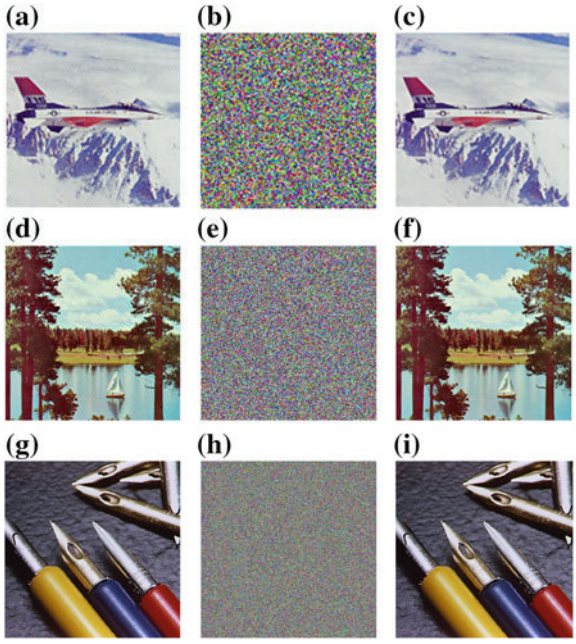

Fig. 2 Comparison after encrypted and recovered **a** *Airplane*, **b** encrypted *Airplane*, **c** recovered *Airplane*, **d** *Sailboat*, **e** encrypted *Sailboat*, **f** recovered *Sailboat*, **g** *Pens*, **h** encrypted *Pens*, **i** recovered *Pens*

In Fig. 2, the encrypted image is the noise-like image, and it can't be seen any useful information. Besides, the recovered image is same as the original image. So these experiments demonstrate that the proposed algorithm has an ideal performance.

5 Security Analysis

5.1 Key Space Analysis

In experiment, the key of encryption algorithm is composed of the key key_1, key_2 of the logistic map and the string key key_3 of the pixel XOR. The key key_1, key_2 are the decimal integer with 8 bits, and the key key_3 has at least eight chars. So the smallest key space of the proposed algorithm is 10^{32}. This key space is enough large to resist the exhaustive attack.

5.2 Key Sensitive Test

The key sensitive test is shown as Fig. 3. In experiments, we select *Pens* as the original image. This test encrypts the original image (a) to get the encrypted image (b) with the key K_1 (key_1 = 88888888, key_2 = 88888888, key_3 = abcdefgh) and decrypts the encrypted image (b) to get the recovered image (c) with the same key K_1. Then let the key change Key_1 change a little precision to get the key K_2 (key_1 = 88888889, key_2 = 88888888, key_3 = abcdefgh). This test decrypts the encrypted image (b) to get the recovered image (d) which is still a noise-like image with the key K_2.

This sensitive test shows that the original image can be completely recovered only when the correct security key is being utilized and that the decryption result is completely different from the correct recovered image when the key has a little precision change. Therefore, this demonstrates the proposed algorithm is also sensitive to its key changes during the decryption process.

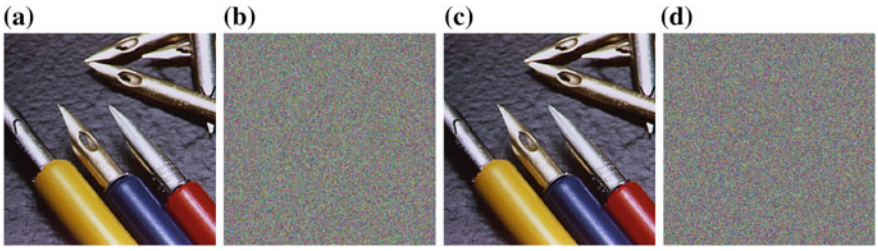

Fig. 3 Key sensitive test

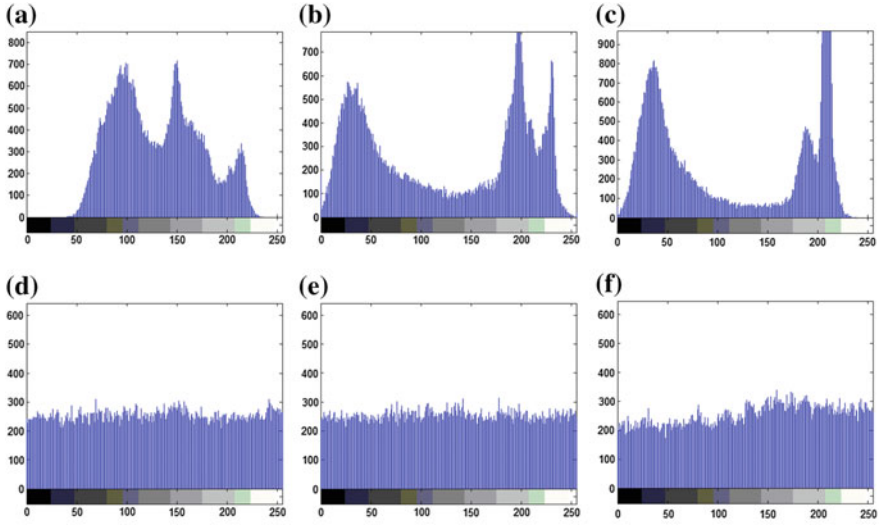

Fig. 4 Comparison of histogram on color components (**a**–**c**) is the histogram of red, green, blue color component in the original image, respectively; (**d**–**f**) is the histogram of red, green, blue color component in the encrypted image respectively

5.3 *Histogram Analysis*

Image histogram represents the intensity distribution of pixels within an image. An encrypted image with a flat histogram is able to resist statistic attacks. In this analysis, we select *Sailboat* image as the original image. Figure 4 compares the encrypted image with the original image on the histogram of red, green, blue color component.

The result of Fig. 4 shows that the histogram of red, green, blue color component in the encrypted image have a big change, which becomes flat and uniform. This change can resist statistic attacks, and the encrypted image is difficult to be decoded from the histogram of the encrypted image.

5.4 *Information Entropy Analysis*

Information entropy (IFE) is designed to evaluate the uncertainty in a random variable as shown in the equation $H_L = \sum_{l=0}^{F-1} P(L = l) \log_2 \frac{1}{P(L=l)}$.

Where F is the gray level and $P(L = l)$ is the percentage of pixels of which the value is equal to l.

Table 1 Information entropy analysis

File name	Image size	Original image	Encrypted image
Airplane.bmp	128 × 128	6.7754	7.9860
Sailboat.bmp	256 × 256	7.7741	7.9903

The IFE can be used for evaluating the randomness of an image. An IFE score of an image close to the maximum IFE value means the excellent random property. For a grayscale image with a data range of [0, 255], its maximum IFE is 8. For a color image, we can convert the color image into the grayscale image, and calculate the IFE. In the experiment of information entropy analysis, we select *Airplane* and *Sailboat* images as the original images. Table 1 shows the IFE scores of images before and after applying the proposed encryption algorithm.

From these results, the IFE scores of all encrypted images with different sizes are close to 8. The IFE scores of the encrypted image are closer to 8 than that of the original image, which means that the encrypted images after applying the proposed encryption algorithm have better random distributions.

5.5 Correlation Analysis

The original image has high correlations between pixels and their neighboring pixels at horizontal, vertical and diagonal directions. The encryption algorithm aims at breaking these pixel correlations in the original images, and transforming them into noise-like encrypted images with little or no correlations. The correlation values can be calculated by the following equation $C_{xy} = \frac{E[(x-\mu_x)(y-\mu_y)]}{\sigma_x \sigma_y}$. Where μ and σ are the mean value and standard deviation respectively, $E[\cdot]$ is the expectation value.

Hence, a good encrypted image should be unrecognized and has the correlation values close to zero. Correlation analysis here is to test the relationships of adjacent pixels in the original and encrypted images. 3000 pairs of two adjacent pixels are randomly chosen from the original and encrypted images in the horizontal, vertical and diagonal directions to perform this analysis. Table 2 compares correlations of the original image with its encrypted versions generated by different encryption algorithm. We select *Sailboat* image as the original image.

Table 2 Correlation values at the horizontal, vertical and diagonal directions

Image	Encryption algorithm	Horizontal	Vertical	Diagonal
Original image		0.9164	0.9140	0.8773
Encrypted image	Proposed algorithm	7.6182×10^{-4}	8.4534×10^{-4}	0.0081
Encrypted image	[6]'s	0.0141	0.0107	0.0097

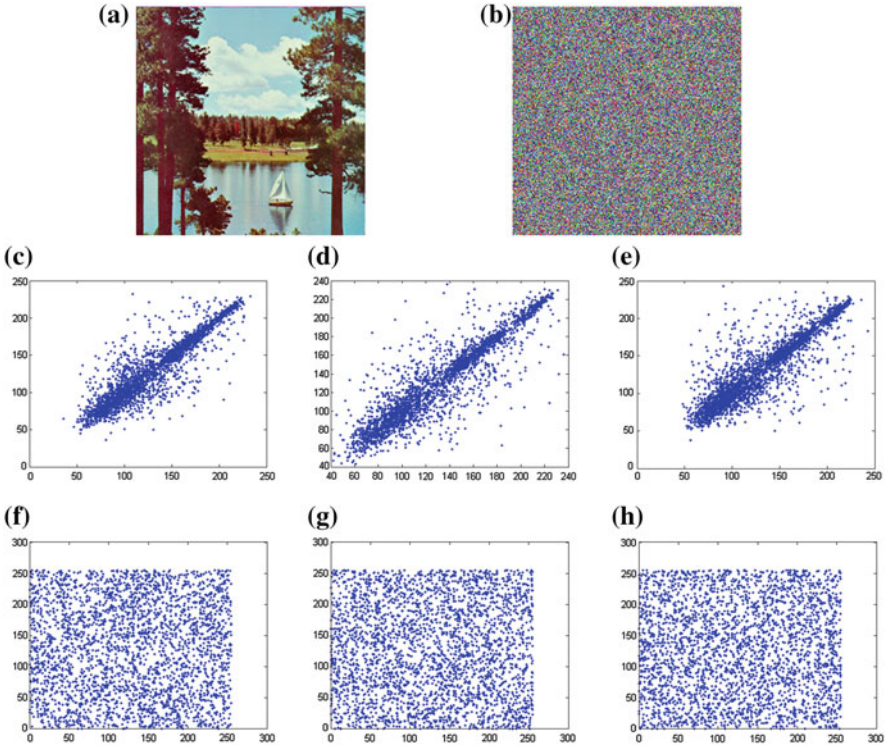

Fig. 5 Correlation of adjacent pixels at different directions

From the results of Table 2, the original image has high correlation values in all directions while all encrypted images have very low correlation values, which shows the excellent performance of image encryption algorithms. Furthermore, compared to the [6]'s algorithm, the proposed algorithm obtains the smaller correlation values in all directions. It out performs [6]'s algorithm with respect to performance of image encryption.

We can set the intensity values of adjacent pixel pairs as the horizontal and vertical axes, respectively. Figure 5 plots their distributions in three directions. In this paper, we select *Sailboat* image as the original image. If two adjacent pixels are equal, their tracks are located in the diagonal line. As can be seen, the distributions of adjacent pixels in the original image are around the diagonal line, which means the original image has high correlation. On the contrary, the distributions of adjacent pixels in the encrypted image disperse in the whole data range of the image, which proves that the encrypted image has extremely low correlation and show high randomness.

Table 3 Comparison of the encryption time of different algorithms

File name	Image size	Proposed algorithm (s)	[7] (s)
Sailboat.bmp	256 × 256	0.273	0.569
Pens.bmp	512 × 512	1.455	2.251

5.6 Speed Analysis

The proposed algorithm is mainly based on modified RC4 and chaotic maps. The basic operation of RC4 algorithm is XOR, whose time-consuming is very little. So the proposed algorithm has a high efficiency on speed. In the speed analysis, we select *Sailboat* and *Pens* images as the original images. Besides, we compare the encryption speed of proposed algorithm with [7]'s. The result is shown as Table 3.

As can be seen from Table 3, the proposed algorithm performs faster than [7]'s, which demonstrates that the proposed algorithm has a fast speed performance and is suitable for real applications.

6 Conclusions

This paper proposes an image encryption algorithm based on modified RC4 and chaotic maps. Through the experiment results and security analysis, the proposed algorithm not only has a high security, but also has a fast speed performance. The main advantages of proposed algorithm are as follows:

(1) The encrypted image is a noise-like image and has a flat and uniform histogram and a good randomness.
(2) The adjacent pixels of the encrypted image have very low correlation.
(3) The proposed algorithm has a large key space and is very sensitive to the key.

Because the main operation of the encrypted algorithm is XOR operation in modified RC4 algorithm and the pixel block is considered as the minimum operation unit, the encryption algorithm has a fast encryption speed.

Acknowledgments This work was supported by the fundamental research funds for the central universities (Grant No.20205001537).

References

1. Liu, H.J., Kadir, A., Niu, Y.J.: Chaos-based color image block encryption scheme using S-box. AEU-Int. J. Electron. Commun. **68**(7), 676–686 (2014)
2. Yin, A.H., Wang, S.K.: A novel encryption scheme based on time stamp in gigabit Ethernet passive optical network using AES-128. Optik **125**(3), 1361–1365 (2014)

3. Kadir, A., Hamdulla, A., Guo, W.Q.: Color image encryption using skew tent map and hyper chaotic system of 6th-order CNN. Optik **125**(5), 1671–1675 (2014)
4. Zhang, Z., Sun, S.L.: Image encryption algorithm based on logistic chaotic system and s-box scrambling. Image Sig. Process. **1**(4), 177–181 (2011)
5. Tong, X.J.: Design of an image encryption scheme based on a multiple chaotic map. Commun. Nonlinear Sci. Numer. Simul. **18**(7), 1725–1733 (2013)
6. Wang, X.Y., Chen, F., Wang, T.: A new compound mode of confusion and diffusion for block encryption of image based on chaos. Commun. Nonlinear Sci. Numer. Simul. **15**(9), 2479–2485 (2010)
7. Liao, X.F., Lai, S.Y., Zhou, Q.: A novel image encryption algorithm based on self-adaptive wave transmission. Sig. Process. **90**(9), 2714–2722 (2010)

Modified Discrete LQ Control Algorithm for Situations with the Scan Period Variance

Jan Cvejn

Abstract Computer-based control systems, especially if they run under general-purpose operating systems, often exhibit variance of the scan period of processing inputs and outputs. Although this fact is usually not taken into account when discrete control algorithms are used, it can cause worse performance of the control loop in comparison to the theoretical case. In this paper we describe a modified discrete LQ control algorithm that takes disturbances of the scan period into account and partially compensates their influence. We also show that such a controller can be implemented even on low-performance hardware platforms, if they are equipped with a sufficient amount of memory.

Keywords Optimal control · LQ controller · Linear systems · Discrete control

1 Introduction

In the control theory analysis and design of control algorithms in the continuous-time domain and in the discrete-time domain are studied separately. Continuous approach is natural for modelling and analysis of real processes and will be always used for designing controllers on the basis of analog components. Discrete approach seems to be natural for technical implementation of control algorithms on microprocessor-based platforms.

Discrete control algorithms rely upon constant period of processing inputs and outputs. However, constant scan period is often not fully guaranteed in real situations. This phenomenon can occur due to handling asynchronous hardware events in computer systems. Although this problem is typical for general-purpose multitasking operating systems, even the most robust hardware platforms such as PLCs

J. Cvejn (✉)
Faculty of Electrotechnics and Informatics, University of Pardubice,
Studentská 95, 532 10 Pardubice, Czech Republic
e-mail: jan.cvejn@upce.cz

R. Silhavy et al. (eds.), *Automation Control Theory Perspectives in Intelligent Systems*, Advances in Intelligent Systems and Computing 466,
DOI 10.1007/978-3-319-33389-2_2

exhibit scan variance. A similar situation could occur at remote control, where the measurements and the control signal are transported over a communication network.

Irregularities of the scan period cause worse performance of the control loop in comparison to the theoretical case. This influence can be neglected if the irregularities occur rarely and the system time constants are large in comparison to the scan period. In the other cases the influence on the closed-loop dynamics can be significant.

In this paper, we describe a modification of the classical linear-quadratic (LQ) discrete control algorithm taking into account irregularities of the scan period. We show that the effect of the scan variance can be partially compensated by mathematical means if a hybrid control law is used, working at discrete steps, but using a continuous-time model for the determination of the control output. In this way the control reliability and performance can be enhanced, especially at time-critical applications.

This problem has been studied already in [1] in a more general form as a stochastic control problem, considering also the optimal estimation. In this paper only the controller part is discussed, but an extended model of the scan period disturbances, which better corresponds to some real situations, is considered. This modification requires a corresponding extension of the control algorithm. The determination of the control action at each step is still not a time-consuming operation, although the enhanced control algorithm needs a table of data stored in the controller memory, which is initialized at the design phase.

2 Motivation

Consider a continuous time-invariant linear system

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

where the dimensions of $\mathbf{x}$ and $\mathbf{u}$ are n and m, respectively, $n \geq m$. $\mathbf{A}$ and $\mathbf{B}$ are known matrices of corresponding dimensions. We are looking for a control history such that

$$J = \frac{1}{2}\int_0^\infty \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)\, dt \;\rightarrow\; \min \tag{2}$$

where $\mathbf{Q}$, $\mathbf{R}$ are given symmetric positive definite matrices. We assume that the current state $\mathbf{x}(t)$, or its estimate, is known.

Although the system nature is continuous, we consider that the measurements and the control actions are taken at discrete-time steps $t_0 < t_1 < \cdots$. Although the scan period T is assumed to be known and constant, due to external factors the

actual difference $t_{i+1} - t_i$ can fluctuate. We assume that each scan is provided with its time mark, which can be usually easily technically realized. Then, from the known sequence $t_1, t_2, \ldots, t_k$ of the previous scan instants it is possible to estimate the future sequence $\bar{t}_{k+1}, \bar{t}_{k+2}, \bar{t}_{k+3}, \ldots$, which is considered to be equidistant, i.e.

$$\bar{t}_{k+2} - \bar{t}_{k+1} = \bar{t}_{k+3} - \bar{t}_{k+2} = \cdots = \overline{T}. \tag{3}$$

But note that $\overline{T} = T$ need not hold in general, because the response can be delayed permanently in some time interval. Although the scan interrupts are generated by hardware clock with the period T, which does not depend on the previous scan instants, the delay can be caused by omitting some scan instants, e.g. due to service of hardware events in the operating system (this problem is discussed below in more detail). This delay can be detected from the sequence of the past measurements t_i, $i \leq k$ and this information also can be used to estimate the future sequence (3), i.e. the expected period $\overline{T}$.

There are several possible ways how to estimate the next scan instant $\bar{t}_{k+1}$. In [1] a simplified model was used, which assumed $\overline{T} = T$. If we denote t_c the instant of the next time interrupt closest to t_k, in most cases the following estimate seems to be more adequate:

$$\bar{t}_{k+1} = t_c + \overline{T} - T. \tag{4}$$

But if $t_c - t_k \leq \alpha T$, where $\alpha \in (0, 1)$ is a known parameter, the control algorithm should be designed to omit the next scan, i.e. in this case

$$\bar{t}_{k+1} = \left(t_c + \overline{T} - T\right) + T = t_c + \overline{T}. \tag{5}$$

This behavior indeed depends on implementation of the controller on given platform. For instance, $t_c - t_k \leq 0.25T$ may indicate that the processor looses the ability to process the hardware events at given moment and in such a case the control algorithm should be designed to drop the following scan and wait until the next one to prevent the system from overloading, which would affect the overall functionality. This modification is especially needed in the case of multi-tasking operating systems, where the control algorithm is a high-priority task, and if processing of the measurements and computation of the control action is a time-consuming operation within the interval $[t_k, t_{k+1}]$.

3 Optimal Control Algorithm

To summarize the considerations of the previous section, at given moment t_k we assume that the estimates of the next scan instant $\bar{t}_{k+1} \geq t_c$ and the future scan period $\overline{T} \geq T$ are known, in general different from the next hardware clock instant t_c and the clock period T.

The criterion (2) value from $t = t_k$ can be expressed as

$$J(t_k) = \frac{1}{2}\sum_{i=k}^{N} \int_{t_i}^{t_{i+1}} \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)\, dt \tag{6}$$

where $N \to \infty$ and $\mathbf{u}(t)$ is constant in each interval $[t_i,\, t_{i+1})$, $i \geq k$. Note that unlike common practice the criterion includes the information about complete state history in $[0,\, t_f]$, and not only about the values at the discrete points t_i.

Denote for simplicity $\mathbf{x}_k = \mathbf{x}(t_k)$ and $\mathbf{u}_k = \mathbf{u}(t_k)$. For given $\mathbf{x}_k$ and $t > t_k$

$$\mathbf{x}(t) = \mathbf{\Phi}(t - t_k)\mathbf{x}_k + \mathbf{\Psi}(t - t_k)\mathbf{u}_k \tag{7}$$

holds, where

$$\Phi(h) = e^{\mathbf{A}h},\ \mathbf{\Psi}(h) = \int_0^h \mathbf{\Phi}(h - \tau)\,\mathbf{B}\, d\tau\ = \int_0^h \mathbf{\Phi}(\tau)\, d\tau\, \mathbf{B}. \tag{8}$$

The term $\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)$ for given $\mathbf{x}_k$ and $t > t_k$ can be written using (8) as

$$\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) = \left[\mathbf{x}_k^T, \mathbf{u}_k^T\right] \begin{bmatrix} \mathbf{\Phi}^T(t - t_k) \\ \mathbf{\Psi}^T(t - t_k) \end{bmatrix} \mathbf{Q} [\mathbf{\Phi}(t - t_k), \mathbf{\Psi}(t - t_k)] \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}. \tag{9}$$

Let us define

$$\begin{aligned} \mathbf{U}(h) &= \int_0^h \left(\begin{bmatrix} \mathbf{\Phi}^T(\tau) \\ \mathbf{\Psi}^T(\tau) \end{bmatrix} \mathbf{Q} [\mathbf{\Phi}(\tau), \mathbf{\Psi}(\tau)] + \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{R} \end{bmatrix} \right) dh \\ &= \int_0^h \begin{bmatrix} \mathbf{\Phi}^T(\tau)\mathbf{Q}\mathbf{\Phi}(\tau) & \mathbf{\Phi}^T(\tau)\mathbf{Q}\Psi(\tau) \\ \mathbf{\Psi}^T(\tau)\mathbf{Q}\mathbf{\Phi}(\tau) & \mathbf{\Psi}^T(\tau)\mathbf{Q}\mathbf{\Psi}(\tau) + \mathbf{R} \end{bmatrix} d\tau = \begin{bmatrix} \mathbf{U}_{11}(h) & \mathbf{U}_{12}(h) \\ \mathbf{U}_{21}(h) & \mathbf{U}_{22}(h) \end{bmatrix} \end{aligned} \tag{10}$$

where $\mathbf{U}_{21}(h) = \mathbf{U}_{12}^T(h)$. Using (10) we can write

$$J(t_k) = \frac{1}{2}\left[\mathbf{x}_k^T\ \mathbf{u}_k^T\right] \mathbf{U}(\bar{t}_{k+1} - t_k) \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} + J(\bar{t}_{k+1}) \tag{11}$$

where

$$J(\bar{t}_{k+1}) = \frac{1}{2}\sum_{i=k+1}^{N} \left[\mathbf{x}_i^T\ \mathbf{u}_i^T\right] \mathbf{U}(\overline{T}) \begin{bmatrix} \mathbf{x}_i \\ \mathbf{u}_i \end{bmatrix}. \tag{12}$$

Let us denote $J^*(t_k)$ the minimal value of $J(t_k)$. By application of Bellman's optimality principle [2–3] we obtain

$$J^*(t_k) = \min_{u_k}\left\{\frac{1}{2}\left[\mathbf{x}_k^T\ \mathbf{u}_k^T\right]\mathbf{U}(\bar{t}_{k+1}-t_k)\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix} + J^*(\bar{t}_{k+1})\right\} \tag{13}$$

where

$$J^*(\bar{t}_{k+1}) = \min_{\{u_{k+1},\ldots,u_N\}}\left\{\frac{1}{2}\sum_{i=k+1}^{N}\left[\mathbf{x}_i^T\ \mathbf{u}_i^T\right]\mathbf{U}(\overline{T})\begin{bmatrix}\mathbf{x}_i\\ \mathbf{u}_i\end{bmatrix}\right\} \tag{14}$$

subject to the dynamic constraints

$$\mathbf{x}_{i+1} = \boldsymbol{\Phi}(\overline{T})\,\mathbf{x}_i + \boldsymbol{\Psi}(\overline{T})\mathbf{u}_i,\quad i \geq k+1. \tag{15}$$

Equations (14) and (15) formulate a discrete deterministic linear-quadratic optimal control problem. The minimal cost-function value of this problem for $N \to \infty$ is in the form

$$\begin{aligned} J^*(\bar{t}_{k+1}) &= \frac{1}{2}\mathbf{x}_{k+1}^T\mathbf{S}\,\mathbf{x}_{k+1} \\ &= \frac{1}{2}\left[\mathbf{x}_k^T,\ \mathbf{u}_k^T\right]\begin{bmatrix}\boldsymbol{\Phi}^T(\bar{t}_{k+1}-t_k)\\ \boldsymbol{\Psi}^T(\bar{t}_{k+1}-t_k)\end{bmatrix}\mathbf{S}[\boldsymbol{\Phi}(\bar{t}_{k+1}-t_k),\ \boldsymbol{\Psi}(\bar{t}_{k+1}-t_k)]\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix} \end{aligned} \tag{16}$$

where $\mathbf{S}$ is a positive-definite symmetric matrix. If we define

$$\mathbf{Z}(h) = \begin{bmatrix}\mathbf{Z}_{11}(h) & \mathbf{Z}_{12}(h)\\ \mathbf{Z}_{21}(h) & \mathbf{Z}_{22}(h)\end{bmatrix} = \mathbf{U}(h) + \begin{bmatrix}\boldsymbol{\Phi}^T(h)\\ \boldsymbol{\Psi}^T(h)\end{bmatrix}\mathbf{S}[\boldsymbol{\Phi}(h),\ \boldsymbol{\Psi}(h)]. \tag{17}$$

where $\mathbf{Z}_{21}(h) = \mathbf{Z}_{12}^T(h)$, the minimizer of $J(t_k)$ for fixed $\overline{T}$ can be written as

$$\begin{aligned} \mathbf{u}_k^* &= \arg\min_{\mathbf{u}_k}\left\{\frac{1}{2}\left[\mathbf{x}_k^T\ \mathbf{u}_k^T\right]\mathbf{U}(\bar{t}_{k+1}-t_k)\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix} + \frac{1}{2}\mathbf{x}_{k+1}^T\mathbf{S}\,\mathbf{x}_{k+1}\right\} \\ &= \arg\min_{\mathbf{u}_k}\left\{\frac{1}{2}\left[\mathbf{x}_k^T\ \mathbf{u}_k^T\right]\mathbf{Z}(\bar{t}_{k+1}-t_k)\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix}\right\}. \end{aligned} \tag{18}$$

The solution can be easily found by differentiation in the form

$$\mathbf{u}_k^* = -\mathbf{Z}_{22}^{-1}(\bar{t}_{k+1}-t_k)\mathbf{Z}_{21}(\bar{t}_{k+1}-t_k)\mathbf{x}_k = \mathbf{C}(\bar{t}_{k+1}-t_k)\mathbf{x}_k. \tag{19}$$

Note that the matrix inversion in (19) always exists, since $\mathbf{Z}_{22}(h)$ is positive definite.

If we put $\bar{t}_{k+1} - t_k = \overline{T}$ in (13) and (19), we have to obtain for optimal $\mathbf{u}_k$

$$J^*(t_k) = \frac{1}{2}\mathbf{x}_k^T \mathbf{S}\, \mathbf{x}_k. \tag{20}$$

This condition can be used to determine $\mathbf{S}$. By substituting (20) into (13) and (16) we obtain:

$$\begin{aligned} J^*(t_k) &= \frac{1}{2}\left[\mathbf{x}_k^T\ \mathbf{u}_k^T\right]\mathbf{Z}\left(\overline{T}\right)\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_k\end{bmatrix} \\ &= \frac{1}{2}\mathbf{x}_k^T\begin{bmatrix}\mathbf{I} & -\mathbf{Z}_{22}^{-1}\left(\overline{T}\right)^{-1}\mathbf{Z}_{21}\left(\overline{T}\right)\end{bmatrix}\mathbf{Z}\left(\overline{T}\right)\begin{bmatrix}\mathbf{I}\\ -\mathbf{Z}_{22}^{-1}\left(\overline{T}\right)^{-1}\mathbf{Z}_{21}\left(\overline{T}\right)\end{bmatrix}\mathbf{x}_k \\ &= \frac{1}{2}\mathbf{x}_k^T\begin{bmatrix}\mathbf{I} & -\mathbf{Z}_{22}^{-1}\left(\overline{T}\right)^{-1}\mathbf{Z}_{21}\left(\overline{T}\right)\end{bmatrix}\begin{bmatrix}\mathbf{Z}_{11}\left(\overline{T}\right) - \mathbf{Z}_{12}\left(\overline{T}\right)\mathbf{Z}_{22}^{-1}\left(\overline{T}\right)\mathbf{Z}_{21}\left(\overline{T}\right)\\ \mathbf{0}\end{bmatrix}\mathbf{x}_k \\ &= \frac{1}{2}\mathbf{x}_k^T\left(\mathbf{Z}_{11}\left(\overline{T}\right) - \mathbf{Z}_{12}\left(\overline{T}\right)\mathbf{Z}_{22}^{-1}\left(\overline{T}\right)\mathbf{Z}_{21}\left(\overline{T}\right)\right)\mathbf{x}_k. \end{aligned} \tag{21}$$

This shows that

$$\mathbf{S} = \mathbf{Z}_{11}\left(\overline{T}\right) - \mathbf{Z}_{12}\left(\overline{T}\right)\mathbf{Z}_{22}^{-1}\left(\overline{T}\right)\mathbf{Z}_{21}\left(\overline{T}\right) \tag{22}$$

must hold. By substituting for $\mathbf{Z}_{ij}\left(\overline{T}\right)$ from (17) it is easily seen that (22) can be rewritten into an algebraic matrix Riccati equation [4, 3].

4 Implementation of the Controller

Obtained expressions are indeed rather complicated to be computed at each control step. The controller matrix $\mathbf{C}(\bar{t}_{k+1} - t_k)$ in (20) is dependent on the expected distance of the next scan $\bar{t}_{k+1} - t_k$ and on $\mathbf{S}$, while $\mathbf{S}$ depends on $\overline{T}$.

For given $\overline{T}$ the solution to the Riccati Eq. (22) can be obtained off-line as a part of the controller design. Denote $\mathbf{Z}\left(\overline{T};\mathbf{S}_i\right)$ the value of $\mathbf{Z}\left(\overline{T}\right)$ for $\mathbf{S} = \mathbf{S}_i$. A basic method of obtaining $\mathbf{S}$ consists in solving

$$\mathbf{S}_{i+1} = \mathbf{Z}_{11}\left(\overline{T};\mathbf{S}_i\right) - \mathbf{Z}_{12}\left(\overline{T};\mathbf{S}_i\right)\mathbf{Z}_{22}^{-1}\left(\overline{T};\mathbf{S}_i\right)\mathbf{Z}_{21}\left(\overline{T};\mathbf{S}_i\right) \tag{23}$$

iteratively until $\|\mathbf{S}_{i+1} - \mathbf{S}_i\| < \varepsilon$, where ε is sufficiently small [4]. More sophisticated methods, preferable both from numerical point of view and for improved efficiency, were proposed in [5, 6].

If we assume $\overline{T} \in [T, T_{\max}]$, $T_{\max} \geq 2T$, the solutions to (23) can be obtained for the values in this interval with a sufficiently small discrete step $\Delta\overline{T}$ at the

initialization phase. These solutions can be stored in the controller memory and in the real-time operation the values corresponding to the current estimate of $\overline{T}$ are being picked from this table. Although it may seem that the initialization phase could be computationally very demanding, if the value of $\mathbf{S}$ is known of some $\overline{T}$, it can be used as a very good estimate of $\mathbf{S}$ for the iterative computation based on (23) corresponding to $\overline{T} + \Delta\overline{\mathrm{T}}$, because $\mathbf{S}$ depends only moderately on $\overline{T}$. Therefore, the computation of the whole table of the values of $\mathbf{S}$ for $\overline{T} \in [T, T_{\max}]$ is not a time-demanding operation.

In the same way, the values of the matrices $\mathbf{\Phi}(h)$, $\mathbf{\Psi}(h)$ and $\mathbf{U}(h)$ have to be computed for $0 < h \leq h_{\max}$, where $h_{\max} \geq 2T$ is known, with a sufficiently small discrete step of h and stored in the controller memory. Formally written, it is needed to solve the following set of differential equations in the interval $h \in [0, h_{\max}]$:

$$\frac{d}{dh}\mathbf{\Phi}(h) = \mathbf{A}\,\mathbf{\Phi}(h) \tag{24}$$

$$\frac{d}{dh}\mathbf{\Psi}(h) = \mathbf{\Phi}(h)\,\mathbf{B} \tag{25}$$

$$\frac{d}{dh}\mathbf{U}(h) = \begin{bmatrix} \mathbf{\Phi}^T(\tau)\mathbf{Q}\mathbf{\Phi}(\tau) & \mathbf{\Phi}^T(\tau)\mathbf{Q}\mathbf{\Psi}(\tau) \\ \mathbf{\Psi}^T(\tau)\mathbf{Q}\mathbf{\Phi}(\tau) & \mathbf{\Psi}^T(\tau)\mathbf{Q}\mathbf{\Psi}(\tau) + \mathbf{R} \end{bmatrix} \tag{26}$$

with the initial conditions

$$\mathbf{\Phi}(0) = \mathbf{I}_n, \quad \mathbf{\Psi}(0) = \mathbf{0}, \quad \mathbf{U}(0) = \mathbf{0}. \tag{27}$$

It is important to mention here that the controller implementation is significantly more efficient in the simplified version where $\overline{T} = T$ is fixed. In this case $\mathbf{S}$ is constant and the controller matrix $\mathbf{C}(h)$ in (19) can be computed in forward and stored, so the matrix inversion in (19) need not be computed in real time.

5 Example

Consider the double-integrator system in the form (1) where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{28}$$

with the initial condition

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{29}$$

The criterion (6) parameters were chosen as

$$\mathbf{Q} = \mathbf{I}_2, \ \mathbf{R} = 1. \tag{30}$$

The scan period is $T = 2\,s$, but each fourth scan is delayed of 50 % and the following scan is omitted. In the initialization phase it was needed to obtain the solution to (24)–(26) for $h \in [0, 2T]$ and to solve (23) for the sequence of values of $\overline{T} \in [T, 2T]$ with the step size of $\Delta\overline{T} = 0.01$. The initialization was not a time-demanding operation.

Figure 1 shows the response of the system when the standard LQ controller is used, i.e. if $\bar{t}_{k+1} = t_k + T$ and $\overline{T} = T$, while Fig. 2 shows the responses if the modified controller (19) is used for $\overline{T} = 5T/4$. This estimate of $\overline{T}$ corresponds to the fact that each fifth scan is omitted.

It can be seen that the control loop behavior was significantly enhanced, although a similar effect indeed could be achieved by decreasing the scan period, if

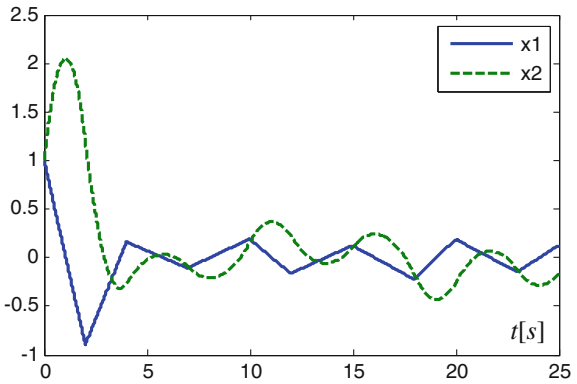

Fig. 1 The history of the state variables—standard LQ controller

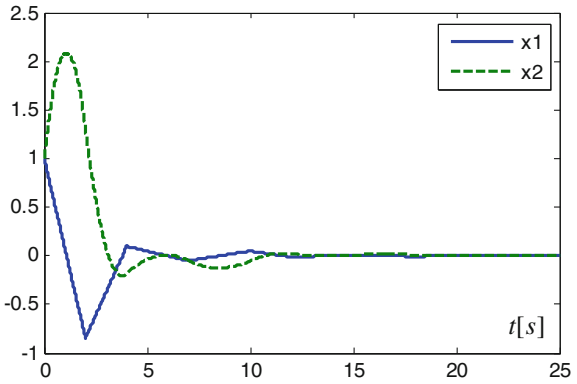

Fig. 2 The history of the state variables—the modified LQ controller, full version

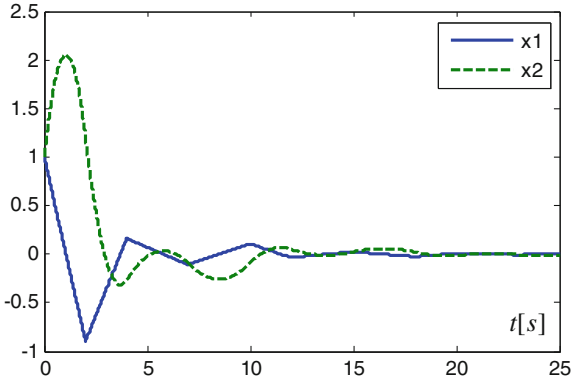

Fig. 3 The history of the state variables—the modified LQ controller, simplified version

it was technically possible. Even bigger differences can be observed if the controller is equipped with the optimal state estimator, which can be also designed so that it takes the variance of the scan period into account, as described in [1].

Figure 3 shows that the responses obtained for the simplified version where $\overline{T} = T$ are similar to the full version. This indicates that the simplified version, which is preferable from the implementation point of view, would be usually sufficient and recommendable for practical use.

6 Conclusions

The modification of the LQ control algorithm described in this paper tries to reduce the influence of the scan-period variance, which can occur in computer-based control systems, on the closed-loop control performance. Although obtained expressions for the control output may be rather complicated to be computed in real time, if sufficient memory in the control system is available, it is possible to carry out most of these computations in forward and the determination of the control output is not a complicated or time-consuming operation. Consequently, such a control algorithm can be implemented even on low-performance hardware platforms. In the simplified version, which seems to be sufficient for practical purposes, the computation of the control action is as demanding as a matrix-vector product, like in the case of standard LQ control algorithm. However, the controller has to be equipped with sufficient amount of memory to store a table of the control matrices, dependent on a single parameter.

References

1. Cvejn, J.: Compensation of the scan-period irregularities in lqg control systems. Arch. Control Sci. **18**, 3 (2008)
2. Bryson, A.E., Ho, Y.C.: Applied Optimal Control. Hemisphere Corp, New York (1975)
3. Fleming, W.H., Rishel, R.W.: Deterministic and Stochastic Optimal Control. Springer, Berlin (1995)
4. Stengel, R.F.: Stochastic Optimal Control: Theory and Application. Wiley, New York (1986)
5. Pappas, T., Laub, A.J., Sandell Jr, N.R.: On the numerical solution of the discrete-time algebraic riccati equation. IEEE Trans. Autom. Control **25**, 631–641 (1980)
6. Kleinman, D.L.: Stabilizing a discrete, constant linear system with application to iterative methods for solving the riccati equation. IEEE Trans. Autom. Control **19**, 252–254 (1974)

Polynomial Approximation of Quasipolynomials Based on Digital Filter Design Principles

Libor Pekař and Pavel Navrátil

Abstract This contribution is aimed at a possible procedure approximating quasipolynomials by polynomials. Quasipolynomials appear in linear time-delay systems description as a natural consequence of the use of the Laplace transform. Due to their infinite root spectra, control system analysis and synthesis based on such quasipolynomial models are usually mathematically heavy. In the light of this fact, there is a natural research endeavor to design a sufficiently accurate yet simple engineeringly acceptable method that approximates them by polynomials preserving basic spectral information. In this paper, such a procedure is presented based on some ideas of discrete-time (digital) filters designing without excessive math. Namely, the particular quasipolynomial is subjected to iterative discretization by means of the bilinear transformation first; consequently, linear and quadratic interpolations are applied to obtain integer powers of the approximating polynomial. Since dominant roots play a decisive role in the spectrum, interpolations are made in their very neighborhood. A simulation example proofs the algorithm efficiency.

Keywords Approximation · Bilinear transformation · Digital filter · MATLAB · Polynomials · Pre-warping · Quasipolynomials

1 Introduction

Mainly due to that delay appears in many real-word systems, such as economical, biological, networked, mechanical, electrical etc. [1, 2], this phenomenon have been intensively studied during recent decades [3, 4]. The most specific feature of time

L. Pekař (✉) · P. Navrátil
Faculty of Applied Informatics, Tomas Bata University in Zlín,
Zlín, Czech Republic
e-mail: pekar@fai.utb.cz

P. Navrátil
e-mail: pnavratil@fai.utb.cz

R. Silhavy et al. (eds.), *Automation Control Theory Perspectives in Intelligent Systems*, Advances in Intelligent Systems and Computing 466,
DOI 10.1007/978-3-319-33389-2_3

delay systems (TDS) and models can be viewed in the fact that they own infinite spectra; thus, they are included in the family of infinite-dimensional systems. Linear time-invariant TDS, considered in this paper, can primarily be described by ordinary difference-differential (or shifted-argument) equations [5] which can be subjected to the Laplace transform [6]. As a consequence, the corresponding transfer function (matrix) is obtained in which its denominator quasipolynomial (QP) called also as the characteristic QP dominantly decides about dynamical and stability system properties [7]. QP roots (or zeros) agree with system poles, except for some special cases. Hence, the knowledge about the spectrum of QP roots is a crucial matter for TDS analysis.

Various tools and methods have been developed and designed for TDS spectrum computation or estimation, mainly in the state space domain, see e.g. [8]. These results represent a certain kind of TDS discretization as well. In the input-output Laplace space, the Quasi-Polynomial mapping Rootfinder (QPmR) has proved to be a very effective and practically usable when computing QP zeros within the determined region of the complex plane [9–11]. This method omits any QP simplification or approximation and a special software package was developed for its practical usability. TDS model reduction or rationalization represents another way how to cope with the problem [12–14]. However, expect for the well-known Padé approximation of exponential terms, these methods have been designed primarily for approximation of the complete TDS model, not only of the QP itself, and their applicability is mostly worsen due to a high mathematical knowledge level required out of the user. Delta models [15] represent an easy-to-handle rationalization and discretization methodology for practitioners, usable for TDS as well [16], which are closely related to the notion of the bilinear transformation [17] and give a polynomial discrete-time approximation representation of the system model.

The goal of this paper is to design a sufficiently simple, fast and practically usable technique for polynomial approximation of a QP without the necessity of advanced mathematical knowledge or using uncommon software tools. It is based on two main principles adopted from digital filter designing: As first, exponential terms in the QP are subjected to the innate time shifting and consequently to linear or quadratic interpolation such that the eventual shifts are integer multiplies of a basic time period. As second, s-powers representing derivatives are recursively put through the bilinear transformation the efficiency of which is further enhanced by pre-warping [17] that preserves the particular selected frequency under discretization. Since the dominant (i.e. the rightmost) pole (or the pair) has the decisive impact to system dynamics, all the interpolations and extrapolations are performed in the neighborhood of a close dominant QP root estimation.

The rest of the paper is organized as follows: Basic properties of zeros of a retarded QP and herein utilized techniques and tools are provided in the next, preliminary, section. Afterward, the reader is acquainted with the approximating procedure in details. A numerical example is given to illustrate the accuracy and efficiency of the technique; then the paper is concluded.

2 Preliminaries

Prior to the description of the approximation algorithm, retarded quasipolynomials and their spectral features ought to be introduced. This section, moreover, provides the reader with necessary mathematical tool and techniques that are utilized during the approximation procedure.

2.1 Retarded Quasipolynomial and Its Spectrum

A retarded QP can be expressed as

$$X(s,\mathbf{x},\boldsymbol{\tau}) = s^n + \sum_{i=0}^{n-1}\sum_{j=1}^{h_i} x_{ij}s^i \exp\left(-s\sum_{k=1}^{L}\lambda_{ij,k}\tau_k\right) \tag{1}$$

where $\boldsymbol{\tau} = [\tau_1,..\tau_L] \in \mathbb{R}_+^L, \tau_i > 0$ represents independent delays, $\lambda_{ij,k} \in \mathbb{N}_0$, $\mathbf{x} = [x_{01},..x_{n-1,h_{n-1}}] \in \mathbb{R}^{\sum_{i=0}^{n-1} h_i} \neq 0$.

Let $\Sigma := \{s_i\}$ be the spectrum of roots (zeros) of (1).

Property 1 [2, 5]. *For (1) it holds that*

1. *If there exist nonzero $x_{ij}, \lambda_{ij,k}$ for some positive τ_k and some i, j, k, then the number of QP zeros is infinite.*
2. *For any fixed real $\beta > -\infty$, the number of roots with $\mathrm{Re}\, s_i > \beta$ is finite.*
3. *Isolated roots behave continuously and smoothly with respect to $\boldsymbol{\tau}$ on $\mathbb{C}$.*

Definition 1 The *spectral abscissa*, $\alpha(\boldsymbol{\tau})$, is the function

$$\alpha(\boldsymbol{\tau}) := \boldsymbol{\tau} \mapsto \sup \mathrm{Re}\Sigma \tag{2}$$

Property 2 [18]. *For function $\alpha(\boldsymbol{\tau})$, it holds that:*

1. *It may be nonsmooth, and hence not differentiable, e.g. in points with more than one real root or conjugate pairs with the same maximum real part.*
2. *It is non-Lipschitz, for instance, at points where the maximum real part has multiplicity greater than one.*

To sum up main findings from Properties 1 and 2, although the rightmost part of the spectrum contains isolated roots, the position of which is continuously changed with $\boldsymbol{\tau}$, the abscissa might evince abrupt changes in its value.

Definition 2 The *leading* (dominant) *root*, s_L, or pair $\{s_L, \bar{s}_L\}$ satisfies

$$\alpha(\boldsymbol{\tau}) = \operatorname{Re} s_L = 0 \tag{3}$$

i.e. it represents the rightmost root or the pair from Σ.

2.2 Discretization Techniques and Tools

As mentioned above, the approximation algorithm is designed to be practically implementable by means of the most standard programs without the necessity of the use of special software and the knowledge of advanced math.

Derivatives in (1) are expressed by s-powers. The idea of the derivative approximation is based on the iterative digital-filter-like discretization of the QP depending on the current leading root estimation via the bilinear (or Tustin) transformation

$$s \to \frac{2}{T}\frac{1-q}{1+q} \tag{4}$$

where q means the shifting operator that agrees with z^{-1} in the z-transform and T is the sampling period.

Let $X(s, \mathbf{x}, \boldsymbol{\tau})$ be the characteristic quasipolynomial of a system. Since, however, transformation (4) does not preserve frequencies (namely, the system eigenfrequency), it is desirable to find another mapping that keeps "continuous" frequencies (i.e. those in the s-plane) and "discrete" frequencies (i.e. those in the z-plane) identical. It can be derived [17] that this requirement is satisfied if the following modified mapping is used

$$s \to \frac{\omega}{\tan(\omega T/2)}\frac{1-q}{1+q} \tag{5}$$

where ω stands for the desired frequency. Note that the operation of the frequency preservation is called *pre-warping*. Because of the decisive role of leading roots, we have set $\omega = \operatorname{Im} s_L$.

From the point of view of derivative discretization or delta models, the value of T in (4) or (5) should be sufficiently small; however, the lower T is, the higher resulting approximating polynomial degree is obtained. In the contrary, the z-transform demands significantly lower values; for instance, in [19] the following recommendation for periodic systems is given

$$T = [0.2/\omega_0, 0.5/\omega_0] \tag{6}$$

where ω_0 expresses the frequency of undumped oscillations. Note that $\omega_0 \approx |s_L|$.

Regarding terms expressing delays in (1), i.e. the exponentials, they can be subjected to inherent shifting

$$\exp(-\vartheta s)X(s) \hat{=} x(t-\vartheta) \hat{=} q^{\vartheta/T}x(k) \hat{=} z^{-\vartheta/T}X(z) \tag{7}$$

Nevertheless, in general, delay value ϑ might not be an integer multiple of T; hence, term $z^{-\vartheta/T}$ should be interpolated by a linear combination of integer powers of z. The following lemma gives two possible solutions of this task.

Lemma 1 *Consider a term* $z^{-\left(\lfloor\vartheta\rfloor+\bar{\vartheta}\right)}$, $\lfloor\vartheta\rfloor \in \mathbb{N}_0$, $0<\bar{\vartheta}<1$. *In the vicinity of* $z_0 \in \mathbb{C}$, *the term can be interpolated linearly as (8) or quadratically as (9):*

$$(1-\lfloor\vartheta\rfloor)z_0^{-\lfloor\vartheta\rfloor}z^{-\bar{\vartheta}} + \lfloor\vartheta\rfloor z_0^{-\lfloor\vartheta\rfloor+1}z^{-\left(\bar{\vartheta}+1\right)} \tag{8}$$

$$\begin{aligned} &0.5(2-\lfloor\vartheta\rfloor)(1-\lfloor\vartheta\rfloor)z_0^{-\lfloor\vartheta\rfloor}z^{-\bar{\vartheta}} + \lfloor\vartheta\rfloor(2-\lfloor\vartheta\rfloor)z_0^{-\lfloor\vartheta\rfloor+1}z^{-\left(\bar{\vartheta}+1\right)} \\ &+ 0.5\lfloor\vartheta\rfloor(\lfloor\vartheta\rfloor-1)z_0^{-\lfloor\vartheta\rfloor+2}z^{-\left(\bar{\vartheta}+2\right)} \end{aligned} \tag{9}$$

The proof is omitted due to the limited space. In this study, the value of z_0 is selected as

$$z_0 = z_L = \exp(Ts_L) \tag{10}$$

which agrees with the z-transform and is closely related to mapping (5), and it is consistent with the idea of the leading root importance.

3 Polynomial Approximation Algorithm

Two versions of the algorithm are presented. The former one can be utilized whenever the leading root of a QP sufficiently close to the studied QP. The latter can be used even if no leading root estimation is known and, naturally, it is expected to give less accurate results and is computationally more complex.

Algorithm 1 Input: The QP $X(s, \mathbf{x}, \boldsymbol{\tau})$ to be approximated.

Step 1: Consider that there exists a QP $X(s, \mathbf{x}_0, \boldsymbol{\tau}_0)$ with $\|X(s, \mathbf{x}, \boldsymbol{\tau}) - X(s, \mathbf{x}_0, \boldsymbol{\tau}_0)\| < \Delta$, for a sufficiently small $\Delta > 0$, the leading root of which, $s_0 = \hat{s}_0$, is known exactly. Set $\varepsilon > 0$.
Step 2: Compute polynomial $P(z^{-1}|\hat{s}_0)$ according to (4)–(9). Define and compute

$$\hat{s}_1 := \left\{\arg\min|s-\hat{s}_0| : s = T^{-1}\log(z), P\left(z^{-1}|\hat{s}_0\right) = 0\right\} \tag{11}$$

Step 3: While $|\hat{s}_1 - \hat{s}_0| \geq \varepsilon$, set $\hat{s}_0 := \hat{s}_1$ and go to Step 5.

Output: $P(z^{-1})$ and its roots.

Remark 1 The norm in Step 1 of Algorithm 1 can simply be computed as a point norm in s_0, i.e. $\|X(s_0, \mathbf{x}, \boldsymbol{\tau})\|$. The problem may appear if $|s_1 - s_0| > \delta$ for some $\delta > 0$ and any value of $\|X(s_0, \mathbf{x}, \boldsymbol{\tau})\|$ due to Property 2.

Algorithm 2 Input: The QP $X(s, \mathbf{x}, \boldsymbol{\tau})$ to be approximated.

Step 1: Define the mesh grid $\tau_{k,j+1} = \tau_{k,j} + \Delta\tau_{k,j}$, $\tau_{k,0} = 0$, $\boldsymbol{\tau} = \left[\tau_{1,N}, \tau_{2N}, \ldots, \tau_{L,N}\right]$, $k = 1 \ldots L$, $j = 0 \ldots N-1$, and set $\varepsilon > 0$.
Step 2: Compute

$$\hat{s}_{0,\ldots,0} = s_{0,\ldots,0} := \{\arg\ \max \mathrm{Re}\, s : X(s, \mathbf{x}, \mathbf{0}) = 0\} \tag{12}$$

exactly.
Step 3: For ($j_1 = 0 \ldots N-1$, for ($j_2 = 0 \ldots N-1$,… (for $j_L = 0 \ldots N-1$ do: If $\exists j_l \neq 0, l = 1 \ldots L$ do Steps 4–6))) (nested loops).
Step 4: Define $M := \max\{k : j_k \neq 0\}$ and set $\bar{\tau} = \left[\tau_{1,j_1}, \tau_{2,j_2} \ldots, \tau_{L,j_L}\right]$, $\hat{s}_0 = \hat{s}_{j_1,\ldots,j_{M-1},j_M-1,0\ldots0}$.
Step 5: Compute polynomial $P(z^{-1}|\bar{\tau}, \hat{s}_0)$ according to (4)–(9) and $\hat{s}_1$ by means of (11).
Step 6: While $|\hat{s}_1 - \hat{s}_0| \geq \varepsilon$, set $\hat{s}_0 := \hat{s}_1$ and go to Step 5.

Output: $P(z^{-1})$ and its roots.

4 Numerical Example

Consider a skater on the remotely swaying bow sketched in Fig. 1. The skater controls the power input, $P(t)$, to the servo giving rise to the angle deviation, $u(t)$, from the horizontal position and, consequently, the angle between the skater and the bow symmetry axis, $y(t)$, emerges. If the friction is neglected but the skater's reaction time and servo latency included, the following particular transfer function can be written [20]

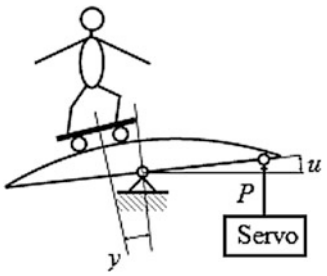

Fig. 1 A skater on the remotely controlled swaying bow

$$G(s) = \frac{0.2\exp(-(\tau_1+\tau_2)s)}{s^2(s^2-\exp(-\tau_2 s))} \tag{13}$$

A particular generalized (third-order) finite-dimensional linear proportional-integrative-derivative controller stabilizes system (13) with $\boldsymbol{\tau} = [\tau_1, \tau_2] = [0.08, 0.08]$, which yields the feedback characteristic quasipolynomial (14), see [21].

$$\begin{aligned} X(s,\cdot,\boldsymbol{\tau}) = {} & s^2\left(s^2-\exp(-0.08s)\right)\left(s^3+469418.6s^2+640264.6s+10560107\right) \\ & + 0.2\exp(-0.16s)\left(82226506s^3+106523134s^2+26247749s+5617613\right) \end{aligned} \tag{14}$$

the roots of which decide about control system stability.

Let us test Algorithm 1 for various values of T first. With respect to the upper bound of condition, the sampling period can be written as $T = (k_T|s_L|)^{-1}, k_T \geq 2$, the value of which is updated in every iteration step (see Steps 2 and 3 in Algorithm 1). Assume that the leading pair, $s_0 = -9.9669\text{e}-3 \pm 3.9672704\text{i}$, of $X(s, \mathbf{x}_0 = \mathbf{x}, [0.08, 0.07])$ is known exactly and set $\varepsilon = \text{e}-6$. Selected algorithm results are summarized in Table 1. Due to limited space, some observations are further commented rather than being included in the table.

Table 1 Results of Algorithm 1

Method	k_T	Dominant pair of roots		Num. of iter.
(4), (8)	2	–2.11963e–2 + 3.8592406i	–9.1702e–2 – 3.917348i	5
	5	–2.80361e–2 + 3.9172242i	–3.60554e–2 – 3.9223709i	4
	10	–2.90655e–2 + 3.9254363i	–3.30879e–2 – 3.9277296i	3
	20	–2.9325e–2 + 3.927487i	–3.05583e–2 – 3.9281274i	3
	30	–3.8027e–3 + 3.9947106i	–3.504e–3 – 3.9948482i	8
(5), (8)	2	–2.83932e–2 + 3.9282875i	–0.1062955 – 3.9922307i	5
	5	–2.92479e–2 + 3.9281888i	–3.78099e–2 – 3.9336952i	4
	10	–2.93707e–2 + 3.9281745i	–3.34359e–2 – 3.9304912i	3
	20	–2.94013e–2 + 3.92817i	–3.06333e–2 – 3.9288096i	3
	30	–3.77434e–3 + 3.9950239i	–3.47358e–3 – 3.9951627i	14
(4), (9)	2	–2.11963e–2 + 3.8592406i	–6.70344e–2 – 3.8332474i	5
	5	–2.80361e–2 + 3.9172242i	–2.96717e–2 – 3.9154286i	4
	10	–2.90655e–2 + 3.9254364i	–2.93613e–2 – 3.924999i	3
	20	–2.93249e–2 + 3.9274854i	–2.93571e–2 – 3.9274282i	3
	30	–1.29309e–2 + 3.9607762i	–1.29307e–2 – 3.960776i	13
(5), (9)	2	–2.83932e–2 + 3.9282875i	–7.80977e–2 – 3.8996772i	5
	5	–2.92479e–2 + 3.9281888i	–3.09832e–2 – 3.9262835i	4
	10	–2.93707e–2 + 3.9281746i	–2.96686e–2 – 3.9277336i	3
	20	–2.94014e–2 + 3.9281712i	–2.94335e–2 – 3.9281141i	3
	30	–1.29634e–3 + 3.9610884i	–1.29632e–3 – 3.961088i	14

Table 2 Values of dominant pairs for selected values of k_T

Method	k_T	Dominant pair of roots		Num. of iter.
(4), (8)	10	–2.90655e–2 + 3.9254363i	–3.30879e–2 – 3.9277296i	3
	20	–2.93249e–2 + 3.927486i	–3.05583e–2 – 3.9281274i	3.05
(5), (8)	10	–2.93707e–2 + 3.9281745i	–3.34359e–2 – 3.9304912i	3
	20	–2.94013e–2 + 3.9281699i	–3.06333e–2 – 3.9288096i	3.01
(4), (9)	10	–2.90655e–2 + 3.9254363i	–2.93613e–2 – 3.924999i	3
	20	–2.9325e–2 + 3.927487i	–2.93572e–2 – 3.9274298i	3
(5), (9)	10	–2.93707e–2 + 3.9281745i	–2.96686e–2 – 3.9277336i	3
	20	–2.94014e–2 + 3.9281706i	–2.94334e–2 – 3.9281135i	3.05

The exact leading pair of roots of $X(s, \mathbf{x}, \boldsymbol{\tau})$ found by the QPmR reads $s_L = -0.0294116 \pm 3.9281699\text{i}$. However, in fact, leading roots of the approximating polynomial do not constitute a conjugate pair because of its complex coefficients. We have observed from the test that higher values of k_T (limited by an upper bound of approx. $k_T \approx 30$) give less number of iterations within Steps 2 and 3, better approaching of leading roots and higher leading root estimation accuracy. The beneficial impact of pre-warping (5) can be seen in significantly better leading root estimation (mainly in the imaginary part of the root); however, such an improvement is not confirmed in the case of less dominant roots. The advantage of the quadratic interpolation (9) compared to the linear one (8) can also be observed. It distinctively improves less-dominant pairs mutual approaching and has a slight impact to their loci estimation and the approaching of the leading pair. In the contrary, it has no effect on the leading poles estimation.

However, the discretization procedure gives rise to "parasitic" high-frequency roots not included in the original QP. These polynomial roots are located very close to the imaginary axis with a high imaginary part value. Such an observation has been made by Vyhlídal and Zítek [16] as well; they have utilized a delta models in their work.

Algorithm 2 applied to (14) starts with $s_{0,\ldots,0} = 0.1214757 \pm 4.5573833\text{i}$, see (12), and let $\Delta\tau_{k,j} = 0.01$. Eventual values of dominant pairs for very selected values of k_T (to be concise) are displayed in Table 2. Apparently, the results are very close to those in Table 1, which implies that the polynomial approximation based on the information about the dominant (leading) root is sufficiently robust with respect to successive procedure of delay values shifting introduced in Algorithm 2.

5 Conclusion

The presented paper has been aimed at the possible polynomial approximation of a quasipolynomial by means of tools and techniques used for digital filters design; namely, the bilinear transformation with/without pre-warping and a specific linear

and quadratic interpolation for the acquisition of commensurate delays. This activity is useful mainly for stability and dynamical analysis of time delay systems when the characteristic quasipolynomial, as the transfer function denominator, is analyzed. Since the decisive information is contained in a small number of the rightmost, i.e. leading, quasipolynomial zeros, the approximating polynomial has been found iteratively based on the leading root estimation. A rather tricky step is a proper choice of the discretization step and the sampling period.

The presented simulation example has indicated the beneficial impact of the use of pre-warping and more complex, i.e. quadratic, interpolation to the leading root estimation accuracy and the approaching of both dominant roots in the complex conjugate pair, respectively. Besides the leading pair, a small number of less-dominant pairs must also be observed. This feature might play a leading role in the process of the determination of our future research in this field.

Acknowledgments The work was performed with the financial support by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme project No. LO1303 (MSMT-7778/2014) and also by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References

1. Chiasson, J., Loiseau, J.J.: Applications of Time Delay Systems. Springer, New York (2007)
2. Sipahi, R., Vyhlídal, T., Niculescu, S.-I., Pepe, P.: Time Delay Systems: Methods, Applications and New Trends. LNCIS, vol. 423. Springer, New York (2012)
3. Richard, J.P.: Time-Delay systems: an overview of some recent advances and open problems. Automatica **39**, 1667–1694 (2003)
4. Loiseau, J.J., Michiels, W., Niculescu, S.-I., Sipahi, R.: Topics in Time Delay Systems: Analysis, Algorithm and Control. LNCIS, vol. 388. Springer, Berlin (2009)
5. Hale, J.K., Verduyn Lunel, S.M.: Introduction to Functional Differential Equations. Applied Mathematical Sciences, vol. 99. Springer, New York (1993)
6. Zítek, P., Víteček, A.: Control Design of Time-Delay and Nonlinear Subsystems. CTU Publishing (1999) (in Czech)
7. Gu, K., Kharitonov, V.L., Chen, J.: Stability of Time-Delay Systems. Birkhäuser, Boston (2003)
8. Breda, D., Maset, S., Vermiglio, R.: Pseudospectral differencing methods for characteristic roots of delay differential equations. SIAM J. Sci. Comput. **27**, 482–495 (2005)
9. Vyhlídal, T., Zítek, P.: Quasipolynomial mapping algorithm rootfinder for analysis of time delay systems. In: Proceedings of the 4th IFAC Workshop on Time-Delay Systems (TDS 2003). Rocquencourt, France (2003)
10. Vyhlídal, T., Zítek, P.: Mapping based algorithm for large-scale computation of quasipolynomial zeros. IEEE Trans. Autom. Control **54**, 171–177 (2009)
11. Vyhlídal, T., Zítek, P.: QPmR—Quasi-Polynomial Root-Finder: Algorithm Update and Examples. In: Vyhlídal, T., Lafay, J.-F., Sipahi, R. (eds.) Delay Systems: From Theory to Numerics and Applications, pp. 299–312. Springer, New York (2014)
12. Partington, J.R.: Some frequency-domain approaches to the model reduction of delay systems. Ann. Rev. Control **28**, 65–73 (2004)
13. Pekař, L.: On a controller parameterization for infinite-dimensional feedback systems based on the desired overshoot. WSEAS Trans. Syst. **12**, 325–335 (2013)

14. Seuret, A., Özbay, H., Bonnet, C., Mounier, H.: Low Complexity Controllers for Time Delay Systems. Advances in Delays and Dynamics, vol. 2. Springer, New York (2014)
15. Middleton, R.H., Goodwin, G.C.: Digital Control and Estimation: A Unified Approach. Prentice Hall, Detroit (1990)
16. Vyhlídal, T., Zítek, P.: Discrete Approximation of a Time Delay System and Delta Model Spectrum. In: Proceedings of the 16th IFAC World Congress, p. 636. IFAC, Prague (2005)
17. Oppenheim, A.: Discrete Time Signal Processing. Pearson Higher Education, Upper Saddle River, NJ (2010)
18. Vanbiervliet, T., Verheyden, K., Michiels, W., Vandewalle, S.: A nonsmooth optimization approach for the stabilization of time-delay systems. ESIAM Control Optim. Ca. **14**, 478–493 (2008)
19. Balátě, J.: Automatic Control. BEN Publishing, Prague (2004). (in Czech)
20. Zítek, P., Kučera, V., Vyhlídal, T.: Meromorphic observer-based pole assignment in time delay systems. Kybernetika **44**, 633–648 (2008)
21. Pekař, L.: A Simple DDS Algorithm for TDS: An Example. In: Proceedings of the 29th European Conference on Modelling and Simulation (ECMS 2015), pp. 246–251. European Council for Modelling and Simulation (ECMS), Varna, Bulgaria (2015)

An Implementation of a Tilt-Compensated eCompass

Martin Sysel

Abstract This paper describes implementation of an electronic compass and calibration method. Firstly describes used hardware and then focus on the method of sensor calibration parameters. The eCompass uses a three-axis accelerometer and three-axis magnetometer. The compass heading is a function of all three accelerometer readings and all three magnetometer readings. The accelerometer measures the components of the earth gravity and provide pitch and roll angle information which is used to correct the magnetometer data. The magnetometer measures the components of earth's magnetic field (called geomagnetic field) to determine the heading angle to the magnetic north.

Keywords eCompass · MEMS · Accelerometer · Magnetometer · Calibration

1 Introduction

The first compass was probably a magnetized stone, that when suspended, would always point the same way. No one knows who first discovered the compass. The Chinese understood its use 3,000 years before Europeans learnt to travel without using the sun or the stars. Marco Polo is reputed to have brought the compass back to Europe on his return from Cathay in 1260. Todays all non-electronics compasses use a magnetized steel needle, supported in the middle. The important points of a compass are North, East, South and West; always read clockwise around the circle. North is found at 0° (which is also 360°), East becomes magnetic azimuth 090 (90°), South becomes magnetic azimuth 180 (180°), West becomes magnetic azimuth 270 (270°). More accurate directions are given by using all the numbers in between these. Always give readings in degrees, it is more accurate [1].

M. Sysel (✉)
Faculty of Applied Informatics, Tomas Bata University in Zlín, Zlín,
Czech Republic
e-mail: Sysel@fai.utb.cz

R. Silhavy et al. (eds.), *Automation Control Theory Perspectives in Intelligent Systems*, Advances in Intelligent Systems and Computing 466,
DOI 10.1007/978-3-319-33389-2_4

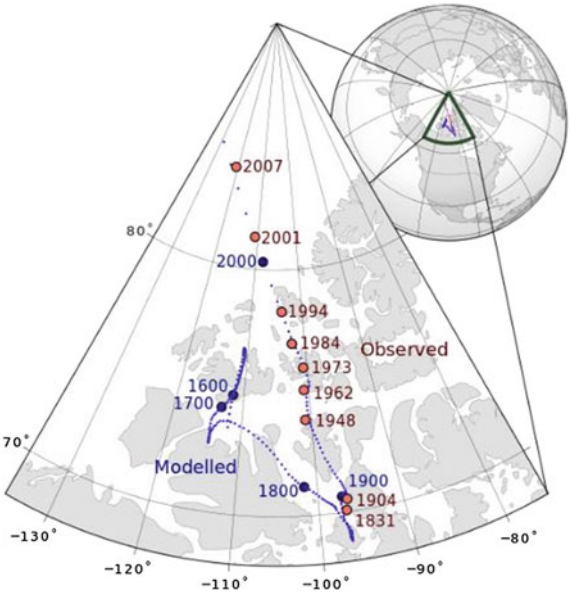

Fig. 1 Observed north dip poles

The compass points to the magnetic north, not true geographic North Pole. The magnetic north pole is actually south of the geographic North Pole. The compass therefore points a little to the side of the geographic North Pole, and this varies depending on where you are on Earth, and also with time. In fact, the magnetic pole migrate over time, it moves about 55 km a year, and the movement has speeded up recently [2]. Figure 1 shows observed north dip poles during 1831–2007 which are marked as yellow circles. Modeled pole locations from 1590 to 2020 are blue circles.

The Earth acts like a large spherical magnet: it is surrounded by a magnetic field. That changes with time and location. At any point and time, the Earth's magnetic field is characterized by a direction and intensity which can be measured.

The intensity of the magnetic field is about 0.25–0.65 gauss (25000–65000 nT) and has a component parallel to the earth surface that always points toward the magnetic North Pole. In the northern hemisphere this field point down. At the equator it points horizontally and in the southern hemisphere it points up. This angle between the earth's magnetic field and horizontal plane is defined as an inclination angle. Another angle between the earth's magnetic north and geographic north is defined as a declination angle in the range of ±20° depending on geographic location [3]. A magnetic compass needle tries to align itself with the magnetic field lines. However, near the magnetic poles, the fields are vertically converging. The strength and direction tend to "tilt" the compass needle up or down into the Earth.

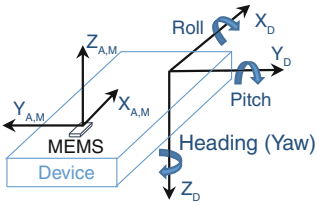

Fig. 2 eCompass coordinate system [3]

A tilt compensated electronics compass system uses a three-axis accelerometer sensor and three-axis magnetic sensor. The accelerometer is used to measure the tilt angles of pitch and roll for compensation. The magnetic sensor is used to measure the earth's magnetic field. Then it is possible to calculate the compass heading angle (yaw) with respect to the magnetic north. The declination angle at the current geographic location should be used for compensation of the heading to obtain true geographic North Pole. The Fig. 2 shows coordinate system which uses the industry standard "NED" (North, East, Down) to label axes on the device. The X_D axis of the device is the eCompass pointing direction, the Y_D axis points to the right and Z_D axis points downward. Positive yaw angle is defined to be a clockwise rotation about the positive Z_D axis.

Similar, positive pitch and roll angles are defined as clockwise rotations. When MEMS is installed in a device as shown in Fig. 2 then the sign of $Y_{A,M}$ and $Z_{A,M}$ from the sensor measurements needs to be reversed to make the sensing axes the same direction as the device axes [3, 4].

2 Accelerometer

2.1 Physical Principles and Structure

An accelerometer is an electromechanical device that measures acceleration forces. These forces may be static, like the gravity, or they could be dynamic—caused by moving or vibrating. There are many types of accelerometers and there are many different ways to make an accelerometer. Some accelerometers contain microscopic crystal structures that get stressed by accelerative forces use the piezoelectric effect. Another sensing changes in capacitance. Capacitive sensing has excellent sensitivity.

Typical MEMS accelerometer is composed of movable proof mass with plates that is attached through a mechanical suspension system to a reference frame, as shown in Fig. 3. A MEMS accelerometer differs from integrated circuits in that a proof mass is machined into the silicon. Any displacement of the component causes this mass to move slightly according to Newton's second law, and that change is

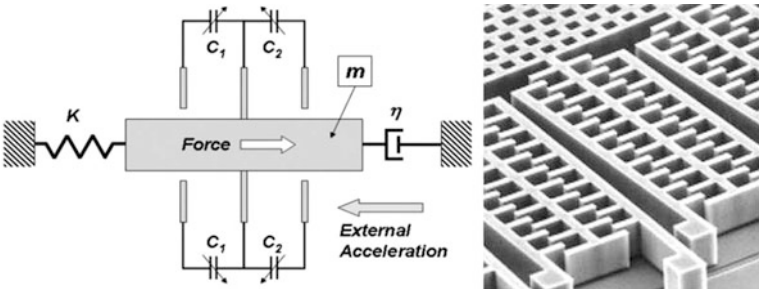

Fig. 3 Schematic and internal structure of a capacitive accelerometer [6]

detected by sensors. Usually the proof mass disturbs the capacitance of a nearby node; that change is measured and filtered. Movable plates and fixed outer plates represent capacitors. The deflection of proof mass is measured using the capacitance difference [5]. The free-space (air) capacitances between the movable plate and two stationary outer plates C_1 and C_2 are functions of the corresponding displacements [6].

The most important specification is the number of axis. The MEMS proof mass can measure only one parameter in each available axis, so a one axis device can sense acceleration in a single direction. Three axis units return sensor information in the X, Y, and Z directions.

Accelerometer sensors measure the difference between any linear acceleration in the accelerometer's reference frame and the earth's gravitational field vector. The earth's gravitational field is defined by a force vector that points directly down towards the earth's core. In the absence of linear acceleration, the accelerometer output is a measurement of the rotated gravitational field vector and can be used to determine the accelerometer pitch and roll orientation angles. The orientation angles are dependent on the order in which the rotations are applied. The most common order is the aerospace sequence of yaw then pitch and finally a roll rotation [7].

Three-axis accelerometers supplied for the consumer market are typically calibrated by the sensor manufacturer using a six-element linear model comprising a gain and offset in each of the three axes. This factory calibration will change slightly as a result of the thermal stresses during soldering of the accelerometer to the circuit board. Rotation of the accelerometer package relative to the circuit board and misalignment of the circuit board to the final product will also add small errors. The original factory accelerometer calibration is adequate for the majority of consumer applications. However, own calibration improve accuracy for high-accuracy applications (tilt applications, accuracies below 2°).