(IJNCAA)

ISSN 2220-9085(ONLINE)
ISSN 2412-3587(PRINT)

INTERNATIONAL JOURNAL OF

New Computer Architectures and their Applications

Volume 7, Issue 1
2017

SDIWC

www.sdiwc.net

International Journal of
NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

Overview

The SDIWC International Journal of New Computer Architectures and Their Applications (IJNCAA) is a refereed online journal designed to address the following topics: new computer architectures, digital resources, and mobile devices, including cell phones. In our opinion, cell phones in their current state are really computers, and the gap between these devices and the capabilities of the computers will soon disappear. Original unpublished manuscripts are solicited in the areas such as computer architectures, parallel and distributed systems, microprocessors and microsystems, storage management, communications management, reliability, and VLSI.

One of the most important aims of this journal is to increase the usage and impact of knowledge as well as increasing the visibility and ease of use of scientific materials, IJNCAA does NOT CHARGE authors for any publication fee for online publishing of their materials in the journal and does NOT CHARGE readers or their institutions for accessing the published materials.

Publisher

The Society of Digital Information and Wireless Communications
20/F, Tower 5, China Hong Kong City, 33 Canton Road, Tsim Sha Tsui, Kowloon, Hong Kong

Further Information

Website: http://sdiwc.net/ijncaa, Email: ijncaa@sdiwc.net,
Tel.: (202)-657-4603 - Inside USA; 001(202)-657-4603 - Outside USA.

The issue date is March 2017.

IJNCAA

ISSN 2220-9085 (Online)
ISSN 2412-3587 (Print)

Volume 7, Issue No. 1

2017

CONTENTS

ORIGINAL ARTICLES

International Journal of New Computer Architectures and their Applications (IJNCAA) 7(1): 1-7
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Reviewing and Analyzing Efficient GCD/LCM Algorithms for Cryptographic Design

Ibrahim Marouf
King Faisal University
i.marouf@outlook.com

Mohamad Musab Asad
King Faisal University
asadmosab@gmail.com

Qasem Abu Al-Haija
King Faisal University
qalhaija@kfu.edu.sa

ABSTRACT

In this paper, we provide a practical review with numerical example and complexity analysis for greatest common divisor (GCD) and Least Common Multiple (LCM) algorithms that are commonly used in the computing coprocessors design such as Cryptoprocessor design. The paper discusses four common GCD algorithms: Dijkstra's algorithm, Euclidian algorithm, Binary GCD algorithm, Lehmer's algorithm, and two LCM algorithms: LCM based Prime Factorizations algorithm and LCM based GCD reduction. It was found that Lehmer's algorithm can be used efficiently to compute GCD and LCM with time complexity of $\boldsymbol{O}(\frac{\boldsymbol{n}}{\boldsymbol{log}(\boldsymbol{n})})$ which enhances the linear time $(\boldsymbol{O}(\boldsymbol{n}))$ complexity of well-known Euclidian algorithm.

KEYWORDS

Greatest Common Divisor (GCD), Dijkstra's GCD, Euclidian GCD, Binary GCD, Lehmer's GCD, Least Common Multiple (LCM).

1 INTRODUCTION

Recently, the vast promotion in the field of information and communication technology (ICT) such as grid and fog computing has increased the inclination of having secret data sharing over the existing non-secure communication networks. This encouraged the researches to propose different solutions to ensure the safe access and store of private and sensitive data by employing different cryptographic algorithms especially the public key algorithms [1] which proved robust security resistance against most of the attacks and security halls. Public key cryptography is significantly based on the use of number theory and digital arithmetic algorithms.

Number theory [2] concerned with elementary arithmetic algorithms over the set of positive integers (i.e. natural numbers) such as divisibility, primes, congruence, GCD and others. It has rapidly grown in recent years in practical importance through its use in areas such as coding theory, cryptography and statistical mechanics.

GCD and LCM operations [2] are both essential elementary number theory algorithms that are commonly used in the design of many public key crytoprocessors such as RSA cryptosystem [3] and Schmidt-Samoa cryptosystem [4]. The good utilization of enhanced implementations of GCD/LCM will contribute in the overall cost enhancement for the coprocessor.

Recently, several solutions were proposed for speeding up GCD/LCM algorithms while few of them were efficient. For instance, Q. A. Al-Haija, M. Al-Ja'fari and M. Smadi [6] targeted Altera Cyclone IV FPGA family to design an efficient GCD (Greatest Common Divisor) coprocessor based on Euclid's method with variable datapath sizes. They tested their design using seven FPGA chip technologies in terms of maximum frequency and critical path delay of the coprocessor. As a result, they recorded the best values of maximum frequencies of 243.934 MHz down to 39.94 MHz for 32 bits and 1024-bit datapath, respectively.

Theoretically, J. Sorenson [10] provided an analysis of Lehmer's Euclidean GCD algorithm and a modified version that is faster than Euclid's algorithm roughly by a factor of k, where k is the number of bits in the multi-precision base of the algorithm. Also, another Euclid's based GCD algorithm is the parallel Lehmer-Euclid GCD algorithm was discussed in [11]. For long integers, Jebelean, T [12] proposed a Double-Digit-Lehmer-

Euclid's algorithm to speed up the computation of finding the GCD for long integers.

In a related work, a generalization of the binary algorithm for GCD computing were proposed in [13] where the authors used the concept of modular conjugate to find the GCD of multi-precision integers that is faster than Lehmer-Euclid method by a factor of two. Accordingly, this new method was suitable for systolic parallelization and in "least-significant digit first" pipelined manners.

Moreover, Wang, P.S [14] discussed an enhanced EZ-GCD algorithm with it implementation aspects. His algorithm was fast and particularly suited for computing GCD of multivariate polynomials. Similarly, other three algorithms for multivariate polynomial GCD were discussed in [15].

Furthermore, an execution time comparison between three different GCD algorithms (namely: Euclidean, binary, plus-minus) was analyzed by T. Jebelean [16]. These algorithms are the known ones. In addition, the author [16] proposed new improvement of Lehmer's GCD and generalization of binary algorithm which resulted in enhancement computing speed of GCD operation.

Finally, S.M. Sedjelmaci presented new parallelization of the extended Euclidean GCD algorithm that match the best existing integer GCD algorithms since it can be achieved in parallel O/sub eps/ (n/log n) using only n/sup 1+eps/ processors on a Priority CRCW PRAM, for any constant eps>0.

In this paper, we will review and summarize four efficient algorithms to compute GCD/LCM of two positive integers. The remaining of this paper is organized as follows: Section 2 discusses the different GCD algorithms with numerical examples and flowcharts. Section 3 provides the review of LCM algorithms with numerical examples and flowcharts. Finally, Section 4 concludes the paper.

2 GREATEST COMMON DIVISOR (GCD)

GCD is well known elementary number theory algorithm that is defined as the largest positive integer number that can divide two non-negative integer numbers without reminder. The very basic method to calculate GCD is using prime factorization method (PF-GCD).

PF-GCD method depends on fact that each number can uniquely written as a product of prime numbers raised to different powers. Then, to find the GCD, take only the smallest power of common primes factors of the two numbers. An example of the algorithm is shown below:

576 = 2632, 135 = 335➔ GCD (576, 135) = 32 = 9

However, more efficient algorithms have been proposed to compute GCD such as Dijkstra's algorithm, Euclidian's algorithm, Binary algorithm and Lehmer's algorithm.

2.1 Dijkstra's Algorithm

Dijkstra's algorithm [5] depends on the idea that if: $m|d$ and $n|d$ then $(m-n)\,|d$ with no reminder, as a result we can see: If m > n, $GCD\ (m,n)$ is the same as $GCD\ (m-n,n)$. The flowchart of this algorithm is shown in Figure 1.

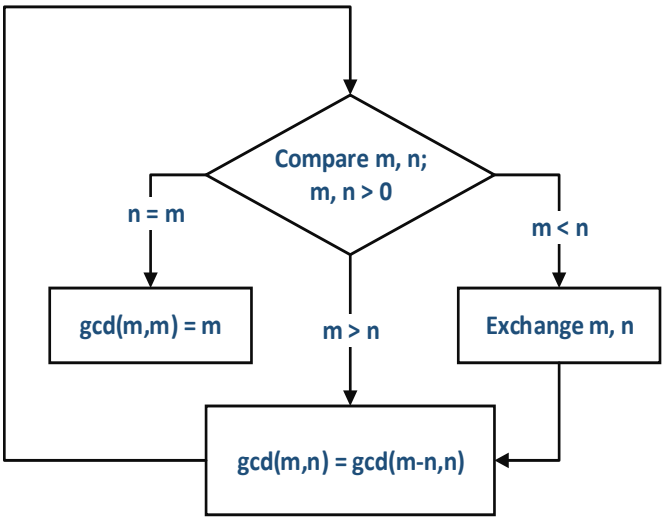


Figure 1. Dijkstra's Algorithm

An example of the algorithm is shown below:

GCD (36, 44) = GCD (36, 8) = GCD (28, 8) = GCD (20, 8) = GCD (12, 8) = GCD (4, 8) = GCD (4, 4) = 4

➔ GCD (36, 44) = 4;

2.2 Euclidian Algorithm

Euclidian algorithm [6] is simple well-known iterative method to calculate GCD. Suppose a and b are both integer numbers and $a > b$, if not exchange between a and b. The algorithm divide a

by b and find the reminder. If the reminder is equal to Zero then $GCD\ (a,b)\ =$ a. If not, then divide b by the reminder and find the new reminder. Continue this procedure until the reminder reach zero, the $GCD\ (a,b)$ is the final reminder. The flowchart of this algorithm is shown in Figure 2. From the figure, we can see that the complexity of Euclidian algorithm depends on $n\ =\ (a+b)$. The number of steps can be linear, for e.g. GCD $(x,1)$, so the time complexity is $O(n)$.

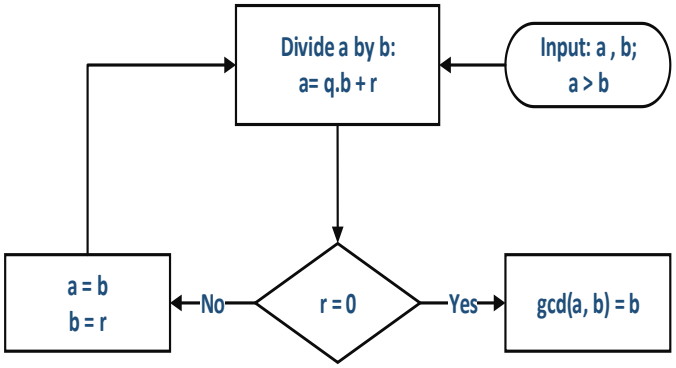


Figure 2. Euclidian Algorithm

An example of the algorithm is shown below:

GCD (24, 16): $24 = 1 \times 16 + 8\ ;8 \neq 0$ ➔
$16 = 2 \times 8 + 0$
Since r = 0 ➔ GCD (24, 16) = 8

2.3 Binary GCD

Binary GCD algorithm [7] replaces the division operations by arithmetic shifts, comparisons, and subtraction depending on the fact that dividing binary numbers by its base 2 is equivalent to the right shift operation. This algorithm is also known as Stein's algorithm which is illustrated in figure 3. The algorithm depends on the following facts:

1. If both a & b are 0, then GCD is zero: $GCD\ (0,0)\ =\ 0$.
2. $GCD\ (a,0)\ =\ a$ & $GCD\ (0,b)\ =\ b$ because everything divides 0.
3. If a & b are both even, $GCD\ (a,b)\ =\ 2 * GCD\ (a/2, b/2)$ because 2 is a common divisor. Multiplication with 2 can be done with bitwise shift operator.
4. If a is even & b is odd, $GCD\ (a,b)\ =\ GCD\ (a/2, b)$. Similarly, if a is odd & b is even, then GCD $(a,b)\ =\ GCD\ (a, b/2)$. It is because 2 is not a common divisor.
5. If both a & b are odd, then $GCD\ (a,b)\ =\ GCD\ (|a-b|/2, b)$ where b is min (a, b). Note that difference of two odd numbers is even
6. Then to complete the algorithm repeat steps 3–5 until a = b, or until a = 0. In either case, $GCD(a,b)\ =\ 2^k.b$, where k is the number of common factors of 2 found in step 3.

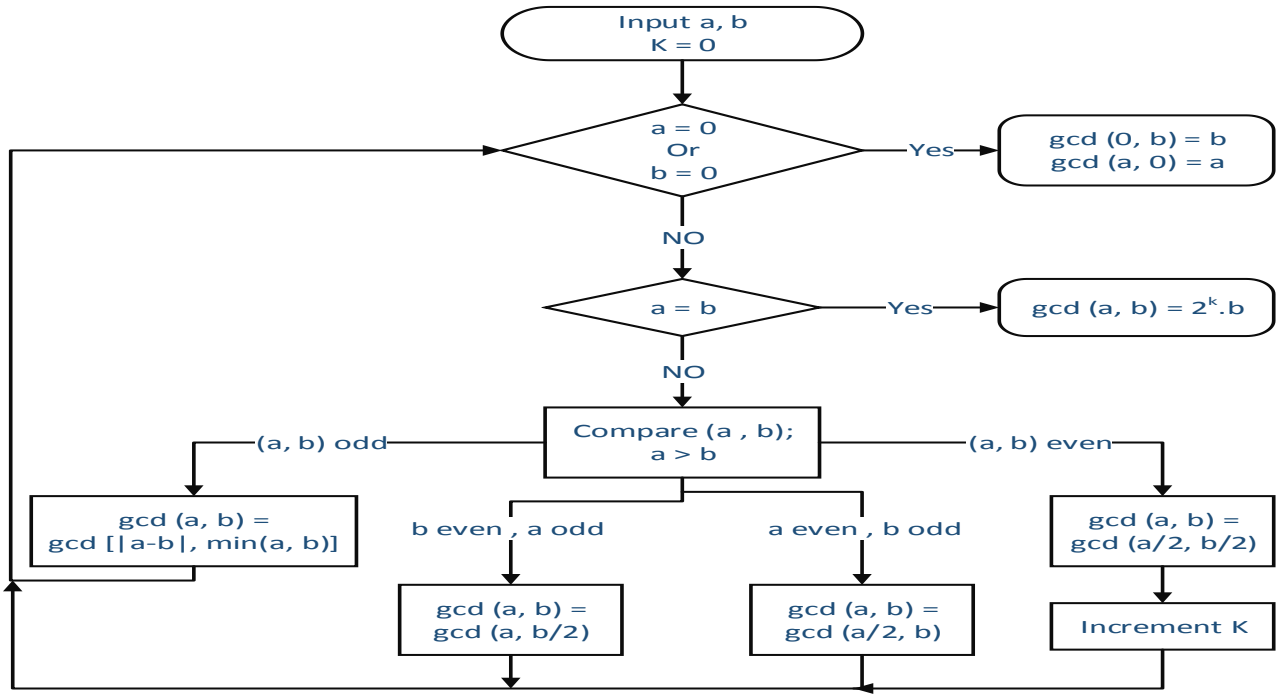


Figure 3. Binary GCD algorithm

The flowchart of this algorithm is shown in fig. 3. It's clearly seen that the complexity of binary GCD algorithm depends of the number of bits in the larger of the two numbers (n) which grow quadratically $(O(n^2))$.

An example of the algorithm is shown below:

GCD (16, 12) = GCD (8, 6) = GCD (4, 3) = GCD (2, 3) = GCD (1, 3) = GCD (2,1) = GCD (1, 1) = GCD (0,1) = 22 = 4

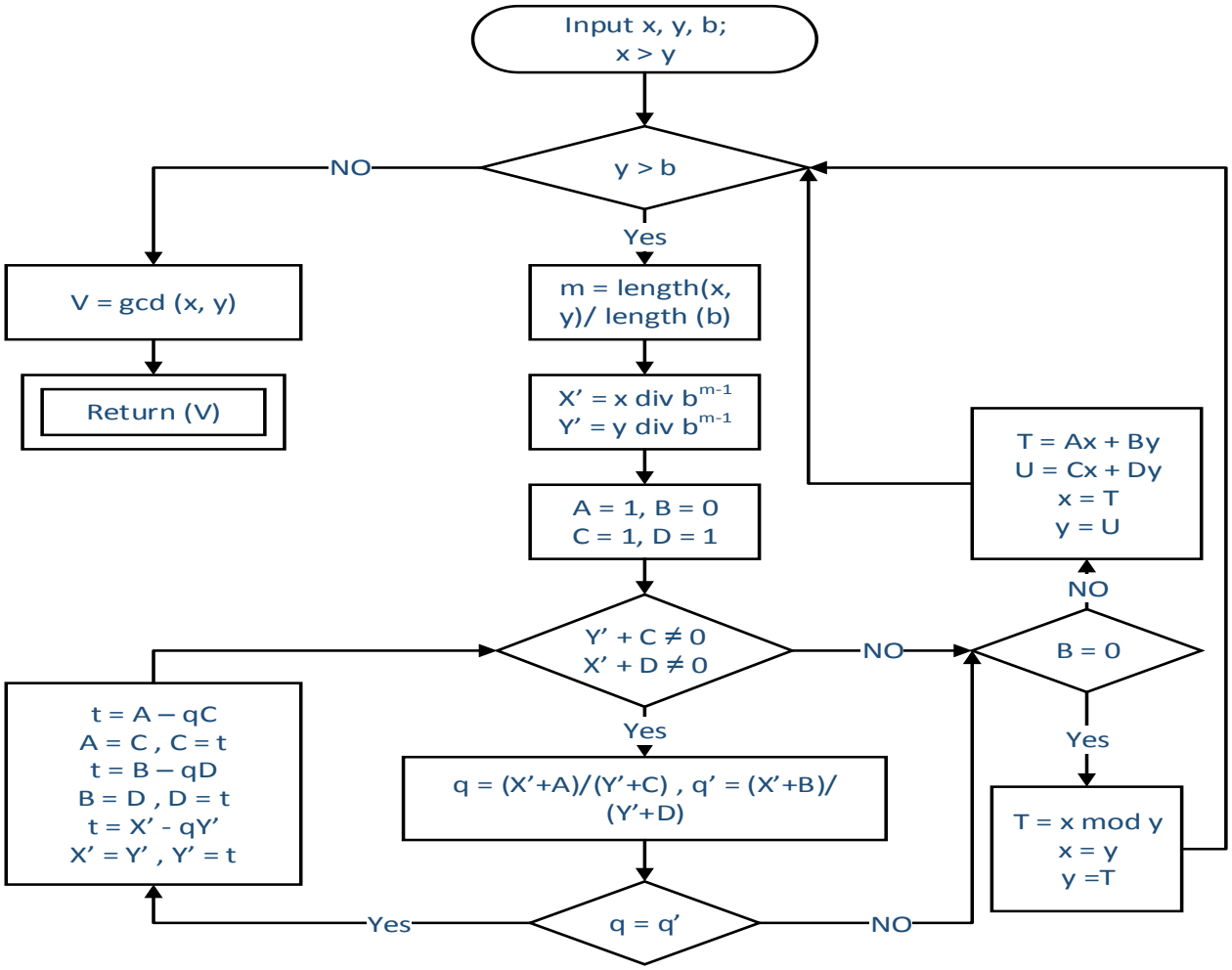


Figure 4. Lehmer's GCD algorithm

2.4 Lehmer's GCD algorithm

When the two numbers of GCD are very long, Euclidean algorithm will take longer time to compute GCD. However, Lehmer's GCD algorithm [8], is a fast GCD algorithm when two numbers are very long. The algorithm decreases the large numbers to smaller numbers of chosen base. The primary advantage of this algorithm is the reduction of the two numbers to the chosen base b. The "div" function in diagram means:

$$X' = x \; div \; b^{1-m} = \left\lfloor \frac{x}{b^{1-m}} \right\rfloor \qquad (1)$$

The function $V = GCD\ (x, y)$ is computed using and simpler GCD algorithms such as Euclidian Algorithm. The flowchart of this algorithm is shown in Figure 4. The cost complexity of lehmer's GCD algorithm is approaching faster with $O\ (n\ /\ log\ n)$.

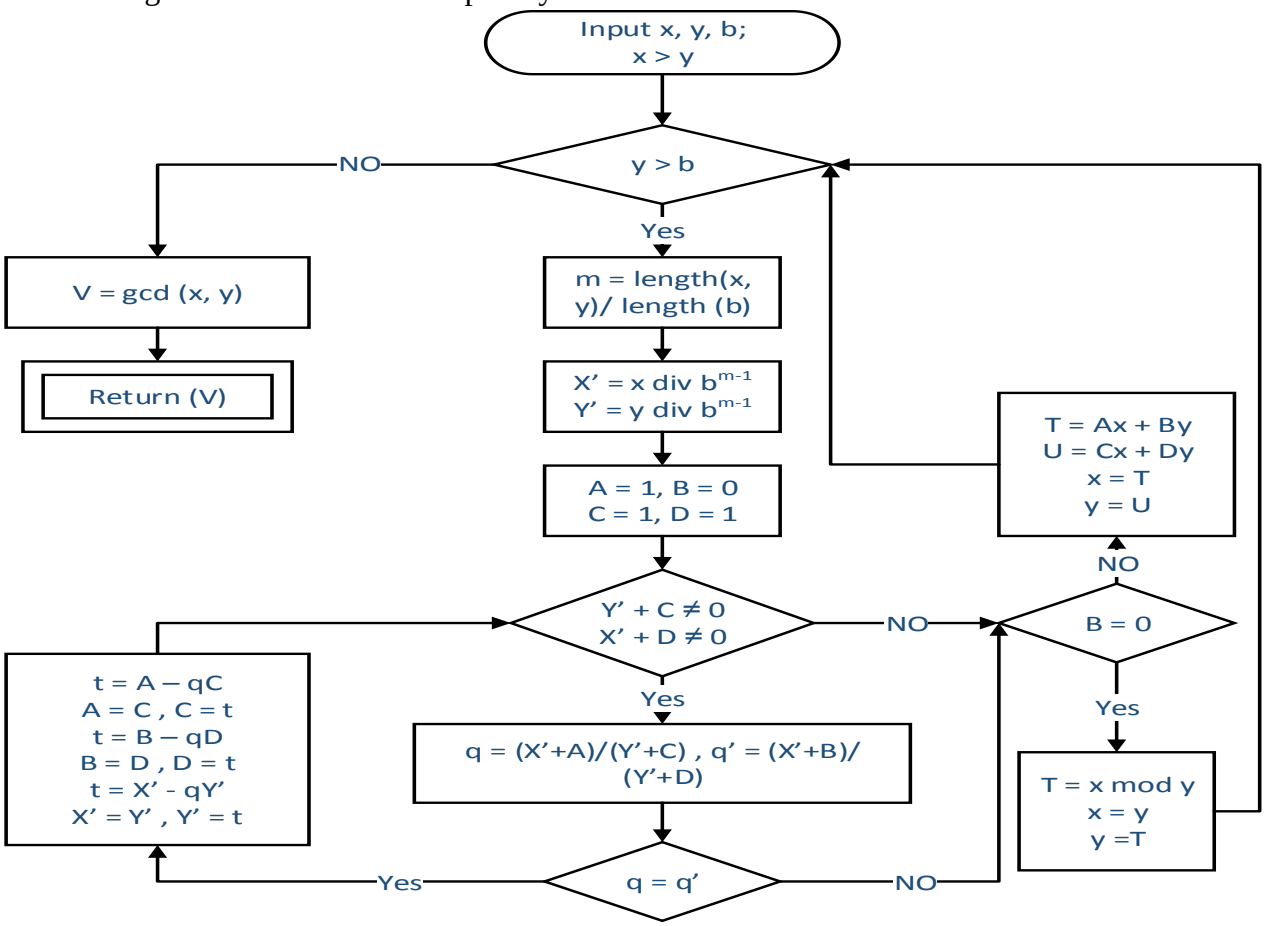


Figure 4. Lehmer's GCD algorithm

An example of the algorithm is shown below:
Let: x= 768,454,923 and y= 542,167,814, find $GCD\ (x, y)$. Firstly, we use Lehmer's algorithm to reduce x and y to the chosen base b = 103. Now, the high-order digit of x and y are x' = 768 and y' = 542 then run the algorithm as in table 1 and the table 2.

Table 2. Lehmer's GCD algorithm Example

x	y	q	q'	Reference
768,454,923	542,167,814	1	1	i
89,593,596	47,099,917	1	1	ii
42,493,679	4,606,238	10	8	
4,606,238	1,037,537	5	2	
1,037,537	456,090	5	2	
456,090	125,357	3	3	iii
34,681	10,657	3	3	iv
10,657	2,710	5	3	
2,710	2,527	1	0	
2,527	183			

Table 2. Reference Table

	x'	y'	A	B	C	D	q	q'
i	768	542	1	0	0	1	1	1
	542	226	0	1	1	-1	2	2
	226	90	1	-1	-2	3	2	2
	90	46	-2	3	5	-7	1	2
ii	89	47	1	0	0	1	1	1
	47	42	0	1	1	-1	1	1
	42	8	1	-1	-1	2	10	5

iii	456	425	1	0	0	1	3	3
	125	81	0	1	1	-3	1	1
	81	44	1	-3	-1	4	1	1
	44	37	-1	4	2	-7	1	1
	37	7	2	-7	-3	11	9	1
iv	34	10	1	0	0	1	3	3
	10	4	0	1	1	-3	2	11

Finally, we get x = 2,527 and y = 183, therefore, $v = GCD\,(x, y) = GCD\,(2{,}527, 183)$ which can be calculated easily using Euclidian's algorithm.

3 LEAST COMMON MULTIPLE (LCM)

LCM is well known elementary number theory algorithm that is defined as the smallest multiple integer of two numbers. The very basic method to calculate LCM is using prime factorization method (PF-LCM).

PF-LCM method depends on fact that each integer number can uniquely written as a product of prime numbers raised to different powers. Then, we can calculate LCM by taking all common factors with greatest power.

An example of the algorithm is shown below:

40 = 235, 26 = 2.13➔ LCM (40, 26) = 23.5.13 = 520

Alternatively, LCM can be efficiently calculated by using the GCD reduction method [9]. LCM is usually calculated using GCD from the formula:

$$LCM\,(a,b) = \frac{ab}{\mathrm{GCD}\;(a,b)} = \left(\frac{a}{\mathrm{GCD}(a,b)}\right) b$$

Where GCD can be calculated using different algorithms without need to factor the numbers.

4 CONCLUSIONS

GCD/LCM operations are essential number theory algorithms that commonly used as underlying operations of many computing processors. The efficient utilization of such algorithms contributes in the overall leverage of system execution. For instance, Lehmer's algorithm reduces the large integers to common base (b) with fast convergence rate at complexity of O (n/log (n)) which make it faster than other algorithms. Eventually, Lehmer's algorithm can be efficiently implemented (in hardware or software) to compute both GCD and LCM operations for any two large integer numbers.

REFERENCES

[1]. A.J. Menezes, P.C. Van Oorschot and S.A. Vanstone. (1996). Handbook of Applied Cryptography", CRC Press, Boca Raton, Florida

[2]. W. Trappe and L. C. Washington, (2002) 'Introduction to Cryptography with Coding Theory', Prentice Hall, vol. 1: p.p. 1-176.

[3]. Q. Abu Al-Haija, M. Smadi, M. Jaffri and A. Shua'ibi, "Efficient FPGA Implementation of RSA Coprocessor Using Scalable Modules", 9th International Conference on Future Networks and Communications (FNC-2014), by Elsevier, Ontario, Canada, 17-20, Aug-2014.

[4]. M. R. K. Ariffin, M. A. Asbullah, N. A. Abu and Z. Mahad, " A New Efficient Asymmetric Cryptosystem Based on the Integer Factorization Problem of $\boldsymbol{N = p^2q}$", Malaysian Journal of Mathematical Sciences 7(S): 19-37 (2013)

[5]. K. J. Goldman, "Recursive Algorithms", Computer Science I, lecture notes, Washington University in St. Louis, 1997.

[6]. Q. A. Al-Haija, M. Al-Ja'fari and M. Smadi, (2016) 'A comparative study up to 1024-bit Euclid's GCD algorithm FPGA implementation & synthesizing', 2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, United Arab Emirates, pp. 1-4.

[7]. D. Stehlé and P. Zimmermann (2004), "A binary recursive gcd algorithm", Algorithmic number theory (PDF), Lecture Notes in Comput. Sci., 3076, Springer, Berlin, pp. 411–425, MR 2138011, doi:10.1007/978-3-540-24847-7_31.

[8]. S. M. Sedjelmac, " On a Parallel Lehmer-Euclid GCD Algorithm", International Symposium on Symbolic and Algebraic Computation (ISSAC 2001), By ACM, UWO, Canada, 2001.

[9]. W. Stein, (2011) 'Elementary Number Theory: Primes, Congruence, and Secrets', Springer, vol. 1.

[10]. J. Sorenson. (1995). "An analysis of Lehmer's Euclidean GCD algorithm", Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation, pp. 254-258. Montreal, Canada, ACM Press.

[11]. Sidi Mohammed Sedjelmaci, "On a parallel Lehmer-Euclid GCD algorithm", Proceedings of the 2001 international symposium on Symbolic and algebraic computation, p.303-308, July 2001, London, Ontario, Canada. doi:10.1145/384101.384142
[12]. T. Jebelean. "A double-digit Lehmer–Euclid algorithm for finding the GCD of long integers", Journal of Symbolic Computation, 19 (1995), pp. 145-157
[13]. Jebelean, T., 1993. "A generalization of the binary GCD algorithm". In Proceedings of the 1993 international symposium on Symbolic and algebraic computation (pp. 111-116). ACM.
[14]. Wang, P.S., 1980. "The eez-gcd algorithm". ACM SIGSAM Bulletin, 14(2), pp.50-60.
[15]. Sasaki, T. and Suzuki, M., 1992. "Three new algorithms for multivariate polynomial GCD". Journal of symbolic computation, 13(4), pp.395-411.
[16]. T. Jebelean, "Comparing Several GCD Algorithms", ARITH-11: IEEE Symposium on Computer Arithmetic, 1993-June.
[17]. S. M. Sedjelmaci, "On a parallel extended Euclidean algorithm," Proceedings ACS/IEEE International Conference on Computer Systems and Applications, Beirut, 2001, pp. 235-241.

International Journal of New Computer Architectures and their Applications (IJNCAA) 7(1): 8-13
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Examining Stock Price Movements on Prague Stock Exchange Using Text Classification

Jonáš Petrovský, Pavel Netolický and František Dařena
Department of Informatics, Faculty of Business and Economics, Mendel University in Brno,
Zemědělská 1, 613 00 Brno, Czech Republic
jontesek@gmail.com, pavel.netolicky@gmail.com, frantisek.darena@mendelu.cz

ABSTRACT

The goal of the article was to examine the relationship between the content of text documents published on the Internet and the direction of movement of stock prices on the Prague Stock Exchange. The relationship was modeled by text classification. As data were used news articles and discussion posts on Czech websites and the value of the PX stock index and stock price of company CEZ. Document's class (plus/minus/constant) was determined by the relative price change that happened between the publication date of a document and the next working day. We achieved a high accuracy of 75% for classification of discussion posts, however the classification accuracy for news articles was about 60%. We tried both binary (documents with constant class were discarded) and ternary classification – the former was in all cases more successful.

KEYWORDS

Text Mining, Classification, Stock Market, Machine Learning

1 INTRODUCTION

Efficient markets theory (EMT) says that investors immediately incorporate all available information about given stock into its price and therefore the stock price is based solely on its fundamental value. However empirical observations contradict the EMT, because some price movements cannot be explained by change of fundamental figures [1]. Here comes the Behavioral finance theory which says that emotions may deeply influence behavior and decision making of individuals as well as whole human societies [2]. This means that prices on capital markets are (more or less) influenced by emotions, moods and opinions of market participants [3]. These characteristics are difficult to obtain, but could be present in text documents published on the internet (news articles, social media posts, etc.), which express both fundamental facts (rationality) and emotions and opinions of people (irrationality) [4]. To determine if the texts actually contain such information and that it is connected with stock price, we need to show and quantify a connection between texts and stock price movements. In other words, examine the influence of the (i)rationality of investors on stock market.

2 CURRENTLY USED METHODS

When trying to model behavior of a stock price we can use classification or regression. Many studies (e.g. [4]) in this area chose the former approach and decided to examine not the actual numeric price value, but only change of the value – they used direction of the change (up, down or constant) as a class. We focused on this approach as well, because we are not interested in the actual stock price value, but only in its change.

In fact, the problem represents a typical text classification task – given a document, determine to which class it belongs. For this, virtually any supervised learning algorithm may be used. However there are two main difficulties. Firstly, documents' classes have to be defined in a meaningful and useful way. Lee et al. [5] used 1% threshold value for determining the direction of a stock price change. Secondly, a suitable and effective set of features must be chosen. Many studies used as features just single words (unigrams) in so-called bag-of-words model with

satisfactory results [6]. The studies used different types of classifiers. Strength of the connection between texts and stock prices was evaluated by classification metrics (e.g. by accuracy) which are based on how many times the classifier assigns correct class to the given text.

3 DATA AND METHODOLOGY

The goal of the work was to examine the connection between content of text documents published on the Internet and direction of stock price movements, by using classification. A suitable approach had to be taken for working with every aspect of this task: handling prices and texts and processing the data via classification algorithms.

3.1 Stock prices

For the main part of the work, we used the PX index which reflects all companies traded on the Prague Stock Exchange (BCPP). The data were downloaded from the stock exchange's website (https://www.pse.cz). For every trading day, we used the closing value of the index. We also decided to examine discussion posts for one company (CEZ). Because BCPP contains data only since 2012, we downloaded it from www.akcie.cz.

3.2 Text data

The examined text data (documents) were downloaded from two sources (see Table 1). All documents were written in Czech language.

Table 1. Examined text data.

Source	Documents type	Number of doc.	Period	Average per day
Patria.cz	News articles about Czech stock market.	1 244	9. 2. 16 to 27. 5. 17 (15 mon.)	2.63
Akcie.cz	Discussion posts about 17 Czech companies.	20 605	14. 3. 08 to 27. 5. 17 (9y.)	6.13

Table 2 shows the information available for every text document. In subsequent analysis, all these fields apart from author were used.

Table 2. Available characteristics of a document with a concrete example of a discussion post regarding company CETV.

Field name	original in Czech	translated to English
datetime	2017-05-18 11:49:00	2017-05-18 11:49:00
author	mmmm	mmmm
title	Za vodou koncila na 94.	Offshore price ended on 94.
text	ja si myslim ze se dostane nekam k 85 ale nemam kouli samozrejme. :))	I think that it will get to 85, but I don't have crystal ball of course. :))

For every discussion post (Akcie.cz), it was known to which company on the stock exchange it belongs. However, for news articles (Patria.cz) this information was unknown. Moreover, it was found out, that a news article usually comments on multiple companies.

3.3 Classification methodology

We used classification to predict, whether a stock price will move up, down or stay constant on the basis of document's text. Each price movement represented a class. To obtain more diverse and possibly better results, we used both two (only up and down) and three classes for classification. It was expected that the ternary classification would perform worse, like mentioned in [7]. We extracted documents' features from the text by using the bag-of-words model. Every document was represented by a vector with values corresponding to the assigned weights of the words present in the document. For the experiments with all discussion posts (Akcie.cz) and news articles (Patria) values of the PX index were used. For one experiment (referred to as "CEZ experiment") stock prices and discussion posts related to only one company (CEZ) were used.

Document class. Assigning a class to a document was based on the relative price

change between two moments and on the threshold value (v) of minimal percentage price change. Formally, the percentage price return R in time t is:

$$R_t = (p_t - p_{t-1}) / p_{t-1}, \quad (1)$$

where price p_{t-1} is the closing price of the day when the document was published (or the last working day) and price p_t is close price the closing price of next working day. If the price return was in the constant interval (–v, +v), the document was either discarded from further processing (for binary classification) or assigned the "constant" class label (for ternary classification). If the price return was equal to or larger than +v, the document was labeled as "plus", otherwise as "minus". We used 0.25, 0.5 and 1.0% as the threshold values.

Text pre-processing and conversion. The text was processed as follows:
1. Join document title and text into one string.
2. Strip diacritics from text (convert "special" Czech letters to their ASCII equivalents).
3. Strip all HTML tags.
4. Lowercase and remove punctuation.
5. Tokenize – get words (using *TreebankWordTokenizer*).
6. Filter words – minimal length of 3 letters, exclude numbers.

The edited text had to be converted into a structured format. For this, a Python library called *scikit-learn* and its Vectorizer class were used. Only words which occurred at least 5 times (for discussion posts) and 10 times (for news articles) in the whole document collection were considered. Those words were converted to a bag-of-words representation using three different weighting schemes [8, p. 21–26]:
• Term Presence (TP): 1 if a term is present in a document, 0 if not.
• Term Frequency (TF): how many times is a term is present in a document.
• TF-IDF: TF (local weight) multiplied by IDF (global weight).

Classification. Converted data were processed again by *scikit-learn*. The data were split into training (60%) and testing (40%) datasets. Class balancing was not performed. Each of the generated vector representations was processed by 20 different classifiers (with default settings – we did not optimize the parameters of the classifiers). The performance of a classifier was rated by the achieved accuracy (proportion of correctly classified instances on all examined instances [9, p. 268]) on the test set.

4 RESULTS AND DISCUSSION

Three different sets of text data, all discussion posts (Akcie.cz), posts related to the CEZ company, and news articles (Patria) together with information about stock prices were used to prepare data for classification. Based on the combination of variable experimental parameters – the number of classes (2 or 3), minimal percentage change (0.25, 0.5 and 1%) and weighting scheme for the term-document matrix (TP, TF, TF-IDF) – 54 different sets were created and subsequently processed by 20 classification algorithms.

In total 1080 classification results were obtained. We evaluated the results for each data set separately and for each classification set, the highest accuracy achieved by any combination of vector type and classification algorithm was found. Our findings are presented in Table 3, Table 4 and Table 5. Class 1 means "minus", class 2 "constant" and class 3 "plus".

Table 3. Classification of Akcie.cz discussion posts

Num. of classes	Percent change	Accuracy	Total samples	Class 1 samples	Class 2 samples	Class 3 samples	Num. of words
2	0.25	0.74	16 673	8 169	0	8 504	10 604
2	0.50	0.76	13 449	6 454	0	6 995	9 181
2	1.00	0.78	8 170	3 939	0	4 231	6 359
3	0.25	0.67	20 576	8 169	3 903	8 504	12 337
3	0.50	0.65	20 576	6 454	7 127	6 995	12 337
3	1.00	0.79	20 576	3 939	12 406	4 231	12 337

Table 4. Classification of Patria news articles

Num. of classes	Percent change	Accuracy	Total samples	Class 1 samples	Class 2 samples	Class 3 samples	Num. of words
2	0.25	0.62	874	360	0	514	2 347
2	0.50	0.59	587	246	0	341	1 788
2	1.00	0.61	222	100	0	122	872
3	0.25	0.40	1 244	360	370	514	3 036
3	0.50	0.52	1 244	246	657	341	3 036
3	1.00	0.79	1 244	100	1 022	122	3 036

Table 5. Classification of CEZ discussion posts

Num. of classes	Percent change	Accuracy	Total samples	Class 1 samples	Class 2 samples	Class 3 samples	Num. of words
2	0.25	0.71	6 162	2 929	0	3 233	5 235
2	0.50	0.72	5 300	2 494	0	2 806	4 601
2	1.00	0.76	3 935	1 797	0	2 138	3 714
3	0.25	0.63	7 281	2 929	1 119	3 233	5 923
3	0.50	0.65	7 281	2 494	1 981	2 806	5 923
3	1.00	0.80	7 281	1 797	3 346	2 138	5 923

If we look at how balanced the datasets are, we can say that for 2 classes they are in all cases relatively well balanced. For 3 classes, there is a clear misbalance. This can be seen especially in Table 6 where 82% of samples belongs to the constant class.

If we compare the classification accuracy for different datasets, we see that for discussion posts it is far higher (+10%) than for news articles. Interesting is that the accuracy obtained by training a classifier on all discussion posts and the PX index (Table 3) is higher than when using only posts and prices for one company (Table 5).

The highest accuracy was achieved always for 3 classes and 1% change. If we consider only 0.25 and 0.50% changes, accuracy for 2 classes is always better than for 3 classes. Generally, it can be said that the higher the percentage change, the higher the accuracy. However, this does not hold for Patria news articles with 2 classes (Table 4), where is the highest accuracy achieved for 0.25% change.

Table 6. Comparison of avg. accuracies for vector type

Vector type	Akcie.cz	CEZ	Patria
TP	0.67	0.63	0.51
TF	0.66	0.63	0.51
TF-IDF	0.67	0.63	0.53

Table 6 tells us that for discussion posts the used vector type was not very important, however for news articles the highest accuracy was achieved by TF-IDF.

Table 7. Comparison of avg. accuracies for classification algorithms

Akcie.cz – discussion posts		Patria – news articles	
Algorithm	Avg. acc.	Algorithm	Avg. acc.
ExtraTrees	0.72	LogisticRegression	0.56
MLP	0.72	CalibratedClassifier	0.56
RandomForest	0.71	SVC	0.56
LogisticRegression	0.71	LogisticRegression	0.53
LinearSVC	0.71	RidgeClassifier	0.53

Table 7 shows classifiers with the best average performance across all experiments. For discussion posts “Extremely randomized trees” ensemble method was the best, closely followed by “Multi-layer Perceptron” neural network. However, out of the best five algorithms for news articles, only one (LogisticRegression) was successful also for the posts. This indicates that for each type of document different algorithms are suitable.

5 CONCLUSION

The goal of the article was to examine the relationship between the content of text documents published on the Internet and the direction of movement of stock prices on the Prague Stock Exchange. For this, text classification was used.

The connection was found as demonstrated by the achieved classification accuracy. When using binary classification (documents with constant class were discarded), we achieved an accuracy of 75-78% for discussion posts and about 60% for news articles. For ternary classification, the accuracy was lower (about 65% and 40-50%). However, for all datasets was the accuracy, when using the highest 1% threshold for minimal price change, 80 %.

During the work, we encountered several problems. The most notable one was a rather small amount of available data – especially the news articles.

It must be noted that the goal was to examine if there is a connection between texts and stock prices, not to achieve the highest possible accuracy for each classification algorithm. Because of this, we used only default settings (parameters’ values) for the algorithms. An optimization of these parameters might bring us a few percent higher accuracy.
There are many options for further research in this area: use clustering/topic models (e. g. LDA) to find document classes based on their content; use bigrams or tri-grams as features; take into account the importance (popularity) of the document, use more values for minimal price change and also other time interval (more or less than 1 day).

REFERENCES

[1] SHILLER, R. J. From efficient markets theory to behavioral finance. The Journal of Economic Perspectives. 2003, vol. 17, no. 1, p. 83–104.

[2] BOLLEN, J., MAO, H. and ZENG, X. Twitter mood predicts the stock market. Journal of Computational Science. 2011, vol. 2, no. 1, p. 1–8.

[3] KAPLANSKI, G. and LEVY, H. Sentiment and stock prices: The case of aviation disasters. Journal of Financial Economics. 2010, vol. 95, no. 2, p. 174–201.

[4] ARIAS, M., ARRATIA, A. and XURIGUERA, R. Forecasting with Twitter Data. ACM Transactions on Intelligent Systems and Technology (TIST). 2013, vol. 5, no. 1, p. 8.

[5] LEE, Heeyoung, et al. On the Importance of Text Analysis for Stock Price Prediction. In: LREC. 2014. p. 1170-1175

[6] Schumaker, R. P., Zhang, Y., Huang, C.-N., and Chen, H. (2012). Evaluating sentiment in financial news articles. Decision Support Systems.

[7] DARENA, F., PETROVSKY, J., ZIZKA, J. and PRICHYSTAL, J. Analyzing the correlation between online texts and stock price movements at micro-level using machine learning," MENDELU Working Papers in Business and Economics 2016-67, Mendel University in Brno, Faculty of Business and Economics. Available at: https://ideas.repec.org/p/men/wpaper/67_2016.htm

[8] WEISS, S. M., INDURKHYA, N. and ZHANG, T. Fundamentals of Predictive Text Mining. London: Springer, 2010.ISBN 978-1-84996-225-4.

[9] MANNING, C. D. and SCHÜTZE, H. Foundations of Statistical Natural Language Processing. Cambridge, USA: The MIT Press, 1999. ISBN 9780262133609.

International Journal of New Computer Architectures and their Applications (IJNCAA) 7(1): 14-17
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Mobile Robot Localization Based on Multi-Sensor Model for Assistance to Displacement of People with Reduce Mobility

Wassila. Meddeber, Youcef. Touati and Arab. Ali-Cherif
Computer Science and Artificial Intelligence Lab. LIASD
University of Paris 8
Saint-Denis, France
meddeberwassila@ ai.univ-paris8.fr, touati@ai.univ-paris8.fr, aa@ai.univ-paris8.fr

***Abstract* —This paper deals multi-sensor data fusion problem for mobile robot localization. In this context, we have used data fusion sensors: encoders and ultrasonic sensor. To improve the robustness of localization and to reduce the estimation error we have proposed a Kalman Particle Kernel Filter (KPKF) approach, which is based on a hybrid Bayesian filter, combining both extended Kalman and particle filters. The KPKF filter using a Gaussian mixture in which each component has a small covariance matrix. The Kalman correction updates the weights in order to bring particles back into the most probable space area. This method can be applied for non-linear and multimodal environment and can improve localization performances and reduced estimation error. The proposed approach is implemented on a LIASD-Wheelchair experimental platform.**

Keywords—Localization; multi-sensor; data fusion; mobile robotics; Kalman filter; particle filter; smart wheelchair.

I. INTRODUCTION

Several works have been undertaken to assists and help the handicapped and elderly people to gain mobility and lead to independent life and particularly those concerning the development of services related to automated wheelchairs. Make a wheelchair intelligent and autonomous, allows us to develop new methodologies taking into account the type of handicap, environment dynamics, new communication technologies such as sensor networks and wireless mesh networks and so on. In this direction, localization process is one of the main services that have been prospected in order to ensure assisted people a better mobility and assistance in their life [1], [2]. It constitutes a key problem in mobile robotics and it consists of estimating the robot's pose (position, orientation) with respect to its environment from sensor data, and the simplest way is integrating of odometric data which, however, is associated with unbounded errors, resulting from uneven floors, wheel slippage, limited resolution of encoders, etc [3], [4]. However, such a technique is not reliable due to cumulative errors occurring over longer runs. Therefore, a mobile robot must be able to localize or estimate its parameters also with respect to an internal world model by using the information obtained with its exteroceptive sensing system [5].

The use of sensory data from a range of disparate multiple sensors, is to automatically extract the maximum amount of information possible about the sensed environment under all operating conditions [6], [7]. The main idea of data fusion methods is to provide a reliable estimation of robot's pose, taking into account the advantages of the different sensors [8].

This paper focuses on robust pose estimation for mobile robot localization. A new hybrid Particle filter method called Kalman-Particle Kernel Filter (KPKF) is proposed to minimize the system estimation error and increase the localization robustness. It's organized as follows: In section II we present and discuss some multi-sensor data fusion methods. In section III, a proposed KPKF is presented. In section IV, an example of a localization process applied on the LIASD-Wheelchair is illustrated. Finally, in section V, conclusion and some perspectives are addressed.

II. RELATED WORKS

The Kalman Filter (KF) is the best known and most widely applied parameter and state estimation algorithm in data fusion methods [9], [10]. It can be considered as a prediction-update formulation. The algorithm uses a predefined linear model of the system to predict the state at the next time step [11], [12]. The prediction and update are combined using the Kalman gain, which is computed to minimize the mean square error of the state estimate. The KF diagram is illustrated in Fig.1.

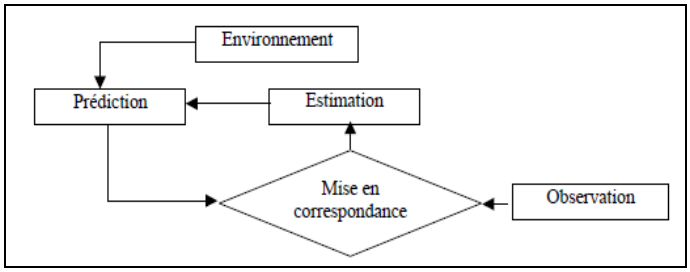


Fig. 1. Kalman filter diagram (KF).

Extended Kalman Filter (EKF) is a new version of KF that can handle non-linear measurement equations. Various EKF based-approaches have been developed. These approaches work well as long as the used information can be described by simple statistics well enough. The lack of relevant information is compensated by the use of various processes models [13]. However, they require assumptions about parameters, which might be very difficult to determine. Assumptions that guarantee optimum convergence are often violated and, therefore, the process is not optimal or it can even converge. The Kalman filter techniques rely on approximated filter, which requires tuning of modelling parameters, such as covariance matrices, in order to deal with model approximations and bias on the predicted pose. In order to compensate such error sources, local iterations, adaptive models and covariance intersection filtering have been proposed [14]. An interesting approach solution was proposed in [15], where observation of the pose corrections is used for updating of the covariance matrices. However, this approach seems to be vulnerable to significant geometric inconsistencies of the world models, since inconsistent information can influence the estimated covariance matrices.

In the localization problem is often formulated by using a unique model, from both state and observation processes point of view. Such an approach, introduces inevitably modelling errors, which degrade filtering performances, particularly, when signal-to-noise ratio is low and noise variances have been poor estimated. Moreover, to optimize the observation process, it is important to characterize each external sensor not only from statistic parameters estimation point of view but also from robustness of observation process point of view [16]. It is then interesting to introduce an adequate model for each observation area in order to reject unreliable readings. In the same manner, a wrong observation leads to a wrong estimation of the state vector and consequently degrades localization algorithm performance.

Particle Filter (PF) based-methods are considered as a sequential version of the Monte Carlo methods [17]. They represent the most effective methods for nonlinear localization of mobile systems. These methods have the ability to manage a set of particles in order to determine positions, and orientations. The principle of PF is to make the particles evolving in the same way as the robot to determine new positions and then comparing its perceptions to those of the particles. We retrieve the model values odometry (prediction) between two successive moments then transmitted to the filter function for correction by the observation model. After a small number of iterations, this process converges into a position where a population of particles is very dense. The PF method is illustrated in Fig. 2.

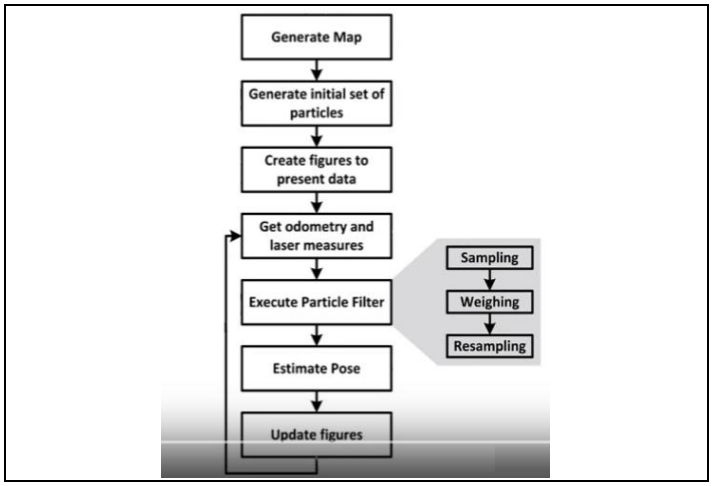


Fig. 2. Particle filter diagram (PF).

However, this filter can be very costly to implement, as a very large number of particles is usually needed, especially in high dimensional system. In case of low dynamical noise, we observe that in multiplying the high weighted particles, the prediction step will explore poorly the state space. The particle clouds will concentrate on few points of the state space. This phenomenon is called particle degeneracy, and causes the divergence of the filter.

Despite the research efforts to improve filters performance for data fusion, their behaviors remain unstable for some applications such as navigation and localization

III. PROPOSED KPKF FOR MULTI-SENSOR DATA FUSION

The Kalman-Particle Kernel Filter (KPKF) combines both an EKF and a PF for a robust localization system by adjusting the state of mobile system and reducing the estimation error. This new filter is a kind of hybrid particle filter. It is based on the representation of the kernel of conditional density and on a local linearization as a Gaussian mixture [18]. The KPKF filter method can be implemened according to three steps, as it is shown in Figure 3:

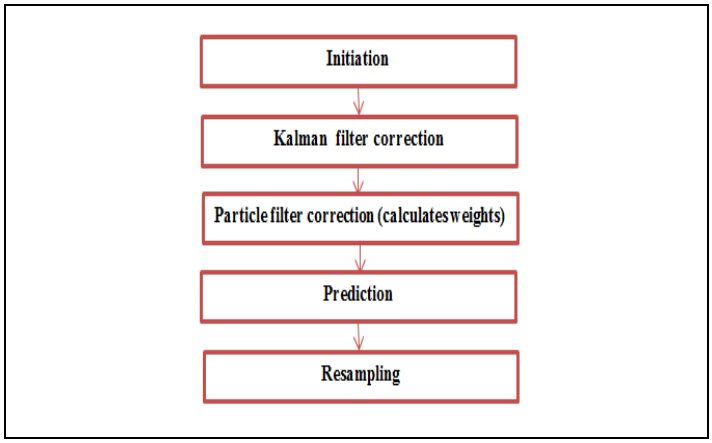


Fig. 3. Kalman-Particle Kernel Filter diagram.

- **Correction step**: is divided into two steps: a Kalman correction and a particle correction. The correction step ensures a mixture of Gaussian distribution of the filter density in order to increase the probability of the presence of the particles in the state space.
- **Prediction step:** therefore, this step is still a mixture of Gaussian distribution. In fact, the predicted density is modeled in the same way as the corrected density.
- **Resampling step**: is introduced to further reduce the divergence of the particulate filter (Monte Carlo).

IV. Implementation

We present an application of the Kalman-Particle Kernel Filter (KPKF) approach for a robust localization adapted to disabled and elderly people. Our approach is implemented and tested on a prototype called LIASD-Smart Wheelchair, developed in our laboratory (Informatics and Artificial Intelligence) [19]. The LIASD-Smart Wheelchair is equipped with data fusion sensors: ultrasonic sensor and, encoders.

LIASD-Wheelchair is an adjustable adults' powered wheelchair (Fig. 4). It is suitable for indoor or outdoor use and implements wired and wireless networks for communication. The wireless communication is based on two standards: IEEE 802.11 and IEEE 802.15.4. A wireless router is integrated to ensure communication between remote computer (server) and wheelchair devices (camera, embedded computer, etc.).

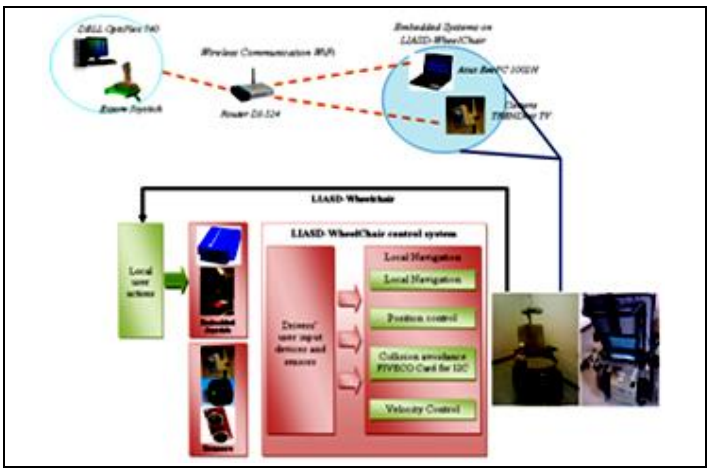


Fig. 4. Global structure of LIASD-WheelChair.

A. Hardware Architecture

The hardware architecture of LIASD-Wheelchair consists of sensory block, control architecture, and communication networks. The presented system includes two optical incremental encoders mounted to a motor, with resolution of 500 Counts per Revolution. Four ultrasonic sensors (US SFR08) are used to localize the wheelchair in the environment. They have a resolution of 3cm and can identify obstacles between 3cm and 6m. The US sensors interact with the computer via TCP/IP server board *FMod-TCP DB* using an I2C interface. In order to ensure navigation and anti-collision objectives a Wireless Internet Camera Server is mounted on the wheelchair headrest.

B. Control Architecture

The LIASD-WheelChair control architecture is divided into three levels: a basic control level, a tactical level and strategy level, as shown in Fig. 5. The strategy level concerns the way the wheelchair system can achieve the main goal. Algorithms such as planning trajectories, localization, etc. are implemented to fulfill the desired task. Elementary actions are, then generated in tactical level aiming to satisfy reached goals specified previously. Basic control level implements PID controller in the PWM/encoders boards with specific parameters for positions and speeds control.

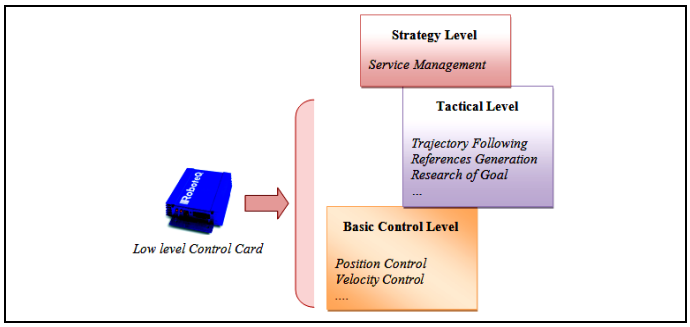


Fig. 5. LIASD-WheelChair control architecture.

Using its measurements and the characteristics of the geometrical model of the dynamic system, we determine at each moment the position and the orientation of the smart wheelchair. The odometry model introduces a major fault of the slipping of the rounds (noise of measurement) An unreliable location. To correct the odometry error, we use the data fusion theoretical method by adding the measurement of the ultrasonic sensor (observation model).

V. Conclusion and Future Works

The localization system is a complex multi-sensor process. To solve the problem of multimodality and non-linearity, we have proposed a new adaptation filter for data fusion, called Kalman-Particle Kernel Filter. The KPKF is a mixture of extended Kalman filter and particulate filter combining the advantages of both filters. Our approach is implemented on a mobile platform developed in our laboratory called LIASD Smart Wheelchair. The aim is to improve the quality of service in terms of mobility and assistance to displacement of persons with disabilities. A full test of our system is still in progress to demonstrate that our filtering approach is very effective for the robustness of system localization. Therefore, this method could use the research work that deals with the issue of the localization and navigation of stand-alone vehicles.

Acknowledgment

This article is part of the research development and innovation project (LIASD-Wheelchair). It deals with the problem of mobile system indoor localization for navigation.

Special thanks to professor ARAB ALI CHERIF, Director of the LIASD laboratory at University Saint Denis Paris 8 and all the colleagues involved in this project.

REFERENCES

[1] A. Lankenau, T. Rofer. A Versatile and Safe Mobility Assistant[J]. IEEE Robotics & Automation Magazine, 2001, Vol.8 (1), pp.29-37.

[2] E. Prassler, J. Scholz and P Fiorini. A robotics wheelchair for crowded public environment[J]. IEEE Robotics Automation Magazine, 2001, Vol.8(1), pp.38-45.

[3] J. Borenstein, B. Everett, L. Feng, Navigating Mobile Robots: Systems and Techniques, A.K. Peters, Ltd., Wellesley, MA, 1996.

[4] K.O. Arras, N. Tomatis, B.T. Jensen, R. Siegwart, Multisensor on-the-fly localization: precision and reliability for applications, Robotics and Autonomous Systems 34 (2001) 131–143.

[5] S. Borthwick, M. Stevens, H. Durrant-Whyte, Position estimation and tracking using optical range data, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1993, pp. 2172–2177.

[6] J.A. Castellanos, J.D. Tardós, Laser-based segmentation and localization for a mobile robot, in: F.P.M. Jamshidi, P. Dauchez (Eds.), Robotics and Manufacturing: Recent Trends in Research and Applications, vol. 6, ASME Press, New York, 1996, pp. 101–109.

[7] M. Jenkin, E. Milios, P. Jasiobedzki, Global navigation for ARK, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1993, pp. 2165–2171.

[8] J.J. Leonard, H.F. Durrant-Whyte, Mobile robot localization by tracking geometric beacons, IEEE Transactions on Robotics and Automation 7 (3) (1991) 376–382

[9] J. Neira, J.D. Tardós, J. Horn, G. Schmidt, Fusing range and intensity images for mobile robot localization, IEEE Transactions on Robotics and Automation 15 (1) (1999) 76–84.

[10] H. Wang, M. Kung, T. Lin, Multi-model adaptive Kalman filters design for maneuvering target tracking, International Journal of Systems Sciences 25 (11) (1994) 2039-2046.

[11] L. Jetto, S. Longhi, G. Venturini, Development and experimental validation of an adaptive Kalman filter for the localization of mobile robots, IEEE Transactions on Robotics and Automation 15 (2) (1999) 219–229.

[12] J.B. Gao, C.J. Harris, Some remarks on Kalman filters for the multi-sensor fusion, Journal of Information Fusion 3 (2002) 191-201.

23-24.

[13] C. Harris, A. Bailley, T. Dodd, Multi-sensor data fusion in defense and aerospace, Journal of Royal Aerospace Society 162 (1015) (1998) 229-244.

[14] G.A. Borges, M.J. Aldon, Robustified estimation algorithms for mobile robot localization based geometrical environment maps, Robotics and Autonomous Systems 45 (2003) 131-159.

[15] X. Xu, S. Negahdaripour, Application of extended covariance intersection principle for mosaic-based optical positioning and navigation of underwater vehicles, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2001, pp. 2759–2766.

[16] X. R. Li, "Engineer's guide to variable-structure multiple-model estimation for tracking," in *Multitarget-Multisensor Tracking: Applications and Advances*, Y. Bar-Shalom and D.W. Blair, Eds. Boston, MA: Artech House, 2000, vol. III, ch. 10, pp. 499–567.

[17] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Automat. Contr.*, vol. 33, pp. 780–783, Aug. 1988.

[18] X. R. Li, "Hybrid estimation techniques," in *Control and Dynamic Systems: Advances in Theory and Applications*, C. T. Leondes, Ed. New York: Academic, 1996, vol. 76, pp. 213–287.

[19] Y. Touati, H. Aoudia, and A. Ali-Chérif, Intelligent Wheelchair localization in wireless sensor network environment: A fuzzy logic approach, 5 th IEEE International Conference on Intelligent Systems, 2010, London, UK , pp.408-413.

International Journal of New Computer Architectures and their Applications (IJNCAA) 7(1): 18-28
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print))

Dimensionality Reduction for State-action Pair Prediction based on Tendency of State and Action

Masashi Sugimoto[1*] Naoya Iwamoto[1] Robert W. Johnston[1]
Keizo Kanazawa[1] Yukinori Misaki[1] Hiroyuki Inoue[2]
Manabu Kato[2] Hitoshi Sori[2] Shiro Urushihara[1]
Kazunori Hosotani[2] Hitoshi Yoshimura[3] Kentarou Kurashige[4]

[1]National Institute of Technology, Kagawa College
[2]National Institute of Technology, Tsuyama College
[3]National Institute of Technology, Tomakomai College
[4]Muroran Institute of Technology

Email([*]: corresponding author): sugimoto-m@es.kagawa-nct.ac.jp

ABSTRACT

This study investigates the effectiveness of reduction of training sets and kernel space for action decision using future prediction. For future prediction in a real environment, it is necessary to know the properties of the state and disturbance resulting from the outside environment, such as a ground surface or water surface. However, obtaining the properties of the disturbance depends on the specifications of the target processor, especially its sensor resolution or processing ability. Therefore, sampling-rate settings are limited by hardware specifications. In contrast, in the case of future prediction using machine learning, prediction is based on the tendency obtained from past training or learning. In such a situation, the learning time is proportional to training data. At worst, the prediction algorithm will be difficult to implement in real time because of time complexity.

Here, we consider the possibility of carefully analyzing the algorithm and applying dimensionality reduction techniques to accelerate the algorithm. In particular, to reduce the training sets and kernel space based on the recent tendency of disturbance or state, we focus on the use of the fast Fourier transform (FFT) and pattern matching. From this standpoint, we propose a method for dynamically reducing the dimensionality based on the tendency of disturbance. As a future application, an algorithm for operating unmanned agricultural support machines will be planned to implement the proposed method in a real environment.

KEYWORDS

Online SVR, Prediction and Control using State-action Pair, Dimensionality Reduction, Training Set

1 INTRODUCTION

For an action decision based on future prediction, it is necessary to know the properties of disturbance resulting from the outside environment [1]. However, obtaining the properties of the disturbance depends on the specifications of the target processor, especially its sensor resolution or processing ability. Therefore, sampling-rate settings are limited by hardware specifications. In contrast, in the case of future prediction using machine learning, prediction is based on the tendency obtained from past training or learning. In such a situation, the learning time is proportional to training data [1, 2].

Previously, a state-action pair prediction method had been proposed. In this method, the prediction performance [3] and action-decision methods [4, 5] had been considered based on some prediction results. These studies considered the behavior of a robot when an unknown periodic disturbance signal is continuously given to the robot. On the other hand, in these studies, the learning space had not been considered in action decision or future prediction. In general, a non-linear clustering (or regression, as in this work) kernel function has been used that allows the growth of the sup-

port vector machine (SVM) or support vector regression (SVR) solution, which starts invading other space; this "other space" is called the feature space. This allows us to change the information from one linear space to another one, permitting us to better classify (or perform regression analysis on) examples. However, the speed of learning depends mostly on the number of support vectors, which can significantly influence performances. Therefore, the complexity of learning is simply proportional to the size of training sets. If the recent tendency of disturbance or state, or these period can be obtained, the size of training sets can be reduced. Moreover, the length of training sets will be fixed even if a new training set is added.

Therefore, we consider the possibility of carefully analyzing the algorithm and applying dimensionality reduction techniques to accelerate the algorithm. In particular, to reduce the training sets and kernel space based on the recent tendency of disturbance or state, we focus on the use of the fast Fourier transform (FFT) and pattern matching. From this standpoint, we propose a method for dynamically reducing the dimensionality based on the tendency of disturbance.

The remainder of this paper is organized as follows. In section II, we present the approach for reducing a learning space (feature space) dynamically. Further, details on how to decide a learning space are presented based on the nearest-neighbor one-step-ahead forecasts and Nyquist-Shannon sampling theorem. In section III, the configuration of verification experiments is described. Section IV presents a summary of this work and identifies avenues for future work.

2 APPROACH FOR REDUCING THE LEARNING SPACE BASED ON THE FREQUENCY OF THE DISTURBANCE SIGNAL

2.1 Our Previous Work

In our previous study, we mentioned that, for controlling a robot in a dynamic environment, an action can be chosen based on the current result by predicting the future state using previous actions and states. In this paper, we consider that obtaining the optimal action is equivalent to minimizing the body pitch angle of an inverted pendulum when the predictive disturbance is continued using the prediction the state-action pair that had been proposed in our previous study [3]. Therefore, in this paper, we consider a system that decides the optimization action using the proposed method shown in Fig. 1 based on the previous study [4].

In Fig. 1, the system will apply optimal control using a gain $\mathbf{K}_f$ as the optimal feedback gain, and in parallel, it will decide the action to be taken in the future using the prediction of the state-action pair. In Fig. 1, ${}^{t-l}\hat{\mathbf{u}}(t+j)$ describes the prediction result of the control input $\mathbf{u}(t+j)$ when that input predicted in time $(t-l)$. Hence, the proposed method revises the current action by combining the optimal control and the result of the state-action pair prediction. The structure of prediction of the state-action pair is named "N-ahead state-action pair predictor," and the internal structure is shown in Fig. 2 [3]. The proposed method obtains a series of actions in time $(t+N)$ in the distant future from the current time t by using the N-ahead state-action pair predictor. From

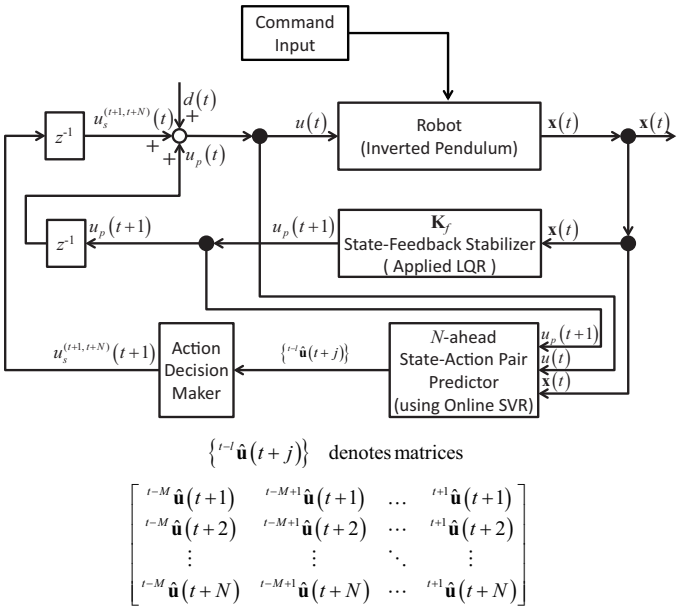


Figure 1. Outline of the decision of optimal action for a robot using the prediction of the state-action pair [5].

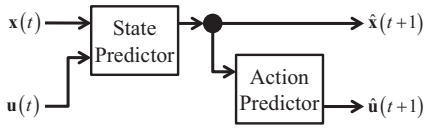


Figure 2. Outline of the prediction system of state and action [3].

this prediction series, the system will be able to revise the current action by combining "*the action that will be taken in the future*" and the prediction series of the action.

In this system, the optimal compensation control input $\mathbf{u}(t)$ is given as follows:

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{u}_s^{(t+1,t+N)}(t) + \mathbf{d}(t). \quad (1)$$

In this equation, $\mathbf{u}_p(t)$ denotes an optimal control action with optimal feedback control gain, $\mathbf{u}_s^{(t+1,t+N)}(t)$ is generated from the "action decision maker," and $\mathbf{d}(t)$ denotes an unknown periodic disturbance input signal. Then, $\mathbf{u}_s^{(t+1,t+N)}(t)$ can be defined as follows:

$$\mathbf{u}_s^{(t+1,t+N)}(t) = \sum_{i=1}^{N} \alpha_i \hat{\mathbf{u}}(t+i). \quad (2)$$

Moreover, in this study, the coefficient α_i is defined as follows:

$$\alpha_i = \frac{N+i-1}{100 \cdot N}. \quad (3)$$

From this technique, we create an action corresponding to a future time and obtain the optimal action that will be performed in the future.

2.2 Basic Idea

In section I, we mentioned the relationship between dimensionality reduction and the size of training sets for learning and predicting. From this viewpoint, we can reduce the training time and the size of the learning space for predicting future states and action if we can reduce the size of training sets.

SVR defines a "support vector," and the number of support vectors is less than the number of other training sets. In detail, the support vector describes the properties of an unknown function, eliminating the necessity for a precision training set for the unknown function. Thus, removing any training samples that are not relevant to support vectors might have no effect on the construction of the proper decision function [6]. In other words, in this study, the training sets can be reduced if an unknown periodic disturbance is applied to the plant model. In addition, a prediction model can be built for an almost-periodic disturbance signal if the support vectors can denote the properties of an almost-periodic function; that is, in the proposed method, the training sets will be reduced based on the recent tendency of disturbance or state, or the period of the disturbance signal. Moreover, the length of training sets will be fixed even if a new training set is added. In addition, the support vectors of "one period" of an almost-periodic disturbance will be used repeatedly. Therefore, in the proposed method, the prediction model can predict and adapt the plant if an unknown periodic disturbance is applied, similar to previous studies, in spite of dynamic dimensionality reduction based on the tendency of disturbance. However, this method provides prediction results under the assumption that "the given unknown periodic disturbance will not change in the future." Therefore, there will be a possibility that the compensation action will become inappropriate in the future if the tendency of disturbance is changed while behavioral correction is being made by the state-action pair prediction. Hence, this study focuses on the *difference* between the tendency of disturbance obtained by the prediction and the current tendency.

2.3 Estimation of the Frequency of a Disturbance Signal and Reduction of the Learning Space

As mentioned above, an unknown periodic signal will be used as a disturbance signal. Therefore, we will attempt to analyze the properties of the disturbance signal to reduce the learning space (Fig. 3). In this case, the disturbance signal will be represented with a tendency similar to a pattern.

From this property, the nearest-neighbor one-

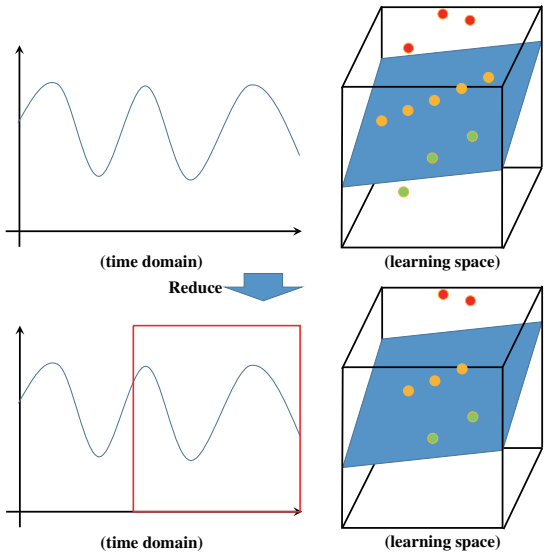


Figure 3. Outline of reduction of the learning space according to the period of the disturbance signal.

step-ahead forecasts [7] will be applied to detect the cycle period of a disturbance signal. Let us illustrate the nearest-neighbor one-step-ahead forecasts. As shown in Fig. 4, the target function $f(t)$ has a repeating tendency. In this case, we want to predict at time $t+1$ the next value of the series f. The pattern $f(t-12), f(t-6)$ is the most similar to the pattern. Then, the prediction will be calculated. Consequently, the nearest-neighbor one-

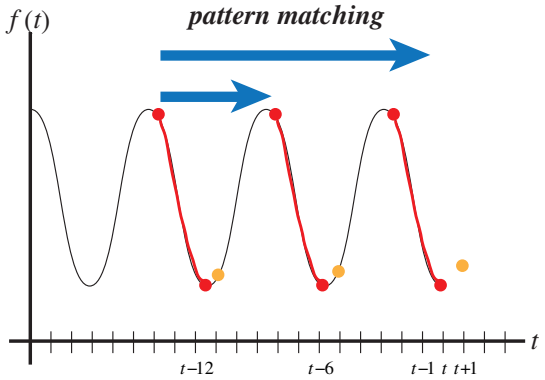


Figure 4. Outline of nearest-neighbor one-step-ahead forecasts.

step-ahead forecasts provide not only a one-step prediction result, but also the cycle period. From these results, the cycle period $T_{\text{disturbance}}$ is calculated as follows:

$$T_{\text{disturbance}} = |t_{\text{max, disturbance}} - t_{\text{min, disturbance}}| \times 2. \tag{4}$$

Here, $t_{\text{max, disturbance}}$ denotes the time when the maximum values of disturbance signal $d(t)$ have been reached, and $t_{\text{min, disturbance}}$ denotes the time when the minimum values of disturbance signal $d(t)$ have been reached.

On the other hand, as we have stated above, we attempt to analyze the properties of the disturbance signal to reduce the size of training sets. Therefore, the Nyquist-Shannon sampling theorem is applied to reduce the size of training sets and determine the properties of the disturbance signal. The theorem statement is as follows. *A sufficient sample rate is* $2B$ *samples/second or greater. Equivalently, for a given sample rate* f_s*, perfect reconstruction is guaranteed for a bandlimit* $B < f_s/2$. From this theorem, the sampling rate t'_s is defined as follows:

$$t'_s \leq \frac{T_{\text{disturbance}}}{2}, \tag{5}$$

Here, the sampling rate t'_s is an integral multiple of the original sampling rate t_s. Then, the result obtained by dividing t'_s by t_s is the training set that builds a prediction model. Therefore, new training sets are defined as follows:

$$N = \frac{t'_s}{t_s}, \tag{6}$$

$$S = \{s_{t-N}, s_{t-N+1}, \ldots, s_t\}. \tag{7}$$

In the above equations, S is a list of support sets.

The proposed method predicts events given in the training sets; however, it does not reduce former training sets. In this section, we describe how to implement the future prediction.

In this case, a next state $\hat{x}_{t+1,i}, i \in \dim \hat{\mathbf{x}}_{t+1}$ (i denotes an element of all the robot's states) is estimated using the state and action defined by $\mathbf{z}_t = \begin{bmatrix} x_{t,1} & \ldots & x_{t,n} & | & a_t \end{bmatrix}$. Therefore, the vector $\mathbf{z}_t$ is an $(n+1) \times 1$ vector. Next, let us consider the sum-of-squares error function J_S from the training set $\{\mathbf{x}_j, y_j\}$ described by the

SVR model $y(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$ [8].

$$J_S(\mathbf{w}) = \frac{1}{2}\sum_{j=t-N}^{t}\left\{\mathbf{w}^\top\phi\left(\mathbf{x}_j\right) + b - y_j\right\}^2 + \frac{\lambda}{2}\mathbf{w}^\top\mathbf{w} \quad (\lambda \geq 0), \quad (8)$$

where $\mathbf{w}^\top$ indicates the transpose of $\mathbf{w}$. Here, λ represents the regularization parameter, and $\mathbf{w}$ represents the weight matrix of the SVR model. The weight matrix $\mathbf{w}$ is determined by setting the gradient for minimizing the sum-of-squares error function J_S to zero (thus, $\partial J_S(\mathbf{w})/\partial\mathbf{w} = 0$). Hence,

$$\begin{aligned}\frac{\partial}{\partial\mathbf{w}}J_S(\mathbf{w}) =& 2\times\frac{1}{2}\sum_{j=t-N}^{t}\left[\left\{\mathbf{w}^\top\phi\left(\mathbf{x}_j\right) + b - y_j\right\}\right.\\ &\left.\phi\left(\mathbf{x}_j\right)\right] + \frac{\lambda}{2}\mathbf{w} + \frac{\lambda}{2}\mathbf{w} = 0,\\ 0 =& \sum_{j=t-N}^{t}\left[\left\{\mathbf{w}^\top\phi\left(\mathbf{x}_j\right) + b - y_j\right\}\right.\\ &\left.\phi\left(\mathbf{x}_j\right)\right] + \lambda\mathbf{w},\\ \mathbf{w} =& -\frac{1}{\lambda}\sum_{j=t-N}^{t}\left\{\mathbf{w}^\top\phi\left(\mathbf{x}_j\right) + b - y_j\right\}\phi\left(\mathbf{x}_j\right)\\ =& \sum_{j=t-N}^{t}a_j\phi\left(\mathbf{x}_j\right) = \Phi^\top\mathbf{a},\end{aligned} \quad (9)$$

$$\textit{where}\ \ \mathbf{a} = \begin{bmatrix}a_{t-N} & \ldots & a_t\end{bmatrix}^\top,$$

$$a_j = -\frac{1}{\lambda}\left\{\mathbf{w}^\top\phi\left(\mathbf{x}_j\right) + b - y_j\right\}.$$

Now, Φ is called the design matrix, and the j-th row is described by $\phi\left(\mathbf{x}_j\right)^\top$. Here, the parameter vector $\Phi\mathbf{a}$ replaces $\mathbf{w}$:

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^\top\Phi\Phi^\top\Phi\Phi^\top\mathbf{a} - \mathbf{a}^\top\Phi\Phi^\top\mathbf{y} + \frac{1}{2}\mathbf{y}^\top\mathbf{y} + \frac{\lambda}{2}\mathbf{a}^\top\Phi\Phi^\top\mathbf{a}. \quad (10)$$

Next, the Gramian matrix $\mathbf{K} = \Phi\Phi^\top$ will be defined. Here, the matrix coefficient of $\mathbf{K}$ is given by

$$K_{jm} = \phi\left(\mathbf{x}_j\right)^\top\phi\left(\mathbf{x}_m\right) = k\left(\mathbf{x}_j, \mathbf{x}_m\right) = Q_{jm}. \quad (11)$$

This matrix coefficient is the symmetric kernel matrix. Now, let us rearrange the sum-of-squares error function J_S by using the Gramian matrix:

$$J_S(\mathbf{a}) = \frac{1}{2}\mathbf{a}^\top\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}^\top\mathbf{K}\mathbf{y} + \frac{1}{2}\mathbf{y}^\top\mathbf{y} + \frac{\lambda}{2}\mathbf{a}^\top\mathbf{K}\mathbf{a}. \quad (12)$$

The equation is rearranged by isolating $\mathbf{a}$:

$$\mathbf{a} = \left(\mathbf{K} + \lambda\mathbf{I}_N\right)^{-1}\mathbf{y}. \quad (13)$$

Here, $\mathbf{I}_N$ represents the $N \times N$ identity matrix. Therefore, the prediction result $\hat{y}(\mathbf{x})$ for the SVR model to input $\mathbf{x}$ can be derived from the equation as

$$\begin{aligned}\hat{y}(\mathbf{x}) &= \mathbf{w}\phi(\mathbf{x}) + b = \mathbf{a}^\top\Phi\phi(\mathbf{x}) + b\\ &= \mathbf{k}(\mathbf{x})^\top\left(\mathbf{K} + \lambda\mathbf{I}_N\right)^{-1}\mathbf{y} + b\end{aligned} \quad (14)$$

$$\text{where}\quad \mathbf{k}(\mathbf{x}) = \begin{bmatrix}k\left(\mathbf{x}_{t-N}, \mathbf{x}\right) & \ldots & k\left(\mathbf{x}_j, \mathbf{x}\right)\end{bmatrix}^\top.$$

In this time, the prediction result and Kernel matrix are updated as follows:

$$\begin{aligned}\hat{y}(\mathbf{x}) &= \mathbf{w}\phi(\mathbf{x}) + b = \mathbf{a}^\top\Phi\phi(\mathbf{x}) + b\\ &= \mathbf{k}(\mathbf{x})^\top\left(\mathbf{K} + \lambda\mathbf{I}_N\right)^{-1}\mathbf{y} + b\end{aligned} \quad (15)$$

$$\text{where}\quad \mathbf{k}(\mathbf{x}) = \begin{bmatrix}k\left(\mathbf{x}_{t-N}, \mathbf{x}\right) & \ldots & k\left(\mathbf{x}_t, \mathbf{x}\right)\end{bmatrix}^\top$$

$$Q = \begin{bmatrix}Q_{s_{t-N},s_{t-N}} & \cdots & Q_{s_{t-N},s_t}\\ \vdots & \ddots & \vdots\\ Q_{s_t,s_{t-N}} & \cdots & Q_{s_t,s_t}\end{bmatrix}. \quad (16)$$

The matrix Q contains the values of the kernel function and is called the kernel matrix. In these equations, the learning space Q and training sets $\mathbf{x}$ will be re-constructed at each sampling time by adding new training data. Therefore, the learning space and training sets are reduced at each sampling time. As mentioned before, the speed of learning depends mostly on the number of support vectors, which can significantly influence performance. Consequently, the speed of learning will be improved over that in previous studies.

Now, let us consider the tendency of disturbance change. When applying state-action pair prediction, it is possible to obtain prediction results at each sampling time. The prediction results are regarded as time-series data,

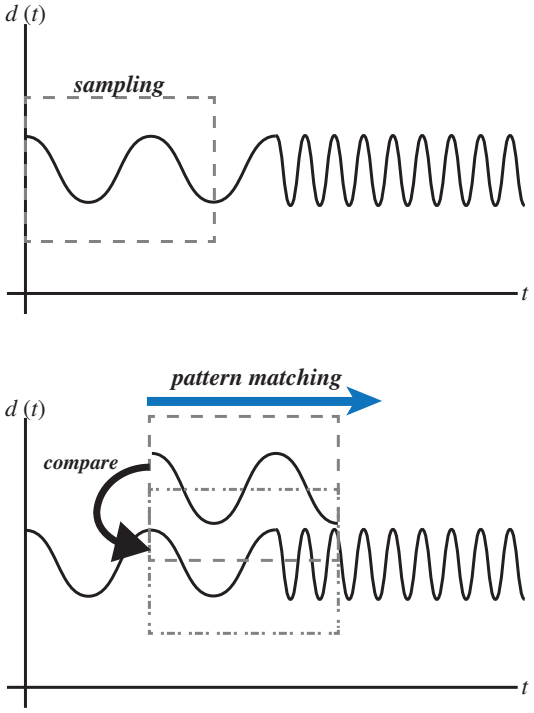


Figure 5. Comparison of difference of disturbances.

and its period is calculated (as shown in Fig. 5). Firstly, the input prediction values for control input $\mathbf{u}(t+j)$ using the nearest-neighbor one-step-ahead forecasts are calculated to obtain the period $T_{cycle,prediction}$ and amplitude $A_{cycle,prediction}$ of prediction values of $\mathbf{u}(t+j)$ at each time. Secondly, $T_{cycle,current}$ and amplitude $A_{cycle,current}$ at time $(t+j)$ are obtained. Using this information, the following equations are solved to determine whether the tendency of disturbance will change.

$$|T_{cycle,prediction} - T_{cycle,current}| \leq \epsilon, \quad (17)$$
$$|A_{cycle,prediction} - A_{cycle,current}| \leq \epsilon. \quad (18)$$

These equations indicate that there will be no tendency change if the period or amplitude is ϵ or less. Else, the tendency of disturbance will change; moreover, the proposed method will learn and control new changes in combination with a linear-quadratic regulator (LQR).

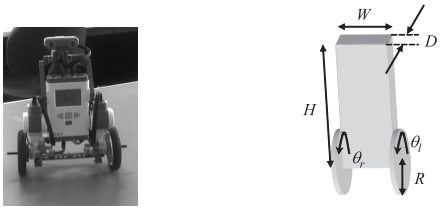
3 VERIFICATION EXPERIMENT – COMPUTATIONAL SIMULATION USING THE PROPOSED METHOD

3.1 Outline of the Experiment

In this experiment, we stabilize the posture of a two-wheeled self-propelled inverted pendulum "NXTway-GS" (Fig. 6) as an application by using the computer simulation. In this verification experiment, we compared the control response of the proposed method with that of the conventional method. Furthermore, in the proposed method, for postural control, the predictor only used the proximate prediction result obtained by repeatedly training data from 0 [s] (no reduction) or training sets reduced at each sampling time.

3.2 Simulation Setup – the NXTway-GS Model

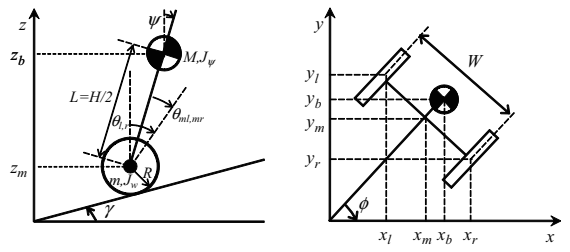
NXTway-GS (Fig. 6) can be considered an inverted pendulum model, as shown in Fig. 7. Figure 7 shows a side view and plane view of the model. The coordinate system used in 3.3 is described in Fig. 7. In figure 7, ψ denotes the body pitch angle, and $\theta_{ml,mr}$ denotes the DC motor angle (l and r indicate *left* and *right*, respectively). The physical parameters of NXTway-GS are listed in Table 1.



(a) NXTway-GS

(b) Model of NXTway-GS

Figure 6. Two-wheeled inverted pendulum "NXTway-GS."



(a) Side view (b) Plane view

Figure 7. Side view and Plane View of NXTway-GS [10]-[11].

3.3 Simulation Setup – Modeling of NXTway-GS

We can derive the equations of motion of the inverted pendulum model using the Lagrange equation based on the coordinate system in Fig. 7. If the direction of the model is in the positive x-axis direction at $t = 0$, the equations of motion for each coordinate are given as ([10]-[11])

$$\left[(2m + M) R^2 + 2J_w + 2n^2 J_m\right] \ddot{\theta} + (MLR - 2n^2 J_m) \ddot{\psi} - Rg (M + 2m) \sin\gamma = F_\theta \tag{19}$$

$$\left(MLR - 2n^2 J_m\right) \ddot{\theta} + \left(ML^2 + J_\psi + 2n^2 J_m\right) \ddot{\psi} - MgL\psi = F_\psi \tag{20}$$

$$\left[\frac{1}{2} mW^2 + J_\phi + \frac{W^2}{2R^2} \left(J_w + n^2 J_m\right)\right] \ddot{\phi} = F_\phi \tag{21}$$

Here, we consider the variables $\mathbf{x}_1, \mathbf{x}_2$ as the state variables and $\mathbf{u}$ as the input variable ($\mathbf{x}^\top$ indicates the transpose of $\mathbf{x}$).

$$\mathbf{x}_1 = \begin{bmatrix}\theta & \psi & \dot{\theta} & \dot{\psi}\end{bmatrix}^\top \tag{22}$$

$$\mathbf{x}_2 = \begin{bmatrix}\phi & \dot{\phi}\end{bmatrix}^\top \tag{23}$$

$$\mathbf{u} = \begin{bmatrix}v_l & v_r\end{bmatrix}^\top \tag{24}$$

Consequently, we can derive the state equations of the inverted pendulum model from Eq. (19), (20), and (21).

$$\frac{d}{dt}\mathbf{x}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1\mathbf{u} + \mathbf{S} \tag{25}$$

$$\frac{d}{dt}\mathbf{x}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2\mathbf{u} \tag{26}$$

In this paper, we only use the state variable $\mathbf{x}_1$. Because $\mathbf{x}_1$ includes the body pitch angle as important variables ψ and $\dot{\psi}$ for the control of self-balancing, we will not consider plane motion ($\gamma_0 = 0, \mathbf{S} = \mathbf{0}$).

3.4 Simulation Setup – How to Apply the Online SVR to the State Predictor

In this method, we use Online SVR [9] as the learner. Moreover, we applied the RBF kernel [13] as the kernel function for Online SVR. The RBF kernel on two samples $\mathbf{x}$ and $\mathbf{x}'$, represented as feature vectors in some input space, is defined as

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\beta \left|\left|\mathbf{x} - \mathbf{x}'\right|\right|^2\right). \tag{27}$$

The learning parameters of Online SVR are listed in Table 2. In table 2, $i \in \{1, 2, 3, 4\}$.

3.5 Simulation Setup – How to Apply the Linear-quadratic Regulator to the Action Predictor

In this experiment, we apply an LQR for action prediction (and a predictor). Therefore, we design the controller as an action predictor based on modern control theory. This LQR calculates the feedback gain $\mathbf{k}_f$ so as to minimize the cost function J_C given as follows:

$$J_C = \int_0^\infty \left[\mathbf{x}^\top(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^\top(t)\mathbf{R}\mathbf{u}(t)\right] dt. \tag{28}$$

The tuning parameter is the weight matrix for the state $\mathbf{Q}$ and input $\mathbf{R}$. In this paper, we choose the following weight matrix $\mathbf{Q}$ and $\mathbf{R}$:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 6 \times 10^5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \times 10^2 \end{bmatrix} \tag{29}$$

$$\mathbf{R} = 1 \times 10^3 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{30}$$

Then, we obtain the feedback gain $\mathbf{k}_f$ by minimizing J_C. Therefore, we apply $\mathbf{k}_f$ as an action predictor [3]. Hence, in this experiment, we do not consider the plane movement of

Table 1. Physical parameters of NXTway-GS

Symbol	Value	Unit	Physical property
g	9.81	[m/s^2]	Gravity acceleration
m	0.03	[kg]	Wheel weight [10]
R	0.04	[m]	Wheel radius
J_w	$\frac{mR^2}{2}$	[kgm^2]	Wheel inertia moment
M	0.635	[kg]	Body weight [10]
W	0.14	[m]	Body width
D	0.04	[m]	Body depth
H	0.144	[m]	Body height
L	$\frac{H}{2}$	[m]	Distance of center of mass from wheel axle
J_ψ	$\frac{ML^2}{3}$	[kgm^2]	Body pitch inertia moment
J_ϕ	$\frac{M(W^2+D^2)}{12}$	[kgm^2]	Body yaw inertia moment
J_m	1×10^{-5}	[kgm^2]	DC motor inertia moment [11]
R_m	6.69	[Ω]	DC motor resistance [12]
K_b	0.468	[V·s/rad.]	DC motor back EMF constant [12]
K_t	0.317	[N·m/A]	DC motor torque constant [12]
n	1	[1]	Gear ratio [11]
f_m	0.0022	[1]	Friction coefficient between body and DC motor [11]
f_W	0	[1]	Friction coefficient between wheel and floor [11]

Table 2. Learning parameters of Online SVR

Symbol	Value	Property
C_i	300	Regularization parameter or predictor of x_i
ϵ_i	0.02	Error tolerance for predictor of x_i
β_i	30	Kernel parameter for predictor of x_i

the two-wheeled inverted pendulum. In other words, we consider that $\phi = 0, \theta_{ml} = \theta_{mr}$, and $\mathbf{u} = u, \mathbf{d}(t) = d(t)$.

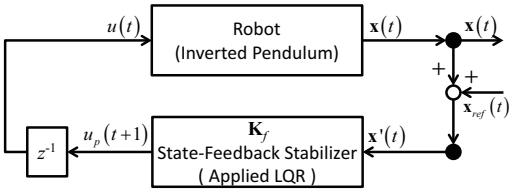


Figure 8. Control input obtained by mixing the action and disturbance inputs.

3.6 Conditions of Simulation – Acquiring the Training Sets

In this experiment, we mix the action signal with a known disturbance signal $\mathbf{d}(t)$ (Figs. 8and 9), where $\mathbf{d}(t)$ is given as

$$d(t) = \begin{cases} 2\sin\left(\frac{1}{2}\pi t\right) & \text{if } t \leq 12[\text{s}] \\ \frac{1}{2}\sin\left(2\pi t\right) & \text{if } t > 12[\text{s}] \end{cases} \tag{31}$$

Then, the signal $\mathbf{d}(t)$ is mixed in the model.

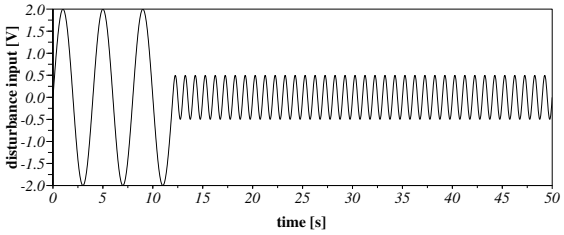


Figure 9. Disturbance signal in control inputs $\mathbf{d}(t)$.

Thereby, we can acquire the training sets from the two-wheeled inverted pendulum. Figures 10 to 12 show training sets that were obtained from the computer simulation of the stabilization control of the two-wheeled inverted pendulum. The properties of disturbance that we provide as input and the other conditions of the simulation are listed in Table 3.

3.7 Simulation Results

Figures 10 and 11 show compensation results of the state of $\mathbf{x}_1$, and Fig. 12 shows the prediction of the control input and compensation input using the prediction result of $\mathbf{u}$.

In this section, we will not consider the part

Table 3. Parameters for the simulation

Sybmol	Value	Unit	Physical property
ψ_0	0.0262	[rad.]	Initial value of body pitch angle
γ_0	0.0	[rad.]	Slope angle of movement direction
t_s	0.05	[s]	Sampling rate
$t_{d,\text{start}}$	0.0	[s]	Start time of application of predictable disturbance
$t_{d,\text{finish}}$	45.0	[s]	Finish time of application of predictable disturbance
N_s	60	—	Initial dataset length
$N_{\max}$	950	—	Maximum dataset length for the prediction
N	30	—	Step size of outputs for N-ahead state-action pair predictor's outputs

that is given in real training sets. Thus, we will only argue and focus on the part of the graph pertaining to the state-predicted part shown in T (at $t = 3.00$ [s]) of Figs. 10 and 11.

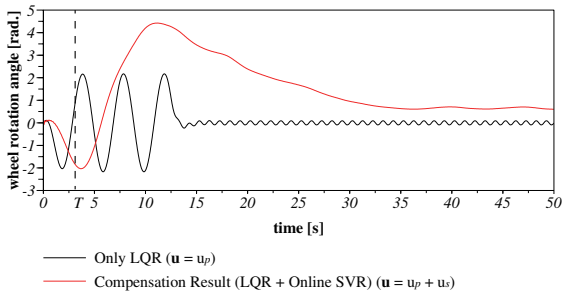


Figure 10. Control response of the wheel rotation angle θ.

3.8 Discussion of Simulation Results

Here, the start and prediction of the state-predicted point are shown at $t = 3.00$ [s] in T.

According to these results (Figs. 10 through 12), compensation results obtained using the proposed method (shown by the red solid line) approach zero with time. Next, we focus on each result.

In this study, the learning space and training sets are changed "suddenly," when the period of the disturbance signal is estimated. At

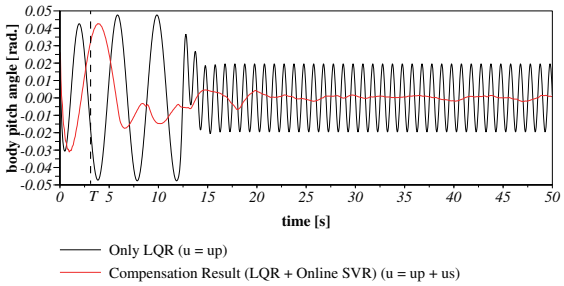


Figure 11. Control response of the body pitch angle ψ.

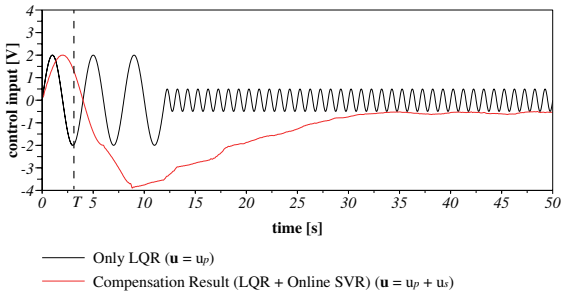


Figure 12. Control response of the control input u.

this time, the current action and states are also suddenly changed by the compensation control input based on the former learning space and training sets. In a previous study, the training sets and learning space included significant prediction error in the early phase of training. Therefore, the state-action pair prediction uses former training sets and adds new training sets to the current learning space. That is, the results of prediction and revised action are influenced by the past prediction error. On the other hand, in the proposed method, the kernel matrix ignores early learning results and early compensation results. In other words, it uses results that are less affected by prediction error, and the time elapsed by the training set out of the disturbance tendency period is not used. Further, this system acquires the data at each sampling time. By using these changed results, the proposed system derives an action that multiplies states with the optimal feedback gain for the future state. Consequently, this system stabilizes the inverted pendulum by using current outside data, previous states, and an action. Consequently, it can be said that the internal states and action will converge to zero with time. From these viewpoints, we conclude that

the experimental results are reasonable.

4 CONCLUSION

In this paper, we investigated the relationship between the learning space and training sets for prediction and the frequency of the disturbance signal given by the outside environment. On the basis of our previous studies, we proposed a method to reduce training sets and learning space dynamically for prediction based on prediction results obtained from the recent tendency of disturbance frequency by using the Nyquist-Shannon sampling theorem. When the proposed method was applied, the body pitch angle of NXTway-GS converged to zero with time, despite a tendency change. In other words, the compensated action for rapid convergence was obtained, similar to the conventional method of training sets and learning space.

From the results of verification experiments, the proposed method could be converged to a desirable state in a manner similar to fixed training sets. More specifically, the slope of the body pitch angle of NXTway-GS converged to zero based on the state and action prediction and decision. Accordingly, we conclude that the proposed method can be adapted to any disturbance frequency. From the results of verification experiments, it can be concluded that the proposed system can predict what be defined by training sets and learning space that can be obtained the properties of a disturbance signal based on the Nyquist-Shannon sampling theorem. In addition, as future work, we will confirm the response of the proposed system for actual robots such as a crawler-type robot. Moreover, the proposed method will be extended to plane motion to extend the dimensions of output prediction. As a future application, an algorithm for operating unmanned agricultural support machines will be planned to implement the proposed method in a real environment.

REFERENCES

[1] P. Pivoňka, V. Veleba, M. Šeda, P. Ošmera, and R. Matoušek, "The Short Sampling Period in Adaptive Control," In Proceedings of the World Congress on Engineering and Computer Science 2009, Vol II WCECS, pp.724-729, 2009.

[2] A. L. Blum, and P. Langley, "Selection of Relevant Features and Examples in Machine Learning," Artificial Intelligence, Vol. 97 No. 1-2, pp.245-271, 1997.

[3] M. Sugimoto, and K. Kurashige, "A Study of Effective Prediction Methods of the State-action Pair for Robot Control using Online SVR," Journal of Robotics and Mechatronics, Vol. 27 No. 5, pp.469-479, 2015.

[4] M. Sugimoto, and K. Kurashige, "Real-time Sequentially Decision for Optimal Action using Prediction of the State-Action Pair," In Proceedings of 2014 International Symposium on Micro-NanoMechatronics and Human Science, pp.199-204, Nov.9-12, Nagoya, Japan, 2014.

[5] M. Sugimoto, and K. Kurashige, "Future Motion Decisions using State-action Pair Predictions," International Journal of New Computer Architectures and their Applications, Vol. 5 No. 2, pp.79-93, 2015.

[6] R. Koggalage, and S. Halgamuge, "Reducing the Number of Training Samples for Fast Support Vector Machine Classification," Neural Information Processing – Letters and Reviews, Vol. 2 No. 3, pp.57-65, 2004.

[7] G. Bontempi, Machine Learning Strategies for Time Series Prediction, Machine Learning Summer School, 2013. [Online]. Available: http://www.ulb.ac.be/di/map/gbonte/ftp/time_ser.pdf

[8] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, 2006.

[9] F. Parrella, Online Support Vector Regression. PhD thesis, Department of Information Science, University of Genoa, Italy, 2007.

[10] M. Sugimoto, H. Yoshimura, T. Abe, and I. Ohmura, "A Study on Model-Based Development of Embedded System using Scilab/Scicos," In Proceedings of the Japan Society for Precision Engineering 2010 Spring Meeting, Saitama, D82, pp.343-344, 2010.

[11] Y. Yamamoto, NXTway-GS Model-Based Design –Control of self-balancing two-wheeled robot built with LEGO Mindstorms NXT–. Cybernet Systems Co., Ltd., 2009.

[12] R. Watanabe, Ryo's Holiday LEGO Mindstorms NXT, 2008.

[13] Y. Chang, C. Hsieh, K. Chang, M. Ringgaard, and C. Lin, "Training and Testing Low-degree Polynomial Data Mappings via Linear SVM," J. Machine Learning Research, Vol. 11, pp.1471-1490, 2010.

International Journal of New Computer Architectures and their Applications (IJNCAA) 7(1): 29-36
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Application of Continuous Genetic Algorithms for Optimization of Logistic Networks Governed by Order-Up-To Inventory Policy

Przemysław Ignaciuk and Łukasz Wieczorek
Institute of Information Technology, Lodz University of Technology
215 Wólczańska St., 90-924 Łódź, Poland
przemyslaw.ignaciuk@p.lodz.pl, 801144@edu.p.lodz.pl

ABSTRACT

The paper addresses the optimization problem of goods distribution process in logistic networks. The controlled nodes in the considered class of networks form a mesh structure. The stock level at the nodes is replenished from external sources and other nodes in the controlled network. The external demand is imposed on any node without prior knowledge about the requested quantity. Inventory control is realized through the application of order-up-to policy implemented in a distributed way. The aim is to provide high customer satisfaction while minimizing the total holding costs. In order to determine the optimal reference stock level for the policy operation a continuous genetic algorithm is applied and adjusted for the analyzed class of application centered problems.

KEYWORDS

Logistic networks, order-up-to policy, optimization, continuous genetic algorithm, inventory management.

1 INTRODUCTION

The optimization of goods flow in logistic networks is a computationally challenging task. For this reason, the related research mainly focuses on simple network structures and topologies. The complex mathematical dependencies and delayed interaction of system components (e.g., in a practical system the goods cannot be transferred immediately among the nodes) make the numerical analysis of multi-node networks resource prohibitive. In particular, determination of the cost (or fitness) function is time consuming. Moreover, the presence of nonlinearities may lead to many local minima. In the scientific literature, the optimization of logistic systems is examined mainly in the case of fundamental configurations, e.g., when each internal node has only one goods supplier [1]. The most common types of such structures are:

- single-echelon systems [2, 3] – where a separate external provider is connected to each controlled node;
- serial interconnection [4, 5] – in which all the nodes are connected to each other in sequence;
- tree-like organization [6–8] – wherein the flow of goods proceeds along parallel paths and the external demand is placed at a focal point.

These studies are not sufficient for the current logistic systems, where the actually deployed architectures are much more complex. One may argue that, nowadays, the general availability of powerful computing machines creates new opportunities for solving realistic optimization problems. However, due to curse of dimensionality, performing extensive numerical treatment becomes possible only when an efficient method is selected, e.g., within the evolutionary computation domain [9].

The purpose of this paper is to evaluate the usefulness of genetic algorithms (GAs) in the optimization of logistic network performance when subjected to the control of the classical – order-up-to (OUT) [10] – inventory policy. The research is focused on the practical case of a system with mesh-type topology. In the analyzed structure type, a particular node – connected to multiple nodes – may play the role of supplier and goods provider to effectuate the stock replenishment decisions. The decisions are taken according to the indications of the OUT policy, deployed in a distributed way (independently at each node). The optimization objective is to determine the reference stock level (RSL) for individual nodes so that the holding costs in the

entire system are minimized while at the same time a given service level is maintained.

Since the considered problem has a continuous search domain, applying the generic form of a GA would require translating the system variables (and associated operations) into the binary domain. Therefore, unlike the typical GA binary-value implementation, a continuous search space is used. Moreover, as opposed to the standard GA tuning procedures, proposed for the "artificial" optimization problems where multiple cost function evaluations are permissible [11], the long time of obtaining the fitness function value in the considered class of systems shifts the GA tuning effort towards constraining the number of iterations. The effectiveness of GA in reaching the optimal network state is evaluated in strenuous simulations.

2 SYSTEM DESCRIPTION

2.1 Actors in Logistic Processes

The paper analyzes the process of goods distribution among the nodes (warehouses, stores, etc.) of a logistic network. The nodes are connected with each other with mesh topology permitted. Each connection is characterized by two attributes:

- delivery delay time (DDT) – the time from issuing an order for goods acquisition until their delivery to the ordering node;
- supplier fraction (SF) – the percentage of ordered quantity to be retrieved from a particular source selected by the ordering node from its neighbors in the controlled network or external suppliers.

Apart from the initial stock at the nodes, the main source of goods in the network are the external suppliers. There are no isolated nodes that would not be linked to any other controlled node or external supplier, neither the nodes that would supply the stock for themselves. In addition, there is a finite path from each controlled node to at least one external source, which means that the network is connected. The system driving factor is the external demand imposed on the controlled nodes. The demand can be placed at any node and, as in the majority of practical cases [10, 12], its future value is not known precisely at the moment of issuing an order. The business objective is to ensure high customer satisfaction through fulfilling the external demand, at the same time avoiding unnecessary increase of the operational costs. Thus, the optimization purpose is to obtain a high service level at the lowest possible cost of goods storage at the nodes, i.e., minimizing the total network holding cost (*HC*).

2.2 Actor Interaction

The considered logistic network consists of N nodes n_i, where index $i \in \Theta_N = \{1, 2, \ldots, N\}$, and M external sources m_j, where $j \in \Theta_M = \{1, 2, \ldots, M\}$. The set containing all the indices $\Theta = \{1, 2, \ldots, N + M\}$. Let $l_i(t)$ denote the on-hand stock level (the quantity of goods currently stored) and $d_i(t)$ the external demand imposed on node i in period t, $t = \{0, 1, 2, \ldots, T\}$, T being the optimization time span. The connection between two nodes i and j is unidirectional, characterized by two attributes $(\alpha_{ij}, \gamma_{ij})$, where:

- α_{ij} – the SF between nodes i and j, $\alpha_{ij} \in [0, 1]$;
- γ_{ij} – the DDT between nodes i and j, $\gamma_{ij} \in [1, \Gamma]$, where Γ denotes the maximum DDT between any two directly interconnected nodes.

Fig. 1 illustrates the operation sequence at a network node occurring in each period.

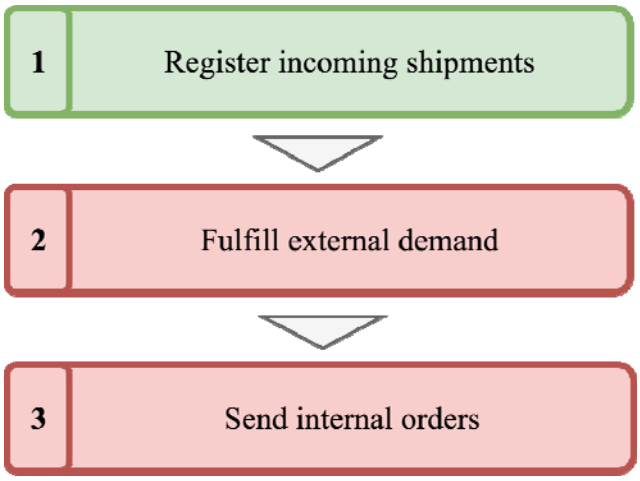


Figure 1. Node operational sequence.

Detailed mathematical description of node interaction is given in [12]. Below, only the fundamental issues required for the algorithm implementation are covered.

Let us introduce variables:

- $\Omega_i^S(t)$ – amount of goods sent by node *i* in period *t*,
- $\Omega_i^R(t)$ – amount of goods received by node *i* in period *t*.

The stock level at node *i* evolves according to

$$l_i(t+1) = \left(l_i(t) + \Omega_i^R(t) - d_i(t)\right)^+ - \Omega_i^S(t), \quad (1)$$

where $(f)^+$ denotes the saturation function $(f)^+ = \max\{f, 0\}$. The satisfied external demand $s_i(t)$ at node *i* in period *t* (the goods actually sold to the customers) may be expressed as

$$s_i(t) = \min\left\{l_i(t) + \Omega_i^R(t), d_i(t)\right\}. \quad (2)$$

Consequently, (1) may be rewritten as

$$l_i(t+1) = l_i(t) + \Omega_i^R(t) - s_i(t) - \Omega_i^S(t). \quad (3)$$

Let $o_i(t)$ denote the total quantity of goods ordered by node *i* in period *t*. $o_i(t)$ covers the orders to be fulfilled by other controlled nodes as well as external sources. Then, the quantity sent by node *i* in period *t* in response to the orders from its neighbors

$$\Omega_i^S(t) = \sum_{j\in\Theta_N} \alpha_{ij}(t) o_j(t). \quad (4)$$

On the other hand, the quantity of goods received by node *i* in period *t* from all its suppliers

$$\Omega_i^R(t) = \sum_{j\in\Theta} \alpha_{ji}(t-\gamma_{ji}) o_i(t-\gamma_{ji}). \quad (5)$$

The nodes try to answer both the external and internal demand. In case of insufficient stock to fulfill all the requests, the ordered quantity is reduced accordingly, yet

$$\forall_i \; 0 \le \sum_{j\in\Theta} \alpha_{ji}(t) \le 1. \quad (6)$$

When a node receives a request from another controlled node in the network and have accumulated enough resources to fulfill it, then $\alpha_{ij}(t) = \alpha_{ij}$. Otherwise, $\alpha_{ij}(t) < \alpha_{ij}$. It is assumed that the external sources are able to satisfy every order originating from the network (they are not restricted by capacity limitation).

2.3 State-Space Description

For the purpose of convenience of further study, a state-space model of the considered network will be introduced. The dynamic dependencies can be grouped into

$$\begin{aligned}\boldsymbol{l}(t+1) = \boldsymbol{l}(t) + \sum_{\gamma=1}^{\Gamma} \boldsymbol{M}_{\boldsymbol{\gamma}}(t-\gamma)\boldsymbol{o}(t-\gamma) \\ + \boldsymbol{M}_{\boldsymbol{0}}(t)\boldsymbol{o}(t) - \boldsymbol{s}(t),\end{aligned} \quad (7)$$

where:

- $\boldsymbol{l}(t)$ – vector of stock levels

$$\boldsymbol{l}(t) = \left[l_1(t), l_2(t), ..., l_N(t)\right]^T, \quad (8)$$

- $\boldsymbol{o}(t)$ – vector of replenishment orders

$$\boldsymbol{o}(t) = \left[o_1(t), o_2(t), ..., o_N(t)\right]^T, \quad (9)$$

- $\boldsymbol{s}(t)$ – vector of satisfied demands

$$\boldsymbol{s}(t) = \left[s_1(t), s_2(t), ..., s_N(t)\right]^T, \quad (10)$$

- $\boldsymbol{M}_{\boldsymbol{\gamma}}(t)$ – matrices specifying the node interconnections; for each $\gamma \in [1, \Gamma]$,

$$\boldsymbol{M}_{\boldsymbol{\gamma}}(t) = \begin{bmatrix} \sum_{i\Gamma_{i1}=\gamma} \alpha_{i1}(t) & 0 & \cdots & 0 \\ 0 & \sum_{i\Gamma_{i2}=\gamma} \alpha_{i2}(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sum_{i\Gamma_{iN}=\gamma} \alpha_{iN}(t) \end{bmatrix}, \quad (11)$$

- $\boldsymbol{M}_{\boldsymbol{0}}(t)$ – matrix describing the stock depletion due to internal shipments

$$\boldsymbol{M_o}(t) = -\begin{bmatrix} 0 & \alpha_{12}(t) & \cdots & \alpha_{1N}(t) \\ \alpha_{21}(t) & 0 & \cdots & \alpha_{2N}(t) \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{N1}(t) & \alpha_{N2}(t) & \cdots & 0 \end{bmatrix}. \quad (12)$$

2.4 OUT Inventory Policy

One of the popular stock replenishment strategies applied in logistic systems is the OUT inventory policy. This policy attempts to elevate the current stock to a predefined reference one. A replenishment order is issued if the sum of the on-hand stock level and goods quantity from pending orders at a node is below the reference level. The reference level should be set so that high percentage of the external demand is satisfied, yet excessive stock accumulation is avoided. The network optimization procedures discussed in this paper provide guidelines for the reference stock level selection under uncertain demand (the future demand is not known exactly while issuing the stock replenishment orders). The operational sequence of the OUT policy is presented in Fig. 2.

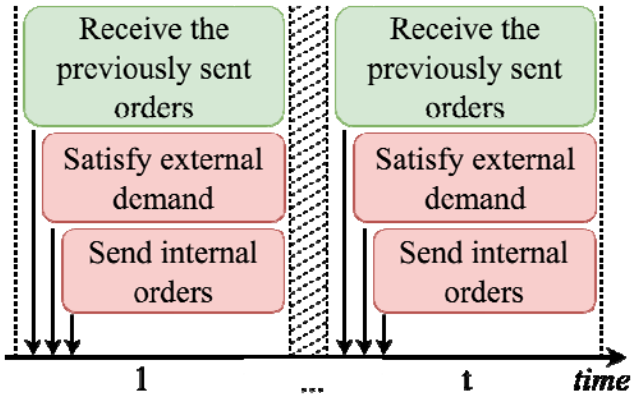


Figure 2. OUT policy operational sequence.

According to [10], the quantity of the replenishment order placed by node i in period t may be calculated as

$$o_i(t) = l_i^r - l_i(t) - \Phi_i(t), \quad (13)$$

where:

- l_i^r – the reference stock level set for node i, $i \in [1, N]$,
- $\Phi_i(t)$ – the amount of goods from pending orders issued by node i (the orders already placed by not yet realized due to lead-time delay).

In order to allow for efficient implementation of the optimization procedures variables (13) are represented in a vector form:

$$\boldsymbol{o}(t) = \boldsymbol{l}^r - \boldsymbol{l}(t) - \sum_{k=1}^{\Gamma} \sum_{\gamma=t-k}^{t-1} \boldsymbol{M}_i(\gamma)\boldsymbol{o}(\gamma), \quad (14)$$

where $\boldsymbol{l}^r$ groups the reference stock levels for all the nodes.

Any logistic system should retain a high service level despite imprecise knowledge about the demand future evolution. This objective is quantified here through the fill rate, i.e., the percentage of actually realized customer demand imposed on all the nodes. The goal of the optimization procedure is to indicate a reference stock level for each node so as to preserve the lowest possible *HC* while keeping the fill rate close to a predefined one – ideally 100%. As a first approximation, using only the knowledge about the highest expected demand in the system $\boldsymbol{d}_{\max}$, the 100% fill rate is obtained if the reference stock level is selected according to the following formula [12]

$$\boldsymbol{l}^r = \left(\boldsymbol{I}_N + \sum_{\gamma=1}^{\Gamma} \gamma \boldsymbol{M}_\gamma \right) \boldsymbol{M}^{-1} \boldsymbol{d}_{\max}, \quad (15)$$

where $\boldsymbol{I}_N$ is an $N \times N$ identity matrix.

3 GENETIC ALGORITHM

The key factor behind cost-efficient operation of the OUT policy is proper selection of RSL. In the analyzed class of systems it should be done simultaneously for all the controlled nodes, which is challenging due to coupled relationships in the mesh structure. GA enables automation of this process through numerical adjustments.

The continuous domain of the search space indicates direct GA application, i.e., without the typical conversion to the binary form [11]. With respect to the GA terminology, the RSL in a given node reflects the allele and the candidate solution (individual) is represented by the vector containing the RSLs of all the controlled nodes. A particular population comprises a set of such vectors. The genotypes of each individual correspond to the phenotypes of RSLs. Fig. 3

outlines the algorithm operational sequence. Its individual steps are described in latter sections.

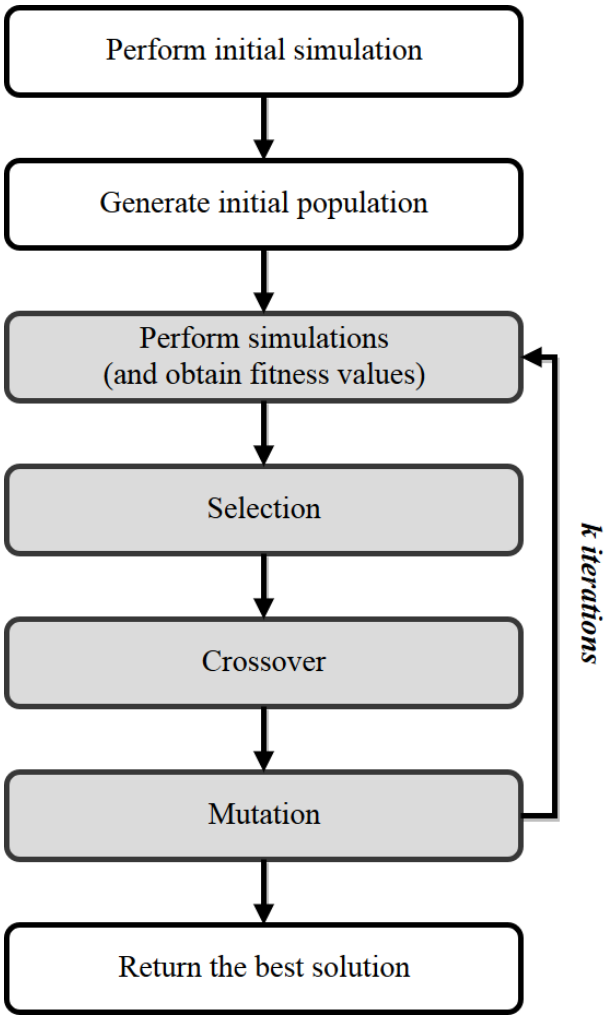


Figure 3. GA flowchart.

3.1 Initialization

The first two steps of the flowchart presented in Fig. 3 constitute the initialization phase of the optimization process. First, the RSLs are calculated according to formula (15). In the initial simulation run it is assumed that at each node a persistent external excitation (demand) is exerted during the entire simulation time equal to the maximum estimate for this node. The maximum holding cost $HC_{\max}$ and RSLs are determined accordingly and set as the boundary value for further calculations. Although the initial RSL ensures full customer satisfaction, the holding cost is very high – an excessive amount of goods is stored. The aim of further phases of the optimization process is to reduce *HC* while maintaining a high customer satisfaction rate.

3.2 Fitness function

A good choice of the fitness function is fundamental for GA operation and its efficiency in solving optimization problems. This function specifies dependencies between the individual system components and their relevance for the ensuing solution. It allows one to relate a particular solution to the expected optimum and influences the formation of successive populations. A well-defined fitness function should be normalized and efficient to compute. In the considered type of networks, two factors have a decisive impact on the solution:

- HC – holding cost, $HC \in [0, HC_{\max}]$,
- FR – fill rate, $FR \in [0, 1]$.

HC denotes the total cost of storing goods in all the nodes throughout the simulation interval. *FR* provides a percentage value of how the logistic network has succeeded in fulfilling the customer demand. The goal of the optimization process is to minimize the total holding cost of the network while maintaining the highest possible fill rate. For this reason, the following fitness function has been applied to the algorithm:

$$Fitness = \left(1 - \frac{HC}{HC_{initial}}\right)^{\varphi} FR^{\beta}, \quad (16)$$

where φ and β are tuning parameters which allow one to investigate the impact of prioritizing cost reductions vs. customer satisfaction in finding the optimal solution. The fitness function enables one to measure how well adjusted is a candidate solution and compare one to another.

3.4 Selection

The selection operation in the considered GA is realized using roulette-wheel approach, also called fitness proportionate selection. It is illustrated in Fig. 4. Selection is performed once the fitness value of all the individuals in a population have been calculated. The fitness value is needed to determine the probability of choosing each chromosome in the recombination process. From the computational perspective, each individual has assigned a fraction within the

range [0, 1] proportional to its fitness value relative to the rest of the current generation. Using a random selector, the entire population is divided into pairs.

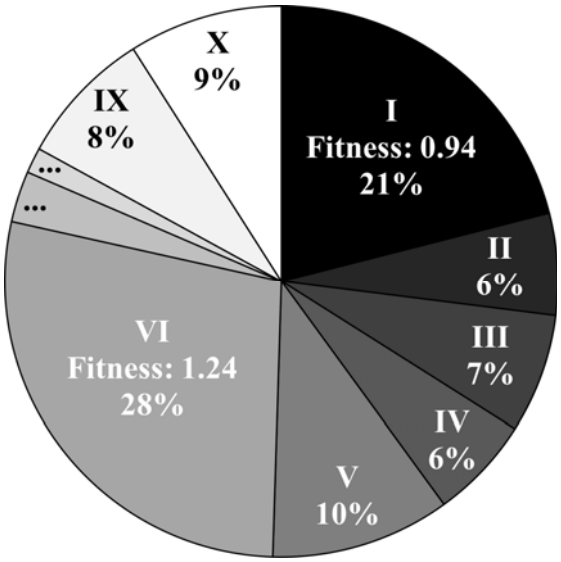


Figure 4. Roulette-wheel for a particular generation.

3.3 Crossover

The crossover operation enables evolution of populations. A pair of individuals from the source population (parents) is taken and used to obtain two child solutions forming part of the new population. For this purpose, a random natural number Δ, $\Delta \in [0, N]$, is selected. Then, each candidate solution from the parent pair is divided into two sub-vectors and two child candidate solutions are formed through swapping these sub-vectors. For two individuals $A = [l_{A1}, l_{A2}, \ldots, l_{AN}]$ and $B = [l_{B1}, l_{B2}, \ldots, l_{BN}]$ the crossover at a point Δ results in

- $C_1 = [l_{A1}, l_{A2}, \ldots, l_{A\Delta}, l_{B(\Delta+1)}, \ldots, l_{BN}]$,
- $C_2 = [l_{B1}, l_{B2}, \ldots, l_{B\Delta}, l_{A(\Delta+1)}, \ldots, l_{AN}]$.

3.4 Mutation

The final step of the GA operation is mutation. The mutation rate is defined as one of the initial parameters of GA. The mutation depends on the chosen coefficient and in the discussed implementation occurs infrequently. Basically, mutation means to replace a randomly selected gene with a random value from the considered domain. If the rate equals 0.01 each gene of the individual after crossover operation has the probability of 1% that its value will mutate.

4 NUMERICAL STUDY

In order to assess the performance of continuous GA in simulation-based optimization of logistic networks, a MATLAB-based application has been created (sources available on-line [13]). It enables one to investigate various network topologies and influence of different sets of input parameters, i.e., the number of controlled nodes and external suppliers, connectivity structure, and demand pattern. In the application, once the network structure is defined, the optimization process using either exhaustive search (for simpler topologies) or GA is performed for different RSL vectors. The obtained results are processed, logged into a text file, graphically visualized for the user.

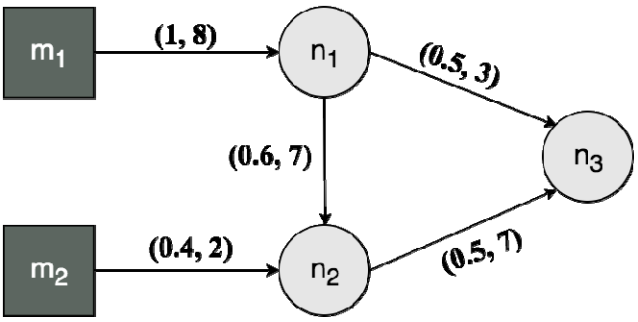


Figure 5. Logistic network scheme.

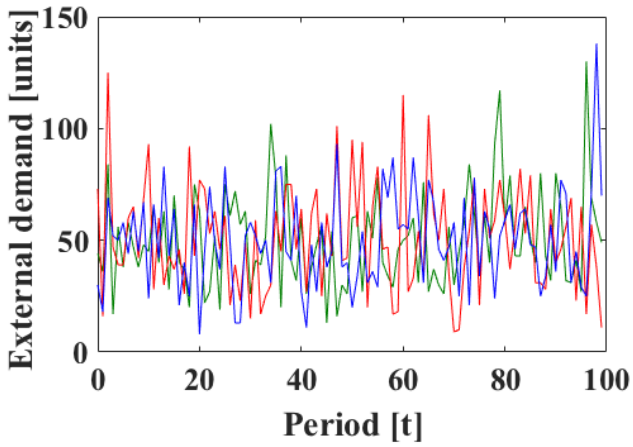


Figure 6. External demand imposed on controlled nodes.

Fig. 5 depicts the network topology selected for illustration purposes. The structure encompasses two external sources ($M = 2$) and three internal nodes ($N = 3$). The pair of numbers describing each node interconnection comprises the nominal quantity partitioning and delivery delay time (SF, DDT), e.g., node 2 acquires 60% of the established order from node 1 with lead-time delay of 7 periods. During the simulation, all the controlled nodes receive external demands. The

random external demand has been generated using the Gamma distribution with parameters *shape* = 5 and *scale* = 10. The demand requests with the values in the range [8, 138] units, are illustrated in Fig. 6. The simulation lasts $T = 100$ periods. The size of the GA population has been set as 10 individuals. The overall initial holding cost, considering all three controlled nodes, equals $3 \cdot 10^5$ units.

Table 1 groups the data with regard to different fitness function shaping coefficients. In turn, the dependence between the population size and the number of iterations needed to achieve a similar optimization result is illustrated via Table 2. The presented values are the averages from multiple simulation runs taken to leverage the GA inherent randomness (sometimes the best solution has already been found in the first few iterations).

Table 1. Optimization results.

Fitness function coefficients		Optimization results		Iteration number
φ	β	Fill rate	Holding cost	
1	1	0.9890	18625	976
1	10	0.9946	30932	189
1	50	0.9987	31948	233
10	1	0.9339	10183	63
10	10	0.9833	16756	345
10	50	0.9958	25006	556
50	1	0.5445	1851	33
50	10	0.9457	10548	451
50	50	0.9881	19678	485

Table 2. Population size dependence.

Population size	Iterations needed	Time reduction
2	13184	1.0
4	4160	0.54
10	607	0.23
20	251	0.17
50	22	0.05

The analysis of the obtained data indicates that even a small change of the fitness function coefficients might have a significant impact on the holding costs, customer satisfaction, and the process of determining the optimal solution. Depending on the particular objectives, the relative importance of those factors can be modified to achieve a desirable solution. Increasing φ (equation (16)) raises the importance of holding cost reduction, whereas increasing β provides a higher fill rate (improved customer satisfaction). Simultaneous increase of both coefficients directs the system to a desirable state of maximum service rate attained with minimum holding costs. The number of iterations to reach convergence is inversely proportional to φ and grows with β.

Fig. 7 illustrates the progress of optimization process defined as the improvement of the fitness function value during successive GA iterations. The fitness function shaping parameters have been set as $\varphi = 10$ and $\beta = 50$. The graph shows that the fitness function grows fast in the first several iterations, and then improves approximately linearly. Since the optimal solution is not known *a priori*, the stopping criterion is enforced through a predefined maximum number of iterations. As the second stopping criterion, besides the simulation duration, a threshold for the number of iterations without improvement of the fitness function value is specified. In the case under consideration the threshold equals 500 iterations. The dashed line in Fig. 7 indicates the best solution established through the exhaustive search. The exhaustive search requires significantly larger number of iterations to reach the optimum than GA. It is computationally infeasible for more complex network structures.

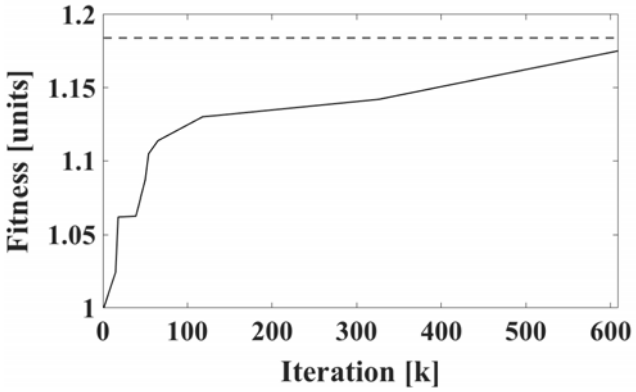


Figure 7. Fitness adjustment progress.

Figs. 8 and 9 display the stock level evolution at the controlled nodes for the initial and final (optimum) RSL setting. As can be noted from these graphs, the GA algorithm, during about

600 iterations, successfully eliminates superfluous resources. Thus, the holding cost is reduced, yet the stock level is kept positive most of the time, which implies a high fill rate.

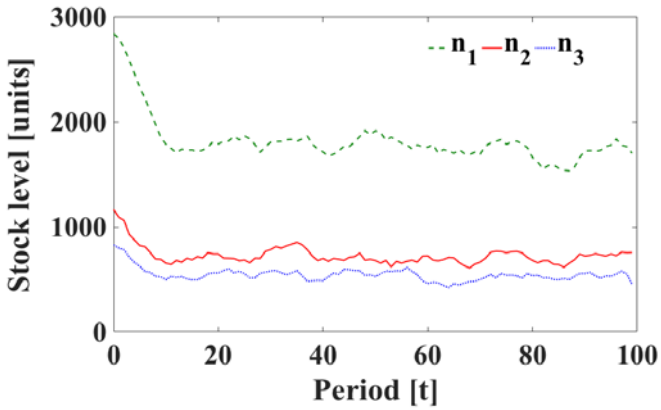


Figure 8. Stock level at the nodes for the initial generation.

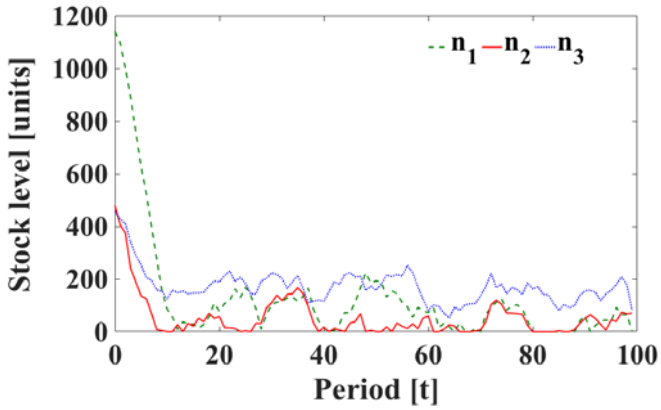


Figure 9. Stock level at the nodes for the final (optimal) generation.

4 CONCLUSIONS

The paper explores the use of continuous-domain GAs for the optimization mesh-type logistic networks governed by the OUT policy. The optimization purpose is to reduce the holding costs yet ensuring high customer satisfaction. It is achieved by adjusting the RSL at the network nodes. The fitness function of GA has been defined to allow a smooth balance between the holding costs (financial measure) and customer satisfaction through the adjustment of two algebraic coefficients. The generation size occurs to have a decisive impact on the GA operation and convergence time. The quantity of individuals in population is inversely proportional to the number of iterations needed to find the optimal solution. However, by increasing the generation size, the number of calculations and memory usage in each iteration grows fast. The numerous tests, executed for various network topologies, GA parameters, and fitness function shape coefficients, indicate that the application of GAs for RSL selection in logistic networks is advisable. As opposed to the full-search approach, the desired balance between the holding cost reduction and elevating the customer satisfaction is can be established in a reasonable time frame using common computers.

REFERENCES

1. Sarimveis, H., Patrinos, P., Tarantilis, C.D., Kiranoudis, C.T.: “Dynamic modeling and control of supply chain systems: A review,” Computers and Operations Research, vol. 35, pp. 3530--3561 (2008).
2. Hoberg, K., Bradley, J.R., Thonemann, U.W.: “Analyzing the effect of the inventory policy on order and inventory variability with linear control theory,” European Journal of Operational Research, vol. 176, pp. 1620--1642 (2007).
3. Ignaciuk, P., Bartoszewicz, A.: “Linear-quadratic optimal control of periodic-review perishable inventory systems,” IEEE Transactions on Control Systems Technology, vol. 20, pp. 1400--1407 (2012).
4. Boccadoro, M., Martinelli, F., Valigi, P.: “Supply chain management by H-infinity control,” IEEE Transactions on Automation Science & Engineering, vol. 5, pp. 703--707 (2008).
5. Movahed, K.K., Zhang, Z.-H.: “Robust design of (s, S) inventory policy parameters in supply chains with demand and lead time uncertainties,” International Journal of Systems Science, vol. 46, pp. 2258--2268 (2015).
6. Kim, C.O., Jun, J., Baek, J.K., Smith, R.L., Kim, Y.D.: “Adaptive inventory control models for supply chain management,” The International Journal of Advanced Manufacturing Technology, vol. 26, pp. 1184--1192 (2005).
7. Ignaciuk, P.: “LQ optimal and robust control of perishable inventory systems with multiple supply options,” IEEE Transactions on Automatic Control, vol. 58, pp. 2108--2113 (2013).
8. Ignaciuk, P.: “Nonlinear inventory control with discrete sliding modes in systems with uncertain delay,” IEEE Transactions on Industrial Informatics, vol. 10, pp. 559--568 (2014).
9. Chołodowicz, E., Orłowski, P.: “Comparison of a perpetual and PD inventory control system with Smith Predictor and different shipping delays using bicriterial optimization and SPEA2,” Pomiary Automatyka Robotyka, vol. 20, pp. 5--12 (2016).
10. Axsäter, S.: Inventory Control, Springer International Publishing (2015).
11. Simon, D.: Evolutionary Optimization Algorithms, John Wiley & Sons (2013).
12. Ignaciuk, P.: “State-space modeling and analysis of order-up-to goods distribution networks with variable demand and positive lead time,” Information Systems Architecture and Technology: Proceedings of 37th International Conference on Information Systems Architecture and Technology – ISAT 2016 – Part IV, pp. 55--65 (2017).
13. Wieczorek, Ł.: “Application sources,” [on-line]. Available: https://github.com/UksonCode/logistic-networks-ga-optimization.

International Journal of
NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

The *International Journal of New Computer Architectures and Their Applications* aims to provide a forum for scientists, engineers, and practitioners to present their latest research results, ideas, developments and applications in the field of computer architectures, information technology, and mobile technologies. The IJNCAA is published four times a year and accepts three types of papers as follows:

1. **Research papers**: that are presenting and discussing the latest, and the most profound research results in the scope of IJNCAA. Papers should describe new contributions in the scope of IJNCAA and support claims of novelty with citations to the relevant literature.
2. **Technical papers**: that are establishing meaningful forum between practitioners and researchers with useful solutions in various fields of digital security and forensics. It includes all kinds of practical applications, which covers principles, projects, missions, techniques, tools, methods, processes etc.
3. **Review papers**: that are critically analyzing past and current research trends in the field.

Manuscripts submitted to IJNCAA **should not be previously published or be under review** by any other publication. Plagiarism is a serious academic offense and will not be tolerated in any sort! Any case of plagiarism would lead to life-time abundance of all authors for publishing in any of our journals or conferences.

Original unpublished manuscripts are solicited in the following areas including but not limited to:

- Computer Architectures
- Parallel and Distributed Systems
- Storage Management
- Microprocessors and Microsystems
- Communications Management
- Reliability
- VLSI