

Volume 9 Issue 2 June 2016 ISSN 2088-7051

Jurnal Ilmu Komputer dan Informasi

Journal of Computer Science and Information





DYNAMIC AND INCREMENTAL EXPLORATION STRATEGY IN FUSION ADAPTIVE RESONANCE THEORY FOR ONLINE REINFORCEMENT LEARNING

Budhitama Subagdja

Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, N4-B3b-06 Nanyang Technological University 50 Nanyang Avenue, 639798, Singapore

E-mail: budhitama@ntu.edu.sg

Abstract

One of the fundamental challenges in reinforcement learning is to setup a proper balance between exploration and exploitation to obtain the maximum cumulative reward in the long run. Most protocols for exploration bound the overall values to a convergent level of performance. If new knowledge is inserted or the environment is suddenly changed, the issue becomes more intricate as the exploration must compromise the pre-existing knowledge. This paper presents a type of multi-channel adaptive resonance theory (ART) neural network model called fusion ART which serves as a fuzzy approximator for reinforcement learning with inherent features that can regulate the exploration strategy. This intrinsic regulation is driven by the condition of the knowledge learnt so far by the agent. The model offers a stable but incremental reinforcement learning that can involve prior rules as bootstrap knowledge for guiding the agent to select the right action. Experiments in obstacle avoidance and navigation tasks demonstrate that in the configuration of learning wherein the agent learns from scratch, the inherent exploration model in fusion ART model is comparable to the basic E-greedy policy. On the other hand, the model is demonstrated to deal with prior knowledge and strike a balance between exploration and exploitation.

Keywords: *Reinforcement Learning, Exploration Strategy, Adaptive Resonance Theory*

Abstrak

Salah satu permasalahan mendasar dari Reinforcement Learning adalah mengatur keseimbangan antara eksplorasi dan eksploitasi untuk mendapatkan ganjaran (reward) maksimal secara kumulatif dalam jangka waktu yang lama. Ketika pengetahuan awal diikutsertakan, masalah muncul karena eksplorasi yang dilakukan harus dikompromikan dengan pengetahuan sebelumnya yang telah dipelajari. Makalah ini menampilkan salah satu jenis jaringan saraf tiruan adaptive resonance theory (ART) berkanal ganda yang dikenal juga dengan sebutan fusion ART yang juga merupakan aproksimator Fuzzy untuk reinforcement learning dengan kemampuan meregulasi strategi eksplorasi sebagai sifat dasarnya. Model ini menawarkan proses pembelajaran yang stabil tetapi inkremental serta mampu melibatkan pengetahuan awal yang memilih aksi yang benar. Eksperimen menggunakan navigasi dan menghindari rintangan sebagai domain masalah menunjukkan bahwa konfigurasi pembelajaran menggunakan sifat dasar untuk meregulasi eksplorasi sebanding dengan metoda umum yang menggunakan aturan E-greedy. Di lain pihak, model yang diusulkan ini juga menunjukkan kemampuan dalam menggunakan pengetahuan awal serta mencapai keseimbangan dalam eksplorasi dan eksploitasi pengetahuan.

Kata Kunci: *Reinforcement Learning, Strategi Explorasi, Adaptive Resonance Theory*

1. Introduction

Reinforcement learning studies the techniques of how autonomous agents learn a task by directly interacting with the environment. The learning is often formalized as MDP (Markov Decision Process) in which the agent learns through cycles of sense, act, and learn [1]. Classical solutions to the reinforcement learning problem generally involve learning policy function which maps each state to a desired action and value function which associa-

tes each pair of state and action to a utility value (Q-values) [2].

A fundamental issue in the original formulation of reinforcement learning is how to setup a proper balance between exploration and exploitation so that the agent can maximally obtain its cumulative reward in the long run [3]. A common approach to address this issue is to apply a stochastic algorithm which selects between a random choice and a learnt action based on a probability value (E-greedy). In practice, the rate of explora-

tion changes in an ad hoc manner so that the agent tends to explore the environment at the beginning and gradually make use of its learnt knowledge. Another approach uses softmax policy that makes all actions to have an equal probability to be chosen at the beginning, and gradually tend to choose more on the action with the best (maximum) value estimate. This kind of approach assumes that the agent at the start of its activity explores the environment in a random fashion and gradually makes use of its own learnt knowledge.

The stochastic approach including the ones that use the gradual change is an effective way for reinforcement learning when the agent learns off-line from scratch (learning is conducted in a simulation and separated from the actual performance or usage in the environment). However, in online learning (the learning is conducted while performing actions in the actual environment and tasks), it is difficult if not impossible to determine the right probability for action selection and/or the proper rate of its change so that the overall performance can be optimal.

Another pertinent issue in reinforcement learning is how to conduct learning when a prior or additional knowledge is given on the run. The stochastic approach of action selection policy may not be suitable as the algorithm makes the agent explore the environment at the beginning regardless the imposition of pre existing knowledge. Recently, many approaches are starting to address this issue by incorporating transfer learning to reinforcement learning [4]. Most works focus on how the knowledge learnt from one task can be transferred to another reinforcement learner to boost its performance in learning the new task. Only few approaches like bayesian reinforcement learning [5] are considering to solve it through the internal action selection mechanism of reinforcement learning.

This paper presents a self-organizing neural network model called fusion ART (Adaptive Resonance Theory) that can be used as a function approximator for reinforcement learning. The neural network model has an inherent property to grow its nodes (neurons) and connections on-the-fly which can be useful to address the exploration-exploitation dilemma as proposed in this paper. Another intrinsic feature of the neural model is the direct correspondence between a sub-network in the model and a rule representation that maps a state to an action and the reward. This feature enables the neural network to be inserted with pre-given rules on-the-fly which can tackle the issues of transfer learning. In this paper I suggest that the growing neural network provides a self-regulating mechanism to control the rate of exploration. As a part of the inherent feature of the ART network,

Experiments in a navigation task domain show that the agent's performance while learning using the inherent selection mechanism is as optimal as learning with the E-greedy policy while facilitating the use of prior knowledge. The rest of the paper is organized as follows. Section 2 discusses the issues and challenges in implementing the right action selection policy and its influence to the use of prior knowledge. Section 3 introduces the self-organizing neural network model as a function approximator for reinforcement learning. The section also explains how the neural network can support prior and inserted knowledge and tackle the action selection issues. Section 4 introduces the navigation simulation task and presents the experimental results. The final section concludes and discusses future work.

2. Methods

Action Selection and Prior Knowledge in Reinforcement Learning

Reinforcement learning is typically formulated as learning to optimize a policy function π or a function that determines the action a to take by an agent (that is being controlled) given the current state s (see Figure 1). The optimization criteria is based on how much the agent is expected to receive rewards in the long run. When the model that provides good or optimal solution for policy π is not available, the agent must explore the environment to learn the policy based on experience. In this case, the policy π can be considered as the target knowledge to learn by the agent. To acquire the knowledge, the agent needs to select which action either to test (exploit) or to explore.

The main purpose of action selection in reinforcement learning is to trade off exploration and exploitation in such a way that the expected reward into the future can be maximized. Standard selection policies such as E-greedy or those that based on a simulated annealing procedure (e.g. softmax policy) consider the whole learning phase as a search process towards the maximum rewarding solution. Although it is unlikely that those action selection techniques can find the optimum solution, they offer practical approaches to find a very good solution in a fast manner over some noisy data.

When prior knowledge (e.g. as an initial policy consisting of mappings of states, actions, and rewards) are inserted before learning, intuitively some conditions are expected. The expected characteristics are as follows: 1) the learner tries to exploit the pre-inserted knowledge as much as possible to behave in the environment; 2) if no prior knowledge is suitable (e.g. the state is not much

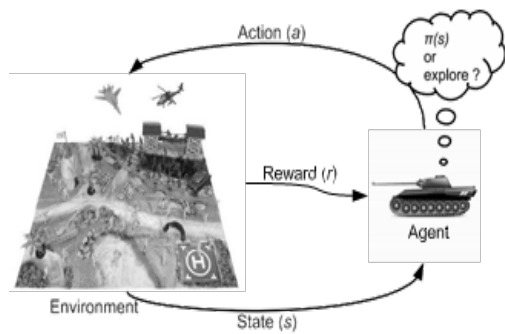


Figure 1. Exploration vs Exploitation in Reinforcement Learning.

explored) for the current state, explore the environment (e.g. random selection); 3) if prior knowledge is wrong or inaccurate, adjust the knowledge and gradually shift to full exploitation.

However, the expectations above contradict the application of the standard action selection policy (e.g. E-greedy) in reinforcement learning. The gradual change of exploration procedure always makes the agent to explore (random selection) at the beginning regardless the status of the agent's prior knowledge. Although it will shift eventually to its exploitation mode, the set of acquired knowledge during exploration makes the pre-given knowledge insignificant. The stochastic approach is suitable only for learning from scratch, but the inclusion of pre-inserted knowledge becomes useless.

Some non-greedy (non-stochastic) approaches of exploration strategy have been devised to be based on the context of explored states or the knowledge of the agent [3]. Methods such as counter based exploration [3] and interval estimation algorithm [6] make use of additional information to direct the selection of exploration behavior during online learning. For each state action pair (s , a), a certain value v is maintained as a visiting counter or a special utility value. A more recent version of the kind is R-Max, which also makes use of visiting counters incorporated with known/unknown flag on each state action pair [7]. The algorithm has also been extended to deal with large and continuous environment (R-Max-Lspi) [8]. Another approach uses Bayesian methods to address the exploration-exploitation trade off while facilitating transfer learning to incorporate prior knowledge [5]. The Bayesian approach encodes value function (e.g. transitions, rewards) as a probability distribution. The action is selected based on this probability distribution. Prior knowledge can be provided as a collection of transition and reward mappings incorporating a probability distribution.

Although the above exploration strategies can speed up the search towards the optimum so-

lution while taking the state of learnt knowledge into consideration, they are still not addressing the issue of prior knowledge effectively. It is impractical to set up a map of states, actions, and rewards together with entries like visited numbers or context dependent values prior to learning as those values should be obtained after the agent interacts with the environment. In the Bayesian approach of reinforcement learning, providing a prior probability distribution may be impractical in some cases especially those with large state spaces. Moreover, computing the posterior probability for a large state space may be too complex and intractable.

The next section shows another alternative of balancing exploration and exploitation in reinforcement learning with the inclusion of prior knowledge. The approach uses a neural network architecture supporting a direct mapping between the neural connections and a symbolic rule representation. Based on the existing neural connections, the learner chooses either the learnt action or explores the environment.

Multi-Channel Adaptive Resonance Theory

When the task of reinforcement learning is large and involving continuous values, neural networks can be used as function approximators that can generalize the state space reducing computational effort and substantially increasing the speed of learning. The neural network substitutes the look-up table for representing the value function. Our approach in this paper uses a multichannel adaptive resonance theory neural network called fusion ART as a function approximator for reinforcement learning.

Principles of Adaptive Resonance Theory

Adaptive Resonance Theory (ART) is a theory about how the brain autonomously learns to categorize, recognize, and predict objects and events in a dynamic environment [9]. It explains how a human brain acts as a self-organizing system that can rapidly learn huge amounts of data in real time from a changing world but still conduct it in a stable manner without catastrophically forgetting previously learnt knowledge. As a neural network architecture [10], ART solves the stability plasticity dilemma by employing two complementary iterating processes: bottom-up activation and top-down matching.

In Figure 2, the bottom-up process categorizes the input pattern by activation and selection of representative neurons (node j) in category field (F_2). The selected category primes and focuses the input pattern through the attentional system.

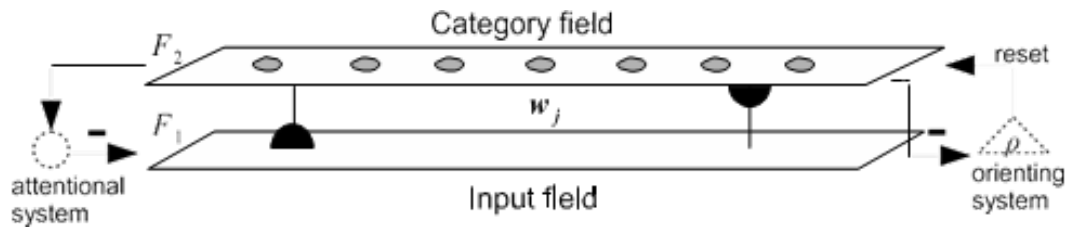


Figure 2. Simplified diagram of ART (Adaptive Resonance Theory) neural network architecture.

The top-down matching thus judges the degree it fits with the input pattern (F_2) using the orienting system. A resonant state occurs when there is a good enough match between the bottom-up and top-down patterns. The two processes resonate with each other so that one process reinforces and is reinforced by the other. At this state, learning is initiated to update weights w_j .

When the input pattern does not meet the top-down match according to the orienting system criteria, the current selected category is reset and suppressed until a match is found. However, if no match can be found possibly because the input pattern represents a truly novel experience, the search process recruits uncommitted neurons to learn the pattern as a novel information.

In this case, stable learning is achieved by allowing memories (or weights) to change when the input pattern is close enough to the expectation from top-down matching. The stability is also achieved by allowing the formation of a new category when the input is totally new.

Unlike other types of neural networks, in this bidirectional complementary process, there is no separation between phases of learning and activation. The categorization and learning are integrated parts of the network activity. Another characteristic of ART is that the recruitment of uncommitted neurons for representing novel patterns allows the network to grow so that network can learn incrementally in the long run in fast but stable manner.

Fusion ART Dynamics

A multi-channel ART is a variant of ART network that incorporate multiple input (output) neural fields. Each input (output) field is associated with its own input (or output) vector and some adjustable parameters. The multi-channel ART network is a flexible architecture that serves a wide variety of purposes. The neural network can learn and categorize inputs and can be made to map a category to some predefined fields by a readout process to produce the output. Similar to ART, learning can be conducted by adjusting the weighted connections while the network searches and selects the best matching node in the category field. When no

existing node can be matched, a new uncommitted node is allocated to represent the new pattern. The network thus grows dynamically in response to incoming patterns.

Depending on the task domain, the neural network may also apply different types of vector encoding on its different input fields. To handle continuous values, it is possible to employ vector calculation to process the activation of neurons which has been used in ART 2 architecture [11]. A simpler but more effective approach is to use Fuzzy operation to process inputs and categorization [12]. In this paper, we focus on the use of fuzzy operation (Fuzzy ART) rather than vector calculation (ART 2) as the former offers dynamic generalization mechanism and more expressive vector representation for prior know-ledge. Multichannel ART employing Fuzzy operations is also called fusion ART which is used throughout this paper.

In fusion ART, certain fields may apply complement coding to prevent category proliferation and enable generalization of input (output) attributes [12].

The generic network dynamics based on fuzzy ART operations, can be described as follows:

Input vectors: Let $I^k = (I_1^k, I_2^k, \dots, I_n^k)$ denote the input vector, where $I_1^k \in [0, 1]$ indicates the input i to channel k . With complement coding, the input vector I^k is augmented with a complement vector \bar{I}^k such that $\bar{I}_1^k = 1 - I_1^k$.

Activity vectors: Let $x^k = (x_1^k, x_2^k, \dots, x_n^k)$ denote the F_1^k activity vector, for $k = 1, \dots, n$. Initially, $x^k = I^k$. Let y denote the F_2 activity vector.

Weight vectors: Let w_j^k denote the weight vector associate with j th node in F_2 for learning the input patterns in F_1^k . Initially, F_2 contains only one uncommitted node and its weight vectors contain all 1's.

Parameters: The network dynamics is determined by learning rate parameters $\beta^k \in [0, 1]$ that sets how much the update is applied to the weight vector of the corresponding k -channel, contribution parameters $\gamma^k \in [0, 1]$ that corresponds to the

importance of the field k during bottom-up activation, and vigilance parameters $\rho^k \in [0, 1]$ which indicates how sensitive field k towards differences during template matching operation. In this case $k = 1, \dots, n$. choice parameter $\alpha^k \geq 0$ indicates the importance of field k among the other fields. It also used to avoid division by zero.

Code activation: A bottom-up activation firstly takes place in which the activities (known as choice function values) of the nodes in the F_2 field are computed. Given the activity vectors x^1, \dots, x^3 , for each F_2 node j , the choice function T_j is computed as follows:

$$T_j = \sum_{k=1}^K \gamma^k \frac{|x^k \wedge w_j^k|}{\alpha^k + |w_j^k|} \quad (1)$$

Where the fuzzy AND operation \wedge is defined by $(p \wedge q)_1 \equiv \min(p_1, q_1)$, and the norm $|\cdot|$ is defined by $|p| \equiv \sum_1 p_1$ for vectors p and q .

Code competition: A code competition process follows under which the F_2 node with the highest choice function value is identified. The winner is indexed at J where

$$T_j = \max\{T_j; \text{for all } F_2 \text{ node } j\} \quad (2)$$

When a category choice is made at node J , $y_j = 1$; and $y_j = 0$ for all $j \neq J$. this indicates a winner-take-all strategy.

Code activation: Before the node J can be used for prediction and learning, a template matching process checks that the weight templates of node J are sufficiently close to their respective input patterns. Specifically, resonance occurs if for each channel k , the match function m_j^k of the chosen node J meets its vigilance criterion:

$$m_j^k = \frac{|x^k \wedge w_j^k|}{x^k} \geq \rho^k \quad (3)$$

If any of the vigilance constraints is violated, mismatch reset occurs in which the value of the choice function T_j is set to 0 for the duration of the input presentation. The search process is guaranteed to end as it will either find a committed node that satisfies the vigilance criterion or activate an uncommitted node which would definitely satisfy the criterion due to its initial weight values of 1s.

Template learning: Once a resonance occurs, for each channel k , the weight vector w_j^k is modified by following learning rule:

$$w_j^{k(new)} = (1 - \beta^k)w_j^{k(old)} + \beta^k(x^k \wedge w_j^{k(old)}) \quad (4)$$

The learning rule adjusts the weight values towards the fuzzy AND of their original values and the respective weight values. The rationale is to learn by encoding the common attribute values of the input vectors and the weight vectors. For an uncommitted node J , the learning rates β^k are typically set to 1. For committed nodes, β^k can remain as 1 for fast learning or below 1 for slow learning in a noisy environment. When an uncommitted node is selecting for learning, it becomes committed and a new uncommitted node is added to the F_2 field. The network thus expands its nodes and connections dynamically in response to the input patterns.

Fusion ART can be used by itself as a stand-alone neural architecture. It has been applied as a reinforcement learning architecture for autonomous agents called FALCON [13], [14]. The reinforcement learner has been applied to control a non-player character (NPC) in a competition for autonomous bots in real-time first-person-shooter video game [15]. A fusion ART network can also be considered as a building block for more complex memory or cognitive architecture. In [16], it is used as the building block for modelling episodic memory comprising memory formation forgetting, and then consolidation processes. This model of episodic memory has been demonstrated to handle memory tasks in a fast-paced real-time first-person shooter video game environment [17]. Recently, the memory model has been demonstrated to exhibit transitive inference [18] and has been integrated with a larger more complex multi-memory system [19]. It is also worth mentioning that the principles of attention, expectation, and resonance search as mentioned in [9] have been successfully applied to demonstrate the transient formation of goal hierarchy, the implementation of planning, and the online acquisition of hierarchical procedural knowledge in [20] using fusion ARTs as the building blocks.

Reinforcement learning in fusion ART

For the reinforcement learning, the architecture makes use of 3-channel fusion ART architecture (Figure 4) comprising a categorization field F_2 and three input fields, namely a sensory field F_1^1 for representing the current state, an action field F_1^2 for representing the action to take, and Q (reward) field F_1^3 for representing the reinforcement value. Each input field employs independent parameters for code activation and pattern matching

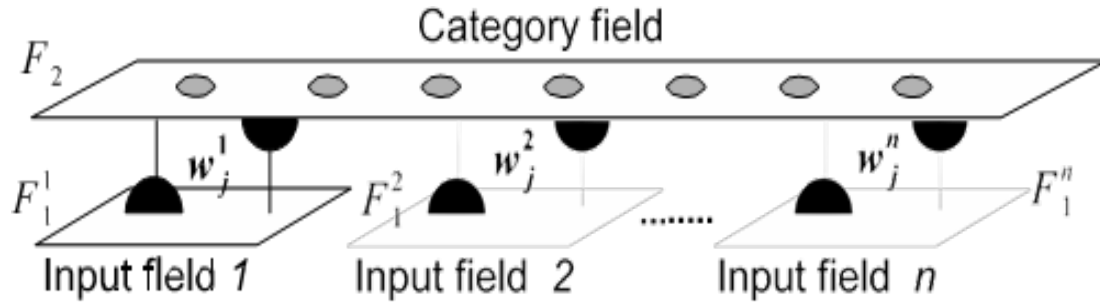


Figure 3. Fusion ART Architecture.

implying that different input (output) fields may be processed differently. It also implies that the neural network serves as a fuzzy approximator for reinforcement learning.

The basic reinforcement learning algorithm with fusion ART (also known as FALCON [14]) acquires an action policy by learning the mapping of the current state to the corresponding desirable action from experience. The system adjusts its internal representation upon receiving a reward feedback after performing each action. In a realistic environment, it may take a long sequence of actions before a reward or penalty is finally given. This is known as a temporal credit assignment problem in which we need to estimate the credit of an action based on what it will lead to a rewarding state eventually.

Temporal Difference Learning. Instead of learning a function that maps states to actions directly from immediate rewards, the FALCON incorporates Temporal Difference (TD) methods to estimate and learn the value function of state-action pairs $Q(s, a)$ that indicates the goodness for a learning system to take a certain action a in a given state s . In this way, TD methods can be used for multiple-step prediction problems, in which the merit of an action can only be known after several steps into the future (delayed reward).

Given the current state s , the neural network is used to predict the value of performing each available action. Upon input presentation, the activity vectors are initialized as $x^1 = S = (s_1, s_2, \dots, s_n)$ where $s_1 \in [0, 1]$ indicates the value of sensory input I , $x^2 = A = (a_1, a_2, \dots, a_n)$ where $a_i = 1$ if a corresponds to the action a , $a_i = 0$ for $i \neq I$, and $x^3 = Q = (1, 1)$. The Q values are estimated through resonance search processes for every possible action a_i . The value functions are then processed to select an action based on the action selection policy. Upon receiving a feedback (if any) from the environment after performing the action, a TD formula is used to compute a new estimate of the Q value for performing the chosen action in the current state. The new Q value is then used as the teaching signal for neural network to learn the as-

sociation of the current state and the chosen action to the estimated value.

One key component of the FALCON as a reinforcement learner is the iterative estimation of value functions $Q(s, a)$ using a temporal difference equation $\Delta Q(s, a) = \alpha TDerr$, where $\alpha \in [0, 1]$ is the learning parameter $TDerr$ and is the temporal error term. Using Q-learning, $TDerr$ is computed by equation(5).

$$TDerr = r + \gamma \max_{a'} Q(s', a') - Q(s, a) \quad (5)$$

Where r is the immediate reward value, $\gamma \in [0, 1]$ is the discount parameter, and $\max_{a'} Q(s', a')$ denotes the maximum estimated value of the next states'. It is important to note that the Q values involved in estimating $\max_{a'} Q(s', a')$ are computed by the same fusion ART network itself and not by a separate reinforcement learning system. The Q-learning update rule is applied to all states that the agent traverses. With value iteration, the value function $Q(s, a)$ is expected to converge to $r + \gamma \max_{a'} Q(s', a')$ over time.

Whereas many reinforcement learning systems have no restriction on the values of rewards r and thus the value function $Q(s, a)$, ART systems typically assume that all input values are bounded between 0 and 1. A solution to this problem is by incorporating a scaling term into the Q-learning updating equation directly. The Bounded Q-learning rule is given by equation(6).

$$\Delta Q(s, a) = \alpha TDerr(1 - Q(s, a)) \quad (6)$$

By introducing the scaling term $1 - Q(s, a)$, the adjustment of Q -values will be self-scaling so that they will not be increased beyond 1. The learning rule thus provides a smooth normalization of the Q values. If the reward value r is constrained between 0 and 1, we can guarantee that the Q values will remain to be bounded between 0 and 1.

The standard action selection policy used in FALCON, which is also applied in [21], is E-greedy policy which selects the action with the high-

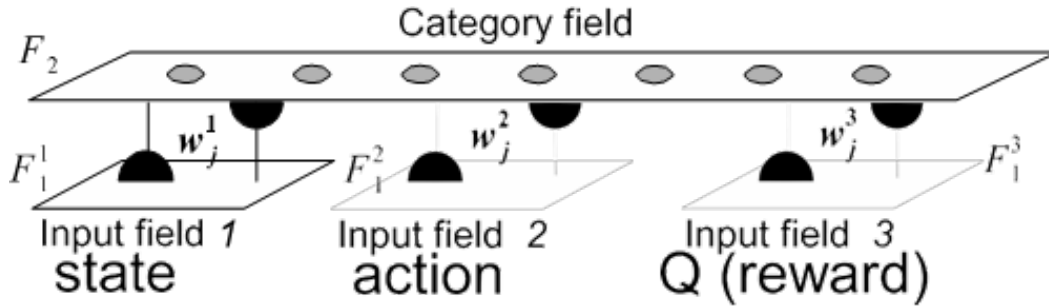


Figure 4. Fusion ART for Reinforcement Learning.

Algorithm 1: TD Learning with fusion ART

- 1 Initialize the neural network;
- 2 **repeat**
- 3 Sense the environment and formulate a state s ;
- 4 **for** each action $a \in A$ **do**
- 5 Predict the value of each $Q(s,a)$ by presenting the corresponding state and action vectors S and A to the corresponding input (output) fields;
- 6 **end**
- 7 Based on the computed value functions, select an action a based on the action selection policy;
- 8 Perform a action a ;
- 9 Observe the next state s' , receive a reward r if any;
- 10 Estimate $Q(s,a)$ following $\Delta Q(s, a) = \alpha TDerr$;
- 11 Present the corresponding s,a and Q to the corresponding vectors S, A and Q input (output) fields for learning;
- 12 Update $s \leftarrow s'$;
- 13 **Until** s is a terminal state;

Algorithm 2: Direct code retrieval in TD learning with fusion ART

- 1 Initialize the neural network;
- 2 **repeat**
- 3 Sense the environment and formulate a state s ;
- 4 Present vector S , vector $A^*=(1, \dots, 1)$ and vector $Q^*=(1,0)$ to corresponding fields to retrieve the best code and read it out to the corresponding vector S, A and Q ;
- 5 **If** according to the action policy, it is for exploration
- 6 **Then**
- 7 Select a random action and update the corresponding vector A accordingly;
- 8 Perform a action a ;
- 9 Observe the next state s' , receive a reward r if any;
- 10 Estimate $Q(s,a)$ following $\Delta Q(s, a) = \alpha TDerr$;
- 11 Present the corresponding s,a and Q to the corresponding vectors S, A and Q input (output) fields for learning;
- 12 Update $s \leftarrow s'$;
- 13 **Until** s is a terminal state;

est value with a probability of $1 - E$ and takes a random action with probability E . A decay E -greedy policy is thus adopted to gradually reduce the value of E over time so that the exploration is encouraged in the initial stage and gradually optimize the performance by exploiting familiar paths in the later stage. The standard TD learning algorithm using fusion ART is shown in Algorithm 1.

In [22], it has been proven that selecting the best action from the list of value of every possible action to take in the current state (as shown in line 4 to 7 in Algorithm 1) is equivalent to the retrieval of the action with the highest or maximal Q (reward) value given the current state. The search for the best action through the loop of value checking over all possible actions in the current state can be replaced by simply presenting the state S , the action A , and the maximum value vector Q^* , in which $Q^*=(1,0)$, to the corresponding fields in fusion ART to retrieve the code with the best action to take.

Algorithm 1 can be replaced with Algorithm 2 which simplifies the selection of the best action by directly retrieving the code with maximum Q value. This simplified algorithm also implies that

the implementation of policy function is equivalent to the retrieval of the best matching code in the network.

Rules and Adaptive Exploration

The network dynamics of fusion ART can be regarded as a myriad of learning operations namely similarity matching, associative learning, learning by instruction, and reinforcement learning. Through learning by similarity matching, the network identifies key situations in its environment that are of significance to the task at hand. Through learning by association (or directly as instructions), it learns the association between typical situations and their corresponding desired actions. Finally, through the reinforcement signals given by the environment, it learns the value of performing a specific action in a given situation.

Rule Insertion

The recognition categories learnt in F_2 layer are compatible with a class of IF-THEN rules that maps a set of input attributes (antecedents) in one

pattern channel (field) to a disjoint set of output attributes (consequents) and the estimated reward value in the other channel. In this way, instructions in the form of IF-THEN rules (accompanied by reward values) can be readily translated into the recognition categories at any stage of the learning process. Particularly, each corresponding rule can have the following format:

$$\begin{aligned} & \text{IF } C_1 \wedge C_2 \wedge \dots \wedge C_N \\ & \text{THEN } A_1 \wedge A_2 \wedge \dots \wedge A_M (Q = R) \end{aligned}$$

Where \wedge indicates a logical AND operator. Hence, the rules are conjunctive in the sense that the attributes in the IF clause and in the THEN clause have AND relationships. Each conditional attribute c_i and action attribute a_j correspond to each element of the state and action vector respectively.

Algorithm 3: Rule insertion algorithm

- 1 Set all ρ^k to 1;
- 2 **For** each rule r , **do**
- 3 Translate antecedent, consequent and reward of r , into vector S , A and Q respectively;
- 4 Present vector S , A and Q to the corresponding input fields;
- 5 Invoke the network dynamics procedure (section 2.4) to insert the translate rule;
- 6 **end**

Imposing fusion ART with domain knowledge through explicit instructions may serve to improve learning efficiency and predictive accuracy. To insert rules into the network, the IF-THEN clauses and reward values of each instruction (rule) can be translated into corresponding input vectors. let s , a , and q are the state vector, the action vector, and the reward vector respectively. after the translation, the state-action-reward (s , a , q) triad of vectors are inserted into the network through the iterative performance of the code activation, code competition, template matching and template learning procedure (section fusion art dynamics). during rule insertion, the vigilance parameters (ρ^k) are set to 1s to ensure that each distinct rule is encoded by one category node. The procedure for inserting rules is shown in algorithm 3.

Adaptive Exploration

It is suggested in this paper that the feature of fusion ART that grows nodes and connections during learning can actually be used to regulate exploration and exploitation. The regulation can also be inherently conducted according to the status of the agent's existing knowledge. The intuition is simply that when the neural network fails to find any existing matching category for the current state, an uncommitted node is allocated for a new rule.

The allocation implies that the agent needs to explore the environment as the action for that particular state is unknown or novel. Otherwise, it just follows the original step to select the best category (rule) for further execution. Consequently, the action selection process involving E-greedy can be replaced with a simpler model that adaptively turns to exploration by selecting a random action only if no matching category can be found.

Algorithm 4 shows the action selection procedure using the adaptive exploration strategy of fusion ART.

Algorithm 4: Action selection policy with adaptive exploration strategy

- 1 Given state s an the best code retrieved using direct code method (Algorithm 2);
- 2 **If** the code is newly allocated or just committed node in F_2
- Then**
- Set the policy to exploration;
- else**
- Set the policy to exploitation;
- end**

The next section shows some experiments results confirming that the characteristics of the proposed action selection policy is comparable to the standard E-greedy policy even when prior rules are provided. The performance improvement rate is not influenced by any constant value or parameters.

This simplification of selection policy can be achieved thanks to the principle of overcoming the stability plasticity dilemma in ART neural networks. The exploration strategy can lead to convergent if the trajectory towards the rewarding state is known or has been familiarized. However, the policy will switch back to exploration mode if novel situation is detected.

3. Results and Analysis

To conduct test the new exploration policy against the standard E-greedy, an experiment is conducted

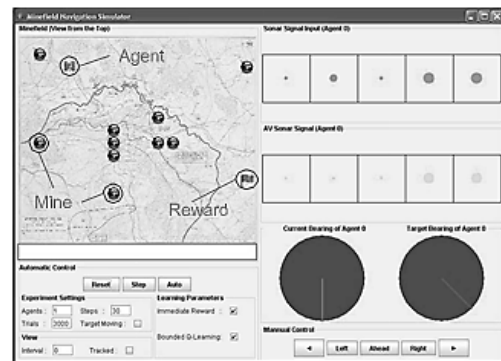


Figure 5. Screenshot of the minefield simulator

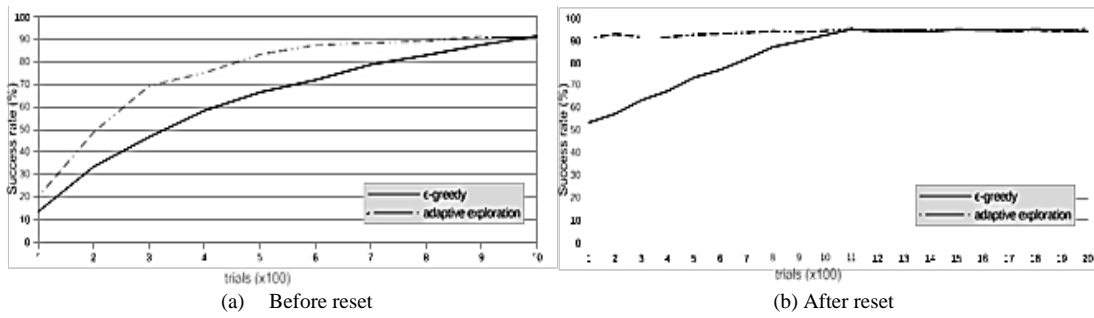


Figure 6. The effect of E reset (a) before the reset; (b) after the reset.

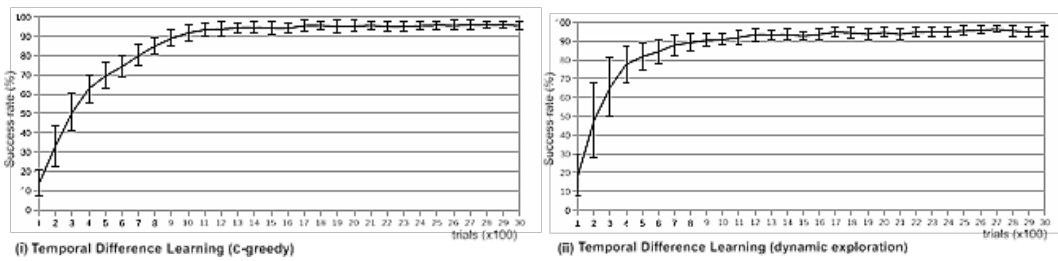


Figure 7. The learning performance of TD learning with *E*-greedy and inherent exploration policy.

using a simulation. The simulation described in this paper is similar to the underwater navigation and mine avoidance domain developed by Naval Research Lab (NRL) [22]. The objective is to navigate through a minefield to a randomly selected target position in a specified time frame without hitting a mine. In each trial, an auto-nomous vehicle (AV) starts from a randomly chosen position in the field, and repeats the cycles of sense, act, and learn. A trial ends when the system reaches the target (success), hits a mine (failure), or exceeds 30 sense-act-learn cycles (out of time).

As the configuration of the minefield is generated randomly and changes every time a trial is started, the system needs to learn strategies that can be carried trials. In addition, the system has a rather coarse sensory capability with a 180 degree forward view based on five sonar sensors. For each direction *I*, the sonar signal is measured by $s_i = 1/1+d_i$ where d_i is the distance to an obstacle (that can be a mine or the boundary of the minefield) in the *i* direction. Other input attributes of the sensory (state) vector include the bearing of the target from the current positin. In each step, the system can choose on out of the five possible actions, namely move left, move diagonally left, move straight ahead, move diagonally right and move right. Figure 5 shows the screenshot of the simulator used in the experiment.

We conduct experiments with the delayed evaluative feedback (TD). At the end of a trial, a reward of 1 is given when the AV reaches the target. A reward of 0 is given when the AV hits a mine or runs out of time. The neural network con-

sists of 18 nodes in the sensory fields (representing 5x2 complement-coded sonar signals and 8 target bearing values), 5 nodes in the action field, and 2 nodes in the reward field (representing the complement-coded function value). The TD fusion ART model employs a set of parameter values as follows: $\alpha = 0.1$ (all fields), $\beta = 1.0$, and $\gamma = 1.0$ (all fields). Baseline vigilances ρ^k for all fields are set to 0. ρ^k value increases by a small factor (0.001) during the resonance search until a resonant state is achieved (in which ρ^k is reset to 0 after that). This approach of changing the vigilance parameter during resonance search is also known as match tracking [19]. For *E*-greedy action selection policy, the decaying rate is $E = 0.001$.

The experiment is conducted to see whether the existence of prior knowledge influences the performance improvement in the neural network both with *E*-greedy and the adaptive exploration policy. At first, each reinforcement learning configuration (TD with *E*-greedy and TD with the adaptive action selection) runs for 1000 trials to let it learns some rules (Figure 6 (a)). After that, the simulation parameters are reset their baseline level and a refreshed run continues until 2000 trials (Figure 6 (b)) which make 3000 trials in total. The baseline reset is made to investigate the effect of prior knowledge in TD learning using the *E*-greedy policy considering the knowledge has been captured in the first 1000 trials. Figure 6 shows the effect of the *E* reset on success rates of TD learning with multi-channel ART for both the standard *E*-greedy policy and the inherent policy.

The results are averaged over 30 independent experiment runs. After 1000 trials, the performance of TD with E-greedy is dropped to about half of its maximum performance. This is due to the set back of E value after the reset that makes the agent to do the exploration once more. On the other hand, the one with the adaptive exploration policy keeps steady on its optimal level regardless the E reset. The figure indicates that the new policy can still continue its improvement based on its current state of knowledge (rules).

The next stage of the experiment is conducted to observe whether the adaptive exploration strategy can still make the performance converges just as the E-greedy policy when no prior rules are given. In this stage, each success rate is complemented with its standard deviation. Figure 7 shows the performance rate and the standard deviation of both E-greedy and the adaptive policy in TD learning averaged over 30 independent experiment runs. It is shown that in both E-greedy and the adaptive exploration policy, the performance converges. In fact, it is shown that using the adaptive policy, the success rate converges faster than the other one using E-greedy although the difference is marginal.

The results of the experiment indicate that the adaptive exploration strategy is comparable to the standard strategy using E-greedy policy. However, the proposed strategy can overcome the discontinuity in learning despite the change in the environment configuration since the learnt knowledge is maintained. On the other hand, the interrupt during the trial sets back the performance gained to the initial state since the current E level is not kept in the learnt knowledge although the last set of learnt knowledge is maintained over trials.

4. Conclusion

We have presented a multi-channel neural network framework for realizing a reinforcement learning. The neural network serves as fuzzy approximator to handle tasks and situations involving multiple channels, multimodality, large state space and continuous values. It is also possible to incorporate prior knowledge into the network by mapping rules into neural nodes and connections. Besides learning from scratch, the agent can be provided with a set of useful knowledge as bootstraps so that it can behave more effectively in a given environment.

A new action-selection policy is also introduced which also reveals that the multi-channel ART neural network has inherent features supporting online reinforcement learning. The inherent policy for adaptive exploration can replace the E-greedy policy so that prior knowledge can be ex-

ploited more effectively by putting the context of the decision on the status of the agent's knowledge. Our comparative experiments show that the inherent policy is comparable with the E-greedy.

In any case, the advantage of using the fusion ART's inherent dynamic exploration is that the performance rate becomes dependent to the availability of prior knowledge rather than some external parameters or heuristics. It is demonstrated in this paper that any useful pre-existing rules will be exploited instead of being ignored by the new policy. Consequently, the policy would instantly select the exploration mode whenever a lack of knowledge is identified. In that case, fusion ART offers an inherent self-regulating control for exploration and exploitation in reinforcement learning.

Our future work will involve applying the inherent action selection strategy to more complex and challenging domains. Another possible extension is to study the relationships between the neural network parameters (e.g vigilance, contribution, choice, and learning rate parameters) and the effectiveness of reinforcement learning. Moreover, the adaptive exploration policy described in this paper is still quite simple that decision is limited to either to explore or to exploit the knowledge. Another more sophisticated strategy deserves to be investigated which may combine the policy with other strategies while exploiting the inherent features in the ART neural network as much as possible.

Acknowledgement

This research is supported by the National Research Foundation, Prime Ministers Office, Singapore under its IDM Futures Funding Initiative.

References

- [1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge: MIT Press, 1998.
- [2] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3/4, pp. 279–292, 1992.
- [3] S. B. Thrun, "The role of exploration in learning control," in *Hand book of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*, D. A. White and D. A. Sofge, Eds. Florence: Van Nostrand Reinhold, 1992, pp. 527–599.
- [4] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, vol. 10, pp. 1633–1685, 2009.

- [5] R. Dearden, N. Friedman, and S. Russell, "Bayesian q-learning," in AAAI '98/IAAI '98: Proceedings of the fifteenth national / tenth conference on Artificial intelligence / Innovative applications of artificial intelligence. Menlo Park: American Association for Artificial Intelligence, 1998, pp. 761–768.
- [6] L. P. Kaelbling, *Learning in Embedded Systems*. Cambridge: MIT Press, 1993.
- [7] R. I. Brafman and M. Tennenholtz, "R-max a general polynomial time algorithm for near optimal reinforcement learning," *Journal of Machine Learning Research*, vol. 3, pp. 213–231, 2002.
- [8] L. Li, M. L. Littman, and C. R. Mansley, "Online exploration in least-squares policy iteration," in *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, 2009, pp. 733–739.
- [9] S. Grossberg, "Adaptive resonance theory: How a brain learns to consciously attend, learn, and recognize a changing world," *Neural Networks*, vol. 37, no. 2013, pp. 1–47, 2013.
- [10] G. Carpenter and S. Grossberg, "Adaptive Resonance Theory," in *The Handbook of Brain Theory and Neural Networks*, M. Arbib, Ed. Cambridge, MA: MIT Press, 2003, pp. 87–90.
- [11] G. A. Carpenter and S. Grossberg, "ART 2: Stable self-organization of pattern recognition codes for analog input patterns," *Applied Optics*, vol. 26, pp. 4919–4930, 1987.
- [12] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759–771, 1991.
- [13] A.-H. Tan, N. Lu, and D. Xiao, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 230–244, 2012.
- [14] A.-H. Tan, "Falcon: A fusion architecture for learning, cognition, and navigation," in *Proceedings of 2004 IEEE International Joint Conference on Neural Networks (IJCNN'04)*, 2004, pp. 3297–3302.
- [15] A.-H. Tan, N. Lu, and D. Xiao, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 230–244, 2012.
- [16] D. Wang, B. Subagdja, A.-H. Tan, and G.-W. Ng, "Creating human-like autonomous players in real-time first person shooter computer games," in *Proceedings of the Twenty-First Annual Conference on Innovative Applications of Artificial Intelligence (IAAI'09)*, 2009, pp. 173–178.
- [17] W. Wang, B. Subagdja, A.-H. Tan, and J. A. Starzyk, "Neural modeling of episodic memory: Encoding, retrieval, and forgetting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 10, pp. 1574–1586, 2012.
- [18] —, "Neural modeling of sequential inferences and learning over episodic memory," *Neurocomputing*, vol. 161, no. 2015, pp. 229–242, 2015.
- [19] W. Wang, B. Subagdja, A.-H. Tan, and Y.-S. Tan, "A self-organizing multi-memory system for autonomous agents," in *Proceedings of 2012 IEEE International Joint Conference on Neural Networks (IJCNN'12)*, 2012, pp. 480–487.
- [20] B. Subagdja and A.-H. Tan, "iFALCON: A neural architecture for hierarchical planning," *Neurocomputing*, vol. 86, pp. 124–139, 2012.
- [21] —, "Direct code access in self-organizing neural architectures for reinforcement learning," in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007)*, 2007, pp. 1071–1076.
- [22] D. Gordan and D. Subramanian, "A cognitive model of learning to navigate," in *Proceedings, Nineteenth Annual Conference of the Cognitive Science Society*, 1997, pp. 271–276.

DYNAMICS BASED CONTROL OF A SKID STEERING MOBILE ROBOT

Osama Elshazly¹, Hossam S. Abbas², and Zakarya Zyada³

¹Mechatronics and Robotics Eng. Dept., Egypt-Japan University of Science and Technology (E-JUST),
Qesm Borg Al Arab, Alexandria, 21934, Egypt

²Electrical Eng. Dept., Faculty of Engineering, Assiut University, Al Hamraa Ath Thaneyah, Qesm Than
Asyut, Assiut, 71515, Egypt

³Faculty of Mechanical Engineering (FKM), Universiti Teknologi Malaysia, 81310 Johor, Malaysia.
("On-leave" Mechanical Power Engineering, Tanta University, 31511 Tanta, Egypt.)

E-mail: osama.elshazly@ejust.edu.eg¹, chossam.abbas10@gmail.com², zakarya@utm.my³

Abstract

In this paper, development of a reduced order, augmented dynamics-drive model that combines both the dynamics and drive subsystems of the skid steering mobile robot (SSMR) is presented. A Linear Quadratic Regulator (LQR) control algorithm with feed-forward compensation of the disturbances part included in the reduced order augmented dynamics-drive model is designed. The proposed controller has many advantages such as its simplicity in terms of design and implementation in comparison with complex nonlinear control schemes that are usually designed for this system. Moreover, the good performance is also provided by the controller for the SSMR comparable with a nonlinear controller based on the inverse dynamics which depends on the availability of an accurate model describing the system. Simulation results illustrate the effectiveness and enhancement provided by the proposed controller.

Keywords: *Skid Steering Mobile Robots, Reduced order model, Linear Quadratic Regulator, Feed-Forward Compensation, Inverse Dynamics.*

Abstrak

Dalam paper ini, pengembangan *reduced order, augmented* model *dynamics-drive* yang menggabungkan kedua dinamika dan subsistem drive dari *skid steering mobile robot* (SSMR) ditampilkan. Sebuah algoritma kontrol Linear Quadratic Regulator (LQR) dengan kompensasi *feed-forward* dari *disturbances part* termasuk dalam *reduced order augmented* model *dynamics-drive* dirancang. Pengendali yang diusulkan memiliki banyak keuntungan seperti kesederhanaan dalam hal desain dan implementasi dibandingkan dengan skema kontrol nonlinear kompleks yang biasanya dirancang untuk sistem ini. Selain itu, kinerja yang baik juga disediakan oleh pengendali untuk SSMR sebanding dengan pengendali nonlinear berdasarkan dinamika *inverse* yang tergantung pada ketersediaan dari model yang akurat yang menggambarkan sistem. Hasil simulasi menggambarkan efektivitas dan peningkatan oleh pengendali yang diusulkan.

Kata Kunci: *Skid Steering Mobile Robots, Reduced order model, Linear Quadratic Regulator, Kompensasi Feed-Forward, Dinamika Inverse.*

1. Introduction

Nowadays, mobile robotics constitute an attractive research field with a high potential for practical applications. Skid steering mobile robots (SSMR) are widely used as outdoor mobile robots. Simple and robust mechanical structure, faster response, high maneuverability, strong traction, and high mobility make SSMR to be suitable for many applications such as loaders, farm machinery, mining and military applications [1][2]. Due to complex kinematic constraints and wheel/ground interactions, considering the dynamical model and designing a proper controller for skid-steering mobile robots (SSMR)

are challenging tasks. For the topic of skid steering mobile robots modelling and control, several research papers have been published to cover this research field. The stability of wheeled skid steering mobile robots has been studied using model based nonlinear control techniques by explicitly considering dynamics and drive models [3-5].

Moreover, the kinematics have been addressed as the relation of linear and angular velocities with the position of the vehicle in some works [6] [7]. However, major skid effects have not been considered, which arise at a lower level, in the relation between drive velocities and vehicle velocities. Estimation of tire/ground friction of a simpli-

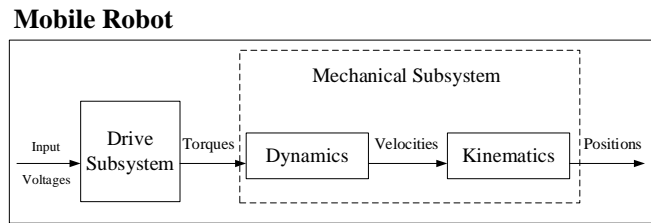


Figure 1. An electrically driven mobile robot decomposition

fied dynamic model for wheeled skid steering mobile robots is considered using an online adaptive control [8]. A trajectory tracking control were presented in [9] to control a wheeled skid-steering mobile robot moving on a rough terrain based on different control methods such as practical fuzzy lateral control, longitudinal control and sensor pantilt control; the authors used ADAMS and MATLAB co-simulation platform to assess these control laws. In [10] [11], a thorough dynamic analysis of a skid-steered vehicle has been introduced; this analysis considers steady-state (i.e. constant linear and angular velocities) dynamic models for circular motion of tracked vehicles.

The control system design for such SSMR dynamics and drive subsystems was done independently without considering the combination between dynamic and drive model. Moreover, the robot motor voltages/torques may be the true control inputs of some mobile robots. In other words, the low level control of such mobile robot may be essential for trajectory tracking problem to consider robot dynamics. Therefore, the main contribution in this paper is to design a dynamics-based controller to control the augmented dynamics-drive model which has the inputs of motors voltages and the outputs of robots are the longitudinal and angular velocities.

In this paper, a reduced order, augmented dynamics-drive model that combines both the dynamics and drive subsystems of the SSMR based on [3] is developed. Then, a Linear Quadratic Regulator (LQR) with feed-forward compensation of the disturbances part included in the reduced order augmented dynamics-drive model is developed. For comparison, an inverse dynamics controller is designed. The main advantage of the proposed LQR controller is simplicity of design and experimental implementation in comparison with nonlinear controllers which are highly complex for implementation. The two controllers are used for a reference tracking of SSMR.

The remaining of the paper is organized as follows: A reduced order augmented dynamics-drive model of SSMR is presented in a systematic way in section 2. Section 2 is also devoted to present the development of the controller using LQR with feed forward compensation for the system. An inverse

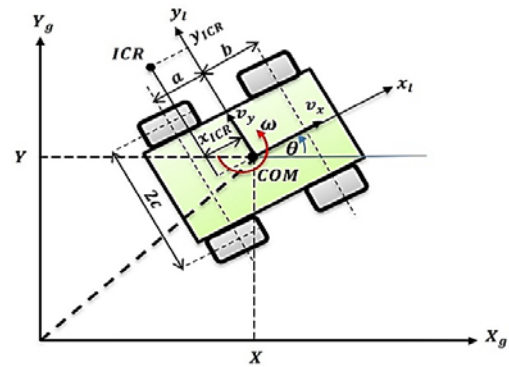


Figure 2. Schematic diagram of SSMR

dynamics controller is presented to ensure robustness to the nonlinearities of dynamical model. Section 3 is dedicated for extensive simulation results considering trajectory tracking problem. Conclusions and ideas for future work are given in Section 4. Finally, acknowledgments and references complete the paper.

2. Methods

Robot Model

In this subsection, the dynamics mathematical description of an SSMR moving on a planar surface is reviewed. The mobile robot mathematical model [3] can be divided into three parts: kinematics, dynamics and drive subsystems, see Figure 1. In this paper, we focus on the first two blocks, i.e., the drive and dynamics subsystems, and we use them for reference tracking control of both the linear and angular velocities.

Model of Dynamics Subsystem

The dynamics effects play an important role in SSMR vehicles. Such dynamics consider the forces and torques required to cause the robot motion. Considering forces and torques is important in mobile robot control to follow a desired trajectory.

Moreover, dynamics consider the wheel ground interactions which affect the performance of the SSMR vehicles. So, this subsection is dedicated for the dynamics properties of the SSMR moving on a

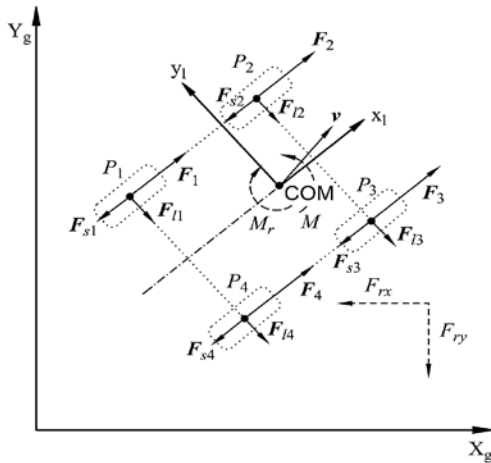


Figure 3. Active and resistive forces of the vehicle [3]

planar surface as shown in Figure 2 to be described [3][4]. It is assumed that the mass distribution of the vehicle is homogeneous, kinetic energy of the wheels and drives is neglected and detailed derivations of the tire relations is omitted and the reader can return to [3] for more details.

Using the Euler-Lagrange principle with Lagrange multipliers to include non-holonomic constraints, the dynamics equation of the robot can be obtained. The planar motion of SSMR allows us to assume that the potential energy of the robot is $PE(q) = 0$. Therefore the Lagrangian L of the system equals the kinetic energy as given by equation(1):

$$L(q, \dot{q}) = T(q, \dot{q}) \quad (1)$$

The following equation(2) can be developed if the kinetic energy of the vehicle is considered and the energy of rotating wheels is neglected:

$$T(q, \dot{q}) = \frac{1}{2} m v^T v + \frac{1}{2} I \omega^2 \quad (2)$$

where m and I represent the mass of the robot and moment of inertia of the robot about the COM, respectively. Equation(2) can be rewritten in the following equation(3) (since $v^T v = v_x^2 + v_y^2 = \dot{X}^2 + \dot{Y}^2$).

$$T(q, \dot{q}) = \frac{1}{2} m (\dot{X}^2 + \dot{Y}^2) + \frac{1}{2} I \dot{\theta}^2 \quad (3)$$

The inertial forces can be obtained after calculating the partial derivative of kinetic energy and its time-derivative as given by equation(4).

$$\frac{d}{dt} \left(\frac{\partial T(q, \dot{q})}{\partial \dot{q}} \right) = \begin{bmatrix} m \ddot{X} \\ m \ddot{Y} \\ I \ddot{\theta} \end{bmatrix} = M \ddot{q} \quad (4)$$

where:

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \quad (5)$$

Based on Fig.3, the generalized resistive forces can be calculated as the following equation(6).

$$R(\dot{q}) = [F_{rx}(\dot{q}) \quad F_{ry}(\dot{q}) \quad M_r(\dot{q})]^T \quad (6)$$

such that, $F_{rx}(\dot{q})$ and $F_{ry}(\dot{q})$ are the resultant forces expressed in the inertial frame which can be calculated as given by equation(7) and equation(8).

$$F_{rx}(\dot{q}) = \cos \theta \sum_{i=1}^4 F_{si}(v_{ix}) - \sin \theta \sum_{i=1}^4 F_{li}(v_{iy}) \quad (7)$$

$$F_{ry}(\dot{q}) = \sin \theta \sum_{i=1}^4 F_{si}(v_{ix}) + \cos \theta \sum_{i=1}^4 F_{li}(v_{iy}) \quad (8)$$

Also, the resistant moment around the center of mass $M_r(\dot{q})$ can be obtained as given by equation (9).

$$M_r(\dot{q}) = -a \sum_{i=1,4} F_{li}(v_{iy}) + b \sum_{i=2,3} F_{li}(v_{iy}) + c \left[-\sum_{i=1,2} F_{si}(v_{ix}) + \sum_{i=3,4} F_{si}(v_{ix}) \right] \quad (9)$$

such that F_{si} results from the rolling resistant moment τ_{ri} as shown in Figure 4 and F_{li} denotes the lateral reactive force. F_{si} and F_{li} can be considered as wheel friction forces and can be described as given by equation(10).

$$\begin{aligned} F_{si} &= \mu_{sci} m g \operatorname{sgn}(v_{ix}) \\ F_{li} &= \mu_{lci} m g \operatorname{sgn} \end{aligned} \quad (10)$$

where μ_{sci} and μ_{lci} denote the coefficients of the longitudinal and lateral friction forces. g is the gravity of acceleration. The active force F_i is linearly dependent on the wheel control input τ_i by the inverse of the wheel radius (r) given by equation(11).

$$F_i = \frac{\tau_i}{r} \quad (11)$$

The active forces generated by the actuators which make the robot move can be expressed in the

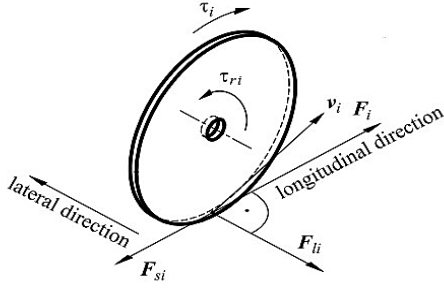


Figure 4. Forces acting on a wheel

inertial frame as the following equation(12).

$$\begin{aligned} F_x &= \cos \theta \sum_{i=1}^4 F_i \\ F_y &= \sin \theta \sum_{i=1}^4 F_i \end{aligned} \quad (12)$$

The active torque around the COM is calculated as defined in equation(13).

$$M = c(-F_1 - F_2 + F_3 + F_4) \quad (13)$$

In consequence, the vector F of active forces has the following equation(14).

$$F = [F_x \quad F_y \quad M]^T \quad (14)$$

Using equations(11) to equation(13), and assuming that the radius of each wheel is the same, we get equation(15).

$$F = \frac{1}{r} \begin{bmatrix} \cos \theta \sum_{i=1}^4 F_i \\ \sin \theta \sum_{i=1}^4 F_i \\ c(-F_1 - F_2 + F_3 + F_4) \end{bmatrix} \quad (15)$$

A new torque control input τ is defined for notation simplification as equation(16).

$$\tau = \begin{bmatrix} \tau_L \\ \tau_R \end{bmatrix} = \begin{bmatrix} \tau_1 + \tau_2 \\ \tau_3 + \tau_4 \end{bmatrix} \quad (16)$$

where τ_L and τ_R denote the torques produced by the wheels on the left and right sides of the vehicle, respectively. Using equation(15) and equation (16), we can obtain the following equation(17).

$$F = B(q)\tau \quad (17)$$

such that B is the input transformation matrix defined as equation(18).

$$B = \frac{1}{r} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ -c & c \end{bmatrix} \quad (18)$$

The following dynamics model (equation (19)) is obtained using equation(4), equation(6) and equation(17).

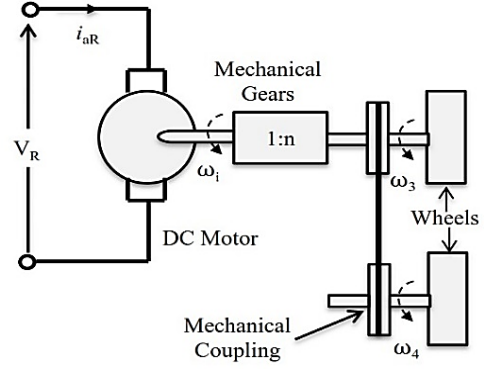


Figure 5. Drive system on the right side of the vehicle

$$M(q)\ddot{q} + R\dot{q} = B(q)\tau \quad (19)$$

A velocity constraint can be considered in order to complete the kinematic model as equation(20).

$$v_y + x_{ICR}\omega = 0 \quad (20)$$

Equation(20) describes a non-holonomic constraint which can be written in the Pfaffian form as equation(21).

$$[-\sin \theta \quad \cos \theta \quad x_{ICR}][\dot{X} \quad \dot{Y} \quad \dot{\theta}]^T = A(q)\dot{q} = 0 \quad (21)$$

The equation that describe the kinematics of the robot is given by [3]:

$$\dot{q} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & x_{ICR} \sin \theta \\ \sin \theta & -x_{ICR} \cos \theta \\ 0 & c \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} = S(q)\eta \quad (22)$$

here $\eta = [v_x \quad \omega]^T$, v_x is the longitudinal velocity, ω is the angular velocity of the robot, τ is the torque control input, and $q = [X \quad Y \quad \theta]^T$ represents the generalized coordinates of the center of mass (COM) of the robot, i.e., the COM position, with X and Y ; and θ is the orientation of the local coordinate frame with respect to the inertial frame.

Equation(19) describes the dynamics of a free body only and does not include the non-holonomic constraint described by equation(21), so a constraint has to be imposed on equation(19). A vector of Lagrange multipliers, λ , is introduced to include the non-holonomic constraint into the dynamics equation as the following equation(23).

$$M(q)\ddot{q} + R\dot{q} = B(q)\tau + A^T(q)\lambda \quad (23)$$

It would be more suitable to express equation (23) as a function of the internal velocity vector η for control purposes. So, by multiplying equation (23) by $S^T(q)$ from the left, we can obtain the following equation(24).

$$S^T(q) M(q)\ddot{q} + S^T(q) R\dot{q} = S^T(q) B(q)\tau + S^T(q) A^T(q)\lambda \quad (24)$$

By taking the time derivative of equation(22), we obtain equation(25).

$$\ddot{q} = \dot{S}(q)\dot{\eta} + S(q)\ddot{\eta} \quad (25)$$

From equation(22), equation(23), and equation(25), the dynamic equation as introduced in [3] is defined in equation(26).

$$\bar{M}(q)\dot{\eta} + \bar{C}(\dot{q})\eta + \bar{R}(\dot{q}) = \bar{B}(q)\tau \quad (26)$$

The matrices \bar{M} , \bar{C} , \bar{R} , and \bar{B} are given, respectively, by equation(27) to equation(30).

$$\bar{M} = \begin{bmatrix} m & 0 \\ 0 & mx_{ICR}^2 + I \end{bmatrix} \quad (27)$$

$$\bar{C} = \begin{bmatrix} 0 & mx_{ICR}^2\dot{\theta} \\ -mx_{ICR}^2\dot{\theta} & mx_{ICR}^2\dot{x}_{ICR} \end{bmatrix} \quad (28)$$

$$\bar{R} = \begin{bmatrix} F_{rz}(\dot{q}) \\ x_{ICR}F_{ry}(\dot{q}) + M_r \end{bmatrix} \quad (29)$$

$$\bar{B} = 1 \begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ \frac{-c}{r} & \frac{c}{r} \end{bmatrix} \quad (30)$$

Equation(26) can be written as the following equation(31).

$$\dot{\eta} = \bar{M}^{-1}(q)\bar{B}(q)\tau - \bar{M}^{-1}(q)\bar{C}\eta - \bar{M}^{-1}(q)\bar{R}(\dot{q}) \quad (31)$$

where \bar{M} is nonsingular for all q , $2c$ is the vehicle width, and the coordinate of the instantaneous center of rotation (ICR) is defined as (x_{ICR}, y_{ICR}) .

Model of Drive Subsystem

In this subsection, the drive subsystem of an SSMR model is developed. It is assumed that the robot is driven by two DC motors, one at each side, with mechanical gears. Moreover, it is also assumed that there is no slip in the belt which connect each two wheels on each side and there are no nonlinearities in the mechanical coupling in the drive subsystem. In Figure 5, a simplified scheme of the drive on the right side of the robot is depicted. Considering only one drive and assuming that the two motors and gears have the same parameters, the relation between the torque τ_R and voltage V_R can be written as the following equation(32).

$$\tau_R = nK_m i_{aR} \quad (32)$$

where i_{aR} is the armature current, K_m is the motor torque constant, n is the gear ratio such that $(n > 1)$.

$$V_R = L_a \frac{d}{dt} i_{aR} + R_a i_{aR} + nK_e \omega_R \quad (33)$$

where L_a and R_a denote the series inductance and resistance of the rotors, respectively, K_e is the electromotive force, and ω_R is the right hand side motor angular velocity.

The left ω_L and right ω_R sides angular velocities can be obtained from the following equation (34) and equation(35).

$$\omega_L = \frac{v_x - c\omega}{r} \quad (34)$$

$$\omega_R = \frac{v_x + c\omega}{r} \quad (35)$$

In order to obtain the overall model of the two motors of the drive system, Equation(33) can be written as the following equation(36) and equation (37).

$$\begin{bmatrix} \dot{i}_{aR} \\ \dot{i}_{aL} \end{bmatrix} = \frac{1}{L_a} \begin{bmatrix} V_L \\ V_R \end{bmatrix} - \frac{R_a}{L_a} \begin{bmatrix} i_{aL} \\ i_{aR} \end{bmatrix} - \frac{nK_e}{rL_a} \begin{bmatrix} v_x - c\omega \\ v_x + c\omega \end{bmatrix} \quad (36)$$

$$\begin{bmatrix} \dot{i}_{aR} \\ \dot{i}_{aL} \end{bmatrix} = \begin{bmatrix} -nK_e & ncK_e \\ rL_a & rL_a \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} - \frac{R_a}{L_a} \begin{bmatrix} i_{aL} \\ i_{aR} \end{bmatrix} + \frac{1}{L_a} \begin{bmatrix} V_L \\ V_R \end{bmatrix} \quad (37)$$

where V_R and V_L are the right and left side motor voltage signals, respectively.

Dynamics-Drive Augmented Model

In previous work, a control law was designed for dynamics and drive models independently. However, in this research it is required to design one controller for both dynamics and drive models. So, the drive and the dynamics subsystems are combined in one state space representation for such purpose. Substitution from equation(32) into equation(31), gives this equation(38).

$$\begin{bmatrix} \dot{v}_x \\ \dot{\omega} \end{bmatrix} = nK_m \bar{M}^{-1} \bar{B} \begin{bmatrix} i_{aL} \\ i_{aR} \end{bmatrix} - \bar{M}^{-1} \bar{C} \begin{bmatrix} v_x \\ \omega \end{bmatrix} - \bar{M}^{-1} \bar{R} \quad (38)$$

which can be mathematically manipulated to obtain equation(39).

$$\begin{bmatrix} \dot{v}_x \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{nK_m i_{a1}}{mr} & \frac{nK_m i_{a2}}{mr} \\ -ncK_m i_{aL} & ncK_m i_{aR} \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} - \begin{bmatrix} X_{ICR}\omega \\ -mx_{ICR}\omega \\ mx_{ICR}^2 + I \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} - \begin{bmatrix} F_{rx} \\ m \\ -x_{ICR}F_{ry} + M_r \\ mx_{ICR}^2 + I \end{bmatrix} \quad (39)$$

The dynamics-drive model of the SSMR vehicle described by equation(37) and (39) can be re-

presented by the following state space representation:

$$\begin{bmatrix} \dot{v}_x \\ \dot{\omega} \\ \dot{i}_{aR} \\ \dot{i}_{aL} \end{bmatrix} = A_1 \begin{bmatrix} v_x \\ \omega \\ i_{aL} \\ i_{aR} \end{bmatrix} + B_1 \begin{bmatrix} V_L \\ V_R \end{bmatrix} + D_1 \quad (40)$$

where

$$A_1 = \begin{bmatrix} 0 & -x_{ICR}\omega & \frac{nK_m}{mr} & \frac{nK_m}{mr} \\ \frac{mx_{ICR}\omega}{mx_{ICR}^2+I} & 0 & \frac{-ncK_m}{r(mx_{ICR}^2+I)} & \frac{ncK_m}{r(mx_{ICR}^2+I)} \\ \frac{-nK_e}{rL_a} & \frac{ncK_e}{rL_a} & \frac{-R_a}{L_a} & 0 \\ \frac{-nK_e}{rL_a} & \frac{-ncK_e}{rL_a} & 0 & \frac{-R_a}{L_a} \end{bmatrix} \quad (41)$$

$$B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{L_a} & 0 \\ 0 & \frac{1}{L_a} \end{bmatrix}, \quad D_1 = \begin{bmatrix} \frac{-F_{rx}}{m} \\ -x_{ICR}F_{ry}-M_r \\ 0 \\ 0 \end{bmatrix} \quad (42)$$

It may be difficult to design a controller for a system of fourth order described by equation(40). So, it will be simpler if the system is reduced to a lower order. The order of the model described by equation(40) can be reduced to 2 by neglecting the motor inductance, i.e. $L_a = 0$. Therefore, equation (33) can be written defined in equation(43).

$$V_R = R_a i_{aR} + nK_e \omega_R \quad (43)$$

From equation(43), we can obtain the current of the motors as equation(44).

$$\begin{bmatrix} i_{aL} \\ i_{aR} \end{bmatrix} = \begin{bmatrix} \frac{1}{R_a} & 0 \\ 0 & \frac{1}{R_a} \end{bmatrix} \begin{bmatrix} V_L \\ V_R \end{bmatrix} + \begin{bmatrix} \frac{-nK_e}{rR_a} & \frac{ncK_e}{rR_a} \\ \frac{-nK_e}{rR_a} & \frac{-ncK_e}{rR_a} \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} \quad (44)$$

The motors torque can be calculated as:

$$\begin{bmatrix} \tau_L \\ \tau_R \end{bmatrix} = \begin{bmatrix} \frac{nK_m}{R_a} & 0 \\ 0 & \frac{nK_m}{R_a} \end{bmatrix} \begin{bmatrix} V_L \\ V_R \end{bmatrix} + \begin{bmatrix} \frac{-n^2K_mK_e}{rR_a} & \frac{n^2cK_mK_e}{rR_a} \\ \frac{-n^2K_mK_e}{rR_a} & \frac{-n^2cK_mK_e}{rR_a} \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} \quad (45)$$

By using equation(45), the state space representation of the reduced order overall system can be given by equation(46).

$$\begin{bmatrix} \dot{v}_x \\ \dot{\omega} \end{bmatrix} = A_2 \begin{bmatrix} v_x \\ \omega \end{bmatrix} + B_2 \begin{bmatrix} V_L \\ V_R \end{bmatrix} + D_2 \quad (46)$$

where:

$$A_2 = \begin{bmatrix} \frac{-2n^2K_mK_e}{mr^2R_a} & -x_{ICR}\omega \\ \frac{mx_{ICR}\omega}{mx_{ICR}^2+I} & \frac{-2n^2c^2K_mK_e}{r^2R_a(mx_{ICR}^2+I)} \end{bmatrix} \quad (47)$$

$$B_2 = \begin{bmatrix} \frac{nK_m}{mrR_a} & \frac{nK_m}{mrR_a} \\ \frac{-ncK_m}{rR_a(mx_{ICR}^2+I)} & \frac{ncK_m}{rR_a(mx_{ICR}^2+I)} \end{bmatrix} \quad (48)$$

$$D_2 = \begin{bmatrix} \frac{-F_{rx}}{m} \\ -x_{ICR}F_{ry}-M_r \\ \frac{mx_{ICR}^2+I}{} \end{bmatrix} \quad (49)$$

Equation(46) can be written in the general form of the state space representation as the following equation(50).

$$\dot{X} = A_2(X)X + B_2U + D_2(X) \quad (50)$$

where the system states are represented by equation (51).

$$X = [v_x \quad \omega]^T \quad (51)$$

The control input vector U is represented by $U = [V_L \quad V_R]^T = [V_1 \quad V_2]^T$.

Finally, we consider $D_2(X)$, the last term of equation(50) as a disturbance; this will be neglected in the design of the Linear Quadratic Regulator (LQR) in the next subsection; however, it will be considered in the design of the feed-forward controller to overcome its effect on the system as shown below in the next subsection.

Dynamics-Based Controller

Motor voltages/torques may be the true control inputs of some mobile robots. In other words, the low level control of such mobile robot may be essential for trajectory tracking problem to consider robot dynamics. So, dynamics-based controller is presented to control the augmented dynamics-drive model which has the inputs of motors' voltages (V_L and V_R) and the outputs of robot's velocities (v_x and ω). In this section, two control laws are introduced, LQR with feed-forward compensation and inverse dynamics controller to be used for a reference tracking of the augmented dynamics-drive model presented in the previous section.

Linear Quadratic Regulator (LQR)

LQR control plays a crucial role in optimal control systems and it has many applications, e.g. airplane flight control, chemical process control, and motor control. The main purpose of LQR control is to ob-

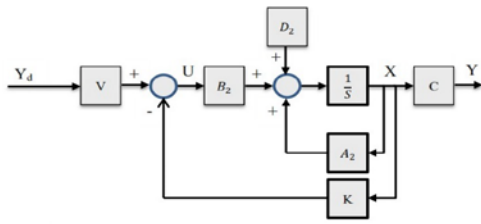


Figure 6. LQR closed loop system block diagram

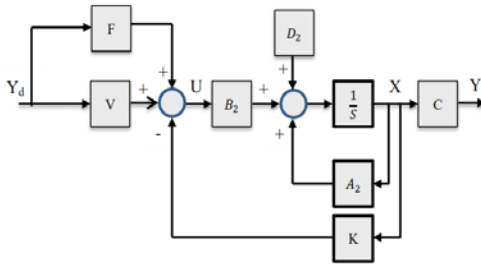


Figure 7. LQR with feed-forward compensation.

tain an optimal control law in order to minimize a cost function along the trajectory of a linear system. Consider the state space representation of a system [12]:

$$\dot{X} = AX + BU \quad (52)$$

$$Y = CX \quad (53)$$

with $X(t) \in R^n$, $U(t) \in R^m$ and the initial condition is $X(0)$. We assume here that all the states are measurable and seek to find a state-variable feedback control law as equation(54).

$$U = -KX \quad (54)$$

that gives desirable closed-loop properties such that K is the feedback gain vector. The optimal feedback state regulation minimizes the quadratic cost function defined by equation(55).

$$J = \frac{1}{2} \int_0^{\infty} (X^T Q X + U^T R U) dt \quad (55)$$

where Q is a symmetric positive semi-definite matrix and R is a symmetric positive definite matrix. The optimal feedback gain vector can be calculated by equation(56).

$$K = R^{-1} B^T P \quad (56)$$

where P is the solution of the Algebraic Riccati Equation defined by equation(57).

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \quad (57)$$

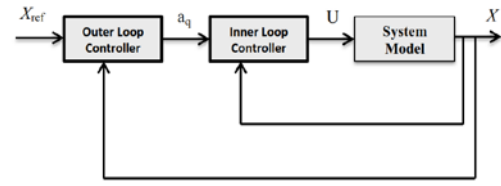


Figure 8. Inner-outer loop control for inverse dynamics

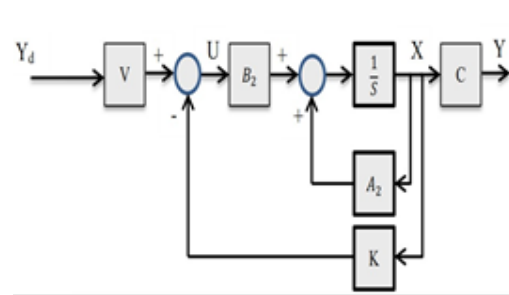


Figure 9. LQR closed loop system block diagram without the $D2(x)$ term

Our objective is to design an optimal control law to provide a reference tracking controller of a linearized model of the skid steering mobile robot vehicle using the voltages of the two motors as the inputs to the system. The closed loop reference tracking control system for the SSMR is described by equations (52) and (53). The control law for reference tracking LQR controller can be defined as equation(58) [13]:

$$U = -KX + VY_d \quad (58)$$

where $V = -(C^T(A - BK)^{-1}B)^{-1}$ to insure zero steady-state error and Y_d is the desired output.

The block diagram that describes the closed loop system for the augmented dynamics-drive model which was described by equation(50) with the LQR control law defined by equation(58) is depicted in Figure 6.

Feed-Forward Compensation

In the simulation, the term $D2(x)$ of equation(50) is considered as a disturbance; therefore, the system response is tested with and without this term to check its effect. A feed-forward compensation is proposed to overcome the effect of this term.

The overall closed-loop block diagram of the system with the LQR controller and the feed-forward compensation is depicted in Figure 7, where F is a vector which can be calculated by equation (59).

$$F = \begin{bmatrix} \frac{R_{qr}}{2nK_l} \left[\left(\frac{-x_{1CR}\omega F_{ry}}{c} - \frac{M_r}{c} \right) + F_{rx}(\dot{q}) \right] \\ \frac{R_{qr}}{2nK_l} \left[\left(\frac{-x_{1CR}\omega F_{ry}}{c} + \frac{M_r}{c} \right) + F_{rx}(\dot{q}) \right] \end{bmatrix} \quad (59)$$

This vector represents the compensation of the two components of $D_2(x)$.

Inverse Dynamics Controller

As presented before, the LQR controller with feed-forward compensation was designed for a linearized model of the SSMR and it is more favorable to design a controller for the nonlinear model of the SSMR directly. So, we design an inverse dynamics controller for the nonlinear system model based on the idea presented in [14]. Consider again the SSMR dynamics model described by equation(50). The idea of inverse dynamics is to seek a non-linear feedback control law described by equation(60).

$$U = f(X, t) \quad (60)$$

The block diagram of the scheme of the inverse dynamics control is shown in Figure 8. Consider a new input to the system a_q such the equation(61).

$$a_q = A_2(X)X + B_2U + D_2(X) \quad (61)$$

By mathematical manipulation (equation(62)) of equation(61), we can obtain the control input U as equation(63).

$$a_q - A_2(X)X - D_2(X) = B_2U \quad (62)$$

$$U = B_2^{-1}(a_q - A_2(X)X - D_2(X)) \quad (63)$$

The new control input a_q can be given by equation (64).

$$a_q = k(X - X_{ref}) \quad (64)$$

where k is the gain to be designed for the controller and X_{ref} is the reference system states.

3. Results and Analysis

In this simulation, the dynamic and drive models described by the reduced order model (46) is considered, so the inputs to the system are the two motor voltages V_1 and V_2 and the outputs of the system are the linear velocity v_x and angular velocity ω . The system parameters applied for simulation are shown in Table 1. In practice, it is difficult to measure x_{ICR} value, so it is assumed here to be equation(65) [3] [4].

$$x_{ICR} = constant = x_0 \quad x_0 \in (-a, b) \quad (65)$$

where a and b are positive kinematic parameters of the robot depicted in Figure 2.

TABLE 1
PARAMETERS OF THE MODEL

Variable	Value	Unit
$a = b$	39	mm
c	34	mm
r	26.5	mm
m	1	Kg
l	0.0036	Kg.m ²
x_0	-15	mm
R_a	3.9	Ω
K_i	8.55	mN.m/A
K_e	8.55	mV.s/rad
n	12	—

LQR with Feed-Forward Compensation Controller Results

For the LQR controller, the matrices Q and R , which are described in equation(57), are chosen by trial and error to be equation(66).

$$Q = \begin{bmatrix} 0.1 & 0 \\ 0 & 0 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (66)$$

In simulation, we test three cases for the system. Firstly, we assume that the system described by equation(50) is without the disturbance $D_2(x)$ term as shown in Figure 9. Secondly, this term $D_2(x)$ is added to the system as depicted in Figure 6 to check its effect on the response and asses the controller performance. Finally, the feed-forward compensation part is included as shown in Figure 7 to overcome the effect of the disturbance part $D_2(x)$.

Figure 10 shows the system response to a square reference for both the linear and angular velocities. The system can track the desired inputs quickly and without any overshoot. Also, Figure 10 provides the control signals of the system motors and it is obvious that the control signals are in the limits of the maximum motor voltages ($\pm 24VDC$).

In order to demonstrate the effectiveness of the controller, the term $D_2(x)$ is added to the system and the system response and control signals are shown in Figure 11. It can be shown that the system is affected by this part which destabilizes the angular velocity output.

In order to overcome the effect of $D_2(x)$, the feed-forward compensation is added to the controller. The system response and control signals are shown in Figure 12. It is shown that the system can again track the desired references. Moreover, we see that the control signals are different from those that are depicted in Figure 10 and Figure 11 as the control law is changed, see Figure 7.

From the above analysis, the simulation results illustrate that the LQR performance is enhan-

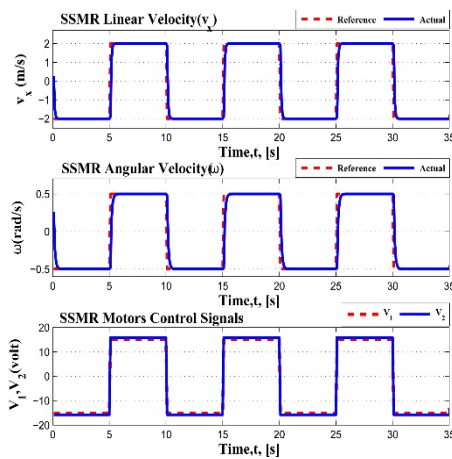


Figure 10. System response (without the nonlinear term

$$D_2(x))$$

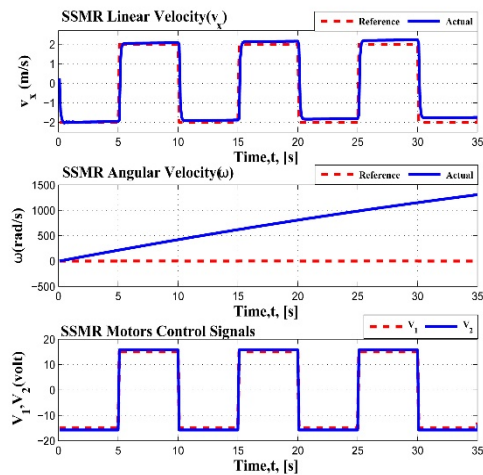


Figure 11. System response (including $D_2(x)$ as part of the system)

ced for the linearized model; however, the stability and tracking of the reference input may not be guaranteed if this controller is applied to the nonlinear system as it was applied to a linearized model with disturbance term only and the stability and tracking is not guaranteed.

Inverse Dynamics Controller Results

Next, we apply the inverse dynamics controller to the nonlinear system described by equation(50) directly. The gain k found in equation(64) was selected to be 10. The system response and control signals are shown in Figure 13. These results clearly illustrate the performance enhancement of the

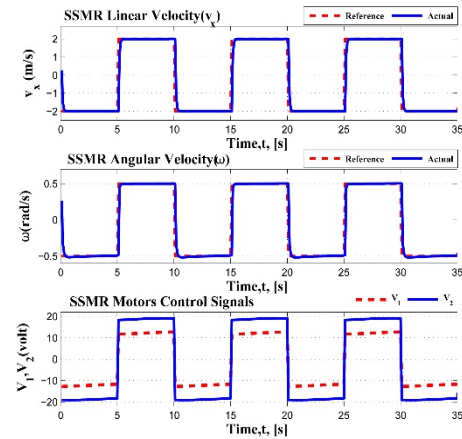


Figure 12. System response with feed-forward compensation

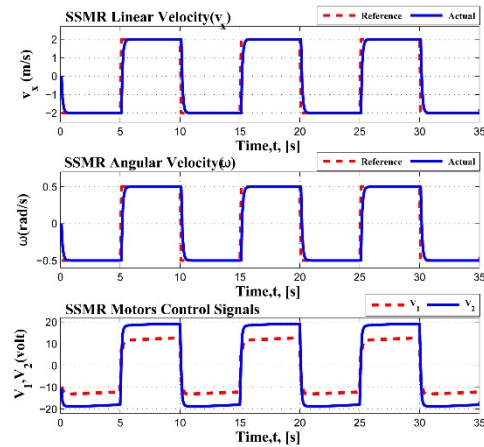


Figure 13. System response with an inverse dynamic controller

inverse dynamics controller that can deal with nonlinear systems. Also, the two control signals are within the limits of the motor voltages. But, it is clear from the inverse dynamics controller design that an accurate model of the nonlinear system is needed as its proper design depends on the system matrices. Finally, it should be noted that the inverse dynamics controller is applied to the nonlinear model of the SSMR directly while the LQR with the feed-forward compensation is applied to a linearized model of the system.

4. Conclusion

An LQR with feed-forward compensation algorithm is presented in this paper for controlling a reduced order model of augmented dynamic and drive models of an SSMR. The controller is considered by merging a linear quadratic regulator controller with a feed-forward compensation to impro-

ve tracking accuracy and overcome effects of nonlinearities. For comparison, an inverse dynamics controller is designed. The LQR controller with the feed-forward compensation shows satisfactory results.

In the future work, a development of a time-varying LQR controller to deal with the nonlinearities of the system will be investigated. Experimental implementation will be considered as well.

References

- [1] W. Yu, O. Y. Chuy, E. G. Collins, and P. Hollis, "Analysis and experimental verification for dynamic modeling of a skid-steered wheeled vehicle," *IEEE transactions on robotics*, vol. 26, no. 2, pp. 341–353, 2010.
- [2] A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. G. Cerezo, and J. Gonzalez, "Experimental kinematics for wheeled skid-steer mobile robots," in *In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, USA, 2007, pp. 1222–1227.
- [3] K. Kozłowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot," *Int. J. Appl. Math. Comput. Sci.*, vol. 14, no. 4, pp. 477–496, 2004.
- [4] Z. Jian, W. S. Shuang, L. Hua, and L. Bin, "The sliding mode control based on extended state observer for skid steering of 4-wheel-drive electric vehicle," in *In Proc. of the 2nd International Conference on Consumer Electronics, Communications and Networks (CE-CNet)*, China, 2012, pp. 2195–2200.
- [5] L. Caracciolo, A. D. Luca, and S. Iannitti, "Trajectory tracking control of a four-wheel differentially driven mobile robot," in *In Proc. of IEEE International Conference on Robotics and Automation*, USA, 1999, pp. 2632–2638.
- [6] E. Maalouf, M. Saad, and H. Saliyah, "A higher level path tracking controller for a four-wheel differentially steered mobile robot," *Journal of Robotics and Autonomous Systems*, vol. 54, pp. 23–33, 2006. Fig. 13. System response with an inverse dynamic controller.
- [7] K. Kozłowski and D. Pazderski, "Practical stabilization of a skid steering mobile robot - a kinematic-based approach," in *In Proc. of IEEE 3rd International Conference on Mechatronics*, Hungary, 2006, pp. 519–524.
- [8] J. Yi, D. Song, J. Zhang, and Z. Goodwin, "Adaptive trajectory tracking control of skid-steered mobile robots," in *In Proc. of the IEEE International conference on robotics and automation*, Italy, 2007, pp. 2605–2610.
- [9] Y. Yi, F. Mengyin, Z. Hao, X. Guangming, and S. Changsheng, "Control methods of mobile robot rough-terrain trajectory tracking," in *In Proc. of the 8th IEEE International Conference on Control and Automation*, China, 2010, pp. 731–738.
- [10] J. Y. Wong, *Theory of Ground Vehicles*, 3rd ed. New York: John Wiley & Sons, 2001.
- [11] J. Y. Wong and C. F. Chiang, "A general theory for skid steering of tracked vehicles on firm ground," *In Proc. of Institution of Mechanical Engineering, Part D, J. Automobile Eng.*, vol. 215, pp. 343–355, 2001.
- [12] K. Ogata, *Modern Control Engineering*, 5th ed. Pearson Prentice-Hall, 2010.
- [13] M. Ruderman, J. Krettek, F. Hoffmann, and T. Bertram, "Optimal state space control of dc motor," in *In Proc. of the 17th World Congress The International Federation of Automatic Control*, Korea, 2008, pp. 5796–5801.
- [14] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, 5th ed. New York: John Wiley & Sons, 2006.

A NOVEL APPROACH TO STUTTERED SPEECH CORRECTION

Alim Sabur Ajibola, Nahrul Khair bin Alang Md. Rashid, Wahyu Sediono, and Nik Nur
Wahidah Nik Hashim

Mechatronics Engineering Department, International Islamic University Malaysia, Jalan Gombak,
Kuala Lumpur, 53100, Malaysia

E-mail: moaj1st@yahoo.com, wsediono@iiu.edu.my

Abstract

Stuttered speech is a dysfluency rich speech, more prevalent in males than females. It has been associated with insufficient air pressure or poor articulation, even though the root causes are more complex. The primary features include prolonged speech and repetitive speech, while some of its secondary features include, anxiety, fear, and shame. This study used LPC analysis and synthesis algorithms to reconstruct the stuttered speech. The results were evaluated using cepstral distance, Itakura-Saito distance, mean square error, and likelihood ratio. These measures implied perfect speech reconstruction quality. ASR was used for further testing, and the results showed that all the reconstructed speech samples were perfectly recognized while only three samples of the original speech were perfectly recognized.

Keywords: *stuttered speech, speech reconstruction, LPC analysis, LPC synthesis, objective quality measure*

Abstrak

Stuttered speech adalah *speech* yang kaya dysfluency, lebih banyak terjadi pada laki-laki daripada perempuan. Ini terkait dengan tekanan udara yang tidak cukup atau artikulasi yang buruk, meskipun akar penyebabnya lebih kompleks. Fitur utama termasuk *speech* yang berkepanjangan dan berulang-ulang, sementara beberapa fitur sekunder meliputi, kecemasan, ketakutan, dan rasa malu. Penelitian ini menggunakan *LPC analysis* dan *synthesis* algoritma untuk merekonstruksi *stuttered speech*. Hasil dievaluasi menggunakan jarak cepstral, jarak Itakura-Saito, mean square error, dan rasio *likelihood*. Langkah-langkah ini terkandung kualitas *speech reconstruction* yang sempurna. ASR digunakan untuk pengujian lebih lanjut, dan hasilnya menunjukkan bahwa semua sampel *speech* yang terekonstruksi dikenali dengan sempurna sementara hanya tiga sampel dari *speech* asli dikenali dengan sempurna.

Kata Kunci: *stuttered speech, speech reconstruction, LPC analysis, LPC synthesis, objective quality measure*

1. Introduction

The aim of this study is to develop a novel approach for stuttered speech correction using speech reconstruction. Human beings express their feelings, opinions, views and notions orally through speech. Speech includes articulation, voice, and fluency [1,2]. It is a complex naturally acquired human motor skills, an action characterized in normal grownups by the production of about 14 different sounds per second via the coordinated actions of about 100 muscles connected by spinal and cranial nerves. The ease with which human beings speak is in contrast to the complexity of the act, and that complexity may help explain why speech can be exquisitely sensitive to the nervous system associated diseases [3]. Nearly 2% and 5% of adults and children stutter respectively [4,5].

Stuttering can also be defined as a disruption

in the normal flow of speech unintentionally by dysfluencies, which include repetitive pronunciation, prolonged pronunciation, blocked or stalled pronunciation at the phoneme or the syllable level [6-8]. Stuttering cannot be permanently cured, however, it may go into remission after some time, or stutterers can learn to shape their speech into fluent speech with the appropriate speech pathology treatment. This shaping has its effects on the tempo, loudness, effort, or duration of their utterances [7,9].

Stuttering has been found to be more prevalent in males than females (ratio 4:1) [1,2,6,9,10]. Stutterers and non-stutterers alike have speech dysfluencies, which are gaffes or disturbances in the flow of words a speaker plans to say, but dysfluencies are more observable in stutterers' speech [11]. Stuttered speech is rich in dysfluencies, usually repetitions. Classical approaches to the

analysis of dysfluencies are in very short intervals, which is sufficient for recognition of simple repetitions of phonemes [12].

In order to achieve the reconstruction, the linear prediction coefficient (LPC) was used. It was used because its algorithm models the human speech production. The reconstructed speech was then evaluated using objective speech quality measures such as cepstral distance (CD), mean square error (MSE), Itakura-Saito distance (IS) and likelihood ratio (LR). Automatic speaker recognition (ASR) system was developed to further evaluate and compare between the original speech and the reconstructed speech.

2. Methods

The methodologies used for the actualization of this research are described in this section. The LPC analysis and synthesis, the line spectral frequency (LSF) for feature extraction and the multilayer perceptron (MLP) as classifier are explained.

LPC Speech Reconstruction

Linear predictive coding (LPC) is most widely used for medium or low bit-rate speech coders [13]. From each frame of the speech samples, the reflection coefficients are computed. Because important information about the vocal tract model is extracted in the form of reflection coefficients, the output of the LPC analysis filter using reflection coefficients will have less redundancy than the original speech. Thus, less number of bits is required to quantize the residual error. This quantized residual error along with the quantized reflection coefficients are transmitted or stored. The output of the filter, termed the residual error signal, has less redundancy than original speech signal and can be quantized by a smaller number of bits than the original speech. The speech is reconstructed by passing the residual error signal through the synthesis filter. If both the linear prediction coefficients and the residual error sequence are available, the speech signal can be reconstructed using the synthesis filter.

Speech Analysis Filter

Linear Predictive Coding is the most efficient form of coding technique [14, 15] and it is used in different speech processing applications for representing the envelope of the short-term power spectrum of speech. In LPC analysis of order 'p' the current speech sample $s(n)$ is predicted by a linear combination of p past samples k , and given by equation(1) [16].

$$\hat{s}(n) = \sum_{k=1}^p a_p(k).s(n-k) \quad (1)$$

where $\hat{s}(n)$ is the predictor signal and $\{a_p(1), \dots, a_p(p)\}$ are the LPC coefficients. The residual signal $e(n)$ is derived by subtracting $\hat{s}(n)$ from $s(n)$ and the reduced variance is given by the equation-(2).

$$\begin{aligned} e(n) &= s(n) - \hat{s}(n) \\ &= s(n) - \sum_{k=1}^p a_p(k).s(n-k) \end{aligned} \quad (2)$$

By applying the Z-transform to the equation which gives rise to the equation(3).

$$E(z) = A_p(z).S(z) \quad (3)$$

where $S(z)$ and $E(z)$ are the transforms of the speech signal and the residual signal respectively, and $A_p(z)$ is the LPC analysis filter of order 'p' as given by equation(4).

$$A_p(z) = 1 - \sum_{k=1}^p a_p(k) z^{-k} \quad (4)$$

The short-term correlation of the input speech signal is removed by giving an output $E(z)$ with more or less flat spectrum. After implementation of analysis filter, the quantization techniques are implemented and the speech signal is to be brought from the quantized signal at the receiver and so the quantized signal is to be synthesized to get the speech signal.

Speech Synthesis Filter

The short-term power spectral envelope of the speech signal can be modelled by the all-pole synthesis filter as given by equation(5) [16]:

$$H_p(z) = \frac{1}{A_p(z)} = \frac{1}{1 - \sum_{k=1}^p a_p(k) z^{-k}} \quad (5)$$

The equation(5) is the basis for the LPC analysis model. On the other hand, the LPC synthesis model consists of an excitation source $E(z)$, which provides input to the spectral shaping filter $H_p(z)$, which will give the synthesized output speech $S(z)$ as given by equation(6) [14]:

$$S(z) = H_p(z).E(z) \quad (6)$$

In order to identify the sound whether it is voiced or unvoiced, the LPC analysis of each frame can act as a decision-making process. The impulse train is used to represent voiced signal, with non-zero taps occurring for every pitch period. To determine the correct pitch period/frequency, a pitch-detecting algorithm is used. The pitch period can be estimated using autocorrelation function. However, if the frame is unvoiced, then the white noise is used to represent it and a pitch period of $T=0$ is transmitted [14-15].

Therefore, either white noise or impulse train becomes the excitation of the LPC synthesis filter. Hence, it is important to emphasize on the pitch, gain and coefficient parameters that will be varying with time and from one frame to another. The above model is often called the LPC Model. This model speaks about the digital filter (called the LPC filter) whose input is either a train of impulses or a white noise sequence and the output is a digital speech signal [14-15].

Feature Extraction

In general, most speech feature extraction methods fall into the following two categories: modelling the human voice production system or modelling of the peripheral auditory system [17]. Feature extraction consists of computing representations of the speech signal that are robust to acoustic variation but sensitive to linguistic content [18]. It is executed by converting the speech waveform to some type of parametric representation for further analysis and processing. This representation is effective, suitable and discriminative than the original signal [19]. The feature extraction plays a very important role in speech identification. As a result of irregularities in human speech features, human speech can be sensibly interpreted using frequency-time interpretations such as a spectrogram [20].

Line Spectral Frequency (LSF)

Line Spectral Frequency (LSF) exhibits ordering and distortion independence properties. These properties enable the representation of the high frequencies associated with less energy using fewer bits [21]. LSF's are an alternative to the direct form predictor coefficients or the lattice form reflection coefficients for representing the filter response. The direct form coefficient representation of the LPC filters is not conducive to an efficient quantization. Instead, nonlinear functions of the reflection coefficients are often used as transmission parameters. These parameters are preferable because they have a relatively low spectral sensitivity [22]. It has been found that the line sp-

ectral frequency (LSF) representation of the predictor is particularly well suited for quantization and interpolation. Theoretically, this can be motivated by the fact that the sensitivity matrix relating the LSF-domain squared quantization error to the perceptually relevant log spectrum is diagonal [23].

Classification

In order to classify and recognize the eight speakers, an MLP (multilayer perceptron) type of neural network was used. Since neural networks are very good at mapping inputs to target outputs, this feature was used to the advantage of this study. The MLP was used to map the input to the output and it is described below.

Multilayer Perceptron (MLP)

Multilayer perceptron (MLP) is one of many different types of existing neural networks. It comprises a number of neurons connected together to form a network. This network has three layers which are input layer, one or more hidden layer(s) and an output layer with each layer containing multiple neurons [24]. A neural network is able to classify the different aspects of the behaviours, knows what is going on at the instant, diagnoses whether it is correct or faulty, forecasts what it will do next, and if required responds to what it will do next. For an MLP network with b input nodes, one-hidden-layer of c neurons, and d output neurons, the output of the network is given by equation(7) [25-26]:

$$Y_k = \phi_k \left(\sum_{j=1}^c w_{jk} \phi_j \left(\sum_{i=1}^b w_{ij} x_i \right) \right) \quad (7)$$

where ϕ_j and ϕ_k are the activation functions of the hidden-layer neurons and the output neurons, respectively; w_{ij} and w_{jk} are the weights connected to the output neurons and to the hidden-layer neurons, respectively; x_i is the input.

All nodes in one layer are connected with a specific weight to every node in the following layer, without interconnections within a layer. Learning takes place in the perceptron by varying connection weights after each piece of data is processed, based on the quantity of error in the output judged against the anticipated result. This is an example of supervised learning and is achieved through back propagation, a generalization of the least mean squares algorithm [27]. However, a common problem when using MLP is the way to choose the number of neurons in the hidden layer [28].

TABLE 1
SUMMARY OF SAMPLES USED FOR THE ASR

Sample	Stutterer type			
	B	R	BL	I
F1			x	x
F2	x	x	x	x
F3	x	x	x	
F4	x	x	x	
M1	x		x	x
M2	x		x	x
M3	x		x	x
M4	x	x		x

Performance Analysis

Performance analysis is the process of evaluating how the designed system is or would be functioning. By evaluating the system, it is possible to determine if something could be done to speed up a task, or change the amount of memory required to run the task without negatively impacting the overall function of the system. Performance analysis also helps to adjust components in a manner that helps the design make the best use of available resources. The confusion matrix labelling for the computation of the ROC.

The major metrics that are extracted from the confusion matrix are sensitivity, accuracy, specificity, precision, and misclassification rate [29]. Sensitivity (Sen) or recall is a measure of the proportion of actual positives which were correctly identified (true positive rate), accuracy (Acc) is a measure of the degree of closeness of the predicted values to the actual values, precision (Pres) is a measure of repeatability or reproducibility and misclassification rate (MR) is the number of incorrectly identified instances divided by the total number of instances.

3. Results and Analysis

The stuttered speech samples that were obtained for use in this research is the University College London Archive of Stuttered Speech (UCLASS) release 1 database. The recordings of the stuttered speech were collected at University College London (UCL) over a number of years. The recordings are mostly from children who were referred to clinics in London for assessment of stuttering. The Release One recordings have only monolog speech with an age range from 5 years 4 months to 47 years. For the convenience of users, they were prepared in CHILDES, PRAAT, and SFS formats, all of which are freeware available on the Internet. The speech recordings included both male and female speakers. Table 1 shows the eight samples used and the types of stuttering present

TABLE 2
MODIFIED ANALYSIS TOOL

Range		Assigned class
0 - <= 0.10	(0 - <= 10%)	negligible (N)
> 0.10 - <= 0.20	(> 10 - <= 20%)	poor (P)
> 0.20 - <= 0.40	(> 20 - <= 40%)	low (L)
> 0.40 - <= 0.60	(> 40 - <= 60%)	moderate (M)
> 0.60 - <= 0.80	(> 60 - <= 80%)	substantial (S)
> 0.80 - <= 0.90	(> 80 - <= 90%)	Considerable (C)
> 0.90 - <= 1.00	(> 90 - <= 100%)	high (H)

in them. The categories of the stuttering present are burst stuttering (B), reciprocating stuttering (R), blocking stuttering (BL), and interjection (I).

The dysfluencies associated with stuttering can be classed into the following categories [2, 8, 9, 11, 30]:

Bursts stuttering (B)

A syllable is repeated when speaking (“He wa-wa-wa was a good king”) or (caaaaaaaaaaaaaaaaaaake).

Reciprocating stuttering (R)

Some syllables are repeated when speaking (“He ww was a good king”) or (“u-um-um”) or prolonged (“uuuum”) or repeated syllable before pronunciation (“wa wa wa water”).

Blocking stuttering (BL)

A word is difficult to pronounce in a sentence, for a few seconds unsuccessful (“He w—as a good king”).

Interjection (I)

Some interjections are added to the sentence (“I have um, um, a test to-day”) or (“School is, well, fine”) or (“The test was, you know, hard”).

The analysis tool for evaluating the automatic speaker recognition (ASR) systems was a modification of the analysis tool developed by Best in 1981. In order to cater for more distinction at the boundaries of the analysis tool, it was modified to enhance its ability to effectively handle probability values that are exactly on the edges such as 20, 40, 60 and 80. Furthermore, the categories *negligible* and *high* were divided into 2 each. This was done in order to enhance the grouping of probabilities into the two classes and to reduce the band of the two classes. The modifications introduced are described in Table 2.

Objective Measure

The cepstral distance (CD), mean square error (MSE), Itakura-Saito distance (IS) and likelihood ratio (LR) measure between the original speech and the reconstructed speech can be seen in Table

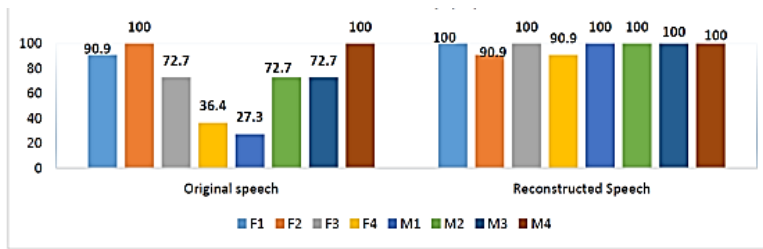


Figure 1. Sensitivity of the ASR.

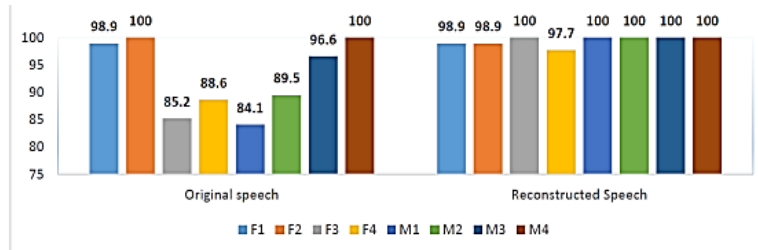


Figure 2. Accuracy of the ASR.

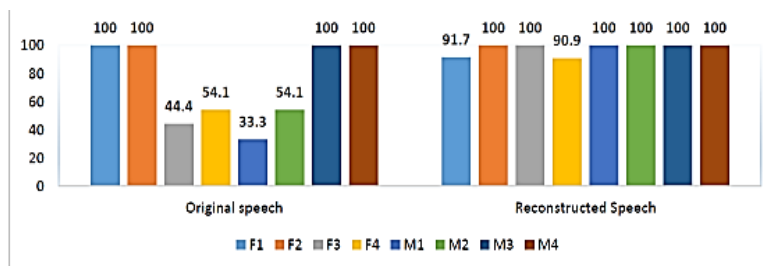


Figure 3. Precision of the ASR

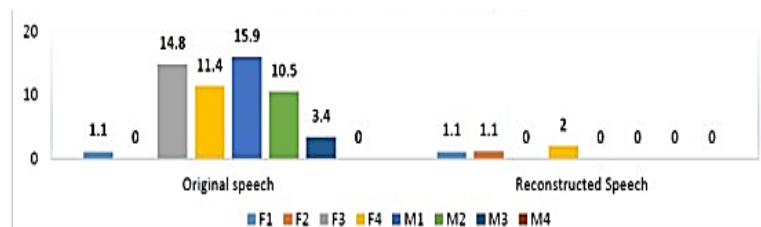


Figure 4. Misclassification Rate of the ASR.

3. The MSE between the original speech and the reconstructed speech for all the speech samples is zero, implying that the reconstruction was perfect with excellent quality of speech and a mirror reflection of the original speech. Similarly, the IS between the original and the reconstructed speech signals was zero. Since the MSE for all samples implied excellent reconstruction, it could be inferred that IS value of zero means perfect reconstruction quality.

The LR for all the 8 samples is also zero. The CD for samples F2, F4, M1 and M4 are not zero, implying that these four samples do not have perfect reconstruction. This, however, is in contrast

to the result interpretation of the other 3 metrics. And because it is known that whenever signal processing techniques are applied to any signal, reversing the process cannot give exactly the same signal as the original signal. Either there is an improvement of the signal or it is degraded.

ASR Evaluation

The LSF-MLP, feature extractor classifier was selected for developing the ASR. The developed ASR was applied on the original stuttered speech signal; this is to serve as a benchmark for the purpose of comparison with the reconstructed spe-

TABLE 3
OBJECTIVE MEASURES OF THE RECONSTRUCTED SPEECH

Sample	CD	MSE	IS	LR
F1	0	0	0	0
F2	0.0291	0	0	0
F3	0	0	0	0
F4	0.0305	0	0	0
M1	0.0499	0	0	0
M2	0	0	0	0
M3	0	0	0	0
M4	0.0882	0	0	0

TABLE 4
SUMMARY OF THE PERFORMANCE ANALYSIS OF LSF-MLP FOR THE ORIGINAL SPEECH

Sample	Sen	Acc	Prec	MR
F1	H	H	H	N
F2	H	H	H	N
F3	S	C	M	P
F4	L	C	M	P
M1	L	C	L	P
M2	S	C	M	P
M3	S	H	H	N
M4	H	H	H	N

ech. The performance metrics that have been discussed above were applied to evaluate the responsiveness of the ASR to the original speech. The performance metrics of systems are plotted in Figures 1-4.

The reconstructed speech was also used with the developed ASR system. Though the reconstructed speech has been evaluated using some objective measure, the methodologies used are just distance measures and the MSE. These distance measures only evaluate the closeness between the original speech and the reconstructed speech. And these measures were not able to effectively differentiate between the original speech and the reconstructed speech. As a result, using ASR for proper evaluation of how the speech would be recognized is compulsory. The results of the performance of the ASR are as discussed below.

For the original speech, the sensitivity of the ASR to samples F1, F2 and M4 was high, while the system sensitivity to F3, M2 and M3 was substantial and F4 and M1 had low sensitivity. The accuracy was high for F1, F2, M3 and M4 and considerable for the remaining four samples. The precision of the system to M1 was low, F3, F4, and M2 were moderate and high for the other samples. The misclassification rate was poor for F3, F4, M1 and M2 and negligible for the other samples.

For the reconstructed speech, the ASR had a sensitivity of group high for all the samples, with only F2 and F4 that below 100%. Similarly, all the accuracies can as well be put in the group high,

TABLE 5
SUMMARY OF THE PERFORMANCE ANALYSIS OF LSF-MLP FOR THE RECONSTRUCTED SPEECH

Sample	Sen	Acc	Prec	MR
F1	H	H	H	N
F2	H	H	H	N
F3	H	H	H	N
F4	H	H	H	N
M1	H	H	H	N
M2	H	H	H	N
M3	H	H	H	N
M4	H	H	H	N

with F1, F2 and F4 slightly below 100%. The values of the precision are also in the group high, with F1 and F4 below 100%. In addition, all the misclassification rates are in the category negligible, while only F1, F2, and F4 have values slightly more than 0%.

Tables 4 and 5 show the summary that reduces the calculations and explanations of Figures 1-4. The hyperbolic tangent sigmoid activation function (tansig) was used for both the hidden layer and the output layer. From Table 5, it can be seen that the ASR excellently senses each input, puts them in their correct classes, with no misfiring. Similarly, all the inputs had very small values for the misclassification rates, implying that almost all the samples were correctly classified. Comparing it with the ASR results of the original speech signals, it would be observed that only the accuracy was very good. The results for the sensitivity and precision had a mixture of the different categories. Also, the misclassification rates are all negligible, with values not as low as they should be. Only 4 of the 8 samples had below 5% while the other 4 had values more than 10%.

4. Conclusion

The use of speech reconstruction for stuttered speech for correcting stuttered speech has been enumerated in this study. Since the LPC algorithm used models the human speech production, the reconstructed speech was very similar to the original speech as interpreted by the objective measures. The ASR gave a better picture of the reconstructed speech as all the speech samples were perfectly recognized while only 3 samples of the original speech were perfectly recognized. Therefore, it could be concluded that the reconstructed speech would be better perceived by the stutterers.

References

- [1] M. Hariharan, V. Vijejan, Y. Chong, and Y. Sazali, "Speech stuttering assessment using sample entropy and Least Square Support Vector Machine," in 8th International Collo-

- quium on Signal Processing and its Applications (CSPA), 2012, pp. 240–245.
- [2] G. Manjula and M. Kumar, “Stuttered Speech Recognition for Robotic Control,” *Int. J. Eng. Innov. Technol.*, vol. 3, no. 12, pp. 174–177, 2014.
- [3] J. Duffy, “Motor speech disorders: clues to neurologic diagnosis,” in *Parkinson’s Disease and Movement Disorders*, Humana Press, 2000, pp. 35–53.
- [4] E. G. Conture and J. S. Yaruss, “Treatment Efficacy Summary,” *Am. speech- language Hear. Assoc.*, no. 1993, p. 20850, 2002.
- [5] C. Oliveira, D. Cunha, and A. Santos, “Risk factors for stuttering in disfluent children with familial recurrence,” *Audiol. Res.*, vol. 18, no. 1, pp. 43–49, 2013.
- [6] L. S. Chee, O. C. Ai, M. Hariaran, and S. Yaacob, “MFCC based recognition of repetitions and prolongations in stuttered speech using k-NN and LDA,” in *2009 IEEE Student Conference on Research and Development and Development (SCOReD)*, 2009, pp. 146–149.
- [7] M. Hariharan, L. S. Chee, and S. Yaacob, “Analysis of infant cry through weighted linear prediction cepstral coefficients and Probabilistic Neural Network,” *J. Med. Syst.*, vol. 36, no. 3, pp. 1309–15, Jun. 2012.
- [8] J. Zhang, B. Dong, and Y. Yan, “A Computer-Assist Algorithm to Detect Repetitive Stuttering Automatically,” in *2013 International Conference on Asian Language Processing (IALP)*, 2013, pp. 249–252.
- [9] S. Awad, “The application of digital speech processing to stuttering therapy,” in *IEEE Sensing, Processing, Networking, Instrumentation and Measurement Technology Conference, IMTC 97*, 1997, pp. 1361–1367.
- [10] L. S. Chee, O. C. Ai, M. Hariharan, and S. Yaacob, “Automatic detection of prolongations and repetitions using LPCC,” in *International Conference for Technical Postgraduates 2009, TECHPOS 2009*, 2009, pp. 1–4.
- [11] K. Hollingshead and P. Heeman, “Using a uniform-weight grammar to model disfluencies in stuttered read speech: a pilot study,” Oregon, 2004.
- [12] J. Pálffy and J. Pospichal, “Pattern search in dysfluent speech,” in *2012 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2012, pp. 1–6.
- [13] I. Mansour and S. Al-Abed, “A New Architecture Model for Multi Pulse Linear Predictive Coder for Low-Bit-Rate Speech Coding,” *Dirasat Eng. Sci.*, vol. 33, no. 2, 2010.
- [14] M. Suman, “Enhancement of Compressed Noisy Speech Signal,” *Koneru Lakshmaiah Education Foundation*, 2014.
- [15] D. Jones, S. Appadwedula, M. Berry, M. Hanun, J. Janovetz, M. Kramer, D. Moussa, D. Sachs, and B. Wade, “Speech Processing: Theory of LPC Analysis and Synthesis,” *Connexions*. June, 2009.
- [16] L. Rabiner and R. Schafer, *Digital processing of speech signals*. Prentice-Hall, 1978.
- [17] Q. P. Li, *Speaker Authentication*. Springer-Verlag Berlin Heidelberg, 2012.
- [18] R. L. Venkateswarlu and R. Vasanthakumari, “Neuro Based Approach for Speech Recognition by using Mel-Frequency Cepstral Coefficients,” *Int. J. Comput. Sci. Commun.*, vol. 2, no. 1, pp. 53–57, 2011.
- [19] C. Cornaz, U. Hunkeler, and V. Velisavljevic, “An automatic speaker recognition system,” *Lausanne, Switzerland*, 2003.
- [20] G. T. Tsenov and V. M. Mladenov, “Speech recognition using neural networks,” in *Neural Network Applications in Electrical Engineering (NEUREL)*, 2010 10th Symposium on, 2010, pp. 181–186.
- [21] V. Namburu, “Speech Coder Using Line Spectral Frequencies of Cascaded Second Order Predictors,” *VirginiaTech*, 2001.
- [22] P. Kabal and R. Ramachandran, “The computation of line spectral frequencies using Chebyshev polynomials,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 34, no. 6, pp. 1419–1426, 1986.
- [23] W. B. Kleijn, T. Bäckström, and P. Alku, “On line spectral frequencies,” *IEEE Signal Process. Lett.*, vol. 10, no. 3, pp. 75–77, 2003.
- [24] R. Kumar, R. Ranjan, S. K. Singh, R. Kala, A. Shukla, and R. Tiwari, “Multilingual Speaker Recognition Using Neural Network,” in *Proceedings of the Frontiers of Research on Speech and Music, FRSM*, 2009, pp. 1–8.
- [25] M. A. Al-Alaoui, L. Al-Kanj, J. Azar, and E. Yaacoub, “Speech recognition using artificial neural networks and hidden Markov models,” *IEEE Multidiscip. Eng. Educ. Mag.*, vol. 3, no. 3, pp. 77–86, 2008.
- [26] S. S. Haykin, *Neural networks and learning machines*, vol. 3. Pearson Education Upper Saddle River, 2009.
- [27] S. Agrawal, A. K. Shruti, and C. R. Krishna, “Prosodic feature based text dependent speaker recognition using machine learning algorithms,” *Int. J. Eng. Sci. Technol.*, vol. 2, no. 10, pp. 5150–5157, 2010.
- [28] C. L. Tan and A. Jantan, “Digit recognition using neural networks,” *Malaysian J. Comput. Sci.*, vol. 17, no. 2, pp. 40–54, 2004.
- [29] R. Kohavi and F. Provost, “Glossary of terms,” *Mach. Learn.*, vol. 30, no. 2–3, pp.

- 271–274, 1998.
- [30] M. Hariharan, L. S. Chee, O. C. Ai, and S. Yaacob, “Classification of speech dysfluencies using LPC based parameterization techniques,” *J. Med. Syst.*, vol. 36, no. 3, pp. 1821–1830, 2012.

DOCUMENT CLUSTERING BY DYNAMIC HIERARCHICAL ALGORITHM BASED ON FUZZY SET TYPE-II FROM FREQUENT ITEMSET

Saiful Bahri Musa, Andi Baso Kaswar, Supria, and Susiana Sari

Department of Informatics Engineering, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember, Jl. Teknik Kimia, Gedung Teknik Informatika, Surabaya, 60111, Indonesia

E-mail: saiful14@mhs.if.its.ac.id

Abstract

One of ways to facilitate process of information retrieval is by performing clustering toward collection of the existing documents. The existing text documents are often unstructured. The forms are varied and their groupings are ambiguous. This cases cause difficulty on information retrieval process. Moreover, every second new documents emerge and need to be clustered. Generally, static document clustering method performs clustering of document after whole documents are collected. However, performing re-clustering toward whole documents when new document arrives causes inefficient clustering process. In this paper, we proposed a new method for document clustering with dynamic hierarchy algorithm based on fuzzy set type-II from frequent item set. To achieve the goals, there are three main phases, namely: determination of keyterm, the extraction of candidates clusters and cluster hierarchical construction. Based on the experiment, it resulted the value of F-measure 0.40 for Newsgroup, 0.62 for Classic and 0.38 for Reuters. Meanwhile, time of computation when addition of new document is lower than to the previous static method. The result shows that this method is suitable to produce solution of clustering with hierarchy in dynamical environment effectively and efficiently. This method also gives accurate clustering result.

Key Words: *Dynamic Hierarchical Algorithm, Fuzzy Set Type-II, Document Clustering.*

Abstrak

Salah satu cara untuk mempermudah proses information retrieval adalah dengan melakukan pengklasteran terhadap koleksi dokumen yang ada. Dokumen teks yang ada seringkali tidak terstruktur, formatnya bervariasi, dan pengelompokannya ambigu. Hal ini menimbulkan kesulitan dalam proses information retrieval. Selain itu, setiap detik dokumen baru bertambah dan perlu untuk dikelompokkan. Pada umumnya, metode pengklasteran dokumen statis melakukan pengklasteran dokumen setelah keseluruhan dokumen terkumpul. Namun, melakukan pengklasteran ulang terhadap keseluruhan dokumen ketika dokumen baru tiba mengakibatkan proses pengklasteran menjadi tidak efisien. Penelitian ini mengusulkan metode baru untuk pengklasteran dokumen dengan algoritma hierarki dinamis berbasis fuzzy set type-II dari frequent itemset. Untuk mencapai tujuan tersebut, terdapat 3 tahapan utama yang akan dilakukan, yaitu; ekstraksi keyterm, ekstraksi kandidat kluster dan pembangunan hirarki kluster. Berdasarkan eksperimen yang telah dilakukan diperoleh nilai F-Measure 0,40 untuk Newsgroup, 0,62 untuk Classic, dan 0,38 untuk Reuters. Sedangkan waktu komputasi pada saat penambahan dokumen dapat direduksi dibanding dengan metode statis sebelumnya. Hasil percobaan terhadap beberapa dataset koleksi dokumen menunjukkan bahwa metode ini tidak hanya sesuai untuk menghasilkan solusi pengklasteran secara hirarki dalam lingkungan yang dinamis secara efektif dan efisien, tetapi juga memberikan hasil pengklasteran yang akurat.

Kata Kunci: *Algoritma Hirarki Dinamis, Fuzzy Set Tipe-II, Pengklasteran Dokumen.*

1. Introduction

Plentiful information as a result from development of information technology is a big advantage for the seeker of information. However, at the same time the big problem appears as a result of the increase of exist data where it's difficult to determine needed information from large quantity of unnecessary/unimportant data. So, information retrieval

(IR) and information extraction (IE) are present to handle the problem. Managing, accessing, searching, and big browsing repository from text document need efficient organizing from the information. In that case, document clustering have important role as tool that organize document collection to be meaningful cluster collection to increase information retrieval efficiency and document management [1].

However, the increasing of text document explosively on the internet and must be clustered generally have unstructured form and their groupings are ambiguous. So they cause the difficulties in seeking and managing document. One of methods that functions to organize document collection is document clustering hierarchically. Document clustering into structure of tree hierarchically is able to increase efficiency of IR [2–4]. However, there are some challenges in hierarchy document clustering, namely; high dimensionality, scalability, accuracy, easy of browsing and meaningful cluster label [2-5].

Some researchers [2–4,6] use frequent item-set from association rule for document management. The method is able to solve the problem like reduction of dimension, input of cluster amount and the ease of seeking by meaningful label. Next, Chen et al [4] show that Fuzzy Frequent Item-set-based on Hierarchical Clustering method (F2IHC) can avoid overlapping cluster and increase the accuracy of document clustering result. However the method uses fuzzy set type-1. Fuzzy set type-1 that has function of distinct membership is not able to model uncertainty directly [7].

On the other hand, method of fuzzy set type-2 has interval membership function which is able to model uncertainty in defining membership function on fuzzy set type-1 [7-8]. Then, Sari et al [9] build a document clustering method hierarchically

based on fuzzy set type-2 from frequent item-set to increase the quality of clustering result. Fuzzy set type-2 trapezoidal as upper membership function and fuzzy type-II triangular as bottom membership function toward frequent item-set that gotten from association rule mining to increase the accuracy of document clustering. The proposed method is able to produce qualified cluster and to solve the ambiguity. However the recommended method is only implemented in the static document. In environment of dynamic information like World Wide Web is necessary to apply adaptive method for organizing of document.

Static clustering methods generally do clustering where the entire document had been ready before applying clustering algorithms. When adding a new document, it is necessary to do re-clustering toward the whole documents. However, performing re-clustering toward the whole documents when new documents come afterward, caused the process of clustering are not efficient.

In this paper, we proposed a new method for document clustering with Dynamic Hierarchical Algorithm based on Fuzzy Type-II from the Frequent Item-set. Type-II fuzzy sets used to overcome the problem of ambiguity and dynamic clustering method with dynamic hierarchical algorithm can process documents added or eliminated to or from the collection. Dynamic algorithm has the capability to renew clustering when data is added or

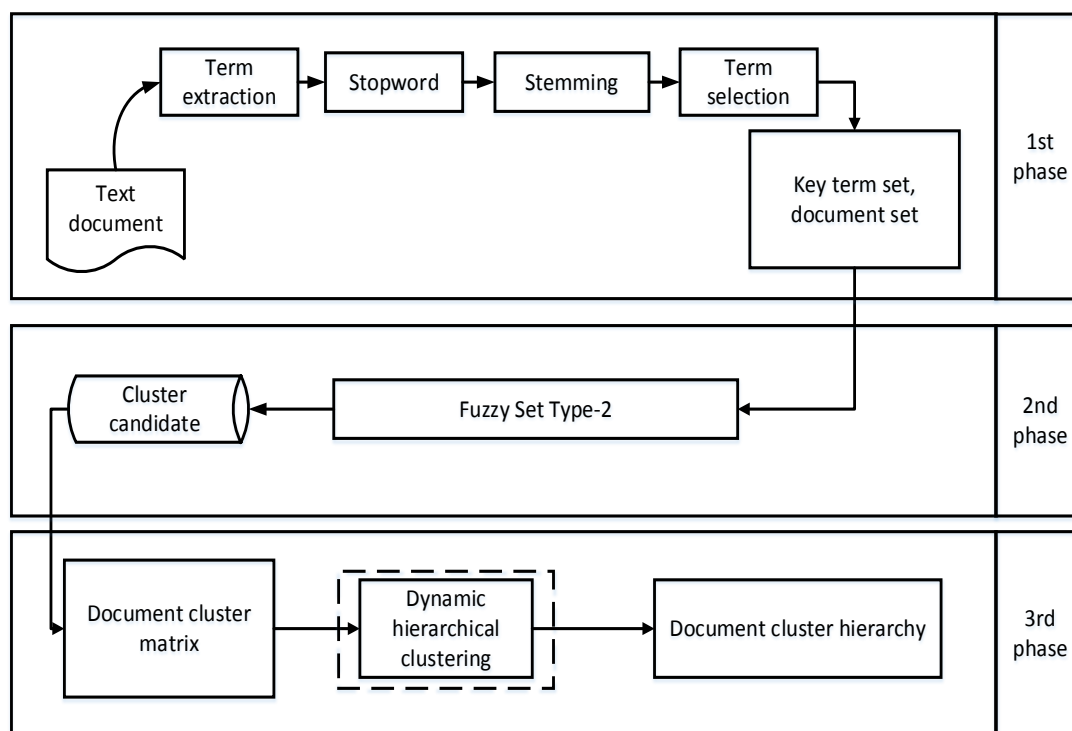


Figure 1. Document clustering phase.

TABLE 1
FUZZY MEMBERSHIP FUNCTION TYPE-II TRAPEZOIDAL AND TRIANGULAR DEFENITION.

Upper Membership Function (UMF)		Lower Membership Function (LMF)	
$W_{ij}^{L,U}(f_{ij}) = \begin{cases} 1, & a \leq f_{ij} \leq b \\ \frac{(f_{ij} - c)}{(b - c)}, & b \leq f_{ij} \leq c \\ 0, & f_{ij} \geq c \end{cases}$	$a, d, b_1 = \min(f_{ij})$ $b = \frac{(\min(f_{ij}) + c_1)}{2}$	$W_{ij}^{L,L}(f_{ij}) = \begin{cases} \frac{(f_{ij} - a_1)}{(b_1 - a_1)}, & a_1 \leq f_{ij} \leq b_1 \\ \frac{(f_{ij} - c_1)}{(b_1 - c_1)}, & b_1 \leq f_{ij} \leq c_1 \end{cases}$	$a_1 = 0$
$W_{ij}^{M,U}(f_{ij}) = \begin{cases} \frac{(f_{ij} - d)}{(e - d)}, & d \leq f_{ij} \leq e \\ 1, & e \leq f_{ij} \leq f \\ \frac{(f_{ij} - g)}{(f - g)}, & f \leq f_{ij} \leq g \end{cases}$	$c, h = \text{avg}(f_{ij})$ $e = \frac{(c_1 + \text{avg}(f_{ij}))}{2}$ $f = \frac{(\text{avg}(f_{ij}) + f_1)}{2}$	$W_{ij}^{M,L}(f_{ij}) = \begin{cases} \frac{(f_{ij} - d_1)}{(e_1 - d_1)}, & d_1 \leq f_{ij} \leq e_1 \\ \frac{(f_{ij} - f_1)}{(e_1 - f_1)}, & e_1 \leq f_{ij} \leq f_1 \end{cases}$	$c_1, d_1 = \frac{(\min(f_{ij}) + \text{avg}(f_{ij}))}{2}$ $e_1 = \text{avg}(f_{ij})$
$W_{ij}^{H,U}(f_{ij}) = \begin{cases} \frac{(f_{ij} - h)}{(i - h)}, & h \leq f_{ij} \leq i \\ 1, & i \leq f_{ij} \leq j \end{cases}$	$g, j, h_1 = \max(f_{ij})$ $i = \frac{(g_1 + \max(f_{ij}))}{2}$	$W_{ij}^{H,L}(f_{ij}) = \begin{cases} 0, & f_{ij} \leq g_1 \\ \frac{(f_{ij} - g_1)}{(h_1 - g_1)}, & g_1 \leq f_{ij} \leq h_1 \end{cases}$	$f_1, g_1 = \frac{(\text{avg}(f_{ij}) + \max(f_{ij}))}{2}$

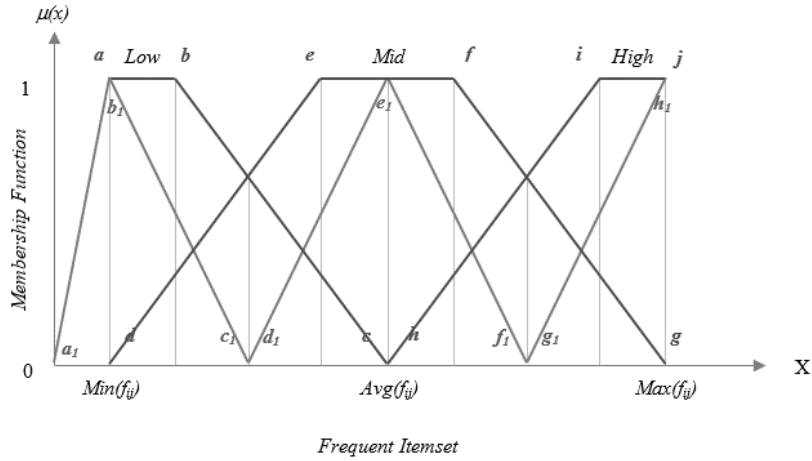


Figure. 2. Fuzzy membership function type-II trapezoidal and triangular.

eliminated from the collection. This algorithm enables us dynamically to track the large-scale of information constantly changing, either inserted to the web every day, without having to perform a complete clustering. The proposed method is expected to perform dynamical documents clustering so re-clustering toward all existing documents are not necessary to do as well as overcoming the problem of ambiguity so it can provide accurate clustering results at the same time.

2. Methods

There are several phases are done for document clustering with dynamic hierarchical clustering algorithm, namely: keyterm extraction, extracion of candidate clusters and cluster construction (Figure 1). Keyterm extraction is a process to obtain the most representative terms for the document. Ex-

traction of cluster candidate is a process done to give value of fuzzy for frequent item-set and to get 1-itemset candidate. Cluster construction is a process to build a cluster hierarchy for frequent item-set.

Keyterm Extraction

The first phase is the process of keyterm extraction. The aim of keyterm extraction is to get the most representative term to be input for cluster candidate extraction phase. The input of this phase is the collection of document that will be clustered, while the output of this phase is in the keyterms. The procedure that must be done in this phase is the extraction of term, stop word removal, stemming, and term selection.

Term extraction is the process of term extraction from a document collection that is denoted by

$$W = \begin{matrix} & t_1 & t_2 & \dots & t_p \\ \begin{matrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{matrix} & \begin{bmatrix} W_{11}^{max-R_j} & W_{12}^{max-R_j} & \dots & W_{1p}^{max-R_j} \\ W_{21}^{max-R_j} & W_{22}^{max-R_j} & \dots & W_{2p}^{max-R_j} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n1}^{max-R_j} & W_{n2}^{max-R_j} & \dots & W_{np}^{max-R_j} \end{bmatrix} & \end{matrix} \quad n \times p$$

Figure 3. Document Term Matrix.

$$G = \begin{matrix} & \tilde{c}_1 & \dots & \tilde{c}_2 & \tilde{c}_3 & \dots & \tilde{c}_k \\ \begin{matrix} t_1 \\ t_2 \\ \vdots \\ t_p \end{matrix} & \begin{bmatrix} g_{11}^{max-R_j} & \dots & g_{12}^{max-R_j} & g_{13}^{max-R_j} & \dots & g_{1k}^{max-R_j} \\ g_{21}^{max-R_j} & \dots & g_{22}^{max-R_j} & g_{23}^{max-R_j} & \dots & g_{2k}^{max-R_j} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g_{p1}^{max-R_j} & \dots & g_{p2}^{max-R_j} & g_{p3}^{max-R_j} & \dots & g_{pk}^{max-R_j} \end{bmatrix} & \end{matrix} \quad p \times k$$

Figure 4. Term-Cluster Matrix.

($D = \{d_1, d_2, d_3, \dots, d_n\}$) in order to obtain the set of term $T_D = \{t_1, t_2, t_3, \dots, t_n\}$. The set of term T_D still contains common words (stop word) such as "and, with, what, etc.", so it's needed the stop word-removal process to eliminated the stop word. Stop word removal is used contains 571 words in English.

Stemming process is performed then on the remaining term by stemming WordNet 2.0. This process aims to return the exist word into their basic form. To get the most representative keyterm, the term selection is based on calculations $tf \cdot idf$, $tf \cdot df$ and tf^2 . Term which has value more than the minimum threshold value $tf \cdot idf$ (α), minimum threshold $tf \cdot df$ (β), and minimum threshold tf^2 (γ) are defended as a set of keyterms. To get the value of keyterms set, we use equation(1) for $tf \cdot idf$, equation(2) for $tf \cdot df$ and equation(3) for tf^2 .

$$tf \cdot idf_{ij} = \frac{f_{ij}}{\sum_{j=1}^m f_{ij}} * \log \left(\frac{|D|}{|\{d_i | t_j \in d_i, d_i \in D\}|} \right) \quad (1)$$

$$tfdf_{ij} = TF * DF, \quad (2)$$

$$\text{where } TF = \frac{f_{ij}}{\sum_{j=1}^m f_{ij}}, DF = \frac{|\{d_i | t_j \in d_i, d_i \in D\}|}{|D|}$$

$$tf_{ij}^2 = tfidf_{ij} * tfdf_{ij} \quad (3)$$

Cluster Candidate Extraction

The second phase is performing the cluster candidate extraction to provide value of fuzzy for frequent item set and get 1-itemset candidate. In giving

fuzzy value for frequent item-set, using membership functions of fuzzy set type-II as the upper trapezoidal membership function (UMF) and triangular membership function as lower membership function (LMF) will be optimal (Figure 2).

Term-frequency (tf) fuzzy set of documents d_i denoted as $(F_{ij}, W_{ij}^{r,z})$. F value has the range $[0, 1]$. In F_{ij} consists of three regionals, namely Low (L), MID (M) and High (H). F_{ij} denoted as $\{W_{ij}^{L,U}(F_{ij})/t_j \cdot L \cdot U, W_{ij}^{M,U}(F_{ij})/t_j \cdot M \cdot U, W_{ij}^{H,U}(F_{ij})/t_j \cdot H \cdot U\}$, $\{W_{ij}^{L,L}(F_{ij})/t_j \cdot L \cdot L, W_{ij}^{M,L}(F_{ij})/t_j \cdot M \cdot L, W_{ij}^{H,L}(F_{ij})/t_j \cdot H \cdot L\}$. $t_j \cdot r \cdot z$

are regional fuzzy t_j . Z can serve as UMF (U) or LMF (L). For a term (t_j, f_{ij}) in the document (d_i), $W_{ij}^{r,z}(f_{ij})$ is the membership de-gree t_j in d_i which is defined in Table 1.

Where $\min(f_{ij})$ is the minimum frequency term in D , $\max(f_{ij})$ is the term maximum frequency of the term in D and $avg(f_{ij}) = (\min(f_{ij}) + \max(f_{ij}))/2$.

Fuzzy frequent item-set which has higher support value than minimum support value will take into account as candidate 1-itemset. The support value calculation of a term derived from the ration between the value of the fuzzy frequent itemset and the amount document. Candidate cluster (\tilde{C}) can be obtained from the collection of documents (D). \tilde{C} may also be denoted as $\tilde{C}_{(t_1, t_2, \dots, t_q)}$ or $\tilde{C}_{(\tau)}$. Candidate cluster has 2-tuples denoted as $\tilde{c} = (\tilde{D}_c, \tau)$, where \tilde{D}_c is part of D and τ is the fuzzy frequent item-set to describe \tilde{c} . τ is denoted as $\tau = \{t_1, t_2, \dots, t_q\} \subseteq K_D$. K_D is collection of keyterms

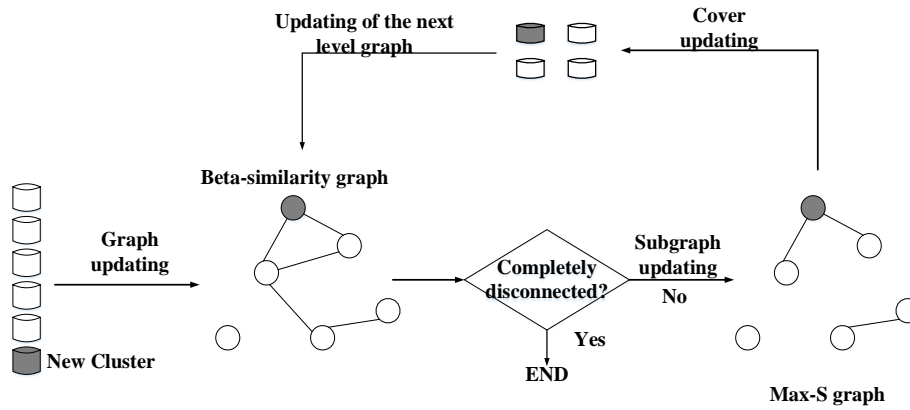


Figure 5. Dynamic hierarchical algorithm.

D and q are the number of keyterms that are included in τ . Collection of cluster candidate is denoted as $\tilde{C}_D = \{\tilde{c}_1, \tilde{c}_2, \tilde{c}_3, \dots, \tilde{c}_k\}$, where k is the total amount of cluster candidate.

Cluster Hierarchy Construction

The third phase is the tree cluster construction. The process consists of two steps, namely: Document-Cluster Matrix (DCM) and construct a cluster hierarchy using dynamic hierarchical clustering algorithm. DCM functions to place each document to the proper cluster, so each c_i^q contains subset of the document. To achieve this goal, de-fined two matrices, that is Document-Term Matrix (DTM) and Term-Cluster Matrix (TCM). DTM (W) is the weight of term t_j in the document d_i and $t_j \in L_1$. The first step is to consider each candidate cluster $\tilde{c}_{(\tau)} = \tilde{c}_{(t_1, t_2, \dots, t_q)}$ by fuzzy frequent item-set τ . τ is considered as a reference to generate target of cluster. To present the importance of document in the candidate cluster \tilde{c}_1 , then calculated the similarity of the terms in d_i and \tilde{c}_1 (Figure 3).

$W_{ij}^{max-R_j}$ is the weight of term t_j in the document $d_i \in \tilde{c}_1$ and λ is the minimum value of the confidence. TCM (G) is a matrix $p \times k$ (Figure 4). TCM for $1 \leq j \leq p, 1 \leq l \leq k$ calculated by using equation (4) where score calculated by using equation (5).

$$g_{jl}^{max-R_j} = \frac{score(\tilde{c}_l^q)}{\sum_{i=1}^n W_{ij}^{max-R_j}} \quad (4)$$

Each $g_{jl}^{max-R_j}$ in TCM presents the degree of importance of keyterms t_j in a candidate cluster $\tilde{c}_{(\tau)}$ by referring to all documents which have z .

In cluster construction, the proposed framework is an agglomerative method based on Figure 5 [10] where consist of two graphs. In the framework, we use the vertex as the cluster of document.

$$score(\tilde{c}_l^q) = \left\{ \begin{array}{l} \sum_{d_i \in \tilde{c}_l^q, t_j \in L_1} W_{ij}^{max-R_j} \text{ if } q = 1, \\ \frac{\sum_{d_i \in \tilde{c}_l^q, t_j \in L_1} W_{ij}^{max-R_j}}{\lambda}, \text{ else} \end{array} \right\} \quad (5)$$

$$\beta_{Sim}(c_x, c_y) = \frac{\sum_{d_1 \in c_x, c_y} v_{ix} \times v_{iy}}{\sqrt{\sum_{d_1 \in c_x} (v_{ix})^2 \times \sum_{d_1 \in c_y} (v_{iy})^2}} \quad (6)$$

The first graph is an undirected graph, where the vertices are the cluster and there is an edge between node i and j . Furthermore, the graph is called β -similarity graph. In this graph, an edge is formed between vertex i and j , if the vertex j is β -similar to the vertex i . Two clusters are β -similar if the similarity of both is greater than or equal to β , where β is determined parameter by the user who presents a minimum similarity threshold. β -cluster similarity between two target clusters c_x and $c_y, c_x \neq c_y$, is defined by equation(6).

The second graph is called max- S graph. Max- S graph relies on the maximum β -similarity relationship and it is a sub-graph of the first one. Vertices of the graph is the same as vertices in the graph β -similarity. Vertices i and j given the edge, if cluster i is the most β -similiar to cluster j . The use of Max- S graph not only reduce time and room utilize (because it has a little edges) but also produce dense cluster.

Being given a cluster hierarchy that previously was created using the algorithm. If there is a new document addition of the cluster, the cluster at all levels of the hierarchy should be revised. When a new document arrives, singleton will be created and β -similarity graph at the bottom level is updated.

Then update the max- S graph, where this process can produce or remove a vertex and can also produce a new edge and remove the others. Let N be the set of cluster to add to max- S graph. Add all

TABLE 2
F-MEASURE RESULT

Addition process	Classic		Reuters		NewsGroup	
	Proposed	Static method	Proposed	Static method	Proposed	Static method
1	0.62	0.62	0.39	0.39	0.39	0.39
2	0.62	0.62	0.39	0.39	0.40	0.39
3	0.62	0.62	0.38	0.38	0.40	0.39

TABLE 3
TIME EXECUTION

Addition process	Classic		Reuters		NewsGroup	
	Proposed (s)	Static method (s)	Proposed (s)	Static method (s)	Proposed (s)	Static method (s)
1	98	149	1229	1316	979	940
2	36	231	403	1608	357	1303
3	54	252	573	2207	615	1877

vertices of \mathbf{N} to max- \mathbf{S} graph. Find the most β -similar vertices of each vertex in \mathbf{N} and add the corresponding edges to max- \mathbf{S} graph. Find all vertices for which a vertex in \mathbf{N} is its most β -similar and update the corresponding edges. The value of β -similarity the same for all levels of hierarchy levels.

A cover routine applied to the max- \mathbf{S} graph to renew cluster. Let \mathbf{N} be the set of vertices added to the max- \mathbf{S} graph. Let, also, \mathbf{NE} be the set of edges added to the max- \mathbf{S} graph. Let \mathbf{Q} be a queue with the vertices to be processed, $\mathbf{Q} \neq \emptyset$. Put the remaining vertices of the clusters into \mathbf{Q} . Remove these clusters from the list of the existing clusters. Put all vertices of \mathbf{N} into the queue \mathbf{Q} . Build the connected components from the vertices in \mathbf{Q} and add them to the list of existing clusters. For each edge of \mathbf{NE} , merge the clusters to which its vertices belong. When a cluster is created on the hierarchy level, β -similarity graph next level should be updated. This process is repeated until the graph completely disconnected.

3. Results and Analysis

The implementation of the method is supported by Intel® Core™ i3-2120 CPU @ 3.30GHz (4CPUs) processor, with a RAM of 4 GB and Java Development Kit 6 update 31 with Netbeans IDE 8.0.1. Regarding the evaluation of the method, we used 1000 documents from Classic, 1930 documents from Reuters and 1000 document from Newsgroup. Therefore the evaluation of the proposed method is based on the scalability and F-Measure of the hierarchical cluster.

Overall F-Measure will be measured using equation(7).

$$F(C) = \sum_{l_j \in L} \frac{|l_j| \max\{F\}}{|D| c_{i \in C}}, \quad (7)$$

where $|D|$ is the amount of all document in dataset D . C is a cluster obtained from system. L is the class label that obtained from dataset. $|c_i|$ is the amount of document in cluster C . $|l_j|$ is the amount of document in class L . F is the F-measure, P is the precision and R is recall obtained from equation(8) to equation(10) respectively.

$$F = \frac{2PR}{P+R} \quad (8)$$

$$P = \frac{|c_i \cap l_j|}{|c_i|} \quad (9)$$

$$R = \frac{|c_i \cap l_j|}{|l_j|} \quad (10)$$

In the keyterm extraction process, the amount of keyterms depends on the minimum threshold. The keyterms that has tf , idf , $tf \cdot df$, and tf^2 more than threshold value will be consider as the keyterm. The obtained keyterms are supposed to be the representative keyterm toward document collection.

In the candidate extraction phase of the clusters using fuzzy set type-II and with *minsup* higher than 20%, we obtained 4 clusters from the dataset Classic, 9 clusters from Reuters and 15 from Newsgroup. If the *minsup* used is too low, it will obtained many cluster candidate. Therefore, it is possible to results a low clustering accuracy. On the other hand, if we use high *minsup* value, it is possible that it will obtained a little cluster candidate which doesn't represent the whole documents within the document collection.

In the cluster construction phase, the graph that was formed was a vertex that was obtained using type-II fuzzy sets. First of all, all documents that were placed in the various cluster, uses the DCM method. Thus, from the above process, each of the cluster candidates have got their own members. Based on the result of clustering and construction of the hierarchy structure, we obtained the values of F-Measure as mention in Table 2. The experiment was done by doing the process of documents addition periodically. Based on the experiments that were done, we obtained value F-Measure as much as 0.40 from Newsgroup, 0.62 from classic and 0.38 from Reuters after clustering the whole documents. The F-Measure value decreased in each additional document process because the more documents, the more possibility of cluster formed. Which means, each document will occupy a cluster that is not supposed.

To show the dynamic and efficiency of the proposed method, documents clustering was initially carried out. Next, with the assumption that there are new documents, these new documents are clustered immediately. From Table 3, we can see that in every clustering process, the proposed method can reduce document clustering processing time consideredly when compared with the static method which means, the proposed method can improve the efficiency of the document clustering method in term of time execution if there are new documents.

The proposed method can performs clustering process faster than the previous method because the proposed method only apply the whole processes toward the document comes afterward then find the similarity value toward the existing clusters (vertex). Therefore, the document that has just been added can join in cluster that has been formed before or form new hierarchy. While the previous method performs clustering toward the whole documents either the document comes afterward or the document that has been clustered before.

Beside that we can also know, the proposed method can give value F-measure as good as the method before. This can happen because of the obtained result from cluster candidate extraction using fuzzy set type-II that overcome ambiguity problem and each exist document has been joined in its own cluster. Therefore, we can know that the proposed method is effective and more efficient in clustering the document.

For addition, proposed method can improve the efficiency in terms of RAM usage. In every clustering process/when the the addition of new documents, RAM usage increases only in small amounts. Whereas conventional method require larger RAM allocation. For example, in the first iteration of the clustering process for 500 documents Class-

ic, it is requires allocation of 18.98 MB RAM by using the proposed method and 19.09 MB by using conventional method. Then we added 200 new documents in the second iteration. RAM usage using the proposed method only increased by 4 MB. While conventional method increased about 20.19 MB.

4. Conclusion

In this paper we proposed a new document clustering method by dynamic hierarchical algorithm based on fuzzy set type II from frequent item-set. Dynamic hierarchical algorithm used to perform dynamic document clustering and fuzzy set type II used to solve ambiguity problems when clustering. From the obtained results, the proposed method can improve efficiency in terms of time of clustering and RAM usage because the proposed method only apply the whole processes toward the document comes afterward then find the similarity value toward the existing clusters (vertex). Besides that, it can be seen that the proposed method gives good accuracy of the clustering.

Reference

- [1] T. M. Nogueira, H. A. Camargo, S. O. Rezende, R. W. Luís, and S. Carlos-sp, "Fuzzy Rules for Document Classification to Improve Information Retrieval," *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 3, pp. 210–217, 2011.
- [2] F. Beil, M. Ester, B. Bc, and C. Va, "Frequent Term-Based Text Clustering," 2002.
- [3] B. C. M. Fung, "Hierarchical Document Clustering using Frequent Itemsets," 2002.
- [4] C.-L. Chen, F. S. C. Tseng, and T. Liang, "Mining Fuzzy Frequent Itemsets for Hierarchical Document Clustering," *Inf. Process. Manag.*, vol. 46, no. 2, pp. 193–211, 2010.
- [5] J. Han and M. Kamber, *Data Mining Concept and Techniques*. 2006.
- [6] T. Hong, K. Lin, and S. Wang, "Fuzzy Data Mining for Interesting Generalized Association Rules," *Fuzzy Sets Syst.*, vol. 138, pp. 255–269, 2003.
- [7] J. M. Mendel and R. I. B. John, "Type-2 Fuzzy Sets Made Simple," *Fuzzy Syst. IEEE Trans.*, vol. 10, no. 2, pp. 117–127, 2002.
- [8] J. T. Starczewski, "Efficient Triangular Type-2 Fuzzy Logic Systems," *Int. J. Approx. Reason.*, vol. 50, no. 5, pp. 799–811, 2009.
- [9] S. Sari and A. Z. Arifin, "Clustering Dokumen Secara Hierarki Berbasis Fuzzy Set Tipe-2 Trapezoidal dan Triangular dari Frequent Itemset," in *Prosiding Seminar Nasio-*

- nal Manajmemen Teknologi XVI*, 2012, pp. 1–8.
- [10] R. Gil-garcía and A. Pons-porrata, “Dynamic Hierarchical Algorithms for Document Clustering,” *Pattern Recognit. Lett.*, vol. 31, no. 6, pp. 469–477, 2010.

FRACTAL DIMENSION AND LACUNARITY COMBINATION FOR PLANT LEAF CLASSIFICATION

Mutmainnah Muchtar, Nanik Suciati, and Chastine Fatichah

Department of Infomatics Engineering, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember Surabaya, Jl. Teknik Kimia, Gedung Teknik Informatika, Kampus ITS Sukolilo, Surabaya, 60111, Indonesia

E-mail: muchtarmutmainnah@gmail.com, nanik@if.its.ac.id, chastine@if.its.ac.id

Abstract

Plants play important roles for the existence of all beings in the world. High diversity of plant's species make a manual observation of plants classifying becomes very difficult. Fractal dimension is widely known feature descriptor for shape or texture. It is utilized to determine the complexity of an object in a form of fractional dimension. On the other hand, *lacunarity* is a feature descriptor that able to determine the heterogeneity of a texture image. *Lacunarity* was not really exploited in many fields. Moreover, there are no significant research on fractal dimension and *lacunarity* combination in the study of automatic plant's leaf classification. In this paper, we focused on combination of fractal dimension and *lacunarity* features extraction to yield better classification result. A box counting method is implemented to get the fractal dimension feature of leaf boundary and vein. Meanwhile, a gliding box algorithm is implemented to get the *lacunarity* feature of leaf texture. Using 626 leaves from *flavia*, experiment was conducted by analyzing the performance of both feature vectors, while considering the optimal box size r . Using support vector machine classifier, result shows that combined features able to reach 93.92 % of classification accuracy.

Keywords: *leaf classification, fractal dimension, lacunarity, box counting, gliding box*

Abstrak

Tumbuhan memegang peranan penting dalam kehidupan manusia. Tingginya keberagaman spesies tumbuhan membuat metode pengamatan manual dalam klasifikasi daun menjadi semakin sulit. Dimensi fraktal merupakan deskriptor bentuk dan tekstur yang mampu mendeskripsikan kompleksitas dari suatu objek dalam bentuk dimensi pecahan. Di sisi lain, *lacunarity* adalah deskriptor tekstur berbasis fraktal yang mampu mendeskripsikan heterogenitas dari citra tekstur. Namun *lacunarity* belum cukup dieksplorasi dalam banyak kasus dan belum ada usaha yang cukup signifikan dalam mengkombinasikan dimensi fraktal dan *lacunarity* dalam bidang klasifikasi tumbuhan secara otomatis. Penelitian ini berfokus pada ekstraksi dan kombinasi fitur dimensi fraktal dan *lacunarity* untuk meningkatkan akurasi klasifikasi. Metode *box counting* diterapkan untuk memperoleh dimensi fraktal dari bentuk pinggir dan urat daun, sementara metode *gliding box* diterapkan untuk memperoleh fitur *lacunarity* dari tekstur daun. menggunakan 626 citra daun dari *flavia*, percobaan dilakukan dengan menganalisis performa dari kedua fitur dengan mempertimbangkan ukuran kotak r yang paling optimal. Klasifikasi dengan *support vector machine* menunjukkan bahwa hasil kombinasi kedua fitur mampu mencapai rata-rata akurasi hingga 93.92%.

Kata Kunci: *klasifikasi daun, dimensi fraktal, lacunarity, box counting, gliding box*

1. Introduction

Science associated with plant identification and classification plays an important role in many fields that affects human life, including in the fields of food and agriculture, medicine, industry, environment, and so on. Plant Morphology is a study that focuses on how to examine and identify a plant based on its physical characteristics that can be seen with human naked eyes. With the rising number of plant species acknowledged today, it is im-

portant to protect the plants or collect them in the form of information that offers diversity of flora. The introduction of computer-based plant classification system able to recognize the diversity of flora is certainly will be helpful for many researchers in agriculture and plantations, botanist, doctors, and it is also can be used as a learning tools for student in school. There are some characteristics that can be used for identifying a plant. Some plants can be identified by its physical features like flower, fruit, leaf, root or stem [1].

Leaf is the most frequent part that used in plant classification, manually or automatically [2]. Leaf has many special characteristics that can be used as a feature in the classification process, like color, shape, texture, or a combination of these features [3,4]. Color based research in leaf classification faced some problems, since most of leaf have a green color and some types of leaves change their color in certain seasons. Therefore, shape and leaf texture are widely studied in automatic plant identification. Some example of leaves shape features have been analyzed in several studies, like geometric descriptors [5] and fractal dimension [6,7]. Examples of research related to leaf texture analyzing is GLCM and LBP [8] and Gabor [9]. A shape and texture based approach in recognizing shape and texture features of plant leaf is also proposed in this study.

The application of fractal concept for fractal or non-fractal object has been commonly used in image analysis and pattern recognition, where fractal dimension is used to measure the complexity of geometric shapes and textures of an objects in term of fractional dimension [10,11]. However, there is a possibility that two objects with different pattern will likely to display the same fractal dimension's value. Mandelbrot [12] later introduced the concept *lacunarity* that able to measure the spatial distribution of gap with certain size on image texture [13]. Thus, it is stated that *lacunarity* will likely to complement this drawbacks. Low *lacunarity* value indicates that the texture is homogeneous if all gaps indicate the same size. While high *lacunarity* indicates that the texture is heterogeneous. *Lacunarity* has been applied in several areas of texture-based research, such as in the field of spatial data mapping [14,15], medical [16,17], and the agricultural industry [18].

Box counting method [19], is the most common approach used in calculating fractal dimension of an object, with its ability to represent the complexity of the image and its easy implementation [20]. Therefore, Bruno *et al.* [6] perform a leaf identification based on the complexity of the internal and external shape of leaf to obtain the fractal dimension using box counting method. The result shows a good performance but the misclassification rate was still quite high, so a fractal based texture recognition feature like *lacunarity* might be considered as a good feature to be combined with the fractal dimension in shape analysis to improve the classification accuracy. Although the fractal dimension is widely used in different areas, its only represents an object only by one unique real number. This becomes a drawback for recognition purposes since we may find a lot of objects with the same fractal dimension but completely diverse appearance. To overcome this drawback, we propose

to also use all difference values between adjacent element of $\log r$ and $\log N(r)$ from box counting methods. This technique is expected to be useful since fractal dimension is always extracted from the slope of the straight line of log-log plot.

One of the methods developed to obtain *lacunarity* feature is gliding box by Plotnick [21]. Gliding box is a box of a certain size applied to grayscale or binary image from left to right. This method has disadvantage since its applying a global thresholding. One of box gliding method proposed by Backes *et al.* [13] is the application of a local binary pattern of the input image with local thresholding, where thresholding stage is performed on each box. However, in these studies thresholding value is determined only by simple average gray value. This study will also try to apply a thresholding method that are more developed like Otsu methods. Given the importance of the binary pattern in improving the discriminatory feature of *lacunarity*, application of thresholding on each box in gliding box is expected to maintain the texture of local information that usually lost when applying global threshold.

Meanwhile, Kilic and Abiyev [20] mention that the fractal dimension and *lacunarity* have been examined separately, and there is no significant effort in combining the two features in a better synergy. Therefore, this study proposes the combination of fractal dimension features of leaf shape and *lacunarity* features of leaf texture to improve the classification accuracy compared to previous fractal dimension and *lacunarity* methods. To obtain fractal dimension features, a box-counting method by Bruno *et al* [6] was implemented, with modification in amount of features being extracted. *Lacunarity* features were obtained by using one of the gliding-box methods developed by Backes [13] by applying a local binary pattern, with using more advancing Otsu thresholding methods. Also we propose to add more feature along the calculation of *lacunarity*, with various box size r .

2. Methods

The proposed method is consist of several steps. The first step is preprocessing and segmentation of

TABLE 1
ASPECT RATIO AND ITS CORRESPONDING SIZE

R = major axis length/ minor axis length	Size (mxn)
$R \leq 1.4$	450 x 450
$1.4 < R \leq 2$	300 x 450
$2 < R \leq 2.4$	210 x 450
$2.4 < R \leq 3$	150 x 450
$3 < R \leq 5$	98 x 450
$5 < R \leq 13$	68 x 450
$R > 13$	15 x 450

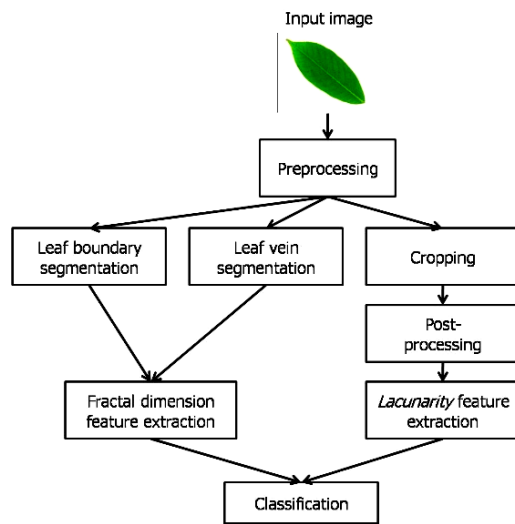


Figure 1. Proposed system

leaf input. In this step, the shape of leaf contour and leaf vein where obtained along with cropped image texture. The next step is post-pre-processing for leaf texture. The fractal dimension feature of leaf shape combined with *lacunarity* feature of leaf texture. The leaf classification task then finally performed at the end of process. Figure 1 shows the proposed system.

Preprocessing and Segmentation

Before an image goes into further steps, it is necessary to do the image preprocessing, a stage in which the image is being prepared in order to produce the desired output image. Image pre-processing result is expected to be used optimally at the next steps. In this study, image pre-processing results are used in the process of segmenting the leaf boundary and leaf veins (boundary and veins).

In the process of analyzing the leaf shape, the rotation stage is first performed so that the final image can be invariant towards rotation. The process started by converting a 1600x1200 RGB image into a grayscale image. The image is then rotated by aligned it toward the horizontal line, where the rotation angle is equal to an angle between major axis to horizontal axis of the image. To obtain the leaf boundary, grayscale image is converted into a binary image and the hole filling method is then applied. The binary and grayscale image is cropped until they only fits the bounding rectangle to make the features translation-invariant. All preprocessed leaf image result in varying image size since the image aspect ratio *R* of each leaf is different. Hence, the image is resized based on 7 type of predefined size to avoid distortion or variation in scale which is result in unbalanced feature length. Table 1 shows the predefined aspect ratio *R* with its

corresponding size. A canny edge detection method is applied in to the binary image to get the corresponding leaf boundary.

Detecting a leaf vein and segmenting it from the leaf objects is quite complicated because of the very low contrast difference between leaf veins and leaf objects [6-7]. We propose to apply the multi-thresholding method in segment-ing the leaf veins and obtained more than one image of the leaf veins. In this study, a canny edge detection method with more than one sigma σ value was applied to the gray image to obtain some images of leaf veins. Furthermore, the stage is ended with masking process between leaf boundary and leaf veins. Figure 2 shows the stages of preprocessing and veins segmentation.

Extracting leaf texture is done by converting input image into a grayscale image. The grayscale image is cropped into a 128x128 pixel size image. Different post-processing stage are then applied to the cropped image. The post-processing step to get the various texture image are consist of the following step:

Histogram equalization

Histogram equalization methods is aim to enhance the image contrast by transforming the values in an intensity image, or the values in the colormap of an indexed image. The enhancement will make the histogram of the output image approximately matches a specified histogram.

Kirsch Operator

The Kirsch edge detector module detects edges using eight compass filters [23]. All eight filters are applied to the image with the maximum being retained for the final image. The eight filters are a rotation of a basic compass convolution filter. The filters are of the form:

$$\begin{aligned}
 NW &= [5 \ -3 \ -3; 5 \ 0 \ -3; 5 \ -3 \ -3]; \\
 SW &= [-3 \ -3 \ -3; 5 \ 0 \ -3; 5 \ 5 \ -3]; \\
 SE &= [-3 \ -3 \ -3; -3 \ 0 \ -3; 5 \ 5 \ 5]; \\
 NE &= [-3 \ -3 \ -3; -3 \ 0 \ 5; -3 \ 5 \ 5]; \\
 N &= [-3 \ -3 \ 5; -3 \ 0 \ 5; -3 \ -3 \ -3]; \\
 W &= [-3 \ 5 \ 5; -3 \ 0 \ 5; -3 \ -3 \ -3]; \\
 S &= [5 \ 5 \ 5; -3 \ 0 \ -3; -3 \ -3 \ -3]; \\
 E &= [5 \ 5 \ -3; 5 \ 0 \ -3; -3 \ -3 \ -3];
 \end{aligned}$$

Canny edge detector

The process of Canny edge detection algorithm consist of 5 different steps [22]: 1) apply Gaussian filter to smooth the image in order to remove the noise; 2) find the intensity gradients of the image; 3) apply non-maximum suppression to remove spurious response to edge detection; 4) apply double threshold to determine potential edges; 5) track edge by hysteresis: finalize the detection of edges by

suppressing all the other edges that are weak and not connected to strong edges.

The Canny edge detector uses a Gaussian filter. The image is convolved with the filter. The filter blurs the image to a degree specified by σ to minimize the effect of unwanted information. The equation for a Gaussian filter kernel of size $(2k+1) \times (2k+1)$ is given by equation(1).

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-k-1)^2+(j-k-1)^2}{2\sigma^2}\right) \quad (1)$$

where the parameter σ (sigma) determines the width of the filter and hence the degree of blurring i.e. the greater the value of sigma the more the blurring is. If the value of sigma is high then faint edges will not be detected. On the other hand if sigma is very low then noise may also get detected as edges.

Local Thresholding

Local thresholding method is done by using a moving window that calculate the local binary value of an image by converting the grayscale image into a binary image [13].

Median Filtering

The median filtering is done by applying a filter that find the median value of grayscale image in a specific box size and resulted in a new filtered image. The median filtering is applied to the local the local thresholding result.

Skeletonization

The skeletonization step is aim to reduce all objects in an image to lines, without changing the essential structure of the image. Figure 3 shows the output of preprocessing results applied to the cropped leaf image.

Fractal Dimension Measurement

Fractal has a main characteristic called self similarity. These characteristics make the fractal has the ability to model complex and irregular natural objects, unlike euclidean geometry which is only able to determine the integer dimensions of an object. Fractal geometry involves various approaches to define fractional dimensions. The most common

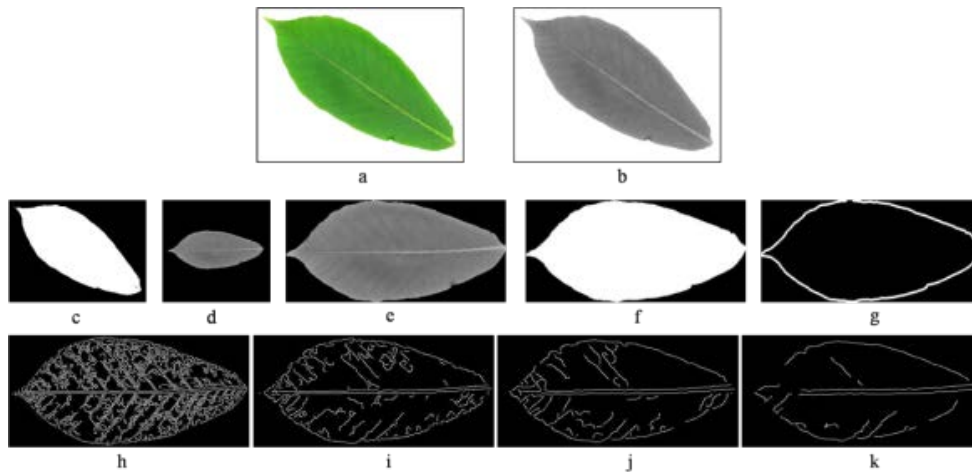


Figure 2. Preprocessing step to get leaf boundary and venation. a: original image; b: grayscale image; c: binary image; d: rotated image; e: shrunken gray image; f: shrunken binary image; g: leaf boundary; h,i,j,k: various leaf veins.

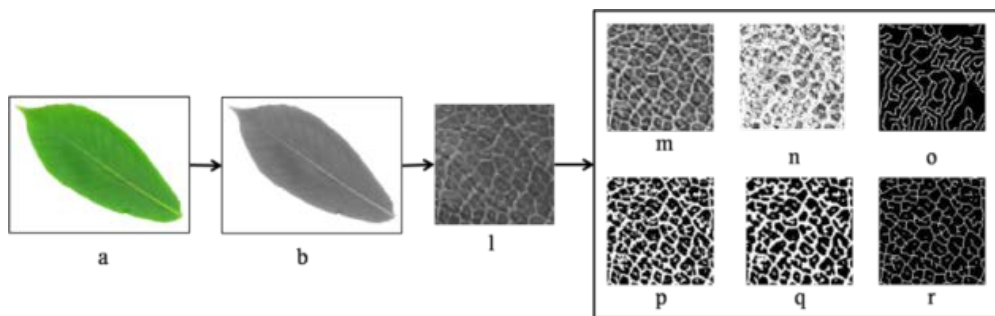


Figure 3. Preprocessing step to get various leaf texture models. a: original image; b: preprocessing image; l: cropped image texture; m: histogram equalization result; n: kirsch filter result; o: edge detection result; p: local thresholding result; q: median filter result; r: skeletonization result.

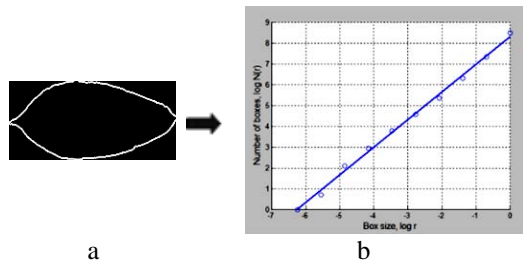


Figure 4. Fractal dimension measurement of a leaf boundary. a: input image; b: corresponding log-log plot

method used for calculating the fractal dimension of an image is a Box Counting method [6,19]. Fractal dimensions of an image with this method is calculated by the following equation(2).

$$D(r) = \lim_{r \rightarrow 0} \frac{\log N(r)}{\log r^{-1}} \quad (2)$$

where $N(r)$ denotes the number of r -sized box that contains information (pixels) object and $D(r)$ is the fractal dimension of the object with the box's size. The algorithm to measure the fractal dimension of an image using box counting method proposed in this study is as follow:

- 1) divide image into squares with a size r . The numerical value of boxe size are 2^n , with $n = 0,1,2,\dots$, and so on. 2^n should not be larger than the size of the image. When the image size is $2m \times 2m$, then the value of n will stop until m ;
- 2) calculate the number of boxes $N(r)$ containing occupied object in the image. The value of $N(r)$ is highly dependent on r ;
- 3) calculate $D(r)$ with equation(2) for the entire value of r ;
- 4) create a straight line based on the value of $\log N(r)$ (y-axis) and the values of $\log(r)$ (the x-axis) for each value of r , then calculate the slope of the straight line. The value of the slope is the fractal dimension of an image. The slope of a straight line calculated using the least squares method.

Figure 4 (a) shows an example of leaf boundary image while Figure 4 (b) is the image corresponding log-log plot for fractal dimension measurement. Table 2 shows the result of equation of fractal dimension of the image.

The fractal dimension measurement from Table 2 will result in only one single value. Therefore, we propose to add more features along the fractal dimensional value. This feature is all difference values between adjacent element of $\log r$ and $\log N(r)$.

This will produce n value that highly correlated with the corresponding fractal dimension value at each axes. The different value between $N(r)$

TABLE 2
EXAMPLE OF FRACTAL DIMENSION MEASUREMENT

r	$N(r)$	$\log r$	$\log N(r)$
512	1	-6.24	0
256	2	-5.55	0.69
128	8	-4.85	2.08
64	19	-4.16	2.94
32	44	-3.47	3.78
16	95	-2.77	4.55
8	214	-2.08	5.37
4	541	-1.39	6.29
2	1522	-0.69	7.33
1	4813	0	8.48

and r are then being divided to get the n final values. The fractal dimension feature vectors of each image in this study can be written as follow: $\overline{fd} = [D_0, D_1, D_2, \dots, D_n]$ where D_0 is the final fractal dimension value, while D_1 to D_n is feature vectors extracted from the log-log plot with n is maximum amount of differences value extracted from the log log plot.

Lacunarity Measurement

Lacunarity comes from the Latin (*lacuna*) which is also the origin of the word lake in English, refers to a concept which was also introduced by the “father” of the concept of fractals, Mandelbrot, in 1982. This concept defines that an object will be “lacunar” if gap (hole) on an object tends to be large. Low *lacunarity* indicates that the texture is homogeneous, while high *lacunarity* indicates that the texture is heterogeneous [13,20]. High *lacunarity* value means that the pixels spread out over a wider range and surrounded by many and large gaps [20].

Initially, *lacunarity* introduced to describe the fractal characteristics that have the same dimensions but have a different appearance [11,13]. Thus it able to overcome the drawbacks of widely used fractal dimension. Until now *lacunarity* concept being developed in analyzing the texture and is scale-dependents [14,15,25]. One of most common and simple approach to calculate the *lacunarity* of a binary image map is the gliding-box algorithm, introduced by Allain and Cloitre [10]. This algorithm analyzes the image by applying an overlapping box with size $r \times r$ that glides over an image from upper left to the right. S is the number of occupied sites or mass of the gliding box. The number of boxes of size r containing S occupied sites is designated by $n(S,r)$ and the total number of boxes of size r by $N(r)$. If the map is M , then the total number of boxes is calculated using equation(3).

$$N(r) = (M - r + 1)^2 \quad (3)$$

This frequency distribution is converted into a probability distribution $Q(S,r)$ by dividing it to the total number of boxes using equation(4).

$$Q(S,r) = n(S,r)/N(r) \quad (4)$$

The first and second moments of this distribution are now determined in equation(5).

$$Z^{(1)} = \sum S Q(S,r) \quad (5)$$

$$Z^{(2)} = \sum S^2 Q(S,r) \quad (6)$$

So the lacunarity value $\Lambda(r)$ of the image with box size r can be defined as equation(7).

$$\Lambda(r) = Z^{(2)}/(Z^{(1)})^2 \quad (7)$$

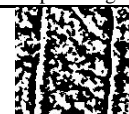
Leaf texture image is analyzed by using the method of gliding box. Figure 5 is example of leaf texture from each class for lacunarity analysis. At this stage, the box with the size r move above the grayscale or binary image started from top left to the bottom right. Once the gliding box is finished, the frequency distribution of the mass of the box r is calculated so that the value of lacunarity can be obtained through the equation(6). The resulted lacunarity feature vectors described as follow: $\overline{\Lambda r} = [\Lambda r_1, \Lambda r_2, \Lambda r_3, \dots, \Lambda r_m]$ where Λr_1 is lacunarity value at smallest box size r , and Λr_m is lacunarity value at maximum box size r , with $m = 2^n$ and m was lies between 1 to maximum image size. Table 3 is an example of lacunarity measurement using box size $r=2$ and a 128x128 pixels binary image.

Feature combination is conducted by simply concatenate one feature vector into another feature vector. Therefore, the fractal dimension of leaf shapes (boundary and veins) are concatenated with lacunarity



Figure 5. Example of leaf texture from each class for lacunarity analysis

TABLE 3
EXAMPLE OF LACUNARITY MEASUREMENT (R=2)

Input Image	$Z^{(1)}$	$Z^{(2)}$	$\Lambda(r)$
	7078124	5.9956×10^9	1.9302

curarity feature vector of leaf texture to produce a feature vector with length $1 \times n$, where n consists of a combination of features D_1 , D_2 and Λr . D_1 is the fractal dimension of the shape of leaf boundary, D_2 is the fractal dimension of leaf veins, and Λr is lacunarity of leaf texture with box size r . The length of the feature vectors will be vary based on the amount of box r being applied and the input images.

Data set

Dataset used in this study was the *flavia* dataset that available for public using. Figure 3 shows examples of leaves dataset from *flavia*. Image with a white background has previously been acquired by using the scanner to produce images with a size of 1600x1200 pixels and have a *jpeg file format. The dataset can be downloaded at the site <http://flavia.sourceforge.net>. The fundamental properties of the data sets are shown in Table 4.

Experiments

The experiments were conducted to answer the research question of this study: whether there is a better synergy between fractal dimension and lacunarity using proposed methods to increase leaf classification accuracy. Three classifiers were used to compare the classification result. The performances of the proposed methods were evaluated using



Figure 6. Example of leaf using in this study

TABLE 4
DATASETS AND THEIR PROPERTIES

Class Label	Species Name	Amount data per-class
1	<i>pubescent bamboo</i>	34
2	<i>Chinese horse chestnut</i>	36
3	<i>Chinese redbud</i>	37
4	<i>true indigo</i>	48
5	<i>Japanese maple</i>	37
6	<i>goldenrain tree</i>	29
7	<i>Chinese cinnamon</i>	37
8	<i>Japanese cheesewood</i>	36
9	<i>Sweet osmanthus</i>	29
10	<i>ginkgo</i>	36
11	<i>Crepe myrtle</i>	39
12	<i>oleander</i>	33
13	<i>yew plum pine</i>	32
14	<i>Ford Woodlotus</i>	27
15	<i>Tangerine</i>	26
16	<i>Japan Arrowwood</i>	36
17	<i>Beales Barberry</i>	39
18	<i>Glossy Privet</i>	35

10-fold cross validation [24]. All datasets were split into 10 data subsets. One subset was used for testing and the other nine subsets were used as training. This procedure was repeated 10 times for all of data sets. To get the accuracy value take the average value of all fold classification result as defined in equation(8).

$$Accuracy = \frac{n_{correctly\ classified\ data}}{number\ of\ all\ data} \times 100\% \quad (8)$$

The experiment we performed is consist of three experiments as follows: 1) an experiment using only fractal dimension feature. In this scenario, all datasets were analyzed to get the leaf boundary and leaf veins. Since we propose to use more than one sigma σ value for edge detection, then we will analyze the effect of the edge detection towards resulted fractal dimension features. The box counting method was applied on each resulted images to get fractal dimension feature vectors; 2) an experiment using only *lacunarity* feature by applying different box size r . The experiment is aim is to evaluate the best r which able to produce best classification. In this scenario, the amount of grayscale and binary image was observed to see whether it would affect the classification accuracy. When right amount of input images is obtained, the gliding box method was applied to get the expected *lacunarity* feature vectors; 3) an experiment that combines both fractal dimension and *lacunarity* feature vectors. In this scenario, both fractal dimension and *lacunarity* feature vectors are concatenated and then analyzed to see the synergy between the combined features. We then do the comparison between the proposed fractal dimension and *lacunarity* combination to two previous methods.

3. Results and Analysis

Fractal dimension analysis

Using box counting method, we conduct an experiment to analyze performance of the system. To ob-

TABLE 5
COMPARISON OF DIFFERENT SIGMA VALUE OF CANNY
EDGE DETECTOR

Input Image	Feature length	Classification accuracy (%)
Leaf boundary, Leaf vein ($\sigma = 1$)	1x20	60.376
Leaf boundary, Leaf vein ($\sigma = 1$ and $\sigma = 2$)	1x30	71.556
Leaf boundary, Leaf vein ($\sigma = 1, \sigma = 2, \sigma = 3$)	1x40	73.458
Leaf boundary, Leaf vein ($\sigma = 1, \sigma = 2, \sigma = 3, \sigma = 4$)	1x50	76.979

tain fractal dimension features of the leaf shape (boundary and vein), a canny edge detector is applied. For first test, a single value of sigma $\sigma = 1$ is applied to the canny edge detection operator to obtain the leaf veins. At this try, one leaf boundary image and one leaf vein image is extracted. The fractal dimensions of both images are being measured using proposed box counting methods. Classification results show a success rate of 60.376%.

However, each of leaf has different brightness and contrast [6]. So it is very difficult to segment with one unified gray level threshold after converted to grayscale images. Therefore, the next testing was conducted to analyze the effect of the amount of sigma value at canny operator to obtain proper leaf veins. The second testing apply the value of $\sigma=1$ and $\sigma=2$ for canny operator, followed by the third experiment with $\sigma=1, \sigma=2,$ and $\sigma=3,$ and the fourth testing which combine $\sigma=1, \sigma=2, \sigma=3,$ and $\sigma=4$ altogether. Table V describes the comparison of classification accuracy when using different sigma values. The highest result shown in the fourth trial, with an average accuracy of 76.979%. From this result, we can see that using only one single leaf vein image was only resulted in 60.376% of accuracy. Figure 7 shows the comparison of classification result using conventional fractal dimension measurement by Backes [6] and our proposed methods. Result shows that using only one single fractal dimension value will produce less classification result compared to our proposed methods. This also suggest that using all difference values between adjacent element of log box size r and log amount of box $N(r)$ as a leaf features was able to increase the classification accuracy.

Lacunarity analysis

To analyze the performance of gliding box method in measuring the *lacunarity* value, we change the size of box r that varies according to the input image in order to find an optimal r value. Figure 8 shows the classification accuracy with various box size r . Result shows that the smaller the box size r , the better the accuracy. While the greater the size of the box, then the accuracy decreases.

The result also shows that the overall accuracy is not very high. This is due to the high similarity between classes of leaf textures. This also suggest that *lacunarity* measurement was better performed for data with a few number of classes, like what have been done in previous study [14-18]. At first, we use only one input value, but the result is not quite satisfying. Therefore, we try to combine several images as an input for *lacunarity* measurement. We finally use six input images that represent the uniqueness of image texture. The combined image resulted in 50,169% of classification accuracy.

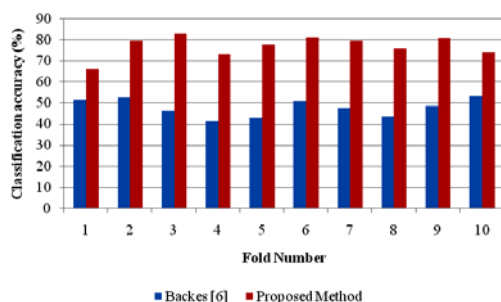


Figure 7. Comparison of previous methods and proposed methods

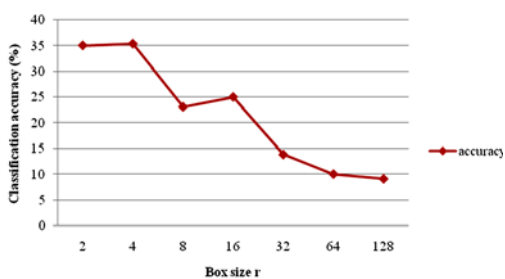


Figure 8. The effect of box size *r* towards classification result

TABLE 6
CLASSIFICATION RESULT USING ONLY ONE IMAGE FOR LACUNARITY ANALYSIS







No	Input image	Accuracy (%)
1	Histogram equalization 	14.22
2	Kirsch filter 	14.54
3	Local thresholding[13] 	19.83
4	Median filter 	17.09
5	Canny edge detector 	16.15
6	Skeletonization 	15.32
Combination of 1,2,3,4,5,6		50.169

Table 6 describes the classification result if we use the single input image separately, while Table 7 shows the classification accuracy (using all of the 6 images) with various box size *r*.

Fractal Dimension and Lacunarity Combination Analysis

The final experiment conducted to answer the research question in this study, whether there are a synergy between a combination of fractal dimension of leaf shape and lacunarity of leaf texture. The

TABLE 7
THE EFFECT OF BOX SIZE IN LACUNARITY

box size <i>r</i>	Feature length	accuracy (%)
2	1x6	35.14
4	1x6	35.47
8	1x6	23.18
16	1x6	25.09
32	1x6	13.90
64	1x6	10.07
128	1x6	9.129
2,4,8,16, 32	1x30	50.169

TABLE 8
COMPARISON OF OVERALL CLASSIFICATION ACCURACY

Fold number	Fractal dimension	Lacunarity	Combination of both features
1	66.129	53.226	93.548
2	79.365	39.682	93.650
3	82.539	53.968	95.238
4	73.016	60.317	93.650
5	77.777	39.682	95.238
6	80.952	41.269	93.650
7	79.365	60.317	98.412
8	75.806	56.451	93.548
9	80.645	46.774	95.161
10	74.193	50	87.096
Average accuracy	76.979%	50.169%	93.919%

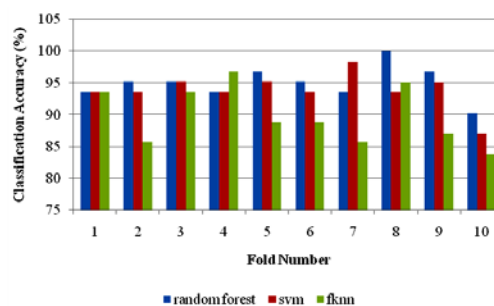


Figure 9. Classification accuracy of proposed method with three classifiers

resulted features \vec{v} concatenated from fractal dimension and lacunarity measurement can be described as $\vec{v} = [D_{10}, D_{11}, D_{12}, \dots, D_{1max}, D_{20}, D_{21}, \dots, D_{2max}, A_1, A_2, \dots, A_{max}]$ where D_1 is a fractal dimension of leaf boundary, D_2 is fractal dimension of leaf veins, while A is lacunarity value of leaf texture. The feature vectors used in this experiment is consist of combination of best fractal dimension and lacunarity feature vectors obtained from previous experiment.

Table 8 shows the comparison result of each methods when applied alone or combined together using 10-fold cross validation system and SVM classifier. When using fractal dimension of leaf shape alone, the system is able to reach 82.539% of accuracy with average classification accuracy is 76.979%, while analysis of lacunarity feature resulted in 50.169% accuracy with highest classification rate found at 4th fold (60.317%). Combining

both features able to improve the average accuracy up to 93.916% with highest classification result shows in 7th fold that reaches 98.412% of classification accuracy.

We also compare the classification result using other classifiers aside of SVM, which are Random Forest and Fuzzy k -Nearest Neighbor to see the robustness of this combined features. Figure 9 shows the comparison of classification accuracy when we use different classifiers. Result shows that the ensemble classifier, Random Forest, is able to outperform SVM and F-Knn classifier with average 95.048% of classification accuracy, while SVM resulted in 93.92% and Fknn produce 89.93% of accuracy. From this experiment, we can see the robustness of the proposed feature extraction and combination methods. We also able to prove the hypothesis that expects a better synergy between fractal dimension and *lacunarity* when combined together rather than using each feature alone. This also suggests that using fractal dimension and *lacunarity* in leaf classification task will lead to a promising result.

4. Conclusion

We have presented a study to analyze the synergy of combined features of fractal dimension and *lacunarity* to improve plant leaf classification accuracy. Experiment is performed using 626 of leaf images from *flavia* leaf dataset. To obtain the leaf boundary and vein, an edge detection operator with multi-threshold value was applied to obtain the most representative features. Then we propose a method to extract a sequence of fractal dimension value from a log-log plot after applying a box counting methods. *Lacunarity* value obtained by applying a gliding box methods on leaf texture image. Parameter of box size r was analyzed in the *lacunarity* calculation to determine an optimal r value. Furthermore, the extracted fractal dimension feature vector was concatenated with *lacunarity* feature vectors. Experiment result shows that highest success rate in the calculation of fractal dimension value can be obtained when combining images with sigma value of 1 to 4.

Meanwhile, best results at *lacunarity* calculation is obtained when the size of the box r used is a combination of $r = 2,4,8,16,32$. Combination of each of the best features of fractal dimension analysis of leaf shape and *lacunarity* analysis of leaf texture able to achieve an average 95.948%, 93.92% and 89.93% of classification accuracy using Random Forest, SVM, and F-Knn classifiers respectively. These result indicates that the combining both fractal dimension and *lacunarity* features are better than using these methods separately. It is also able to prove the hypothesis that there is a synergy bet-

ween the two features. In the future, a fractal based feature combination might considered as a good references in the field of plant leaf classification.

References

- [1] Backes, Andre Ricardo, Dalcimar Casanova, and Odemir Martinez Bruno. "Plant leaf identification based on volumetric fractal dimension." *International Journal of Pattern Recognition and Artificial Intelligence* 23.06, pp. 1145-1160. 2009.
- [2] Novotný, Petr, and Tomáš Suk. "Leaf recognition of woody species in Central Europe." *biosystems engineering* 115.4, pp. 444-452. 2013.
- [3] Beghin, Thibaut, et al. "Shape and texture based plant leaf classification." *Advanced Concepts for Intelligent Vision Systems*. Springer Berlin Heidelberg, 2010.
- [4] Kadir, Abdul, et al. "Leaf classification using shape, color, and texture features." *arXiv preprint arXiv: 1401.4447*. 2013.
- [5] Du, Ji-Xiang, Xiao-Feng Wang, and Guo-Jun Zhang. "Leaf shape based plant species recognition." *Applied mathematics and computation* 185.2, pp. 883-893. 2013.
- [6] Bruno, Odemir Martinez, et al. "Fractal dimension applied to plant identification." *Information Sciences* 178.12, pp. 2722-2733. 2008.
- [7] Du, Ji-xiang, Chuan-Min Zhai, and Qing-Ping Wang. "Recognition of plant leaf image based on fractal dimension features." *Neuro-computing* 116, pp.150-156. 2013.
- [8] Arun, C. H., WR Sam Emmanuel, and D. Christopher Durairaj. "Texture feature extraction for identification of medicinal plants and comparison of different classifiers." *Int J Comput Appl* 62.12, pp.1-9. 2013.
- [9] Lin, Feng-Yan, et al. "Multiple classification of plant leaves based on Gabor transform and LBP operator." *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*. Springer Berlin Heidelberg, pp. 432-439. 2008.
- [10] Allain, C., and M. Cloitre. "Characterizing the lacunarity of random and deterministic fractal sets." *Physical review A* 44.6, pp. 3552. 1991.
- [11] Dong, Pinliang. "Test of a new lacunarity estimation method for image texture analysis." *International Journal of Remote Sensing* 21.17, pp. 3369-3373. 2000.
- [12] Mandelbrot, Benoit B. *The fractal geometry of nature*. Vol. 173. Macmillan, 1983.
- [13] Backes, André Ricardo. "A new approach to

- estimate lacunarity of texture images.” *Pattern Recognition Letters* 34.13, pp. 1455-1461. 2013.
- [14] Myint, Soe Win, and Nina Lam. “A study of lacunarity-based texture analysis approaches to improve urban image classification.” *Computers, environment and urban systems* 29.5, pp. 501-523. 2005.
- [15] Dong, Pinliang. “Lacunarity analysis of raster datasets and 1D, 2D, and 3D point patterns.” *Computers & Geosciences* 35.10, pp. 2100-2110. 2010.
- [16] Borys, Przemyslaw, et al. “Lacunarity as a novel measure of cancer cells behavior.” *Bio-systems* 94.3, pp. 276-281. 2008.
- [17] Neves, Leandro Alves, et al. “Multi-scale lacunarity as an alternative to quantify and diagnose the behavior of prostate cancer.” *Expert Systems with Applications* 41.1, pp. 5017-5029. 2014.
- [18] Li, Lei, et al. “Multifractal analysis and lacunarity analysis: A promising method for the automated assessment of muskmelon (*Cucumis melo* L.) epidermis netting.” *Computers and electronics in agriculture* 88, pp. 72-84. 2012.
- [19] Voss, Richard F. “Characterization and measurement of random fractals.” *Physica Scripta* 1986.T13, pp. 27. 1986.
- [20] Kilic, Kemal Ihsan, and Rahib Hidayat Abiyev. “Exploiting the synergy between fractal dimension and lacunarity for improved texture recognition.” *Signal Processing* 91.10, pp.2332-2344. 2011.
- [21] Plotnick, Roy E., Robert H. Gardner, and Robert V. O'Neill. “Lacunarity indices as measures of landscape texture.” *Landscape ecology* 8.3, pp. 201-211. 1993.
- [22] Canny, John. “A computational approach to edge detection.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 6, pp. 679-698. 1986.
- [23] Kirsch, Russell A. “Computer determination of the constituent structure of biological images.” *Computers and biomedical re-search* 4.3, pp. 315-328. 1971.
- [24] Kohavi, Ron. “A study of cross-validation and bootstrap for accuracy estimation and model selection.” *Ijcai*. Vol. 14. No. 2. 1995.
- [25] Plotnick, Roy E., et al. “Lacunarity analysis: a general technique for the analysis of spatial patterns.” *Physical review E* 53.5, pp. 5461. 1996.

FEATURE SELECTION METHODS BASED ON MUTUAL INFORMATION FOR CLASSIFYING HETEROGENEOUS FEATURES

Ratri Enggar Pawening¹, Tio Darmawan², Rizqa Raaiqa Bintana^{2,3}, Agus Zainal Arifin² and Darlis Herumurti²

¹Department of Informatics, STT Nurul Jadid Paiton, Jl. Pondok Pesantren Nurul Jadid Paiton, Probolinggo, 67291, Indonesia

²Department of Informatics Engineering, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember (ITS), Kampus ITS Sukolilo, Surabaya, 60111, Indonesia

³Department of Informatics, Faculty of Science and Technology, UIN Sultan Syarif Kasim Riau, Jl. H.R Soebrantas, Pekanbaru, 28293, Indonesia

E-mail: enggar.r@gmail.com¹, agusza@cs.its.ac.id²

Abstract

Datasets with heterogeneous features can affect feature selection results that are not appropriate because it is difficult to evaluate heterogeneous features concurrently. Feature transformation (FT) is another way to handle heterogeneous features subset selection. The results of transformation from non-numerical into numerical features may produce redundancy to the original numerical features. In this paper, we propose a method to select feature subset based on mutual information (MI) for classifying heterogeneous features. We use unsupervised feature transformation (UFT) methods and joint mutual information maximization (JMIM) methods. UFT methods is used to transform non-numerical features into numerical features. JMIM methods is used to select feature subset with a consideration of the class label. The transformed and the original features are combined entirely, then determine features subset by using JMIM methods, and classify them using support vector machine (SVM) algorithm. The classification accuracy are measured for any number of selected feature subset and compared between UFT-JMIM methods and Dummy-JMIM methods. The average classification accuracy for all experiments in this study that can be achieved by UFT-JMIM methods is about 84.47% and Dummy-JMIM methods is about 84.24%. This result shows that UFT-JMIM methods can minimize information loss between transformed and original features, and select feature subset to avoid redundant and irrelevant features.

Keywords: *Feature selection, Heterogeneous features, Joint mutual information maximation, Support vector machine, Unsupervised feature transformation*

Abstrak

Dataset dengan fitur heterogen dapat mempengaruhi hasil seleksi fitur yang tidak tepat karena sulit untuk mengevaluasi fitur heterogen secara bersamaan. Transformasi fitur adalah cara untuk mengatasi seleksi subset fitur yang heterogen. Hasil transformasi fitur non-numerik menjadi numerik mungkin menghasilkan redundansi terhadap fitur numerik original. Dalam tulisan ini, peneliti mengusulkan sebuah metode untuk seleksi subset fitur berdasarkan *mutual information* (MI) untuk klasifikasi fitur heterogen. Peneliti menggunakan metode *unsupervised feature transformation* (UFT) dan metode *joint mutual information maximation* (JMIM). Metode UFT digunakan untuk transformasi fitur non-numerik menjadi fitur numerik. Metode JMIM digunakan untuk seleksi subset fitur dengan pertimbangan label kelas. Fitur hasil transformasi dan fitur original disatukan seluruhnya, kemudian menentukan subset fitur menggunakan metode JMIM, dan melakukan klasifikasi terhadap subset fitur tersebut menggunakan algoritma *support vector machine* (SVM). Akurasi klasifikasi diukur untuk sejumlah subset fitur terpilih dan dibandingkan antara metode UFT-JMIM dan Dummy-JMIM. Akurasi klasifikasi rata-rata dari keseluruhan percobaan yang dapat dicapai oleh metode UFT-JMIM sekitar 84.47% dan metode Dummy-JMIM sekitar 84.24%. Hasil ini menunjukkan bahwa metode UFT-JMIM dapat meminimalkan informasi yang hilang diantara fitur hasil transformasi dan fitur original, dan menyeleksi subset fitur untuk menghindari fitur redundansi dan tidak relevan.

Kata Kunci: *Fitur heterogen, Joint mutual information maximation, Seleksi fitur, Support vector machine, Unsupervised feature transformation*

1. Introduction

Data and features which have high-dimensional are the main problems in the classification of supervised and unsupervised learning, which is becoming even more important with the recent explosion of the size of the available datasets both in terms of the number of data samples and the number of features in each sample. The rapid training time and the enhancement of classification accuracy can be obtained when dimension of data and features are decreased as low as possible.

Dimensionality reduction can be conducted using feature extraction and feature selection methods. Feature extraction methods transform the original features into a new feature which has lower dimension. The common used methods are principal component analysis (PCA) [1-2] and linear discriminant analysis (LDA) [3-4]. Feature selection methods is conducted by selecting some important features which minimises a cost function.

Feature selection methods are divided into two categories in terms of evaluation strategy, in particular, classifier dependent and classifier independent. Classifier dependent is divided into two methods, wrapper and embedded methods. Wrapper methods evaluate subsets of variables to detect the possible interactions between variables by measuring the prediction accuracy of a classifier. Wrapper methods had researched by [5-6]. They perform well because the selected subset is optimised for the classification algorithm. Wrapper methods may suffer from over-fitting to the learning algorithm and has very expensive in computational complexity, especially when handling extremely high-dimensional data. It means that each change of training models will decrease the function of subsets.

The feature selection stage in the embedded methods is combined with the learning stage [6]. Embedded methods perform variable selection as part of the learning procedure and are usually specific to given learning machines. These methods are less computational complexity and over-fitting. However, they are very specific and difficult for generalisation.

Classifier independent can be called as filter methods. Filter methods assess the relevance of features by looking only at the intrinsic properties of the data. The advantages of filter methods are: they can scale of high-dimensional datasets, they are computationally simple and fast, and they are independent of the classification algorithm. The disadvantage of filter methods is that they ignore the interaction between the features and the classifier (the search in the feature subset space is separated from the search in the hypothesis space), and

most proposed techniques are univariate. Feature selection using filter methods is researched by [7]. These methods rank features according to their relevance to the class label in the supervised learning. The relevance score is calculated using mutual information (MI).

Information theory has been widely applied in filter methods, where information measures such as mutual information are used as a measure of the features's relevance and redundancy. MI can overcome problems of filter methods. Some methods which apply MI are MIFS [8], mRMR [9], NMIFS [10], and MIFS-ND [11]. These methods optimize the relationship between relevance and redundancy when selecting features. The problems of these methods is the overestimation of the significance of the feature candidates. The method for selecting the most relevant features using joint mutual information (JMI) is proposed by [12]. Joint Mutual Information Maximization (JMIM) is the development of JMI that adds maximum of the minimum method.

Datasets with heterogeneous features can affect feature selection results that are not appropriate because it is difficult to evaluate heterogeneous features concurrently. Feature transformation (FT) is another way to handle heterogeneous features subset selection. FT methods unify the format of datasets and enable traditional feature selection algorithms to handle heterogeneous datasets. FT methods for heterogeneous features using unsupervised feature transformation (UFT) has proposed by [13]. The results of transformation from non-numerical into numerical features may produce redundancy to the original numerical features. The redundant features can be handled by selecting of the significant feature.

In this paper, we propose a method to select feature subset based on mutual information (MI) for classifying heterogeneous features. This paper is organized as follows. Section 2 describes research methodology of the proposed methods. Secti-

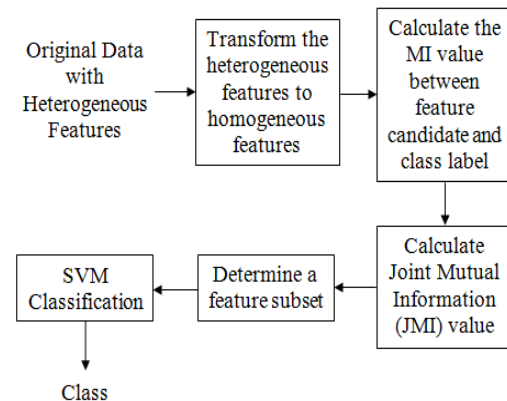


Figure 1. The proposed methods

on 3 describes the conducted experiments and discusses the results. Section 4 concludes this study.

2. Methods

General description of the research methods is shown in Figure 1. The stages of UFT-JMIM methods in this study are transformation of heterogeneous features, calculation of MI value between feature candidate and class label, calculation of JMI value, determine a feature subset, and classification by using SVM.

Transformation of heterogeneous features

In the transformation stage, we transform the datasets that have heterogeneous features to homogeneous features. Feature transformation is conducted by using UFT methods. UFT is derived from the analytical relationship between MI and entropy. The purpose of UFT is to find a numerical X' to substitute the original non-numerical feature X , and X' is constrained by $I(X';X) = H(X)$. This constraint makes the MI between the transformed X' and the original X to be the same as the entropy of the original X .

This condition is critical because it ensures that the original feature information is preserved, when non-numerical features are transformed into numerical features. It is also worth noting that the transformation is independent of class label, so that the bias introduced by class label can be reduced. After it is processed by UFT methods, the datasets's format which have heterogeneous features can be combined to numerical features entirely. The solution for UFT methods is shown by equation(1) [13]. Based on equation(1), UFT methods can be formalized as shown by Algorithm 1, which also details equation(1) together.

$$\mu_i^* = [(n - i) - \sum_{k=1}^i (n - k) p_k] \sqrt{(1 - \sum_i p_i^3) / \sum_{i \neq j} p_i p_j (i - j)^2}, \quad (1)$$

where $\sigma_i^* = p_i \quad i \in \{1, \dots, n\}$

Calculation of MI value between feature candidate and class label

MI is the amount of information that both variables share, and is defined as equation(2). Each feature f_i which is a member of F is calculated the value of MI (I) to class label C . By adopting equation(2), the value of MI for each feature f_i is obtained by using equation(3).

$$I(X; C) = H(C) - H(C|X) \quad (2)$$

Algorithm 1: UFT

Input: dataset D , which have heterogeneous feature $f_{i,j} \in \{1, \dots, m\}$

Output: transformed dataset D' with pure numerical features

```

1: for  $j = 1$  to  $m$  do
2:   if feature  $f_j$  is non-numerical then
3:      $n = \text{size}(\text{unique}(f_j))$ ;
4:      $\{s_i | i = 1, \dots, n\}$  is the set of non-numerical values in feature  $f_j$ 
5:      $p_i$  is the probability of  $s_i$ 
6:     for  $i = 1$  to  $n$  do
7:        $\mu_i = [(n - i) - \sum_{k=1}^i (n - k) p_k] \sqrt{(1 - \sum_i p_i^3) / \sum_{i \neq j} p_i p_j (i - j)^2}$ ;
8:        $\sigma_i = p_i$ ;
9:       use Gaussian distribution  $\mathcal{N}(\mu_i, \sigma_i)$  to generate numerical data and substitute the values equal to  $s_i$  in feature  $f_j$ 
10:    end for
11:  end if
12: end for

```

$$I(f_i; C) = H(C) - H(C|f_i) \quad (3)$$

$I(f_i; C) = H(C) - H(C|f_i)$ where $H(C)$ is defined as equation(4). $H(C)$ is the entropy of class label C .

$$H(C) = -\sum_{i=1}^N p(c_i) \log(p(c_i)) \quad (4)$$

The value of $p(c_i)$ probability function is obtained by using equation(5).

$$p(c_i) = \frac{\text{number of instants with value } c_i}{\text{total number of instants } (N)} \quad (5)$$

To fill the first subset, find $I(f_i, C)$ which has a maximum value. Feature f_i is more relevant to the class label C than feature f_j in the context of the already selected subset S if it satisfies equation(6).

$$I(f_i, S; C) > I(f_j, S; C) \quad (6)$$

Calculate JMI value

Let $S = \{f_1, f_2, \dots, f_k\}$, JMI of f_i and each feature in S with C is calculated. The minimum value of this mutual information is selected based on the lowest amount of new information of feature f_i that is added to subset. The feature that produces the maximum value is the feature that adds maximum information to that shared between S and C , it means that the feature is most relevant to the class label C in the context of the subset S according to equation(6).

The features are selected by JMIM according to equation(7), where JMI $I(f_i, f_s; C)$ is defined as

Algorithm 2: Forward greedy search

-
1. (Initialisation) Set $F \leftarrow$ "initial set of n features"; $S \leftarrow$ "empty set."
 2. (Computation of the MI with the output class) For $\forall f_i \in F$ compute $I(C; f_i)$.
 3. (Choice of the first feature) Find a feature f_i that maximises $I(C; f_i)$; set $F \leftarrow F \setminus \{f_i\}$; set $S \leftarrow \{f_i\}$.
 4. (Greedy selection) Repeat until $|S| = k$: (Selection of the next feature) Choose the feature $f_i = \arg \max_{f_i \in F} s(\min_{f_s \in S} (I(f_i, f_s; C)))$; set $F \leftarrow F \setminus \{f_i\}$; set $S \leftarrow S \cup \{f_i\}$.
 5. (Output) Output the set S with the selected features.
-

equation(8).

$$f_{JMIM} = \arg \max_{f_i \in F-S} (\min_{f_s \in S} (I(f_i, f_s; C))) \quad (7)$$

$$I(f_i, f_s; C) = [-\sum_{c \in C} p(c) \log(p(c))] - \left[\sum_{c \in C} \sum_{f_i \in F-S} \sum_{f_s \in S} \log \left(\frac{p(f_i f_s; c/f_s)}{p(f_i/f_s) p(c/f_s)} \right) \right] \quad (8)$$

Determine a feature subset

The method uses the following iterative forward greedy search algorithm to find the relevant feature subset of size k within the feature space (Algorithm 2).

Classification process

At this stage, classification process is conducted to determine the class of the object. In this study, the classification uses support vector machine (SVM) multiclass One-Against-One (OAO) with polynomial kernel. Polynomial kernel function (K) is shown by equation(9):

$$K(x, x_i) = [(x \cdot x_i) + 1]^q \quad (9)$$

where x_i is dimensional input ($i = 1, 2, \dots, l$, l is the number of samples) belong to class 1 or another and q is power of polynomial kernel function.

Datasets

Datasets are used in this study from UCI Repository (table I). They are Acute Inflammations, Adult, Australian Credit Approval, German Credit Data, and Hepatitis. Data type of Acute Inflammations dataset is multivariate. The attribute types of this dataset are categorical and integer. This dataset contains 1 numerical feature and 5 non-numerical features. All of the non-numerical features

only have two probability values, yes or no value. This dataset has two classes of data, they are yes for the inflammation of urinary bladder and no for not.

Data type of Adult dataset is multivariate. This dataset contains 14 features that composed by categorical and integer values. The attribute types of this dataset are categorical and number. Every feature has different number of values. This dataset has two data classes.

Australian Credit Approval dataset has multivariate data type. This dataset contains 14 features that composed by categorical, number, and real values. There are 6 numerical features and 8 categorical features. This dataset has two data classes. They are + (positive) class for approved credit and - (negative) class for rejected credit.

Data type of German Credit Data dataset is multivariate. This dataset contains 20 features that composed by categorical and number. There are 7 numerical features and 13 categorical features. This dataset has two classes of data, they are 1 as good credit consumer and 2 as bad credit consumer.

Data type of Hepatitis dataset is multivariate. The dataset contains 20 features that composed by categorical, number, and real values. There are 6 numerical features and 13 categorical features. This dataset has two classes of data, they are 1 for die and 2 for live.

3. Results and Analysis

To validate the results of proposed methods, five datasets from UCI Repository are used in the experiment (Table 1). In the datasets used, the type of non-numerical features is categorical data which is nominal and ordinal data type. The number of non-numerical features in each dataset is different (Table 2).

Scenario of testing is conducted by transforming non-numerical features using UFT methods and dummy variable. The transformation using dummy variable is conducted by changing the data to the numbers manually, for example feature of sex which has male and female data is changed by numeral 1 (for male) and 2 (for female).

It means Dummy-JMIM has lower complexity than UFT-JMIM but we do not know it is good for changing the categorical value manually or no. The transformed and the original features are combined entirely, then determine features subset by using JMIM methods, and classify them using SVM algorithm. The classification accuracy are measured for any number of selected feature subset and compared between UFT-JMIM methods and Dummy-JMIM methods.

Dummy variable is an defined variable whi-

TABLE 1
DESCRIPTION OF REAL-WORLD DATASETS

	1	2	3	4	5	6
Acute Inflammations		120	2	1	5	6
Adult		1992	2	6	8	14
Australian		690	2	6	8	14
Credit Approval						
German Credit Data		1000	2	7	13	20
Hepatitis		80	2	6	13	19

Titles of column heads:
1: Datasets; 2: Instances; 3: Classes; 4: Numerical features
5: Non-numerical features; 6: Features

TABLE 2
NON-NUMERICAL FEATURES IN DATASETS

Datasets	Position of non-numerical features in datasets
Acute Inflammations	2, 3, 4, 5, 6
Adult	2, 4, 6, 7, 8, 9, 10, 14
Australian Credit Approval	2, 4, 5, 6, 8, 9, 11, 12
German Credit Data	1, 3, 4, 6, 7, 9, 10, 12, 14, 15, 17, 19, 20
Hepatitis	2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 19

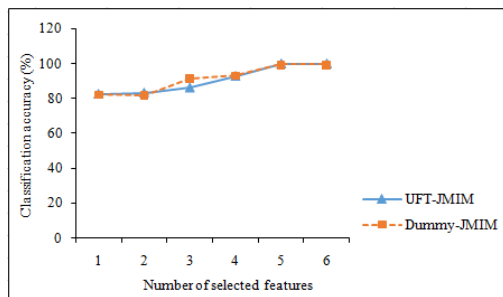


Figure 4. The classification accuracy that is achieved by the Acute Inflammations dataset

ch is created to represent an attribute with two or more different categories or levels. We use dummy variable as another way for the transformation of features by using defined variable. Dummy variable is used as a reference to ensure that the results of transformation from non-numerical into numerical features by using UFT methods does not have significant difference to the results of transformation of features using defined variable. So that, indicating that the original feature information is not lost.

Figures 2-6 show the classification accuracy of the five datasets. The classification accuracy is computed for the whole size of the selected subset (from 1 feature up to 20 features). Thus, all features of each dataset in this experiment was selected for each testing of k value (number of selected features). As shown in Figure 2, it illustrates the experiment with the acute inflammations dataset. UFT-JMIM achieves the highest average accuracy (100%) with 5 and 6 selected features, which is higher than the accuracy of Dummy-JMIM with 5 features (99.5%) and 6 features (99.4%).

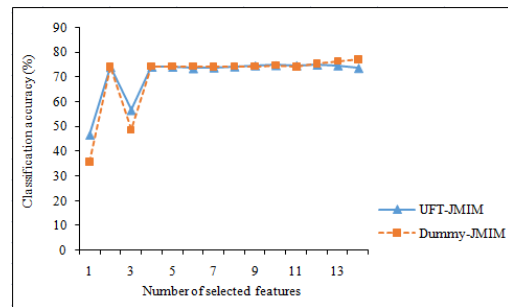


Figure 3. The classification accuracy that is achieved by the Adult dataset

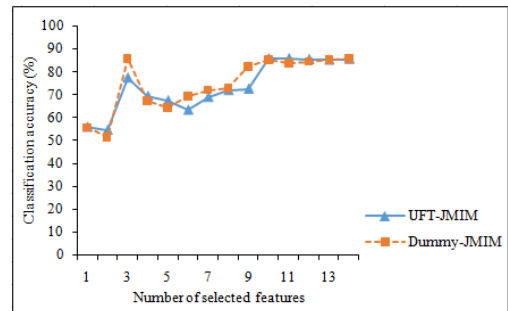


Figure 2. The classification accuracy that is achieved by the Australian Credit Approval dataset

In Figure 3 which illustrates the accuracy of the adult dataset, UFT-JMIM cannot achieve the highest classification accuracy. It can only achieve the 74.9% with 12 selected features. Meanwhile, Dummy-JMIM can achieve classification accuracy 77.1% with 14 features. Figure 4 shows the results for australian credit approval dataset. UFT-JMIM can achieve the highest classification result (85.83%) with 11 selected features. Dummy-JMIM can achieve the closest classification accuracy (85.80%) with 14 features.

The classification accuracy of UFT-JMIM for german credit data dataset is shown by Figure 5. It achieves 72.3% (15 selected features). Whereas, the classification accuracy produced by Dummy-JMIM can only achieve 70.6% as the best result with 1 selected feature. Figure 6 shows the UFT-JMIM performance for the hepatitis dataset which achieves the highest classification accuracy (89.3%) with 10 selected features. Meanwhile, Dummy-JMIM can only achieve 88.2% with 10 selected features.

UFT-JMIM can get better classification accuracy because of MI that used in UFT and JMIM methods. In UFT methods, MI preserves the information of data when transformation of features is conducted from non-numerical to numerical features. So when the data is transformed, MI minimizes information loss. For this case, the MI value between the transformed and the original features must be the same as the entropy of the ori-

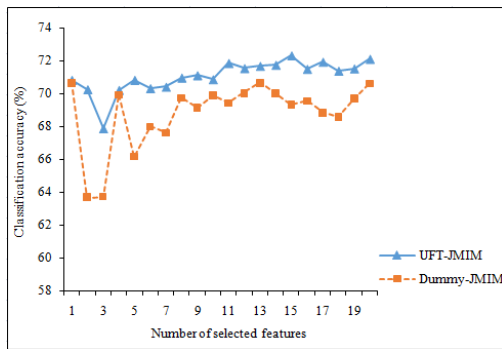


Figure 6. The classification accuracy that is achieved by the German Credit Data dataset

ginal features to preserve the original feature information.

In addition, MI applied in JMIM methods is used to measure of relevant and redundant features when select feature subset. It studies relevancy and redundancy, and takes into consideration the class label when calculating MI. In this methods, the candidate feature that maximises the cumulative summation of joint mutual information with features of the selected subset is chosen and added to the subset. JMIM methods employs joint mutual information and the ‘maximum of the minimum’ approach, which should choose the most relevant features. The features are selected by JMIM according to criterion as equation(7). In JMIM methods, the iterative forward greedy search algorithm is used to find the best combination of k features within subset. It causes the performance of finding to feature subset to be suboptimal because of high computation.

4. Conclusion

Feature selection based on MI using trans-formed features can reduce the redundancy of the selected feature subset, so that it can improve the accuracy of classification. The average classification accuracy for all experiments in this study that can be achieved by UFT-JMIM methods is about 84.47% and Dummy-JMIM methods is about 84.24%. This result shows that UFT-JMIM methods can minimize information loss between transformed and original features, and select feature subset to avoid redundant and irrelevant features.

For future work, further improvement can be made by studying to determine the best size k to find the relevant feature subset from heterogeneous features automatically in which it may make computation to be low.

References

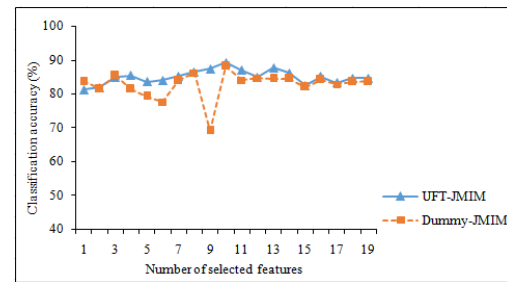


Figure 5. The classification accuracy that is achieved by the Hepatitis dataset

- [1] Bajwa, I., Naweed, M., Asif, M., & Hyder, S., “Feature based image classification by using principal component analysis,” *ICGST International Journal on Graphics Vision and Image Processing*, vol. 9, pp. 11–17. 2009.
- [2] Turk, M., & Pentland, A., “Eigenfaces for recognition,” *Journal of Cognitive Neuro-science*, vol. 3, pp. 72–86. 1991.
- [3] Tang, E. K., Suganthana, P. N., Yao, X., & Qina, A. K., “Linear dimensionality reduction using relevance weighted LDA,” *Pattern Recognition*, vol. 38, pp. 485–493. 2005.
- [4] Yu, H., & Yang, J., “A direct LDA algorithm for high-dimensional data with application to face recognition,” *Pattern Recognition*, vol. 34, pp. 2067–2070. 2001.
- [5] Bennasar, M., Hicks, Y., & Setchi, Rossitza, “Feature selection using Joint Mutual Information Maximisation,” *Expert Systems With Applications*, vol. 42, pp. 8520–8532. 2015.
- [6] Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L.A., *Feature extraction foundations and applications*, Springer Studies infuzzi-ness and soft computing, New York/Berlin, Heidelberg, 2006.
- [7] Saeys, Y., Inza, I., & Larrañaga, P., “A review of feature selection techniques in bioinformatics,” *Bioinformatics Advance Access*, vol. 23, pp. 2507–2517. 2007.
- [8] Battiti, R., “Using mutual information for selecting features in supervised neural net learning,” *IEEE Transactions on Neural Networks*, vol. 5, pp. 537–550. 1994.
- [9] Peng, H., Long, F., & Ding, C., “Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1226–1238. 2005.
- [10] Estévez, P.A., Tesmer, M., Perez, A., & Zurada, J. M., “Normalized mutual information feature selection,” *IEEE Transactions on Neural Networks*, vol. 20, pp. 189–201. 2009.

- [11] Hoque, N., Bhattacharyya, D. K., & Kalita, J. K., "MIFS-ND: a mutual information based feature selection method," *Expert Systems with Applications*, vol. 41, pp. 6371–6385. 2014.
- [12] Yang, H., & Moody, J., "Feature selection based on joint mutual information" *In Proceedings of international ICSC symposium on advances in intelligent data analysis*, pp. 22–25, 1999.
- [13] Wei, M., Chow, T. W.S., & Chan, R. H.M., "Heterogeneous feature subset selection using mutual information-based feature transformation," *Neurocomputing*, vol. 168, pp. 706-718. 2015.

IMPLEMENTATION OF SERIAL AND PARALLEL BUBBLE SORT ON FPGA

Dwi M J Purnomo¹, Ahmad Arinaldi¹, Dwi T Priyantini¹, Ari Wibisono¹, and Andreas Febrian²

¹Faculty of Computer Science, Universitas Indonesia, Kampus Baru UI, Depok, 16424, Indonesia

²Department of Engineering Education, Utah State University, 4160 Old Main Hill
Logan, Utah, 84322, United States of America

E-mail: dwimarhaendro@gmail.com, ari.w@cs.ui.ac.id

Abstract

Sorting is common process in computational world. Its utilization are on many fields from research to industry. There are many sorting algorithm in nowadays. One of the simplest yet powerful is bubble sort. In this study, bubble sort is implemented on FPGA. The implementation was taken on serial and parallel approach. Serial and parallel bubble sort then compared by means of its memory, execution time, and utility which comprises slices and LUTs. The experiments show that serial bubble sort required smaller memory as well as utility compared to parallel bubble sort. Meanwhile, parallel bubble sort performed faster than serial bubble sort to implement the algorithm on FPGA.

Keywords: *Sorting, bubble sort, serial bubble sort, parallel.*

Abstrak

Sorting adalah proses yang banyak dilakukan di dunia komputasi. Pemanfaatannya meliputi berbagai macam bidang, dari penelitian hingga industri. Dewasa ini terdapat banyak macam algoritma untuk *sorting*. Salah satu algoritma yang paling sederhana tapi cukup akurat adalah *bubble sort*. Pada studi ini, *bubble sort* diimplementasikan pada FPGA. Implementasi dilakukan pada pendekatan serial dan paralel. *Bubble sort* serial dan paralel dibandingkan penggunaan memori, waktu yang diperlukan untuk mengimplementasi, dan utilitas yang terdiri dari *slice* dan LUT. Eksperimen yang dilakukan menunjukkan bahwa *bubble sort* serial memerlukan lebih sedikit memori dan utilitas dibandingkan dengan *bubble sort* paralel. Sementara itu, *bubble sort* paralel memerlukan waktu lebih sedikit untuk mengimplementasikan algoritma di FPGA.

Kata Kunci: *Sorting, bubble sort, bubble sort serial, bubble sort parallel.*

1. Introduction

Sorting is a process that can be utilized in many applications. The applications vary from its origin computer science to other fields such as management, economic, etc. In computer science, sorting can be used to sort data either ascending or descending which is usually required in algorithm such as evolutionary algorithm. In management, one example is in risk management. The decision is taken based upon risk calculation. Before the decision is made the risk will be sorted to find the smallest. The number of data to be sorted are also vary from small number (e.g. less than 100) to large number of data depends on the application. There are also many variations of the algorithm in sorting process.

Several algorithms to undertake the sorting process are selection sort, merge sort, insertion sort, Heap sort, Radix sort, and bubble sort. Heap sort, Radix sort, and merge sort are powerful to sort large number of data [1]. Meanwhile the selection

sort, insertion sort and bubble sort are powerful for few data.

Selection and bubble sort are almost similar in term of the algorithm. The difference of those two algorithms lie on the array utilization in selection sort. In the selection sort, the sorting is based on the maximum value on each array, whereas in bubble sort each component is swapped one by one. Therefore it leads to the complexity different for both the algorithms. The complexity of bubble sort is $O(4n^2)$, whereas selection sort is $O(2n^2)$ [2]. Nevertheless, for small number of input, bubble sort is the simple but powerful algorithm compared to the others [3].

Bubble sort can be utilized to sort N numbers whether ascending or descending. Basically, bubble sort is undertaken in three main steps [4]. Firstly, the algorithm would step each input from the first to the last. The first position is occupied whether by smallest number or largest number, depend on the orientation of the sorting process (i.e. ascending

or descending). Secondly, the two adjacent input then would be compared. Finally, if the two adjacent inputs are in wrong order, the algorithm would swap it to the right order. The aforementioned steps would be repeated until no swap is required. Therefore, the final result would be numbers from the smallest to the largest for ascending or from the largest to the smallest for descending.

Research on bubble sort have been conducted in various application. Firstly, parallel bubble sort to utilize the concept in parallel computing [5]. Parallel computing means several calculations undertaken simultaneously [5]. Secondly study on bubble sort to assess the performance of visualization to promote the theory of understanding based on application rather than syntax [6]. Thirdly, bubble sort approach for channel routing [7]. Finally, study that compare serial and parallel computing on bubble sort with statistical bond [8]. This paper used statistical approach rather than mathematical approach. Therefore, it is represented in variant system rather than exact value like in mathematical approach [8].

Sorting is a basic process, hence it can be implemented in various platform. Field Programmable Gate Array (FPGA) is one of a desirable platform to implement sorting process. FPGA is still widely employed, especially in the industry. The advantage of using FPGA over another device is that in FPGA there will be no redundancy because the gate utilized in FPGA has not been prescribed yet [9]. Therefore, the utilization of resource is energetically effective [10]. Moreover, it is also mentioned that FPGA has considerable performance [10].

Sorting implementation on FPGA have been researched by many researchers. Srivasta et al. [10] proposed hybrid design for large scaling sorting on FPGA. Other research on high-speed parallel scheme for data sorting on FPGA. [11,12]. Parallel sorting which was conducted by Sogabe [13] and Martínez [14]. Last but not least is comparison study of many sorting algorithms covering parallel merge sort, parallel counting sort, and parallel bubble sort on FPGA [15].

The focus of this study is to implement bubble sort algorithm on FPGA. Bubble sort was chosen due to its simplicity and accuracy if the number of input data is fairly small. This study would contribute on two aspects. Firstly, the implementation of bubble sort on FPGA both serial and parallel approach. Secondly, the comprehensive comparison between serial and parallel approach on bubble sort algorithm implementation on FPGA.

The remainder of this paper is organized as follows: section two would elaborate bubble sort section three would explain the research method,

section four would shows the results and give some discussion, and the last section is conclusions.

2. Methods

Serial Bubble sort

Generally, serial bubble sort sequentially compares two number from leftmost to rightmost [3]. The order depends on whether it is ascending or descending. If it is ascending the largest number would be in the rightmost, otherwise would be in the leftmost. The schematic of serial bubble sort is delineated in Figure 1.

As shown in Figure 1, each consecutive step is undertaken by comparing two adjacent number. On the top left of the figure, two leftmost adjacent numbers are compared. In case of ascending the smaller number would be put in left side. On the contrary, in case of descending the larger number would be positioned in the left side.

The next step is to compare the next two numbers. Similar to the first step those number are compared and swapped. The same steps continue until the last number and called first stage. After all the number in the first stage has been compared, the rightmost of the number will be fixedly positioned. Therefore in the next stage, the aforementioned number would not be compared again.

The second stage then compared the 2 leftmost number again. It continues until one number before the last number similar to the first stage. In Figure 1 the fixedly last number is colored orange. Thus, it distinct which number can still be compared (blue), which cannot (orange).

The next stage until last stage is executed similarly, the only difference is that in the next stage, the rightmost number is reduced by one number as shown in Figure 1. The stage finish when all the

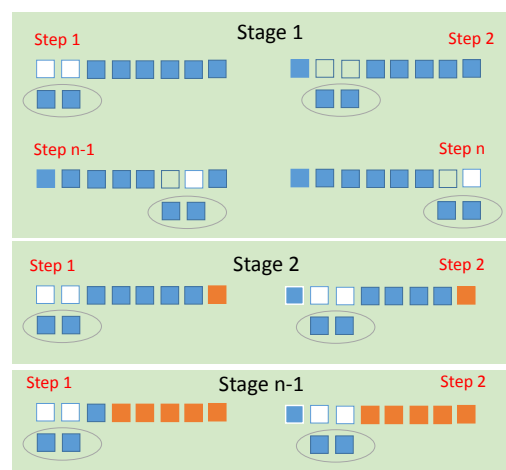


Figure. 1. Scheme of serial bubble sort.

number in each position has been compared to the number which position is right. Therefore, the position of each number will be prescribed based on the value of the number. The pseudo code for serial bubble sort is shown in algorithm 1.

As shown in the pseudo code, there are two loops in total bubble sort algorithm. The first loop is the top loop. The top loop define the stage of the algorithm. The number of stage is the number of data minus 1.

Meanwhile, the second loop is the bottom loop. It defines the step of the algorithm. The number of the step depends on the stage. It is number of data minus the undergoing stage.

Parallel Bubble sort

The idea of parallel bubble sort is to create parallel swapping. When in the serial bubble sort there is only 1 comparing process in parallel bubble sort there are $n/2$ (n is total input) comparing process. The schematic of one sorter is depicted in Figure 2.

As shown in Figure 2, there are n input data that would be sorted (i.e. $in_1, in_2, \dots, in(n)$). The first step is comparing each two adjacent numbers which is undertaken in swapper. After the first swap, the first and last number is located into the first and last output respectively (i.e. out_1 and out_n). Meanwhile, the other number after the first swap will be compared again with its adjacent number. However, the adjacent number is now changed. In the first swap process, the numbers to be compared

```

Algorithm 1: Serial Bubble Sort
1  % Initialization
2  Input_Data;
3  Number_of_Data;
4
5  for i in 1 to Number_of_Data-1 do
6    for j in 1 to Number_of_Data-i
7      do
8        compare Input_Data(i)with
9        Input_Data(i+1);
10       If Input_Data(i)>
11       Input_Data(i+1)then swap;
12     end for;
13   end for;

```

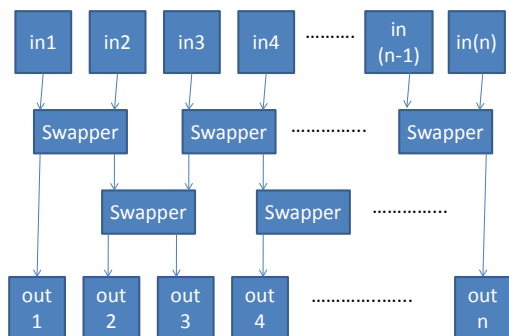


Figure 2. Scheme of steps in sorter in one stage.

are odd number followed by even number. Whereas in the second swap the numbers to be compared are even number followed by odd number.

Similar with the serial bubble sort, in parallel bubble sort there are also many stages. In each stage, contains several steps which has been previously elaborated. The schematic of the total stages in parallel bubble sort is shown in Figure 3. As shown in Figure 3, there are $n-1$ sorter. Each sorter would sort all the input (n input). Inside each sorter, there are two swappers as mentioned in the previous explanation.

Therefore, each number will compared to all the other numbers and placed in the right position. The pseudo code of parallel bubble sort is shown in Algorithm 2. As shown in the pseudo code, there are three loops in total bubble sort algorithm. The first loop is the top loop. The top loop define the stage of the algorithm. The number of stage is the number of data minus 1.

Meanwhile, the second and third loop is the bottom loop. It defines the step of the algorithm.

```

Algorithm 2: Parallel Bubble Sort
1  % Initialization
2  Input_Data;
3
4  for i in 1 to Number_of_Data-1
5    do
6      for i in 0 to
7        Number_of_Data/2-1 do
8        compare Input(2*i+1) with
9        Input(2*i+2);
10       If Input(2*i+1)>
11       Input(2*i+2) then swap;
12     end for;
13
14     for i in 1 to
15       Number_of_Data/2-1 do
16       compare Input(2*i) with
17       Input(2*i+1);
18       If Input(2*i)> Input(2*i+1)
19       then swap;
20     end for;
21   end for;

```

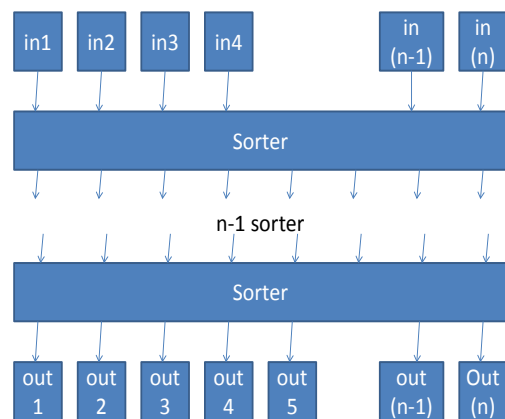


Figure 3. Scheme of stages in parallel bubble sort.

The second loop is the first swapper, whereas the third loop is the second swapper. The number of the step in the first swapper is half of number of data minus. Meanwhile, in the second swapper is half of number of data minus 1.

Serial Bubble Sort Implementation on FPGA

The implementation of serial bubble sort on FPGA contains two main steps. The steps are number representation and component implementation. This sub-section would rigorously discuss those steps.

Number Representation

In this study, the number is represented as 6 bit two's complement. The representation of the number is shown in Figure 4.

Component Implementation

There are four levels of component in serial bubble sort. Firstly, the top level is bubble sort itself which sort all the input into the right position. Secondly, swapper which swap two inputs if the condition is met. Thirdly, comparator which compare two inputs. Lastly, adder which was utilized to subtract two inputs to define the larger number in the comparator component.

Figure 5 shows one example of bubble sort component. In this example, only 8 inputs and outputs used. In the real experiment the number of input and output is varied. In this component the in-

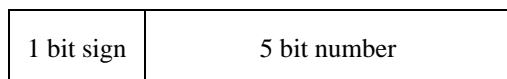


Figure 4. Number representation.

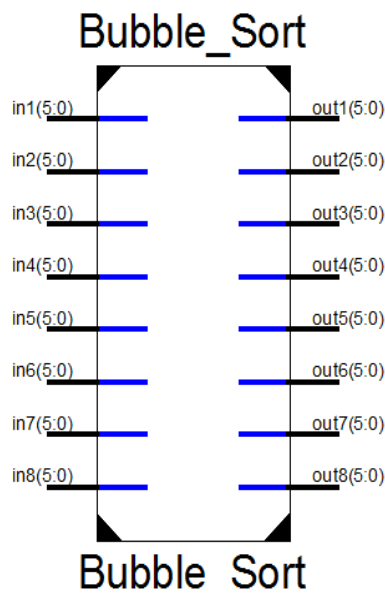


Figure 5. Bubble sort.

put would be ordered. The outputs are input that have been arranged whether ascending or descending. Inside this component, there are many swappers which is shown in Figure 6.

Figure 6 delineates the structure of the swapper which form top level bubble sort. In the first stage there are n-1 swappers i.e. to compare 2 adjacent numbers from 1 to n-1 with 1 increment. The next stage the number of swapper is reduced one each level. It means that the swapping process is reduced one step from the previous stage.

Inside swapper component there is a comparator component and multiplexer. The structure of swapper component is depicted in Figure 7. The comparator is delineated as one component, whereas the multiplexer is the combination of logic gates. The multiplexer is used to define the output after swapping process. Therefore, it has 2 outputs.

Inside comparator component there is an N bit adder. The structure of swapper component is depicted in Figure 8. The comparing process is undertaken by adding the first input with the negative of the second input. Finally, inside N bit adder there are 6 full adders which undertake the adding process.

Parallel Bubble Sort Implementation on FPGA

Similar to the serial bubble sort, the implementation of parallel bubble sort on FPGA contains two main steps. The steps are number representation and component implementation. The number representation in parallel bubble sort is similar with serial bubble sort. Therefore, this sub-section wou-

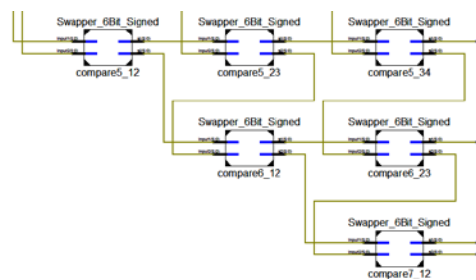


Figure 6. Structure of the Bubble Sort.

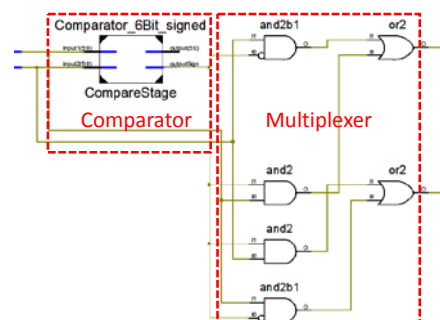


Figure 7. Structure of Swapper.

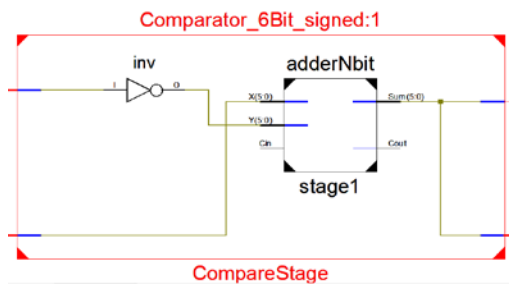


Figure 8. Structure of Comparator.

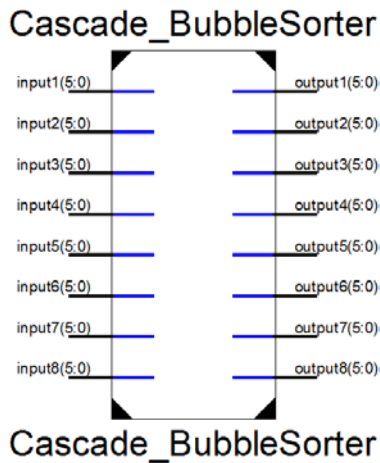


Figure 9. Top level sorter.

Id only sophisticatedly discuss the component implementation.

Component Implementation

There are five levels of component in parallel bubble sort. Firstly, top level sorter which sort all the input into the right position. Thirdly, bottom level sort which sort each two numbers. Thirdly, swapper which swap two inputs if the condition is met. Fourthly, comparator which compare two inputs. Lastly, adder which was utilized to subtract two inputs to define the larger number in the comparator component.

Figure 9 shows one example of top level sorter component. In this example, only 8 inputs and outputs used. In the real experiment the number of input and output is varied. In this component the input would be ordered. The outputs are input that have been arranged whether ascending or descending. Inside this component, there are many swappers which is shown in Figure 10.

Figure 10 delineates the structure of the bottom level sorter which form top level sorter. There are n-1 bottom level sorter to form top level sorter with n inputs and n outputs. The large view of the bottom level sorter is delineated in Figure 11.

Bottom level sorter consist of n inputs and n outputs (n = 8). Each bottom sorter represent the

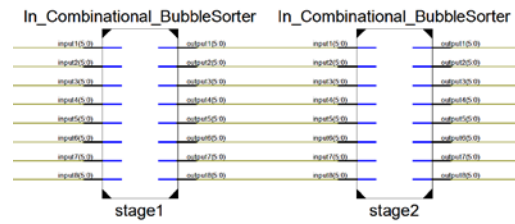


Figure 10. Structure of bottom level sorter.

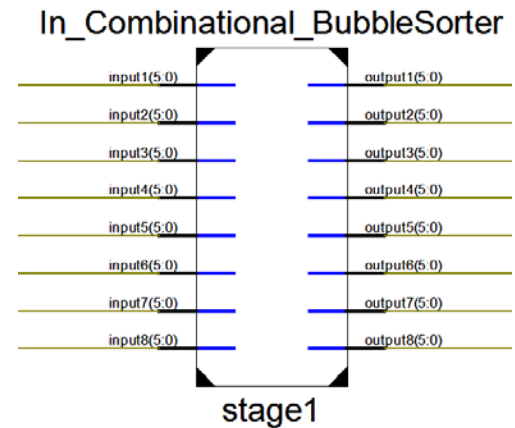


Figure 11. Bottom level sorter.

sorter in the block diagram which has previously been elucidated in Figure 3. Inside bottom level sorter there are n-1 swapper as shown in Figure 12.

The n input from top level sorter will be mapped in bottom level sorter. The inputs would enter the swapper and positioned as shown in Figure 12. If the order is wrong, then it would be swapped. Otherwise, no swapping would be undertaken. Inside swapper there is comparator in which have adder and multiplexer inside it. Swapper, comparator, and adder component in parallel bubble sort are similar with those in serial bubble sort.

Scenario

In this study, both serial and parallel bubble sort would be implemented on FPGA. The input would be varied from 4 to 40. Its increment is 2. Therefore, the input variations are 4, 6, 8, 10, 12, 14, ..., 40. The increment is defined to be two, because in parallel bubble sort it would be efficient if the number of input is even number.

The device utilized is Spartan 3A. Meanwhile the value of the inputs are varied from negative and positive. There are also similar number to test the performance of the sorter completely.

There are three main performance criteria employed in this study. The first criteria is execution time. The time required by the algorithm too implemented in the FPGA is calculated and compared.

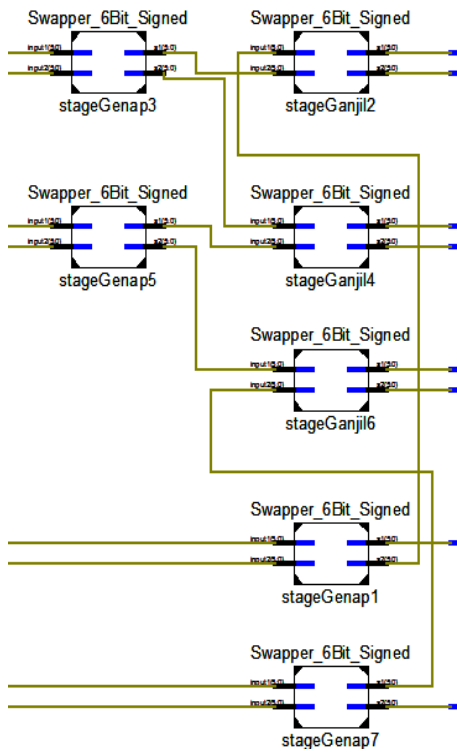


Figure 12. Structure of swapper in parallel bubble sorter.

The second criteria is memory. Memory required by the algorithm to operate is assessed and compared. Lastly, resources needed by the algorithm. Resources that compared cover slices and LUTs.

3. Results and Analysis

Serial Bubble Sort

From the experiments it is evident that serial bubble sort can be employed to sort the numbers. From 4 to 40 inputs, all variation could successfully be sorted by serial bubble sort. One example of 8 input is shown in Figure 13. The representation of the number before and after sorted are listed in Table 1.

The accuracy of serial bubble sorter is 100%. This is due to the fact that there is no number that is not compared by the others. Therefore, it covered the worst case.

To improve the system by means of its memory as well as its execution time, optimization procedure can be taken into account. By utilizing optimization process the unnecessary processes can be overlooked. Hence the time and resources can be reduced.

Parallel Bubble Sort

From the experiments it is apparent that parallel bubble sort can be utilized to sort the numbers. From 4 to 40 inputs could successfully be sorted by paral-

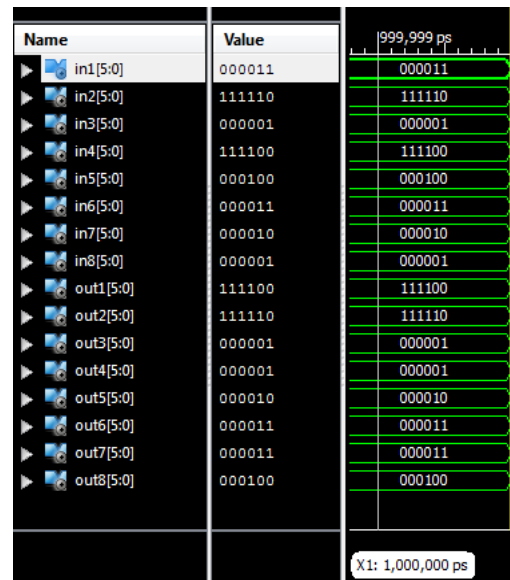


Figure 13. Result of 8 inputs serial bubble sorter.

TABLE 1
RESULT OF SERIAL BUBBLE SORT

Before Sorted		After Sorted	
Binary	Decimal	Binary	Decimal
000011	3	111100	-4
111110	-1	111110	-1
000001	1	000001	1
111100	-4	000001	1
000100	4	000010	2
000011	3	000011	3
000010	2	000011	3
000001	1	000100	4

lel bubble sort. One example of 8 input is shown in Figure 14. The representation of the number before and after sorted are listed in Table 2.

The accuracy of parallel bubble sorter is 100%. This is due to the fact that all number in the input is compared by the others. Therefore, it cope with the worst case.

To improve the system i.e. reducing memory as well as its execution time, optimization procedure can be empowered. By employing optimization process the unnecessary processes can be eradicated. Therefore, the time and resources utilization can be reduced.

Comparison of Serial and Parallel Bubble Sort

In this study, serial bubble sort and parallel bubble sort are compared. The comparison was focused on the execution time and resources. The execution time comparison is depicted in Figure 15. Meanwhile the resources by means its memory is delineated in Figure 16.

In execution time, parallel bubble sort perform better than serial bubble sort. Parallel bubble sort can sort the input faster than serial bubble sort.

TABLE 2
RESULT OF PARALLEL BUBBLE SORT

Before Sorted		After Sorted	
Binary	Decimal	Binary	Decimal
111110	-2	111100	-4
111111	-1	111101	-3
111101	-3	111110	-2
111100	-4	111111	-1
000100	4	000001	1
000011	3	000010	2
000010	2	000011	3
000001	1	000100	4

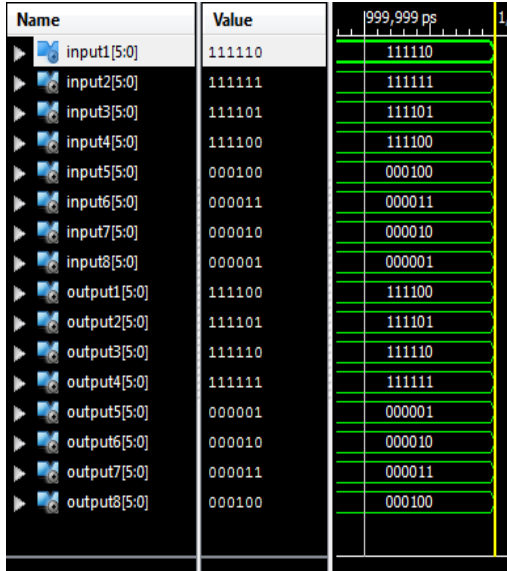


Figure 14. Result of 8 inputs parallel bubble sorter.

TABLE 3
MEMORY AND TIME COMPARISON BETWEEN SERIAL AND PARALLEL BUBBLE SORT

Number of Input	Memory (kB)		Time(s)	
	Serial	Parallel	Serial	Parallel
4	288288	301104	49.18	17.7
8	301472	326896	50.34	26.42
12	331936	376496	76.35	25.81
14	349344	443696	91.27	32.05
16	353184	482608	106.68	40.84
18	378784	529712	129.84	53.52
20	388640	534448	136.52	70.39
22	417440	590896	191.17	94.22
24	450080	656304	201.25	135.36
26	483872	725488	266.02	206.48
28	522336	802160	365.61	276.3
30	561248	883824	333.17	370.87
32	603552	967088	415.3	422.21
34	651808	1068848	497.61	535.47
36	701536	1166000	711.03	634.98
38	754336	1242144	918.59	634.12
40	808224	1351264	1065.18	904.92

This is due to the fact that several process in parallel bubble sort can be executed parallel. Therefore, it does not have to wait the previous process done to be executed. Meanwhile, in serial bubble sort, the next process would be undertaken after the previous process have finished. Even though the more process executed at the same time would cause the

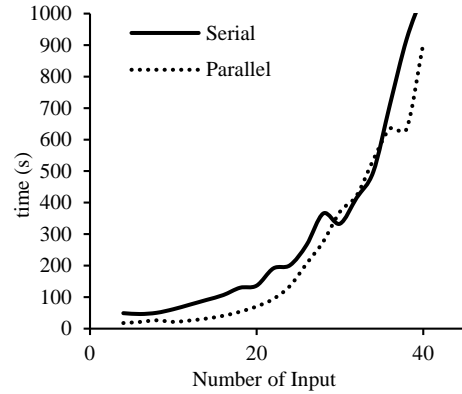


Figure 15. Execution time comparison between serial and parallel bubble sort.

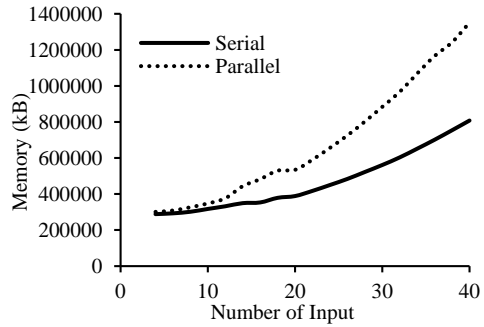


Figure 16. Memory comparison between serial and parallel bubble sort.

system take longer time to finish it, the result shows that the compensation does not affect the final result.

In terms of memory, parallel bubble sort required memory larger than serial bubble sort. Parallel bubble sort required larger memory because it undertook several processes at the same time. Therefore, each process would take larger capacity of memory.

This phenomenon is the compensation of faster process in parallel bubble sort. Serial bubble sort can be employed with limited memory capacity, whereas parallel bubble sort can operate faster. The detail comparison between serial bubble sort and parallel bubble sort are listed Table 3 for memory and time comparison.

4. Conclusion

In this paper, bubble sort has been successfully implemented on FPGA. Serial bubble sort required smaller memory as well as utilities, i.e. slices and LUTs compared to parallel bubble sort. Meanwhile, parallel bubble sort is faster than serial bubble sort to undertake the processes.

References

- [1] R. Abirami, "VHDL Implementation of Merge Sort Algorithm" *International Journal of Computer Science and Communication Engineering*, Vol. 3, No. 2, pp. 15-18, 2014.
- [2] R. Edjlal, A. Edjlal, and T. Moradi, "A Sort Implementation Comparing with Bubble Sort and Selection Sort" *In 3rd International Conference on Computer Research and Development*, pp. 380-381, 2011.
- [3] W. Min, "Analysis on Bubble Sort Algorithm Optimization" *In International Forum on Information Technology and Applications*, pp. 208-211, 2010.
- [4] H. Sutopo, "Multimedia Based Instructional Development: Bubble Sort Visualization" *In 3rd International Conference on Software Engineering and Service Science*, pp. 791-794, 2015.
- [5] R. Rashidy, S. Yousefpour, and M. Koochi, "Parallel Bubble Sort Using Stream Programming Paradigm" *In 5th International Conference on Application of Information and Communication Technologies*, pp. 1-5, 2011.
- [6] P. Bellström and C. Thorén, "Learning How to Program through Visualization: A Pilot Study on the Bubble Sort Algorithm" *In 2nd International Conference Applications of Digital Information and Web Technologies*, pp. 90-94, 2009.
- [7] S. S. Chen, C. H. Yang, and S. J. Chen, "Bubble-Sort Approach to Channel Routing" *In IEE Proceedings - Computers and Digital Techniques*, pp. 415-422, 2000.
- [8] S. K. Panigrahi, S. Chakraborty, and J. Mishra "Statistical Bound of Bubble Sort Algorithm in Serial and Parallel Computations" *In Electrical, Electronics, Signals, Communication and Optimization*, pp. 1-6, 2015.
- [9] D. M. J. Purnomo, M. R. Alhamidi, A. Wibisono, and M. I. Tawakal, "Investigation of Flip-Flop Performance on Different Type and Architecture is Shift Register with Parallel Load Applications" *Jurnal Ilmu Komputer dan Informasi*, pp. 87-95, 2015.
- [10] A. Srivastava, R. Chen, V. K. Prasanna, and C. Chelmis, "A Hybrid Design for High Performance Large-scale Sorting on FPGA" *In ReConfigurable Computing and FPGAs*, pp. 1-6, 2015.
- [11] S. Dong, X. Wang, and X. Wang, "A Novel High-Speed Parallel Scheme for Data Sorting Algorithm Based on FPGA" *In 2nd International Congress on Image and Signal Processing*, pp. 1-4, 2009.
- [12] F. A. Alquaied and M. A. AlShaya, "A Novel High-Speed Parallel Sorting Algorithm Based on FPGA" *In Saudi International Electronics, Communications and Photonics Conference*, pp. 1-4, 2011.
- [13] Y. Sogabe and T. Maruyama, "FPGA Acceleration of Short Read Mapping based on Sort and Parallel Comparison" *In 24th International Conference on Field Programmable Logic and Applications*, pp. 1-4, 2014.
- [14] J. Martínez, R. Cumplido, and C. Feregrino, "An FPGA-based Parallel Sorting Architecture for the Burrows Wheeler Transform" *In International Conference on Reconfigurable Computing and FPGAs*, pp. 7-14, 2005.
- [15] S. Bique, W. Anderson, M. Lanzagorta, and R. Rosenberg, "Sorting Using the Xilinx Virtex-4 Field Programmable Gate Arrays on the Cray XD1" *In CUG 2008 Proceedings*, pp. 1-12, 2008.